AAAD: Asynchronous Inter-variable Relationship-Aware Anomaly Detection for Multivariate Time Series

Hongyi Liu[1], Xiaosong Huang[2], Mengxi Jia[1], Lingzhe Zhang[1], Tong Jia[3,4], Zhonghai Wu[1,5], Ying Li[1,5*]
[1]*School of Software and Microelectronics*, *Peking University*, Beijing, China
[2]*School of Computer Science*, *Peking University*, Beijing, China
[3]*Institute for Artificial Intelligence*, *Peking University*, Beijing, China
[4]*National Key Laboratory of Data Space Technology and System*, Beijing, China
[5]*National Engineering Research Center for Software Engineering*, *Peking University*, Beijing, China
{hongyiliu,mxjia,jia.tong,wuzh,li.ying}@pku.edu.cn, {hxs,zhang.lingzhe}@std.pku.edu.cn

***Abstract*—Anomaly detection in multivariate time series (MTS) plays a crucial role in various domains, particularly in multimedia. While significant progress has been made in modeling normal data patterns and detecting anomalies based on deviations, existing methods face challenges in capturing complex inter-variable relationships, particularly in the presence of asynchronous dependencies. To address the challenges of detecting anomalies in MTS with complex inter-variable relationships, we propose an asynchronous inter-variable relationship-aware anomaly detection method. This approach simultaneously extracts both synchronous and asynchronous feature pairs between variables and leverages attention mechanisms to automatically learn unified inter-variable relationships across these dependencies. Additionally, we utilize a memory network to store and dynamically update the normal patterns of inter-variable relationships, computing anomaly scores based on deviations in these relationships. Extensive experiments demonstrate that our method outperforms existing baseline approaches, achieving an average F1 score of 97.05% across five benchmark datasets. Ablation studies further validate the effectiveness of each component of our method.**

***Index Terms*—multivariate time series, anomaly detection, inter-variable relationships, asynchronous**

I. Introduction

Multivariate Time Series (MTS) anomaly detection (AD) is a critical technique with widespread applications across various fields, particularly in multimedia. Examples include detecting factory equipment failures from audio data [1], identifying crimes from video data [2], monitoring failures in cyber-physical systems such as aerospace [3] and water treatment plants [4], and diagnosing faults in software systems (e.g., cloud services) using performance metrics [5]–[7]. In real-world scenarios, MTS data are typically collected from multiple channels of a monitored system, aiming to identify anomalies as data patterns deviating from the majority [8]. Due to the labor-intensive process of acquiring labeled data [9], most prior work adopts unsupervised settings, assuming that the majority of data are normal. These methods model the normal patterns in MTS and detect anomalies based on deviations from these patterns [10].

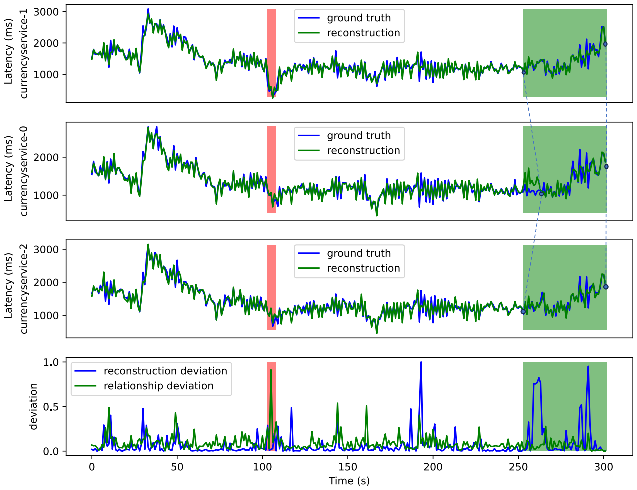

Fig. 1. Motivational example.

Existing approaches can be categorized into three types based on their normal pattern modeling strategies. *Density estimation-based methods* [11], [12] assume that normal data follow specific distributions and estimate parameters using historical data. *Clustering-based methods* [13], [14] rely on the assumption that normal data are dominant and form clusters. However, both methods are limited in real-world applications due to the difficulty in predefining distribution types or cluster numbers. The most prevalent approaches are *reconstruction-based methods* [3], [5], [15], [16], which learn the distribution of normal data through reconstructing unlabeled data. These methods reconstruct normal data well while failing to reconstruct anomalies, enabling anomaly detection based on reconstruction errors.

