Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077521

ARTICLE

# Prototype Memory and Contrastive Learning Based Unsupervised Anomaly Detection for Time Series

**Xi Li**[1], **Yingjie Chang**[1], **Peng Chen**[1,*], **Ang Bian**[1] **and Ning Lu**[1,2,*]

[1]School of Computer and Software Engineering, Xihua University, Chengdu, China
[2]School of Art and Design, Xihua University, Chengdu, China

*Corresponding Authors: Peng Chen. Email: chenpeng@mail.xhu.edu.cn; Ning Lu. Email: luning@mail.xhu.edu.cn

Received: 11 December 2025; Accepted: 11 March 2026; Published: 08 May 2026

**ABSTRACT:** Multivariate time series anomaly detection (MTSAD) is a critical task for real-time risk control and fault diagnosis in industrial monitoring, aerospace, and financial domains. Unsupervised MTSAD confronts three core challenges: label scarcity in practical scenarios, diverse anomaly patterns that demand adaptive modeling, and weak feature discriminability between normal and anomalous samples. To address these challenges, we propose a Prototype Memory and Contrastive Learning Based Unsupervised Anomaly Detection for Multivariate Time Series method named PC-UAD. PC-UAD comprises three core modules with hierarchical functionalities: (1) A Temporal PatchEmbedder, which adopts learnable positional encoding for dynamic temporal representation and incorporates channel projection to model adaptive cross-sensor dependencies in multivariate data; (2) A Prototype Memory Encoder, which embeds a prototype attention mechanism to explicitly memorize typical normal patterns, forming a "normal pattern dictionary" that enhances the model's perception of normal behavioral boundaries; (3) A ContrastFusion module, which leverages contrastive learning to amplify feature distribution discrepancies between normal and anomalous data, strengthening the model's ability to distinguish subtle anomalies. Experiments on five public multivariate time series datasets demonstrate that our method achieves superior detection accuracy compared to eight state-of-the-art approaches, with the average F1-score and ROC-AUC both ranking first.

**KEYWORDS:** Multivariate time series; unsupervised anomaly detection; prototype memory; contrastive learning; transformer

## 1 Introduction

Multivariate time series data, characterized by its temporal continuity and sequential dependence, permeates nearly every corner of modern society—from industrial sensor readings monitoring manufacturing equipment health, to financial transaction records tracking market fluctuations, and healthcare vital signs reflecting patient physiological states. Anomalies in such data, often manifesting as unexpected deviations from normal patterns, frequently signal critical events: a sudden spike in a turbine's vibration amplitude may precede mechanical failure, an abnormal sequence of financial transactions could indicate fraudulent activity, and erratic changes in a patient's heart rate might warn of life-threatening conditions. Consequently, multivariate time series anomaly detection (MTSAD), the task of identifying these rare yet high-impact deviations automatically, has become an indispensable cornerstone in fields requiring real-time risk control, fault diagnosis, and situational awareness.

Despite decades of research, unsupervised MTSAD remains a formidable challenge, primarily due to three inherent characteristics of real-world scenarios. First, label scarcity plagues most practical applications: anomalies are inherently rare, and manually annotating them requires domain expertise and enormous labor costs, making supervised learning paradigms impractical. Second, diversity of anomaly patterns complicates detection—anomalies may present as instantaneous spikes, gradual drifts, cross-period deviations, or low-amplitude mutations, each demanding different temporal modeling capabilities. Third, complex data properties, such as non-stationarity, high dimensionality and noise interference, further obscure the boundary between normal and anomalous behaviors, placing strict demands on the robustness and generalization of detection models.

To address these challenges, researchers have proposed a spectrum of unsupervised and self-supervised methods, yet existing approaches still suffer from notable limitations that hinder their performance in complex real-world scenarios. Reconstruction-based methods (e.g., AutoEncoders, GANs) rely on the "reconstruction assumption"—that models trained on normal data will fail to reconstruct anomalies—but this assumption often collapses in high-capacity models, which may overgeneralize and reconstruct anomalies effectively. Prediction-based methods (e.g., LSTM, TCN) use future prediction errors as anomaly indicators, but their ability to capture long-range or slow-evolving anomalies is constrained by fixed window sizes and limited temporal dependency modeling. Transformer-based methods, while excelling at global context modeling, suffer from high computational complexity and lack explicit memory mechanisms for "normal pattern prototypes," limiting their performance in sample-scarce industrial scenarios. Memory-augmented methods introduce external memory to memorize normal patterns, but they often lack active mechanisms to enhance the discriminability between normal and anomalous features, leading to suboptimal detection accuracy for subtle anomalies. Recent advances, such as diffusion models with self-conditioning guidance [1], adaptive bottleneck-based frameworks with dual adversarial decoders [2], and reconstruction trend-focused Transformer networks [3], have explored novel paradigms for representation learning and anomaly separation.

Contrastive learning has emerged as a powerful self-supervised framework for time series analysis, enhancing feature discriminability between normal and anomalous data by optimizing pairwise sample similarity. However, its reliance on pairwise comparisons without anchoring to stable "normal patterns" leaves it vulnerable to noise and subtle anomalies. In contrast, memory-augmented methods inherently leverage "pattern memory" to establish a stable reference for normal patterns, thereby effectively mitigating overgeneralization. Yet, these approaches typically lack explicit mechanisms to widen the discriminative margin between normal and anomalous samples in the feature space. The necessity of combining these two paradigms is thus compelling: pattern memory anchors the model's understanding of normal behavior, while contrastive learning strengthens anomaly discriminability.

To bridge these gaps, this paper proposes PC-UAD, a novel Transformer-based framework that integrates learnable prototype memory and contrastive fusion to address the core challenges of MTSAD. The key innovations and contributions of this work are summarized as follows:

(1) Enhanced Temporal Feature Extraction: A Temporal PatchEmbedder replaces fixed sinusoidal positional encoding with learnable positional encodings and adds a channel projection layer, enabling adaptive position representation learning and explicit modeling of cross-sensor dependencies in multivariate time series.

(2) Explicit Normal Pattern Memorization: A Prototype Memory Encoder embeds learnable prototype matrices into the Transformer's multi-head attention, forming a "normal pattern dictionary" that fuses local patch features with global typical patterns, ensuring normal samples cluster tightly around prototypes while anomalies remain distinguishable.

(3) Active Feature Discrimination: A ContrastFusion module generates negative samples via Gaussian noise injection and adopts asymmetric contrastive loss and denoising loss to actively expand feature distribution distances between normal and anomalous data, polarizing latent-space representations.
(4) Comprehensive Validation: PC-UAD is evaluated on five real-world multivariate time series benchmark datasets covering aerospace, space station, internet service, water treatment, and micro-service scenarios. Experimental results confirm its superiority over state-of-the-art baselines in F1-score and ROC-AUC, verifying robustness across diverse anomaly patterns and data types.

