GRELEN: Multivariate Time Series Anomaly Detection from the Perspective of Graph Relational Learning

Weiqi Zhang[1,2], **Chen Zhang**[3 *], **Fugee Tsung**[1,2]
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
[3]Tsinghua University
wzhangcd@connect.ust.hk, zhangchen01@tsinghua.edu.cn, season@ust.hk

Abstract

System monitoring and anomaly detection is a crucial task in daily operation. With the rapid development of cyber-physical systems and IT systems, multiple sensors get involved to represent the system state from different perspectives, which inspires us to detect anomalies considering feature dependence relationship among sensors instead of focusing on individual sensor's behavior. In this paper, we propose a novel Graph Relational Learning Network (GReLeN) to detect multivariate time series anomaly from the perspective of between-sensor dependence relationship learning. Variational AutoEncoder (VAE) serves as the overall framework for feature extraction and system representation. Graph Neural Network (GNN) and stochastic graph relational learning strategy are also imposed to capture the between-sensor dependence. Then a composite anomaly metric is established with the learned dependence structure explicitly. The experiments on four real-world datasets show our superiority in detection accuracy, anomaly diagnosis, and model interpretation.

1 Introduction

Time series anomaly detection is an important research topic and has wide applications. For example, in industry, sensors are mounted in the system for production line monitoring. Generally, a system has different sensors to describe its global state and requires multivariate time series anomaly detection techniques to trigger system-level alarm [Su *et al.*, 2019].

As sensor dimension increases, dependence relationship between sensors becomes more important for efficient anomaly detection. For example, Fig.1 plots time series signals of four sensors in a secure water treatment system (SWaT). It shows one anomaly happened on sensor *LIT-101*. Then sensor *FIT-101* and actuator *MV-101* follow the anomaly and drop to low level values, while sensor *P-101* remains uninfluenced. Yet from the detection perspective, it does not mean sensor *P-101* is useless, since its dependence relationship between others is actually changed and will help detect anomaly. Similarly, the relationship between attacked sensor and other unattacked sensors will also be the indicator of anomaly. It motivates that monitoring dependence relationship would be more powerful in complex systems, compared with merely focusing on monitoring each individual sensor's behavior.

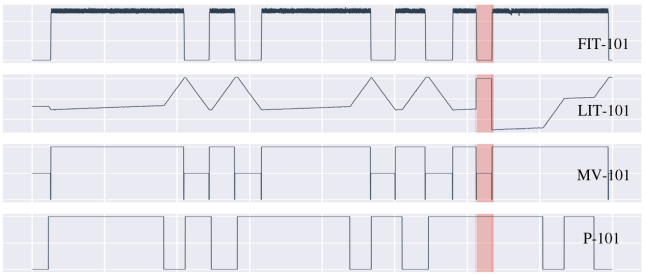

Figure 1: An example of anomaly in multivariate time series for a secure water treatment system: Signals of four different sensors are plotted and one anomalous period is highlighted by red block.

So far numerous works have been proposed for multivariate time series anomaly detection. Neural networks are widely used given high data dimension and generally have better performance than traditional statistical methods. Considering the anomalies are generally unpredictable with various patterns, unsupervised-learning-based methods are more attractive. The basic framework is to use no-anomaly data to establish a model describing the normal system pattern. Then testing data that cannot be well fitted by the pre-trained model would be regarded as anomalies.

However, most existing methods cannot model the anomaly from the perspective of multivariate time series dependence relationship. Though there are also some graph learning based methods considering dependence relationship in modeling, their intrinsic information loss will consequently undermine the detection power, which will be illustrated in Section 4.6.

In this work, we propose a novel anomaly detection framework, called Graph Relational Learning Network (GReLeN), to fill the gaps mentioned before. Our contributions could be summarized as follows:

(a) GReLeN utilizes VAE structure to learn a probabilistic relation graph for multiple sensors, where the latent vari-

*Corresponding author.

able of VAE is used to capture the dependence relationship between sensors.

(b) We propose to detect anomaly from the perspective of graph relational learning for the first time. The anomaly detection score is established based on the explicit dependence relationship learned from model.

