Adversarial Graph Neural Network for Multivariate Time Series Anomaly Detection

Bolong Zheng, *Member, IEEE*, Lingfeng Ming, Kai Zeng, Mengtao Zhou, Xinyong Zhang, Tao Ye, Bin Yang, Xiaofang Zhou, *Fellow, IEEE*, and Christian S. Jensen, *Fellow, IEEE*

***Abstract*—Anomaly detection is one of the most significant tasks in multivariate time series analysis, while it remains challenging to model complex patterns for improving detection accuracy and to interpret the root causes of anomalies. However, existing studies either consider only the temporal dependencies, or simply reconstruct the original input for detection, both neglecting the hidden relationships among multivariate. We propose an adversarial graph neural network based anomaly detection model, called SGAT-AE, which consists of a Self-learning Graph ATtention network (SGAT), an Auto-Encoder (AE), and an adversarial training component. Specifically, SGAT is a prediction model that discovers the graph dependency relationships among multivariate and acts as a sample generator to confuse AE, while AE reconstructs the samples and acts as a discriminator that distinguishes a real sample from a generated one. A novel adversarial training between SGAT and AE is applied to amplify the errors of anomalies such that the prediction performance of SGAT is improved and the overfitting of AE is avoided. In addition, we aggregate the prediction error, the reconstruction error, and the adversarial error for anomaly detection, and develop a graph based anomaly interpretation method that locates the root causes from both local and global perspectives. Extensive experiments with five real-world data offer evidence that the proposed solution SGAT-AE is capable of achieving better performance when compared with the state-of-the-art proposals.**

***Index Terms*—Anomaly detection, anomaly interpretation, graph neural network, time series.**

I. Introduction

In real-world applications, sensing technologies are being deployed that produce increasing volumes of multivariate time series data, which enables the monitoring capabilities with the objective of supporting a variety of applications [1], [2], [3]. The anomaly detection in time series is one of the most significant tasks, refers to identifying abnormal states within a specific time period. For instance, database administrators (DBAs) monitor thousands of key performance indicators (KPIs) to determine whether performance failures happen in online transaction processing (OLTP) workloads [4]. In industrial infrastructures, operators monitor equipments through sensors to ensure the safety from attack [5].

As labeling anomalies by human experts is time consuming and labor intensive, time series often come without anomaly labels, so the anomaly detection is usually considered as an unsupervised learning problem. In the past few years, classical unsupervised statistical learning methods [6], [7], [8] are proposed to deal with the dependencies among multivariate in a linear manner.

In order to extract more complex features of multivariate time series data, deep learning based methods have been proposed for anomaly detection, which can be mainly divided into prediction-based [1], [9] and reconstruction-based methods [10], [11], [12]. In particular, prediction-based methods model the normal patterns of historical time series and predict the future trend, but they are easily affected by noise. The reconstruction-based methods learn latent representations of normal patterns and use it to reconstruct the original input, and then use the reconstruction error to detect anomalies. Although these methods can reduce the influence of noise, they are prone to overfitting. Generative adversarial networks (GANs) [13], [14] and variational autoencoders (VAEs) [12] are generative models that learn the distributions of normal data, and detect anomalies with reconstruction probabilities. More recently, ensemble models [15], [16] are proposed to improve the performance by combining the advantages of multiple base models. In addition, transformer-based methods [17], [18] are proposed that use attention-based autoencoders for reconstruction.

However, anomaly detection in multivariate time series remains challenging due to the following reasons:

(1) Hidden Spatial Dependency: In addition to the inherent temporal dependencies in time series, the spatial dependencies (hidden complex relationships) among multivariate exist and are unknown to operators. Consider an example in a water treatment plant where sensor 1 records the state of a valve and sensor 2 records the flow rate. The change of sensor 1 leads to the change of sensor 2, and this affection may also be delayed. Although principal component analysis (PCA) [19], [20] and auto-encoder (AE) [21], [22] can model the correlations of latent representations among multivariate, they do not explicitly present these dependency relationships to operators. The ambiguous spatial dependencies make it difficult to detect and interpret anomalies correctly.

Manuscript received 24 November 2022; revised 22 April 2024; accepted 23 June 2024. Date of publication 2 July 2024; date of current version 13 November 2024. This work was supported in part by NSFC under Grant 62372194 and Grant 62322205 and has already been applied in Huawei's GaussDB (DWS) system." Recommended for acceptance by P. Tsaparas. *(Corresponding author: Bolong Zheng.)*

Bolong Zheng and Lingfeng Ming are with the Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: zblchris@gmail.com).

Kai Zeng, Mengtao Zhou, Xinyong Zhang, and Tao Ye are with Huawei, Shenzhen 518129, China.

Bin Yang is with East China Normal University, Shanghai 200050, China.

Xiaofang Zhou is with the Hong Kong University of Science and Technology, Hong Kong, SAR, China.

Christian S. Jensen is with Aalborg University, 9220 Aalborg, Denmark.

Digital Object Identifier 10.1109/TKDE.2024.3419891

1041-4347 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

(2) Indistinguishably Small Anomaly Score: Existing models learn the normal patterns during training and take the error between the model output and the ground truth as the anomaly score, so a sample with a high score is labeled as an anomaly. However, if the anomaly is relatively close to the normal data, the error might be too small to detect correctly. Therefore, the core of improving the performance of anomaly detection is to amplify the errors, especially for the anomalies with low scores. Although USAD [5] proposes two AEs sharing a same encoder for adversarial training to amplify the errors, it dose not model the spatial and temporal dependencies, which limits the performance.

(3) Accurate Anomaly Interpretation: Most existing studies only focus on detecting the anomaly, but neglect the anomaly interpretation that analyzes which main variables cause the anomaly that can help speed up troubleshooting. Both DAEMON [23] and OmniAnomaly [2] simply identify the top-k variables with the largest errors or with the lowest reconstruction probabilities as the interpretation of the anomaly. However, they fail to consider the dependency relationships among multivariate and may miss the true root cause variables.

To address the challenges above, we propose a novel adversarial graph neural network, called SGAT-AE, which consists of a Self-learning Graph-structured ATtention network (SGAT), an Auto-Encoder (AE), and an adversarial training component. SGAT is a prediction-based model that combines a self-learning graph and an attention network to discover the hidden spatial dependencies among multivariate (*Challenge 1*). Specifically, different from most graph learning methods [24], [25], SGAT learns a directed graph structure since the dependence patterns among multivariate time series are asymmetric. SGAT uses the multiscale 1D convolution to extract the nonlinear temporal dependencies of multivariate time series. Different from GDN [26] that only captures the spatial dependencies, SGAT enables to learn both the spatial and temporal dependencies simultaneously. AE is a reconstruction-based model that learns the latent representations of normal patterns and reconstructs the original input.

As mentioned previously, the reconstruction-based methods enable to learn the normal patterns but are easy to get overfitting, while the prediction-based methods are susceptible to noise. Therefore, to take the best of the both worlds, an adversarial training process between SGAT and AE is adopted. The input data is first processed by SGAT whose role is to reconstruct the input as close as possible. Then AE tries to distinguish the data generated by SGAT from the original input. It is worth noting that AE is able to identify whether an input sample is normal or not before reconstruction. So we introduce an adversarial training that allows to amplify the reconstruction errors from data generated by SGAT to avoid over-parameterized (*Challenge 2*). Different from GANs [13], the reason why we use AE as the discriminator is to get rid of the mode collapse and non-convergence owing to the stability of AE [5].

We compute the anomaly score by aggregating the prediction error, the reconstruction error, and the adversarial error together. In addition, we use the self-learning graph to clarify the relationships among multivariate, and interpret effectively the anomaly from both local and global perspectives (*Challenge 3*).

Overall, the contributions are summarized as follows:

1) We propose a novel adversarial graph neural network SGAT-AE for anomaly detection in multivariate time series. SGAT learns graph dependency relationships among multivariate to better extract the spatial dependencies and temporal dependencies. The idea of SGAT-AE is to employ an adversarial training between SGAT and AE to amplify the errors of anomalies.
2) We develop a novel anomaly interpretation method that effectively locates the root cause variables of the anomalies by considering the dependency relationships among multivariate, which are obtained from the self-learning graph in SGAT.
3) We conduct extensive experiments in five real-world datasets, and the experimental results show that SGAT-AE outperforms state-of-the-art baseline methods. A case study also shows that the anomaly interpretation method effectively locates the root causes of anomalies.

II. Preliminaries

We proceed to introduce the settings and formally define the problem of anomaly detection in multivariate time series.

A. Settings

Definition 1 (Time Series): A multivariate time series sequence is denoted as $\mathcal{X} = \{X_1, \ldots, X_t, \ldots, X_T\}$, where $X_t \in \mathbb{R}^{N \times d}$ is an observation at time t that contains N variables with d-dimensional features, and T is the length of the time series.

To model the strong temporal dependencies of time series, we pre-process the original sequences into time windowed sequences.

Definition 2 (Time Windowed Sequence): We denote a windowed sample at time t with length w as $W_t = \{X_{t-w+1}, \ldots, X_t\}$, which contains the historical observations from time $t-w+1$ to t in $\mathcal{X}$. Therefore, $\mathcal{X}$ is transformed into a time windowed sequence $\mathcal{W} = \{W_1, \ldots, W_t, \ldots, W_{T'}\}$, where $T' = T - w + 1$.

