Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Structure-Aware Cross-Layer Gated Attention Transformer for Unsupervised Multivariate Server Time-Series Anomaly Detection

Jing Quan

jquan@cqut.edu.cn

Chongqing University of Technology

Can Zeng

Chongqing University of Technology

Ting Xie

Chongqing University of Technology

Research Article

Keywords:

Posted Date: November 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11594384/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Structure-Aware Cross-Layer Gated Attention Transformer for Unsupervised Multivariate Server Time-Series Anomaly Detection

Jing Quan[1*], Zeng Can[2] and Ting Xie[3]

[1*]School of Mathematical Sciences, Chongqing University of Technology, Huaxi Street, Banan, 400054, Chongqing, China.
[2,3]School of Mathematical Sciences, Chongqing University of Technology, Huaxi Street, Banan, 400054, Chongqing, China.

*Corresponding author(s). E-mail(s): jquan@cqut.edu.cn;

Abstract

Unsupervised anomaly detection in multivariate server time series is challenging because temporal dependencies vary across variables and time scales. Reconstruction-based Transformers can model them, but they often lose fine-grained temporal detail, reuse cross-layer context indirectly, and rely mainly on reconstruction error. This paper proposes the Cross-Layer Gated Attention Transformer (CLGAT), an unsupervised reconstruction framework for server monitoring data. CLGAT combines four components. Overlapping Patch Embedding preserves local context while restoring the original temporal resolution through linear interpolation. Cross-Layer Dynamic Context Residual retrieves representations from preceding encoder layers and injects them into the current layer through time-step gating. Dual-stage Gated-Attention modulates temporal attention weights and aggregated feature responses. The Window Attention Structure aggregates selected long-range attention responses, normalizes them through temperature-controlled chunk-wise Softmax, and fuses the resulting structural score with a Mean–Top-$\boldsymbol{K}$ reconstruction score. The threshold is set at the 99.5th percentile of validation anomaly energy. On the SMD benchmark, CLGAT achieves 94.59% Precision, 96.04% Recall, and 95.31% F1-score under the point-adjusted protocol, outperforming 15 baselines. Ablation confirms all four components contribute, with the largest drop after removing the Window Attention Structure.

Keywords: Transformer; Gated-Attention; Cross-layer residual; Server Multivariate Time Series; Anomaly Detection

1 Introduction

Server clusters support cloud computing, edge computing, and distributed data services under high concurrency and tightly coupled system conditions. Traditional statistical, density-based, and threshold methods provide practical baselines [1, 2], but their performance depends on predefined criteria or decision settings. Because abnormal samples are scarce and manual annotation is costly, unsupervised time-series anomaly detection is a suitable approach for server monitoring [3].

Existing unsupervised methods mainly follow reconstruction, prediction, or dependency-modeling paradigms [4, 5]. Reconstruction and prediction models learn normal temporal patterns from historical observations and flag deviations from the learned representation. Recurrent, variational, and Transformer architectures provide the main backbones for this task. DDConv introduces a reconstruction network for server anomaly detection, with emphasis on local temporal structures [6]. A depthwise-separable convolution and Transformer framework combines local and global context through sliding-window reconstruction [7]. RE-ADTS separates reconstruction and anomaly decision modules [8], while Transformer-based reconstruction provides a generic framework for temporal deviation detection [9]. TranAD adds self-conditioning and adversarial training [10], and CT-DDPM combines copula dependency modeling with diffusion-based reconstruction [11]. USAD reconstructs multivariate sequences through an adversarial encoder-decoder structure [14]. Some methods adapt the anomaly criterion to specific data characteristics, including F1-score and cross-entropy objectives for sparse binary data [12] and noise-to-norm reconstruction for industrial anomaly localization [13]. Ensemble and stochastic latent-variable methods, such as EMRL [15] and OmniAnomaly [16], further show the benefit of richer representations. These studies provide effective reconstruction and prediction backbones, but their final decisions are usually based on numerical residuals. They do not explicitly use changes in learned long-range temporal associations as anomaly evidence.

Another line of work models dependencies among variables and time steps. Graph neural networks and self-attention architectures learn inter-variable and temporal relationships rather than treating monitoring channels independently [17, 18]. Anomaly Transformer uses association discrepancy to compare temporal association patterns [19]. Parallel attention has been applied to electrical equipment monitoring [20], and fragment-level weak supervision has been used to localize abnormal video regions [21]. Temporal neighborhood attention emphasizes informative input features for decoder-based detection [22], while attention-based Siamese networks support few-shot anomaly detection [23]. Deviation-guided attention combines latent embeddings, residual direction, and hierarchical reconstruction errors [24]. Recurrent convolutional encoding with attention has also been used for chaotic time-series detection [25]. These methods show that learned dependencies can improve anomaly discrimination. However, attention is often used as an internal representation mechanism, while relative deviations in selected long-range attention responses are rarely included in the final anomaly score. In stacked encoders, earlier-layer representations are also reused mainly through adjacent layer-to-layer paths.

For multivariate server time series, three issues therefore remain unresolved. First, patch-based and window-based representations can lose fine-grained temporal detail,

which weakens point-level localization. Second, deeper layers do not directly reuse the contextual information accumulated across all preceding layers. Third, reconstruction and prediction criteria do not explicitly represent changes in long-range temporal structure. These issues are further complicated by heterogeneous temporal dynamics across variables and operating states. Existing reviews emphasize the need for representations at different temporal granularities [26], while reconstruction and prediction remain the dominant anomaly criteria [27].

To address these gaps, we propose the Cross-Layer Gated Attention Transformer (CLGAT), an unsupervised reconstruction framework for multivariate server time-series anomaly detection. CLGAT combines four components: overlapping patch embedding, CLDC Residual, dual-stage Gated-Attention, and the Window Attention Structure (WAS). The main contributions of this work are summarized as follows.

- **Overlapping patch embedding with resolution restoration.** We introduce overlapping temporal patches to incorporate local context into each latent representation and use linear interpolation to restore the original temporal resolution. This preserves point-wise temporal detail while allowing each representation to depend on neighboring observations.
- **Cross-layer dynamic context residual (CLDC Residual).** We retrieve contextual information from all preceding encoder-layer outputs through cross-layer attention and inject the aggregated context into the current layer through time-step gating. This gives deeper layers direct access to representations accumulated at different encoding depths.
- **Dual-stage gated attention.** We develop a shared-gating attention mechanism that modulates temporal attention weights and aggregated feature responses at two levels. The first gate regulates each query-specific temporal response; the second adjusts the resulting latent features.
- **Window attention structure for structure-aware scoring.** We aggregate selected bidirectional long-range attention responses into a structural score and normalize it through temperature-controlled chunk-wise Softmax. The score is fused with the Mean–Top-K reconstruction score to form an anomaly energy that uses both numerical deviation and temporal structure.
- **Evaluation on server monitoring data.** We evaluate CLGAT on the Server Machine Dataset (SMD). It achieves a Precision of 94.59%, a Recall of 96.04%, and an F1-score of 95.31%, outperforming the 15 baseline methods. Ablation experiments quantify the contribution of each proposed component.

The remainder of this paper is organized as follows. Section 2 presents the CLGAT architecture and anomaly detection procedure. Section 3 reports the experimental results, ablation studies, and parameter analysis. Section 4 concludes the paper and discusses future work.

2 Methods

This section presents the methodological framework of CLGAT for unsupervised multivariate time-series anomaly detection.

2.1 Problem Formulation

Multivariate server monitoring data consist of multiple hardware and system variables observed simultaneously at successive timestamps.

Let the normal training sequence and test monitoring sequence be denoted by $\mathbf{X}_{\text{train}} \in \mathbb{R}^{L_{\text{train}} \times C}$ and $\mathbf{X}_{\text{test}} \in \mathbb{R}^{L_{\text{test}} \times C}$, respectively, where L_{train} and L_{test} denote the corresponding numbers of sampling timestamps and C denotes the number of monitoring variables.

Because different monitoring variables may exhibit substantially different numerical ranges, feature-wise z-score standardization is performed before window construction. The mean and standard deviation of each variable are estimated exclusively from the normal training sequence:

$$\mu_c = \frac{1}{L_{\text{train}}} \sum_{t=1}^{L_{\text{train}}} \mathbf{X}_{\text{train}}[t, c], \tag{1}$$

and

$$\sigma_c = \sqrt{\frac{1}{L_{\text{train}}} \sum_{t=1}^{L_{\text{train}}} \left(\mathbf{X}_{\text{train}}[t, c] - \mu_c\right)^2}. \tag{2}$$

The resulting training-derived statistics are applied consistently to the training, validation, and test sequences. For a generic monitoring sequence $\mathbf{X}$, the standardized continuous sequence is denoted by $\widetilde{\mathbf{X}}$, with

$$\widetilde{\mathbf{X}}[t, c] = \frac{\mathbf{X}[t, c] - \mu_c}{\sigma_c}. \tag{3}$$

This procedure places different monitoring variables on comparable numerical scales while preventing statistics from the test sequence from affecting input normalization.

The standardized sequences are segmented into overlapping windows of temporal length L. For an anchor timestamp $t = L, L{+}1, \ldots, L_{\text{seq}}$, where L_{seq} denotes the length of the corresponding standardized sequence, the input window ending at timestamp t is defined as

$$\mathbf{W}_t = \left[\widetilde{\mathbf{X}}[t - L + 1], \widetilde{\mathbf{X}}[t - L + 2], \ldots, \widetilde{\mathbf{X}}[t]\right] \in \mathbb{R}^{L \times C}. \tag{4}$$

Adjacent windows are extracted with a stride of one timestamp and therefore share $L - 1$ observations. For a sequence of length L_{seq}, the total number of overlapping windows is

$$N_{\text{win}} = L_{\text{seq}} - L + 1. \tag{5}$$

With $L = 100$, the training, validation, and test sequences yield 708306, 141582, and 708321 overlapping windows, respectively. Training windows are used for model optimization, whereas the validation subset is additionally used for model selection and anomaly-threshold calibration.