(c) The experiments on four real world datasets show that our proposed model could describe the latent dependence among sensors well and has good interpretability corresponding to domain knowledge. Our proposed model also shows superiority regarding detection accuracy and anomaly diagnosis.

2 Related Work

2.1 Multivariate Time Series Anomaly Detection

Considering the quick development of neural network, only deep-learning based models are reviewed in this subsection. The current works could be classified as prediction-based methods and reconstruction-based methods.

(a) Prediction-based models: Prediction-based models utilize advanced deep learning components to capture spatial-temporal dependence, and use the prediction error as anomaly score for detection. Well designed modules could achieve more accurate prediction, and consequently be helpful to detect abnormal ones. [Deng and Hooi, 2021] proposes a GNN based method to aggregate the information between sensors. [Zhao *et al.*, 2020] combines feature-oriented Graph Attention Network (GAT) and time-oriented GAT to handle spatial dependence and temporal dependence while predicting.

(b) Reconstruction-based models: Reconstruction-based methods hope to find a latent representation for the entire time series for data reconstruction. Loss function of model is a common choice for anomaly score. [Li *et al.*, 2019] uses Long Short-Term Memory (LSTM) as basic cells, and considers the entire variable set concurrently with a Generative Adversarial Network (GAN) framework. In [Su *et al.*, 2019], the proposed *OmniAnomaly* uses stochastic Recurrent Neural Network (RNN) to find robust representations for multivariate time series. [Audibert *et al.*, 2020] proposes an autoencoder architecture whose adversarial-style learning is inspired by GAN. Recent work [Abdulaal *et al.*, 2021] utilizes spectral analysis on the latent representation and produces a synchronized representation for multivariate data. However, in these works, no dependence relationship between variables has been considered and modeled explicitly.

2.2 Graph Learning in Multivariate Time Series

GNN is commonly used in multivariate time series prediction, where they regard each single time series as a node and their dependence as edges in graph. A common assumption is that the graph structures should be pre-defined based on domain knowledge. It makes the model very sensitive to the choice of graph, and loses generality. Recent studies adopt a data-driven way (i.e. graph learning strategy) to learn the graph structure between different time series automatically. In [Shang *et al.*, 2020; Wu *et al.*, 2020], graph structure learning is used for improving multivariate time series prediction. In [Kipf *et al.*, 2018], a Neural Relational Inference (NRI) model is used to learn the system dynamics from observational data based on VAE. However, their node-to-edge and edge-to-node operations on graph result in heavy computation complexity, which hinders their application in high-dimensional time series.

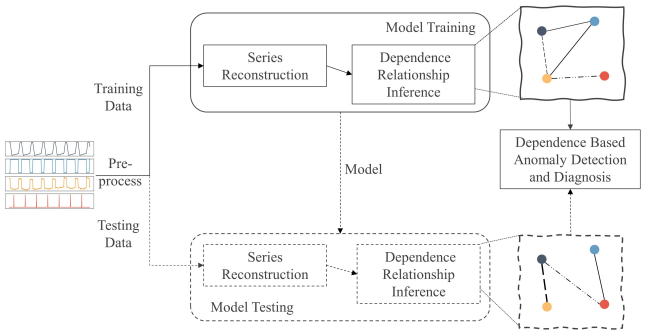

Figure 2: Proposed structure for anomaly detection with proposed Graph Relational Learning Network (GReLeN) model.

[Deng and Hooi, 2021] is the most similar work to ours. It uses node embedding to learn graph structure and do prediction-based detection. However, its learning strategy assumes the graph structure should be determined in advance instead of automatically learned, which may lead to low model robustness. Besides, it assumes the graph structure is sparse and unchanged over time, which leads to information loss. Furthermore, its learned graph structure only acts as an intermediate module to improve prediction, but is not used to indicate any anomaly explicitly. More comparisons will be explained in Section 4.6.

