Physical Process Guided Graph Neural Networks for Anomaly Detection in CPSs

Mengzhou Gao[1] , **Zehao Liu**[1] , **Yifan Lu**[1] and **Pengfei Jiao**[1,2]
[1]Hangzhou Dianzi University
[2]Data Security Governance Zhejiang Engineering Research Center
{mzgao, zehaoliu, lyfcan, pjiao}@hdu.edu.cn

Abstract

Given to abundant time series derived from Cyber Physical Systems (CPSs), various data-driven anomaly detection methods are proposed. Most existing methods are focused on capturing complex (e.g. temporal and inter-variable) relationships among sensors and actuators, while ignoring the underlying physical process which is paramount in CPSs and is essential to reveal changes and anomalies. In this work, we propose the physical process guided graph neural networks for modeling the hidden dynamics in CPSs and predicting time series for anomaly detection. In detail, we design an end-to-end auto-encoder framework for revealing the complex and dynamic changes among the time series. Instead of traditional sensors-actuator graph, system state graph is constructed with learnable and changable structure, which is more effective in model learning and time series prediction. Experiments on real-world datasets show that the proposed model detects anomalies more accurately than baseline approaches.

1 Introduction

Cyber Physical Systems (CPSs) consist of computing and physical processes which are integrated via sensors, actuators and communications links to enable the control of the underlying physical processes [Umer *et al.*, 2022]. With the abundant data acquisition from sensors, controllers and actuators, numerous CPS time-series data are collected and can be utilized to extract potential information or signals of anomaly. Anomaly detection, the process of detecting unusual patterns or unexpected behaviors, is paramount to ensure the stable and reliable operation of CPSs.

Anomaly detection methods have attracted substantial interest and can be broadly classified into two categories. On one hand, physics-based anomaly detection methods are developed which rely on precise first-principle models or extensive expert knowledge[Giraldo *et al.*, 2018], however, become challenging due to the increasing complexity of CPSs. On the other hand, data-driven methods have garnered significant attention due to the benefit of abundant data acquisition from sensors, controllers, and actuators. In specific, data-driven models are trained in an unsupervised way to either reconstruct the original time series or predict the subsequent one.

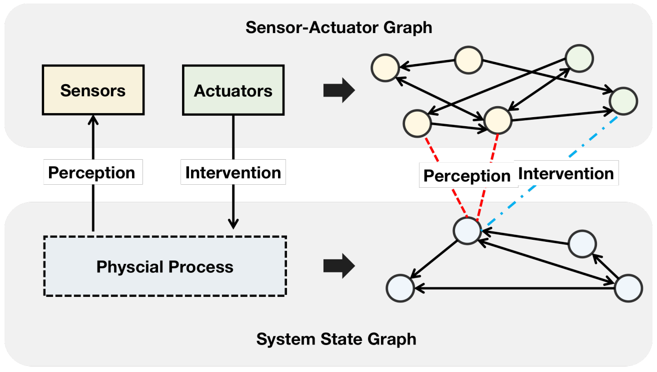

Figure 1: Cyber-Physical Systems. Actuators apply interventions to the underlying physical process of the CPS, and sensor values are the perception of the physical process. The association between the two at the data level can be transformed into the relationships between hidden system state nodes.

In the category of data-driven anomaly detection, methods based on machine learning (ML) and deep learning (DL) have demonstrated effective. Confronting the complex relationships within CPS, conventional ML-based methods encounter difficulties in effectively modeling intricate aspects like spatial-temporal correlations.[Inoue *et al.*, 2017] Recently, DL-based methods exhibit efficiency in handling high-dimensional multivariate time-series data, such as DAGMM[Zong *et al.*, 2018], MAD-GAN[Li *et al.*, 2019], USAD[Audibert *et al.*, 2020], NSIBF[Feng and Tian, 2021].

However, ML-based and DL-based methods fail to capture the inherent pairwise interdependence among variables, consequently reducing their capacity to detect intricate anomaly events. Graph neural networks (GNNs), which can explicitly model temporal and inter-variable relationships, possess the ability of capturing high-order complex interdependencies [Jiao *et al.*, 2021] and have shown great promise in CPS anomaly detection, such as GDN[Deng and Hooi, 2021], FuSAGNet[Han and Woo, 2022], MAD-SGCN[Qi *et al.*, 2022], Grelen[Zhang *et al.*, 2022a] and GANF[Dai and Chen, 2022]). However, most GNN-based anomaly detection methods consider every sensor and actuator as an individual node, forming a sensor-actuator graph in where node features are derived from multivariate time series data.