In response to the three core challenges of unsupervised MTSAD outlined earlier: Label scarcity is addressed by the Prototype Memory Encoder, which memorizes typical normal patterns from unlabeled data to form a "normal pattern dictionary" and eliminates reliance on annotated anomalies; diverse anomaly patterns are tackled by the Temporal PatchEmbedder, whose learnable positional encodings and multi-scale patch modeling adaptively capture dynamic temporal dependencies and cross-sensor correlations; weak feature discriminability is resolved by the ContrastFusion Module, which leverages asymmetric contrastive loss to expand feature distribution gaps between normal and anomalous samples, enhancing the model's ability to distinguish subtle anomalies.

The remainder of this paper is structured as follows: Section 2 reviews related work on time series anomaly detection. Section 3 details the architectural design and mathematical formulation of PC-UAD. Section 4 presents comprehensive experiments, including performance comparisons, ablation studies, and sensitivity analyses. Section 5 concludes the paper and discusses future directions.

## 2 Related Work

In the past five years, unsupervised and self-supervised MTSAD methods have developed rapidly, with innovations focusing on model architecture optimization, representation learning enhancement, and multi-mechanism fusion.

### *2.1 Reconstruction-Based Methods*

Reconstruction-based methods are traditional and mainstream unsupervised MTSAD paradigms, whose core assumption is that models trained on normal data can accurately reconstruct normal samples but fail to reconstruct anomalous samples, with anomaly scores determined by reconstruction errors. In recent years, researchers have optimized this framework to address the overgeneralization problem of high-capacity models and the asynchronous correlation modeling of multivariate time series.

Dai and wang [4] proposed a hash memory network-enhanced autoencoder, which stores normal pattern features in hash-coded memory units and constrains the decoder to retrieve memory for reconstruction, reducing the model's ability to reconstruct anomalies. For microservice fault diagnosis scenarios, Li et al. proposed a parallel convolutional anomaly multi-classification model based on reconstruction principles, which achieves effective anomaly localization while ensuring detection accuracy [5].

However, reconstruction-based methods still face inherent bottlenecks: high-capacity models can still reconstruct subtle anomalies, leading to reduced detection accuracy; meanwhile, their ability to model long-range and multi-periodic temporal dependencies is limited by network depth and receptive field.

### *2.2 Transformer-Based Methods*

The Transformer architecture has become a research hotspot in MTSAD due to its strong global temporal dependency modeling capability, especially for long-sequence and multivariate time series.

In recent years, related work has focused on optimizing attention mechanisms and reducing computational complexity.

Anomaly Transformer [6] pioneered an association discrepancy-based anomaly attention mechanism, quantifying anomaly degrees by measuring attention weight inconsistency across positions and achieving state-of-the-art performance on MSL and PSM datasets. TranAD [7] proposed a two-stage self-conditioning inference framework for MTSAD: the first stage generates preliminary reconstructions and computes focus scores, while the second stage amplifies signals in high-error regions, and integrates adversarial training to enhance generalization under limited data. TimesNet [8] modeled time-series two-dimensional variations via temporal 2D convolution, providing an efficient feature extraction scheme for Transformer-based anomaly detection models.

Despite their advantages, Transformer-based methods have two critical limitations: vanilla self-attention incurs $O(T^2)$ computational complexity, making edge deployment infeasible; and the lack of explicit normal-pattern memory modules limits performance in sample-scarce industrial scenarios.

### *2.3 Memory-Augmented Methods*

To address the overgeneralization problem of reconstruction and Transformer models, memory-augmented methods use external memory units or learnable prototypes to explicitly memorize normal patterns, forming a "normal pattern dictionary" to enhance anomaly discriminability.

Li et al. [9] proposed a prototype-oriented unsupervised method for MTSAD, which clusters normal features into prototype centers and calculates anomaly scores via feature-prototype distance, enabling effective detection of slow-evolving anomalies. Dai and Wang [4] integrated hash memory networks into autoencoders, using hash coding to store normal sub-patterns and retrieving memory during decoding to constrain reconstruction, significantly improving anomaly-normality distinguishability. In cloud application performance diagnosis, Xin et al. [10] reviewed trustworthy AI-based systems, noting that memory-augmented models are widely used for their ability to retain historical normal patterns, but require further optimization in real-time performance.

The main limitation of existing memory-augmented methods is the lack of active feature discrimination mechanisms: they can memorize normal patterns but cannot actively expand the feature distance between normal and anomalous samples, leading to low detection accuracy for subtle anomalies.

### *2.4 Contrastive Learning-Based Methods*

Contrastive learning has brought breakthroughs to self-supervised anomaly detection in recent years, which enhances feature discriminability by constructing positive and negative sample pairs, and has become an important direction to solve the problem of poor feature separability in traditional methods.

Xiao et al. [11] explored dual-view graph structures for multivariate time series analysis, integrating graph and hypergraph representations into contrastive learning, then designed a cross-view contrastive loss to align and distinguish feature representations from both views. Xiao et al. [12] proposed the MulGad model, which introduces multi-granularity contrastive learning for MTSAD, extracting fine and coarse-grained temporal features to capture complex anomaly patterns and enhance the model's perception of diverse anomalies via cross-granularity sample pair construction and contrastive optimization.

The core challenge of contrastive methods lies in the construction of high-quality positive and negative pairs: improper pair design will lead to feature collapse or overfitting; at the same time, most methods lack explicit normal pattern memory mechanisms, resulting in a disconnect between feature discriminability and normal pattern memorization.

### *2.5 Research Gaps and Motivation of this Paper*

Comprehensive analysis of existing methods reveals three core research gaps in current MTSAD: (1) Overgeneralization of reconstruction/Transformer models: High-capacity models can reconstruct subtle anomalies, and the lack of explicit normal pattern constraints reduces detection accuracy. (2) Disconnect between memory and contrastive mechanisms: Memory-augmented methods lack active feature discrimination, while contrastive learning methods lack normal pattern memorization, failing to balance "pattern memorization" and "feature separation". (3) Insufficient modeling of multi-scale temporal features: Most methods focus on single-scale temporal dependencies and ignore the hybrid characteristics of real-world anomalies.

To address these gaps, this paper proposes PC-UAD, which integrates learnable prototype memory and contrastive fusion, achieving explicit normal pattern memorization and active feature discrimination, while enhancing multi-scale temporal feature extraction via learnable positional encoding and patch embedding. The framework draws on cross-domain research insights [6] to ensure robustness across diverse real-world scenarios.

## 3 Methods

For a MTSAD task, assume an industrial system with $d$ sensors, where observations over.