3 Proposed Model

3.1 Problem Formulation

Consider a system with N sensors, our sequential data collected over T timestamps could be denoted as $\mathbf{X} \in R^{N \times T}$. We denote the data collected at each sensor i and each timestamp t by using subscript and superscript respectively. That is, $X_i = \mathbf{X}_{i,:}, X^t = \mathbf{X}_{:,t}$. In our unsupervised assumption, the system operates in normal condition during the first T_{train} timestamps, which could be regarded as training data. Our task here is to identify potential anomaly in the following timestamps $t > T_{train}$. All the data coming after T_{train} will be regarded as testing set.

A sliding window w is used for constructing the samples: $S^t = X^{t-w+1:t}$. The task of our anomaly detection model f is to provide a set of binary labels indicating whether there's any anomaly for certain timestamp of the testing set: $y^t = f(S^t), y^t \in \{0, 1\}, 1 \leq t \leq T_{test}$. The mathematical notations are shown in Table 1.

3.2 Anomaly Detection Process

As shown in Fig.2, our proposed anomaly detection structure is composed of an offline training module and an online testing module. The offline training phase uses normal training data to learn the normal sensor dependence relationship together with the data reconstruction mechanism. Nor-

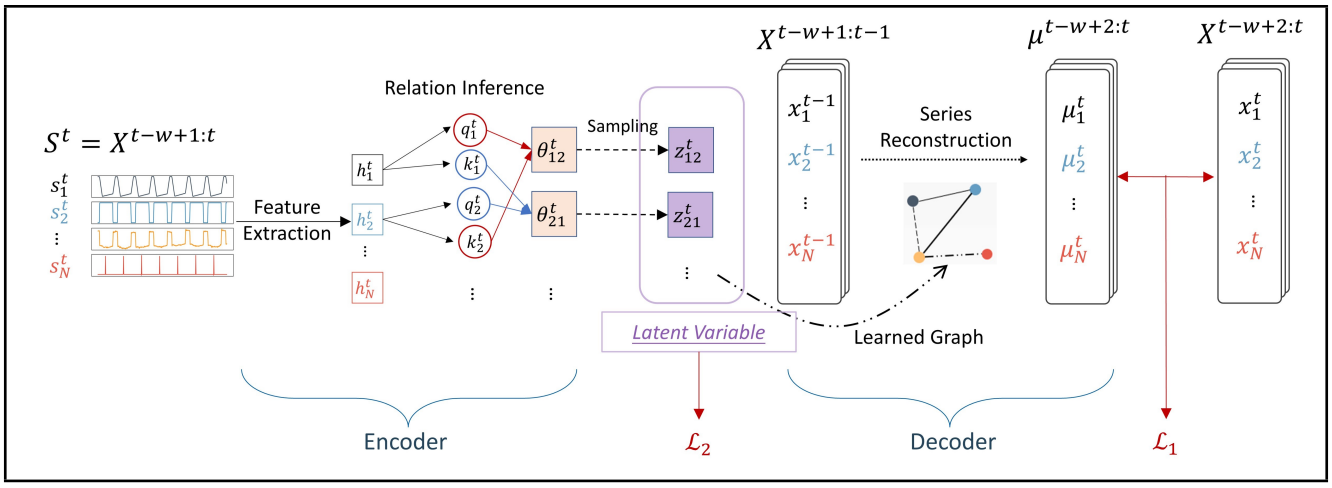

Figure 3: Proposed Graph Relational Learning Network (GReLeN) for multivariate time-series anomaly detection with graph relational inference.

Indices	
N, T	Number of sensors and timestamps in collected data
X_i, X^t	Data collected at sensor i and timestamp t
w	Length of sliding window to generate sample
S^t	Sliding windowed sample data at timestamp t
h_0	Number of types of dependence relationship we hope to learn
Θ^t	Probability parameters for dependence relationship
$\mathcal{Z}^t$	Practical dependence relationship.
μ_i^t, σ	Parameters of Gaussian distribution while reconstructing
$\tilde{p}_k^t, \tilde{q}_k$	Learned and prior probability to fall into dependence type k

Table 1: Notations

mal pattern of time-varying multivariate system and the pre-trained model could be obtained. In the online testing phase, the model is used for testing data stream without additional training. By monitoring the sensor dependence relationship of the testing data, anomaly detection and diagnosis could be achieved.