In order to extract temporal patterns and reduce the impact of noise, we adopt a feature enrichment method [27], [28] that uses the moving average to smooth short-term fluctuations and to highlight long-term trends. We denote X_t^m as the moving average value of the latest m times at time t. Note that, a large m indicates an emphasis on long-term trends. Based on practitioner's experiences, we adopt the moving average features of X_t^3, X_t^5 for short-term patterns, and those of X_t^{10}, X_t^{20} for long-term patterns, respectively, in order to obtain abundant temporal features and to reduce the impacts of noises by both the short and long-term patterns. For ease of presentation, we use $X_t^\star$ to represent all these moving average features, i.e., $X_t^\star = [X_t^3; X_t^5; X_t^{10}; X_t^{20}]$, where $[;]$ represents the concatenation operation. Then, we transform $X_t^\star$ to $W_t^\star$.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

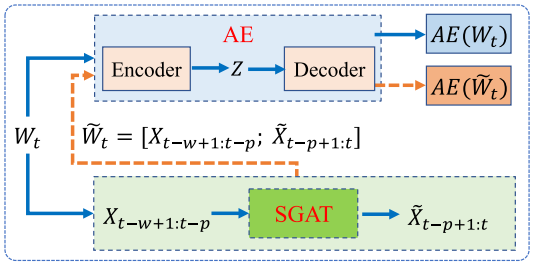


Fig. 1. The SGAT-AE model.

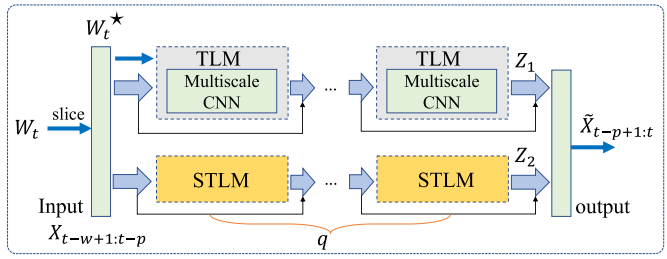


Fig. 2. The architecture of SGAT.

B. Problem Statement

We study two problems as follows.

1) *Anomaly Detection:* Given a windowed sample W_t and the corresponding moving average feature $W_t^\star$ at time t, SGAT-AE aims to label the observation X_t with $y_t \in \{0, 1\}$ by computing an anomaly score S_t based on the error between the original input and the model output. If S_t is larger than a threshold τ, we label X_t as an anomaly ($y_t = 1$); otherwise, X_t is labeled as normal ($y_t = 0$).
2) *Anomaly Interpretation:* The anomaly interpretation is to answer the question of why an observation is detected as an anomaly. Given a labeled anomaly X_t, we need to clarify the dependency relationships among multivariate and diagnose the root variables that cause the anomaly.

III. SGAT-AE

We proceed to introduce the details of SGAT-AE that consists of a Self-learning Graph-structured ATtention network (SGAT), an Auto-Encoder (AE), and an adversarial training component between SGAT and AE. SGAT models the historical patterns of multivariate time series to predict the future trends, while AE learns to reconstruct the original input. The idea of SGAT-AE is to employ SGAT and AE for adversarial training to amplify the anomaly error so that the performance of the multivariate time series anomaly detection is improved.

A. Network Architecture

The model architecture of SGAT-AE is illustrated in Fig. 1. For the current input W_t:

1) *SGAT:* To extract the spatial and temporal dependencies of multivariate time series, we propose a graph neural network-based multivariate time series forecasting model SGAT that employs the first $(w - p)$-step historical observations to predict the following p-step values in W_t. When $p = 1$, it is a single-step prediction. When $p > 1$, it is a multi-step prediction.
2) *AE:* To reduce the dimension of multivariate time series, AE is to learn the latent representations for normal patterns and then back them into the original input space.
3) *Adversarial Training:* To improve the prediction accuracy of SGAT and to avoid overfitting of AE, SGAT is trained adversarially with AE (shown by the orange dashed arrow in Fig. 1). Specifically, SGAT works as a generator by merging its input and prediction output to obtain a generated sample $\tilde{W}_t$, while AE discriminates whether the input is an original input W_t or a generated input $\tilde{W}_t$. As a result, SGAT-AE enables to amplify the errors of anomalies through the adversarial training between the two networks.

B. Multivariate Time Series Forecasting

We propose a graph neural network-based multivariate time series forecasting model SGAT that aims to model the historical variable patterns to predict the future trends. Specifically, we employ a self-learning graph and a graph attention network to discover spatial relationships among multivariate, while we utilize multiscale 1D CNNs to extract nonlinear temporal dependencies.

Forecasting Problem: At current time t, we slice W_t to obtain $X_{t-w+1:t-p}$ and $X_{t-p+1:t}$, which are used as the input and the target of SGAT, respectively. Similarly, $W_t^\star$ is sliced to obtain $X^\star_{t-w+1:t-p}$. The goal of SGAT is to learn a mapping function f that is able to predict $\tilde{X}_{t-p+1:t}$ given $X_{t-w+1:t-p}$ and $X^\star_{t-w+1:t-p}$.

$$[X_{t-w+1:t-p}, X^\star_{t-w+1:t-p}] \xrightarrow{f} \tilde{X}_{t-p+1:t}, \tag{1}$$

As illustrated in Fig. 2, SGAT consists of an input layer, q temporal learning modules (TLMs) and spatial-temporal learning modules (STLMs), and an output layer. The value of q depends on the problem complexity and we set q to 2. Both of the input layer and the output layer are fully connected layers. The input layer first projects the input $X_{t-w+1:t-p}$ into a latent space. Then, it is processed in parallel by the following chains.

1) In the upper chain, it is first concatenated with the moving average feature $X^\star_{t-w+1:t-p}$ to enrich the temporal features, and is then processed by TLMs to extract nonlinear temporal dependencies.
2) In the lower chain, it is processed by STLMs to mine hidden spatial dependencies and to extract temporal dependencies simultaneously.

The results of the two chains are concatenated and then taken into the output layer to return the following p-step result, i.e., $\tilde{X}_{t-p+1:t} \in \mathbb{R}^{p \times N \times d}$. It is worth noting that residual and skip operations [29] are employed to prevent network degradation. The loss function of SGAT is:

$$Loss_{pred} = ||X_{t-p+1:t} - \tilde{X}_{t-p+1:t}||_2^2, \tag{2}$$

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

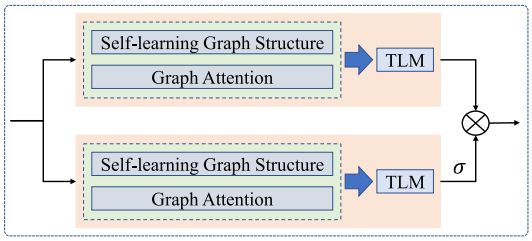


Fig. 3. Spatial-temporal learning module (STLM).

where $||\cdot||_2$ represents L2-norm.

1) Temporal Learning Module (TLM): As different variables have different trends, the temporal dependencies in multivariate time series are diversified. To address this issue, we employ 1D convolution filters [30] with multiple sizes to extract richer temporal features. Therefore, different convolution filters can adapt to temporal patterns of different variables.

The temporal learning module (TLM) consists of 1D convolution filters with multiple sizes [29], i.e., $\mathbf{\Gamma}_{2\times1} \in \mathbb{R}^2$, $\mathbf{\Gamma}_{3\times1} \in \mathbb{R}^2$, and $\mathbf{\Gamma}_{5\times1} \in \mathbb{R}^2$. The intuition is that these filters can be combined into multiple temporal lengths, such as 7, 9, 12, and so on. It is worth noting that we can choose more convolution filters according to the problem complexity. Consider the current input Z_{in}, the output of TLM is computed as follows:

$$Z_{out} = Concat(Z_{in} * \mathbf{\Gamma}_{2\times1}, Z_{in} * \mathbf{\Gamma}_{3\times1}, Z_{in} * \mathbf{\Gamma}_{5\times1}), \quad (3)$$

where $*$ denotes the convolution operation and $Concat(\cdot)$ denotes the concatenation operation. Note that, a sequence with time axis length h is reduced to $h-k+1$ after passing through a filter of $\mathbf{\Gamma}_{k\times1}$ with stride 1 and no padding. Therefore, the outputs of the three filters are truncated to a same length according to the filter $\mathbf{\Gamma}_{5\times1}$ and then concatenated in the channel dimension.

2) Spatial-Temporal Learning Module (STLM): In addition, it is also challenging to extract the complex spatial dependencies in multivariate time series since the causal relationships among multivariate are unknown. To address this issue, we transform the multivariate time series forecasting problem to a graph node prediction problem that aims to learn a graph with dependency relationships among multivariate.

Fig. 3 shows the details of the spatial-temporal learning module (STLM) that consists of two identical structures. For each structure, we consider variables as nodes in a graph and employ a self-learning graph to build topological relationships via node embedding technology [31]. Thereafter, a graph attention network is used to extract spatial dependencies among nodes. By combing with a TLM, we extract the spatial dependencies and temporal dependencies simultaneously. We use a gate linear unit (GLU) [32] to model the complex nonlinearity. The one followed by a sigmoid activation function works as a gate to control the information output from the other one, where σ is the sigmoid function and $\otimes$ denotes the Hadamard product.

