The Journal of Supercomputing (2024) 80:8529–8549
https://doi.org/10.1007/s11227-023-05772-5

A graph attention network-based model for anomaly detection in multivariate time series

Wei Zhang[1] · Ping He[2] · Chuntian Qin[2] · Fan Yang[1] · Ying Liu[3]

Accepted: 27 October 2023 / Published online: 25 November 2023

Abstract
Anomaly detection of multivariate time series plays a growingly crucial role in intelligent operation and maintenance. Most existing anomaly detection models tend to focus on extracting temporal information while essentially ignoring the relationships among multiple sensors. Graph neural networks simulate multivariate inter-series relationships but suffer from the independent updating of graph structure from data. To overcome such limitation, we present a graph attention network-based model to learn the sensors relationships. It is equipped with a sustainable updating similarity-constrained graph structure learning method and a time series encoder. The graph learning method performs continuous updating of the graph structure. The encoder generates augmented and representative views along the temporal dimension. Our proposed model not only efficiently learns sensor relationships but also improves the ability of anomaly detection. Experiments are performed on publicly benchmark datasets, including SWaT, WADI, and HAI, with $F1$ scores of 92.98%, 91.19%, and 87.12%, respectively. This confirms that the proposed model outperforms the state-of-the-art in anomaly detection performance.

Keywords Anomaly detection · Multivariate time series · Graph neural network · Similarity constraint

1 Introduction

Modern industrial systems are becoming increasingly complex due to technological advancements, changing customer demands, globalization, and regulatory requirements [1]. In such systems, the operation and maintenance of interconnected sensors generate massive multivariate time series data with high dimensionality and spatio-temporal complexity [2, 3]. Efficient and precise anomaly detection techniques for multivariate time series enable companies to continuously monitor their key indicators and timely alerts for potential events [4, 5]. By employing these techniques,

Extended author information available on the last page of the article

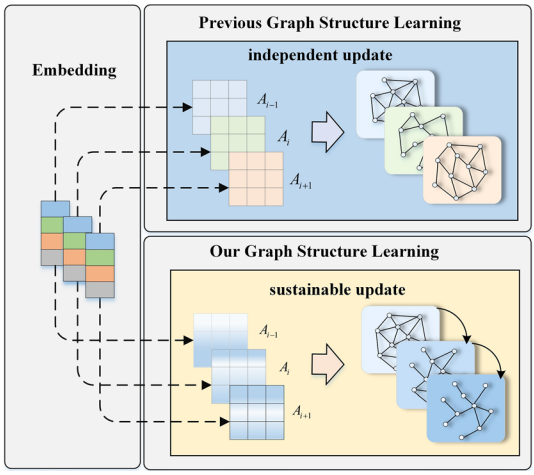


Fig. 1 Comparison of graph structure learning between existing method and our proposed method

operators can perform scheduled maintenance based on detected anomalies, minimizing downtime and emergency repairs. In terms of security, anomaly detection helps monitor data traffic and system behavior, identifying cyber security threats or unauthorized access attempts and triggering necessary security protocols. Recently, rapid progress in neural network-based methods has led to significant performance improvements in anomaly detection models.

Some typical methods [6–9] adopt models such as autoencoder (AE) and recurrent neural network (RNN) as their core modules to obtain representation and construct networks. However, such models completely focus on the extraction of temporal information, ignoring explicitly the correlations in multivariate time series. This correlation manifests that sensors in actual industrial systems are complex and nonlinearly interconnected. For instance, the flow change will affect the temperature and pressure [10]. Determining whether the entire system is running normally based on a single sensor may be challenging. Consequently, such correlations should be considered for anomaly detection in multivariate time series.

Note that graph neural networks (GNNs) [11, 12] can effectively extract and discover features and patterns from graph structured data. Several works have employed GNNs to model topological structure relationships among multiple sensors [10, 13–16]. As depicted in the top part of Fig. 1, such methods employ embedding vectors to capture each sensor's distinctive characteristics. Then, at iteration i, the graph structure relationships A_i are only learned from the sensor embedding. However, in these approaches, the learning of graph structure relationships between sensors is completely determined by the sensor embedding, i.e., the update of graph structure relationships A_i is not subject to A_{i-1}. Each iteration learns independently, neglecting the valuable information from previous ones. This learning scheme of graph structure between sensors lacks continuity modeling relationships, leading to inadequate modeling of sensor relationships. As a result, the motivation of this paper is to introduce a sustainable graph structure learning process. This sustainable updating ensures that the knowledge gained in each iteration is not discarded, leading to more accurate and robust anomaly detection.

In this paper, we present a Time-Series Graph Attention network (TS-GAT) approach to improve anomaly detection performance. As shown in the bottom part of Fig. 1, we propose a sustainable updating graph structure learning method that can continually learn relationships between sensors from the sensor embedding. Specifically, the graph structure relationships A_i are not only learned from the sensor embedding but also subject to A_{i-1}. Our proposed learning scheme can completely exploit the learned graph structure relationships A_{i-1} to implement sustainable updating, which improves the efficiency of modeling relationships between sensors. Besides, we construct a time series encoder for generating temporal views to help with multivariate time series. Our contributions are summarized as follows:

- We present a novel graph attention network-based model that concurrently learns from temporal and spatial perspectives, thereby providing more knowledge of the relationships between multivariable sensors.
- We present a sustainable updating method for graph structure learning combining a similarity-constrained loss and a threshold selection strategy. It facilitates more efficient learning of topological sensor dependencies.
- Experimental results on three publicly available real-world datasets demonstrate that our model surpasses the state-of-the-art approaches.

The remaining sections are organized below. Section 2 expounds relevant literature in deep learning anomaly detection models. Section 3 describes the methodology, including the model framework, training loss function, and anomaly score calculation. Section 4 explains the experimental results, which are displayed visually through tables and graphs. Section 5 briefly summarizes the proposed model.

2 Related work

Currently, numerous deep approaches are flourishingly implemented in anomaly detection. This section will thoroughly sort out the mainstream models.

AE has a strong nonlinear representation ability and employs the reconstruction residuals as a criterion for anomaly discrimination. Zong et al. [6] proposed an end-to-end hybrid model that combined AE with the Gaussian method. Another study [7] adopted the architecture of AE, consisting of one encoder and two decoders, and leveraged an adversarial learning technique to train the model. Naito et al. [17] adopted a two-stage AE model to enhance the interpretability and accuracy of anomaly detection. Moreover, RNNs capture the temporal characteristics for dealing with time series tasks. Park et al. [9] proposed a framework combining variational autoencoder (VAE) and long short-term memory network (LSTM) to tackle the issue of anomaly detection on multimodal data. Fährmann et al. [18] also was a hybrid model of VAE and LSTM, but it focused on lightweight. To characterize the temporal correlation of time series distributions, Li et al. [8] built a generative adversarial network-like architecture using LSTM and RNN as basic modules and performed anomaly detection synthetically through reconstruction and discrimination errors. In addition, the prevalent transformer [17] also appears in some studies.

