



# HAN-CAD: hierarchical attention network for context anomaly detection in multivariate time series

Haicheng Tao<sup>1</sup> · Jiawei Miao<sup>1</sup> · Lin Zhao<sup>2</sup> · Zhenyu Zhang<sup>2</sup> · Shuming Feng<sup>2</sup> ·  
Shu Wang<sup>3</sup> · Jie Cao<sup>4</sup>

Received: 9 January 2023 / Revised: 22 March 2023 / Accepted: 23 March 2023 /

Published online: 10 May 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Anomaly Detection in multivariate time series (MTS) plays an important role in many real-world Web services such as the Web traffic monitoring system. With abundant MTS data, exploiting the relationships among different variables, i.e., inter-variable relationships, is crucial for detecting anomalies. Recent studies have made substantial efforts to promote relationship learning from graph neural network. However, existing methods mostly neglect the distinctive characteristics of inter-variable relationships under different contexts, i.e., dynamics of inter-variable relationships. Therefore, we propose a “Hierarchical Attention Networks for Context Anomaly Detection” (HAN-CAD) model to fully exploit the inter-variable relationships and their dynamics. More concretely, we model each time series segment (context sequence) as a graph, where variables in the sequence are nodes and edges denote correlation patterns among variables. Then, the first graph attention layer is built on this graph to obtain the variable representation, which captures the relationships among different variables. Thereafter, the second attention layer outputs the sequence representation by integrating inter-variable relationships within the current context sequence. Finally, anomalies can be detected based on the reconstruction model, i.e., AutoEncoder. Extensive experiments on real-world datasets demonstrate that the proposed method can effectively detect anomalies in MTS and outperforms recent state-of-the-art methods.

**Keywords** Anomaly detection · Multivariate time series · Graph neural network · Hierarchical attention network

## 1 Introduction

Anomaly detection aims to identify rare observations that differ considerably from the majority of other ones [12, 20, 32]. In recent years, diverse research communities, e.g., cyber security [9], anomalous activity detection [24] and image processing [39], etc., have done

---

✉ Jie Cao  
cao\_jie@hfut.edu.cn

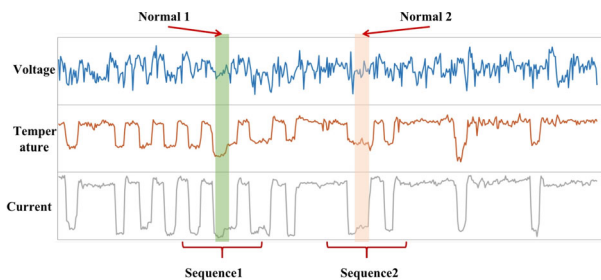
Extended author information available on the last page of the article

tremendous work on anomaly detection. Especially, in real-world scenarios, a wide range of applications [2] w.r.t. anomaly detection are involved with time series data. For instance, detecting malicious or abnormal activities in sequential sensor readings is vital for the control system of a power grid [18]. Hence, we focus on detecting anomalies in time series data.

Early work on anomaly detection mostly focused on univariate time series, which consider only a single time-related variable or metric. The key to detecting anomalies with a single metric is to learn the temporal dependencies of the time series. Traditional methods [7, 37] often use statistics such as mean, ARIMA [23] and Hidden Markov Models [29] to determine the temporal trends of sequences and thus obtain the expected value of a specified point. Recently, numerous deep learning-based methods [1, 33] have been proposed to enhance the ability to represent time series data and better capture its temporal dependencies, e.g., CNN [25], RNN [4] and LSTM [22], etc. However, in many circumstances, multivariate time series data are involved, e.g., detecting anomalies in server machines [17] based on multiple metrics such as CPU usage, bandwidth and network throughput. Considering the characteristics of MTS data, e.g., high dimensionality, complex interactions and temporal dependency among variables, it thus remains a challenging problem for anomaly detection.

Conventionally, MTS data consist of multiple univariate time series, and thus MTS anomaly detection can be divided into several univariate time series anomaly detection problems [14]. However, this intuitive approach of applying univariate-based methods completely ignores the relationships among different variables in MTS data. To address this issue, a few methods [11, 31] employ dimensionality reduction techniques for high-dimensional time series data, and then they apply univariate-based methods. To capture more complex relationships among variables, various methods [15, 21, 35] based on deep learning techniques have been proposed, such as Gated Recurrent Unit (GRU) [6] with AutoEncoder [27]. Nonetheless, most of these methods only model the multivariate relationships implicitly, which still have limits [8]. To overcome this problem, several deep learning methods [8, 36] have been proposed recently to explicitly construct the relationships among different variables using graph neural networks [19].

However, existing methods [8, 17, 36], whether modeling relationships among different variables implicitly or explicitly, tend to neglect that variable relationships can be different under different context sequences. In other words, previous methods model static correlations, while in reality, these correlations are indeed dynamic or evolving over time. As shown in Figure 1, a smart grid is equipped with three sensors, i.e., voltage sensor, temperature sensor and current sensor, to monitor health status. It is normal for the three sensors to follow the same trend, where temperature will always rise with increasing voltage or current, as shown in sequence 1. However, in sequence 2, temperature violates this trend. However, we can not treat it as an anomaly since the temperature in the plant is always brought down manually



**Figure 1** Illustration of dynamic relationships among variables

when the outdoor air temperature is high, which is obviously a common problem in a real-life world. Hence, the questions are how to capture temporal dependency under different contexts and integrate them with relationships among different variables.