Self-learning Graph: The self-learning graph aims to learn an adjacency matrix A without any additional prior knowledge through an end-to-end manner using the stochastic gradient algorithm. However, there exist only few studies focusing on this problem. Graph WaveNet [33] builds an adaptive graph structure through node similarity, and its adjacency matrix A is symmetric and bidirectional. In fact, the relationships among nodes (variables) may be directed. Consider an example in a water treatment plant. A change in sensor 1 will cause a change in sensor 2, but not vice versa. MTGNN [29] proposes a directed graph, which is more realistic in real-world applications. However, the edge weights may be negative which results in unnatural relationships. To solve the problem, we build directed relationships among multivariate time series as follows:

$$\begin{aligned} &for\ i = 1, 2, \ldots, N: \\ &A[:, i] = SoftMax(tanh(\eta E_2 \Theta_2 E_1^T[i, :])), \\ &A[i, :] = SoftMax(tanh(\eta E_1[i, :] \Theta_1 E_2^T)), \end{aligned} \quad (4)$$

where Θ_1, Θ_2 are learnable parameters, E_1, $E_2 \in \mathbb{R}^{N\times c}$ are node embeddings and c is the embedding dimension. These parameters are optimized by the back-propagation algorithm after random initialization during training. η is the zoom factor. The in-degree weights $A[:, i]$ and the out-degree weights $A[i, :]$ of each node are different, which ensures that A is asymmetric. We use the $tanh$ activation function to scale the weights of edges to the same interval and then normalize through the $SoftMax$ function. Note that A is a fully-connected graph in (4). To eliminate weak connections, for each node i $(i = 1, 2, \ldots, N)$, we only select k nodes with the largest edge weights as its neighbors, and set the weights of the other nodes to 0.

Graph Attention Network: We employ a graph attention-based approach to aggregate the node's information with its neighbors base on the self-learning graph A in terms of the correlation weights among nodes. Different from the classic GAT [34] that only considers the time series node features to obtain attention scores, we take into account the node embeddings obtained from the process of graph construction. The intuition is that the information is transmitted among nodes through A, while node embeddings represent the graph structure. Therefore, the attention score α_{ij} between nodes i and j is computed as follows:

$$e_{ij} = \frac{< [v^i; MW_t^i], [v^j; MW_t^j] >}{\sqrt{Dim}}, \quad (5)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}(i)} \exp(e_{ik})}, \quad (6)$$

where v^i and v^j denote the embedings of nodes i and j, W_t^i and W_t^j represent the time series features of nodes i and j at time t, respectively. M is a trainable weight matrix. MW_t^i represents the transformed feature of node i. $< \cdot, \cdot >$ denotes the inner product operation, and Dim is the dimension of $[v^i; MW_t^i]$. Then, the attention score is normalized.

Thereafter, the aggregated representation Z_t^i of node i at time t is computed as follows:

$$Z_t^i = \sigma\left(MW_t^i + \sum_{j\in\mathcal{N}(i)} \alpha_{ij} MW_t^j\right), \quad (7)$$

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

where $\mathcal{N}(i) = \{j | A_{ji} > 0\}$ represents the neighbors of node i obtained from the self-learning graph A. σ denotes the sigmoid activation function.

C. Auto-Encoder

The Auto-Encoder (AE) learns to reconstruct the original input. It consists of an encoder E and a decoder D. The encoder maps the origin input W_t into a latent representation Z and then the decoder maps it back into the input space to obtain a reconstruction $AE(W_t)$. The reconstruction error between the original input W_t and the reconstruction $AE(W_t)$ is calculated as follows:

$$Loss_{rec} = ||W_t - AE(W_t)||_2^2, \tag{8}$$

where $AE(W_t) = D(Z)$ and $Z = E(W_t)$. On the one hand, AE is supposed to achieve accurate reconstruction by minimizing the error $Loss_{rec}$. On the other hand, if AE is too sensitive to the input data, AE may only learn a mapping relationship from the input to the reconstruction, resulting in overfitting. To avoid this problem, AE is usually optimized by choosing a reasonable dimension of the latent representation Z or adding a regularization term. We train AE in an adversarial manner to avoid overfitting.

D. Adversarial Training

SGAT and AE are two different anomaly detection methods. The former is a prediction-based method that considers the prediction error as the anomaly score, while the latter is a reconstruction-based method that uses the reconstruction error as the anomaly score. They both regard observation samples with high scores as anomalies. However, they both have shortcomings. SGAT models the historical patterns of multivariate time series, which is susceptible to noise. AE is over-parameterized [15] and is easy to fit with anomalies, which means that the reconstruction error of AE is small when the anomaly is close to the normal data, making it difficult to detect the anomaly correctly. Therefore, the core of improving anomaly detection performance is how to amplify the error of anomalies. This problem can be solved if AE is able to identify whether an input sample is normal or not before reconstruction. The generative adversarial network (GAN) technique [13] is a feasible solution that is based on a two-player minimax adversarial game between two networks trained simultaneously. However, the classic GAN neglects the spatial and temporal dependencies in multivariate time series. Therefore, we employ the prediction model SGAT and AE for adversarial training to combine their advantages to break through the bottleneck and to improve the performance of anomaly detection in multivariate time series.

Specifically, the input and the prediction result of SGAT are merged to obtain a generated sample $\tilde{W}_t$, which is reconstructed by AE into $AE(\tilde{W}_t)$. Therefore, the adversarial loss function and the objective function of adversarial training are computed as follows:

$$Loss_{adv} = ||W_t - AE(\tilde{W}_t)||_2^2, \tag{9}$$

$$\min_{SGAT} \max_{AE} Loss_{adv}, \tag{10}$$

Algorithm 1: SGAT-AE Training.

Input: Number epochs $MaxEpoch$, Normal windows Dataset $\mathcal{W} = \{W_1, W_2, \ldots, W_{T'}\}$ and $\mathcal{W}^\star = \{W_1^\star, W_2^\star, \ldots, W_{T'}^\star\}$
Output: Trained SGAT and AE

SGAT, AE ← initialize weights
for $n = 1$ *to* $MaxEpoch$ **do**
  for $t = 1$ *to* T' **do**
    // **Train SGAT**
    $X_{t-w+1:t-p}, X_{t-p+1:t} \leftarrow W_t$;
    $X^\star_{t-w+1:t-p}, X^\star_{t-p+1:t} \leftarrow W^\star_t$;
    $\tilde{X}_{t-p+1:t} \xleftarrow{SGAT} [X_{t-w+1:t-p}, X^\star_{t-w+1:t-p}]$;
    $\tilde{X}_{t-p+1:t} \xleftarrow{f(\cdot)} [X_{t-w+1:t-p}, X_{ma,t-w+1:t-p}]$
    $Loss_{pred} \leftarrow ||X_{t-p+1:t} - \tilde{X}_{t-p+1:t}||_2^2$;
    SGAT ← update weights using $Loss_{pred}$;
    // **Train AE**
    $AE(W_t) \xleftarrow{AE} W_t$;
    $Loss_{rec} \leftarrow ||W_t - AE(W_t)||_2^2$;
    AE ← update weights using $Loss_{rec}$;
    // **Adversarial Training**
    Compute $Loss_{pred}$ and $Loss_{rec}$ as above;
    $\tilde{W}_t \leftarrow [X_{t-w+1:t-p}; \tilde{X}_{t-p+1:t}]$;
    $AE(\tilde{W}_t) \xleftarrow{AE} \tilde{W}_t$;
    $Loss_{adv} \leftarrow ||W_t - AE(\tilde{W}_t)||_2^2$;
    $Loss_1 \leftarrow \frac{1}{n} Loss_{pred} + (1 - \frac{1}{n}) Loss_{adv}$;
    SGAT ← update weights using $Loss_1$;
    $Loss_2 \leftarrow \frac{1}{n} Loss_{rec} - (1 - \frac{1}{n}) Loss_{adv}$;
    AE ← update weights using $Loss_2$;
Return SGAT, AE;

On the one hand, SGAT acts as a generator with the goal of minimizing adversarial error $Loss_{adv}$, as it wishes to make accurate predictions to generate sufficiently realistic sample $\tilde{W}_t$ to confuse AE. Consider the prediction error of SGAT itself in (2), the adversarial training objective of SGAT is computed as follows:

$$\begin{aligned} Loss_1 &= Loss_{pred} + Loss_{adv} \\ &= ||X_{t-p+1:t} - \tilde{X}_{t-p+1:t}||_2^2 + ||W_t - AE(\tilde{W}_t)||_2^2, \end{aligned} \tag{11}$$

On the other hand, AE acts as a discriminator with the goal of maximizing the adversarial error $Loss_{adv}$ as it aims to distinguish whether the input is a real sample W_t or a generated sample $\tilde{W}_t$ from SGAT. Consider the reconstruction error of AE itself in (8), the adversarial training objective of AE is computed as follows:

$$\begin{aligned} Loss_2 &= Loss_{rec} - Loss_{adv} \\ &= ||W_t - AE(W_t)||_2^2 - ||W_t - AE(\tilde{W}_t)||_2^2, \end{aligned} \tag{12}$$

It is worth noting that SGAT and AE both have two optimization objectives, the weights of the two optimization objectives should change with time [5]. So $Loss_1$ and $Loss_2$ are modified as follows:

$$Loss_1 = \frac{1}{n} ||X_{t-p+1:t} - \tilde{X}_{t-p+1:t}||_2^2$$

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

$$+\left(1-\frac{1}{n}\right)||W_t - AE(\tilde{W}_t)||_2^2, \tag{13}$$

$$Loss_2 = \frac{1}{n}||W_t - AE(W_t)||_2^2 - \left(1-\frac{1}{n}\right)||W_t - AE(\tilde{W}_t)||_2^2, \tag{14}$$

where n denotes a training epoch. According to the above description, SGAT tries to minimize both the prediction error and the reconstruction error of $\tilde{W}_t$ during the adversarial training process. Besides, AE tries to minimize the reconstruction error of W_t but to maximize that of $\tilde{W}_t$ to amplify the reconstruction discrepancy of abnormal input samples. In order to speed up the convergence of SGAT-AE, we adopt a strategy of combining individual training and adversarial training for SGAT and AE, which means that the parameters of each network are updated twice for each mini-batch samples. The specific training process is summarized in Algorithm 1.

IV. Anomaly Detection and Interpretation

We proceed to introduce the anomaly detection approach. In addition to detecting the anomalies, we also aim to locate the root causes of the anomalies based on the relationships in the self-learning graph. We first compute an anomaly score to determine whether an observation is anomalous or not. Then, the anomaly interpretation is applied.