The extracted windows are subsequently organized into mini-batches. Given B window samples, the standardized model input is constructed as

$$\mathbf{X}^* = \mathrm{Stack}\left(\mathbf{W}_{t_1}, \mathbf{W}_{t_2}, \ldots, \mathbf{W}_{t_B}\right) \in \mathbb{R}^{B \times L \times C}. \tag{6}$$

The three dimensions correspond to window samples, temporal positions, and monitoring variables. The resulting tensor $\mathbf{X}^* \in \mathbb{R}^{B \times L \times C}$ is subsequently provided to the Patch Embedding module.

During inference, the window-level reconstruction and structural scores obtained from overlapping windows are mapped back to their corresponding positions on the original temporal axis to recover full-length point-wise score sequences. The detailed score aggregation, anomaly-energy construction, and anomaly-decision procedures are presented in the subsequent anomaly-scoring section.

All experiments are conducted on SMD, a widely adopted benchmark for unsupervised multivariate time-series anomaly detection. Its server monitoring sequences contain multiple simultaneously observed system variables with heterogeneous temporal patterns and cross-variable relationships, providing an appropriate setting for evaluating temporal representation learning, dependency modeling, and point-wise anomaly discrimination.

2.2 Overview of the CLGAT Architecture

As illustrated in Figure 1, the proposed CLGAT architecture consists of four complementary components: Patch Embedding, Cross-Layer Dynamic Context Residual (CLDC Residual), Gated-Attention, and Window Attention Structure (WAS). These components jointly enhance local temporal representation, cross-layer contextual aggregation, adaptive temporal dependency modeling, and structure-aware anomaly evaluation.

The standardized input $\mathbf{X}^* \in \mathbb{R}^{B \times L \times C}$ is first processed by Patch Embedding. The resulting embedded representation $\mathbf{X}_{\text{final}} \in \mathbb{R}^{B \times L \times d_{\text{model}}}$ is then provided to the stacked Transformer encoder.

The encoder consists of L_{enc} stacked layers incorporating CLDC Residual and Gated-Attention. The head-wise outputs are then concatenated and projected back to the model feature space.

Based on the gated attention responses generated across encoder layers, WAS extracts and aggregates selected bidirectional long-range temporal attention responses to obtain the raw structural score $s_{\text{topo}}(b, t)$.

Finally, the reconstruction and structural scores obtained from overlapping windows are mapped back to the original temporal axis and averaged over all covering windows. Detailed formulations of the individual components are presented in the following subsections.

2.3 Patch Embedding

Traditional point-wise embedding methods process multivariate observations at individual timestamps and therefore provide limited capability for explicitly capturing

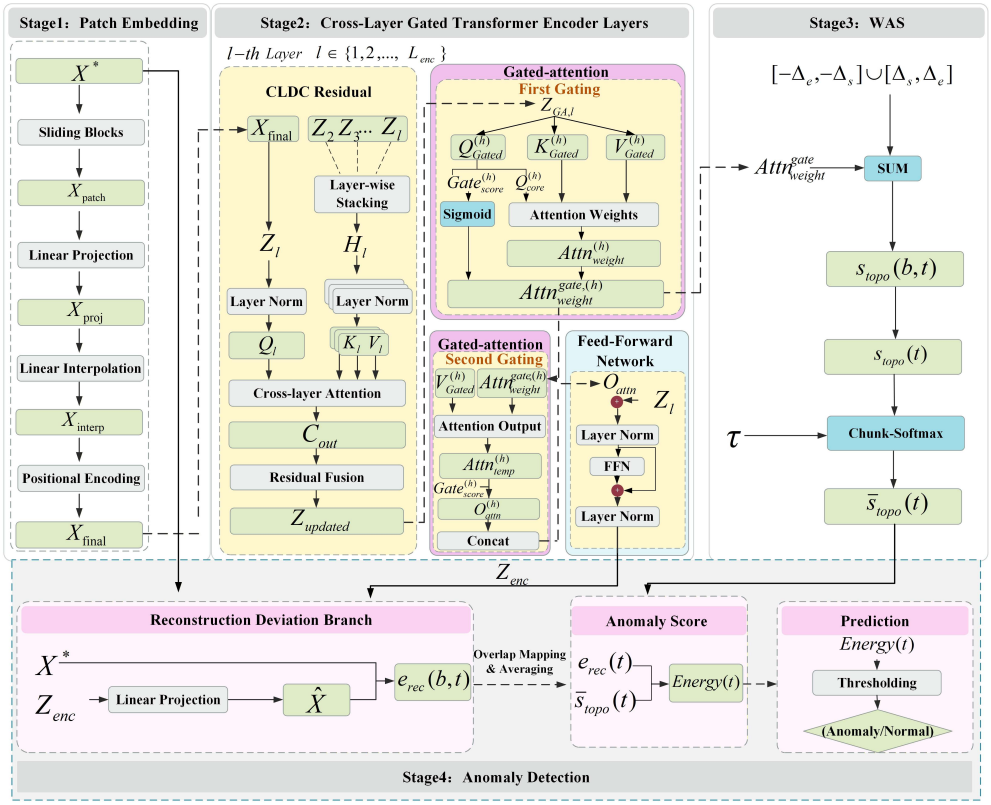


Fig. 1: Overall architecture of the proposed CLGAT model.

local temporal dependencies distributed across consecutive observations. To balance local contextual representation and temporal resolution, we design a Patch Embedding mechanism consisting of four sequential stages: Sliding Blocks, Linear Projection, Linear Interpolation, and Positional Encoding, as illustrated in Figure 2.

Sliding Blocks.

Following feature-wise standardization and sliding-window segmentation, the standardized multivariate input $\mathbf{X}^* \in \mathbb{R}^{B\times L\times C}$ is directly fed into the embedding module, where B, L, and C denote the batch size, temporal window length, and number of monitoring variables, respectively. Let P denote the temporal length of each block and S denote the sliding stride. The required compensation length is defined as follows:

$$L_{\text{pad}} = \begin{cases} S - (L \bmod S), & L \bmod S \neq 0, \\ 0, & L \bmod S = 0. \end{cases} \tag{7}$$

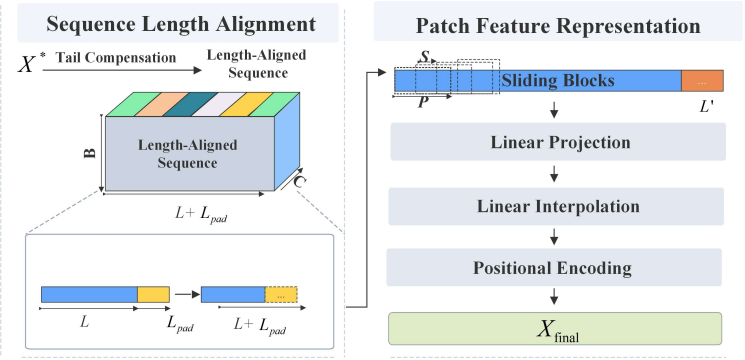


Fig. 2: Architecture of the Patch Embedding module.

The sequence length after tail compensation is calculated as follows:

$$L' = L + L_{\text{pad}}. \tag{8}$$

The length-aligned sequence is subsequently partitioned into overlapping blocks of length P using stride S. The number of valid block positions is calculated as follows:

$$N_P = \left\lfloor \frac{L' - P}{S} \right\rfloor + 1, \tag{9}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation. $\mathbb{R}^{B \times N_P \times (P \cdot C)}$. By jointly organizing consecutive observations and multiple monitoring variables within each local block, this operation provides a structured representation of short-term temporal fluctuations and joint temporal-variable patterns while maintaining continuity between neighboring temporal regions.

Linear Projection.

The flattened block representation is subsequently mapped into a unified latent feature space through a learnable linear transformation.

$$\mathbf{X}_{\text{proj}} = \mathbf{X}_{\text{patch}} \mathbf{W}_{\text{proj}} + \mathbf{b}_{\text{proj}}, \tag{10}$$

where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{(P \cdot C) \times d_{\text{model}}}$ and $\mathbf{b}_{\text{proj}} \in \mathbb{R}^{d_{\text{model}}}$ denote the learnable projection matrix and bias vector, respectively. The projected representation satisfies $\mathbf{X}_{\text{proj}} \in \mathbb{R}^{B \times N_P \times d_{\text{model}}}$. Because the linear transformation operates on the complete flattened block, information from multiple consecutive timestamps and monitoring variables within the same local region is jointly encoded into each latent vector. The resulting representation therefore preserves the local contextual structure established by the Sliding Blocks stage while transforming the original multivariate observations into a compact latent space shared by all block positions.

Linear Interpolation.