In fact, the physical process, serving as the core of CPSs, is essential to reveal changes and anomalies, but has not received sufficient attention. As depicted in Fig.1, the underlying physical process is perceived and intervened by sensors and actuators, respectively. In other words, the time series of sensors and actuators are derived and affected by the underlying physical process. Furthermore, system states inherently represent the physical process and must adhere to specific system dynamics, regardless of the occurrence of anomalies. Consequently, the evolution of system states according to these dynamics becomes a valuable aspect for prediction-based anomaly detection.

How to use GNNs model the system states and their potential correlations within the physical process based on time series from sensors and actuators? There are two challenges need to be solved: 1) system states are often unobservable and unavailable variables, how to derive them from the observable and available time series from sensors and actuators? 2) system states will evolve following specific system dynamics, how to represent the evolution of system states by graphs? Inspired by system identification, we proposed a novel system state graph to estimate the hidden system state and system dynamics within the underlying physical process. Firstly, the hidden system state node embeddings are generated and estimated using convolution neural networks (CNNs), which employ different learnable parameters for time series of sensors and actuators considering their diverse effects on the physical process. Secondly, motivated by [Zhang *et al.*, 2022b], we regard the system dynamics as a parameterized message passing process on a given graph structure. We employ Graph Convolutional Networks (GCNs) to illustrate the progression of system states. The outcomes of the message passing process are treated as estimates for predicting the system states in the subsequent step.

In this work, we propose our novel Physical Process Guided Graph Neural Networks for Anomaly Detection (PhyGAD) in CPSs. To summarize, the main contributions of our work are:

- We predict future system states innovatively through a message-passing mechanism, and encapsulate specific system dynamics within a learnable and changable graph structure influenced by both sensors and actuators.

- We introduce PhyGAD to reveal complex and dynamic changes among the time series in an end-to-end auto-encoder framework, in which a novel system state graph is utilized to depict the underlying physical process.

- We demonstrate through experiments that our proposed method outperforms the state-of-the-art methods on anomaly detection for CPS, and show good prediction performance even when trained on times series data from normal but unstable system operation period.

2 Related Work

2.1 Anomaly Detection in CPSs

Initially, physics-based models are formulated by extrapolating from fundamental physical equations (e.g. Newton's laws, fluid dynamics, or electromagnetic principles). Alternatively, they can be obtained through observations employing a methodology known as system identification (e.g. ARMAX, linear state-space models).[Urbina *et al.*, 2016] However, the increasing complexity of CPSs presents a challenge in implementing these approaches for anomaly detection, as their dependence on accurate first-principle models or substantial expert knowledge becomes impractical within this context.

Subsequently, machine learning methods that operate autonomously of domain-specific knowledge and rely solely on data have been created; however, they often struggle to model the intricate relationships inherent in CPSs, such as spatial-temporal correlations. [Luo *et al.*, 2021] To this end, deep learning-based methods have been proposed, such as DAGMM[Zong *et al.*, 2018], MAD-GAN[Li *et al.*, 2019], USAD[Audibert *et al.*, 2020], and etc.. While the proposed deep learning frameworks can efficiently scale through high-dimensional multivariate time-series data, they fail to explicitly model the inherent pairwise interdependence among variables, thus diminishing the ability to detect complex anomaly events. [Zheng *et al.*, 2023]

2.2 GNN-based Anomaly detection in CPSs

GNNs, which can explicitly model inter-temporal and inter-variable relationships, and have shown great promise in anomaly detection [Jin *et al.*, 2023]. However, a predefined graph for CPSs is frequently unavailable. how to design an appropriate graph structure to effectively model the correlations among time series gradually becomes a pivotal challenge. Pioneering efforts have suggested learning and constructing pairwise correlation relationships among variable pairs through either a straightforward graph learning layer (e.g. GDN[Deng and Hooi, 2021], [Han and Woo, 2022; Qi *et al.*, 2022]) or a probabilistic framework (e.g. Grelen[Zhang *et al.*, 2022a] and GANF[Dai and Chen, 2022]).

These models have achieved good performance for anomaly detection, whereas another important system characteristic, called the dynamic property, has not been taken into account. In fact, CPSs are essentially dynamical systems due to their loops and feedback, highlighting the significance of dynamic graph structures for precisely modeling system behaviors. [Wu *et al.*, 2023] Although research, such as ESG ([Ye *et al.*, 2022]), is underway to capture evolving graph structures in multivariate time series, progress in this area remains relatively restricted.