A. Anomaly Score

The anomaly score is closely related to the anomaly detection model since it is computed based on the error between the ground truth and the model output. As SGAT-AE consists of SGAT, AE, and an adversarial training component, the output brings the prediction error, the reconstruction error, and the adversarial error. Consider these three kinds of errors, we first compute individual scores for each variable, and then aggregate them into a single score. Specifically, the anomaly score S_t of observation X_t is computed as follows:

(1) Time Dimension Aggregation Error: To pay more attention to the moment with the largest gap between the ground truth and the model output, we use the max function as the aggregation function and compute the error of each variable over time as follows:

$$\begin{aligned} Err_{pred} &= \max_t |X_{t-p+1:t} - \tilde{X}_{t-p+1:t}|, \\ Err_{rec} &= \max_t |W_t - AE(W_t)|, \\ Err_{adv} &= \max_t |W_t - AE(\tilde{W}_t)|, \end{aligned} \tag{15}$$

where $Err_{pred} \in \mathbb{R}^N$, $Err_{rec} \in \mathbb{R}^N$, $Err_{adv} \in \mathbb{R}^N$ represent the prediction error, the reconstruction error, and the adversarial error, respectively. Since different variables may have different properties, they may also have different variation ranges. It is possible that the bias generated by one variable over-dominates the other variables. To avoid this issue, we normalize the error of each variable.

(2) Variable Aggregation Error: Since the anomalies mainly affect a small number of variables, or even a single variable, we use the max function to aggregate the errors of N variables to obtain signal anomaly scores as follows:

$$\begin{aligned} Score_{pred} &= \max_{i\in\{1,2,\ldots,N\}} Err^i_{pred}, \\ Score_{rec} &= \max_{i\in\{1,2,\ldots,N\}} Err^i_{rec}, \\ Score_{adv} &= \max_{i\in\{1,2,\ldots,N\}} Err^i_{adv}, \end{aligned} \tag{16}$$

where i represents the ith variable, and Err^i denotes the normalization error of the ith variable.

(3) Aggregation Anomaly Score of Three Components: By combining the three kinds of errors, the final anomaly score S_t is aggregated as follows:

$$S_t = \beta_1 Score_{pred} + \beta_2 Score_{rec} + \beta_3 Score_{adv}, \tag{17}$$

where β_1, β_2, and β_3 are the corresponding weights of the three kinds of anomaly scores, and $\beta_1 + \beta_2 + \beta_3 = 1$. Different weights may result in different anomaly score S_t, and they are used to trade off between false negatives and false positives. A large β_1 means that we pay more attention to the prediction error, so as a large β_2 to the reconstruction error and a large β_3 to the adversarial error. The regulation of β_1, β_2, and β_3 allows parameterizing the sensitivity of SGAT-AE to meet the requirements of a production environment. For example, in an industry company, managers may prefer a low sensitivity level, limiting the number of false positives but warning them only in case of important incidents, while technicians may prefer a high sensitivity level that limits the number of false negatives. Therefore, we adopt 0.05, 0.05, 0.9 as the values of β_1, β_2, and β_3 since the adversarial error is amplified by SGAT-AE, in order to achieve a high sensitivity from a technician view.

Finally, an observation X_t at time t is labeled as an anomaly if S_t exceeds a threshold τ. Since the performance of anomaly detection is sensitive to the threshold τ, it needs to be set reasonably. On the one hand, if τ is too large, only a small number of the observations are labeled as abnormal, while most of the anomalies are not identified. For the false negatives, they may cause major equipment problems and even heavy losses. On the other hand, if the threshold τ is too small, most of the observations including the normal data are labeled as abnormal. For the false positives, the risk of excessive security warning exists that may result in an unnecessary waste of resources. Therefore, the threshold τ requires a trade-off between false negatives and false positives. We employ the selection method proposed in OmniAnomaly [2] to set the threshold τ.

B. Anomaly Interpretation

Different from anomaly detection, the anomaly interpretation is a more complex task. For a labeled anomaly X_t, the anomaly interpretation is to analyze which main variables cause the anomaly so that the operators can locate quickly the root cause. DAEMON [23] identifies the top-k variables with the largest errors as the interpretation of anomaly. It makes sense

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

to some extent, but it fails to take the dependency relationship among multivariate into account. Consider an example in a water treatment plant. The change of sensor 1 will cause the change of sensor 2, but the value change of sensor 1 is smaller than that of sensor 2. When an anomaly occurs, DAEMON treats sensor 2 as the root cause which is incorrect since sensor 1 is the true root cause variable of the anomaly.

To avoid the issue, we propose a graph structure-based anomaly interpretation method. Specifically, we clarify the dependency relationships among variables before performing anomaly interpretation and then locate the root variables from both local and global perspectives. For a labeled anomaly X_t.

- *Local Anomaly Interpretation:* First, the node (variable) with the largest prediction error determined by (16) is taken as an anomaly node. Then, its neighbors are obtained according to the self-learning graph structure. Specifically, its in-degree neighbors and itself are regarded as the candidate nodes of the root cause variables of the anomaly, while its out-degree neighbors help understand the direction of anomaly error propagation. As in the above example, we take sensor 2 as an anomaly node. The directed dependency relationship between sensors 1 and 2 is learned by SGAT-AE and is reflected in the self-learning graph structure, where sensor 1 is sensor 2's in-degree neighbor. In a result, both of the two nodes are regarded as candidate nodes that cause the anomaly.
- *Global Anomaly Interpretation:* According to the self-learning graph, the nodes with only out-degree neighbors and no in-degree neighbors are regarded as a main cause node set, and the nodes with in-degree neighbors are dependent nodes. We select the largest top-k prediction errors in the main cause node set as the root cause variable of the anomaly, regardless of the dependent nodes. As in the above example, we regard sensor 1 as the main cause node and sensor 2 as the dependent node. Therefore, sensor 1 is correctly regard as the root cause of the anomaly.

We take the candidate results from both local and global perspectives as the final interpretation results, so that we can diagnose the anomalies quickly and correctly. Section V-C2 demonstrates case studies.

V. Experimental Study

A. Experimental Settings

The SGAT-AE model is implemented in Python3 with Pytorch. All experiments are run on a Linux (Ubuntu 18.04 LTS) machine with a NVIDIA Tesla V100 GPU card. We use five publicly available datasets with labeled ground-truth anomalies to evaluate our model. Table I summaries the dataset statistics.

- *SWaT* (Secure Water Treatment):[1] The dataset originates from a test bench for a real water treatment plant. It consists of 11 days of continuous operation: the first 7 days collected under normal operations are regarded as the training set and the last 4 days collected with attack scenarios are regarded as the testing set.

[1]https://github.com/JulienAu/Anomaly_Detection_Tuto

TABLE I
Dataset Statistics

Dataset	# Train	# Test	# Dimension	Anomaly(%)
SWaT	496800	449919	51	11.98
WADI	1048571	172801	123	5.99
SMD	708405	708420	28*38	4.16
SMAP	135183	427617	55*25	13.13
MSL	58317	73729	27*55	10.72

- *WADI* (Water Distribution):[2] The dataset is an extension of SWaT dataset. It consists of 16 days of continuous operation: the first 14 days collected under normal operations are regarded as the training set and the last 2 days with attack scenarios are regarded as the testing set.
- *SMD* (Server Machine Dataset):[3] The dataset is a new 5-week-long dataset collected from a large Internet company. It is collected from 28 different machines that form 28 subsets. Each subset is divided into two parts with equal length for training and testing.
- *SMAP* (Soil Moisture Active Passive) and **MSL** (Mars Science Laboratory):[4] These two datasets are labeled by experts from NASA and are anonymized w.r.t. time. The testing set size of SMAP is more than 3 times of the training set while the testing set size of MSL is roughly 1.2 times of the training set.

Methods for Comparison: To study the performance of SGAT-AE for multivariate time series anomaly detection, we compare it with the following methods:

- *AE* [10]: A encoder-decoder architecture with three fully-connected layers that compresses the input into a latent representation and then maps it back to the input space.
- *LSTM-VAE* [35]: A reconstruction-based model that replaces the feed-forward network of VAE with LSTM to extract temporal dependencies.
- *LSTM-NDT* [1]: A prediction-based model with a non-parametric anomaly threshold, which employs LSTM to predict future trend based on the historical observations.
- *DAGMM* [36]: A prediction-based model that utilizes a deep AE to generate a low-dimensional representation and reconstruction error for each input, which is further fed into a Gaussian Mixture Model (GMM).
- *BeatGAN* [14]: A reconstruction-based model that employs an AE to work as a generator and uses an adversarial generation network to guarantee the robustness.
- *OmniAnomaly* [2]: A reconstruction-based model that adopts a stochastic recurrent neural network model that glues GRU and VAE to learn the normal patterns of multivariate time series and uses the reconstruction probability for anomaly judgement.
- *USAD* [5]: A reconstruction-based model that consists of two AEs sharing the same encoder for adversarial training.
- *GDN* [26]: A prediction-based model that combines a structure learning approach with graph neural networks,

[2]https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#wadi
[3]https://github.com/smallcowbaby/OmniAnomaly
[4]https://github.com/khundman/telemanom

