DART-GNN: A Dynamic Recurrent Graph Neural Network for Multivariate Time Series Anomaly Detection

Yuan Li, Yuhang Zhou, Qingzhong Yan, Xiang Wu, and Yuming Bo

***Abstract*—Detecting anomalies in multivariate time series is critical for the safety and reliability of complex cyber-physical systems. Graph Neural Networks (GNNs) have shown great promise in this area by explicitly modeling the relational structure between sensors to improve detection. However, the performance of most GNNs is constrained by their reliance on static graphs, which are unable to capture the evolving nature of relationships between sensors in dynamic environments. To address this limitation, we propose the Dynamic Attention Recurrent Two-step Graph Neural Network (DART-GNN). Our framework constructs a time-specific dependency graph by first using a Gated Recurrent Unit (GRU) to encode temporal context and then applying a self-attention mechanism to infer relationships. A two-step Graph Attention Network (GAT) then performs deep aggregation on this dynamic graph, enabling the model to capture complex, higher-order interactions by propagating information from second-order neighbors. We validated our model's performance through rigorous experiments on the widely-used SWaT and WADI public benchmarks. The results confirm that DART-GNN achieves a new state-of-the-art, demonstrating superior performance compared to a broad spectrum of baseline methods.**

***Key Words*—Anomaly Detection, Graph Neural Network, GRU, Multivariate Time Series.**

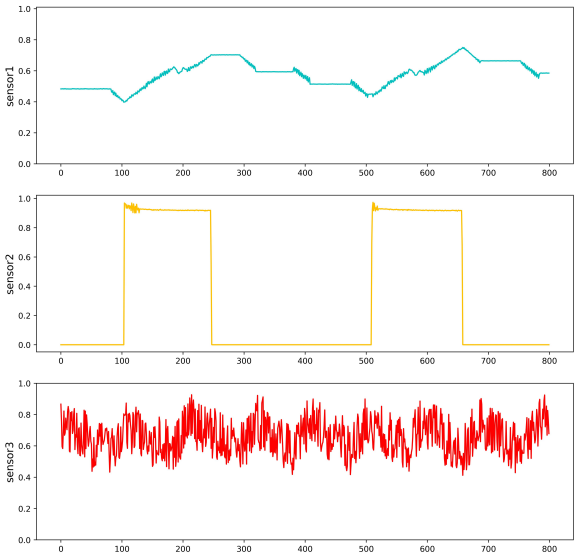


Fig. 1. An example of multivariate time series data from three different sensors.

I. Introduction

ANOMALY detection in time series [1] generated by cyber-physical systems (CPS) is a critical task for ensuring their robust and dependable operation. As these systems become more integrated with IoT devices and social structures, they produce increasingly complex data that reflects the intricate interactions between physical processes and computational controls, as illustrated in Fig. 1. Therefore, the ability to automatically identify anomalous patterns is a cornerstone for building intelligent and trustworthy CPS, as the detection of these patterns provides valuable diagnostic insight into underlying issues, from physical component wear to malicious software activity.

Given the unpredictable nature of anomalies and the scarcity of labeled data in real-world scenarios, unsupervised methods are widely employed. These approaches can be broadly categorized by their underlying strategy. Distribution-based methods aim to model the distribution of normal data in a feature space. This category includes subspace-based techniques like PCA [2], which identifies anomalies based on high reconstruction errors, and boundary-based techniques like One-Class SVMs [3], which define a frontier around the normal data cluster. In contrast, prediction-based approaches like AFMF [4] and MTAD [5] first model the temporal patterns of normal data and then flag instances where actual observations significantly deviate from the model's predictions. However, both categories often struggle in dynamic environments with shifting data distributions, highlighting the need for more adaptable solutions.

The rise of deep learning has brought about more robust analytical frameworks. Recurrent models, such as Long Short-Term Memory (LSTM) networks [6], have shown particular skill in modeling temporal dependencies within individual sensor streams. At the same time, generative models like Variational Auto-Encoders (VAE) [7] and Generative Adversarial Networks (GAN) [8] learn to replicate the distribution of normal data, identifying outliers that cannot be accurately reconstructed. A notable limitation of these advanced models, however, is that they usually process each time series in isolation, ignoring the complex network of relationships between

Manuscript received August 4, 2025; revised August 11 and August 18, 2025; accepted September 15, 2025. This article was recommended for publication by Associate Editor Shujin Qin upon evaluation of the reviewers' comments.

Y. Li, Y. Zhou, and Q. Yan are with the School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: liyuan1@njust.edu.cn; xuan18835033523@njust.edu.cn; qingzhongyan@njust.edu.cn).

X. Wu, and Y. Bo are with the School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: wuxiang1@njust.edu.cn; byming@njust.edu.cn).

Corresponding Author: Xiang Wu

different variables.

To explicitly model these complex correlations, Graph Neural Networks (GNNs) have become a key technology in multivariate anomaly detection [9]. Architectures like Graph Attention Networks (GATs) [10] have advanced beyond simpler models like Graph Convolutional Networks (GCNs) [11] by learning to weigh the importance of different neighbors during information aggregation. Nonetheless, the performance of most GNN-based methods is fundamentally constrained by a static graph assumption. These models typically infer a single, fixed graph based on static node embeddings [12], a representation that cannot adapt to the evolving operational states of a real-world system and largely disregards the valuable temporal context embedded in the time series data itself.

To address these challenges, we propose the Dynamic Attention Recurrent Two-step Graph Neural Network (DART-GNN). Our framework is designed to learn a new graph structure for each time instance directly from the data. First, a Gated Recurrent Unit (GRU) [13], chosen for its comparable performance to LSTM with greater computational efficiency and fewer parameters, processes the recent history of each sensor to produce a context-aware temporal representation. Next, a self-attention mechanism [14] uses these representations to compute a time-specific adjacency matrix, capturing the most salient relationships for that moment. Finally, a novel two-step GAT performs a deep aggregation on this dynamic graph. By propagating information across two hops, the model can capture more complex, higher-order dependencies, which is crucial for the precise identification of subtle anomalies.