To address aforementioned problems, we propose a novel Hierarchical Attention Network for Context Anomaly Detection (HAN-CAD) model to fully exploit the relationships among different variables and their temporal characteristics with regard to various context sequences. We propose using GRUs to obtain the initial feature representation of variables and sequences. Then, we construct a similarity graph for the variables and apply graph attention to capture variable-level correlations based on the similarity graph. Furthermore, another attention layer is proposed to learn the sequence-level temporal relationships between variables and sequences. By hierarchically integrating temporal relationships, we propose using the reconstruction model, i.e., AutoEncoder, to detect anomalies without requiring any ground-truth information. Specifically, our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to use graph attention mechanisms to capture dynamics of variable relationships and sequences for MTS context anomaly detection.
- Based on the hierarchical attention structure, we can obtain the temporal-aware and context-aware representation so as to better detect anomalies.
- We perform comprehensive experiments on three real-world datasets. Experimental results show that our proposed method is effective and outperforms the state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2, we overview the related work. Then, our proposed method is described in Section 3, including the problem statement and details of the proposed model. Experiments and empirical evaluations are reported in Section 4. Finally, Section 5 concludes the paper.

## 2 Related work

Anomaly detection in MTS is an important and challenging task in many real-world applications [2]. Extensive studies have been carried out by academic researchers and industry practitioners. In this section, we briefly review the related deep learning work for MTS anomaly detection since our proposed method is based on deep learning models.

Recent work on deep learning-based MTS anomaly detection can be categorized into three groups: prediction-based models, reconstruction-based models and hybrid models. All of these models [10, 13, 21, 28, 35, 40] follow a similar procedure, which involves feature extraction for MTS using deep learning techniques and construction of different task models. The major difference lies in how the anomaly score is determined, i.e., by the prediction error for prediction-based models, the reconstruction error for reconstruction-based models, and both errors for hybrid models. The core idea of prediction-based models is to predict the observation at time step  $\tilde{x}_t$  based on previous observations. Then, the observation at time step  $t$  can be determined as an anomaly if the prediction error between the true observation  $x_t$  and  $\tilde{x}_t$  is larger than the defined threshold. For instance, Bontemps et al. [3] proposed the first LSTM network for collective anomaly detection with several measures of predicted errors. Hundman et al. [13] proposed a dynamic thresholding method based on LSTM to predict future observations for spacecraft. Furthermore, Siami-Namini et al. [26] compared the performance for different variants in time series data and concluded that BiLSTM [34] is more suitable for time series prediction. Reconstruction-based models are widely used for anomaly detection, which try to obtain the representation of the whole sequence and compute

the reconstructed error of the observation at each time step. Most of the reconstruction-based models are based on two deep generative models, namely AutoEncoders (AEs) and Generative Adversarial Networks (GANs) [30]. For instance, Malhotra et al. [21] proposed the use of LSTM Encoder-Decode network, in which sequences are represented by LSTM and the reconstruction process is based on AE. To address the overfitting problem of AE, Zhou et al. [38] proposed a robust anomaly detection approach based on GAN by augmenting the data using the time warping technique. For more related work about deep learning methods, readers can refer to [2, 5].

It is worth noting that our work aims to capture variable relationships and its dynamic based on the graph neural network and attention mechanism. Among existing related work, both Zhao et al. [36] and Deng et al. [8] propose using graph attention technique to model inter-variable correlations. Zhao's model is based on a fully connected network, while in Deng's work, they propose using top  $K$  directed graph to learn the relationships between variables, which is more flexible. Li et al. [17] also propose the similar approach to capture inter-variable correlations based on hierarchical Variational AutoEncoder. In comparison, all previous work only considers the static correlations between different variables. Our proposed method employs a hierarchical attention mechanism to characterize the dynamic correlations for MTS anomaly detection based on graph neural network.

### 3 Methodology

In this section, we present the details and implementation of the proposed method.

#### 3.1 Problem statement

Let  $\mathcal{X}_N = \{\mathbf{x}_t\}_{t=1}^N \in R^{d \times N}$  denote a set of multivariate time-series data of length  $N$ , where  $\mathbf{x}_t \in R^d$  indicates the observation with  $d$  variables or features at time step  $t$  and  $N$  is the maximum length of timestamps. In this paper, we aim to detect whether the sequence  $\mathcal{X}_{i:j} = \{\mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_j\}$  contains abnormal activities, *i.e.*, anomalies, without using any groundtruth information.

As shown in Figure 1, inconsistent trends among different variables can indicate anomalies and are changing as time goes by. Therefore, to effectively detect anomalies in multivariate time-series, it is essential to capture relationships among multiple variables and learn their dynamic characteristics as anomalies evolve. We address these challenges by proposing a hierarchical attention network that focuses on two key issues: 1) capturing inter-variable correlations using a graph attention network from the variable-level perspective, and 2) characterizing the dynamic relationships between sequences and variables using GRUs and attention mechanism from the sequence-level perspective. Finally, anomalies can be detected based on an AutoEncoder network.

#### 3.2 Overview of proposed model

The overall framework of the proposed method is shown in Figure 2, which involves four main modules:

- **Feature Learning Module:** obtains the time-related features for variables and sequences using GRUs.