3.3 Model Overview

The overall architecture of our model GReLeN is shown in Fig.3. GReLeN follows the main structure of VAE. In particular, three sub-modules are contained in our proposed framework.

(a) **Encoder**: In the encoder part, we use linear layers to extract temporal features for each time series input. Self-attention-based [Vaswani *et al.*, 2017] mechanism would be applied for efficient high-level feature capturing and dependence relationship inference.

(b) **Latent variable with sampling**: The output of encoder is used to parameterize the distribution of dependence relationship among different time series. Reparametrization trick would be used for enabling backpropagation.

(c) **Decoder**: In the decoder part, our goal is to reconstruct the input series recurrently. A spatial-temporal cell would be applied here based on the relational graph we have sampled by latent variable.

3.4 Encoder

The input of encoder is a sample series S^t generated by sliding window w. The target of encoder is to learn the distribution q_ϕ that the latent dependence relationship $Z^t|S^t$ follows. In particular, we can denote that q_ϕ is parameterized by $\Theta^t = \{\theta_{i,j}^t, i, j = 1, \ldots, N\}$.

We first use a linear layer to extract temporal feature H^t from the raw input series, where $W_l \in R^{T \times c_1}$ is the learnable parameters and c_1 is the dimension of high-level extracted temporal features: $H^t = S^t W_l$.

Self-attention-based mechanism would be applied to calculate Θ^t based on $H^t = \{h_i^t, i = 1, \ldots, N\}$. More specific, high-level extracted feature will be projected into latent query, key subspace Q^t, K^t with learnable parameters W_q, W_k. Then the dependence relationship distribution parameters can be obtained by dot-product attention:

$$Q^t = H^t W_q, K^t = H^t W_k, \Theta^t = Q^t {K^t}^T = \left[\theta_{i,j}^t\right]_{1 \le i,j \le N}$$

Further, it is natural to extend the multi-head attention strategy here to learn multiple types of dependence and improve the representation ability of latent variables. For the attention with h_0 heads, we can get Θ^t as follows:

$$\Theta^t = \text{Softmax}\left(\text{Concat}\left(head_1, ..., head_{h_0}\right)\right) = \left[\boldsymbol{\theta}_{i,j}^t\right]_{1 \le i,j \le N}$$

$$Q_h^t = H^t W_{q,h}, K_h^t = H^t W_{k,h}, head_h = Q_h^t {K_h^t}^T, 1 \le h \le h_0$$

where $W_{q,h}$ and $W_{k,h}$ are learnable parameters of the h^{th} head, $\Theta^t \in R^{h_0 \times N \times N}$.

Here, each pair-wise parameter $\boldsymbol{\theta}_{i,j}^t$ is a vector with length h_0, since it means we have h_0 types of dependence in total. It's intuitive that we assume $\boldsymbol{\theta}_{i,j}^t$ is the parameter for categorical distribution (e.g. multinomial distribution). Therefore, we use the softmax function to guarantee the sum of the probability of all these types equals 1.

3.5 Sampling

The above encoder formulates the distribution q_ϕ to generate latent variable $\boldsymbol{z}_{ij}^t \in R^{h_0}$, which is supposed to be a one-hot vector indicating which type of dependence it belongs to. However, the discretization of $\boldsymbol{z}_{ij}^t$ hampers backpropagation and brings additional trouble in the training process.

A recent solution is to use Gumbel-Softmax categorical reparameterization trick [Jang *et al.*, 2017]. It utilizes the continuous variable approximation to make backpropagation possible by

$$\boldsymbol{z}_{ij}^t = \text{softmax}\left(\frac{\boldsymbol{\theta}_{ij}^t + \boldsymbol{g}}{\tau}\right)$$

where $\boldsymbol{g}$ is a h_0 dimension vector of i.i.d. samples drawn from a Gumbel(0,1) distribution and the continuous distribution converges to expected one-hot samples if we have temperature parameter $\tau \to 0$.