To summarize, our main contributions are as follows:

1) We introduce a novel method for learning dynamic relational structures, where a GRU captures evolving sensor states to guide an attention mechanism in constructing a dependency graph at each time instance.
2) We propose a deep aggregation mechanism using a two-step GAT. This allows our model to move beyond immediate neighbors and capture complex, higher-order dependencies by integrating information from the two-hop neighborhood.
3) Our model achieves the best performance on public benchmarks compared with literature methods and provides enhanced explainability by visualizing the learned dynamic relationships.

II. Related Work

This section reviews existing methods for anomaly detection in multivariate time series, which can be broadly classified into three categories: machine learning methods, deep learning methods, and graph neural network methods.

A. Machine Learning Methods

Anomaly detection is firmly grounded in the realm of unsupervised machine learning, where its primary objective lies in the identification of irregular patterns without recourse to labeled data. Characteristically, these methodologies operate by constructing a model that encapsulates the normal behavioral patterns inherent in the underlying data distribution. Clustering-based techniques, for example, delineate anomalies as data points exhibiting a significant degree of divergence from all cluster centroids. This category encompasses prominent algorithms such as K-Means [15] and the distance-based K-Nearest Neighbor (KNN) [16]. Density-based approaches, typified by the Local Outlier Factor (LOF) [17], adhere to a comparable principle, designating data points situated within low-density regions as anomalous. Another prevalent strategy is partitioning, exemplified by the Isolation Forest [18], which isolates anomalies by leveraging their propensity to be segregated from the majority of the data. However, a major drawback of these methods is that they create a static model of what is normal. As a result, they might have difficulty in effectively detecting new types of anomalies that did not exist in the training data distribution.

B. Deep Learning Methods

Endowed with the capability to learn complex, non-linear representations from raw data, deep learning techniques have emerged as a cornerstone of modern anomaly detection. These methodologies can be broadly categorized into reconstruction-based and prediction-based approaches. Reconstruction-based models, which include AEs [19] and VAEs [20], are trained to encode normal data into a low-dimensional latent space and subsequently decode it back to its original form. Anomalies are identified based on their high reconstruction errors. GANs [8] present a more potent alternative, as they can learn the distribution of normal data with greater precision. Models such as MAD-GAN [21] have successfully adapted the GAN framework for multivariate time series by employing LSTMs as both the generator and the discriminator.

Prediction-based models capitalize on the sequential nature of time series data. RNNs [22] and LSTMs [23] excel at modeling temporal dependencies; however, their sequential processing manner may lead to inefficiency. Transformer-based models [24] tackle this inefficiency through a parallel attention mechanism, rendering them effective in capturing long-range contextual patterns. Despite their inherent strengths, a prevalent limitation of these deep learning methods is that they frequently treat each time series as an independent channel, thus failing to explicitly model the critical relational dependencies across different variables.

C. Graph Neural Networks Methods

GNNs have recently emerged as a powerful solution to directly address the limitations of previous methods, as they explicitly model the relational dependencies between variables. By treating each time series as a node, GNNs learn node representations through the core mechanisms of message passing and relational aggregation, enabling effective capture of inter-variable interactions. Among the foundational architectures in this domain, GCNs [25] stand out for their ability to perform convolutional operations over a node's neighborhood, aggregating feature information from adjacent nodes to update the target node's representation. Building on this, GAT [26] introduces an attention mechanism that dynamically assigns

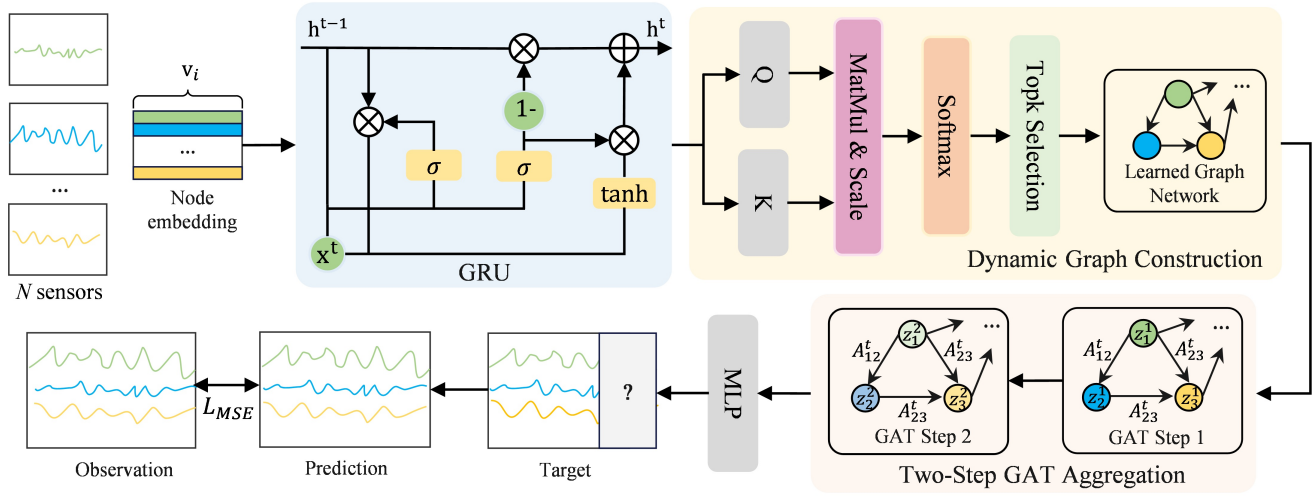


Fig. 2. An overview of DART-GNN.

different weights to neighboring nodes based on their relevance, allowing for more flexible and fine-grained feature fusion. Meanwhile, GIN [27] excels in generating node representations by iteratively aggregating multi-scale neighborhood information.

A key challenge in applying GNNs to time series data lies in the fact that the graph structure is often implicit rather than explicitly given. Early models like STG [28] use manual graph construction, impractical for unknown topologies. Advanced methods like GDN [12] integrate structural learning with GNNs, inferring relationships from node embedding similarity and using attention to capture sensor dependencies, aiding anomaly detection and interpretation. GATAMAF [29] advances this by using GATs to extract temporal features from time series nodes, merging graph relational modeling with temporal dynamics, then using masked autoregressive flow for density estimation. Yet most approaches learn a static graph, failing to adapt to evolving system dynamics or fully use temporal context. This highlights the need for dynamic graph frameworks responsive to temporal variations.