3 Proposed Framework

3.1 Problem Statement

In this paper, sensor values y_t are modeled as a function $h(\cdot)$ of the system state x_t, with the system state itself being modeled as a function $f(\cdot)$ of the previous instant's system state x_{t+1} [Jagannathan, 2001]. In CPS, the actuator state values u_t serve as the control inputs of the system, essentially representing feedback control of the system state. This determines

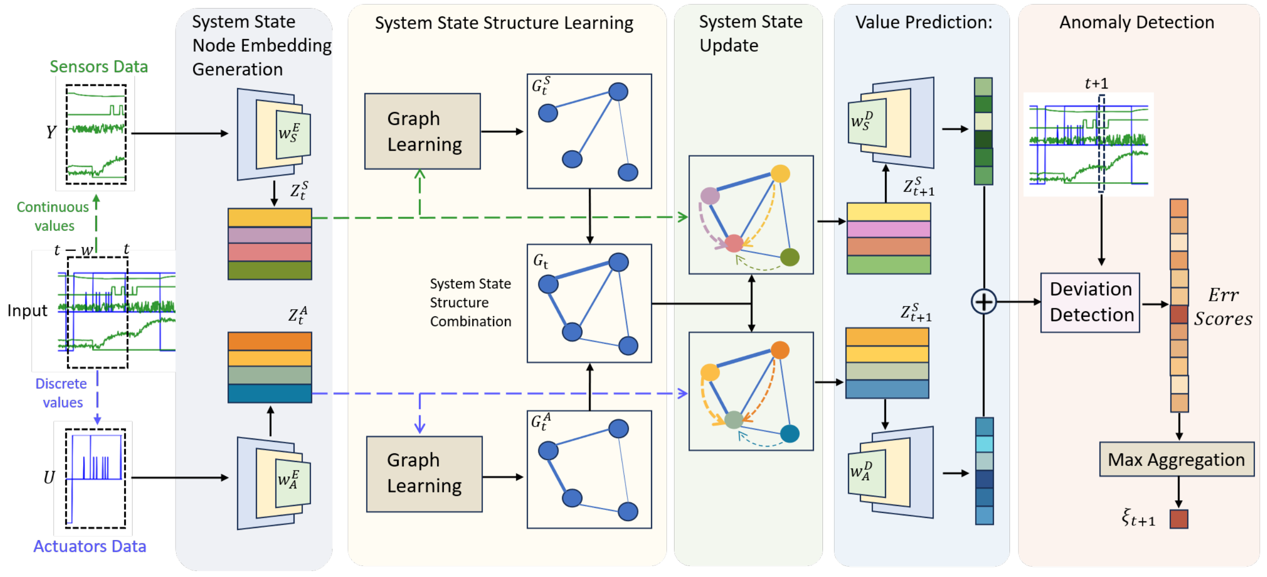

Figure 2: Overview of our proposed framework.The sensor measurements Y and actuator state values U are inputted respectively to generate the corresponding system state embeddings. These embeddings then enter the system state structure learning stage to learn the dynamic system state structure. During the system state update stage, the evolution of the system state is modeled. The updated system state is then decoded for prediction. Finally, anomaly detection is performed based on the deviation of the predicted values.

that the actuator state values are a continuous function $g(\cdot)$ of the system state [Haddad and Lee, 2022]. Thus, in this paper, the following general CPSs model is considered:

$$\begin{cases} x_{t+1} = f_t(x_t) \\ y_t = h(x_t) \\ u_t = g(x_t) \end{cases} \tag{1}$$

where $x_t \in \mathbb{R}^n$ denotes the hidden system state, $u_t \in \mathbb{R}^m$ denotes the control input that is executed by actuators, $y_t \in \mathbb{R}^p$ is the measured output. The mappings $f_t(\cdot)$, $h(\cdot)$ and $g(\cdot)$ are nonlinear. Of note is that $f_t(\cdot)$ represents the dynamics that map the state of the system forward in time and are considered time-varying.

Given to the above CPSs model, multivariate time series data of N sensors and M actuators are collected on T time ticks, which can be denoted as $S = [S_1, S_2, \ldots, S_T]$, $S_t \in \mathbb{R}^N$ and $A = [A_1, A_2, \ldots, A_T]$, $A_t \in \mathbb{R}^M$. In each time tick t, $S_t = [s_1^t; s_2^t; \ldots; s_N^t]$ and $A_t = [a_1^t; a_2^t; \ldots; a_M^t]$. Considering the large amount of data, the times series are often processed with a sliding window of size W as the input $Y_t \in \mathbb{R}^{N\times W}$ and $U_t \in \mathbb{R}^{M\times W}$.