MTS normal pattern modeling comprises two aspects: temporal modeling and inter-variable relationship modeling. For

* Corresponding author: Ying Li. Hongyi Liu and Xiaosong Huang are equal contributed to this paper. This work was supported by Key R&D Project of Guangdong Province (No.2020B010164003).

temporal pattern modeling, methods typically use LSTMs [3] or Transformers [17] to capture temporal dependencies and VAEs [5] to model randomness. For *inter-variable relationships*, existing methods generally fall into two categories: *feature compression-based methods* [5], [17] implicitly extract inter-variable relationships by compressing multivariate features into higher-level representations, and *graph neural network (GNN)-based methods* [15], [16], [18], which treat variables as graph nodes and inter-variable relationships as edges, learning these relationships through self-supervised tasks like MTS reconstruction.

Despite advancements in reconstruction-based methods, these approaches have limitations. First, **detecting inter-variable relationship anomalies remains challenging when intra-variable values are normal but inter-variable relationships deviate significantly**. For example, in a cloud service dataset [7] (Figure 1), an anomaly occurs when 'currencyservice-1' experiences network packet loss (red region). While the intra-variable values of 'currencyservice-1' appear normal, its relationship with other variables changes, leading to missed anomaly detection due to small reconstruction errors. To address this, explicit modeling and storage of normal inter-variable relationships are required to assess deviations effectively.

Second, existing methods predominantly focus on modeling *synchronous relationships* [15], [16], [18], neglecting ***asynchronous relationships*, which are crucial for capturing precise normal inter-variable patterns and reducing false positives**. Asynchronous relationships involve lagged co-occurrence patterns between variables. For instance, in Figure 1, the normal green region shows significant delays in 'currencyservice-0' compared to others. Methods solely considering synchronous relationships might incorrectly label such regions as anomalies. Although a few methods [6], [19] model asynchronous relationships using phase synchronization, these approaches assume identical frequencies and similar waveforms, limiting their applicability to scenarios with frequency inconsistencies or negative correlations.

To address these challenges, we propose an asynchronous inter-variable relationship-aware anomaly detection method, as shown in Figure 2. We design an *extension embedding* approach that expands variable values into feature vectors, enabling the extraction of feature pairs for any variables at any time points within a window. This supports pointwise modeling of inter-variable relationships and the computation of anomaly scores. To *model asynchronous relationships*, we extract both synchronous and asynchronous feature pairs and leverage attention mechanisms to learn unified inter-variable relationships. A *memory network* is utilized to store and update normal inter-variable relationship patterns dynamically, calculating deviations as anomaly scores. Reconstruction errors are also incorporated into anomaly scores to ensure robust temporal anomaly detection. In summary, the contributions of this paper are as follows: (1) We design an *extension embedding* approach and a *memory network*-based learning framework to model inter-variable relationships pointwise and detect anomalies based on relationship deviations. (2) We propose an asynchronous inter-variable relationship-aware detection method, extracting both synchronous and asynchronous feature pairs and learning unified relationships through attention mechanisms. Combined with reconstruction errors, this enables accurate detection of both inter-variable and intra-variable anomalies. (3) Extensive experiments demonstrate the superiority of our method over state-of-the-art approaches on multiple open datasets. Ablation studies validate the effectiveness of each component. Additionally, we release our code and datasets to facilitate replication and future research [20].

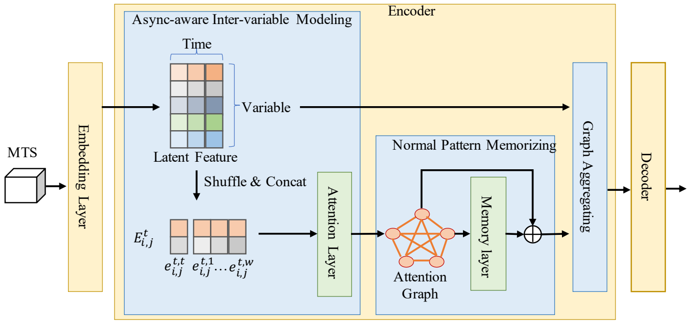

Fig. 2. The overview of the proposed method.