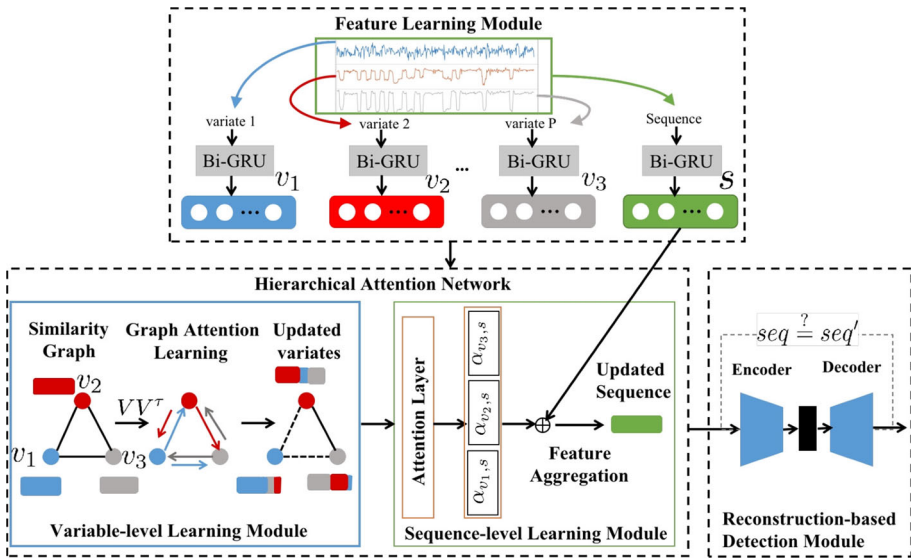


Figure 2 Overview of our proposed framework

- **Variable-level Learning Module:** learns the interactions between different variables using a graph attention network.
- **Sequence-level Learning Module:** learns the evolving relationships between sequences and variables using GRUs and attention mechanism.
- **Reconstruction-based Detection Module:** detects the sequence anomalies using an AutoEncoder network.

### 3.3 Feature representation

The Feature Representation Module takes a sequence of length  $L$ , i.e.,  $\mathcal{X}_L \in R^{d \times L}$ , as input and outputs the dense vectors as features for variables and sequences, respectively. In this work, we denote  $v_i$  and  $s$  as the dense vector for variable  $i$  and sequences, respectively. More specifically, variables in sequences are much related with different time steps. For instance, temperature sensors in a smart grid system exhibit varying statuses at different times of the day. High temperatures during midnight hours may indicate potential device malfunctions or abnormal operating conditions. These nuanced relationships can ultimately aid in identifying anomalous patterns.

To capture temporal dependencies and acquire better representations for variables and sequence, we use a Bidirectional Gated Recurrent Unit (Bi-GRU) network which can leverage information from both previous time steps (forward direction) and later time steps (backward direction). Specifically, Let  $v_i = \{x_1^i, x_2^i, \dots, x_L^i\}$  be the initial representation containing consecutive  $L$  observations for variable  $i$ . Then, the updated representation goes through following non-linear transformations sequentially:

$$z_t^i = \sigma(W_z^i[h_{t-1}^i, x_t^i]) \tag{1}$$

$$r_t^i = \sigma(W_r^i[h_{t-1}^i, x_t^i]) \tag{2}$$

$$g_t^i = \tanh(W^i[r_t^i h_{t-1}^i, x_t^i]) \tag{3}$$

and

$$\vec{h}_t^i = (1 - z_t^i)h_{t-1}^i + z_t^i g_t^i \tag{4}$$

where  $z$ ,  $r$  and  $g$  are the update gate, reset gate and candidate hidden state by integrating the reset gate, respectively.  $W_z^i$ ,  $W_r^i$  and  $W^i$  are all trainable weights.  $h_{t-1}^i$  is the output at time step  $t - 1$  for variable  $i$ .  $\vec{h}_t^i$  is the output for the forward directional GRU and meanwhile we get the backward directional GRU output, i.e.,  $\overleftarrow{h}_t^i$ . Thus, the final representation for variable  $i$  can be formulated as follows:

$$\mathbf{v}_i = h_L^i = \vec{h}_L^i + \overleftarrow{h}_L^i \tag{5}$$

Following the same procedure, we can also get the representation for the whole sequence  $\mathbf{s}$ .

### 3.4 Variable-level learning

In multivariate time-series data, learning variable feature independently cannot fully capture the characteristics of anomalies. Moreover, relationships among variables indeed reveal the distinctive time-related patterns, which are also favorable for detecting anomalies. Hence, rather than learning the variables independently, we investigate to leverage their mutual impacts and aggregate features of these variables at variable level. To address these issues, we propose to use graph attention network to model relationships and get updated features for variables.

Firstly, we construct a similarity graph among different variables, i.e., variable-level graph, in which nodes and edges represent variables and relationships, respectively. The variable-level graph  $G = \{V, E\}$  contains a node set  $V = \{v_1, v_2, \dots, v_d\}$  with features extracted by Bi-GRU, i.e.,  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d\}$ . The similarities between variables are computed using Eq.6 and then sorted in descending order. Then, we take top  $K$  most similar pairs as edges. Thereafter, Graph attention mechanism is introduced to model the interactions among variables and conduct variable-level feature learning. Furthermore, we propose using multi-head attention to extract robust features for variables. Finally, the feature representation for each variable is formed by a weighted sum of all connected node features, which can be formulated as follows:

$$s_{ij} = \frac{\mathbf{v}_i^T \mathbf{v}_j}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\|} \tag{6}$$

$$\mathbf{v}_i = \sigma\left(\frac{1}{H} \sum_{h=1}^H \sum_{j \in \mathcal{N}_i} \alpha_{ij}^h \mathbf{v}_j\right) \tag{7}$$

where  $H$  is the number of heads,  $\mathcal{N}_i$  is the set of neighbors of variable  $i$  and  $\alpha_{ij}$  is the attention weight which indicates the relevance of variable  $j$  to variable  $i$  which can be computed by:

$$r_{ij} = \text{LeakyReLU}(W^r(\mathbf{v}_i \oplus \mathbf{v}_j)) \tag{8}$$

$$\alpha_{ij} = \frac{\exp(r_{ij})}{\sum_{j=1}^{|\mathcal{N}_i|} \exp(r_{ij})} \tag{9}$$

where  $\oplus$  denotes concatenation and  $W^r$  is the trainable weight.