Relying on the inputs Y_t and U_t, our goal is to identify the physical process underneath the data, including the hidden states $\hat{x}_t$ and dynamics $\hat{f}_t(\cdot)$ of CPSs, and its mapping relationships (e.g. $\ell(\cdot), \hat{h}(\cdot), \hat{g}(\cdot)$) with time series data. Thus, the hidden system state $\hat{x}_t$ can evolved into the next state $\hat{x}_{t+1}$ which is then converted to get the predictions of sensor and actuator data as $\hat{y}_{t+1}$ and $\hat{u}_{t+1}$. Finally, anomaly detection could be accomplished based on predictions.

$$\begin{cases} \hat{x}_t(\theta_t) = \ell_\theta(\theta_t) & \text{(2a)} \\ \hat{x}_{t+1}(\theta_t) = \hat{f}_t(\hat{x}_t(\theta_t)) & \text{(2b)} \\ \hat{y}_{t+1} = \hat{h}(\hat{x}_{t+1}(Y_t)) & \text{(2c)} \\ \hat{u}_{t+1} = \hat{g}(\hat{x}_{t+1}(U_t)) & \text{(2d)} \end{cases}$$

It's worth noting that the mappings between hidden states $\hat{x}_t$ and time series data may vary based on different types of time series data. As a result, $\ell(\cdot)$ is denoted as ℓ_θ depending on the input θ. Also, the system dynamics might exhibit temporal variability and are represented as $\hat{f}_t(\cdot)$.

3.2 Overview

PhyGAD depicts the mapping relationships between hidden states and time series data. The system state graph $\mathcal{G} = \{\mathcal{V}, G, Z\}$ can be built, where $\mathcal{V}$ is the set of system state nodes, G is the relation matrix and Z is the feature matrix of nodes. Furthermore, the prediction is realized through the evolution of system states in accordance with the system dynamics through the use of Graph Convolutional Networks (GCNs). In general, PhyGAD consists of the following five components:

1. System State Node Embedding Generation: employs embeddings as the features of system state nodes, obtained by encoding the time series of sensors and actuators independently.
2. System State Structure Learning: merges the two learned graph structures, each arising from modeling the interdependencies among system state nodes derived from sensors or actuators.
3. System State Update: constructs system state graphs for sensors and actuators, respectively, and updates the

system states values by aggregating information from neighboring nodes.

4. Value Prediction: decodes the upgraded features of system state nodes into prediction values for the subsequent time step values of sensors and actuators.
5. Anomaly Detection: compare the predicted values with the actual values of sensors and actuators, and compute an anomaly score to indicate anomalies.

3.3 System State Node Embedding Generation

Due to the different impacts of sensors and actuators on the physical process, as depicted in Fig. 1, time series from sensors and actuators have significant differences in their patterns. Therefore, the system state nodes are obtained by encoding the time series of sensors and actuators, respectively, and the corresponding embeddings are employed as the node features. Following that, (2a) can be represented as :

$$\begin{aligned}\hat{x}_t(Y_t) &= \ell_Y(Y_t) = Enc_Y(Y_t),\\ \hat{x}_t(U_t) &= \ell_U(U_t) = Enc_U(U_t).\end{aligned} \tag{3}$$

Specifically, encoder Enc_S and Enc_A are dertermined as two multi-layer 2D convolutional neural network (CNN) with different parameters. In a convolution layer, temporal relationships within the same channel and physical correlations between different channels are extracted simultaneously. We utilize F filters to convolute the time series Y_t and U_t using:

$$\begin{aligned}Z_S^k &= \sigma(Z_S^{k-1} * W_S^k + b_S^k),\\ Z_A^k &= \sigma(Z_A^{k-1} * W_A^k + b_A^k),\end{aligned} \tag{4}$$

where W_S^k and W_A^k are the k-th filter in Enc_S and Enc_A, which are often a small, e.g. 2×2 matrix with randomly initialized values, b_S^k and b_A^k are the corresponding biases; symbol $*$ denotes the 2D convolution operation; and σ is an activation fucntion, $k = \{1, \ldots, K\}$, $Z_S^0 = Y_t$ and $Z_A^0 = U_t$.