$T$ time steps form a time series $X = (x_1, x_2, \ldots, x_T)$, with $x_T \in \mathbb{R}^d$ denoting the $d$-dimensional measurement vector at time $t$. The goal is to predict an anomaly label sequence $y \in \{0, 1\}^T$, where $y_i = 1$ indicates an anomaly at time $i$, and $y_i = 0$ denotes a normal state.

### *3.1 Overview*

PC-UAD consists of three core modules: the Temporal PatchEmbedder, the Prototype Memory Encoder, and the ContrastFusion Module. These modules respectively undertake the tasks of feature extraction, normal pattern memorization, and anomaly discrimination, forming a complete technical chain to address the core challenges of MTSAD. As illustrated in Fig. 1, the Temporal PatchEmbedder first processes raw time series to generate structured patch features; the Prototype Memory Encoder then fuses these patch features with learnable normal prototypes to establish an explicit "normal pattern reference"; finally, the ContrastFusion Module enhances the separability of normal and anomalous features through contrastive learning. Specifically, the Temporal PatchEmbedder extracts patch-based time tokens from the input series. In each layer of the Prototype Memory Encoder, an attention mechanism integrates patch tokens from the previous layer to capture interdependencies between temporal features and prototype features. A single linear layer reconstructs the original features and computes anomaly scores based on reconstruction errors and contrastive similarity. The ContrastFusion module further optimizes feature distributions, thereby improving overall detection performance.

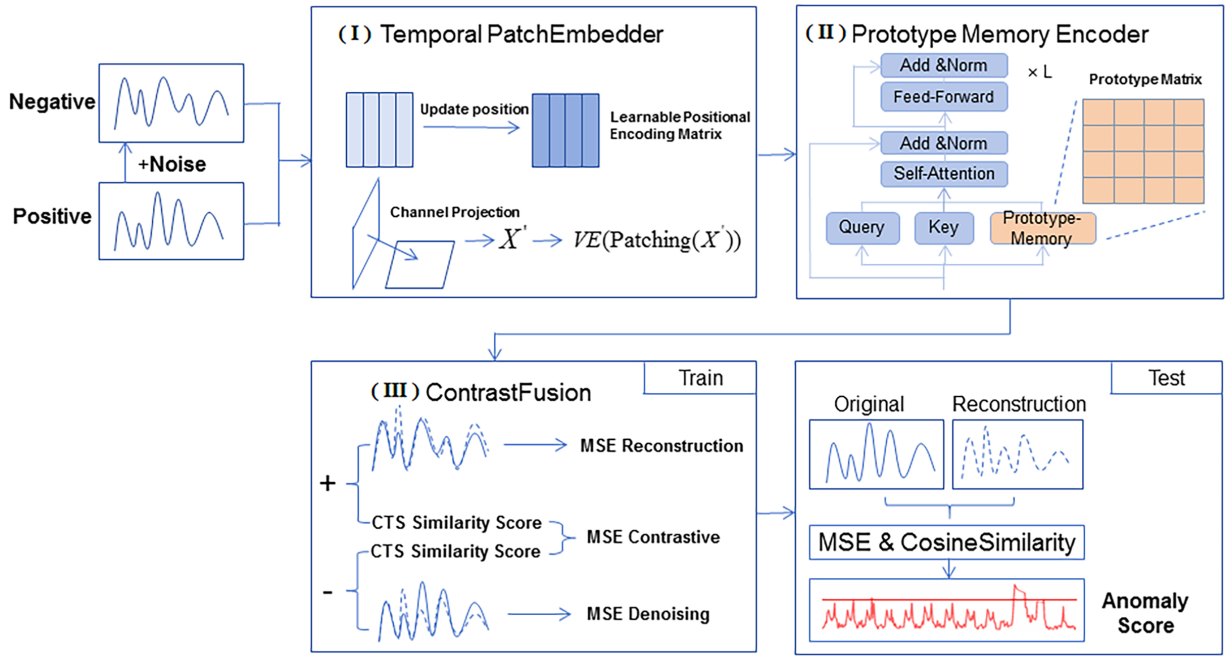


**Figure 1:** The overall architecture of PC-UAD. The framework integrates three key components: (**I**) Temporal PatchEmbedder with learnable positional encodings, (**II**) Prototype Memory Encoder for normal-pattern memorization, (**III**) ContrastFusion module for enhancing normal-anomaly discriminability.

### *3.2 Temporal PatchEmbedder*

To address the limitations of fixed positional encoding in adapting to dynamic temporal characteristics and the need to model cross-sensor correlations, the Temporal PatchEmbedder realizes adaptive feature extraction through three key steps: data normalization, enhanced positional embedding, and patch segmentation projection. In this module, we replace the classical Transformer positional encoding with a set of learnable positional encodings. After incorporating these encodings, a linear layer is applied to project the features at each time step across all channels, enabling the model to capture interdependencies (Fig. 2).

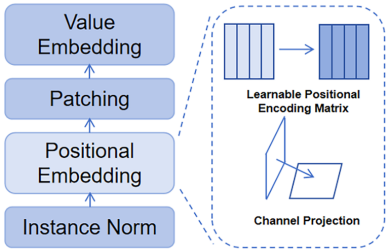


**Figure 2:** Structure of temporal PatchEmbedder.

(1) Data Normalization: Raw time series $X$ is processed with Reversible Instance Normalization [13] to mitigate distribution shift. This operation eliminates scale differences between sensor dimensions while retaining the original temporal trends.

(2) Enhanced Positional Embedding: Learnable positional encodings (instead of fixed sinusoidal offsets) are adopted, which evolve into trainable parameters to adaptively learn optimal position representations. A linear transformation layer further models cross-channel relationship, achieving deep fusion of temporal position information and multi-sensor feature information.

(3) Patch Segmentation and Projection: The normalized sequence $x_T \in \mathbb{R}^{T\times C}$ (where $C$ is feature dimension) is segmented along the temporal dimension into $N$ patches of size $P$, forming $N' \in \mathbb{R}^{N\times P\times C}$. This segmentation converts long time series into structured patch features, reducing computational complexity while preserving local temporal correlations. Patches are reshaped to $N \times (P \cdot C)$ and projected via a linear layer $VE$ into a unified latent space of dimension $N \times D$ (where $D$ represents the model dimension after projection), enabling processing of long temporal windows. The computational procedures are formalized as:

$$X' = PE\left(RevIN\left(X\right)\right) \tag{1}$$

$$N = VE\left(Patching\left(X'\right)\right) \tag{2}$$

### *3.3 Prototype Memory Encoder*

Building on the patch features output by the Temporal PatchEmbedder, the Prototype Memory Encoder embeds learnable prototypes into the Transformer attention mechanism to explicitly memorize normal patterns, addressing the overgeneralization problem of traditional models. This module forms the Prototype Memory Encoder (Fig. 3), which aligns with memory-augmented autoencoders [4] and prototype-based detection methods for multivariate time series, aiming to: (1) adaptively fuse patch information with global typical patterns learned during training; and (2) provide a compact, learnable "normal pattern dictionary," enabling normal samples to cluster within the prototype space while anomalous samples struggling to match with these prototypes.