III. PROPOSED METHOD

To address the challenges of static relational models in time series, we propose a novel framework called DART-GNN as shown in Fig. 2. Our model's core objective is to be deeply aware of two key aspects of the data: the temporal evolution within individual sensor streams and the dynamically changing relationships that connect them. To achieve this, our framework integrates three distinct modules. First, a GRU module is tasked with encoding the recent history of each sensor into a meaningful, context-rich vector. Second, a dynamic graph learning module uses these vectors to infer a directed graph, representing the most salient inter-sensor dependencies at that specific moment. Finally, a multi-step graph aggregation module operates on this dynamic graph, producing robust, context-aware representations used to precisely identify anomalous behaviors.

A. Problem Formulation

The task is to analyze a multivariate time series composed of signals from N distinct sensors. This raw data is transformed into a sequence of input instances by applying a sliding window of a fixed size w. For any given time t, the model receives an input window $\mathbf{X}^{(t)} = \left[\mathbf{x}^{t-w+1}, ..., \mathbf{x}^{t}\right]$, where $\mathbf{x}^{t} = \left[x_1^t, ..., x_N^t\right]$. The model must learn the distribution of normal system behavior from this data without relying on anomaly labels. Subsequently, its objective is to produce a binary label $y(t) \in \{0, 1\}$ for each time step, where $y(t) = 1$ signifies the detection of an anomaly.

B. GRU-based Temporal Context Encoding

To derive a meaningful representation for each sensor, we combine the node embedding with its recent dynamic behavior. Each sensor i is assigned a unique, learnable embedding vector $\mathbf{v_i} \in \mathbb{R}^d$ to represent its intrinsic properties. This embedding vector is then merged with the sensor's dynamic behavior by concatenating it with the observed values within the current time window. This combined information is processed by a GRU, selected for its balance of expressive power and computational efficiency. Then the GRU iterates through the window's data to produce a final hidden state.

$$\begin{aligned} \text{input}_k &= \left[x_i^{t-w+k} \oplus \mathbf{v}_i\right] \quad \text{for } k = 1, \ldots, w \\ \mathbf{h}_i^t &= \text{GRU}\left(\text{input}_1, \ldots, \text{input}_w\right) \end{aligned} \tag{1}$$

where $\oplus$ denotes vector concatenation, and the final hidden state of the GRU, $\mathbf{h}_i^t$ serves as the context-aware representation of sensor i at time t.

C. Dynamic Graph Construction

This stage functions as a graph learning module that infers the system's relational structure in a data-driven manner. The set of all sensor contexts, $\mathbf{H}^t = [\mathbf{h}_1^t, \ldots, \mathbf{h}_N^t] \in \mathbb{R}^{N \times d_h}$, is fed into a self-attention mechanism. This mechanism treats the sensors as nodes in a potential fully-connected graph and learns the strength of the directed connection between

them based on the similarity of their contextual states. This is achieved by projecting the context vectors into Query $\mathbf{Q}$ and Key $\mathbf{K}$ spaces and calculating their scaled dot-product attention:

$$\mathbf{Q} = \mathbf{H}^t\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{H}^t\mathbf{W}_K \tag{2}$$

$$\mathbf{A}'^t = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \tag{3}$$

where $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d_h \times d_h}$ are trainable weight matrices and $\sqrt{d_k}$ ensures numerical stability. After softmax normalization, the resulting attention matrix $\mathbf{A}'^t$ contains scores representing the directed influence between any two sensors. However, a fully-connected graph is often noisy and computationally intensive. We therefore apply a sparsification step by selecting only the top-k most significant relationships for each sensor. This results in a sparse and focused adjacency matrix $\mathbf{A}^t$ that represents the most critical inter-sensor dependencies for the specific moment:

$$A^t_{ij} = \begin{cases} 1 & \text{if } A'^t_{ij} \text{ is one of the top-k value} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

D. Graph-based Anomaly Detection with Two-Step GAT Aggregation

To fully leverage the learned graph, a deep aggregation strategy is required, as anomalies can arise from complex, cascading effects that are invisible to a sensor's immediate neighbors. A deeper aggregation is necessary to model these multi-hop dependencies, so we employ a two-step GAT process.

The first propagation step operates on the initial sensor features $\mathbf{x}_i^{(t)}$. To ensure the attention calculation is aware of both the node embedding and its dynamic state, we first create a fused representation $\mathbf{g}_i^{(t,1)}$ by concatenating the static embedding $\mathbf{v}_i$ with the transformed features:

$$\mathbf{g}_i^{(t,1)} = \mathbf{v}_i \oplus \mathbf{W}^{(1)}\mathbf{x}_i^{(t)} \tag{5}$$

where $\mathbf{W}^{(1)}$ is the learnable weight matrix for the first step. This fused representation is then used to compute the attention scores $\pi^{(t,1)}(i,j)$:

$$\pi^{(t,1)}(i,j) = \text{LeakyReLU}(\mathbf{a}^{(1)T}[\mathbf{g}_i^{(t,1)} \oplus \mathbf{g}_j^{(t,1)}]) \tag{6}$$

where $\mathbf{a}^{(1)}$ is the attention vector for the first step. The scores are normalized via the softmax function across the learned neighborhood A_i^t to get the final coefficients $\alpha_{ij}^{(t,1)}$, which are used to produce the intermediate representation $\mathbf{z}_i^{(t,1)}$:

$$\alpha_{ij}^{(t,1)} = \frac{\exp(\pi^{(t,1)}(i,j))}{\sum_{k \in A_i^t \cup \{i\}} \exp(\pi^{(t,1)}(i,k))} \tag{7}$$

$$\mathbf{z}_i^{(t,1)} = \text{ReLU}\left(\sum_{j \in A_i^t \cup \{i\}} \alpha_{ij}^{(t,1)} \mathbf{W}^{(1)} \mathbf{x}_j^{(t)} \right) \tag{8}$$