Zeng et al. [19] constructed a deep transformer-based model, which was trained in an adversarial training manner and used anomaly scores combining reconstruction residuals and probability during detection. However, the models discussed above are essentially incapable of constructing unequivocally the relationships among multivariate sensors.

To deal with the above issue, some research has shifted to sensor relationship modeling. By virtue of excellent graph relationship modeling capabilities [20–23], GNNs have been extensively applied in anomaly detection. Deng and Hooi [14] adopted a graph attention network (GAT) to learn sensor relationships, which first employed the linear layer to embed sensors, then used the top-k technique to determine the graph structure. Tang et al. [16] was highly similar to [14], except that [16] utilized gate recurrent unit to learn temporal features. Zhan et al. [15] established a reconstruction GAT model focused on multi-scale feature learning. Regarding model design, Zhao et al. [13] and [10] were comparable. Both employed two GATs to simultaneously learn spatio-temporal linkages, optimizing the reconstruction and prediction networks. However, these models have a main drawback: the graph structure learning lacks continuity in modeling sensor relationships.

To compare the above models more clearly, we have compiled a summary Table 1. From Table 1, we can draw two conclusions. First of all, the non-graph models are hybrids of existing deep models, emphasizing the extraction of temporal features. However, these models ignore the modeling of sensor relationships. Secondly, existing graph models use GAT to solve the relationship modeling, but it has the defect of independent updating during training. In contrast to the above methods, our proposed model has the graph structure learning ability of sustainable updating to model sensor relationships. Next, the proposed model in this paper will be explained in detail.

3 Method overview

3.1 Problem statement

The objective of anomaly detection in multivariate time series is to find anomalous data during testing, which is usually implemented using the paradigm of unsupervised learning. Under this paradigm, the training dataset consists solely of normal data, whereas the test dataset contains both normal and abnormal data. The two data types are distinguished based on their patterns and behaviors. As depicted in Fig. 2, the highlighted area indicates an anomaly segment, with apparent fluctuations in the data, while the normal area is relatively stable.

Generally, the multivariate time series data are symbolized as $X = \{x_1, x_2, \ldots, x_n\} \in R^{n^* m}$, where n represents the length of data and m denotes the number of features observed from sensors. For any time step t, $x_t \in R^m$ is a m-dimensional vector. For input data, we perform data normalization and sliding window (window w and step c, as plotted in Fig. 3). The final output is a vector with the value $Y = \{y_1, y_2, \ldots, y_n\}$, where $y_t \in \{0, 1\}$ and $y_t = 1$ declares that the current sample is an anomaly.

Table 1 Comparison of the pros and cons of existing work

Existing work	Brief description	Advantage	Main drawback
Zong et al. [6]	A hybrid model of AE and Gaussian	Optimize hybrid models in an end-to-end manner	Not consider the relationships between sensors
Naito et al. [17]	A two-stage AE model	Provide some interpretability	
Park et al. [9], Fährmann et al. [18]	A hybrid model of VAE and LSTM	Multimodality lightweight	
Audibert et al. [7], Li et al. [8], Zeng et al. [19]	An adversarial model	Build temporal dependencies	
Deng and Hooi [14], Zhan et al. [15], Tang et al. [16]	A GAT model	Build sensors relationships	Independent update of graph relationships
Zhao et al. [13], Zhou [10]	Two GAT models	Build sensors relationships from two views	

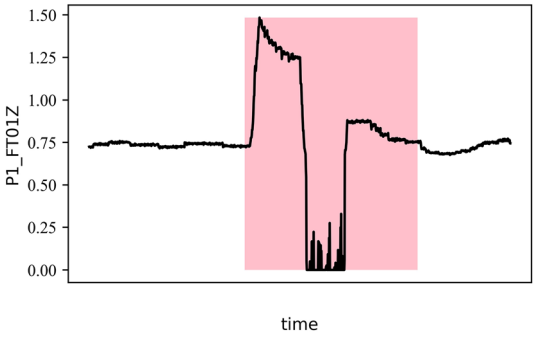


Fig. 2 Example of normal and anomaly data

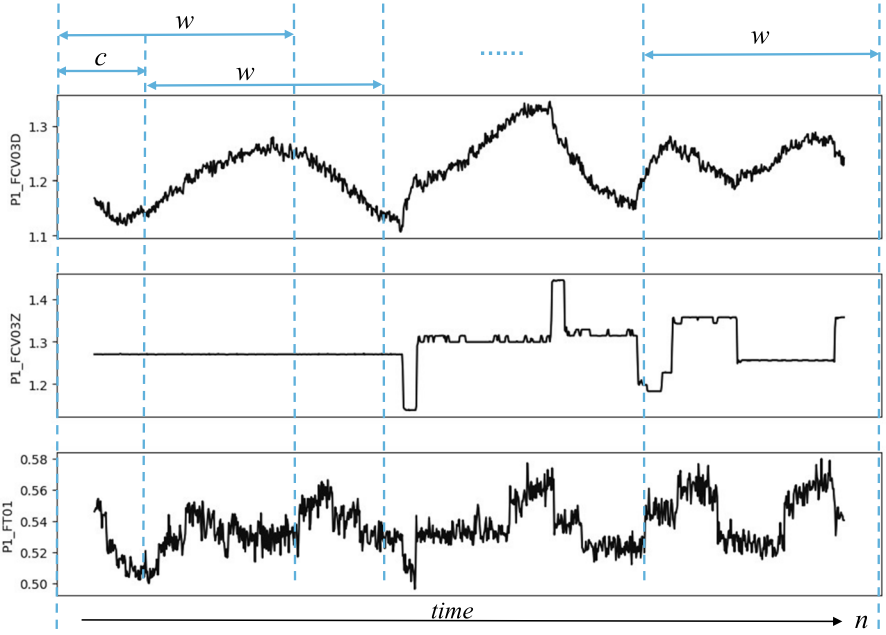


Fig. 3 Example of sliding window

3.2 Model architecture

We comprehensively elucidate the proposed model TS-GAT in this section, as depicted in Fig. 4. Our model mainly comprises three core components: time series encoder, sustainable updating graph structure learning, and forecasting-based decoder. Each is then thoroughly explained.