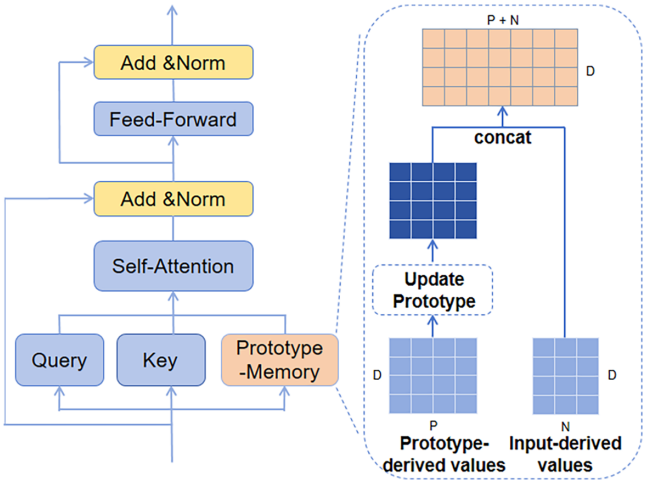


**Figure 3:** Structure of prototype memory encoder.

The module's backbone is a multi-layer Transformer encoder, with a learnable prototype matrix $G \in \mathbb{R}^{M\times D}$ (where $M$ is prototype count and $D$ is the model dimension) integrated into each layer's multi-head self-attention. The prototype matrix $prototype \in \mathbb{R}^{M\times D}$ is initialized using a zero-mean normal distribution with standard deviation 0.02, a standard initialization strategy for Transformer-based models to ensure stable training. It is registered as a trainable parameter via torch.nn.Parameter(), allowing end-to-end optimization through backpropagation to minimize normal-sample reconstruction loss, extracting vectors

that characterize typical normal patterns. The learnable prototype matrix is optimized only by minimizing the reconstruction loss of normal samples, with its update coupled to the Transformer's layer-wise feature learning for regularization. This design constrains prototypes to capture universal normal patterns of the data, effectively avoiding overfitting to local noise.

Unlike conventional attention, values ($V$) are computed separately for input sequences and prototypes, then concatenated into an extended value matrix $V \in \mathbb{R}^{(N+M)\times D}$. Attention scores are also extended to model interactions between input sequences and prototypes, enabling each token to fuse local contextual information with global prototype knowledge.

For the $i$ layer encoder, with input $X^{(i)} \in \mathbb{R}^{N\times D}$ (where the input to the first layer $X^{(1)}$ is the output of the Temporal PatchEmbedder), the query and key for the $K$ attention head are defined as $Q_k^{(i)} = X^{(i)} W_k^Q$ and $K_k^{(i)} = X^{(i)} W_k^K$, with $W_k^Q$, $W_k^K \in \mathbb{R}^{D\times D}$. Values ($V$) are split into two components:

(1) Input-derived values: Consistent with the standard Transformer computation, $\mathrm{V}^{\mathrm{full}} = \mathrm{X}^{(\mathrm{i})}\mathrm{W}_{\mathrm{k}}^{\mathrm{V}}$, where $\mathrm{W}_{\mathrm{k}}^{\mathrm{V}} \in \mathbb{R}^{D\times D}$. For multi-head attention, $D$ is split into $H$ heads (each with dimension $d = D/H$) and the $k$-th head's input value is $V_{orig}^{k(i)} = Rearrange(V^{full}, (N, H\cdot d) \to (H, N, d)$.

(2) Prototype-derived values: Obtained via linear transformation, $V_{proto}^{full} = GW_k^V$, where $V_{proto}^{full} \in \mathbb{R}^{M\times D}$. The $k$-th head's prototype value is $V_{proto}^{k} = Rearrange(V_{proto}^{full}) \in \mathbb{R}^{M\times d}$.

Concatenated the above values to for the final $k$-th head is:

$$V_k^{(i)} = \left[V_{proto}^{k(i)}, V_{orig}^{k(i)}\right] \in \mathbb{R}^{(N+M)\times d} \tag{3}$$

Unlike conventional attention score, our method includes two attention modes: intra-sequence and sequence-prototype attention. First, the prototype key matrix is computed as: $K_{proto,k}^{(i)} = G^{(i)} W_k^{K_{proto,}(i)} \in \mathbb{R}^{M\times d}$. Then, the two types of attention scores are calculated separately:

(1) Intra-sequence attention: Measure correlations between different positions in the input sequence.

$$AttnScores_{seq}^{(i)} = \frac{Q_k^{(i)}(K_k^{(i)})^\top}{\sqrt{d}} \in \mathbb{R}^{N\times N} \tag{4}$$

(2) Sequence-prototype attention: Measures correlations between the input sequence and global prototypes.

$$AttnScores_{proto}^{(i)} = \frac{Q_k^{(i)}(K_{proto,k}^{(i)})^\top}{\sqrt{d}} \in \mathbb{R}^{N\times M} \tag{5}$$

Scores are concatenated and normalized via Softmax function, then multiplied by the value matrix to generate the attention output:

$$AttnScores^{(i)} = \left[AttnScores_{seq}^{(i)}, AttnScores_{proto}^{(i)}\right] \in \mathbb{R}^{N\times(N+M)} \tag{6}$$

$$Z_k^{(i)} = Softmax\left(AttnScores^{(i)}\right)\cdot V_k^{(i)} \tag{7}$$

At the end of each layer, a linear layer aggregates output features $Z_k^{(i)}$ from all attention heads to obtain $Z^{(i)}$. After layer normalization, a Feed-Forward Network (FFN) applies transformation and another layer normalization. The result is applied to produce the next layer's input:

$$X^{(i)'} = LayerNorm\left(X^{(i)} + Z^{(i)}\right) \tag{8}$$

$$X^{(i+1)} = LayerNorm\left(X^{(i)'} + FFN\left(X^{(i)'}\right)\right) \tag{9}$$

The Prototype Memory Encoder strengthens normal-pattern memorization and global context modeling, improving detection performance and generalization; meanwhile, during training, the model is optimized to fuse local patch features with global prototypes, forcing normal data to cluster tightly around prototypes. High-capacity models are thus constrained by prototype-based regularization and cannot freely reconstruct anomalies that deviate from normal patterns.

### *3.4 ContrastFusion Module*

While the Prototype Memory Encoder establishes normal pattern references, it lacks active mechanisms to enhance feature discriminability between normal and anomalous samples. Inspired by contrastive learning [14], the ContrastFusion Module addresses this gap by integrating denoising reconstruction and asymmetric contrastive learning, actively expanding feature distribution distances between normal and anomalous data.