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

TABLE II
COMPARISON WITH BASELINE MODELS

Method	SWaT			WADI			SMD			SMAP			MSL		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
AE	0.991	0.704	0.823	0.397	0.322	0.356	0.883	0.804	0.841	0.722	0.980	0.831	0.854	0.975	0.910
LSTM-VAE	0.712	0.923	0.805	0.480	0.634	0.546	0.870	0.788	0.823	0.716	0.985	0.830	0.860	0.976	0.914
LSTM-NDT	0.677	0.774	0.723	0.602	0.463	0.523	0.557	0.778	0.649	0.922	0.678	0.781	0.630	0.969	0.764
DAGMM	0.829	0.767	0.797	0.727	0.511	0.600	0.673	0.845	0.749	0.633	0.998	0.775	0.756	0.980	0.854
BeatGAN	0.814	0.843	0.828	0.439	0.673	0.531	0.729	0.841	0.781	0.924	0.559	0.696	0.898	0.854	0.875
OmniAnomaly	0.722	0.983	0.833	0.636	0.539	0.584	0.981	0.944	0.962	0.759	0.976	0.854	0.914	0.889	0.901
USAD	0.987	0.740	0.846	0.778	0.535	0.634	0.931	0.962	0.946	0.770	0.983	0.863	0.881	0.979	0.927
GDN	0.948	0.719	0.818	0.796	0.713	0.752	0.999	0.902	0.945	0.911	0.561	0.695	0.876	0.809	0.841
TranAD	0.998	0.688	0.814	0.353	0.827	0.496	0.907	0.997	0.950	0.804	0.999	0.891	0.904	0.999	**0.950**
GTA	0.819	0.802	0.810	0.755	0.716	0.735	0.885	0.974	0.927	0.862	0.494	0.628	0.608	0.623	0.616
RANSynCoders	0.964	0.715	0.821	0.324	0.390	0.354	0.732	0.945	0.825	0.836	0.914	0.873	0.859	0.889	0.874
SGAT-AE	0.949	0.959	**0.954**	0.827	0.999	**0.905**	0.941	0.994	**0.967**	0.885	0.910	**0.897**	0.921	0.951	0.935

and uses attention weights to provide the explainability for the detected anomalies.

- *TranAD* [18]: A reconstruction-based model that employs a encoder-decoder model structure utilizing Transformer. TranAD leverages self-conditioning and adversarial training to amplify errors and gain training stability.
- *GTA* [17]: A prediction-based model. GTA involves automatically learning a graph structure, graph convolution, and modeling temporal dependency using a Transformer-based architecture.
- *RANSynCoders* [37]: RANSynCoders utilizes Auto-Encoder to extract features from the original asynchronous time series, and then performs Fourier transform analysis on the sequence features to learn phase differences. Thus align asynchronous time series and obtain synchronous time series representations.

Evaluation Metrics: We use Precision (Pre), Recall (Rec), and F1-score (F1) to evaluate the performance.

$$Pre = \frac{TP}{TP + FP},\ Rec = \frac{TP}{TP + FN},\ F1 = \frac{2 * Pre * Rec}{Pre + Rec}$$

where TP, TN, FP, and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively.

In practice, the anomalies often occur continuously so that the contiguous anomaly segments are formed. Therefore, it is acceptable if an alert is triggered within any subset of a contiguous anomaly segment. Thus, we adopt a commonly used point-adjust approach [2] to measure the performance. That is, if at least one observation in an anomalous segment is detected correctly, all the other observations in the segment are also treated as detected correctly, even if they are not.

Parameter Settings: We summarize the parameter settings of SGAT-AE as follows.

- *SGAT:* For the self-learning graph, we embed node IDs with length of 32, k with 10, hidden layers with 32 neurons, and the zoom factor η is 3. We use two layers of TLM and STLM. The number of channels in the first and second layers are 32 and 64, respectively.
- *AE:* AE is composed of two symmetrical deep networks with three fully-connected layers. The latent dimension of Z is set by default to 5 in SWaT, 3 in WADI, 3 in SMD, 1 in SMAP, 1 in MSL, respectively.
- *Data Preprocessing and Training:* We set by default the window size w to 15, and predict the future 3-step values ($p = 3$) with $d = 1$ for all datasets. The random seed is 666, the batch size of the random gradient descent is 1024, and the dropout rate is 0.1. After randomly initializing the learnable parameters, we train SGAT-AE up to 80 epochs by using the Adam optimizer with the default learning rate 0.001.

B. Results of Anomaly Detection

1) Comparison With Baseline Models: Table II shows the performances of different methods for anomaly detection in multivariate time series. We have the following observations:

1) In general, the prediction-based methods, such as LSTM-NDT and DAGMM, show unsatisfactory performance. Although these methods can capture temporal dependencies in multivariate time series, they ignore the hidden spatial dependencies, thus restricting their performances. GTA and GDN employ the graph structure to learn the hidden correlations and improves the performance. Although they use down-sampling strategy to reduce the noise impact, true anomalies are still missed.
2) The reconstruction-based methods, including AE and BeatGAN, exhibit poor performance since they are affected by overfitting. USAD employs the adversarial training to amplify the reconstruction error, and has achieved better performance. But these methods all ignore temporal dependencies, which limits the performance improvement. LSTM-VAE, OmniAnomaly, and TranAD adopt RNNs and Transformer to extract temporal dependencies while doing a reconstruction, respectively. They all exhibit good performance in SMD, SMAP, and MSL. However, they all ignore the hidden complex dependencies between multivariate. The deficiency is pronounced obviously in WADI of which the node dimension is highest. It can be observed that the graph learning based methods such as GDN, GTA, and our SGAT-AE outperforms others significantly since the graph structure helps to mine the spatial correlation among nodes. As for RANSynCoders, which is a time series representation learning based method. It exhibits significantly poor performance in WADI when the variation of data dissatisfies periodicity.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

TABLE III
COMPARISON WITH VARIANTS OF SGAT-AE

Method	SWaT			WADI			SMD			SMAP			MSL		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
RemMA	0.992	0.755	0.857	0.585	0.999	0.738	0.938	0.960	0.949	0.854	0.876	0.865	0.943	0.624	0.751
RemAE	0.996	0.696	0.820	0.999	0.181	0.306	0.920	0.971	0.944	0.900	0.611	0.728	0.864	0.752	0.804
RemSGAT	0.992	0.743	0.850	0.905	0.441	0.593	0.990	0.632	0.772	0.930	0.489	0.641	0.738	0.580	0.649
REMAT	0.913	0.801	0.853	0.999	0.573	0.728	0.815	0.999	0.898	0.717	0.580	0.641	0.855	0.919	0.886
LSTM-AE	0.976	0.801	0.880	0.907	0.562	0.694	0.932	0.971	0.951	0.945	0.545	0.692	0.760	0.819	0.788
CNN-AE	0.937	0.760	0.840	0.939	0.562	0.703	0.942	0.961	0.953	0.929	0.482	0.635	0.667	0.932	0.777
AGCRN-AE	0.955	0.893	0.923	0.891	0.761	0.821	0.946	0.980	0.963	0.885	0.881	0.883	0.817	0.899	0.856
SGAT-AE	0.949	0.959	**0.954**	0.827	0.999	**0.905**	0.941	0.994	**0.967**	0.885	0.910	**0.897**	0.921	0.951	**0.935**

3) SGAT-AE achieves the best performance in most datasets except slightly worse than USAD in MSL. Specifically, SGAT extracts temporal dependencies and hidden spatial dependencies simultaneously, and AE successfully reduces the impact of noise. Moreover, its adversarial training combines the advantages of the above two parts, and makes SGAT-AE more robust. More specifically, SGAT-AE improves the F1 score by 12.8% in SWaT, 20.3% in WADI, 1.8% in SMD, and 0.7% in SMAP over the most competitive methods, respectively, and the performance difference in WADI is the most statistically significant. SGAT-AE outperforms the other methods in terms of both the Precision and Recall, where it increases the Precision from 0.796 (GDN) to 0.827 and the Recall from 0.827 (TranAD) to 0.999.

2) Ablation Study: To evaluate the effects of different components in our model, we compare different variants of SGAT-AE, including:

- Removing moving average (**RemMA**).
- Removing AE (**RemAE**).
- Removing SGAT (**RemSGAT**).
- Removing adversarial training (**RemAT**): We remove the adversarial training process and the adjusted anomaly score S_t^* is aggregated as follows:

$$S_t^* = \beta_1 Score_{pred} + \beta_2 Score_{rec} \tag{18}$$

  where we adopt 0.5, 0.5 as the values of β_1 and β_2.
- Replacing SGAT with LSTM (**LSTM-AE**).
- Replacing SGAT with CNN (**CNN-AE**).
- Replacing SGAT with AGCRN [38] (**AGCRN-AE**): AGCRN is also a GNN-based prediction model with an adaptive graph structure. Specifically, it designs a node adaptive parameter learning module and an adaptive graph generation module to capture the dynamic spatial-temporal correlations in multivariate time series.

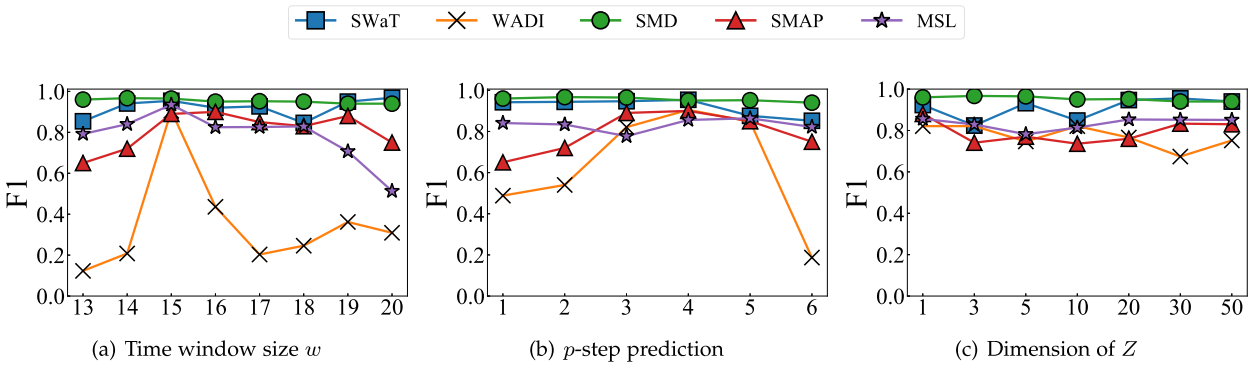
The results are shown in Table III. We have four observations.