II. Approach

A. Problem Statement

Given the MTS data $X = \{x_1, x_2, ...x_T\}$ of length T, the MTS will be split into the sliding windows $S \in \mathbb{R}^{w \times d}$. w is the length of the sliding window, d means the number of variables, $\mathbb{R}$ represents the set of real numbers, and s_i^t represents the value of the i-th variable at timestamp t. The objective is to identify whether a given timestamp x_i is an anomaly or not based on a historical time window w.

B. Model Architecture

1) Embedding: To better capture the inter-variable relationships at each time point, and thus assess whether an anomaly has occurred, the proposed method differs significantly from existing approaches in its embedding strategy. Existing feature embedding methods for capturing inter-variable relationships typically fall into two categories. The first compresses the dimensionality of variables d into a specified dimension vector d' (we called it *mapping embedding*) [5], [17], i.e., $S \rightarrow S', S' \in \mathbb{R}^{w \times d'}$, implicitly extracting inter-variable relationships. However, this approach fails to model the normal patterns of inter-variable relationships and is, therefore, unsuitable for detecting anomalies in inter-variable relationships. The second approach represents each variable by its values within the time window (we called it *raw embedding*) [15], [16], i.e., $S' = S^T, S' \in \mathbb{R}^{d \times w}$, and then explicitly captures inter-variable relationships $\hat{E} \in \mathbb{R}^{d \times d}$ using a graph neural network. However, this method cannot finely model the inter-variable relationships at each timestamp, as it does not distinguish inter-variable relationships at different timestamps within the same time window, thereby limiting the accuracy of anomaly detection in these relationships.

To address the limitations of existing feature embedding methods, we propose an embedding method, we called it *extension embedding*. This approach expands the value of each variable at each timestamp into a specific feature dimension m, i.e., $S \rightarrow S', S' \in \mathbb{R}^{w \times d \times m}$. The embedding layer performs the following operation:

$$h_i^t = Exten(s_i^t) + PE(t), \tag{1}$$

where $h_i^t \in \mathbb{R}^m$ is the embedding vector of the i-th variable at timestamp t. $Exten$ is a 2D-CNN layer that projects s_i^t into an m-dimensional vector, while simultaneously capturing features in both the temporal dimension and the spatial dimension of the variables. PE denotes the positional embedding at position t within the sliding window, using the well-known sine-cosine positional embedding strategy introduced in [21].

2) Encoder: The encoder comprises an async-aware inter-variable modeling module, a normal pattern memorizing module, and a graph aggregating module.

Async-aware Inter-variable Modelling. Rather than merging synchronous and asynchronous inter-variable relationships after they are obtained, directly concatenating synchronous and asynchronous variable feature pairs and using an attention network to generate a unified inter-variable relationship is more beneficial for anomaly detection. The rationale is that concatenating raw synchronous and asynchronous inter-variable features led to less information loss compared to merging normalized (e.g. softmax operation) relationships, as detailed in subsection III-C. Therefore, the proposed method first constructs feature pairs for any two variables at any two timestamps within the window, enabling the extraction of all synchronous and asynchronous features:

$$e_{i,j}^{t,t'} = h_i^t || h_j^{t'}, \tag{2}$$

where $e_{ij}^{tt'} \in \mathbb{R}^{2m}$ is the feature pair formed by variable i at timestamp t and variable j at timestamp t', and $||$ denotes the concatenation operation. So the all inter-variable relationships (including synchronous and asynchronous) within the time window can be represented as $\hat{E} \in \mathbb{R}^{w \times w \times d \times d}$.

Then, we concatenate feature pairs containing asynchronous features between any two variables:

$$E_{i,j}^t = e_{i,j}^{t,t} || e_{i,j}^{t,1} || e_{i,j}^{t,2} || ... || e_{i,j}^{t,w}, \tag{3}$$

where $E_{i,j}^t \in \mathbb{R}^{2m*(w+1)}$ represents the collection of synchronous and asynchronous feature pairs between variable i and variable j at timestamp t. An example of a certain timestamp is shown in Fig. 2, where $d = 5, w = 3, t = 2, i = 1, j = 2$.