Although overlapping block construction enlarges the local temporal receptive field, the projected representation contains only N_P block-level positions and therefore has a lower temporal resolution than the length-aligned sequence of length L'.

$$u_t = \left(t + \frac{1}{2}\right) \frac{N_P}{L'} - \frac{1}{2}. \tag{11}$$

Because the source coordinate may extend beyond the valid range near the sequence boundaries, the effective coordinate is constrained as follows:

$$\tilde{u}_t = \begin{cases} 0, & u_t < 0, \\ u_t, & 0 \le u_t \le N_P - 1, \\ N_P - 1, & u_t > N_P - 1. \end{cases} \tag{12}$$

The left and right neighboring positions associated with $\tilde{u}_t$ are calculated as follows:

$$i_t = \lfloor \tilde{u}_t \rfloor, \tag{13}$$

$$j_t = \min\left(i_t + 1, N_P - 1\right). \tag{14}$$

The interpolation coefficient is calculated as follows:

$$\alpha_t = \tilde{u}_t - i_t. \tag{15}$$

Let $\mathbf{x}_i^{\text{proj}} \in \mathbb{R}^{d_{\text{model}}}$ denote the latent feature vector at the i-th projected block position and $\mathbf{x}_t^{\text{interp}} \in \mathbb{R}^{d_{\text{model}}}$ denote the interpolated latent representation at the t-th target temporal position. The one-dimensional linear interpolation is formulated as follows:

$$\mathbf{x}_t^{\text{interp}} = (1 - \alpha_t)\mathbf{x}_{i_t}^{\text{proj}} + \alpha_t \mathbf{x}_{j_t}^{\text{proj}}. \tag{16}$$

Through this operation, the temporal dimension is restored from N_P block positions to L' dense temporal positions while preserving the latent feature dimension. The resulting representation satisfies $\mathbf{X}_{\text{interp}} \in \mathbb{R}^{B \times L' \times d_{\text{model}}}$. The interpolation process introduces no additional trainable parameters and redistributes the contextual information encoded at the block level onto a dense temporal grid.

Positional Encoding.

After dense temporal resolution is restored, explicit temporal-order information is incorporated to distinguish latent representations associated with different temporal positions. The fixed sinusoidal positional encoding is defined as follows:

$$\begin{cases} PE_{(pos,2r)} = \sin\left(\dfrac{pos}{10000^{2r/d_{\text{model}}}}\right), \\ PE_{(pos,2r+1)} = \cos\left(\dfrac{pos}{10000^{2r/d_{\text{model}}}}\right). \end{cases} \tag{17}$$

The positional representation **PE** shares the temporal and latent dimensions of $\mathbf{X}_{\text{interp}}$ and is added element-wise to the interpolated feature representation. The final embedded representation is formulated as follows:

$$\mathbf{X}_{\text{final}} = \text{Dropout}\left(\mathbf{X}_{\text{interp}} + \mathbf{PE}\right). \tag{18}$$

Before removing the auxiliary temporal positions introduced by tail compensation, $\mathbf{X}_{\text{final}} \in \mathbb{R}^{B\times L'\times d_{\text{model}}}$. Since these auxiliary positions are introduced solely to support regular block extraction and do not correspond to valid temporal positions in the original input window, only the first L positions are retained. Consequently, the final embedded representation satisfies $\mathbf{X}_{\text{final}} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$. When $L_{\text{pad}} = 0$, no temporal truncation is required. The resulting representation preserves fine-grained temporal resolution while incorporating local contextual information from overlapping blocks and explicit temporal-order information, thereby providing the input representation for the subsequent stacked Transformer encoder.

2.4 Cross-Layer Dynamic Context Residual

Traditional Transformer residual connections primarily propagate information between adjacent computational blocks, which limits the direct utilization of representations generated by multiple preceding encoder layers. The overall structure of CLDC Residual is illustrated in Figure 3.

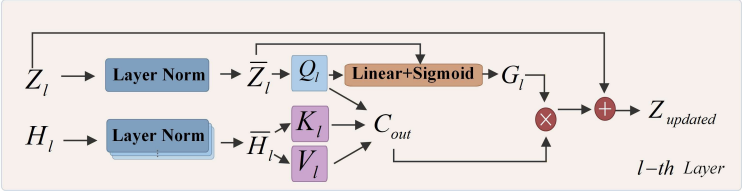


Fig. 3: Architecture of the CLDC Residual.

Following Patch Embedding, the final embedded representation $\mathbf{X}_{\text{final}} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$ is directly fed into the stacked Transformer encoder. Let L_{enc} denote the total number of encoder layers, and let $\mathbf{Z}_l \in \mathbb{R}^{B\times L\times d_{\text{model}}}$ denote the input representation of the l-th encoder layer, where $l = 1, 2, \ldots, L_{\text{enc}}$. The output of the l-th encoder layer is denoted by $\mathbf{Z}_{l+1} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$, which simultaneously serves as the input to the subsequent encoder layer. Accordingly, the initial encoder representation is defined as

$$\mathbf{Z}_1 = \mathbf{X}_{\text{final}}. \tag{19}$$

For the first encoder layer, no preceding encoder representation is available, and CLDC enhancement is therefore omitted.

$$\mathbf{H}_l = \text{Stack}\left(\mathbf{Z}_2, \mathbf{Z}_3, \ldots, \mathbf{Z}_l\right). \tag{20}$$

The resulting tensor satisfies $\mathbf{H}_l \in \mathbb{R}^{(l-1)\times B\times L\times d_{\text{model}}}$. As the encoder depth increases, this historical representation is dynamically updated, enabling the current encoder layer to directly access contextual information generated at multiple preceding encoding depths rather than relying solely on sequential layer-to-layer propagation.

Before cross-layer attention computation, Layer Normalization is applied separately to the current representation and the stacked historical representations:

$$\bar{\mathbf{Z}}_l = \text{LayerNorm}\left(\mathbf{Z}_l\right), \tag{21}$$

$$\bar{\mathbf{H}}_l = \text{LayerNorm}\left(\mathbf{H}_l\right). \tag{22}$$

The normalized current representation is projected into the query space, whereas the normalized historical representations are projected into the key and value spaces. The corresponding query, key, and value representations are formulated as

$$\begin{aligned} \mathbf{Q}_l &= \bar{\mathbf{Z}}_l \mathbf{W}_Q + \mathbf{b}_Q, \\ \mathbf{K}_l &= \bar{\mathbf{H}}_l \mathbf{W}_K + \mathbf{b}_K, \\ \mathbf{V}_l &= \bar{\mathbf{H}}_l \mathbf{W}_V + \mathbf{b}_V. \end{aligned} \tag{23}$$

Here, $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ and $\mathbf{b}_Q, \mathbf{b}_K, \mathbf{b}_V \in \mathbb{R}^{d_{\text{model}}}$ denote the learnable projection parameters. Accordingly, $\mathbf{Q}_l \in \mathbb{R}^{B\times L\times d_{\text{model}}}$, whereas $\mathbf{K}_l, \mathbf{V}_l \in \mathbb{R}^{(l-1)\times B\times L\times d_{\text{model}}}$. This asymmetric projection design allows the current representation to serve as the retrieval query while the preceding encoder outputs provide the historical context to be selectively accessed.

For each historical representation $\mathbf{Z}_m$, where $m = 2, 3, \ldots, l$, the corresponding key and value tensors are denoted by $\mathbf{K}_{l,m} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$ and $\mathbf{V}_{l,m} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$, respectively. The complete cross-layer contextual aggregation is therefore formulated as

$$\mathbf{C}_{\text{out}} = \sum_{m=2}^{l} \text{Softmax}\left(\frac{\mathbf{Q}_l \mathbf{K}_{l,m}^{\top}}{\sqrt{d_{\text{model}}}}\right) \mathbf{V}_{l,m}. \tag{24}$$

The aggregated context satisfies $\mathbf{C}_{\text{out}} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$. Through this operation, the current encoder layer can retrieve temporal information from all available preceding encoder representations and integrate the resulting contextual responses into a unified cross-layer representation.

To regulate the contribution of the aggregated historical context, CLDC Residual further introduces a time-step gating mechanism.

$$\mathbf{G}_l = \sigma\left(\mathbf{Z}_l \mathbf{W}_{\text{gate}} + \mathbf{b}_{\text{gate}}\right), \tag{25}$$

where $\mathbf{W}_{\text{gate}} \in \mathbb{R}^{d_{\text{model}}\times 1}$ and $\mathbf{b}_{\text{gate}} \in \mathbb{R}^1$ denote the learnable gating parameters, and $\sigma(\cdot)$ denotes the sigmoid activation function. The gate therefore controls the contribution of the aggregated historical context at the temporal-position level while preserving the unified feature representation within each time step.

Finally, the aggregated cross-layer context is incorporated into the current representation through gated residual fusion:

$$\mathbf{Z}_{\text{updated}} = \mathbf{Z}_l + \mathbf{G}_l \odot \mathbf{C}_{\text{out}}, \tag{26}$$

where $\odot$ denotes element-wise Hadamard multiplication and $\mathbf{Z}_{\text{updated}} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$. The residual branch preserves the current encoder representation, whereas the gated contextual branch introduces complementary information retrieved from preceding encoder layers.

Overall, CLDC Residual establishes direct contextual interactions between the current encoder layer and preceding encoder representations through historical feature aggregation, cross-layer temporal attention, time-step gating, and residual fusion.

2.5 Gated-Attention

Conventional multi-head self-attention models temporal dependencies by computing pairwise similarities among temporal positions and aggregating contextual information according to the resulting attention distributions. This global interaction mechanism enables each temporal position to access information from the complete input sequence and provides considerable flexibility for modeling complex temporal dependencies. Rather than constructing an independent auxiliary network for gate generation, the proposed mechanism jointly derives the core query representation and gating information through an expanded query projection, allowing the same gating representation to regulate temporal dependency propagation and latent contextual features at two complementary granularities.