1) Without moving average, the model has poor performance. This is because moving average can reduce the effect of noise and is beneficial to extract richer temporal patterns for prediction.
2) No matter removing SGAT or AE from SGAT-AE, the performance degrades. Specifically, RemSGAT is a reconstruction-based model, while RemAE is a prediction-based anomaly detection model. The performance of RemSGAT drops greatly, which indicates that SGAT makes a great contribution to SGAT-AE. Moreover, both models have limited performance without the adversarial training.
3) The performance decreases significantly without adversarial training. The reason is that the model degenerates into two independent parts SGAT and AE. SGAT focuses solely on minimizing prediction errors as its optimization objective, while AE's objective is restricted to minimizing reconstruction errors.
4) The experimental results show an interesting phenomenon. The methods that adopt AE, such as RemSGAT, LSTM-AE, and CNN-AE, perform worse than AE itself in datasets SMAP and MSL. It means that the additional network structures actually have a side effect. We believe the reason for the poor performance of RemSGAT is that the original AE is simplified in our model with fewer layers and parameters. As for the reason for the performance of LSTM-AE and CNN-AE, a probable explanation is that they are not able to construct graph structures and model the spatial relationship among node variables. Therefore, they fail to generate high-quality prediction results and adversarial samples.
5) The performance is improved by considering the hidden relationships between variables. Replacing SGAT with AGCRN has better performance than that with classical LSTM or CNN. The reason is that both LSTM and CNN ignore the hidden relationships between multivariate, while AGCRN discovers the relationship through a self-learning graph and a graph neural network and has more accurate prediction results.
6) SGAT-AE achieves the best performance, followed by AGCRN-AE. Since SGAT learns a directed graph structure while AGCRN learns an undirected one. In fact, the dependence patterns between multivariate are asymmetric, thus a directed graph learned by SGAT-AE makes more sense. Moreover, SGAT employs multiple convolution filters with different sizes to extract richer temporal features. Unlike GRU in AGCRN, the CNNs in SGAT-AE are computed in parallel. Therefore, SGAT-AE is more efficient and suitable for real-time anomaly detection.

Overall, the results indicate that SGAT-AE is capable of outperforming the competitors.

3) Parameter Study: We study the effects of different parameters on the performance of SGAT-AE. Note that, when we vary the value of a parameter, the other parameters are set to the default values.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.



(a) Time window size w (b) p-step prediction (c) Dimension of Z

Fig. 4. The effects of parameters.

Effect of the Window Size w: Since the time series has strong nonlinear temporal dependencies, the current observation X_t is checked whether it is abnormal by taking the sequence W_t with window size w as the input of SGAT-AE. Only short-term temporal dependencies are considered when w is small, while long-term temporal dependencies can be extracted when w is large. However, as w increases, the training speed slows down and the prediction difficulty rises. Fig. 4(a) shows the F1 scores when varying w in all datasets. We can see that when w is 15, the method achieves the best performance. So it is the best trade off between short-term and long-term temporal dependencies, and it enables both fast training and anomaly detection.

Effect of the Prediction Step p: SGAT conducts p-step prediction. On the one hand, the prediction error of SGAT increases as p increases. On the other hand, the predicted value $\tilde{X}_{t-p+1:t}$ deviates more from the ground truth, resulting in a larger deviation of the generated data $\tilde{W}_t$. It is equivalent to increasing the effect of noise disturbance, which is beneficial to avoid the overfitting in AE under the adversarial training. Fig. 4(b) shows that the F1 scores when varying p in all datasets. Specifically, F1 score first gradually increases with the increase of p, which is beneficial to the improvement of AE. Then, F1 score gradually decreases as p increases to a certain extent, indicating that the prediction error of SGAT is too large and weakens the overall performance. Therefore, p is set to a medium-large value 3.

Effect of the Dimension of Z: The dimension of the latent representation Z of AE is assumed to be smaller than that of the original data. On the one hand, if the dimension of Z is large, AE may only learn a mapping relationship from input to reconstruction, resulting in overfitting. On the other hand, the dimension of Z is too small, which may lead to underfitting. However, there is no standard strategy for choosing Z. Different datasets have different values, which are all set by experience and fine-tuned according to the experimental results. Fig. 4(c) shows the F1 scores when varying the dimension of Z in all datasets. We can see that the performance does not change significantly as the dimension of Z increases. It is because that the adversarial training employed in SGAT-AE improves the robustness of AE. According to the results in Fig. 4(c), the dimension of Z is set to 5, 3, 3, 1, and 1 in SWaT, WADI, SMD, SMAP, and MSL datasets, respectively.

TABLE IV
COMPARISON WITH DIFFERENT SCORING WEIGHTS

β_1	β_2	β_3	Pre	Rec	F1
1.0	0.0	0.0	0.954	0.695	0.804
0.0	1.0	0.0	0.810	0.862	0.835
0.0	0.0	1.0	0.893	0.900	0.896
0.5	0.5	0.0	**0.961**	0.877	0.917
0.2	0.2	0.6	0.923	0.870	0.895
0.05	0.05	0.9	0.949	**0.959**	**0.954**

Effect of the anomaly scoring weights. Different weights of β_1, β_2, and β_3 will result in different anomaly score S_t, and they are used to trade off between false negatives and false positives. A larger β_1 corresponds to giving more importance to the prediction of SGAT-AE in S_t, while β_2 corresponds to the reconstruction of AE, and β_3 corresponds to the adversarial training. The changes of scoring weights can tune the detection sensitivity.

We conduct multiple experiments of varying β_1, β_2, and β_3 in SWAT to evaluate the effect of the anomaly scoring weights. Due to the space limitation, Table IV only reports some representative experimental results of Precision, Recall, and F1-Score. It can be observed that the anomaly scoring strategy adopted in our method as $\beta_1 = 0.05$, $\beta_2 = 0.05$, and $\beta_3 = 0.9$ attains the best F1-score and Recall. It indicates that our strategy can not only achieve the best anomaly detection performance but also the highest sensitivity.

C. Results of Anomaly Interpretation

1) Comparison With Baseline Models: For a labeled anomaly X_t, we locate the root variables that cause the anomaly from both local and global perspectives based on the self-learning graph. We use AS_t to represent the root variables suggested by SGAT-AE, which are ordered by the errors. To evaluate the anomaly interpretation quality, we adopt the $HitRate@P\%$ proposed in [2] as follows:

$$HitRate@P\% = \frac{Hit@\lfloor P\% \times |GT_t| \rfloor}{|GT_t|} \tag{19}$$

where $|GT_t|$ is the length of ground truth array containing the variables indeed contributing to anomaly, and P is selected from

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

TABLE V
COMPARISON OF ANOMALY INTERPRETATION

Method	SMD		
	$P = 100$	$P = 150$	$P = 200$
OmniAnomaly	0.800	0.892	0.923
USAD	0.410	0.650	0.853
SGAT-AE	**0.834**	**0.911**	**0.944**

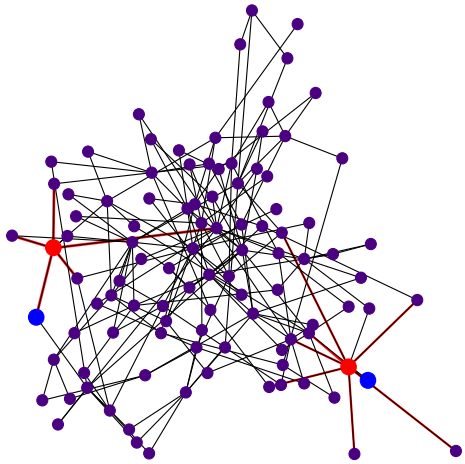
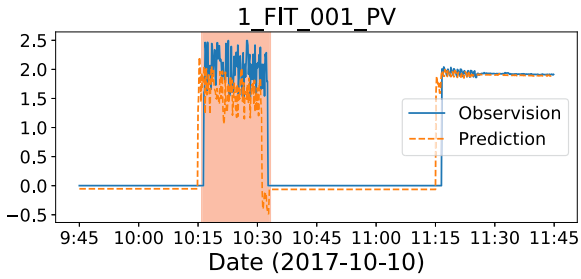
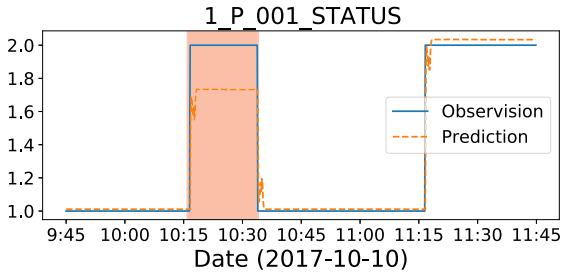


Fig. 5. The self-learning graph in WADI (the red nodes: $1_P_001_STATUS$ and $2_FQ_401_PV$, the blue nodes: $1_FIT_001_PV$ and $2_MCV_007_CO$)



(a) Attack Example 1



(b) Anomaly Prediction

Fig. 6. Attack example 1 on sensor $1_FIT_001_PV$ and the sensor $1_P_001_STATUS$ with the largest anomaly score.

$\{100, 150, 200\}$. $Hit@P\%$ equals the number of overlapping variables between GT_t and the top $\lfloor P\% \times |GT_t| \rfloor$ contributing variables in AS_t over $|GT_t|$. If the root variables we obtain is fewer than $\lfloor P\% \times |GT_t| \rfloor$, we choose the node with the second largest prediction error and its in-degree neighnors as root variables and repeat this until obtaining at least $\lfloor P\% \times |GT_t| \rfloor$ variables.