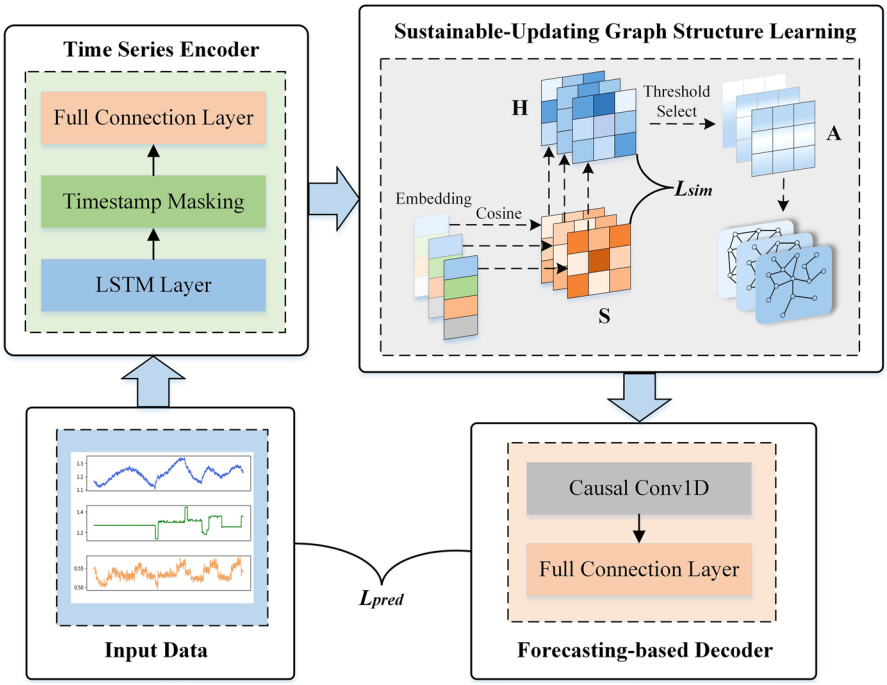


Fig. 4 A high-level framework of TS-GAT

3.2.1 Time series encoder

In multivariate time series data, distinct sensors often have unique properties, which may be related to each other in complex ways. For example, consider a smart home system with sensors for measuring temperature, humidity, and light. It is reasonable to assume that temperature and humidity sensors in different rooms will behave similarly. However, within the same room, there is often a close correlation between these sensors. Humidity decreases may accompany temperature increases. Therefore, a flexible and diverse way to describe the behavior of each sensor is required. Sensor embedding [14] can convert sensor data and features into embedding representations in high-dimensional vector space. Therefore, we propose a time series encoder to obtain embedding vectors from multivariate time series flexibly.

As described in Fig. 4, the time series encoder is formed with an LSTM layer, a timestamp masking layer, and a fully connected layer. Among them, LSTM can better capture long-term dependencies by introducing special memory units and gating mechanisms, as shown in Fig. 5.

For input data x, the LSTM layer handles long-term temporal dependencies in a recurrent and memorized manner, which can effectively gain temporal features. The timestamp masking layer occludes the latent features at stochastically selected timestamps to obtain more robust views. The fully connected layer maps the masked features to produce the representative sensor embedding z, as shown in Eq. 1.

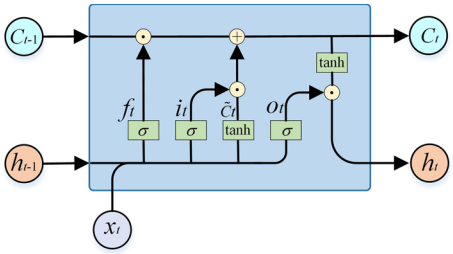


Fig. 5 A separate flow diagram of LSTM model. The LSTM consists of a forget gate f_t, an input gate i_t, an output gate o_t, and a memory unit

$$z = F(x) \tag{1}$$

where F is the time series encoder. $z \in R^d$, d denotes the embedding dimension.

Through the above sensor encoding, the model can map each unique sensor behavior into a semantically rich representation vector. This representation ability helps to understand the patterns and laws behind sensor behavior deeply and provides more information for subsequent task processing.

3.2.2 Sustainable-updating graph structure learning

Based on the sensor embeddings obtained from the above time series encoder, this section will introduce graph structure learning in detail. When modeling sensor relationships, the entire multivariate sensor is treated as a graph structure. Typically, since the relationships of the graph structure do not require symmetry, a directed graph is defined to represent the graph structure. The nodes represent sensors, and the edges represent relationships between sensors.

The proposed sustainable updating method integrates a similarity-constrained loss and a threshold selection strategy. First, we define a global and learnable similarity matrix H, created through random initialization, as shown in Eq. 2. Then, we leverage a threshold select strategy to determine the adjacency matrix A instead of the irrational top-k form, as indicated in Eq. 3. Concretely, assume that the nodes are related if the similarity in H is higher than or equal to threshold δ. The threshold selection strategy avoids redundant graph fully connected mode. Finally, we calculate another similarity S using the node i and all its potential neighbors j in the following Eq. 4. Through the similarity constraint loss of H and S, the learning of H ensures the sustainable updating of A from the data. The graph structure, i.e., the adjacency matrix A, can be continuously updated based on the previous learning, guaranteeing continuity.

$$H = Rand() \tag{2}$$

$$A_{ji} = 1\{(i,j) \in Index(H_{ij} \geq \delta)\} \tag{3}$$

$$S_{ij} = \frac{z_i^\top \cdot z_j}{\|z_i\| \cdot \|z_j\|} \tag{4}$$

where $H \in R^{m^* m}$ denotes a learnable matrix derived from *Rand* stochastic function. *Index* stands for the index pair operation of a matrix. z is the embedding vector obtained from the time series encoder. S denotes the similarity matrix. Both i and j belong to $[1, m]$.

Following the adjacency matrix A, we utilize GAT to capture the sensor dependencies, employing a flexible graph structure to represent the associations between individual sensors. As depicted in Fig. 6, the graph attention mechanism of GAT enables the network to learn the strength of relationships, i.e., attention coefficients, between each node and its neighboring nodes. This coefficient allocation allows the network to focus on neighboring nodes highly relevant to the current node, enhancing the model's expressive ability. Its calculation process is shown in Eqs. 5 and 6.

$$e_{i,j} = \text{Leaky}ReLU\left(a^T \cdot \left(Wz_i \oplus Wz_j\right)\right) \tag{5}$$

$$\alpha_{i,j} = \frac{\exp\left(e_{i,j}\right)}{\sum_{k \in \psi(i) \cup \{i\}} \exp\left(e_{i,k}\right)} \tag{6}$$

where a denotes the learnable weights. W is a learnable weight matrix. z represents the embedding vector obtained from the time series encoder. $\oplus$ means the operation of concatenation. $\psi(i) = \left\{j \mid A_{ji} > 0\right\}$ represents the set of neighbors of node i. $\alpha_{i,j}$ denotes the attention coefficients. By the α, the aggregate representation v_i of node i is defined as Eq. 7.