The Gated-Attention module receives the representation supplied to the attention stage of each encoder layer. To provide a unified notation for the subsequent formulation, the input to Gated-Attention in the l-th encoder layer is denoted by

$$\mathbf{Z}_{\text{GA},l} = \begin{cases} \mathbf{Z}_l, & l = 1, \\ \mathbf{Z}_{\text{updated}}, & l \geq 2. \end{cases} \tag{27}$$

Accordingly, $\mathbf{Z}_{\text{GA},l} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$, where B, L, and d_{model} denote the batch size, temporal length, and latent feature dimension, respectively. Unlike conventional multi-head attention, the query projection is expanded to twice the latent dimensionality so that the projected representation can simultaneously provide the effective query features and gating information, whereas the key and value projections retain the standard latent dimensionality.

The corresponding query, key, and value representations are formulated as

$$\begin{aligned} \mathbf{Q}_{\text{Gated}} &= \mathbf{Z}_{\text{GA},l}\mathbf{W}_Q^{\text{Gated}} + \mathbf{b}_Q^{\text{Gated}}, \\ \mathbf{K}_{\text{Gated}} &= \mathbf{Z}_{\text{GA},l}\mathbf{W}_K^{\text{Gated}} + \mathbf{b}_K^{\text{Gated}}, \\ \mathbf{V}_{\text{Gated}} &= \mathbf{Z}_{\text{GA},l}\mathbf{W}_V^{\text{Gated}} + \mathbf{b}_V^{\text{Gated}}. \end{aligned} \tag{28}$$

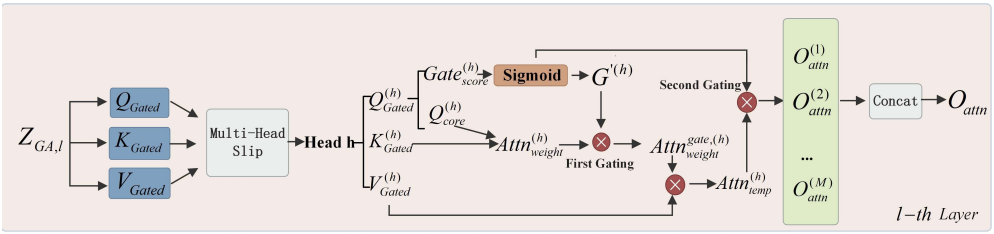


Fig. 4: Architecture of the Gated-Attention module.

Here, $\mathbf{W}_Q^{\text{Gated}} \in \mathbb{R}^{d_{\text{model}} \times 2d_{\text{model}}}$ and $\mathbf{b}_Q^{\text{Gated}} \in \mathbb{R}^{2d_{\text{model}}}$ denote the expanded query projection parameters, whereas $\mathbf{W}_K^{\text{Gated}}, \mathbf{W}_V^{\text{Gated}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ and $\mathbf{b}_K^{\text{Gated}}, \mathbf{b}_V^{\text{Gated}} \in \mathbb{R}^{d_{\text{model}}}$ denote the corresponding key and value projection parameters.

Let M denote the number of attention heads and let $D_{\text{head}} = d_{\text{model}}/M$ denote the latent feature dimension associated with each head. For the h-th attention head, where $h = 1, 2, \ldots, M$, the corresponding expanded query representation is decomposed into two equal-dimensional components:

$$\mathbf{Q}^{(h)}_{\text{Gated}} = \left[\mathbf{Q}^{(h)}_{\text{core}} \parallel \mathbf{Gate}^{(h)}_{\text{score}}\right]. \tag{29}$$

Here, $\parallel$ denotes concatenation along the latent feature dimension. The head-wise expanded query satisfies $\mathbf{Q}^{(h)}_{\text{Gated}} \in \mathbb{R}^{B\times L\times 2D_{\text{head}}}$, whereas $\mathbf{Q}^{(h)}_{\text{core}}$ and $\mathbf{Gate}^{(h)}_{\text{score}} \in \mathbb{R}^{B\times L\times D_{\text{head}}}$. The former serves as the effective query representation for temporal attention computation, whereas the latter provides the shared gating information employed in both modulation stages. Correspondingly, $\mathbf{K}^{(h)}_{\text{Gated}}$ and $\mathbf{V}^{(h)}_{\text{Gated}} \in \mathbb{R}^{B\times L\times D_{\text{head}}}$. Each attention head independently performs the subsequent temporal dependency modeling and gating operations within its own latent subspace, and the head-wise representations are combined only after the complete dual-stage gated computation has been completed.

2.5.1 First Gating: Attention-Weight Modulation

The first gating stage operates directly on the temporal attention weights before contextual feature aggregation.

$$\mathbf{G}^{(h)}_{\text{score}} = \sigma\left(\mathbf{Gate}^{(h)}_{\text{score}}\right). \tag{30}$$

The resulting representation satisfies $\mathbf{G}^{(h)}_{\text{score}} \in \mathbb{R}^{B\times L\times D_{\text{head}}}$, with each gating coefficient bounded within $(0, 1)$. Because the complete feature-wise gating representation contains multiple coefficients at each temporal position, it is averaged along the head-specific latent dimension to obtain a scalar time-step modulation coefficient:

$$\mathbf{G}'^{(h)} = \text{Mean}_{D_{\text{head}}}\left(\mathbf{G}^{(h)}_{\text{score}}\right). \tag{31}$$

Accordingly, $\mathbf{G}'^{(h)} \in \mathbb{R}^{B\times L\times 1}$. Each query position within an attention head is therefore associated with one scalar gating coefficient, which is shared across all key positions corresponding to that query. This design allows the gate to regulate the overall magnitude of the query-specific temporal response without introducing an independent modulation coefficient for every query–key pair.

Because the reconstruction-based anomaly detection task considered in this study does not impose an autoregressive constraint, temporal dependency modeling is performed over the complete input window, allowing each query position to interact bidirectionally with all temporal positions. For the h-th attention head, the temporal dependency distribution is calculated using scaled dot-product attention:

$$\mathbf{Attn}_{\text{weight}}^{(h)} = \text{Softmax}\left(\frac{\mathbf{Q}_{\text{core}}^{(h)}\mathbf{K}_{\text{Gated}}^{(h)\top}}{\sqrt{D_{\text{head}}}}\right). \tag{32}$$

The resulting attention matrix satisfies $\mathbf{Attn}_{\text{weight}}^{(h)} \in \mathbb{R}^{B\times L\times L}$. The Softmax operation is performed over the key-position dimension so that each query position obtains a normalized temporal dependency distribution over the complete input window. The resulting attention distribution is subsequently modulated by the corresponding time-step gating coefficient:

$$\mathbf{Attn}_{\text{weight}}^{\text{gate},(h)} = \mathbf{Attn}_{\text{weight}}^{(h)} \odot \mathbf{G}'^{(h)}. \tag{33}$$

Here, $\mathbf{Attn}_{\text{weight}}^{\text{gate},(h)} \in \mathbb{R}^{B\times L\times L}$, and $\odot$ denotes element-wise Hadamard multiplication. No additional Softmax normalization is applied after the gating operation.

2.5.2 Second Gating: Feature-Level Modulation

The second gating stage operates on the contextual representation obtained after attention-weight modulation.

$$\mathbf{Attn}_{\text{temp}}^{(h)} = \mathbf{Attn}_{\text{weight}}^{\text{gate},(h)}\mathbf{V}_{\text{Gated}}^{(h)}. \tag{34}$$

The resulting contextual representation satisfies $\mathbf{Attn}_{\text{temp}}^{(h)} \in \mathbb{R}^{B\times L\times D_{\text{head}}}$, where each temporal position contains information aggregated from the complete input window according to the gated temporal dependency responses. The feature-wise gating representation is then reused to modulate the aggregated contextual features:

$$\mathbf{O}_{\text{attn}}^{(h)} = \mathbf{Attn}_{\text{temp}}^{(h)} \odot \mathbf{G}_{\text{score}}^{(h)}. \tag{35}$$

The resulting head-wise representation satisfies $\mathbf{O}_{\text{attn}}^{(h)} \in \mathbb{R}^{B\times L\times D_{\text{head}}}$. Because $\mathbf{G}_{\text{score}}^{(h)}$ retains an individual gating coefficient for each latent feature dimension, different latent components at the same temporal position can receive different modulation strengths. The second gating stage therefore provides feature-specific regulation of the contextual information obtained after temporal aggregation.

The two gating stages consequently perform complementary functions within each attention head. After the dual-stage gated computation has been independently completed for all M attention heads, the resulting head-wise representations are concatenated along the latent feature dimension and projected back to the original model space:

$$\mathbf{O}_{\text{attn}} = \text{Concat}\left(\mathbf{O}_{\text{attn}}^{(1)}, \mathbf{O}_{\text{attn}}^{(2)}, \ldots, \mathbf{O}_{\text{attn}}^{(M)}\right) \mathbf{W}_O + \mathbf{b}_O. \tag{36}$$

For each attention head, $\mathbf{O}_{\text{attn}}^{(h)} \in \mathbb{R}^{B\times L\times D_{\text{head}}}$. Since $MD_{\text{head}} = d_{\text{model}}$, the concatenated representation belongs to $\mathbb{R}^{B\times L\times d_{\text{model}}}$. The output projection parameters satisfy $\mathbf{W}_O \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ and $\mathbf{b}_O \in \mathbb{R}^{d_{\text{model}}}$, yielding $\mathbf{O}_{\text{attn}} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$. The resulting attention representation is subsequently passed to the outer residual connection, Layer Normalization, and position-wise feed-forward operations of the current encoder layer.

Overall, the proposed Gated-Attention mechanism introduces two coordinated modulation operations into each attention head through a shared gating representation. This design provides a more selective temporal information propagation mechanism for multivariate monitoring sequences without requiring an additional standalone gating network.