Since only SMD has the ground truth for anomaly interpretation, we compare SGAT-AE with OmniAnomaly and USAD in SMD. Note that, OmniAnomaly identifies the top-k variables with the lowest reconstruction probabilities as the interpretation of anomalies, while USAD obtains the top-k variables with the largest reconstruction errors. Table V shows the values of $HitRate@P\%$ when varying P in SMD. SGAT-AE achieves the best performance as we locate the root cause variables from both local and global perspectives based on the self-learning dependence graph.

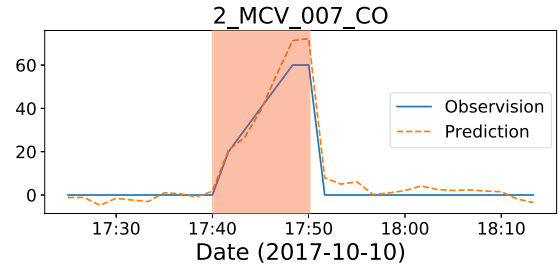
2) Case Study: We proceed to discuss how to locate anomalies. The edges in the graph structure learned by SGAT represent the dependency relationships among multivariate, thereby providing explanation for anomaly localization. Fig. 5 shows the self-learning graph in WADI dataset, followed by case studies to further demonstrate how to use this graph structure to locate and explain an anomaly.

We study a case where the cause of an anomaly is already known. As recorded in WADI dataset, at around 10:00 October 10, 2017, the sensor $1_FIT_001_PV$ that monitored the water flow was attacked, resulting in abnormal value via false reading. As shown in Fig. 6(a), the false reading results in a sharp rise in the observation value, but it is within the normal variation range for this sensor, so detecting this anomaly is nontrivial (The red shaded area indicates the anomalous time period).

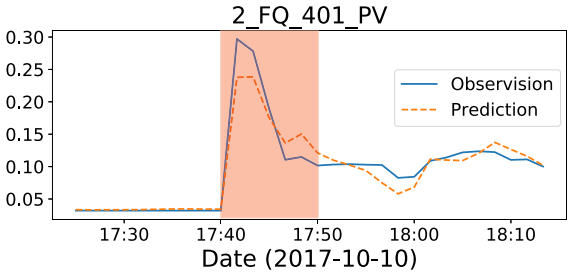
Fig. 6(b) shows the predictions and observations of sensor $1_P_001_STATUS$ with the highest anomaly score during this attack. Since the anomaly score is greater than the threshold τ, the observations are determined to be abnormal and sensor $1_P_001_STATUS$ is considered as an abnormal sensor with the highest score, which indicates that sensor $1_P_001_STATUS$ is the attacked object, or is closely related to the attacked sensor.

The self-learning neighbor relationships indicate the dependencies among nodes (sensors). As shown in Fig. 5, the red node on the right side represents the labeled abnormal sensor $1_P_001_STATUS$ and its neighbors are connected with red edges, while the blue node on the right side represents the real abnormal sensor $1_FIT_001_PV$. From a local perspective, $1_FIT_001_PV$ has the largest graph attention weight among the in-degree neighbors of $1_P_001_STATUS$, so the abnormal pattern of $1_P_001_STATUS$ is likely caused by its neighbor $1_FIT_001_PV$. From a global perspective, $1_P_001_STATUS$ is a dependent node, and $1_FIT_001_PV$ is the main cause node. Then, $1_FIT_001_PV$ is considered to be the cause of the anomaly because it has the highest abnormal score in all main cause nodes. Overall, $1_FIT_001_PV$ is regarded as the main cause of the anomaly from both local and global perspectives. In fact, $1_P_001_STATUS$ shows the valve status of the flow measured by $1_FIT_001_PV$. During the attack, the false reading of $1_FIT_001_PV$ leads its value to arise, and thus $1_P_001_STATUS$ responds to this change. SGAT-AE learns the dependence between $1_FIT_001_PV$

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.



(a) Attack Example 2



(b) Anomaly Prediction

Fig. 7. Attack example 2 on sensor $2_MCV_007_CO$ and sensor $2_FQ_401_PV$ with the largest anomaly score

and $1_P_001_STATUS$. The learned historical change pattern of $1_P_001_STATUS$ fails to predict the mutation situation, resulting in a large prediction error. Then, the cause of the anomaly can be located from the highest anomaly score.

Another anomaly case in WADI is recorded at around 17:40, October 10, 2017 that lasted for ten minutes. The sensor $2_MCV_007_CO$ was attacked, resulting in a continuous rise in the observation value. Fig. 7 shows the predictions and observations of sensor $2_MCV_007_CO$ itself and $2_FQ_401_PV$ with the largest prediction error during the attack. Since the anomaly score of $2_FQ_401_PV$ is greater than the threshold, the observations of this time period are correctly determined to be abnormal even though the anomaly score of $2_MCV_007_CO$ is within the threshold.

We observe that the observation value of sensor $2_FQ_401_PV$ also shows a rise during the attack, which indicates that it is affected by the attacked sensor. It is confirmed in the self-learning graph. As shown in Fig. 5, the attacked sensor $2_MCV_007_CO$ (the blue node on the left side) is the in-degree neighbor of sensor $2_FQ_401_PV$ (the red node on the left side). From the local perspective, the two sensors are regarded as the candidate nodes of the root cause variables of the anomaly with the largest prediction error. From the global perspective, $2_MCV_007_CO$ is considered as the abnormal node since it has the largest prediction error in all main cause nodes. So we accurately identify sensor $2_MCV_007_CO$ as the root cause node from the two perspectives. According to the case studies, we can see that SGAT-AE is effective for multivariate time series anomaly detection and anomaly localization.

VI. Related Work

A. Anomaly Detection

Anomaly detection aims to detect observations which significantly deviate from the other observations. Classical unsupervised learning methods consist of distance-based methods [6], [39], [40], [41], [42], density-based methods [43], cluster-based methods [44], and classification-based methods [8], [45]. Matrix Profile [41], [42], [46] is a series of state-of-the-art distance-based methods. However, most of them deal with dependencies in a linear manner.

Recently, many deep learning-based methods have been proposed to extract more complex nonlinear features. The mainstream methods can be divided into prediction-based methods [1], [9] and reconstruction-based methods [10], [11], [12], [13]. Specifically, prediction-based anomaly detection methods model the normal patterns of historical time series to predict the future trends, and thereafter employ the prediction errors to determine anomalies. These methods extract nonlinear temporal dependencies in multivariate time series, but they are easily affected by noise. Reconstruction-based methods learn latent representations of normal data patterns and then reconstruct the original input, and thereafter use the reconstruction errors to detect anomalies. Although these methods can reduce the influence of noise, they are prone to be overfitting and ignore nonlinear temporal dependencies. GANs [13], [14] and VAEs [12] are generative models for learning normal data distributions and use the reconstruction probabilities to determine anomalies. More recently, ensemble models [15], [16] are proposed to improve the performance by combining the advantages of multiple base models. In addition, transformer-based methods [17], [18] are proposed that use attention-based autoencoders for reconstruction. USAD [5] consists of two AEs sharing the same encoder for adversarial training, which allows to amplify the anomaly errors. RCA [15] employs collaborative AEs to jointly identify normal observations from the data while learning its feature representation. GAN ensemble [16] employs multiple GANs and can better model the distribution of normal data.

However, those methods do not consider the hidden relationships among variables. GDN [26] employs a self-learning structure approach with graph neural networks and then uses the attention weights to provide explainability for detected anomalies. It misses many anomalies with down-sampling strategy, which is not reasonable. Unlike above methods, we employ a self-learning graph and an attention network to discover the hidden relationships among multivariate and employ AE to reconstruct the normal patterns. A novel adversarial training is designed to amplify the anomaly error and improve robustness. In addition, we propose a graph based anomaly diagnosis method to explain anomalies.

B. Graph Based Multivariate Problems

As is known to all, many multivariate problems with spatial correlation can be solved by constructing graph structures since the edges of the graph more or less indicate the correlation of nodes.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

Existing approaches generally model the behavior of multivariate time series based on graph neural network (GNN) to extract spatial-temporal dependencies. They transform the multivariate time series forecasting problem as the graph forecasting problem by employing graphs to represent the dependent relationships between multivariate and employ GNNs to extract spatial dependencies.

The traffic forecasting problem is one of the typical multivariate time series forecasting problems, which has been extensively explored. STGCN [24] captures temporal and spatial dependencies by using 2D convolutional networks and graph convolutional networks, respectively. Graph WaveNet [33] learns a self-adaptive adjacency matrix and combines graph convolution with dilated casual convolution to capture spatial and temporal dependencies. MTGNN [29] is a general GNN-based framework designed specifically for multivariate time series data.

Moreover, in the process of building graph structures, self-learning (self-adaptive) graph structures can be used in such scenario. AGCRN [38] designs a node adaptive parameter learning module and a data adaptive graph generation module to capture the dynamic spatial-temporal correlations. [47] utilizes the lifelong graph learning algorithm when dealing with out of distribution detection problems by proposing a novel neighborhood aggregate method.

Last but not least, graph learning method could be combined with AE or adversarial learning. DAGNN [48] employs the dual generative adversarial networks and the graph neural networks to learn modality-invariant and discriminative common representations for cross-modal retrieval. Similar to DAGNN, [49] propose a method consisting of the dual generative adversarial networks and GNNs. The difference is that [49] employs a novel soft-multi-label contrastive loss for cross-modal retrieval.