After the convolution process, the time series Y_t and U_t are converted into embeddings, denoted as $Z_t^S \in \mathbb{R}^{C \times D}$ and $Z_t^A \in \mathbb{R}^{C \times D}$, where C and D are hyperparameters and C refers to the number of system state nodes.

3.4 System State Structure Learning

System state structure dictates the interconnections among system states and influences the propagation and accumulation of information from neighboring nodes. Therefore, system state structure is treated as the most important factor that determines how the system evolves to the next time.

Because both sensors and actuators would influence the physical process and potentially impact each others. Consider a scenario where actuators influence the open or closed status of a pump or valve. This action could possibly lead to water inflow or outflow changes, ultimately altering the system dynamics, which subsequently affect the sensors values. Hence, we take into account the joint influence of both sensors and actuators on the system state structure.

To achieve this, two distinct graph structures are learned by capturing the interdependencies among system state nodes originating from sensors and actuators. We use relation matrix G_t^S and G_t^A to represent the graph learned from sensors and actuators at time t, where $G_{t,ij}^S$ and $G_{t,ij}^A$ represent the weight of a directed edge from node i to node j in corresponding graphs. To select the dependencies of sensor/actuator i at time t, we compute the similarity between node i's embedding vector, and the embeddings of other nodes.

$$\begin{aligned}e_{t,ji}^S &= \frac{z_{t,i}^S \top z_{t,j}^S}{\|z_{t,i}^S\| \cdot \|z_{t,j}^S\|},\\ e_{t,ji}^A &= \frac{z_{t,i}^A \top z_{t,j}^A}{\|z_{t,i}^A\| \cdot \|z_{t,j}^A\|}.\end{aligned} \tag{5}$$

Then, we select the top k such normalized dot products. Here TopK denotes the indices of top-k values among its input (i.e. the normalized dot products). The value of $k (k \leq C)$ can be chosen by the user according to the desired sparsity level.

$$\begin{aligned}G_{t,ji}^S &= e_{t,ji}^S\{j \in \mathsf{TopK}(\{e_{t,ki}^S\})\},\\ G_{t,ji}^A &= e_{t,ji}^S\{j \in \mathsf{TopK}(\{e_{t,ki}^A\})\}.\end{aligned} \tag{6}$$

Finally, these relation matrices G_t^S and G_t^A are merged together to form relation matrix G_t. Notably, the structure G_t experiences temporal changes influenced by the embedding of the sensors and actuators, making it adaptable and better suited to capture the system dynamics.

$$G_t = G_t^S + G_t^A. \tag{7}$$

3.5 System State Update

The evolution of system states in accordance with the system dynamics is achieved using GCNs which transforms and propagates node features Z_t^S and Z_t^A according to relation matrix G_t by several layers, including linear layers and nonlinear activation.

Utilizing the merged learned system state structure G_t, individual system state graphs for sensors and actuators are constructed as $\mathcal{G}_t^S = \{\mathcal{V}, G_t, Z_t^S\}$ and $\mathcal{G}_t^A = \{\mathcal{V}, G_t, Z_t^A\}$. Then, the same GCN operation is carried out on both Z_t^S and Z_t^A following [Wu *et al.*, 2020] and skip connections are used to retain a portion of the original node features at each hop and information aggregation is performed as follows:

$$H_t^{(l)} = \beta H_t^{(0)} + (1-\beta)\widetilde{A}_t H_t^{(l-1)}, H_t^{(0)} = Z_t, \tag{8}$$

where β is a hyperparameter that controls the proportion of the original features, l is the propagation depth, $\widetilde{A}_t = \widetilde{D}_t^{-1}(G_t + I)$, and $\widetilde{D}_{t,ij} = 1 + \sum_j G_{t,ij}$.

The output of the mix-hop layer is defined as follows:

$$\begin{aligned}\hat{x}_{t+1}(Y_t) &= \sum_{i=0}^{L} H_t^{S(l)} W_S^{(l)},\\ \hat{x}_{t+1}(U_t) &= \sum_{i=0}^{L} H_t^{A(l)} W_A^{(l)},\end{aligned} \tag{9}$$

where L is the propagation depth and is chosen as 2, $H_t^{S(l)}$ and $H_t^{A(l)}$ is the operation (8) of sensors and actuators. $W_S^{(l)}$ and $W_A^{(l)}$ are learnable matrices used to filter out noise that may be introduced by information propagation between nodes without dependency relationships.