### 3.5 Sequence-level learning

In addition, relationships between variables are not stable and always evolve over time. Especially for variables with strong correlations, their anomaly patterns might vary dramatically in different sequences. Previous works treat variables and sequences equally and assigned the same weights to them, which cannot reveal the characteristics of sequence impacts on the variables. In order to capture sequence-level dependency, we propose using another attention mechanism to learn the interaction of variables and sequences. Specifically, the attention weights  $\beta_j$  can be computed as follows:

$$m_j = \text{LeakyReLU}(W^m(\mathbf{v}_j)) \quad (10)$$

$$\beta_j = \frac{\exp(\mathbf{s}^\top m_j)}{\sum_{j=1}^d \exp(\mathbf{s}^\top m_j)} \quad (11)$$

where  $\mathbf{s}$  and  $W^m$  are the feature vector for sequence and the trainable parameters, respectively.

After obtaining the attention weights for sequence, the updated sequence feature representation can be computed as follows:

$$\mathbf{s}' = \beta_0 \mathbf{s} \oplus \sum_{j=1}^d \beta_j \mathbf{v}_j \quad (12)$$

Specifically, the final feature representation is aggregated by concatenating the original representation of sequence and representation of variables with evolving relationships among variables.

### 3.6 Reconstruction-based detection

Following the above hierarchical attention process, we can obtain the final feature representation  $\mathbf{s}$  for a sequence  $\mathcal{X} = \{x_1, x_2, \dots, x_L\}$ . Then, we employ AutoEncoder network to reconstruct the sequence. Let  $f_e(\cdot)$  denote the Encoder and  $f_d(\cdot)$  denote the Decoder. Given the feature vectors  $\mathbf{s}$  for the sequence  $\mathcal{X}$ , the encoder maps the  $\mathbf{s}$  into the latent representation  $\mathbf{z}$  and decoder reversely maps the  $\mathbf{z}$  into the reconstructed  $\hat{\mathcal{X}}$  as follows:

$$\mathbf{z} = f_e(\mathbf{s}', W^e) \quad (13)$$

$$\hat{\mathcal{X}} = f_d(\mathbf{z}, W^d) \quad (14)$$

where both of  $W^e$  and  $W^d$  are trainable parameters. Finally, the reconstruction loss can be defined as follows.

$$\text{Loss} = \frac{1}{L} \sum_{i=1}^L \|x_i - \hat{x}_i\|_2 \quad (15)$$

where  $\|\cdot\|_2$  denotes  $\ell_2$  normal. The sequence can be identified as an anomaly if the reconstruction error is larger than a threshold. In this paper, we adjust the threshold to maximize the F1 score.

The whole learning process of hierarchical attention method is presented in Algorithm 1.

---

**Algorithm 1** Training for HAN-CAD: Hierarchical Attention Network for Multivariate Time-series Anomaly Detection.
 

---

**Require:** a multivariate time-series set:  $\mathcal{X}_N = \{\mathbf{x}_t\}_{t=1}^N \in R^{d \times N}$ ; batch size  $B$ ; training epochs  $I$

- 1:  $epoch = 0$
- 2: **while**  $epoch \leq I$  **do**
- 3:   sample a batch of  $B$  time series;
- 4:   **for** a sample  $\mathcal{X}_p$  in  $B$  **do**
- 5:      $\mathbf{s}$  and  $\mathbf{v} \leftarrow$  Learn the initial feature vectors for the sequence and variables by Eq. 5
- 6:     update the feature vectors  $\mathbf{v}$  for variables by Eq. 7
- 7:     update the feature vectors  $\mathbf{s}$  for sequence by Eq. 12
- 8:   **end for**
- 9:    $Loss \leftarrow$  Calculate samples' reconstruction loss by Eq. 15
- 10:   update all the weights in Eqs. 1, 2, 3, 8, 10, 13 and 14
- 11:    $epoch = epoch + 1$
- 12: **end while**

---

## 4 Experimental results

In this section, we conduct experiments on three real-world datasets and evaluate the effectiveness of the proposed method compared with four state-of-the-art methods.

### 4.1 Datasets and metrics

Experiments are conducted on three publicly available datasets that have ground truth information, which are described as follows:

- **ASD** (Application Server Dataset) [17]: This dataset is a collection of 45-day-long status data from 12 servers in a large internet company. The status of servers is monitored based on 19 metrics ( $d = 19$ ), e.g., CPU-related metrics, memory-related metrics, network metrics and etc.. In our experiments, we only used data from one server to speed up training. Additionally, we used 66.7% of data for training and the rest for testing.
- **SMD** (Server Machine Dataset) [27]: This is another dataset for servers, which collected 5-week-long MTS data. SMD contains 12 servers with 38 metrics, including CPU load, network usage, memory usage etc.. In our experiments, we split the dataset from one server into two parts: 50% of data was used for training, and the remaining 50% was used for testing.
- **WADI** (Water Distribution) [16]: This dataset contains 16-day-long data collected in a water distribution system. Several cyber-attacks were executed, which caused various anomalies in the system. In our experiments, we choose five days of normal data for training and the remaining days containing anomalies for testing.