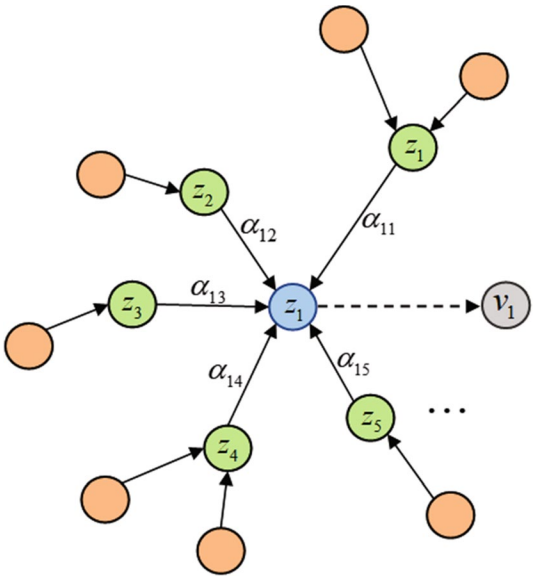


Fig. 6 A simple representation of graph attention mechanism

$$v_i = ReLU\left(\alpha_{i,i} W z_i + \sum_{j \in \psi(i)} \alpha_{i,j} W z_j\right) \tag{7}$$

3.2.3 Forecasting-based decoder

Through the representations v obtained above, we can gain the predicted output. The prediction paradigm can portray the future behavior of sensors by modeling historical observations against normal data. Therefore, we construct a forecasting-based decoder g, formed via a causal convolution [24] and a fully connection layer. The fully connection layer performs the feature mapping. The causal convolution is a strict time-constrained model that obeys the fundamental contextual dependencies of data modeling on temporal order. As shown in Fig. 7, causal convolution only uses historical data when calculating, making the model more suitable for capturing patterns of time series and helping to improve forecasting performance.

To better guarantee the model output $\hat{x}$, we integrate the information from the embedding vectors z and graph attention-based representations v and feed it to the decoder G, as shown in Eq. 8.

$$\hat{x} = G(z \circ v) \tag{8}$$

where $\circ$ is element-wise operation.

3.3 Model training and anomaly detection

Model training. To reduce the discrepancy between input data x and the forecasting output $\hat{x}$, we adopt the mean squared error (MSE) as a prediction loss in Eq. 9. Furthermore, we compute a similarity loss using two similarity matrices S and H, as shown in Eq. 10. Imposing the similarity loss to constrain the H can further ensure the sustainable updating of the adjacency matrix A. In sum, when training the proposed model, we minimize the loss function according to Eq. 11.

$$L_{pred} = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}_i\|_2^2 \tag{9}$$

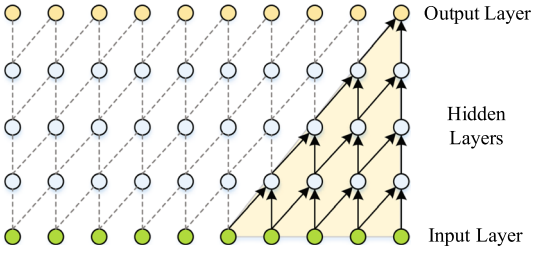


Fig. 7 Visualization of causal convolution

$$L_{sim} = \|S - H\|_2^2 \tag{10}$$

$$L = L_{pred} + \lambda \cdot L_{sim} \tag{11}$$

where $\lambda \in (0, 1]$.

Based on the above loss functions, we provide Algorithm 1 to clearly show the model's training process.

Anomaly detection. Following [14], we leverage graph deviation scoring (GDC) to gain the anomaly scores. GDC computes the anomaly statistics less sensitive to severe biases from specific sensor behavior. Firstly, we obtain the prediction error specified in Eq 12. To prevent the impact caused by different dimensions of sensors, it is more robust to normalize by the median and inter-quartile range (IQR) rather than its mean and standard deviation, as demonstrated in Eq 13. Then, the max function aggregates multiple sensors to achieve the final scores, as shown in Eq 14. When performing anomaly detection, the threshold is selected through the validation dataset. If the anomaly score exceeds the threshold, we label a timestamp in the test dataset as an anomaly.

$$err_i(t) = |x_i(t) - \hat{x}_i(t)| \tag{12}$$

$$p_i(t) = \frac{err_i(t) - \tilde{\mu}_i}{\tilde{\sigma}_i} \tag{13}$$

$$P(t) = \max\left(p_i(t)\right) \tag{14}$$

where $err_i(t)$ indicate the error of sensor i at time t. $\tilde{\mu}_i$ and $\tilde{\sigma}_i$ denotes the median and IQR.

4 Experiment

4.1 Datasets

We leverage three publicly available datasets throughout the experiments: Secure Water Treatment (SWaT) [13], Water Distribution (WADI) [10], and Hardware-in-the-loop-based Augmented Industrial control systems security (HAI) [25]. SWaT and WADI are operational water treatment testbeds primarily utilized for cyberattack and anomaly detection research. HAI, a cyber-physical system, can simulate various complicated processes to generate sophisticated attacks. Table 2 lists each dataset in full. We visualize some multivariate time series observations to observe the data clearly, as illustrated in Fig. 8.