Dataset	SWaT2015		SWaT2019Jul		WADI2019	
Metric	Prec	F1	Prec	F1	Prec	F1
DAGMM	0.6170±0.035	0.6201±0.004	0.1489±0.010	0.2552±0.013	0.2430±0.008	0.3656±0.019
MAD-GAN	0.5029±0.032	0.6278±0.070	0.4333±0.057	0.5872±0.094	0.6789±0.437	0.2662±0.228
USAD	0.2087±0.129	0.3010±0.175	0.3986±0.093	0.4677±0.096	0.1110±0.018	0.1892±0.028
GDN	0.9630±0.033	0.7398±0.013	0.3738±0.031	0.5254±0.021	0.6175±0.077	0.4811±0.013
GANF	0.3889±0.005	0.4925±0.005	0.4504±0.034	0.5846±0.024	0.4074±0.048	0.4695±0.020
PhyGAD	**0.9911±0.007**	**0.7843±0.012**	**0.6181±0.021**	**0.6676±0.011**	**0.6818±0.118**	**0.5001±0.022**

Table 1: Anomaly detection accuracy in terms of precision and F1-score, on three datasets with ground-truth labelled anomalies.

3.6 Value Prediction

Decoders Dec_Y and Dec_U are designed to transform the upgraded features of system state nodes into prediction values for the subsequent time step values of sensors and actuators. According to (2c) and (2d), it can be derived that

$$\begin{aligned}\hat{y}_{t+1} &= \hat{h}(\hat{x}_{t+1}(Y_t)) = Dec_Y(\hat{x}_{t+1}(Y_t)),\\ \hat{u}_{t+1} &= \hat{g}(\hat{x}_{t+1}(U_t)) = Dec_U(\hat{x}_{t+1}(U_t)).\end{aligned} \tag{10}$$

Similar to encoder Enc_Y and Enc_U, decoders Dec_Y and Dec_U consist of two multi-layer 2D CNN and extra multi-layer perceptron(MLP) to map the data to the desired dimension. In specific, We utilize F filters to deconvolute $\hat{x}_{t+1}(Y_t)$ and $\hat{x}_{t+1}(U_t)$ using:

$$\begin{aligned}Z_S^k &= \sigma(Z_S^{k-1} * W_S^k + b_S^k),\\ Z_A^k &= \sigma(Z_A^{k-1} * W_A^k + b_A^k),\end{aligned} \tag{11}$$

where $k = \{K+1, \ldots, 2K\}$, $Z_S^K = \hat{x}_{t+1}(Y_t) and Z_A^K = \hat{x}_{t+1}(U_t)$. Following an MLP, (2c) and (2d) can be derived and represented as

$$\begin{aligned}\hat{y}_{t+1} &= \sigma(Z_S^{2k} \top W_S + b_S),\\ \hat{u}_{t+1} &= \sigma(Z_A^{2k} \top W_A + b_A).\end{aligned} \tag{12}$$

Moreover, the loss function is minimized by using the mean squared error (MSE) between the prediction output and the observed data:

$$L_t^{MSE} = \|\hat{y}_t - y_t\|_2^2 + \|\hat{u}_t - u_t\|_2^2. \tag{13}$$

Anomaly Detection. We adopt the same method as in GDN [Deng and Hooi, 2021] to calculates the prediction deviation for each sensor and actuator:

$$Err_t^i = \begin{cases} \|\hat{y}_t^i - y_t^i\|_2^2, \; i = 1, \ldots, N, \\ \|\hat{u}_t^i - u_t^i\|_2^2, \; i = N+1, \ldots, N+M. \end{cases} \tag{14}$$

The error values for each sensor and actuator are normalized:

$$\xi_t^i = \frac{Err_t^i - \hat{\mu}^i}{\hat{\sigma}^i}. \tag{15}$$

where $\hat{\mu}^i$ and $\hat{\sigma}^i$ represent the median and interquartile range (IQR) in the temporal dimension. Then, the aggregated anomaly score is calculated at time step t as:

$$\xi_t = \max_i \xi_t^i. \tag{16}$$

	Prec	F1
GDN	0.3991±0.120	0.4991±0.065
PhyGAD	**0.5348±0.274**	**0.5648±0.138**

Table 2: Anomaly performance of models trained on startup phase data.

4 Experiment

4.1 Experimental Setup

Datasets. Experiments are conducted on three real-world CPS datasets, including SWaT2015, SWaT2019Jul, and WADI2019, in which the quantity of sensors and actuators are different.

Baseline Methods. SOTA methods include DL-based methods DAGMM [Zong *et al.*, 2018], MAD-GAN [Li *et al.*, 2019], USAD [Audibert *et al.*, 2020], and GNN-based methods GDN [Deng and Hooi, 2021] and GANF [Dai and Chen, 2022].