2.6 Window Attention Structure

Reconstruction-based time-series anomaly detection primarily quantifies abnormality through discrepancies between observed and reconstructed values. Although reconstruction error provides an effective numerical criterion for measuring point-wise deviations, it does not explicitly characterize variations in the temporal dependency structures represented by the attention mechanism. The overall architecture of the proposed WAS module is illustrated in Figure 5.

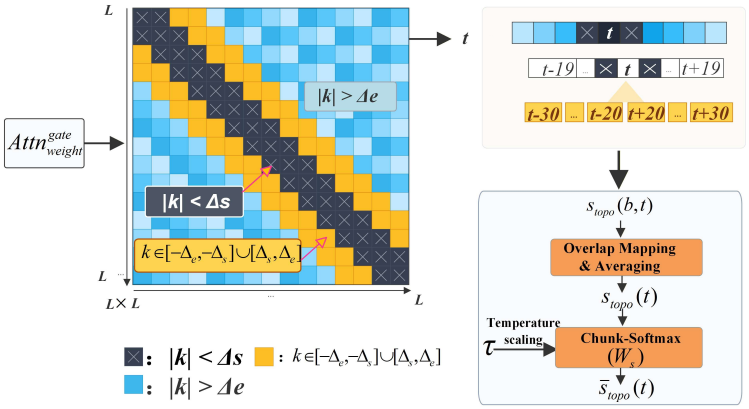


Fig. 5: Architecture of the Window Attention Structure (WAS).

Temporal interactions at different offsets may represent different contextual relationships within multivariate monitoring sequences. Attention responses between

immediately adjacent observations are strongly affected by local temporal continuity, whereas interactions between relatively distant temporal positions provide complementary information beyond neighboring time steps. A smaller Δ_s introduces relatively more local interactions, whereas a larger value shifts the structural aggregation toward more distant temporal dependencies; similarly, Δ_e determines the maximum extent of the temporal interactions included in the resulting structural representation.

Based on the gated attention responses generated by the preceding Gated-Attention module, WAS further performs structural aggregation across both encoder layers and attention heads. Different encoder layers produce representations corresponding to different encoding depths, whereas different attention heads operate within distinct latent subspaces. Consequently, temporal-offset extraction follows a circular boundary-completion strategy, whereby temporal positions extending beyond either boundary of the local window are wrapped to the corresponding positions at the opposite boundary.

Let $\langle q \rangle_L$ denote the one-based circular index obtained by wrapping an arbitrary temporal index q into $\{1, 2, \ldots, L\}$. The raw structural score associated with temporal position t is defined as

$$s_{\text{topo}}(b,t) = \frac{1}{L_{\text{enc}} M} \sum_{l=1}^{L_{\text{enc}}} \sum_{m=1}^{M} \sum_{\Delta=\Delta_s}^{\Delta_e} \Big[\mathbf{Attn}_{\text{weight}}^{\text{gate},(l,m)} \left(b, \langle t-\Delta \rangle_L, t\right) + \mathbf{Attn}_{\text{weight}}^{\text{gate},(l,m)} \left(b, \langle t+\Delta \rangle_L, t\right) \Big]. \tag{37}$$

Here, L_{enc} denotes the number of encoder layers, M denotes the number of attention heads, and $\Delta \in [\Delta_s, \Delta_e]$ denotes the selected temporal offset. The resulting responses are summed over the prescribed offset range and averaged across encoder layers and attention heads, yielding a unified raw structural score for each temporal position.

The raw structural score $s_{\text{topo}}(b,t)$ is initially obtained within each input window. This aggregation procedure preserves temporal alignment between window-based attention computation and the original monitoring sequence while reducing dependence on any single local window.

To further normalize the aggregated structural scores and enhance local contrast, temperature scaling followed by fixed-size non-overlapping chunk-wise normalization is applied. For a timestamp t indexed from 1, the index of the temporal chunk containing t is defined as

$$i(t) = \left\lfloor \frac{t-1}{W_s} \right\rfloor + 1. \tag{38}$$

The valid temporal-index set associated with the i-th chunk is defined as

$$\mathcal{C}_i^{(W_s)} = \{(i-1)W_s + 1,\ (i-1)W_s + 2,\ \ldots,\ \min(iW_s, L_{\text{total}})\}. \tag{39}$$

Here, W_s explicitly determines the temporal extent of each local normalization interval. The resulting normalization index sequence is defined as

$$\widetilde{\mathcal{C}}_i^{(W_s)} = \begin{cases} \mathcal{C}_i^{(W_s)}, & \left|\mathcal{C}_i^{(W_s)}\right| = W_s, \\ \mathcal{C}_i^{(W_s)} \parallel \left\{1, 2, \ldots, W_s - \left|\mathcal{C}_i^{(W_s)}\right|\right\}, & \left|\mathcal{C}_i^{(W_s)}\right| < W_s, \end{cases} \tag{40}$$

where $\parallel$ denotes ordered concatenation.

To explicitly formulate the fixed-size local normalization operation, chunk-softmax$_{W_s}(\cdot)$ denotes Softmax normalization performed independently within each temporal chunk determined by W_s. For an arbitrary full-length scalar sequence $\mathbf{z} = [z(1), z(2), \ldots, z(L_{\text{total}})]^\top$, the chunk-wise normalized value corresponding to timestamp t is defined as

$$[\text{chunk-softmax}_{W_s}(\mathbf{z})]_t = \frac{\exp(z(t))}{\sum_{t' \in \widetilde{\mathcal{C}}_{i(t)}^{(W_s)}} \exp(z(t'))}. \tag{41}$$

For each complete temporal chunk, the normalization is performed over exactly W_s consecutive temporal positions.

Within each temporal chunk, the aggregated raw structural scores are first multiplied by the negative temperature factor $-\tau$ and subsequently processed by the fixed-size chunk-wise Softmax operation.

$$\bar{s}_{\text{topo}}(t) = [\text{chunk-softmax}_{W_s}(-\tau\, \mathbf{s}_{\text{topo}})]_t\,. \tag{42}$$

Here, W_s determines the temporal extent of each local Softmax normalization interval, whereas τ controls the sharpness of the normalized structural-score distribution within each chunk. Therefore, $\bar{s}_{\text{topo}}(t)$ should be interpreted as a relative structural-deviation indicator within its local normalization region rather than as an absolute measure of temporal dependency strength.

Because chunk-softmax$_{W_s}(\cdot)$ is independently applied to different temporal chunks, normalized values are primarily comparable within their corresponding local regions, and minor baseline differences may arise across neighboring chunk boundaries. The resulting structural-deviation score $\bar{s}_{\text{topo}}(t)$ characterizes relative variations in the selected long-range temporal attention responses and is subsequently combined with the reconstruction-based score to construct the final anomaly energy.

2.7 Anomaly Detection

2.7.1 Loss Function

Within each encoder layer, the latent temporal representation is processed by Gated-Attention, residual normalization, and a position-wise feed-forward network, while CLDC Residual is additionally applied from the second encoder layer onward when

historical encoder representations are available. The resulting attention output is subsequently combined with the original input representation $\mathbf{Z}_l$ of the current encoder layer through an outer residual connection, followed by Layer Normalization:

$$\mathbf{Z}_{\text{attn}} = \text{LayerNorm}\left(\mathbf{Z}_l + \text{Dropout}\left(\mathbf{O}_{\text{attn}}\right)\right). \tag{43}$$

The resulting representation is then passed through a two-layer position-wise feed-forward network with GELU activation to perform nonlinear transformation along the latent feature dimension. A second residual connection and Layer Normalization are subsequently applied:

$$\mathbf{Z}_{l+1} = \text{LayerNorm}\left(\mathbf{Z}_{\text{attn}} + \text{FFN}\left(\mathbf{Z}_{\text{attn}}\right)\right). \tag{44}$$

Here, $\mathbf{Z}_{l+1}$ denotes the output representation of the current encoder layer and serves as the input to the subsequent encoder layer.

$$\widehat{\mathbf{X}} = \mathbf{Z}_{\text{enc}}\mathbf{W}_{\text{out}} + \mathbf{b}_{\text{out}}, \tag{45}$$

where $\widehat{\mathbf{X}} \in \mathbb{R}^{B\times L\times C}$ denotes the reconstructed standardized multivariate sequence, $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d_{\text{model}}\times C}$ denotes the output projection matrix, and $\mathbf{b}_{\text{out}} \in \mathbb{R}^{C}$ is the corresponding bias vector.

The model is trained in an unsupervised manner using normal monitoring sequences without anomaly labels. The reconstruction loss is defined as the mean squared error between the reconstructed sequence and the standardized input:

$$\mathcal{L}_{\text{rec}} = \frac{1}{BLC}\sum_{b=1}^{B}\sum_{t=1}^{L}\sum_{c=1}^{C}\left(\widehat{X}_{b,t,c} - X^{*}_{b,t,c}\right)^2. \tag{46}$$

Correspondingly, the topology-related training term is obtained by averaging the raw WAS structural scores over the batch and temporal dimensions:

$$\mathcal{L}_{\text{topo}} = \frac{1}{BL}\sum_{b=1}^{B}\sum_{t=1}^{L} s_{\text{topo}}(b,t). \tag{47}$$

The reconstruction and topology-related objectives are subsequently combined into the following joint training loss:

$$\mathcal{L}^{(e)}_{\text{total}} = \mathcal{L}_{\text{rec}} - \lambda_e\mathcal{L}_{\text{topo}}, \tag{48}$$

where λ_e denotes the epoch-dependent weight assigned to the topology-related term. The negative sign encourages larger selected long-range attention responses for normal training sequences while the model simultaneously minimizes the reconstruction error, thereby allowing reconstruction fidelity and the long-range temporal dependency responses characterized by WAS to be jointly optimized.