The second step takes the intermediate representations $\mathbf{z}_i^{(t,1)}$, which already contain rich, localized contexts as input. This step allows information to propagate across a two-hop distance, providing a much wider receptive field. It computes a new set of attention coefficients and generates the final node representations $\mathbf{z}_i^{(t,2)}$:

$$\mathbf{g}_i^{(t,2)} = \mathbf{W}^{(2)}\mathbf{z}_i^{(t,1)} \tag{9}$$

$$\pi^{(t,2)}(i,j) = \text{LeakyReLU}(\mathbf{a}^{(2)T}[\mathbf{g}_i^{(t,2)} \oplus \mathbf{g}_j^{(t,2)}]) \tag{10}$$

$$\alpha_{ij}^{(t,2)} = \frac{\exp(\pi^{(t,2)}(i,j))}{\sum_{k \in A_i^t \cup \{i\}} \exp(\pi^{(t,2)}(i,k))} \tag{11}$$

$$\mathbf{z}_i^{(t,2)} = \text{ReLU}\left(\sum_{j \in A_i^t \cup \{i\}} \alpha_{ij}^{(t,2)} \mathbf{W}^{(2)} \mathbf{z}_j^{(t,1)} \right) \tag{12}$$

where $\mathbf{a}^{(2)}$ and $\mathbf{W}^{(2)}$ are the learnable parameters for the second step.

The final representations, $\mathbf{Z}^t = \{\mathbf{z}_1^{(t,2)}, \ldots, \mathbf{z}_N^{(t,2)}\}$, are used for the forecasting task. Each node's representation is fused with its static embedding before being processed by a feed-forward network to predict the sensor values at the next time step $\hat{\mathbf{x}}^{t+1}$:

$$\hat{\mathbf{x}}^{t+1} = f_\theta([\mathbf{v}_1 \circ \mathbf{z}_1^{(t,2)}, \cdots, \mathbf{v}_N \circ \mathbf{z}_N^{(t,2)}]) \tag{13}$$

where $\circ$ represents the element-wise product and f_θ is a multi-layer perceptron that acts as the output regressor.

E. Model Training and Inference

The model is trained via an end-to-end optimization process with a forecasting-based objective. The parameters of the entire DART-GNN are adjusted to minimize the Mean Squared Error (MSE) between the model's predictions and the actual future sensor readings. The goal is for the model to become an expert on the system's normal operational patterns, such that its predictions for normal data are highly accurate.

$$L_{\text{MSE}} = \frac{1}{T_{\text{train}} - w} \sum_{t=w}^{T_{\text{train}}-1} ||\hat{\mathbf{x}}^{t+1} - \mathbf{x}^{t+1}||_2^2 \tag{14}$$

where T_{train} is the total number of time ticks in the training data.

During inference, this expertise is used to detect anomalies. When a true anomaly occurs, the sensor's actual behavior will deviate from the model's prediction of normal behavior. The magnitude of this prediction error is therefore a strong indicator of anomalous activity. We calculate a normalized deviation score for each sensor to account for different scales and signal variances. The final anomaly score for a given time step is the maximum deviation observed across all sensors, ensuring that an anomaly flagged on even a single critical node is captured as a system-level event.

$$\begin{aligned} \text{Err}_i(t+1) &= |x_i^{t+1} - \hat{x}_i^{t+1}| \\ \text{Score}(t+1) &= \max_i \left(\frac{\text{Err}_i(t+1) - \mu_i}{\sigma_i} \right) \end{aligned} \tag{15}$$

A time step is then classified as anomalous if its score exceeds a pre-determined threshold.

Algorithm 1 Training and Inference for DART-GNN

Training Procedure
Input: Training data $\mathbf{X}_{train}$, validation data $\mathbf{X}_{val}$, window size w, epochs E, learning rate η, top-k value k.
Output: Trained model parameters Θ, error normalization stats $\boldsymbol{\mu}, \boldsymbol{\sigma}$.
Initialize all model parameters $\Theta = \{\theta_{GRU}, \ldots, \theta_{MLP}\}$.
for $e = 1$ to E **do**
  for each time step t from w to T_{train} **do**
    Let $\mathbf{X}^{(t)}$ be the input window.
    Compute context vectors $\mathbf{H}^t = \text{GRU}(\mathbf{X}^{(t)}, \mathbf{v})$.
    Compute adjacency matrix $\mathbf{A}^t = \text{TopK}(\text{Attn}(\mathbf{H}^t))$.
    $\mathbf{Z}^{(t,1)} \leftarrow \text{GAT}_{\text{step1}}(\mathbf{X}^{(t)}, \mathbf{A}^t, \mathbf{v})$.
    $\mathbf{Z}^{(t,2)} \leftarrow \text{GAT}_{\text{step2}}(\mathbf{Z}^{(t,1)}, \mathbf{A}^t)$.
    Predict next step $\hat{\mathbf{x}}^{t+1} = f_\theta(\mathbf{Z}^{(t,2)}, \mathbf{v})$.
    Calculate loss $L_{MSE} = ||\hat{\mathbf{x}}^{t+1} - \mathbf{x}^{t+1}||_2^2$.
    Update parameters $\Theta \leftarrow \Theta - \eta \nabla_\Theta L_{MSE}$.
  end for
end for
Use trained Θ to get prediction errors $\mathbf{Err}_{val}$ on $\mathbf{X}_{val}$.
For each sensor i, calculate μ_i and σ_i from $\mathbf{Err}_{val,i}$.
return trained parameters Θ and stats $\boldsymbol{\mu}, \boldsymbol{\sigma}$.

Inference Procedure
Input: Trained model Θ, stats $\boldsymbol{\mu}, \boldsymbol{\sigma}$, test data $\mathbf{X}_{test}$, anomaly threshold τ.
Output: Anomaly labels $\mathbf{Y}_{test}$.
for each time step t from w to T_{test} **do**
  Let $\mathbf{X}^{(t)}$ be the input window.
  Predict $\hat{\mathbf{x}}^{t+1}$ using the forecasting process.
  Calculate error $Err_i(t+1)$ for each sensor i.
  Calculate error $Err_i(t+1) = |x_i^{t+1} - \hat{x}_i^{t+1}|$ for each sensor.
  Calculate score $S(t+1) = \max_i \frac{Err_i(t+1) - \mu_i}{\sigma_i}$.
  Assign label $y(t+1)$ based on $S(t+1) > \tau$.
end for
return anomaly labels $\mathbf{Y}_{test}$.