Subsequently, similar to GATv2 [22], a double-layer linear network is used to obtain the unified inter-variable attention:

$$\hat{E}_{i,j}^t = E_{i,j}^t W_0 W_a, \tag{4}$$

where $W_0 \in \mathbb{R}^{2m*(w+1) \times d_m}$ and $W_a \in \mathbb{R}^{d_m \times 1}$ are trainable model parameters, and $\hat{E}_{i,j}^t \in \mathbb{R}$ represents the unified relationship between variable i and variable j at timestamp t.

Normal Pattern Memorizing. To model the normal patterns of relationships between variables and compute the relational deviations for new samples, a memory network is employed to store and update the normal patterns, following the approaches in the existing method [23].

$$\begin{aligned} g =& \sigma(W_1 C + W_2 \hat{E}^t), \\ C =& (1 - g)C + g\hat{E}^t, \end{aligned} \tag{5}$$

where $C \in \mathbb{R}^{d \times d}$ is the memory cell storing historical inter-variable relationship information, initialized as a random vector. $\hat{E}^t \in \mathbb{R}^{d \times d}$ is the the unified relationship at timestamp t. $W_1 \in \mathbb{R}^{d \times d}$ and $W_2 \in \mathbb{R}^{d \times d}$ are trainable model parameters. $g \in \mathbb{R}^{d \times d}$ controls the update rate of the memory cell. The memory cell continually updates to capture the normal patterns of inter-variable relationships. σ is the activation function, here using ReLU. To enhance the information content of features used during reconstruction, the historical inter-variable relationship information is integrated into the hidden layer representation:

$$\hat{E}^{t'} = \hat{E}^t + C. \tag{6}$$

Here, an addition operation is used to integrate both historical and current relationship features. $\hat{E}^{t'} \in \mathbb{R}^{d \times d}$ represents the integrated inter-variable relationship at timestamp t.

Graph aggregating. The feature vectors after graph aggregation are obtained using the inter-variable relationships $\hat{E}_{i,j}^{t'}$ and the embedding vectors of all variables $H^t = \{h_1^t, h_2^t, \ldots, h_d^t\} \in \mathbb{R}^{d \times m}$:

$$\begin{aligned} \alpha_{i,j}^t =& \frac{exp(LeakyReLU(\hat{E}_{i,j}^{t'}))}{\sum_v^d exp(LeakyReLU(\hat{E}_{i,v}^{t'}))}, \\ h_i^{t'} =& \sigma(\sum_j^d \alpha_{i,j}^t h_j^t), \end{aligned} \tag{7}$$

where $\alpha_{i,j}^t \in [0, 1]$ represents the attention between variable i and variable j at timestamp t. The vector $h_i^{t'} \in \mathbb{R}^m$ is the encoded feature of variable i at timestamp t.

3) Decoder: For anomaly detection tasks, the decoder of the reconstruction model typically uses a simple network to prevent it from being too powerful. This ensures that the decoder does not overfit to any input data, thereby ignoring the encoder's extraction of normal patterns, which would reduce the model's ability to distinguish anomalies. Therefore, a simple linear transformation is applied to the encoded vectors to obtain the reconstructed values:

$$\hat{s}_i^t = h_i^{t'} W_d, \tag{8}$$

where $\hat{s}_i^t \in \mathbb{R}$ is the reconstructed value of variable i at timestamp t. The matrix $W_d \in \mathbb{R}^{m \times 1}$ represents the trainable model parameters.

C. Training And Inference

1) Training Phase: During the training phase, for a specific sliding window of training data, the reconstruction loss is defined as:

$$L = \sum_{t=1}^{w} MSE(s^t, \hat{s}^t), \tag{9}$$

where MSE is used to compute the mean squared error between the actual values $s^t \in \mathbb{R}^d$ at timestamp t and the corresponding reconstructed values $\widehat{s^t} \in \mathbb{R}^d$.