The detail statistics of three real-world datasets are shown in Table 1

**Table 1** Dataset Statistics

Datasets	ASD	SMD	WADI
#Features	19	38	112
#Train	8,640	28,479	335,999
#Test	4,320	28,479	172,801
Anomalies (%)	3.40	5.84	5.85



In order to compare with other baselines, we evaluated the performance of the proposed method using three commonly used metrics for detection tasks, i.e., precision, recall and F1-score. Furthermore, it is worth mentioning that any sequence containing at least one anomaly is considered as being correctly detected.

## 4.2 Baselines

We extensively compared the performance of the proposed method with five state-of-the-art MTS anomaly detection methods as follows:

- **LSTM-AE** [21] is a classic reconstruction-based anomaly detection method, which exploits temporal dependencies using LSTM and detects anomalies using AutoEncoder.
- **MAD-GAN** [16]: is another reconstruction-based anomaly detection method based on GAN.
- **MTAD-GAT** [36]: is a state-of-the-art method that can efficiently model the relationships between variables using a graph attention network. Anomaly score can be inferred from the reconstruction error and prediction probability.
- **GDN** [8]: models the structure among different variables using a graph neural network and provides interpretability for anomalies based on the attention weights.
- **InterFusion** [17]: captures the relationships among metrics as well as temporal dependency based on hierarchical variational AutoEncoder. InterFusion also provides interpretations based on MCMC methods.

## 4.3 Experimental setup

In our experiments, the length of sliding window is set as 100, 100 and 30 for ASD, SMD and WADI, respectively. The models are trained using the Adam optimizer with a learning rate  $5e-4$ . The sizes of representation for variables and sequences are both 64. We also use dropout to reduce overfitting and the dropout probability is 0.2. The number of header in multi-headed attention is 2. The state-of-the-art methods and the proposed method are trained on a Windows server with 3.60 GHz Intel I9-9900k CPU and 11 GB Nvidia GeForce RTX 2080 Ti GPU.

## 4.4 Comparison of performance

Firstly, we compare our proposed method with five baselines on three real-world datasets. In particular, experiments are repeated 5 times, and average performance and standard deviation are reported. Table 2 presents the results for all methods using precision, recall and F1 scores, in which the best results are bold-faced. In general, HAN-CAD shows promising results in most cases on Precision and Recall and outperforms all baselines on F1, which demonstrates the effectiveness of our method. Especially, we have the following two observations:

- From the comparison results on the three datasets, we can observe the evident order of six methods from high to low in terms of the three metrics: “HAN-CAD → InterFusion → GDN → MTAD-GAT → MAD-GAN → LSTM-AE”. Further, it is worth noting that all methods capturing correlations between variables, i.e., InterFusion, MTAD-GAT, GDN and our method, perform better than traditional reconstruction-based methods, which reveals the importance of inter-variables relationships for MTS anomaly detection.

**Table 2** Performance(%) comparison of different methods on three real-world datasets

Dataset	Method	Precision	Recall	F1
ASD	LSTM-AE	83.12±2.56	73.42±1.74	79.51±1.17
	MAD-GAN	86.37±4.13	77.19±3.02	81.93±3.38
	MTAD-GAT	88.92±0.79	87.24±1.06	87.25±0.94
	GDN	92.15±0.84	84.72±0.97	87.14±0.86
	InterFusion	<b>97.78±1.05</b>	88.15±0.82	94.52±0.86
	HAN-CAD	96.82± <b>0.76</b>	<b>92.29±0.61</b>	<b>96.18±0.72</b>
SMD	LSTM-AE	87.26±1.63	80.27±1.17	82.94±1.26
	MAD-GAN	98.52±2.08	81.75±2.93	89.37±2.11
	MTAD-GAT	94.45±0.93	87.12±0.81	91.26±0.83
	GDN	92.15±1.45	94.91 ±0.89	93.35±0.92
	InterFusion	94.15±0.96	93.90± <b>0.78</b>	94.02±0.73
	HAN-CAD	<b>97.26±0.42</b>	<b>95.81±0.91</b>	<b>96.16±0.53</b>
WADI	LSTM-AE	81.72±1.28	73.25±2.04	78.14±1.19
	MAD-GAN	89.01±2.81	72.84±1.13	79.08±1.62
	MTAD-GAT	91.62±0.63	77.15±0.94	85.17± <b>0.78</b>
	GDN	85.24±0.84	80.92± <b>0.91</b>	83.15±0.87
	InterFusion	90.15±1.01	<b>87.61 ±0.92</b>	88.15±0.93
	HAN-CAD	<b>96.87±0.61</b>	86.95±1.73	<b>92.73±1.08</b>

• All the baselines obtain lower measure scores on the WADI than on other datasets, which implies anomalies in WADI are more difficult to detect. This is probably because WADI is consisted of 112 variables and thus has more complex relationships among variables. However, our proposed method, HAN-CAD, is able to effectively capture these complex relationships through the use of dynamic context-based modeling. Therefore, HAN-CAD significantly outperforms the baseline methods even on the challenging WADI dataset and achieves high performance measures.