IV. Experiment

A. Datasets

Our model's performance is evaluated on two widely used public benchmark datasets for time series anomaly detection as shown in Table. I, both derived from real-world water treatment test-beds.

The SWaT dataset [30] is generated from a scaled-down yet realistic water treatment plant, representing a modern CPS. Its training data includes recordings of two weeks of normal system operation, while the testing data incorporates multiple controlled physical attack scenarios as ground truth anomalies. This dataset comprises 51 distinct sensor variables.

As an extension of SWaT, the WADI dataset [31] is sourced from a more complex and larger-scale water distribution network. It represents a more comprehensive water treatment, storage, and distribution system with a higher complexity level, featuring 127 sensor variables—making it significantly more high-dimensional than SWaT.

TABLE I General Information about Dataset

Dataset	SWaT	WADI
Dimension	51	123
Training Size	496800	1048571
Testing Size	449919	172801
Anomalies	11.98%	5.99%

B. Implementation Details

All experiments are performed on a server outfitted with an NVIDIA RTX 4090 GPU. The Adam optimizer is employed to train our model, with a learning rate set to 0.001. A fixed sliding window size of 15 is adopted across all experiments. Regarding dataset-specific hyperparameters, the node embedding dimension is configured as 64 for the SWaT dataset and 128 for the more complex WADI dataset. Correspondingly, in the dynamic graph construction phase, the number of top-k neighbors selected is set to 15 for SWaT and 30 for WADI. To guarantee robust training, our models are trained for a maximum of 100 epochs, and an early stopping mechanism with a patience of 15 epochs is implemented to mitigate overfitting.

C. Evaluation Metrics

To quantitatively evaluate the performance of our model, three standard classification metrics are employed: Precision, Recall, and F1-score. Precision refers to the proportion of correctly identified anomalies within all instances classified as anomalous. Recall, alternatively termed sensitivity, quantifies the proportion of all actual anomalies that are successfully detected by the model. The F1-score, calculated as the harmonic mean of Precision and Recall, serves as a comprehensive metric that balances the trade-off between false positives and false negatives. The formal definitions of these metrics are as follows:

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{F}_1 &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \end{aligned} \tag{16}$$

where TP, FP, and FN represent the counts of true positives, false positives, and false negatives, respectively.

D. Baseline Comparison

Our model's performance is compared against nine diverse baseline methods. These include the classical PCA [2] and the prediction-based LSTM [6]. Additionally, several reconstruction-based approaches are evaluated: the hybrid LSTM-VAE [32]; DAGMM [33], which incorporates a Gaussian Mixture Model; the adversarially trained autoencoder USAD [34]; and the GAN-based MAD-GAN [21]. Finally, we include three recent state-of-the-art models: the graph-based TopoGDN [35], the Transformer-based Graphformer [36], and GATAMAF [37], which combines a graph attention network with an autoregressive flow model.

TABLE II Comparison with baselines on SWaT and WADI datasets

Method	SWaT			WADI		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**
PCA	47.12	44.23	45.63	38.26	18.22	24.68
LSTM	59.45	52.76	55.91	72.42	27.93	40.43
LSTM-VAE	95.40	59.49	73.28	45.87	32.12	37.78
DAGMM	70.31	47.13	56.43	54.91	28.69	37.68
MAD-GAN	93.33	62.45	74.83	40.56	38.73	39.62
USAD	96.35	64.46	77.24	86.32	27.87	42.13
TopoGDN	92.01	65.04	76.21	58.08	**44.85**	50.59
Graphformer	97.02	65.48	78.19	68.25	42.41	52.31
GATAMAF	96.86	67.57	79.61	78.51	44.23	56.58
DART-GNN	**98.02**	**68.44**	**80.60**	**90.63**	42.76	**58.11**

As shown in Table. II, our proposed DART-GNN model demonstrates superior performance across both the SWaT and WADI datasets. On the SWaT dataset, DART-GNN achieves the highest F1-score at 80.60%, surpassing all nine baseline methods. Notably, its performance exceeds that of the strongest baseline, GATAMAF, and it also attains the highest scores in both Precision at 98.02% and Recall at 68.44%. This balanced and leading performance highlights our model's effectiveness at both accurately identifying anomalies and capturing a high proportion of true fault events. The model's capabilities are further validated on the more complex WADI dataset. Here, DART-GNN again achieves the highest F1-score at 58.11%, marking a significant improvement over all competitors. This result is driven by an exceptionally high Precision of 90.63%, the best among all methods, which demonstrates the model's robustness in a high-dimensional environment where it can precisely identify anomalies without being overwhelmed by false positives. These comprehensive results validate that our approach of combining dynamic graph learning with deep aggregation effectively models the complex relationships between sensors, leading to more accurate and reliable anomaly detection.

E. Ablation Study

TABLE III Ablation Study on Model Components

Method	SWaT			WADI		
	Prec	Rec	F1	Prec	Rec	F1
DART-GNN	**98.02**	**68.44**	**80.60**	**90.63**	**42.76**	**58.11**
w/o Dynamic Graph	95.23	58.12	72.18	85.40	33.05	47.65
w/o GRU	96.18	64.03	76.88	88.27	38.52	53.63
w/o Two-Step GAT	97.51	64.35	77.53	89.15	41.48	56.62

To verify the individual contribution of our model's key architectural components, we perform a comprehensive ablation study. In this analysis, we compare the full DART-GNN framework against three distinct variants, each with a crucial module disabled: (1) a version reliant on a static graph instead of a dynamic one (**w/o Dynamic Graph**), (2) a version without the GRU-based temporal encoding module (**w/o GRU**), and (3) a version using a conventional single-step GAT (**w/o Two-Step GAT**).