2) Inference Phase: During the inference phase, the anomaly score for each time point within the entire time window is calculated in one pass. The reconstruction error and the inter-variable relationship deviation are combined to determine anomalies. The anomaly score for timestamp t is computed as follows:

$$score(s^t) = MSE(s^t, \widehat{s^t}) \times \frac{exp(MSE(\widehat{E^t}, C)/\tau)}{\sum_{l=0}^{w} exp(MSE((\widehat{E^l}, C)/\tau)}, \tag{10}$$

where $MSE(\widehat{E^t}, C)$ calculates the inter-variable relationship deviation at timestamp t using mean squared error. The matrix $C \in \mathbb{R}^{d \times d}$ represents the memory cell state capturing the normal patterns of inter-variable relationships. The softmax function with temperature τ is applied to the inter-variable relationship deviations. The anomaly score is obtained by multiplying the reconstruction error with the inter-variable relationship deviation. Finally, a threshold is set, which is a predefined hyperparameter based on the anomaly rate in the training dataset, and samples exceeding this threshold are considered anomalous.

III. Evaluation

In this section, we present a comprehensive evaluation of our proposed approach. We assess its performance across various aspects, including the anomaly detection performance, the impact of individual components, and the feasibility study.

A. Experimental Setup

Datasets. We employ five commonly used datasets for time series anomaly detection: SMD [5], SMAP [3], PSM [6], SWaT [4], and AIOps22 [7]. *Baseline Methods*. We compared our proposed method with various types of baseline approaches, including statistical methods that do not perform spatiotemporal modeling (LOF [24], OC-SVM [25]), methods that perform temporal modeling (THOC [14], OmniAnomaly (Omni) [5], Anomaly Transformer (AT) [17], DCdecoder [27], MAN-QSM [26], MEMTO [23], CTAD [29], GPT4TS [28], AnomalyLLM [30]), and methods that incorporate spatiotemporal modeling (MTAD-GAT [16], STGAT [15], RSCoders [6]). **Evaluation Metrics**. To evaluate the anomaly detection performance, we used standard metrics, including precision (P), recall (R), and F1-score. Additionally, we adopt a widely-used point adjustment strategy used in prior works [5], [16], [17], [27]. Under this strategy, if any anomaly is detected within an anomaly segment, all points within that segment are considered to have been correctly identified as anomalies.

B. Main Results

To evaluate the performance of our proposed method for anomaly detection, we compared it to several baseline methods, as shown in Table I. It can be observed that the proposed method achieves the best average performance across five benchmark datasets, with an improvement of 2 percentage points in the average F1 score. Although our method performs slightly worse than AnomalyLLM on the SMD and PSM datasets, it offers significant advantages by eliminating the need for fine-tuning large language models (LLMs). Moreover, compared to LLM-based methods, our approach demonstrates higher inference efficiency, making it more suitable for real-time anomaly detection scenarios.

C. Ablation Study

Effectiveness of Anomaly Detection Criteria. To better understand the impact of these criteria on anomaly detection, we conducted an ablation study, as shown in Fig. 3(a). The results indicate that relying solely on either numeric deviation or relationship deviation does not yield satisfactory performance. This is because a single criterion struggles to comprehensively assess both temporal anomalies and inter-variable relationship anomalies. **Effectiveness of Asynchronous Modeling**. Table II shows that our method significantly outperforms the variant without asynchronous modeling (*w/o async*), confirming the effectiveness of asynchronous modeling. To further explore the rationale behind our proposed method, we compared different strategies for integrating synchronous and asynchronous inter-variable relationships, as shown in Figure 3 (b). The *mean* and *max* variants combine synchronous and asynchronous relationships by taking their mean and maximum, respectively. The *cross_attn* variant uses synchronous relationships as the query and asynchronous relationships as the key and value in a cross-attention mechanism. Finally, our proposed method, *linear* (Our), concatenates the inter-variable features and uses a linear network to learn a unified representation of synchronous and asynchronous relationships. The results indicate that our method's fusion strategy is more effective for anomaly detection. This may be because concatenating the raw features of synchronous and asynchronous relationships preserves more information compared to combining normalized features (e.g., post-softmax), resulting in less information loss. **Effectiveness of Memory Network**. Table II demonstrates that our proposed method significantly outperforms the variant without the memory network (*w/o mem*), confirming the effectiveness of the memory network design. Moreover, our method clearly surpasses the variants that only use the memory network (*w/o async*) or only model asynchronous inter-variable relationships (*w/o mem*), indicating that both components are complementary and contribute to the improvement in anomaly detection.