Input: training dataset x; number of epochs *epochs*; TS-GAT including time series encoder F and forecasting-based decoder G; matrix H.
Output: trained TS-GAT
Init: TS-GAT weights, H
for $epoch = 1$ **to** *epochs* **do**
  $z \leftarrow F(x)$ /* calculate sensor embedding vector z using Eq. 1 */
  $S \leftarrow \text{Cosine}(z)$ /* calculate similarity matrix S using Eq. 4 */
  $A \leftarrow H$ /* determine adjacency matrix A using Eq. 3 */
  $\alpha \leftarrow \text{softmax}(\text{LeakyRelu}(A, z))$ /* calculate attention coefficients α using Eq. 5-Eq. 6 */
  $v \leftarrow \text{Relu}(A, \alpha, z)$ /* calculate aggregate representation v using Eq. 7 */
  $\hat{x} \leftarrow G(z, v)$ /* calculate output $\hat{x}$ using Eq. 8 */
  $L_{pred} \leftarrow x - \hat{x}$ /* calculate prediction loss L_{pred} using Eq. 9 */
  $L_{sim} \leftarrow S - H$ /* calculate similarity loss L_{sim} using Eq. 10 */
  $L \leftarrow L_{pred} + L_{sim}$ /* calculate total loss L using Eq. 11 */
  TS-GAT $\leftarrow$ update weights using L

Algorithm 1 The TS-GAT model training algorithm

Table 2 Dataset statistics

Dataset	Train	Test	Dimensions	Anomaly rate (%)
SWaT	496,800	449,919	51	11.98
WADI	1,048,571	172,801	123	5.99
HAI	309,600	291,600	59	3.96

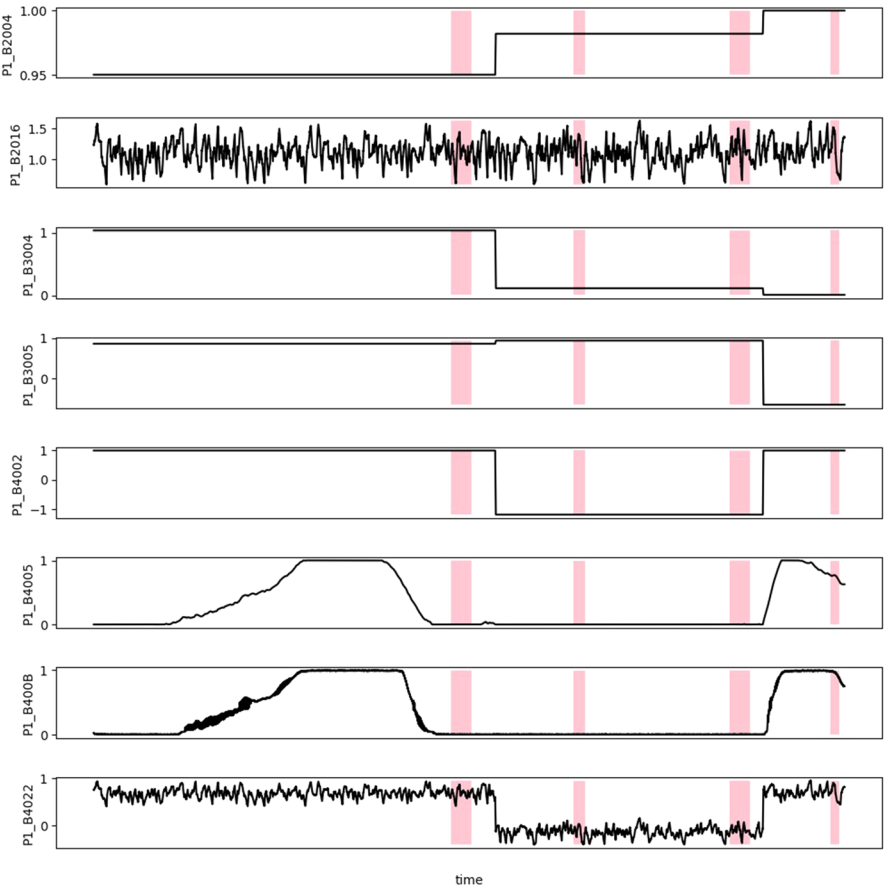


Fig. 8 Visualization of multivariate time series data. The pink highlights represent anomalous segments

4.2 Evaluation metric

As performance metrics for the proposed model, we employ precision (P), recall (R), and F1-score ($F1$). P denotes the proportion of true anomalous samples that the model correctly detects. R indicates the percentage of predicted anomalous points relative to all anomalies. The $F1$ comprehensively considers P and R. The

higher the above $F1$, the better the accuracy of anomaly detection, as defined in Eqs. 15, 16, 17.

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{17}$$

where TP, FP, and FN denote true positive, false positive and false negative, respectively.

Anomalies observations frequently take place continuously for a while generating abnormal segments. The previous work [26] provides a point-adjust technique that, if any anomaly observation within it is correctly recognized, deems the whole abnormal segment to be accurate. Audibert et al. [7], Zhao et al. [13], Su et al. [27] adopted such strategy in evaluation. Additionally, another work [28] focuses on the optimal threshold to evaluate the performance using the best $F1$ score (short $F1$ score hereafter). In this paper, we adopt the above $F1$ and the adjustment strategy to evaluate anomaly detection performance.

4.3 Implementation details

Our experiment uses PyTorch on a machine with NVIDIA RTX 3090 GPU. Following empirical values in the existing literature, the learning rate is 0.001, and the batch size is 128. We adopt Adam [29] as the network optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.999$. We set the training epoch as 100 and leverage the early stopping strategy with patience=8 to prevent overfitting. For the following parameters, we select the optimal value based on the experimental parameter adjustment results. The sliding window w and step c are 15 and 3. The embedding dimension d of the time series encoder is set to 128. The kernel size of casual convolution is 5. The δ in Eq. 3 equals 0.5. The λ in Eq. 11 is 0.8.

4.4 Comparison with state-of-the-art methods

We quantitatively compare the proposed TS-GAT model with existing approaches, including LSTM-VAE [9], DAGMM [6], MAD-GAN [8], USAD [7], MTAD-GAT [13], GDN [14], STGAT-MAD [15], GTA [30], HAD-MDGAT [10], GRN [16]. Table 3 presents the comparison findings of the models for P, R, and $F1$. The most outstanding performance is highlighted in bold, while the second is underlined.