To reduce the influence of the topology-related objective during the initial optimization stage, its contribution is progressively increased over the first three training epochs. Specifically, the epoch-dependent weighting coefficient is defined as

$$\lambda_e = \lambda \min\left(1, \frac{e}{3}\right), \qquad e = 1, 2, \ldots, \tag{49}$$

where λ denotes the preset maximum topology-related weight and e denotes the training epoch indexed from 1.

2.7.2 Energy

During inference, reconstruction deviation provides one source of evidence for identifying anomalous temporal positions. For each temporal position within an input window, the squared reconstruction error is first computed independently over all monitoring variables, after which the mean reconstruction error over the full variable set is combined with the mean of the K largest channel-wise errors:

$$\begin{aligned} e_{\mathrm{rec}}(b,t) = (1-\omega_k)\frac{1}{C}\sum_{c=1}^{C}\left(\widehat{X}_{b,t,c} - X^*_{b,t,c}\right)^2 \\ + \omega_k \,\mathrm{Mean}\left[\mathrm{TopK}_c\left(\left(\widehat{X}_{b,t,c} - X^*_{b,t,c}\right)^2, K\right)\right]. \end{aligned} \tag{50}$$

The number of selected monitoring variables is determined according to the predefined Top-K ratio:

$$K = \max\left(1, \lfloor C r_k \rfloor\right), \tag{51}$$

where r_k denotes the Top-K ratio and $\omega_k \in [0,1]$ controls the relative contribution of the selected large-error channels.

Because adjacent input windows overlap during inference, the same timestamp on the original monitoring sequence may be covered by multiple windows and therefore receive multiple window-level reconstruction and structural scores. The temperature-controlled chunk-wise normalization introduced in the WAS module is subsequently applied to the aggregated structural scores to obtain the normalized structural-deviation score $\bar{s}_{\mathrm{topo}}(t)$.

The final point-wise anomaly energy is then constructed through multiplicative fusion:

$$\mathrm{Energy}(t) = \bar{s}_{\mathrm{topo}}(t) \cdot e_{\mathrm{rec}}(t). \tag{52}$$

This multiplicative formulation assigns relatively large anomaly energy to temporal positions exhibiting both substantial reconstruction deviation and a large normalized structural-deviation response, while reducing the resulting energy when either component remains small.

The continuous anomaly energy is finally converted into point-wise binary predictions using a fixed decision threshold θ:

$$\widehat{Y}(t) = \begin{cases} 1, & \mathrm{Energy}(t) > \theta, \\ 0, & \mathrm{Energy}(t) \le \theta. \end{cases} \tag{53}$$

Here, $\widehat{Y}(t) = 1$ denotes an anomalous temporal position, whereas $\widehat{Y}(t) = 0$ denotes a normal position. The anomaly decision threshold is determined using a predefined percentile-based calibration rule. Under the adopted point-adjusted evaluation protocol, point adjustment is performed only after the threshold-based test predictions have been obtained.

Overall, the proposed anomaly-scoring strategy integrates two complementary sources of anomaly evidence within a unified point-wise discrimination framework.

3 Experiments

3.1 Experimental Setting

3.1.1 Baselines

To comprehensively evaluate the anomaly detection performance of CLGAT, we compare it with 15 representative baseline methods covering a broad range of unsupervised multivariate time-series anomaly detection paradigms, including classical statistical and one-class methods, probabilistic and density-based models, prediction-based approaches, reconstruction and generative models, and attention-based deep architectures.

Classical statistical and one-class baselines include OCSVM [38], IsolationForest [39], and Deep-SVDD [35]. Probabilistic and density-based approaches include MMPCACD [31] and DAGMM [32]. ITAD [33] and THOC [34] learn structured latent representations for anomaly discrimination.

Prediction-based baselines consist of VAR [37], LSTM [4], and CL-MPPCA [36]. Reconstruction and generative baselines include BeatGAN [29], LSTM-VAE [30], and InterFusion [28].

In addition, Anomaly Transformer [19] is included as a strong attention-based baseline.

Together, these baselines cover conventional statistical techniques and representative deep-learning architectures with substantially different anomaly modeling mechanisms, providing a broad basis for quantitative comparison. The same Precision, Recall, and F1-score metrics are used for the subsequent quantitative comparison.

3.1.2 Evaluation Metrics

Three widely adopted metrics, namely Precision, Recall, and F1-score, are employed to quantitatively evaluate anomaly detection performance. Their mathematical definitions are given as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{54}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{55}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{56}$$

Here, TP, FP, and FN denote true positives, false positives, and false negatives, corresponding to correctly detected anomalies, normal points misclassified as anomalies, and undetected anomalies, respectively. Precision, Recall, and F1-score are jointly considered to provide a comprehensive evaluation of anomaly detection performance.

3.1.3 Implementation Details

All experiments use an input window length of $L = 100$ and $C = 38$ monitoring variables. Mini-batches use $B = 256$. Patch Embedding uses $P = 8$ and $S = 1$. The encoder has $L_{\text{enc}} = 3$ layers, $d_{\text{model}} = 128$, $M = 4$ attention heads, $D_{\text{head}} = 32$, and a feed-forward hidden dimension of $d_{\text{ff}} = 256$. The dropout rate is 0.1, and GELU is used in the feed-forward networks.

The complete framework is optimized using the Adam optimizer with an initial learning rate of 1×10^{-4}.

$$\eta_e = \eta_0 \, (0.5)^{e-1} \, , \tag{57}$$

where $\eta_0 = 1 \times 10^{-4}$ denotes the initial learning rate and e denotes the corresponding learning-rate adjustment index. The random seed used in data processing is fixed to 42 to improve the reproducibility of the experimental procedure.

For WAS, the offset bounds are $\Delta_s = 20$ and $\Delta_e = 30$. The topology-related loss coefficient is $\lambda = 10$, introduced through a three-epoch warm-up.

During inference, the Mean–Top-K score uses $r_k = 0.25$, $\omega_k = 0.40$, and $K = 9$. WAS normalization uses $\tau = 50$ and $W_s = 300$.

For anomaly discrimination, the 99.5th percentile of the validation anomaly-energy distribution is used as the threshold.

Following the segment-level point-adjust strategy commonly adopted in multivariate time-series anomaly detection [16, 34, 40], point adjustment is applied only after the threshold-based test predictions have been obtained. Specifically, once at least one timestamp within a ground-truth anomalous segment is successfully detected, the complete anomalous segment is treated as detected for the adjusted evaluation. The ground-truth test labels are used at the final evaluation stage to compute the reported Precision, Recall, and F1-score. This evaluation procedure is used to report the final detection performance of CLGAT.

3.2 Main Results

To evaluate the anomaly detection performance of the proposed CLGAT, we compare it with 15 representative baseline methods on the SMD dataset. The quantitative comparison results are summarized in Table 1.

Table 1 shows that CLGAT achieves 94.59% Precision, 96.04% Recall, and 95.31% F1-score. Deep baselines such as Deep-SVDD, LSTM, and OmniAnomaly outperform the classical methods, but their F1-scores remain below that of CLGAT.

Among the competing approaches, Anomaly Transformer achieves the strongest baseline performance, with a Precision of 89.40%, a Recall of 95.45%, and an F1-score of 92.33%. These results indicate that the proposed framework provides complementary numerical and structural information for anomaly discrimination while maintaining a favorable balance between Precision and Recall.

Table 1: Quantitative comparison of CLGAT with 15 baseline methods on the SMD dataset. P, R, and F1 denote Precision, Recall, and F1-score, respectively.

Method	**P**	**R**	**F1**
OCSVM	0.4434	0.7672	0.5619
IsolationForest	0.4231	0.7329	0.5364
DAGMM	0.6730	0.4989	0.5730
Deep-SVDD	0.7854	0.7967	0.7910
MMPCACD	0.7120	0.7928	0.7502
VAR	0.7835	0.7026	0.7408
CL-MPPCA	0.8236	0.7607	0.7909
ITAD	0.8622	0.7371	0.7948
BeatGAN	0.7290	0.8409	0.7810
LSTM	0.7855	0.8528	0.8178
LSTM-VAE	0.7576	0.9008	0.8230
InterFusion	0.8702	0.8543	0.8622
THOC	0.7976	0.9095	0.8499
OmniAnomaly	0.8368	0.8682	0.8522
Anomaly Transformer	0.8940	0.9545	0.9233
Ours	**0.9459**	**0.9604**	**0.9531**

To further examine how reconstruction information and WAS-derived structural information contribute to the final anomaly energy, two representative 100-timestamp windows extracted from the SMD test sequence are visualized in Figure 6.

As illustrated in Figure 6, the reconstruction and structural branches exhibit distinct but complementary response characteristics. Importantly, the raw WAS score in the third column represents the selected long-range attention response before sign inversion and normalization and should therefore not be directly interpreted as a structural-deviation score.

According to the WAS formulation, the aggregated raw structural score is first multiplied by a negative temperature coefficient and then normalized through fixed-size chunk-wise Softmax. Therefore, the final anomaly energy jointly reflects numerical reconstruction deviation and relative variations in selected long-range temporal attention responses rather than relying on either source independently.

The second representative window further illustrates this complementary behavior over a more extended temporal interval. The visualization therefore illustrates how numerical reconstruction information and long-range structural response information are jointly integrated within the proposed anomaly-scoring mechanism.