TABLE I
MAIN RESULT. THE BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED WITH BOLD FONT AND UNDERLINING FOR CLARITY.

	SMAP			SWaT			SMD			PSM			AIOps22			Average
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	F1
LOF [24] [SIGMOD-00]	58.93	56.33	57.61	72.15	65.43	68.62	56.34	39.86	46.68	57.89	90.49	70.61	35.15	98.99	51.88	59.08
OC-SVM [25] [Neural Comput-01]	53.85	59.07	56.34	45.39	49.22	47.23	44.34	76.72	56.19	62.75	80.89	70.67	42.62	98.66	59.52	57.99
THOC [14] [NeurIPS-20]	92.06	89.34	90.68	83.94	86.36	85.13	79.76	90.95	84.99	88.14	90.99	89.54	88.83	87.12	87.96	87.66
Omni [5] [KDD-19]	92.49	81.99	86.92	81.42	84.31	82.83	83.68	86.82	85.22	88.39	74.46	80.83	92.09	57.24	70.59	81.27
AT [17] [ICLR-22]	94.13	99.41	96.69	91.55	96.73	94.07	89.41	95.45	92.33	96.91	98.91	97.89	86.12	93.31	89.57	94.11
MAN-QSM [26] [ICME-23]	93.63	99.78	96.61	91.06	97.09	93.97	89.51	96.51	92.87	97.44	98.81	98.12	85.66	93.65	89.47	94.21
Dcdetector [27] [KDD-23]	94.25	98.26	96.21	92.94	99.76	96.23	83.59	91.11	87.18	97.14	98.74	97.94	90.19	88.28	89.24	93.36
MEMTO [23] [NeurIPS-23]	93.81	99.61	96.62	94.18	97.54	95.83	89.11	98.41	93.51	97.51	99.21	98.31	85.82	93.49	89.49	94.75
GPT4TS [28] [NeurIPS-23]	88.78	64.72	74.86	90.13	95.61	92.79	89.61	81.13	85.16	97.39	94.13	95.73	84.78	92.22	88.34	87.37
CTAD [29] [TKDE 24]	95.77	96.51	96.13	**97.11**	88.11	92.41	84.94	89.48	87.15	83.01	97.21	89.51	91.65	86.71	89.11	90.86
AnomalyLLM [30] [IJCAI-24]	93.41	**99.81**	96.51	90.84	97.11	93.87	**93.71**	**97.91**	**95.81**	**99.61**	**99.81**	**99.71**	85.46	93.68	89.38	95.05
MTAD-GAT [16] [ICDM-20]	79.14	99.64	88.21	96.81	71.63	82.34	74.89	88.44	75.22	73.62	89.92	80.96	85.29	87.64	86.44	82.63
STGAT [15] [ICASSP-22]	78.01	99.66	87.51	96.51	84.12	89.89	74.09	95.69	83.51	80.31	99.15	88.74	84.07	87.65	85.82	87.09
RSCoders [6] [KDD-21]	94.89	88.67	91.67	95.21	75.21	84.03	90.13	85.15	87.56	90.22	96.03	93.03	83.88	92.05	87.77	88.81
AAAD (Our)	**99.24**	94.75	**96.95**	93.95	**99.82**	**96.81**	92.38	97.51	94.87	98.55	99.33	98.94	**95.75**	**99.66**	**97.67**	**97.05**

TABLE II
THE ABLATION STUDY RESULTS OF THE MODULES OF THE PROPOSED METHOD.