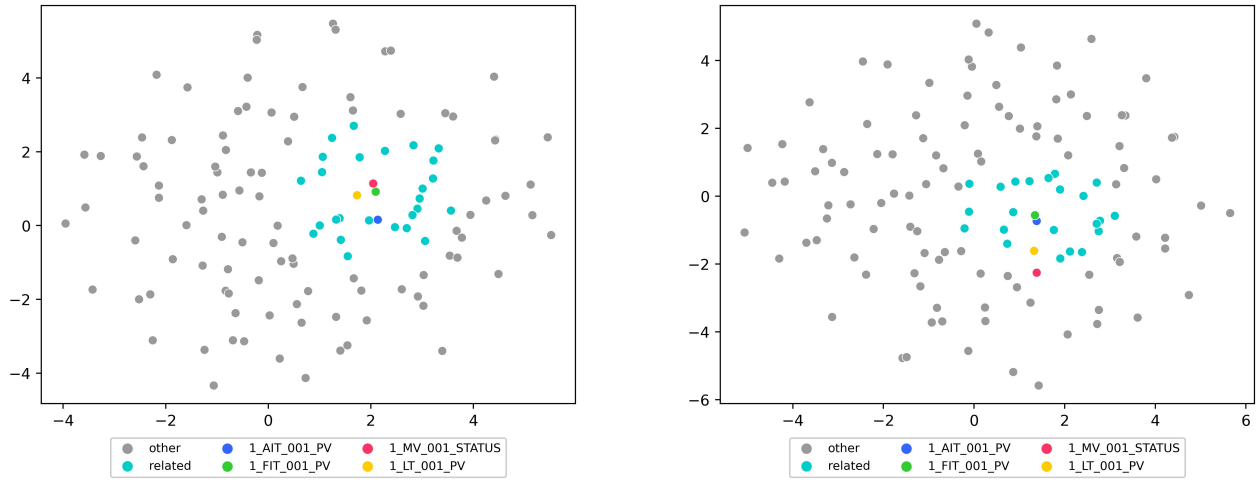
As shown in Table. III, the complete DART-GNN model consistently outperforms all its variants, confirming that each component contributes positively to the final performance. First, the importance of the dynamic graph is highlighted when comparing DART-GNN to the variant with a static graph. The w/o Dynamic Graph variant shows the most substantial performance degradation across all metrics, with its F1-score dropping by 8.42% on SWaT and a significant 10.46% on WADI. This result strongly supports our core hypothesis that a static graph is insufficient for modeling systems where relationships between sensors evolve, and adapting the graph structure at each instance is crucial for accurate detection. Second, removing the GRU module also leads to a notable decline in performance. Without the GRU to encode temporal context, the model's F1-score falls to 76.88% on SWaT and 53.63% on WADI. This demonstrates that capturing temporal context is a vital prerequisite for learning meaningful dynamic graphs, as relying solely on static embeddings makes the inferred relationships less informed and effective. Finally, we analyze the impact of the two-step GAT. When replacing it with a single-step GAT, the F1-score drops from 80.60% to 77.53% on SWaT and from 58.11% to 56.62% on WADI. While this variant performs better than the others, the performance gap confirms the value of our deep aggregation strategy. The shallow aggregation of a single-step GAT has a limited receptive field, whereas the two-step process can capture more complex, higher-order dependencies necessary to identify certain anomalies. The ablation study confirms that the synergy of temporal context encoding, dynamic graph construction, and two-step aggregation is essential to the superior performance of DART-GNN.

TABLE IV Computational Cost Analysis of Ablation Variants

Method	SWaT		WADI	
	Training Time / Epoch (s)	Params (KB)	Training Time / Epoch (s)	Params (KB)
DART-GNN	2.45	231	26.81	834
w/o GRU	2.08	78	20.66	337
w/o Two-Step GAT	1.72	209	17.87	775

In addition to detection accuracy, computational efficiency is a critical factor for the practical deployment of anomaly detection models. The computational cost of DART-GNN is primarily determined by three components: the GRU encoder, the dynamic graph construction, and the two-step GAT. For a multivariate time series with N sensors and a window size of w, the complexity of the GRU encoder is $O(N \cdot w^2)$. The dynamic graph construction, which is dominated by the matrix multiplication of the Query and Key matrices, has a complexity of $O(N^2)$. The two-step GAT, with $L = 2$ layers, has a complexity of approximately $O(L \cdot (\mathcal{E} + N))$, where $\mathcal{E}$ is the number of edges in the sparse graph.

To provide practical insights, we record the training time per epoch and the number of parameters for each model on different datasets, as shown in Table. IV. The w/o GRU variant is the most lightweight in terms of model size, reducing the parameter count by approximately 66% on the SWaT dataset



(a) Normal State Representations (b) Abnormal State Representations

Fig. 3. PCA visualization of sensor representations. (a) During normal operation, functionally related sensors form a tight cluster. (b) During the anomaly, the representation of the faulty sensor 1_FIT_001_PV moves away from its direct controller 1_MV_001_STATUS but remains close to other stable, related neighbors.

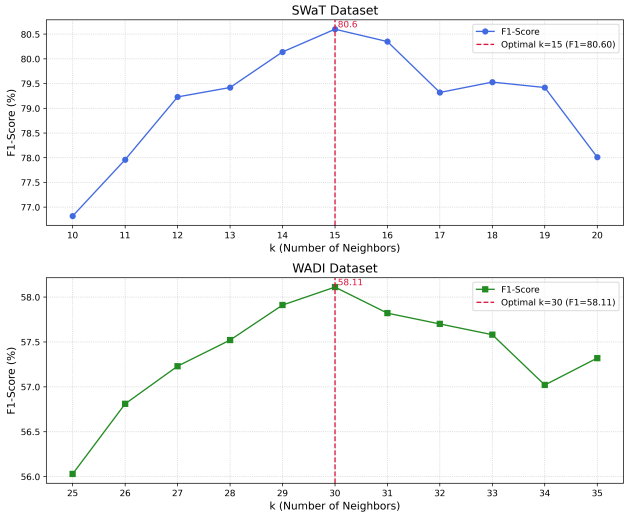


Fig. 4. Hyperparameter sensitivity analysis of Top-k

compared to the full DART-GNN model. This is expected, as the recurrent network is the most significant contributor to the model's complexity. Similarly, replacing the two-step GAT with a single-step version reduces the parameter count and makes it the fastest to train. While these variants offer improvements in efficiency, their reduced performance in the main ablation study confirms that the computational costs of the GRU and the two-step GAT are justified by their substantial contributions to detection accuracy. The full DART-GNN model maintains a reasonable parameter count and efficient training times, validating its practicality for real-world application.