The last two columns of Figure 6 further illustrate the effect of the temperature coefficient on the fusion-score distribution. This behavior is consistent with the role of τ in controlling the sharpness of chunk-wise Softmax normalization without modifying the underlying reconstruction score itself.

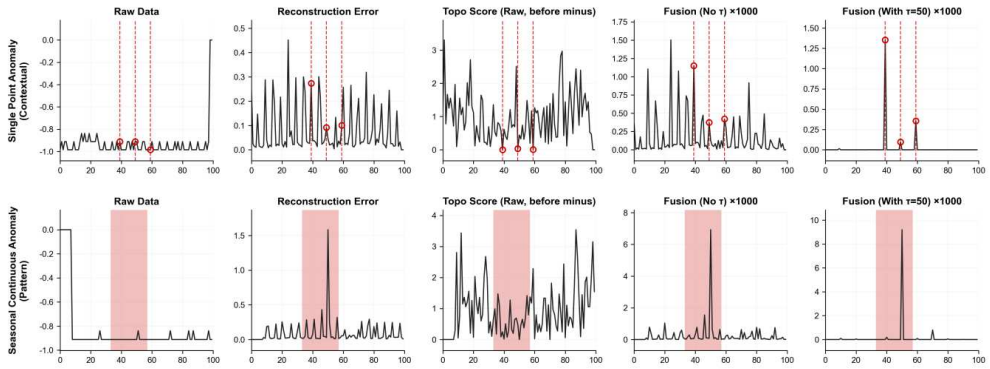


Fig. 6: Visualization of anomaly-score responses in two representative SMD test windows. From left to right, the columns display a selected standardized monitoring channel, the Mean–Top-K reconstruction score, the raw WAS structural score before sign inversion, the baseline fusion score labeled "No τ" in the figure and implemented with $\tau = 1$, and the temperature-scaled fusion score with $\tau = 50$. In the first row, red dashed lines and circles indicate prominent response peaks automatically selected from the final fusion score. In the second row, the red shaded area denotes a predefined temporal interval used for qualitative visualization. Fusion scores are multiplied by 10^3 for visualization.

Overall, the quantitative comparison and qualitative visualization provide complementary evidence for the effectiveness of CLGAT on the SMD benchmark.

3.3 Ablation Studies

To evaluate the individual contributions of the four core components of CLGAT and examine their behavior under joint integration, we conduct a series of controlled ablation experiments on the SMD dataset. This design allows us to assess the contribution of each component within the complete framework and to further examine whether their joint integration provides benefits beyond those obtained by individual components.

Table 2 shows that the complete model reaches 0.9459 Precision, 0.9604 Recall, and 0.9531 F1-score. Removing WAS causes the largest drop, from 0.9531 to 0.8518 F1-score. Removing CLDC Residual, Gated-Attention, and Patch Embedding reduces F1-score to 0.9147, 0.9213, and 0.9283, respectively. Among single-component variants, WAS alone reaches 0.9094 F1-score, followed by Patch Embedding (0.8408), Gated-Attention (0.7612), and CLDC Residual (0.7351). These results show that the four components complement one another.

This behavior is consistent with the progressive processing pipeline of CLGAT. The consistent performance degradation observed after removing any individual component, together with the substantially lower performance of the single-component variants, further supports the complementary roles of the four proposed components within the complete framework.

Table 2: Ablation study of the proposed CLGAT components on the SMD dataset. The best results are highlighted in bold.

CLDC Residual	Gated-Attention	Patch Embedding	WAS	Precision	Recall	F1-score
×	✓	✓	✓	0.9304	0.9041	0.9147
✓	×	✓	✓	0.9215	0.9211	0.9213
✓	✓	×	✓	0.9280	0.9286	0.9283
✓	✓	✓	×	0.8520	0.8515	0.8518
×	×	×	✓	0.9069	0.9119	0.9094
×	×	✓	×	0.8408	0.8407	0.8408
×	✓	×	×	0.7614	0.7609	0.7612
✓	×	×	×	0.7351	0.7350	0.7351
✓	✓	✓	✓	**0.9459**	**0.9604**	**0.9531**

Overall, the ablation results demonstrate the positive contributions of CLDC Residual, Gated-Attention, Patch Embedding, and WAS to CLGAT.

3.4 Parameter Analysis

To systematically investigate the sensitivity of CLGAT to its major hyperparameters, we conduct controlled experiments on five key parameters associated with Patch Embedding, the joint training objective, and WAS, namely the block size P, sliding step S, loss balancing coefficient λ, local Softmax chunk size W_s, and temperature scaling factor τ.

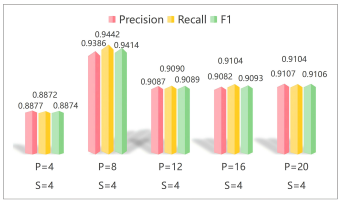
For each target hyperparameter, the remaining model configuration, training settings, and evaluation protocol are kept unchanged unless otherwise specified. Precision, Recall, and F1-score are recorded for all candidate settings to characterize the corresponding performance variation. These sensitivity experiments are used to characterize the response of CLGAT to parameter variations and do not alter the adopted configuration used for the final evaluation.

Choices of block size P and sliding step S.

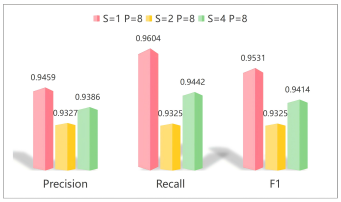
The block size P and sliding step S are the two main parameters controlling the temporal-block construction process in Patch Embedding. Similarly, a smaller S produces more densely overlapping blocks and provides more continuous temporal coverage before the interpolation stage, whereas a larger S reduces the number of generated patches and consequently decreases computational overhead.

To examine the influence of P, we first fix the sliding step at $S = 4$ and evaluate multiple candidate block sizes. As shown in Figure 7a, the detection metrics improve when P increases from relatively small values to an appropriate range. Among the evaluated settings, $P = 8$ exhibits the strongest overall performance when $S = 4$, illustrating that the model benefits from an intermediate temporal block size rather than either excessively short or overly broad local contexts.

After fixing $P = 8$, we further evaluate different values of S. As illustrated in Figure 7b, reducing the sliding step increases the overlap between adjacent temporal blocks and produces denser patch-level representations. These results illustrate the sensitivity of Patch Embedding to the temporal block size and sliding step around the adopted configuration.



(a) Detection performance with fixed $S = 4$ and varying block size P.



(b) Detection performance with fixed $P = 8$ and varying sliding step S.

Fig. 7: Sensitivity of Precision, Recall, and F1-score to the block size P and sliding step S of Patch Embedding on the SMD dataset. The two parameters are examined sequentially by fixing one parameter while varying the other.

Choice of loss balancing coefficient λ.

The coefficient λ controls the relative contribution of the topology-related term in the joint training objective and therefore determines the balance between reconstruction learning and structural-information modeling.

Figure 8 shows that the detection performance improves as λ increases from relatively small values to a moderate range. When the topology-related term receives an excessively large optimization weight, the balance between reconstruction and structural learning is weakened, which can adversely affect the final anomaly score derived from their joint contribution.

Among the evaluated configurations, $\lambda = 10$ exhibits the strongest overall F1-score and provides a favorable balance between the two training objectives. The observed trend illustrates that the model is sensitive to the relative weighting between reconstruction learning and the topology-related objective, while neither an excessively weak nor an excessively strong topology-related contribution is favorable under the current experimental setting.

Choice of local Softmax chunk size W_s.

The local Softmax chunk size W_s determines the temporal range within which the WAS structural responses are normalized. Different values of W_s consequently change the reference range used to transform raw WAS responses into normalized structural-response weights.

When W_s is relatively small, each Softmax operation is performed over a narrow temporal interval.

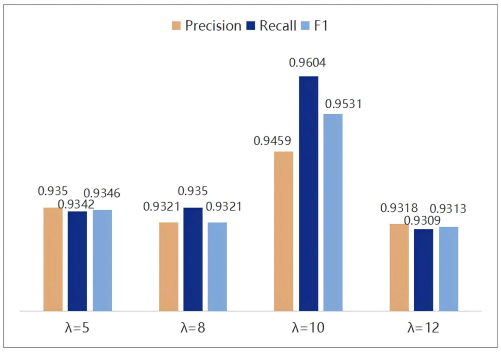


Fig. 8: Sensitivity of Precision, Recall, and F1-score to different values of the loss balancing coefficient λ on the SMD dataset.

However, an excessively large chunk is not necessarily beneficial. Performance first improves as the chunk size increases from relatively small values and then decreases after the normalization range becomes overly broad.

Among the evaluated candidates, $W_s = 300$ exhibits the strongest overall F1-score.

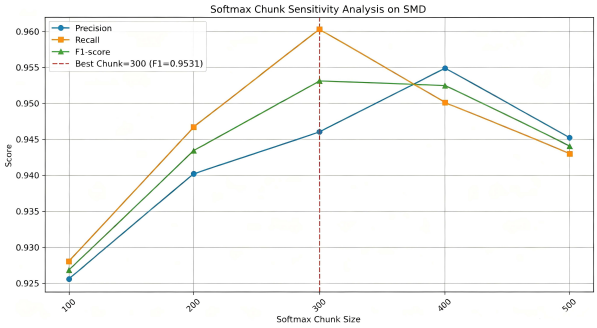


Fig. 9: Sensitivity of Precision, Recall, and F1-score to different local Softmax chunk sizes W_s on the SMD dataset.

Choice of temperature scaling factor τ.