	Anomaly Criterion		Modules		F1-score					F1-score
	Numeric Deviation	*Relationship Deviation*	*Async Modeling*	*Memorizing*	SMD	SMAP	PSM	SWaT	AIOps22	Average
AAAD (Our)	✓	✓	✓	✓	**94.87**	**96.95**	**98.94**	**96.81**	**97.67**	**97.05**
w/o mem	✓	-	✓	-	89.01	96.57	98.49	96.52	59.01	87.92
w/o async	✓	✓	-	✓	89.32	96.61	98.38	96.51	97.24	95.61
w/o mem & async	✓	-	-	-	77.67	69.96	92.25	79.31	58.99	75.64

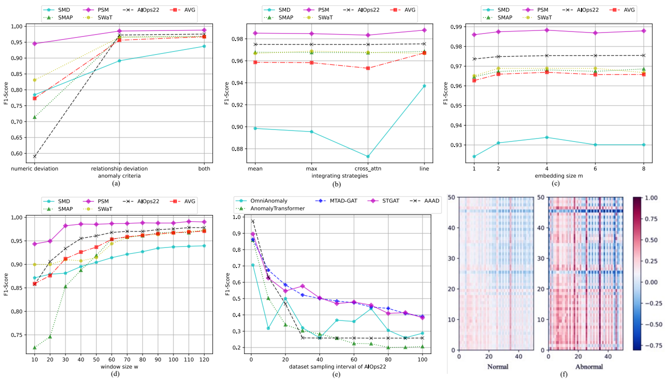

Fig. 3. The experiment results of the feasibility study.

D. Feasibility Study

Embedding Size (m). We systematically varied m and observed its impact on the F1 score for anomaly detection, as shown in Fig. 3(c). As m increases, the F1 score of our method initially rises and then falls. This is likely because, initially, increasing the dimensionality enriches the representation of the features, leading to improved F1 scores. However, when the dimensionality becomes too large, the feature space becomes harder to optimize and more prone to local minima, limiting the model's performance. To balance storage costs and anomaly detection performance, we chose $m = 4$. **Time Window Size** w. We systematically varied w and observed its impact on the anomaly detection F1 score, as shown in Fig. 3(d). As w increases, the F1 score of the proposed method initially increases and then stabilises. This trend can be attributed to the fact that as the time window grows, the model captures temporal dependencies and inter-variable relationships more effectively, leading to improved anomaly detection performance. To achieve the best balance between effectiveness and computational complexity, we select $w = 100$. **Dataset Sampling Interval**. Let the original sampling interval of the dataset be 1. We systematically varied the sampling interval. Fig. 3(e) shows that, the F1 score of all methods decreases across all datasets as the sampling interval increases. The primary reason is that larger sampling intervals result in fewer data points within the anomaly duration, making anomaly detection more challenging. Another thing is as the sampling interval increases, the performance of the proposed method falls behind that of the baseline methods. This occurs because the proposed method of detecting anomalies primarily relies on relational anomalies between variables. As the sampling interval increases, the granularity of inter-variable relationships becomes coarser, making it harder to precisely characterize these relationships. For example, asynchronous relationships degrade into synchronous ones, and the fluctuations in synchronous relationships become smoothed, both of which restrict the performance of the proposed method. Therefore, to achieve optimal anomaly detection performance, we recommend collecting data at the finest granularity possible during the data acquisition process. **Result Visualization**. To further validate the effectiveness of our proposed method, we visualized the inter-variable relationships for normal and anomalous samples in SWaT dataset, as shown in Fig. 3(f). The significant differences between the inter-variable relationships of normal (i.e., the memory cells in our method) and anomalous samples illustrate why deviations in these relationships can be used for anomaly detection.

IV. Conclusion

To address the challenges of detecting anomalies in MTS with complex inter-variable relationships, this paper proposes an innovative asynchronous inter-variable relationship-aware anomaly detection method tailored for these environments. The method simultaneously captures both synchronous and asynchronous features of inter-variable relationships and employs an attention mechanism to automatically learn unified representations of these relationships. This approach enables more precise modeling of inter-variable relationships. Extensive experiments demonstrate that the proposed method outperforms existing baselines, and that each component of the method contributes effectively to the overall performance improvement.

References

[1] Harsh Purohit, Ryo Tanabe, Kenji Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi, "Mimii dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," arXiv preprint arXiv:1909.09347, 2019.

[2] Ruoyu Xue, Jingyuan Chen, and Yajun Fang, "Real-time anomaly detection and feature analysis based on time series for surveillance video," in 2020 5th International Conference on Universal Village (UV). IEEE, 2020, pp. 1–7.

[3] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 387–395.

[4] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur, "A dataset to support research in the design of secure water treatment systems," in Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11. Springer, 2017, pp. 88–99.