F. Hyperparameter Analysis

The hyperparameter k, which determines the number of neighbors each node considers when constructing the dynamic graph, is crucial to the model's performance. To find an optimal value for k, we conduct a sensitivity analysis. We center our search for k around a value corresponding to approximately 25% of the total number of sensors for each dataset, as this provides a reasonable starting point for capturing significant relationships without introducing excessive noise.

The results of this analysis are presented in Fig. 4. For the SWaT dataset, we test k values from 10 to 20. The model's performance peaks at $k = 15$, achieving its best F1-score of 80.60%. This suggests that for the SWaT system, considering roughly 29% of the other sensors as potential neighbors provides the optimal balance between capturing sufficient context and avoiding noise from irrelevant nodes. For the more complex WADI dataset, we test k values from 25 to 35. The results confirm that a larger neighborhood is necessary for this higher-dimensional system. The model achieves its best F1-score of 58.11% at $k = 30$, which corresponds to approximately 24% of the total sensors. As with SWaT, the performance degrades as k moves away from this optimal value, indicating that a carefully selected neighborhood size is key to the model's success. Based on this empirical analysis, we set $k = 15$ for the SWaT dataset and $k = 30$ for the WADI dataset in all our experiments.

G. Qualitative Analysis

To provide deeper insight into our model's capabilities beyond quantitative metrics, this section presents a qualitative evaluation of the DART-GNN framework. We first validate the semantic reasonableness of the sensor representations learned during normal operation, and then we analyze how

these representations shift during an anomaly to provide an interpretable diagnosis.

Validation of Learned Node Representations. A key premise of our model is its ability to understand meaningful relationships between sensors. To verify this, we first examine the sensor representations learned during a period of normal system operation. We extract the final hidden state vectors from our GRU encoder for each sensor and project them into a two-dimensional space using PCA. The result, shown in Fig. 3a, confirms that our model learns semantically meaningful embeddings. Functionally related sensors are positioned closely together, forming a distinct cluster. Specifically, the representation for the flow sensor 1_FIT_001_PV is located in close proximity to its direct controller, the motorized valve 1_MV_001_STATUS. This demonstrates that our model correctly captures the strong, direct dependency between these components during normal operation.

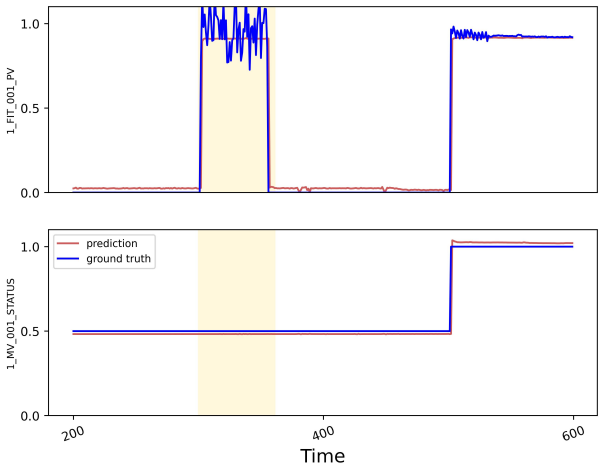


Fig. 5. Comparison of model predictions and ground truth values for sensors

Anomaly Interpretation. Our framework's explainability is further demonstrated by analyzing how these representations change during an anomalous event. Figure 3b visualizes the sensor representations from the moment of the anomaly. The representation of the attacked flow sensor 1_FIT_001_PV has moved significantly away from its direct controller 1_MV_001_STATUS. This visual divergence is a clear representation of the anomaly, showing the model's recognition that the flow sensor's behavior is inconsistent with the state of its valve. However, 1_FIT_001_PV representation remains in proximity to other physically related and stable sensors, such as the pressure sensor 1_AIT_001_PV and the level sensor 1_LT_001_PV. This specific relational shift is what enables the model's robust prediction, which can be seen in Fig. 5. By leveraging the graph structure, the model's forecast for 1_FIT_001_PV is influenced by the collective information from its stable neighborhood, including 1_AIT_001_PV and 1_LT_001_PV. Even though its relationship with the valve has been compromised, the model aggregates the stable context from these other reliable neighbors to predict the expected normal value for the flow. The anomaly is then detected by the large discrepancy between this stable, context-aware prediction and the chaotic ground truth of the attacked sensor. Furthermore, for the 1_MV_001_STATUS sensor, its prediction remains perfectly aligned with its stable ground truth, even during the anomaly. This indicates that the localized fault in 1_FIT_001_PV did not adversely affect the model's predictions for healthy sensors, showcasing the model's ability to precisely localize faults.

V. Conclusion

In this work, we introduce DART-GNN, a novel framework designed to overcome the shortcomings of static graph models in multivariate time series anomaly detection. Our approach integrates a GRU with a self-attention module, enabling the model to generate adaptive graphs that reflect the evolving relationships between sensors. Furthermore, by employing a two-step GAT, our model captures complex, higher-order dependencies that are often missed by conventional methods. Our comprehensive evaluation on public benchmarks confirms that DART-GNN sets a new state-of-the-art, significantly outperforming numerous baselines. An ablation study further validates our design, confirming the essential contribution of each core component. Qualitative analysis also shows that the learned dynamic graphs offer valuable insights for anomaly diagnosis. Future work will focus on investigating more computationally efficient attention mechanisms to enhance scalability, exploring the fusion of data-driven dynamic graphs with static graphs derived from domain knowledge to improve robustness, and adapting the framework for online learning in streaming data environments.

References

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

[2] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "Pca-based multivariate statistical network monitoring for anomaly detection," *Computers & Security*, vol. 59, pp. 118–137, 2016.

[3] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[4] L. Shen, Y. Wei, Y. Wang, and H. Li, "AFMF: time series anomaly detection framework with modified forecasting," *Knowledge Based System*, vol. 296, p. 111912, 2024.