C. Multivariate Time Series Anomaly Interpretation

Anomaly interpretation is to answer the question why an observation is detected as an anomaly, which can help analyze the anomalies and speed up troubleshooting, and thus is often required in practice. OmniAnomaly [2] identifies the top-k variables with the lowest reconstruction probabilities as the interpretation of anomalies, while DAEMON [23] identifies the top-k variables with the largest errors as the root cause variables. Anomaly interpretation is similar to the root cause diagnosis [28], [50]. In fact, anomaly interpretation can be regard as the upstream task of the root cause diagnosis, while the anomaly features obtained from anomaly interpretation are taken as the input of the root cause diagnosis model. Many studies regard the root cause diagnosis as a classification task. iSQUAD [4] is a root cause diagnosis framework designed for DB system, which requires causal hypotheses or labels of root causes. Exathlon [50] labels the ground truth root cause by introducing instances of six different types of anomalous events and then demonstrates explanation discovery tasks for root cause diagnosis. The study [28] uses the explainability scores to evaluate which autoencoder based anomaly detection methods are more explainable, rather than explaining individual anomaly. Different from those approaches, we focus on analyzing which main variables cause the anomaly from the self-learning dependency relationships among multivariate.

VII. Conclusion

We propose a novel adversarial graph neural network SGAT-AE for anomaly detection in multivariate time series. Specifically, SGAT learns the graph dependences among multivariate and models the historical patterns of multivariate time series to predict the future trends, while AE learns the latent representations of normal patterns and reconstruct the original input. The idea of SGAT-AE is to employ SGAT and AE for adversarial training to amplify the anomaly error so that the performance of the multivariate time series anomaly detection is improved. Moreover, we provide an intuitive and effective way to analyze the root cause of anomalies based on the self-learning graph. The extensive experimental results demonstrate that SGAT-AE outperforms the state-of-the-art methods in four real-world datasets, while the largest improvement of F1 score is 20.3% in dataset WADI.

References

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

[2] 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. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2828–2837.

[3] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. Crit. Inf. Infrastructures Secur.*, vol. 10242, 2016, pp. 88–99.

[4] M. Ma et al., "Diagnosing root causes of intermittent slow queries in large-scale cloud databases," *Proc. VLDB Endowment*, vol. 13, no. 8, pp. 1176–1189, 2020.

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

[6] M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.

[7] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," in *Proc. Int. Conf. Very Large Data Bases J.*, vol. 8, no. 3/4, pp. 237–253, 2000.

[8] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1999, pp. 582–588.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] A. Makhzani and B. J. Frey, "K-sparse autoencoders," in *Proc. Int. Conf. Learn. Representations*, 2014.

[11] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[12] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*.

[13] I. J. Goodfellow et al., "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[14] 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.*, 2019, pp. 4433–4439.

[15] B. Liu, D. Wang, K. Lin, P. Tan, and J. Zhou, "RCA: A deep collaborative autoencoder approach for anomaly detection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1505–1511.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.

[16] X. Han, X. Chen, and L. Liu, "GAN ensemble for anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4090–4097.
[17] 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, Jun. 2022.
[18] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *Proc. VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, 2022.
[19] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, 2011.
[20] B. Schölkopf, A. J. Smola, and K. Müller, "Kernel principal component analysis," in *Proc. Int. Conf. Artif. Neural Netw.*, 1997, pp. 583–588.
[21] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 665–674.
[22] A. Goodge, B. Hooi, S. Ng, and W. S. Ng, "Robustness of autoencoders for anomaly detection under adversarial impact," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 1244–1250.
[23] X. Chen et al., "DAEMON: Unsupervised anomaly detection and interpretation for multivariate time series," in *Proc. Int. Conf. Data Eng.*, 2021, pp. 2225–2230.
[24] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.
[25] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1234–1241.
[26] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4027–4035.
[27] H. Yu et al., "Taxi-based mobility demand formulation and prediction using conditional generative adversarial network-driven learning approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3888–3899, Oct. 2019.
[28] T. Kieu et al., "Robust and explainable autoencoders for unsupervised time series outlier detection," in *Proc. Int. Conf. Data Eng.*, 2022, pp. 3038–3050.
[29] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 753–763.
[30] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2015, pp. 103–112.
[31] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.
[32] B. Zheng et al., "SOUP: Spatial-temporal demand forecasting and competitive supply in transportation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 2034–2047, Feb. 2023.
[33] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.
[34] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv: 1710.10903*.
[35] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," 2017, *arXiv: 1711.00614*.
[36] B. Zong et al., "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Representations*, 2018.
[37] A. Abdulaal, Z. Liu, and T. Lancewicki, "Practical approach to asynchronous multivariate time series anomaly detection and localization," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 2485–2494.
[38] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 17804–17815.
[39] D. Amagata, M. Onizuka, and T. Hara, "Fast and exact outlier detection in metric spaces: A proximity graph-based approach," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2021, pp. 36–48.
[40] L. Tran, M. Mun, and C. Shahabi, "Real-time distance-based outlier detection in data streams," *Proc. VLDB Endowment*, vol. 14, no. 2, pp. 141–153, 2020.
[41] Y. Lu, R. Wu, A. Mueen, M. A. Zuluaga, and E. J. Keogh, "Matrix profile XXIV: Scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2022, pp. 1173–1182.
[42] C. M. Yeh et al., "Matrix profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets," in *Proc. Int. Conf. Des. Mater.*, 2016, pp. 1317–1322.
[43] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based GMM," in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 145–154.
[44] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," *Princ. Data Mining Knowl. Discov.*, vol. 2431, 2002, pp. 15–26.
[45] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.
[46] T. Nakamura, M. Imamura, R. Mercer, and E. J. Keogh, "MERLIN: Parameter-free discovery of arbitrary length anomalies in massive time series archives," in *Proc. Int. Conf. Des. Mater.*, 2020, pp. 1190–1195.
[47] M. Hoffmann, L. Galke, and A. Scherp, "Open-world lifelong graph learning," in *Proc. Int. Joint Conf. Neural Netw.*, 2023, pp. 1–9.
[48] S. Qian, D. Xue, H. Zhang, Q. Fang, and C. Xu, "Dual adversarial graph neural networks for multi-label cross-modal retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2440–2448.
[49] S. Qian, D. Xue, Q. Fang, and C. Xu, "Integrating multi-label contrastive learning with dual adversarial graph neural networks for cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4794–4811, Apr. 2023.
[50] V. Jacob, F. Song, A. Stiegler, B. Rad, Y. Diao, and N. Tatbul, "Exathlon: A benchmark for explainable anomaly detection over time series," *Proc. VLDB Endowment*, vol. 14, no. 11, pp. 2613–2626, 2021.



Bolong Zheng (Member, IEEE) received the PhD degree from the University of Queensland in 2017. He is a professor in Huazhong University of Science and Technology (HUST). His research interests include spatial-temporal data management. He has published more than 50 papers in prestigious journals and conferences in data management field such as SIGMOD, ICDE, VLDB, ACM Transactions, and IEEE Transactions.



Lingfeng Ming received the master's degree from the Huazhong University of Science and Technology (HUST). His research interests include spatial-temporal data management, fleet management, and reinforcement learning.



Kai Zeng received the bachelor's degree in software engineering from Zhejiang University and the PhD degree in computer science from UCLA. He is the Chief Architect of Huawei Cloud Data Warehouse Service. His research interest lies in large scale data intensive systems.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.



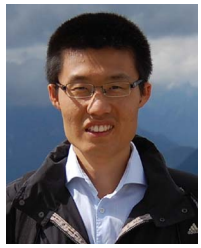
Mengtao Zhou is a reliability architect of Huawei Cloud Data Warehouse Service (DWS), and participating in the design and development of storage engine in DWS. He has many years of experience in cross domain operation and maintenance of network systems and data warehouse. He has participated in the research and development of large-scale network equipment and published 10+ patents (the first author).



Xinyong Zhang received the PhD degree in applied mathmatics from the Department of Mathematics, Tsinghua University in 2018. He is currently a maintenance system engineer of Huawei Cloud Data Warehouse Service (DWS). His research interests include massively parallel processing database and machine learning.



Tao Ye received the PhD degree in computer science from the Huazhong University of Science and Technology (HUST). He is the research and development director of Huawei Cloud Data Warehouse Service. His research interest lies in large scale data intensive systems.



Bin Yang received the PhD degree in computer science from Fudan University, China. He is a professor with East China Normal University, China. He has been a professor with Aalborg University, Denmark and was previously with Aarhus University, Denmark and the Max Planck Institute for Informatics, Germany. His research interests include machine learning and data management.



Xiaofang Zhou (Fellow, IEEE) received the PhD degree in computer science from the University of Queensland in 1994. He is the Otto Poon professor of engineering and chair professor of computer science and engineering with the Hong Kong University of Science and Technology. He is Head of Department of Computer Science and Engineering. His research interests include spatial and multimedia databases, high performance query processing, web information systems, data mining, and data quality management.



Christian S. Jensen (Fellow, IEEE) is a professor of computer science with Aalborg University, Denmark. His research concerns data analytics and management with focus on temporal and spatiotemporal data management. Christian is an ACM and an IEEE fellow, and he is a member of the Academia Europaea, the Royal Danish Academy of Sciences and Letters, and the Danish Academy of Technical Sciences. He has received several national and international awards for his research, most recently the 2019 IEEE TCDE Impact Award and the 2022 ACM SIGMOD Contributions Award. He was Editor-in-Chief of ACM TODS from 2014 to 2020 and an Editor-in-Chief of The VLDB Journal from 2008 to 2014.

Authorized licensed use limited to: University of Electronic Science and Tech of China. Downloaded on October 13,2026 at 12:20:00 UTC from IEEE Xplore. Restrictions apply.