To generate negative samples for contrastive learning without introducing excessive prior knowledge, we use Gaussian noise to generate negative samples: $X_n = X + \alpha J$, where $\alpha$ controls the noise level and $J$ is sampled from a Gaussian distribution. Gaussian noise, as a widely used general perturbation method, simulates the "universal abnormal deviations" existing in most time series data [15]. Inspired by the work on Denoising Autoencoders [16], a denoising loss is added to the objective function to enforce the model's ability to recover clean data from noisy inputs:

$$L_{denoise} = MSE\left(\hat{X}_n, X\right) + \left(1 - CosineSimilarity\left(\hat{X}_n, X\right)\right) \tag{10}$$

where $\hat{X}_n$ is the reconstruction of $X_n$, and $CosineSimilarity$ denotes the cosine similarity function. This module uses two main synergistic processes: denoising reconstruction and feature contrast. Positive (normal) and negative (noisy) samples are fed into the PatchEmbedder and Prototype Memory Encoder to obtain latent representations $N$ and $N_n$. $N$ is then fed into a reconstruction projection layer $\mathcal{P}_1$ and a contrastive projection layer $\mathcal{P}_2$ to obtain the reconstruction output $\hat{X}_n$ and the low-dimensional hidden representation $H = \mathcal{P}_2(N) = Linear(ReLU(Linear(N)))$. Corresponding low-dimensional representations $H$ (normal) and $H_\mathrm{n}$ (anomalous) are derived from $N$ and $N_n$. To avoid training collapse and ensure stable optimization, an asymmetric contrastive loss with Stop Gradient is introduced to generate dynamic targets:

$$\begin{aligned} L_{cont} = {} & MSE\left(H, StopGrad\left(H_n\right)\right) + \left(1 - Similarity\left(H, StopGrad\left(H_n\right)\right)\right) \\ & + MSE\left(H_n, StopGrad\left(H\right)\right) + \left(1 - Similarity\left(H_n, StopGrad\left(H\right)\right)\right) \end{aligned} \tag{11}$$

This loss creates a "repulsive field" in the latent space, maximizing discrepancies between normal and anomalous features to polarize representations. The final joint loss function is:

$$L = L_{rec} + L_{denoise} - \beta L_{cont} \tag{12}$$

where $L_{rec}$ is the reconstruction loss and $\beta$ balances the contrastive term. $\beta$ is dynamically adjusted during training to balance stability and performance: in the early stages, it is assigned a very small weight

to prioritize normal pattern learning via reconstruction/denoising losses, thereby avoiding contrastive learning instability; in the later stages, it is fixed at 0.2 (the maximum value) to maximize the effects of contrastive learning.

The ContrastFusion module breaks through the limitations of reconstruction-only models, enabling the model to not only reconstruct normal patterns but also actively recognize fundamental feature differences between normal and anomalous samples, significantly enhancing generalization for unknown anomalies. Even if high-capacity models attempt to reconstruct subtle anomalies, the contrastive constraint ensures anomalous features remain distinguishable from normal prototypes in the latent space.

## 4 Experiment

### *4.1 Dataset*

PC-UAD is evaluated on five real-world multivariate time series datasets covering diverse domains. **NIPS_TS_Swan** (Swan): Telemetry data from the NASA International Space Station, a benchmark for the NeurIPS 2021 Time Series Competition; **PSM**: Production server monitoring data from eBay, reflecting the operational status of large-scale internet services. Related microservice monitoring work is reported in our previous study [17]. **SWaT**: Sensor and actuator data from a real-world water treatment plant, containing anomalies from physical constraints and attack scenarios. **MSL**: Spacecraft subsystem operational parameters under Martian extreme conditions, widely used to validate MTSAD algorithms for high-reliability aerospace scenarios. **SMD**: collected from a large internet company, recording resource utilization of computer clusters with 38 dimensions and posing challenges for capturing dynamic data trends. The five public datasets were screened for high diversity in anomaly types, data scale, dimensionality, anomaly rates and application backgrounds, covering a wide spectrum of real-world multivariate time series scenarios and anomaly patterns.

### *4.2 Baseline Model*

PC-UAD is compared with eight state-of-the-art unsupervised MTSAD models:

**SimAD** [18]: A dissimilarity-based method that uses enhanced feature extraction, normal-pattern embedding, and contrastive fusion for anomaly discrimination.

**TranAD** [7]: A Transformer-based encoder-decoder model with a two-stage self-conditioning inference mechanism to amplify anomaly signals, integrated with adversarial training for robustness.

**DAGMM** [19]: An end-to-end model combining deep autoencoders with Gaussian Mixture Models, enabling anomaly detection via low-dimensional representation and probability estimation.

**OmniAnomaly** [20]: A stochastic recurrent neural network model that learns normal data distributions in the latent space, using reconstruction probability as the anomaly score.

**USAD** [21]: An adversarial autoencoder model with one encoder and two decoders, which learns normal patterns via an adversarial game between reconstruction and error-amplification decoders.

**DADA** [2]: A general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. It achieves zero-shot anomaly detection.

**RTdetector** [3]: A Transformer-based model leveraging reconstruction trends, incorporating a global attention mechanism based on reconstruction trends and a self-conditioning Transformer with reconstruction trend enhancement.

**AnomalyTransformer** [6]: A Transformer-based model proposing an association discrepancy-based Anomaly-Attention mechanism, quantifying anomaly degrees by measuring attention weight inconsistency across positions.

### 4.3 Evaluation Metrics

To ensure rigorous and unbiased evaluation, F1-score and Area Under the ROC Curve (ROC-AUC) are adopted as core metrics:

**F1-score**: Requires accurate classification of each independent timestamp, avoiding the overestimation bias of point adjustment (PA) and providing a strict measure of per-time-step discriminability;

**ROC-AUC**: Insensitive to decision thresholds, evaluating the model's overall ability to rank normal and anomalous samples, offering a robust benchmark for cross-model comparison.

### 4.4 Experiment Result

As shown in Table 1 and Fig. 4a, PC-UAD achieves optimal or sub-optimal performance in terms of F1-score and ROC-AUC across all five datasets, and demonstrates robust adaptability to different anomaly patterns and data characteristics.