While W_s defines the temporal reference range of local normalization, the temperature scaling factor τ controls the sharpness of the normalized WAS structural-response distribution within each chunk.

When τ is relatively small, differences among the Softmax inputs are less pronounced, and the resulting normalized distribution tends to be smoother. This enhances the contrast of the structural weighting applied to the reconstruction score.

Nevertheless, an excessively large temperature factor can produce an overly concentrated Softmax distribution. Therefore, τ needs to be sufficiently large to provide meaningful structural contrast but not so large that the distribution becomes excessively sharp.

As shown in Figure 10, the overall detection performance increases when τ is raised from relatively small values to an appropriate range, whereas further enlargement leads to performance degradation.

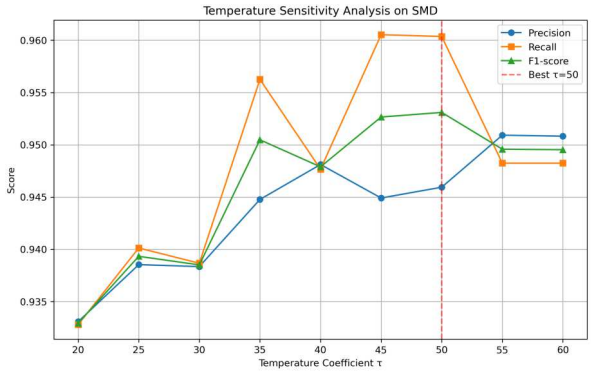


Fig. 10: Sensitivity of Precision, Recall, and F1-score to different temperature scaling factors τ for WAS structural-response normalization on the SMD dataset.

Overall, the sensitivity experiments examine variations around the adopted hyperparameter configuration $P = 8$, $S = 1$, $\lambda = 10$, $W_s = 300$, and $\tau = 50$ on the SMD dataset. These experiments therefore characterize the sensitivity of the proposed framework to the corresponding parameter variations without altering the adopted model configuration.

4 Conclusion

We proposed CLGAT for unsupervised multivariate server time-series anomaly detection. Patch Embedding preserves local temporal context while restoring the original temporal resolution, CLDC Residual improves cross-layer information reuse, Gated-Attention adaptively modulates temporal responses, and WAS combines structural information with reconstruction error. On SMD, CLGAT achieved 94.59% Precision, 96.04% Recall, and 95.31% F1-score, outperforming 15 baselines. Future work will test CLGAT on additional datasets and limited-supervision settings and will adapt its structural constraints and threshold calibration to changing operating conditions.

Declarations

- Funding: This research was supported by the Natural Science Foundation of Chongqing (No. cstb2023nscq-msx0374; cstb2024nscqlzx0093, Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJZD-K202501104)).

- Conflict of interest/Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- Ethics approval and consent to participate: Not applicable.
- Consent for publication: Not applicable.
- The raw SMD dataset txt files are publicly available at https://www.nbsdc.cn/general/dataLinks/16666.11.nbsdc.PvcdrWG1.
- Author contribution: **Jing Quan:** Writing – review & editing, Supervision, Resources, Project administration, Methodology. **Can Zeng:** Writing – original draft, Visualization, Validation, Soft-ware, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ting Xie:** Project administration, Formal analysis, Supervision.

References

[1] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, Jun. 2000.

[2] A. Giannoulidis, A. Gounaris, N. Nikolaidis, A. Naskos, and D. Caljouw, "Investigating thresholding techniques in a real predictive maintenance scenario," *SIGKDD Explor. Newsl.*, vol. 24, no. 2, pp. 86–95, Dec. 2022.

[3] W. Liu, L. Yan, N. Ma, G. Wang, X. Ma, P. Liu, and R. Tang, "Unsupervised deep anomaly detection for industrial multivariate time series data," *Appl. Sci.*, vol. 14, no. 2, Art. no. 774, 2024.

[4] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, London, U.K., 2018, pp. 387–395.

[5] G. Pang, L. Cao, and C. Aggarwal, "Deep learning for anomaly detection: challenges, methods, and opportunities," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, Virtual Event, Israel, 2021, pp. 1127–1130.

[6] Y. Yao and Z.-X. Yang, "DDConv: Disentangled dual branch convolutional network for anomaly detection in server machines," *IEEE Internet Things J.*, vol. 12, no. 18, pp. 37327–37338, 2025.

[7] Y. Gu, Y. Lu, and D. Yan, "Anomaly detection of server data based on depthwise separable convolution and Transformer," in *Proc. IEEE 7th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, 2024, pp. 1814–1818.

[8] T. Amarbayasgalan, V. H. Pham, N. Theera-Umpon, and K. H. Ryu, "Unsupervised anomaly detection approach for time-series in multi-domains using deep reconstruction error," *Symmetry*, vol. 12, no. 8, Art. no. 1251, 2020.

[9] Y. Wang and J. Li, "Anomaly detection method for time series data based on Transformer reconstruction," in *Proc. 12th Int. Conf. Inform., Environ., Energy Appl.*, Singapore, 2023, pp. 58–63.

[10] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," *Proc. VLDB Endow.*, vol. 15, no. 6, pp. 1201–1214, Feb. 2022.

[11] C. Pan, L. Su, L. Xiong, J. Yang, and F. Li, "CT-DDPM: Anomaly detection of multivariate time series with copula and transformer-based denoising diffusion probabilistic models," *Inf. Sci.*, vol. 717, Art. no. 122279, 2025.

[12] H. Qiao, D. Oliveira, and D. Wu, "A novel criterion of reconstruction-based anomaly detection for sparse-binary data," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[13] S. Deng, Z. Sun, R. Zhuang, and J. Gong, "Noise-to-norm reconstruction for industrial anomaly detection and localization," *arXiv preprint arXiv:2307.02836*, 2023.

[14] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: UnSupervised anomaly detection on multivariate time series," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Virtual Event, CA, USA, 2020, pp. 3395–3404.

[15] B. R. Chang, H.-F. Tsai, and G.-R. Chen, "Self-adaptive server anomaly detection using ensemble meta-reinforcement learning," *Electronics*, vol. 13, no. 12, Art. no. 2348, 2024.

[16] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Anchorage, AK, USA, 2019, pp. 2828–2837.

[17] H. Zhao *et al.*, "Multivariate time-series anomaly detection via graph attention network," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2020, pp. 841–850.

[18] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time-series anomaly detection in IoT," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9179–9189, 2022.

[19] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[20] W. Zhou, S. Zhou, Y. Cao, J. Yang, and H. Liu, "Unsupervised anomaly detection method for electrical equipment based on audio latent representation and parallel

attention mechanism," *Appl. Sci.*, vol. 15, no. 15, Art. no. 8474, 2025.

[21] Y. Fan, Y. Yu, W. Lu, and Y. Han, "Weakly-supervised video anomaly detection with snippet anomalous attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 7, pp. 5480–5492, Jul. 2024.

[22] A. Varghese *et al.*, "Temporal autoencoder architectures with attention for ECG anomaly detection," *Int. J. Business Intelligence and Data Mining*, vol. 24, no. 2, pp. 146–159, 2024.

[23] H. Takimoto, J. Seki, S. F. Situju, and A. Kanagawa, "Anomaly detection using siamese network with attention mechanism for few-shot learning," *Applied Artificial Intelligence*, vol. 36, no. 1, Art. no. 2094885, 2022.

[24] G. Xie, X. Hu, Y. Sun, W. Zhang, Y. Bu, H. Fu, and X. Xu, "Deviation-guided attention for semi-supervised anomaly detection with contrastive regularisation," *CAAI Trans. Intell. Technol.*, vol. 11, no. 1, pp. 66–82, 2026.

[25] L. Su, Q. Li, J. Quan, and F. Li, "Enhanced recurrent convolutional encoding with attention-based representation learning for chaotic time series anomaly detection," *Physica Scripta*, vol. 100, no. 11, Art. no. 115215, Nov. 2025.

[26] Y. Yang, C. Zhang, T. Zhou, Q. Wen, and L. Sun, "DCdetector: Dual attention contrastive representation learning for time series anomaly detection," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, Long Beach, CA, USA, 2023, pp. 3033–3045.

[27] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, Art. no. 56, 2021.

[28] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, and D. Pei, "Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding," in *Proc. 27th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Virtual Event, Singapore, 2021, pp. 3220–3230.

[29] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "BeatGAN: Anomalous rhythm detection using adversarially generated time series," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 4433–4439.

[30] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.

[31] T. Yairi, N. Takeishi, T. Oda, Y. Nakajima, N. Nishimura, and N. Takata, "A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction," *IEEE Trans. Aerosp.*

Electron. Syst., vol. 53, no. 3, pp. 1384–1401, Jun. 2017.

[32] B. Zong *et al.*, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[33] Y. Shin *et al.*, "ITAD: Integrative tensor-based anomaly detection system for reducing false positives of satellite systems," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Virtual Event, Ireland, 2020, pp. 2733–2740.

[34] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 13016–13026.

[35] L. Ruff *et al.*, "Deep one-class classification," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80, 2018, pp. 4393–4402.

[36] S. Tariq *et al.*, "Detecting anomalies in space using multivariate convolutional LSTM with mixtures of probabilistic PCA," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Anchorage, AK, USA, 2019, pp. 2123–2133.

[37] C. A. Sims, "Macroeconomics and reality," *Econometrica*, vol. 48, no. 1, pp. 1–48, Jan. 1980.

[38] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[39] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.

[40] H. Xu *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in *Proc. World Wide Web Conf. (WWW)*, Lyon, France, 2018, pp. 187–196.