From Table 3, several significant conclusions can be drawn. Firstly, our model consistently achieves the best $F1$ scores across all available datasets. It also acquires the highest recall in all circumstances other than HAI, where it performs worse than (81.84% vs. 86.58%). However, the precision of our model (93.14%) is dramatically

Table 3 Comparison results of TS-GAT and various models

Method	SWaT(%)			WADI(%)			HAI(%)		
	P	*R*	*F*1	*P*	*R*	*F*1	*P*	*R*	*F*1
LSTM-VAE	76.00	89.50	82.20	87.79	14.45	24.82	57.83	70.51	63.54
DAGMM	89.92	57.84	70.40	54.44	26.99	36.09	21.38	68.79	32.62
MAD-GAN	98.97	63.74	77.54	41.44	33.92	37.30	47.70	32.83	38.89
USAD	99.77	68.79	81.43	18.73	82.96	30.56	9.32	13.35	10.98
MTAD-GAT	90.30	82.10	86.00	72.00	51.80	60.20	13.30	34.29	19.17
GDN	99.35	68.12	81.00	**97.50**	40.19	57.00	88.37	60.32	71.70
STGAT-MAD	96.50	84.10	90.00	79.70	91.00	84.90	89.01	82.31	85.53
GRN	**99.86**	59.09	74.96	35.84	73.98	48.28	79.06	83.46	81.20
GTA	94.83	88.10	91.00	83.91	83.61	84.00	83.22	**86.58**	84.87
HAD-MDGAT	95.30	88.36	91.70	86.10	95.60	90.60	88.39	82.25	85.21
TS-GAT	96.27	**89.92**	**92.98**	85.30	**97.96**	**91.19**	**93.14**	81.84	**87.12**

higher than that of GTA (83.22%) on the same dataset. Secondly, models such as LSTM-VAE, DAGMM, and MAD-GAN, which do not account for sensor relationships, exhibit noticeably lower F1 scores than models that incorporate this relationship. This underscores the criticality of graph relationships in the context of anomaly detection. Thirdly, incorporating sensor relationships enhances the performance of graph models, such as MTAD-GAT, STGAT-MAD, etc. However, the inability to update the continuous graph structure hampers further performance enhancements. In contrast to these approaches, our model addresses this limitation, achieving superior performance. Finally, as obtained by the above models, the slight compromise between high recall and precision has critical implications for real-world applications. In most cases, *FP*s and *FN*s demand the execution of an inevitable trade-off. However, minimizing alarms triggered by *FP*s is prominent for applications' efficiency. But in the long run, detecting as many potential anomalies as possible can enhance the system stability since even rare abnormalities may lead to malfunction of the whole system. The maintenance technical staff with specialized knowledge will incline toward a high-level sensitivity instead of specificity to avoid missing informative and pivotal events [31].

4.5 Anomaly detection visualization

To intuitively show the performance of the TS-GAT model, we visualize the test values, predicted values, and anomaly scores on the HAI dataset in Fig. 9. For sensors *P1_FCV02Z* and *P4_ST_PT01*, values vary quite steadily, and the model detected all the abnormal segments accurately. The sensor *P4_ST_PO* is characterized by dense fluctuations and sharply distinguishes the middle two anomalies. We can see from Fig. 9 that the prediction and the test values essentially follow the same trends. It indicates that the forecasting of the model is precise enough. Under the

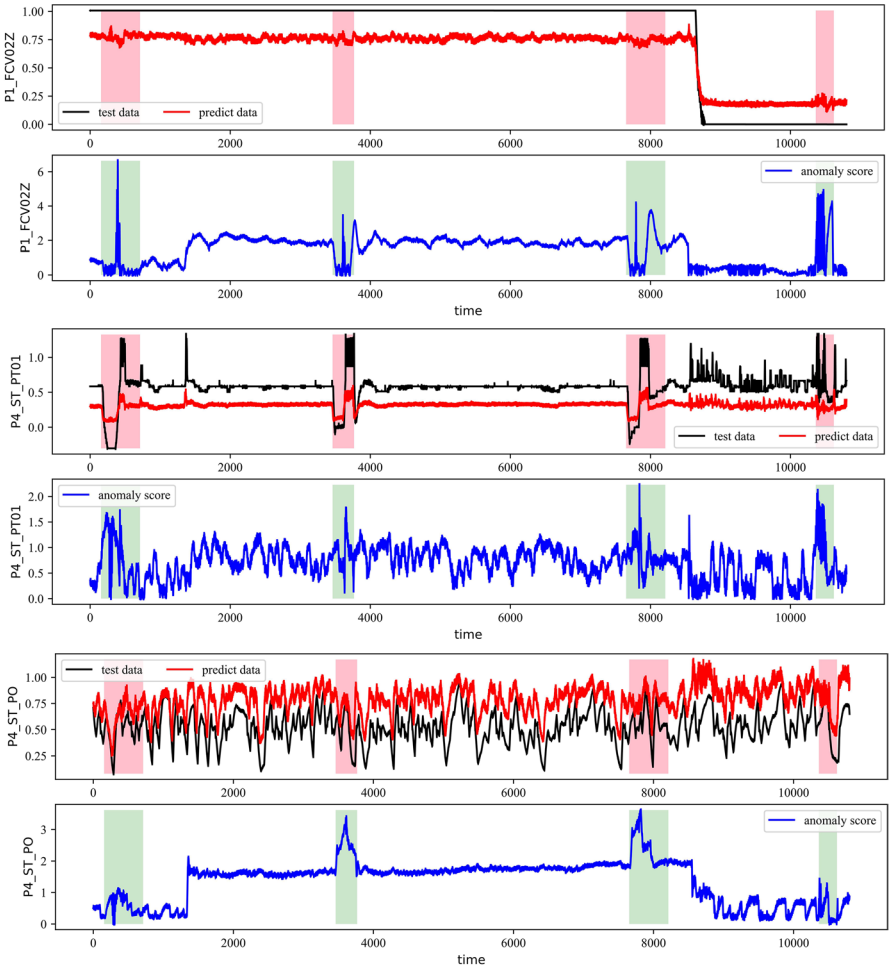


Fig. 9 Display of anomaly detection performance. Each sensor consists of two subgraphs and the highlighted area denotes the abnormal segment. The first represents test and predicted values, while the second denotes the anomaly scores

prediction-based paradigm, the proposed model possesses the ability to detect all anomalies.

4.6 Ablation study

To verify the efficacy and necessity of each component of TS-GAT, the ablation experiments with simplified counterparts of the model are carried out in Table 4. Concretely, we first evaluate the significance of the time series encoder by replacing it with linear embedding. Then, to discuss the effect of threshold selection strategy on graph structure learning, we adopt an entire static graph in which every node is

Table 4 Comparison of TS-GAT ablation experiments on $F1$

Model	SWaT(%)	WADI (%)	HAI (%)
TS-GAT	92.98	91.19	87.12
w/o TS Embedding	88.38	89.12	80.43
w/o Threshold Select	90.23	88.59	86.13
w/o Sim Loss	78.46	87.38	75.89

correlated with all other nodes. Lastly, we discard the similarity loss to evaluate the model's performance.

Obviously, these variants with the corresponding component removed clearly underperform the TS-GAT model. It reveals that each component is indispensable to the model and contributes to performance. In addition, the model's performance significantly deteriorates in the absence of similarity loss, thus confirming the efficacy of the sustainable updating graph structure learning method proposed in this paper.