**Table 1:** Anomaly detection performance (F1-score, top; ROC-AUC, bottom) on public datasets.

| Dataset | SimAD | TranAD | DAGMM | Omni Anomaly | USAD | PC-UAD | DADA | Anomaly Transformer | RTdetector |
|---|---|---|---|---|---|---|---|---|---|
| MSL | 22.36 | 14.56 | 13.50 | 10.95 | 12.10 | 23.98 | **33.50** | 9.69 | 16.49 |
| Swan | 67.53 | 29.47 | 25.56 | 23.97 | 15.67 | **70.62** | 49.17 | 11.69 | 29.18 |
| PSM | 46.59 | 40.21 | 38.65 | 26.65 | 34.69 | **47.62** | 46.88 | 36.42 | 39.47 |
| SWaT | 78.5 | 59.93 | 62.06 | 77.52 | 65.14 | **79.08** | 53.56 | 56.85 | 62.17 |
| SMD | 21.91 | **24.62** | 21.99 | 19.57 | 23.63 | 22.15 | 19.35 | 16.08 | 22.74 |
| MSL | 57.73 | 53.77 | 51.98 | 51.48 | 52.03 | 62.54 | **75.09** | 51.03 | 54.81 |
| Swan | 81.67 | 57.57 | 54.38 | 56.55 | 53.27 | **84.66** | 53.21 | 52.18 | 57.52 |
| PSM | 66.64 | 60.32 | 59.84 | 56.34 | 59.12 | **68.54** | 56.83 | 60.96 | 60.83 |
| SWaT | **91.20** | 71.41 | 72.61 | 82.02 | 80.35 | 91.00 | 82.81 | 60.72 | 72.58 |
| SMD | 77.52 | 57.16 | 56.22 | 55.42 | 56.11 | **77.77** | 72.08 | 52.36 | 56.45 |

Note: The best score in each experiment is in bold.

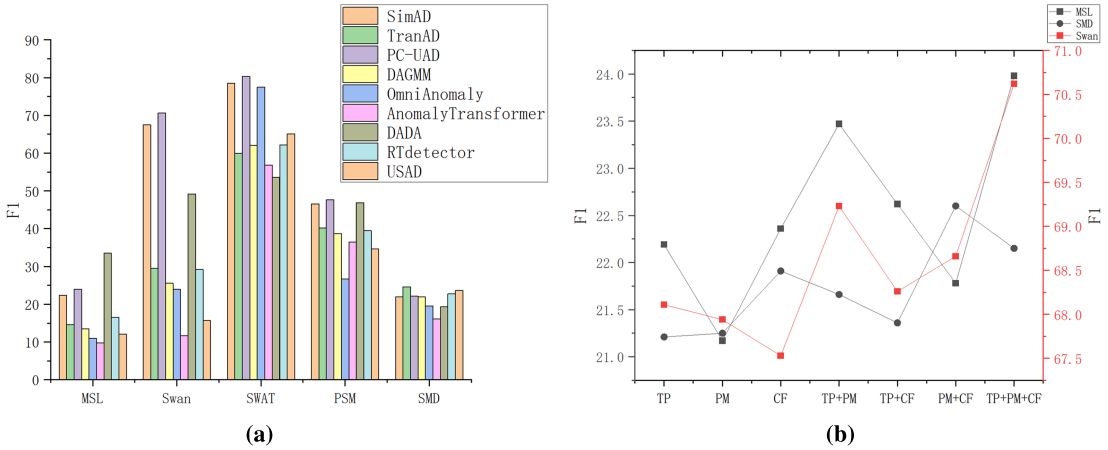


**Figure 4:** Experiment result: (**a**) The F1 score of the comparison model on the datasets; (**b**) Ablation experiment results.

In terms of average F1, the proposed PC-UAD ranks first among all the compared models:

MSL Dataset: Characterized by aerospace subsystem operational parameters with a low anomaly rate (10.72%), subtle deviations are hard to detect. PC-UAD achieves an F1-score of 23.98%, outperforming

mainstream baselines but lower than DADA. This indicates DADA's advantage in capturing subtle anomalies via multi-domain pre-training.

On the Swan dataset, PC-UAD reaches an F1-score of 70.62%, surpassing the closest baseline SimAD by 3.09%, and significantly outperforming AnomalyTransformer and RTdetector by a large margin, reflecting its superiority in modeling multi-type short-term anomalies.

On the PSM dataset (A large-scale internet service monitoring dataset with high noise), PC-UAD achieves the highest F1 of 47.62%, highlighting the prototype memory's ability to resist noise interference.

On the SWaT dataset, PC-UAD obtains an F1-score of 79.08%, exceeding SimAD and OmniAnomaly, and demonstrating strong capability in detecting anomalies with complex spatio-temporal dependencies.

On the SMD, PC-UAD's F1-score is lower than that of TranAD and USAD. The specific reasons were revealed in the ablation experiment. But it still performs closely to RTdetector and outperforms other baselines.

PC-UAD maintains leading ROC-AUC performance across most of the datasets: On MSL, it ranks second only to DADA but outperforms most baselines. On Swan, its score exceeds SimAD and is far higher than the others. On SWaT, it reaches 91.00%, slightly lower than SimAD but surpassing the others. On SMD (low anomaly rate: 4.16%, scattered weak anomalies) and PSM (high anomaly rate: 27.76%, concentrated severe anomalies), it outperforms other baselines, demonstrating the global discrimination capability to scattered anomalies and stability without being overwhelmed by high anomaly density.

Cross-Scenario Robustness: Across datasets with varying anomaly rates, PC-UAD maintains leading ROC-AUC and competitive F1-score, verifying its robustness to both weak and severe anomalies.

### *4.5 Ablation Analysis*

Ablation experiments investigate the contribution of PC-UAD's three core components: Temporal PatchEmbedder (TP), Prototype Memory (PM) Encoder and ContrastFusion (CF) Module. Results in Table 2, and Fig. 4b shows that the complete PC-UAD model achieves the highest average F1-score of 38.92%, confirming the synergistic effect of the three modules.

**Table 2:** Performance of PC-UAD with key components removed (F1-score).

| TP | PM | CF | MSL | Swan | SMD | Average F1 |
|---|---|---|---|---|---|---|
| × | × | √ | 22.36 | 67.53 | 21.91 | 37.27 |
| × | √ | √ | 21.78 | 68.66 | **22.60** | 37.68 |
| √ | × | √ | 22.62 | 68.26 | 21.36 | 37.41 |
| √ | × | × | 22.19 | 68.11 | 21.21 | 37.17 |
| √ | √ | × | 23.47 | 69.23 | 21.66 | 38.12 |
| × | √ | × | 21.17 | 67.94 | 21.25 | 36.79 |
| √ | √ | √ | **23.98** | **70.62** | 22.15 | **38.92** |

Note: The best score in each experiment is in bold.

Removing the PM module causes the most significant average F1 drop (1.51%, 38.92%→37.41%), highlighting its core role in anchoring normal patterns and mitigating overgeneralization. Without the PM module's "normal pattern dictionary", the model loses explicit normal feature references, blurring the boundary between normal and anomalous samples, especially for subtle anomalies in low-anomaly-rate datasets (e.g., MSL).