3.6 Decoder

The decoder is designed to reconstruct the original input series with the guidance of learned dependence relationship. Recall that $\boldsymbol{z}_{i,j}^t$ is an h_0 dimension vector. Furthermore, $\mathcal{Z}^t \in R^{h_0 \times N \times N}$ could be considered as the concatenation of all the h_0 graph adjacency matrices. We model the reconstruction module by following a Gaussian distribution assumption:

$$p_\psi\left(X^{t'+1}|X^{t'}, ..., X^{t-w+1}, \mathcal{Z}^t\right) = \mathcal{N}\left(\boldsymbol{\mu}^{t'+1}, \boldsymbol{\sigma}^2 \boldsymbol{I}\right) \quad (1)$$

for $t' = t-w+1, \ldots, t-1$. We divide the h_0 types of learned dependence into two parts. One of the graphs will be treated as "null dependence", which aims to guarantee the sparsity via prior setting. The other $h_0 - 1$ graphs denote different relationship structures, which will be used as graph adjacency matrix in Diffusion Convolutional Gated Recurrent Unit (DCGRU)[Li *et al.*, 2018] to do the recurrent reconstruction. All the outputs from different graphs will be summed up to obtain the final output.

Mathematically and formally, for any graph dependence structure with adjacency matrix A, DCGRU combines GRU temporal cell and diffusion convolutional GNN:

$$\begin{aligned}
R^{t'} &= \text{sigmoid}\left(W_R \star_A \left[X^{t'} \| H_g^{t'-1}\right] + b_R\right), \\
C^{t'} &= \tanh\left(W_C \star_A \left[X^{t'} \| \left(R^{t'} \odot H_g^{t'-1}\right)\right] + b_C\right), \\
U^{t'} &= \text{sigmoid}\left(W_U \star_A \left[X^{t'} \| H_g^{t'-1}\right] + b_U\right), \\
H_g^{t'} &= U^{t'} \odot H_g^{t'-1} + \left(1 - U^{t'}\right) \odot C^{t'}
\end{aligned}$$

where the graph convolution operation $\star_A$ is defined as:

$$W_Q \star_A Y = \sum_{k=0}^{K} \left(w_{k,1}^Q \left(D_O^{-1}A\right)^k + w_{k,2}^Q \left(D_I^{-1}A^T\right)^k\right) Y$$

Here, D_O and D_I are the out-degree and in-degree matrix of graph. $\|$ denotes concatenation. For $Q = R, U, C$, $w_{k,1}^Q, w_{k,2}^Q, b_Q$ are learnable parameters and K is the hyperparameter reprensenting the diffusion degree while convolution. At each time step t', the hidden state $H_g^{t'}$ serves as our final reconstruction $\mu^{t'+1}$.

3.7 Objective Function and Training

As a VAE-based model, the objective function could be maximizing the evidence lower bound (ELBO) of VAE:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\mathcal{Z}^t|S^t)}\left[\log p_\psi\left(S^t|\mathcal{Z}^t\right)\right] - \text{KL}\left[q_\phi\left(\mathcal{Z}^t|S^t\right) \| p_\psi\left(\mathcal{Z}^t\right)\right]$$

The first term $\mathbb{E}_{q_\phi(\mathcal{Z}^t|S^t)}\left[\log p_\psi\left(S^t|\mathcal{Z}^t\right)\right]$ is the so-called reconstruction loss and $\text{KL}\left[q_\phi\left(\mathcal{Z}^t|S^t\right) \| p_\psi\left(\mathcal{Z}^t\right)\right]$ indicates the Kullback-Leibler (KL) divergence loss.

In the decoder, we could decompose the first term in a recurrent way:

$$p_\psi\left(S^t|\mathcal{Z}^t\right) = \Pi_{t'=t-w+1}^{t-1} p_\psi\left(X^{t'+1}|X^{t'}, ..., X^{t-w+1}, \mathcal{Z}^t\right) \quad (2)$$

Combining Equation (1) and (2), for each sample S^t, the total reconstruction loss could be estimated by

$$\mathcal{L}_1 = -\sum_{i=1}^{N} \sum_{t'=t-w+2}^{t} \frac{||x_i^{t'} - \mu_i^{t'}||^2}{2\sigma^2}$$

The second term in $\mathcal{L}$, i.e., the KL divergence loss, measures how well the dependence relationship we learn from encoder matches the prior distribution.