[5] M. A. Belay, A. Rasheed, and P. S. Rossi, "Mtad: Multiobjective transformer network for unsupervised multisensor anomaly detection," *IEEE Sensors Journal*, vol. 24, no. 12, pp. 20 254–20 265, 2024.

[6] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Söderström, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 387–395.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *International Conference on Neural Information Processing Systems*, 2014, pp. 2672–2680.

[9] E. Dai and J. Chen, "Graph-augmented normalizing flows for anomaly detection of multiple time series," in *International Conference on Learning Representations*, 2021.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[11] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2016.

[12] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4027–4035.
[13] K. Cho, B. van Merrienboer, Çaglar Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.
[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *International Conference on Neural Information Processing Systems*, 2017, p. 6000–6010.
[15] M. Jain, G. Kaur, and V. Saxena, "A k-means clustering and SVM based hybrid concept drift detection technique for network anomaly detection," *Expert Systems with Applications*, vol. 193, p. 116510, 2022.
[16] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *Principles of Data Mining and Knowledge Discovery*, vol. 2431, 2002, pp. 15–27.
[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *International Conference on Management of Data*, 2000, pp. 93–104.
[18] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *International Conference on Data Mining*, 2008, pp. 413–422.
[19] A. Goodge, B. Hooi, S. K. Ng, and W. S. Ng, "Robustness of autoencoders for anomaly detection under adversarial impact," in *International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 1244–1250.
[20] X. Jie, X. Zhou, C. Su, Z. Zhou, Y. Yuan, J. Bu, and H. Wang, "Disentangled anomaly detection for multivariate time series," in *Companion Proceedings of the ACM on Web Conference*, 2024, p. 931–934.
[21] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*, 2019, pp. 703–716.
[22] Y. Liu, C. Gong, L. Yang, and Y. Chen, "Dstp-rnn: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction," *Expert Systems with Applications*, vol. 143, p. 113082, 2020.
[23] I. Kumar, B. K. Tripathi, and A. Singh, "Attention-based lstm network-assisted time series forecasting models for petroleum production," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106440, 2023.
[24] X. Wang, D. Pi, X. Zhang, H. Liu, and C. Guo, "Variational transformer-based anomaly detection approach for multivariate time series," *Measurement*, vol. 191, p. 110791, 2022.
[25] W. Chen, L. Tian, B. Chen, L. Dai, Z. Duan, and M. Zhou, "Deep variational graph convolutional recurrent network for multivariate time series anomaly detection," in *International Conference on Machine Learning*, 2022, pp. 3621–3633.
[26] M. Y. Yağci and M. A. Aydin, "Ea-gat: Event aware graph attention network on cyber-physical systems," *Computers in Industry*, vol. 159, p. 104097, 2024.
[27] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2019.
[28] D. Wang, P. Wang, J. Zhou, L. Sun, B. Du, and Y. Fu, "Defending water treatment networks: Exploiting spatio-temporal effects for cyber attack detection," in *International Conference on Data Mining*, 2020, pp. 32–41.
[29] H. Liu, W. Luo, L. Han, P. Gao, W. Yang, and G. Han, "Anomaly detection via graph attention networks-augmented mask autoregressive flow for multivariate time series," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 368–19 379, 2024.
[30] A. P. Mathur and N. O. Tippenhauer, "Swat: a water treatment testbed for research and training on ICS security," in *International Workshop on Cyber-physical Systems for Smart Water Networks*, 2016, pp. 31–36.
[31] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "WADI: a water distribution testbed for research in the design of secure cyber physical systems," in *International Workshop on Cyber-Physical Systems for Smart Water Network*. ACM, 2017, pp. 25–28.
[32] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.
[33] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.
[34] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: unsupervised anomaly detection on multivariate time series," in *International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 3395–3404.
[35] Z. Liu, X. Huang, J. Zhang, Z. Hao, L. Sun, and H. Peng, "Multivariate time-series anomaly detection based on enhancing graph attention networks with topological analysis," in *International Conference on Information and Knowledge Management*, 2024, pp. 1555–1564.
[36] Y. Wang, H. Long, L. Zheng, and J. Shang, "Graphformer: Adaptive graph correlation transformer for multivariate long sequence time series forecasting," *Knowledge-Based Systems*, vol. 285, p. 111321, 2024.
[37] H. Liu, W. Luo, L. Han, P. Gao, W. Yang, and G. Han, "Anomaly detection via graph attention networks-augmented mask autoregressive flow for multivariate time series," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 368–19 379, 2024.

Yuan Li received his B.S. degree in Computer Science from Nanjing University of Posts and Telecommunications in 2024. He is currently pursuing the M.S. degree in Control Science and Engineering at Nanjing University of Science and Technology. His current research interests include multivariate time series, reinforcement learning and multi-agent systems.

Yuhang Zhou received his B.S. degree in Automation from Nanjing University of Science and Technology (NJUST) in 2023. He is currently pursuing the M.S. degree in Control Science and Engineering at NJUST. His current research interests include reinforcement learning and multi-agent systems.

Qingzhong Yan received his B.S. degree in Automation from Nanjing University of Science and Technology, Nanjing, China, in 2023. He is currently a graduate student pursuing the M.S. degree in Control Science and Engineering at Nanjing University of Science and Technology, Nanjing, China.

Xiang Wu received the B.S. degree in electrical engineering and automation and PHD degree in control science and engineering from Nanjing University of Science and Technology, Nanjing, China, in 2012 and 2019, respectively. From 2016 to 2017, he was a Visiting scholar with the department of Computer Science and Engineering, Michigan State University, East Lansing, USA. He worked as a associate professor in control science and engineering at school of Automation, Nanjing University of Science and Technology, Nanjing, China. His current research interests include multi-agent reinforcement learning and sequential data processing.

Yuming Bo received the B.S., M.S., and Ph.D. degrees in navigation, guidance and control from Nanjing University of Science and Technology, Nanjing, China. He worked as a professor in control science and engineering at school of Automation, Nanjing University of Science and Technology, Nanjing, China. He is a member of the Chinese Association of Automation and Vice Chairman of Jiangsu Branch. His research interests include guidance, navigation and control, filtering and system optimization, and image processing.