In our experiments, we used the sliding window technique to obtain context sequences. To validate the effects of context sequences for different methods, we compared HAN-CAD with MTAD-GAT, GDN and InterFusion with different lengths of sliding windows in terms of F1 score. As shown in Figure 3, our method consistently showed promising results with different lengths on all three datasets. Moreover, our method presented a stable trend, in which HAN-CAN achieved the best F1 when length is 100. Whereas, two graph neural network-based (GNN-based) methods show more fluctuations, which indicates that integrating the inter-variable relationships and context sequence would make GNN-based MTS anomaly detection more robust.

Furthermore, we also investigate how the graph structure impacts the effectiveness of MTS anomaly detection based on GNN. Figure 4 shows the results with different ratios of edges in terms of F1 score for MTAD-GAT, GDN and HAN-CAN on the three datasets. The findings show that our method outperforms the other two GNN-based methods in most settings. In addition, it is observed that all GNN-based methods perform worse on sparse graphs, which may be due to the difficulty in extracting non-linear structural features for relationships in such graphs.

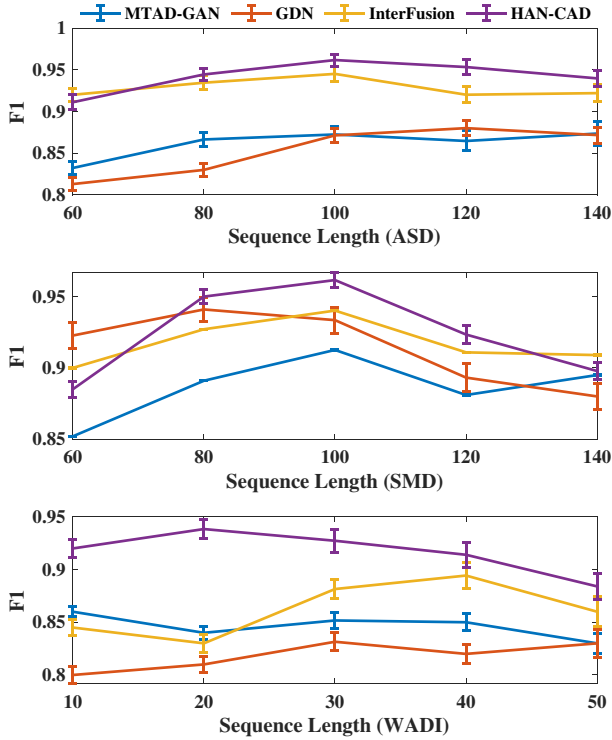


Figure 3 F1 scores with different sliding window lengths

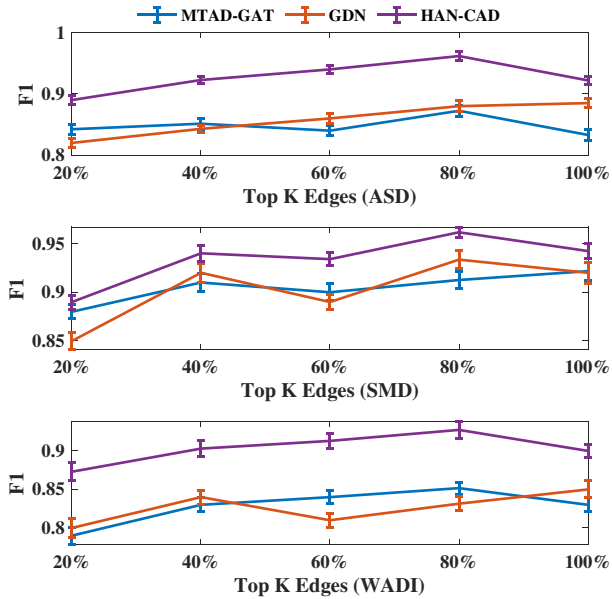


Figure 4 F1 scores with different number of edges

**Table 3** Ablation study

Dataset	Method	Precision	Recall	F1
ASD	HAN-CAD	96.82±0.76	92.29±0.61	96.18±0.72
	w/o feature learning	94.24±0.81	90.46±0.93	93.72±1.04
	w/o variable learning	92.83±0.72	91.83±0.59	90.61±0.62
	w/o sequence learning	95.03±0.81	90.02±0.94	92.03±0.96
SMD	HAN-CAD	97.26±0.42	95.81±0.91	96.16±0.53
	w/o feature learning	96.15±0.83	90.46±1.06	93.71±0.96
	w/o variable learning	92.17±0.47	90.52±0.87	90.72±0.66
	w/o sequence learning	94.95±0.81	92.02±1.01	92.84±0.89
WADI	HAN-CAD	96.87±0.61	86.95±1.73	92.73±1.08
	w/o feature learning	94.25±0.82	84.63±0.96	91.06±0.80
	w/o variable learning	92.15±0.52	83.01±1.14	89.71±0.86
	w/o sequence learning	93.63±1.04	85.25±0.61	91.24±0.92

#### 4.5 Ablation study

Finally, we investigate the impacts of the three components in our method on three datasets. In particular, the first model is trained without using the component of Bi-GRUs, i.e. w/o feature learning. The second model is trained without using the component of graph attention mechanism, i.e. w/o variable learning. The third model is trained without attention mechanism among variables, i.e. w/o sequence learning. Table 3 summarizes the results for the ablation study. We can see that all three components are important, as removing any one of them results in decreased performance in all three measures. Additionally, the comparison results indicate that the graph attention mechanism is the most critical component among the three, suggesting that capturing the relationships among variables is crucial for effectively detecting anomalies in MTS. Moreover, we report the training time for the baselines and the proposed method in Table 4. It is evident that as the amount of data increases, the training time also increases. Among the GNN-based methods, our approach performs the most efficiently, possibly due to the stable training achieved by integrating variable-level learning and sequence-level learning.