Disabling the CF module leads to an average F1 reduction of 0.80% (38.92%→38.12%), verifying its effectiveness in actively enhancing normal-anomaly feature discriminability. Lacking the CF module's asymmetric contrastive loss and Gaussian noise-based negative sampling, the model can only rely on reconstruction errors, which is insufficient for identifying anomalies with small reconstruction deviations (e.g., Swan).

Removing the TP module results in a 1.24% average F1 decrease (38.92%→37.68%), validating its value in adaptive temporal and cross-sensor feature extraction. The loss of learnable positional encoding and channel projection makes the model unable to effectively capture dynamic temporal dependencies and multi-scale features of multivariate time series, leading to the loss of key anomaly-discriminative information.

Notably, the TP module shows a unique trend on the SMD dataset: the model with TP disabled achieves a higher F1-score (22.60%) than with TP enabled. This is due to SMD's characteristics (scattered sparse anomalies, stable cross-sensor dependencies)—the TP module's adaptive feature extraction overfits to local temporal fluctuations and redundant sensor correlations, masking weak anomaly signals, while simple direct feature extraction better preserves sparse anomaly integrity.

Models with two modules removed show more severe performance degradation (average F1 < 37.30%), confirming each module is irreplaceable. Their synergistic integration enables PC-UAD to target the three core challenges of unsupervised MTSAD, achieving superior detection performance across diverse datasets.

### *4.6 Parameter Sensitivity Analysis*

(1) Impact of Prototype Count: Model performance is tested with varying prototype counts (Fig. 5a). The optimal average performance is achieved when the prototype count ($M$) is set to 200 for most datasets, while excessive prototypes increase overfitting risk.

(2) Impact of Patch Size: Patch size directly affects temporal granularity (Fig. 5b). For the Swan dataset, smaller patches (size = 32) perform better. This dataset contains more instantaneous anomalies and local patterns, and these fine-grained temporal features are diluted in larger patches, whereas smaller patches preserve the fine granularity to enable accurate capture of short-term anomalous signals. For other datasets, medium patch sizes (size = 64–128) achieve stable performance by balancing local fluctuation and global trend modeling.

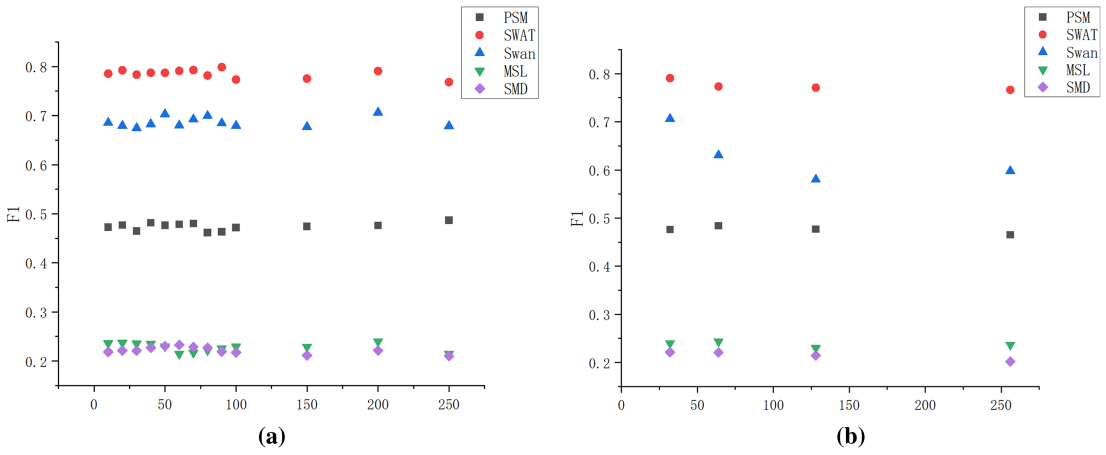


**Figure 5:** Parameter sensitivity analysis: (**a**) F1-score vs. prototype count; (**b**) F1-score vs. patch size.

### *4.7 Inference Speed and Memory Consumption Analysis*

To validate the practical deployability of PC-UAD in real-time monitoring scenarios, we evaluate its inference speed and memory consumption with a batch size of 1 (worst-case latency for real-time processing). As shown in Table 3, PC-UAD achieves low-latency inference speed across all five multivariate datasets, with single-sample processing time ranging from 15.42 to 22.36 μs. This performance meets the real-time requirements of most industrial monitoring and microservice diagnosis scenarios, outperforming federated learning-based methods [22], which often incur additional communication latency.

**Table 3:** Single sample inference speed and memory consumption.

| Dataset | Train Size | Test Size | Dimension | Anomaly Rate | Time/Sample (μs) | Peak GPU Memory (MB) | Peak CPU Memory (MB) | GPU Memory/Sample (MB) |
|---|---|---|---|---|---|---|---|---|
| MSL | 58,317 | 73,729 | 55 | 10.72% | 18.65 | 120.23 | 1063.22 | 120.23 |
| Swan | 587,000 | 425,000 | 38 | 12.5% | 19.73 | 116.73 | 1020.87 | 116.73 |
| PSM | 132,481 | 87,841 | 25 | 27.76% | 15.42 | 115.57 | 1037.13 | 115.57 |
| SWAT | 496,800 | 449,919 | 51 | 11.98% | 16.10 | 119.68 | 2031.09 | 119.68 |
| SMD | 708,405 | 708,420 | 38 | 4.16% | 22.36 | 116.73 | 1365.47 | 116.73 |

PC-UAD exhibits lightweight memory usage, with peak GPU memory consumption ranging from 115.57 to 120.23 MB across datasets. Per-sample GPU memory overhead is consistent at approximately 116–120 MB, indicating that the model's memory usage is not significantly affected by data volume or dimensionality. Peak CPU memory consumption ranges from 1020.87 to 2031.09 MB, which is manageable for standard computing hardware and does not hinder deployment in resource-constrained environments.

Key hyperparameters of PC-UAD are configured as follows: the input time series window length is set to 2048; the dimension of the Transformer model embedding $D$ is 512; the number of learnable prototypes in the memory encoder $M$ is 200; the number of multi-head attention heads is 8; the noise intensity $\alpha$ for Gaussian noise injection in the ContrastFusion module is 0.1; the initial learning rate of the Adam optimizer is 1e−4; and the total training epochs are set to 20.