Implementation details. The model is trained for 50 epochs in each time of training, and all experiments are conducted 10 times to obtain the average results. The embedding depth of system nodes D is 16 and the number of convolutional layers K is 4. The value of $TopK$ is 3 for SWaT2015, and is 5 for SWaT2019Jul and WADI2019.

4.2 Anomaly Detection Performance

The anomaly detection performance is evaluated using F1 score (F1) and precision (Prec). The results are displayed in Table. 1. Note that each model is tested with various anomaly score thresholds and the displayed results are the highest F1 score achieved. PhyGAD has superior performance over all the other five baselines.

Compared to PhyGAD, DAGMM overlooks the temporal correlation between CPS time series which is essential and leads to lower F1 scores. MAD-GAN and USAD utilize time frames to include temporal relationships among CPS time series. However, they rely heavily on data distributions and perform poorly when CPS system states exhibit temporal variability, resulting in lower precision (Prec). As for GDN, it learns the dependency graph between sensors and actuators, however, the dependency relationships are fixed once trained and are not suitable for time-varing system dynamics of CPS. Moreover, GANF is capable of learning the evolution of the graph structure and identifying data distribution drift. However, GANF lacks the ability to capture higher-order correlations within the system, leading to overall lower performance.

Furthermore, we analyse the model's generalization capability for anomaly detection. In the SWaT2015 dataset, it requires around 5-6 hours for the system to transition to a stable state from its initial state[Goh *et al.*, 2017], which is called *startup phase*. In the startup phase, while sensors and actuators operate normally, they exist in an unstable state characterized by abrupt and dramatic variations in values. Hence, we carried out experiments that the models are trained on the data of startup phase and then implement anomaly detection on test dataset. The results in Table 2 indicate a decline in anormaly detection performance when the models are trained on startup phase data. However, PhyGAD exhibit superior generalization capability than GDN. This is attributed to PhyGAD's utilization of system state graphs, encompassing the physical process and effectively extracting higher-order system state information. This approach avoids sole reliance on data distribution and effectively mitigates the impact of data distribution drift.

4.3 Ablation Analysis

To test the validity of each designed module, we give several ablation experiments on SWaT2015, and the results are presented in Table. 3.

Without separation of sensors and actuators. Time series of sensors and actuators are combined into one as an input to a single encoder. Following that, a single GCN module and a single decoder to generate the output. The result shows a decrease of 8.5% in F1 score, which validates the importance of separately processing sensors and actuators. This is because sensors and actuators have different effects on CPS and exhibit significant differences in their form and behavior. Using a single encoder to extract high-order features from both types of data can introduce noise.

Without system state update via GCN. Once the system state update module via GCN is removed, the model's performance metrics exhibit a decrease. Integrating neighboring information for system state updates is recommended, as it proves advantageous for predicting both the system state and subsequent sensor and actuator values.

Without the merge of two learned structure for system state graph. The merge of two learned structure from sensors and actuators, as depicted in (7) is discarded and the system states are updated following the original corresponding structure. The results shows a decrease of 15.4% in F1 score. This indicates that integrating the system structures of sensors and actuators provides richer interaction information and is vital.

Without system state node embeddings. Instead of CNN, we use an MLP to replace the original encoder, resulting in an embedding matrix with the same dimensions as the input data. Also, the learned system state structure is not merged because the number of sensors and actuators is usually not matching. The performance decreases significantly. The F1 score performance decreases by 13.3%. This validates the importance of extracting high-order features from the physical process.

	Prec	F1
PhyGAD	**0.9911±0.007**	**0.7843±0.012**
- Without separation S&A	0.9872±0.006	0.7634±0.018
- Without system state update	0.9888±0.009	0.7284±0.197
- Without merge structures	0.9880±0.009	0.6627±0.259
- Without node embeddings	0.5131±0.231	0.5510±0.247

Table 3: Anomaly detection performance of PhyGAD and its variants on SWaT2015.

5 Conclusion

In our study, we proposed PhyGAD which employed a novel system state graph to understand the hidden physical process of CPSs. The prediction of time series for anomaly detection is carried out by predicting future system states innovatively via GCN. Experimental results on real-world datasets showcase the superiority of our proposed model in accurately detecting anomalies compared to baseline methods. This research provides a new perspective for CPS time series modeling graph by building physical-orient system state graph instead of traditional entity-orient sensor-actuator graph.