4.7 Interpretability of model

Interpretability via time series embedding. We further investigate the interpretability of the time series encoder through the visualization of t-distributed Stochastic Neighbor Embedding (t-SNE) [32] in Fig. 10. We are concerned about whether the features mapped by the encoder can be reflected in the t-SNE space. For instance, the similarity of representation features may reflect similar sensor behaviors. According to Fig. 10, we confirm that some sensors essentially form local clusters. It proves that features obtained from the proposed model may accurately capture the behavioral similarities among local sensors.

Interpretability via learned relationships. According to the learned model, we discover the connections in the graph structure, which can further provide interpretability to know which sensors are relevant. As plotted in Fig. 11a, the sensors *P1_LCV01D* and *P1_FCV03Z* are linked, and their responses to the two abnormal segments are nearly identical. In Fig. 11b, sensor *P1_FT02Z* connects with *P1_FT03*, and both exhibit a strong reaction to the last abnormal segment. This is most likely caused by the connections between these sensors. Therefore, we declare that connected sensors typically behave similarly and are beneficial for anomaly detection.

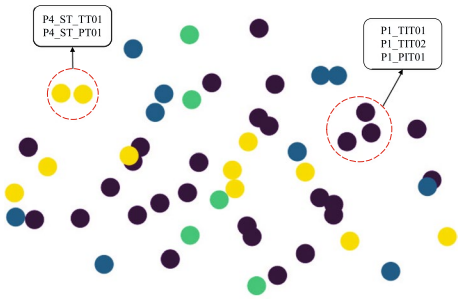
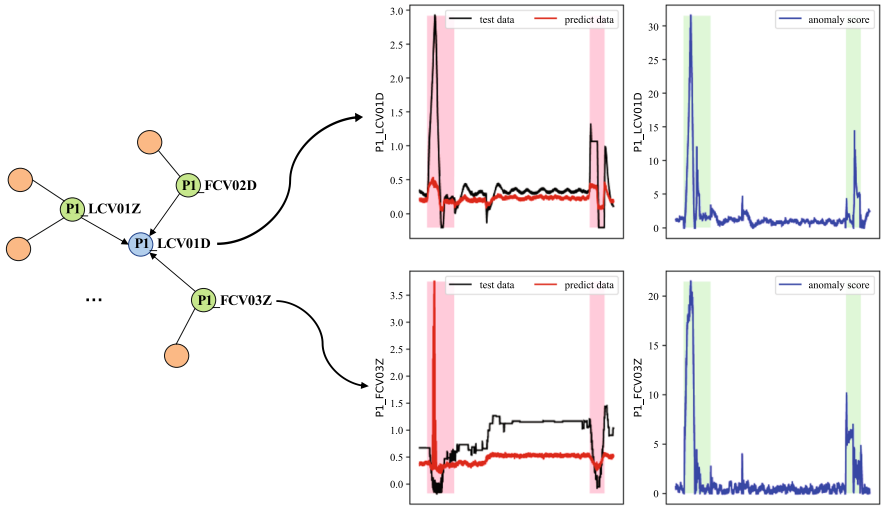
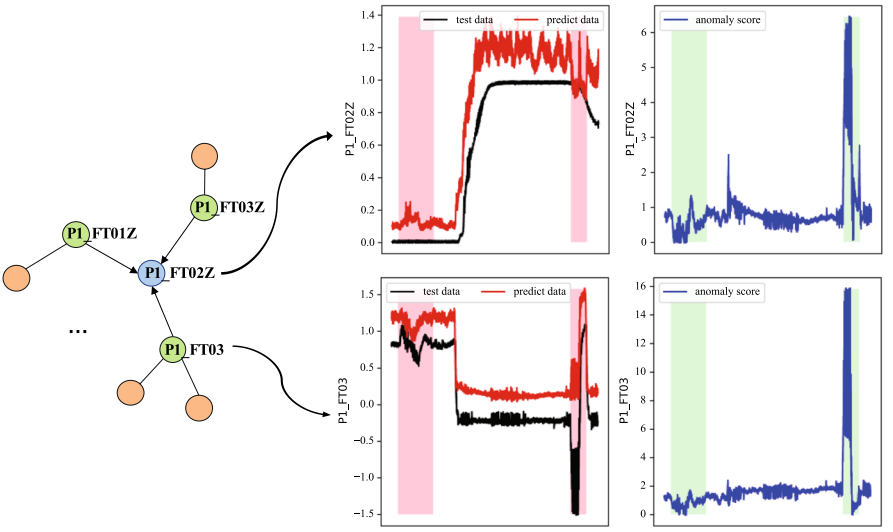


Fig. 10 t-SNE visualization of latent features mapped from time series embedding component on HAI dataset. Different colors indicate distinct processes. The nodes in the red dashed circle basically form a local cluster (color figure online)



(a) *P1_LCV01D* visualization of graph structure relationships on HAI.



(b) *P1_FT02Z* visualization of graph structure relationships on HAI.

Fig. 11 Left side displays the connection relationships, while the right contains four subplots, where the first column represents test and predicted values, and the second denotes anomaly scores. The highlights indicate the anomaly segment

5 Conclusion

In this work, we present a graph attention network-based anomaly detection approach for multivariate time series. The proposed model is integrated with a time series encoder, a sustainable updating graph structure learning module, and a forecasting-based decoder. The encoder has an excellent embedding ability for effectively generating temporal features. The graph learning module can improve the efficiency of modeling sensor relationships. The decoder incorporates causality based on time series, which gives excellent predictive capability. Experimental results on three real-world available datasets indicate the superior performance of the proposed model over state-of-the-art approaches. Remarkably, it also provides good interpretability.

This research monolithically focuses on the relational modeling of multivariate sequences from both temporal and spatial views to better handle anomaly detection. In the future, we will concentrate on detecting tiny abnormalities and explore more possibilities of graph neural networks for anomaly detection.

Funding The authors would appreciate the support from the Major National Science and Technology Special Projects (2016ZX02301003-004-007) and the NaturalScience Foundation of Hebei Province (F2020202067).