[5] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei, "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, 2019, pp. 2828–2837.

[6] Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki, "Practical approach to asynchronous multivariate time series anomaly detection and localization," in Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, 2021, pp. 2485–2494.

[7] "Aiops challenge 2022," https://competition.aiops-challenge.com/home/competition/1496398526429724760.

[8] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano, "A review on outlier/anomaly detection in time series data," ACM computing surveys (CSUR), vol. 54, no. 3, pp. 1–33, 2021.

[9] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller, "A unifying review of deep and shallow anomaly detection," Proceedings of the IEEE, vol. 109, no. 5, pp. 756–795, 2021.

[10] Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi, "Deep learning for time series anomaly detection: A survey," ACM Computing Surveys, vol. 57, no. 1, pp. 1–42, 2024.

[11] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in International conference on learning representations, 2018.

[12] Takehisa Yairi, Naoya Takeishi, Tetsuo Oda, Yuta Nakajima, Naoki Nishimura, and Noboru Takata, "A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction," IEEE Transactions on Aerospace and Electronic Systems, vol. 53, no. 3, pp. 1384–1401, 2017.

[13] David MJ Tax and Robert PW Duin, "Support vector data description," Machine learning, vol. 54, pp. 45–66, 2004.

[14] Lifeng Shen, Zhuocong Li, and James Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," Advances in Neural Information Processing Systems, vol. 33, pp. 13016–13026, 2020.

[15] Jun Zhan, Siqi Wang, Xiandong Ma, Chengkun Wu, Canqun Yang, Detian Zeng, and Shilin Wang, "Stgat-mad: Spatial-temporal graph attention network for multivariate time series anomaly detection," in ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022, pp. 3568–3572.

[16] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang, "Multivariate time-series anomaly detection via graph attention network," in 2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020, pp. 841–850.

[17] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," arXiv preprint arXiv:2110.02642, 2021.

[18] Weiqi Zhang, Chen Zhang, and Fugee Tsung, "Grelen: Multivariate time series anomaly detection from the perspective of graph relational learning.," in IJCAI, 2022, pp. 2390–2397.

[19] Yongqian Sun, Minghan Liang, Zeyu Che, Dongwen Li, Tinghua Zheng, Shenglin Zhang, Pengtian Zhu, Yuzhi Zhang, and Dan Pei, "Efficient multivariate time series anomaly detection through transfer learning for large-scale web services," in 2023 IEEE International Conference on Web Services (ICWS). IEEE, 2023, pp. 145–151.

[20] "Aaad," https://github.com/lhysgithub/AAAD.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[22] Shaked Brody, Uri Alon, and Eran Yahav, "How attentive are graph attention networks?," in International Conference on Learning Representations.

[23] Junho Song, Keonwoo Kim, Jeonglyul Oh, and Sungzoon Cho, "Memto: Memory-guided transformer for multivariate time series anomaly detection," Advances in Neural Information Processing Systems, vol. 36, pp. 57947–57963, 2023.

[24] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander, "Lof: identifying density-based local outliers," in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104.

[25] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson, "Estimating the support of a high-dimensional distribution," Neural computation, vol. 13, no. 7, pp. 1443–1471, 2001.

[26] Jie Zhong, Enguang Zuo, Chen Chen, Cheng Chen, Junyi Yan, Tianle Li, and Xiaoyi Lv, "A masked attention network with query sparsity measurement for time series anomaly detection," in 2023 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2023, pp. 2741–2746.

[27] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun, "Dcdetector: Dual attention contrastive representation learning for time series anomaly detection," in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 3033–3045.

[28] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al., "One fits all: Power general time series analysis by pretrained lm," Advances in neural information processing systems, vol. 36, pp. 43322–43355, 2023.

[29] HyunGi Kim, Siwon Kim, Seonwoo Min, and Byunghan Lee, "Contrastive time-series anomaly detection," IEEE Transactions on Knowledge and Data Engineering, 2023.

[30] Chen Liu, Shibo He, Qihang Zhou, Shizhong Li, and Wenchao Meng, "Large language model guided knowledge distillation for time series anomaly detection," arXiv preprint arXiv:2401.15123, 2024.