## 5 Conclusion

This paper proposes PC-UAD, a novel framework that systematically addresses three key challenges of MTSAD: limited temporal context modeling, insufficient normal-pattern retention, and weak normal-anomaly feature discriminability. PC-UAD's innovations include a Temporal PatchEmbedder with learnable positional encoding, a Prototype Memory Encoder for explicit normal-pattern memorization, and a ContrastFusion module for active feature polarization. Extensive experiments on five real-world datasets demonstrate that PC-UAD outperforms state-of-the-art baselines in F1-score and ROC-AUC, with particular advantages for complex datasets with subtle anomalies. The current negative sample construction relies on Gaussian noise, which may not fully capture complex real-world anomalies (e.g., system-level faults). Future work will focus on: integrating domain knowledge to design hybrid negative sample generation strategies that simulate scenario-specific rare anomalies, introducing few-shot learning mechanisms to enable rapid adaptation to new anomaly patterns with limited labeled data, and exploring cross-domain transfer learning via multi-source pre-training to enhance the model's adaptability to heterogeneous unseen scenarios. These efforts aim to further expand PC-UAD's practical applicability in complex real-world monitoring systems.

**Acknowledgement:** Not applicable.

**Funding Statement:** This research is supported by the National Natural Science Foundation of China under Grant No. 62376043, Sichuan Provincial Natural Science Foundation under Grant No. 2024NSFTD0008, Science and Technology Program of Sichuan Province under Grant No. 2024ZHCG0016, Science and Technology Program of Chengdu under Grant No. 2025-GH02-00020-HZ, Science and Technology Program of Quzhou under Grant No. 2024K008, and the Open Project Program of the State Key Laboratory of CAD and CG(Grant No. A2509), Zhejiang University.

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization, Xi Li and Peng Chen; methodology and writing, Xi Li and Yingjie Chang; software, Yingjie Chang; validation, Ang Bian; project administration and funding acquisition, Peng Chen and Ning Lu. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The authors confirm that the data supporting the findings of this study are available within the article.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Li Y, Wen Z, Chen Z, Mei J, Lin M, Zhu M. Diffusion models with self-conditioning guidance for multivariate time series anomaly detection. Knowl Based Syst. 2025;330(7):114511. doi:10.1016/j.knosys.2025.114511.
2. Shentu Q, Li B, Zhao K, Shu Y, Rao Z, Pan L, et al. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In: Proceedings of the Thirteenth International Conference on Learning Representations (ICLR); 2025 Apr 24–28; Singapore.
3. Liu X, Li X, Li Y, Tang F, Zhao M. RTdetector: deep transformer networks for time series anomaly detection based on reconstruction trend. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence; 2025 Aug 16–22; Montreal, QC, Canada. p. 5788–96. doi:10.24963/ijcai.2025/644.
4. Dai J, Wang YZ. Autoencoder anomaly detection method enhanced by hash memory network. J Chin Comput Syst. 2024;45(6):1301–10. (In Chinese).
5. Li X, Song W, Chen P, Xi Q, He H. An efficient semi-supervised multivariate time series classification model based on multiple prime convolution kernels with adaptive attentions. Intell Data Anal Int J. 2025;30:1195. doi:10.1177/1088467x251387619.
6. Xu J, Wu H, Wang J, Long M. Anomaly Transformer: time series anomaly detection with association discrepancy. In: Proceedings of the Tenth International Conference on Learning Representations (ICLR); 2022 Apr 25–29; Virtual.
7. Tuli S, Casale G, Jennings NR. TranAD: deep transformer networks for anomaly detection in multivariate time series data. Proc VLDB Endow. 2022;15(6):1201–14. doi:10.14778/3514061.3514067.
8. Wu H, Hu T, Liu Y, Zhou H, Wang J, Long M. TimesNet: temporal 2D-variation modeling for general time series analysis. In: Proceedings of the Eleventh International Conference on Learning Representations (ICLR); 2023 May 1–5; Kigali, Rwanda.
9. Li Y, Chen W, Chen B, Wang D, Tian L, Zhou M. Prototype-oriented unsupervised anomaly detection for multivariate time series. In: Proceedings of the 40th International Conference on Machine Learning (ICML); 2023 Jul 23–29; Honolulu, HI, USA.
10. Xin R, Wang J, Chen P, Zhao Z. Trustworthy AI-based performance diagnosis systems for cloud applications: a review. ACM Comput Surv. 2025;57(5):1–37. doi:10.1145/3701740.
11. Xiao Z, Luo C, Hu J, Sa G, Wang Y. Exploring dual-view graph structures: contrastive learning with graph and hypergraph for multivariate time series classification. Neural Netw. 2025;192(10):107859. doi:10.1016/j.neunet.2025.107859.

12. Xiao BW, Xing HJ, Li CG. MulGad: multi-granularity contrastive learning for multivariate time series anomaly detection. Inf Fusion. 2025;119(332):103008. doi:10.1016/j.inffus.2025.103008.
13. Kim T, Kim J, Tae Y, Park C, Choi J, Choo J. Reversible instance normalization for accurate time-series fore-casting against distribution shift. In: Proceedings of the Tenth International Conference on Learning Representations (ICLR); 2022 Apr 25–29; Virtual.
14. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning (ICML); 2020 Jul 13–18; Virtual. p. 1597–607.
15. Zhou S, Zha D, Shen X, Huang X, Zhang R, Chung K. Denoising-aware contrastive learning for noisy time series. In: Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI); 2024 Aug 3–9; Jeju, Republic of Korea. p. 5644–52. doi:10.24963/ijcai.2024/624.
16. Lea C, Flynn MD, Vidal R, Reiter A, Hager GD. Temporal convolutional networks for action segmentation and detection. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 1003–12. doi:10.1109/CVPR.2017.113.
17. Li X, Wen P, Chen P, Chen J, Wen X, Xia Y. An effective parallel convolutional anomaly multi-classification model for fault diagnosis in microservice system. Softw Qual J. 2024;32(3):921–38. doi:10.1007/s11219-024-09672-6.
18. Zhong Z, Yu Z, Xi X, Xu Y, Cao W, Yang Y, et al. SimAD: a simple dissimilarity-based approach for time-series anomaly detection. IEEE Trans Neural Netw Learn Syst. 2025;36(11):19669–80. doi:10.1109/TNNLS.2025.3590220.
19. Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho D, et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: Proceedings of the International Conference on Learning Representations; 2018 Apr 30–May 3; Vancouver, BC, Canada.
20. Su Y, Zhao Y, Niu C, Liu R, Sun W, Pei D. 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 Aug 4–8; Anchorage, AK, USA. p. 2828–37. doi:10.1145/3292500.3330672.
21. Audibert J, Michiardi P, Guyard F, Marti S, Zuluaga MA. USAD: UnSupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020 Jul 6–10; Virtual. p. 3395–404. doi:10.1145/3394486.3403392.
22. Hao J, Chen P, Chen J, Li X. Effectively detecting and diagnosing distributed multivariate time series anomalies via Unsupervised Federated Hypernetwork. Inf Process Manag. 2025;62(4):104107. doi:10.1016/j.ipm.2025.104107.