6 Acknowledgements

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LDT23F01012F01, in part by the Fundamental Research Funds for the Provincial Universities of Zhejiang Grant GK229909299001-008 and the National Natural Science Foundation of China under Grant 62003120 and Grant 62372146.

References

[Audibert *et al.*, 2020] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3395–3404, 2020.

[Dai and Chen, 2022] Enyan Dai and Jie Chen. Graph-augmented normalizing flows for anomaly detection of multiple time series. *arXiv preprint arXiv:2202.07857*, 2022.

[Deng and Hooi, 2021] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4027–4035, 2021.

[Feng and Tian, 2021] Cheng Feng and Pengwei Tian. Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2858–2867, 2021.

[Giraldo *et al.*, 2018] Jairo Giraldo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. A survey of physics-based attack detection in cyber-physical systems. *ACM Computing Surveys*, 51(4):1–36, 2018.

[Goh *et al.*, 2017] 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*, pages 88–99. Springer, 2017.

[Haddad and Lee, 2022] Wassim M Haddad and Junsoo Lee. Finite-time stabilization and optimal feedback control for nonlinear discrete-time systems. *IEEE Transactions on Automatic Control*, 68(3):1685–1691, 2022.

[Han and Woo, 2022] Siho Han and Simon S Woo. 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*, pages 2977–2986, 2022.

[Inoue *et al.*, 2017] Jun Inoue, Yoriyuki Yamagata, Yuqi Chen, Christopher M Poskitt, and Jun Sun. Anomaly detection for a water treatment system using unsupervised machine learning. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1058–1065. IEEE, 2017.

[Jagannathan, 2001] Sarangapani Jagannathan. Control of a class of nonlinear discrete-time systems using multilayer neural networks. *IEEE Transactions on Neural Networks*, 12(5):1113–1120, 2001.

[Jiao *et al.*, 2021] Pengfei Jiao, Tianpeng Li, Yingjie Xie, Yinghui Wang, Wenjun Wang, Dongxiao He, and Huaming Wu. Generative evolutionary anomaly detection in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12234–12248, 2021.

[Jin *et al.*, 2023] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv preprint arXiv:2307.03759*, 2023.

[Li *et al.*, 2019] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*, pages 703–716. Springer, 2019.

[Luo *et al.*, 2021] Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng Yao. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Computing Surveys (CSUR)*, 54(5):1–36, 2021.

[Qi *et al.*, 2022] Panpan Qi, Dan Li, and See-Kiong Ng. Mad-sgcn: Multivariate anomaly detection with self-learning graph convolutional networks. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1232–1244. IEEE, 2022.

[Umer *et al.*, 2022] Muhammad Azmi Umer, Khurum Nazir Junejo, Muhammad Taha Jilani, and Aditya P Mathur. Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection*, 38:100516, 2022.

[Urbina *et al.*, 2016] David I Urbina, Jairo A Giraldo, Alvaro A Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1092–1105, 2016.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 753–763, 2020.

[Wu *et al.*, 2023] Weiqiang Wu, Chunyue Song, Jun Zhao, and Zuhua Xu. Physics-informed gated recurrent graph attention unit network for anomaly detection in industrial cyber-physical systems. *Information Sciences*, 629:618–633, 2023.

[Ye *et al.*, 2022] Junchen Ye, Zihan Liu, Bowen Du, Leilei Sun, Weimiao Li, Yanjie Fu, and Hui Xiong. Learning the evolutionary and multi-scale graph structure for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2296–2306, 2022.

[Zhang *et al.*, 2022a] Weiqi Zhang, Chen Zhang, and Fugee Tsung. Grelen: Multivariate time series anomaly detection from the perspective of graph relational learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, volume 7, pages 2390–2397, 2022.

[Zhang *et al.*, 2022b] Yan Zhang, Yu Guo, Zhang Zhang, Mengyuan Chen, Shuo Wang, and Jiang Zhang. Universal framework for reconstructing complex networks and node dynamics from discrete or continuous dynamics data. *Physical Review E*, 106(3):034315, 2022.

[Zheng *et al.*, 2023] Yu Zheng, Huan Yee Koh, Ming Jin, Lianhua Chi, Khoa T Phan, Shirui Pan, Yi-Ping Phoebe Chen, and Wei Xiang. Correlation-aware spatial-temporal graph learning for multivariate time-series anomaly detection. *arXiv preprint arXiv:2307.08390*, 2023.

[Zong *et al.*, 2018] 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.