**Table 4** Comparison of training times in seconds

Method	ASD	SMD	WADI
LSTM-AE	1,063±12	3,039±31	5,721±38
MAD-GAN	1,692±17	28,931±24	49,723±51
MTAD-GAT	7,915±63	71,819±91	101,792±217
GDN	6,641±48	29,274±85	43,639±169
InterFusion	652±14	2,357±21	5,028±52
w/o feature learning	5,354±21	20,411±55	25,638±85
w/o variable learning	3,541±15	17,339±34	22,416±57
w/o sequence learning	5,601±34	22,119±52	28,379±76
HAN-CAD	6,367±18	23,469±41	34,849±64

Furthermore, we can observe that the training time decreases the most when variable learning is not employed in our method.

## 5 Conclusion

In this paper, we focus on detecting anomalies in multivariate time series. We argue that relationships among different variables are dynamic with regard to context sequences, and capturing the dynamic relationships can improve the accuracy of anomaly detection. Hence, we propose a novel Hierarchical Attention Network for Context Anomaly Detection in Multivariate Time Series. Two attention layers are hierarchically equipped into our model, in which one graph attention is introduced to obtain inter-variable relationships and the other attention is used to capture dynamic relationships. The effectiveness of our method is validated on three real-world datasets. And extensive comparison experiments demonstrate the superiority of our method.

**Author Contributions** Haicheng Tao and Jie cao designed the model, drafted the work and wrote the main manuscript text. Jiawei Miao, Haicheng Tao, Lin Zhao and Zhenyu Zhang analysed the data and carried out the experiment. Shuming Feng and Shu Wang contributed to the interpretation of the results. All authors provided critical feedback and helped shape the research, analysis and manuscript.

**Funding** This research is partially supported by the Key Program of National Natural Science Foundation of China under grant 92046026, in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under grant 21KJB520034.

**Data Availability** The “ASD” dataset that supports the findings of this study is publicly available in “<https://github.com/zhhlee/InterFusion>”. And the “SMD” and “WADI” datasets are available “[https://itrust.sutd.edu.sg/itrust-labs\\_datasets/](https://itrust.sutd.edu.sg/itrust-labs_datasets/)”.

## Declarations

**Competing interests** The authors declare no competing interests.

## References

1. Ahmad, S., Lavin, A., Purdy, S., Agha, Z.: Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* **262**, 134–147 (2017)
2. Blázquez-García, A., Conde, A., Mori, U., Lozano, J.A.: A review on outlier/anomaly detection in time series data. *ACM Comput. Surv. (CSUR)* **54**(3), 1–33 (2021)
3. Bontemps, L., Cao, V.L., McDermott, J., Le-Khac, N.-A.: Collective anomaly detection based on long short-term memory recurrent neural networks. In: *International Conference on Future Data and Security Engineering*, pp. 141–152. Springer (2016)
4. Canizo, M., Triguero, I., Conde, A., Onieva, E.: Multi-head cnn-rnn for multi-time series anomaly detection: An industrial case study. *Neurocomputing* **363**, 246–260 (2019)
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **41**(3), 1–58 (2009)
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. (2014). [arXiv:1412.3555](https://arxiv.org/abs/1412.3555)
7. Dani, M.-C., Jollois, F.-X., Nadif, M., Freixo, C.: Adaptive threshold for anomaly detection using time series segmentation. In: *International Conference on Neural Information Processing*, pp. 82–89. Springer (2015)
8. Deng, A., Hooi, B.: Graph neural network-based anomaly detection in multivariate time series. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 4027–4035. (2021)