Data availability Data will be made available on request.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Zhang Z, Li W, Ding W et al (2022) STAD-GAN: unsupervised anomaly detection on multivariate time series with self-training generative adversarial networks. ACM Trans Knowl Discov Data 17(5):1–18. https://doi.org/10.1145/3572780
2. Zeng Z, Xiao R, Lin X et al (2023) Double locality sensitive hashing bloom filter for high-dimensional streaming anomaly detection. Inf Process Manag 60(3):103306. https://doi.org/10.1016/j.ipm.2023.103306
3. Blázquez-García A, Conde A, Mori U et al (2021) A review on outlier/anomaly detection in time series data. ACM Comput Surv 54(3):1–33. https://doi.org/10.1145/3444690
4. Yin C, Zhang S, Wang J et al (2020) Anomaly detection based on convolutional recurrent autoencoder for IoT time series. IEEE Trans Syst Man Cybern: Syst 52(1):112–122. https://doi.org/10.1109/TSMC.2020.2968516
5. Douiba M, Benkirane S, Guezzaz A et al (2023) An improved anomaly detection model for IoT security using decision tree and gradient boosting. J Supercomput 79(3):3392–3411. https://doi.org/10.1007/s11227-022-04783-y
6. Zong B, Song Q, Min MR, et al (2018) Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations, 1–19, https://openreview.net/forum?id=BJJLHbb0-
7. Audibert J, Michiardi P, Guyard F, et al (2020) USAD: unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 3395–3404, https://doi.org/10.1145/3394486.3403392

8. Li D, Chen D, Jin B, et al (2019) MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks. In: Text and Time Series: 28th International Conference on Artificial Neural Networks, 703–716, https://doi.org/10.1007/978-3-030-30490-4_56
9. Park D, Hoshi Y, Kemp CC (2018) A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. IEEE Robotics and Automation Letters 3(3):1544–1551. https://doi.org/10.1109/LRA.2018.2801475
10. Zhou L, Zeng Q, Li B (2022) Hybrid anomaly detection via multihead dynamic graph attention networks for multivariate time series. IEEE Access 10:40967–40978. https://doi.org/10.1109/ACCESS.2022.3167640
11. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations, 1–14, https://openreview.net/forum?id=SJU4ayYgl
12. Veličković P, Cucurull G, Casanova A, et al (2018) Graph attention networks. In: International Conference on Learning Representations, 1–12, https://openreview.net/forum?id=rJXMpikCZ
13. Zhao H, Wang Y, Duan J, et al (2020) Multivariate time-series anomaly detection via graph attention network. In: 2020 IEEE International Conference on Data Mining, IEEE, 841–850, https://doi.org/10.1109/ICDM50108.2020.00093
14. Deng A, Hooi B (2021) Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence, 4027–4035, https://doi.org/10.1609/aaai.v35i5.16523
15. Zhan J, Wang S, Ma X, et al (2022) STGAT-MAD: Spatial-temporal graph attention network for multivariate time series anomaly detection. In: IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 3568–3572, https://doi.org/10.1109/ICASSP43922.2022.9747274
16. Tang C, Xu L, Yang B, et al (2023) GRU-based interpretable multivariate time series anomaly detection in industrial control system. Comput Secur, p 103094. https://doi.org/10.1016/j.cose.2023.103094
17. Naito S, Taguchi Y, Nakata K, et al (2021) Anomaly detection for multivariate time series on large-scale fluid handling plant using two-stage autoencoder. In: 2021 International Conference on Data Mining Workshops (ICDMW), IEEE, 542–551
18. Fährmann D, Damer N, Kirchbuchner F et al (2022) Lightweight long short-term memory variational auto-encoder for multivariate time series anomaly detection in industrial control systems. Sensors 22(8):2886
19. Zeng F, Chen M, Qian C et al (2023) Multivariate time series anomaly detection with adversarial transformer architecture in the internet of things. Futur Gener Comput Syst 144:244–255
20. Wu S, Sun F, Zhang W et al (2022) Graph neural networks in recommender systems: a survey. ACM Comput Surv 55(5):1–37. https://doi.org/10.1145/3535101
21. Zhang M, Wu S, Yu X et al (2022) Dynamic graph neural networks for sequential recommendation. IEEE Trans Knowl Data Eng 01:1–12. https://doi.org/10.1109/TKDE.2022.3151618
22. Kumari A, Sahoo SP, Behera RK, et al (2020) Supervised machine learning for link prediction using path-based similarity features. In: 2020 IEEE 17th India Council International Conference (INDICON), IEEE, 1–7
23. Kumari A, Behera RK, Sahoo B, et al (2022) Prediction of link evolution using community detection in social network. Computing, 1–22
24. Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271https://doi.org/10.48550/arXiv.1803.01271
25. Shin HK, Lee W, Yun JH, et al (2020) Hai 1.0: Hil-based augmented ics security dataset. In: Proceedings of the 13th USENIX Conference on Cyber Security Experimentation and Test, 1–5
26. Xu H, Chen W, Zhao N, et al (2018) Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In: Proceedings of the 2018 World Wide Web Conference, 187–196, https://doi.org/10.1145/3178876.3185996
27. Su Y, Zhao Y, Niu C, et al (2019) Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 2828–2837, https://doi.org/10.1145/3292500.3330672
28. Li Z, Zhao Y, Han J, et al (2021) Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In: Proceedings of the 27th ACM SIGKDD

Conference on Knowledge Discovery and Data Mining, 3220–3230, https://doi.org/10.1145/3447548.3467075
29. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: International Conference on Learning Representations, 1–15. https://doi.org/10.48550/arXiv.1412.6980
30. Chen Z, Chen D, Zhang X et al (2021) Learning graph structures with transformer for multivariate time-series anomaly detection in IoT. IEEE Internet Things J 9(12):9179–9189. https://doi.org/10.1109/JIOT.2021.3100509
31. Han S, Woo SS (2022) Learning sparse latent graph representations for anomaly detection in multivariate time series. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2977–2986, https://doi.org/10.1145/3534678.3539117
32. Cieslak MC, Castelfranco AM, Roncalli V et al (2020) t-distributed stochastic neighbor embedding (t-SNE): a tool for eco-physiological transcriptomic analysis. Mar Genomics 51:100723. https://doi.org/10.1016/j.margen.2019.100723

Authors and Affiliations

Wei Zhang[1] · Ping He[2] · Chuntian Qin[2] · Fan Yang[1] · Ying Liu[3]

✉ Ping He
heping@scse.hebut.edu.cn

Wei Zhang
zhangv_869@163.com

Chuntian Qin
qinchuntian2023@163.com

Fan Yang
yangfan@hebut.edu.cn

Ying Liu
liuying770315@std.uestc.edu.cn

[1] School of Electronic and Information Engineering, Hebei University of Technology, No. 5340, Xiping Road, Tianjin 300401, Tianjin, China

[2] School of Artificial Intelligence, Hebei University of Technology, No. 5340, Xiping Road, Tianjin 300401, Tianjin, China

[3] School of Computer Science and Engineering, University of Electronic Science and Technology of China, No. 2006, Xiyuan Ave, Chengdu 611731, Sichuan, China