9. Evangelou, M., Adams, N.M.: An anomaly detection framework for cyber-security data. *Comput. Secur.* **97**, 101941 (2020)
10. Gugulothu, N., Malhotra, P., Vig, L., Shroff, G.: Sparse neural networks for anomaly detection in high-dimensional time series. In: *AI4IoT Workshop in Conjunction with ICML, IJCAI and ECAI*, pp. 1551–3203. (2018)
11. Hu, M., Feng, X., Ji, Z., Yan, K., Zhou, S.: A novel computational approach for discord search with local recurrence rates in multivariate time series. *Inf. Sci.* **477**, 220–233 (2019)
12. Huang, L., Zhu, Y., Gao, Y., Liu, T., Chang, C., Liu, C., Tang, Y., Wang, C.-D.: Hybrid-order anomaly detection on attributed networks. *IEEE Trans. Knowl. Data Eng.* (2021) <https://doi.org/10.1109/TKDE.2021.3117842>
13. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 387–395. (2018)
14. Jones, M., Nikovski, D., Imamura, M., Hirata, T.: Exemplar learning for extremely efficient anomaly detection in real-valued time series. *Data Min. Knowl. Discov.* **30**(6), 1427–1454 (2016)
15. Kieu, T., Yang, B., Jensen, C.S.: Outlier detection for multidimensional time series using deep neural networks. In: *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, pp. 125–134. *IEEE* (2018)
16. Li, D., Chen, D., Jin, B., Shi, L., Goh, J., Ng, S.-K.: Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In: *International Conference on Artificial Neural Networks*, pp. 703–716. *Springer* (2019)
17. Li, Z., Zhao, Y., Han, J., Su, Y., Jiao, R., Wen, X., Pei, D.: 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 & Data Mining*, pp. 3220–3230. (2021)
18. Li, S., Pandey, A., Hooi, B., Faloutsos, C., Pileggi, L.: Dynamic graph-based anomaly detection in the electrical grid. *IEEE Trans. Power Syst.* **37**(5), 3408–3422 (2021)
19. Luo, X., Wu, J., Beheshti, A., Yang, J., Zhang, X., Wang, Y., Xue, S.: Comga: Community-aware attributed graph anomaly detection. In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, **33**(6), 2335–2349 (2022)
20. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L.: A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans. Knowl. Data Eng.* **1** (2021). <https://doi.org/10.1109/TKDE.2021.3118815>
21. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: Lstm-based encoder-decoder for multi-sensor anomaly detection. (2016). arXiv preprint [arXiv:1607.00148](https://arxiv.org/abs/1607.00148)
22. Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al.: Long short term memory networks for anomaly detection in time series. In: *Proceedings*, vol. 89, pp. 89–94. (2015)
23. Nelson, B.K.: Time series analysis using autoregressive integrated moving average (arima) models. *Acad. Emerg. Med.* **5**(7), 739–744 (1998)
24. Pawar, K., Attar, V.: Deep learning approaches for video-based anomalous activity detection. *World Wide Web* **22**(2), 571–601 (2019)
25. Sadouk, L.: Cnn approaches for time series classification. *Time Ser. Anal.-Data Methods Appl.* **5** 1–23 (2018)
26. Siami-Namini, S., Tavakoli, N., Namin, A.S.: The performance of lstm and bilstm in forecasting time series. In: *2019 IEEE International Conference on Big Data (Big Data)*, pp. 3285–3292. *IEEE* (2019)
27. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2828–2837. (2019)
28. Tuli, S., Casale, G., Jennings, N.R.: Tranad: Deep transformer networks for anomaly detection in multivariate time series data. (2022). [arXiv:2201.07284](https://arxiv.org/abs/2201.07284)
29. Visser, I.: Seven things to remember about hidden markov models: A tutorial on markovian models for time series. *J. Math. Psychol.* **55**(6), 403–415 (2011)
30. Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., Wang, F.-Y.: Generative adversarial networks: introduction and outlook. *IEEE/CAA J. Autom. Sin.* **4**(4), 588–598 (2017)
31. Wang, X., Lin, J., Patel, N., Braun, M.: Exact variable-length anomaly detection algorithm for univariate and multivariate time series. *Data Min. Knowl. Discov.* **32**(6), 1806–1844 (2018)
32. Xiang, H., Zhang, X.: Edge computing empowered anomaly detection framework with dynamic insertion and deletion schemes on data streams. *World Wide Web* **25**, 2163–2183 (2022)
33. Yin, C., Zhang, S., Wang, J., Xiong, N.N.: Anomaly detection based on convolutional recurrent autoencoder for iot time series. *IEEE Trans. Syst. Man Cybern. Syst.* **52**(1), 112–122 (2020)

34. Zeyer, A., Doetsch, P., Voigtlaender, P., Schlüter, R., Ney, H.: A comprehensive study of deep bidirectional lstm rnns for acoustic modeling in speech recognition. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2462–2466. IEEE (2017)
35. Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., Chawla, N.V.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 1409–1416. (2019)
36. Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., Zhang, Q.: Multivariate time-series anomaly detection via graph attention network. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 841–850. IEEE (2020)
37. Zhou, Y., Arghandeh, R., Spanos, C.J.: Online learning of contextual hidden markov models for temporal-spatial data analysis. In: 2016 IEEE 55th Conference on Decision and Control (CDC), pp. 6335–6341. IEEE (2016)
38. Zhou, B., Liu, S., Hooi, B., Cheng, X., Ye, J.: Beatgan: Anomalous rhythm detection using adversarially generated time series. In: IJCAI, pp. 4433–4439. (2019)
39. Zhou, K., Li, J., Xiao, Y., Yang, J., Cheng, J., Liu, W., Luo, W., Liu, J., Gao, S.: Memorizing structure-texture correspondence for image anomaly detection. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(6), 2335–2349 (2022)
40. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations (2018)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Haicheng Tao<sup>1</sup> · Jiawei Miao<sup>1</sup> · Lin Zhao<sup>2</sup> · Zhenyu Zhang<sup>2</sup> · Shuming Feng<sup>2</sup> ·  
Shu Wang<sup>3</sup> · Jie Cao<sup>4</sup>

Haicheng Tao  
haicheng.tao@gmail.com

Jiawei Miao  
peachblossom.miao@gmail.com

Lin Zhao  
982949302@qq.com

Zhenyu Zhang  
Jerry\_zzy@189.cn

Shuming Feng  
fengsm817@163.com

Shu Wang  
119639464@qq.com

- <sup>1</sup> College of Information Engineering, Nanjing University of Finance & Economic, 3 Wenyuan Road, Nanjing 210023, Jiangsu, China
- <sup>2</sup> Jiangsu Electric Power Information Technology Company, 22 West Beijing Road, Nanjing 210000, Jiangsu, China
- <sup>3</sup> State Grid Jiangsu Electric Power Company, Changzhou, 27 Juqian Street, Changzhou 213000, Jiangsu, China
- <sup>4</sup> School of Management, Hefei University of Technology, 193 Tunxi Road, Hefei 230009, Anhui, China