Given each sensor has influence or could be influenced by only a small proportion of others, we would assume that the dependence relationship is sparse among all these sensors. Therefore, the type denoting "null dependence" should have a larger prior. Among all these h_0 graphs, only $h_0 - 1$ graphs will be used in decoder for reconstruction. The graph which has not been used while decoding would be regarded as "null structure". If two series have connections in "null structure", they will not have dependence in practical because of the discarding.

Denote our prior multinomial distribution parameters as $\tilde{q} = (\tilde{q}_1, \tilde{q}_2, ..., \tilde{q}_{h_0})$. Mathematically, the KL divergence loss could be computed by

$$\mathcal{L}_2 = \text{KL}\left[q_\phi\left(\mathcal{Z}^t|S^t\right) \| p_\psi\left(\mathcal{Z}^t\right)\right] = \sum_{k=1}^{h_0} \tilde{p}_k^t \log\left(\frac{\tilde{p}_k^t}{\tilde{q}_k}\right)$$

where $\tilde{p}_k^t = \sum_{i=1}^{N}\sum_{j=1}^{N} z_{i,j}^t(k)$ is the sum of probability that the learned dependence between sensor i and sensor j belongs to type k.

4 Experiments

4.1 Datasets

Datasets Name	SWaT	WADI	SMD	PSM
Training size	496800	1048571	708377	129784
Testing size	449919	172801	708393	87851
Number of Sensors	51	112	38	26
Number of Attacks	41 (36)	15	327	72
Anomaly Durations	$100 \sim 34208$	$87 \sim 1740$	$2 \sim 3161$	$1 \sim 8861$
Anomaly rate(%)	11.97	5.99	4.16	27.76

Table 2: Detailed characteristics of the four real-world datasets.

We conduct our proposed model on four real-word datasets: SWaT (Secure Water Treatment Testbed) [Goh *et*

Method	SWAT			WADI			SMD			PSM			Average
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	F1
ADD	98.0±0.17	63.4±0.24	77.0±0.19	89.0±0.34	40.3±0.44	55.5±0.47	99.8±0.93	40.9±0.56	58.1±0.62	82.3±1.04	41.5±0.40	55.2±0.40	61.4
IF	80.3±0.17	35.6±0.27	49.4±0.25	99.3±0.24	24.1±1.29	38.9±1.68	99.7±0.36	47.6±0.63	64.5±0.59	75.9±0.81	57.3±0.86	65.3±0.85	54.5
LSTM-VAE	96.1±0.83	59.3±0.91	73.4±0.74	87.4±0.63	13.6±0.71	23.5±1.08	87.0±0.94	79.4±0.74	83.0±0.82	81.0±0.58	58.2±0.88	67.7±0.59	61.9
MAD-GAN	97.3±0.56	64.6±0.49	77.7±0.43	41.0±0.60	34.2±0.67	37.3±0.46	17.5±0.43	92.9±0.80	29.5±0.59	43.4±0.83	63.7±0.35	51.6±0.57	49.1
OmniAnomly	71.7±0.78	96.3±0.76	82.2±0.40	26.9±0.73	98.2±0.86	42.2±0.91	97.8±1.07	94.3±0.70	**96.0±0.61**	96.0±0.39	80.9±0.68	87.8±0.48	77.1
USAD	97.9±0.20	72.7±0.67	83.4±0.43	64.4±0.68	31.6±0.69	42.3±0.63	93.6±0.46	95.5±0.70	94.6±0.57	92.1±1.07	57.6±1.00	70.9±0.81	72.9
GDN	98.2±0.16	67.3±0.41	79.8±0.27	98.2±0.32	39.9±1.11	56.7±1.10	58.1±0.93	56.6±0.41	57.3±0.54	43.4±1.03	76.0±0.61	55.2±0.81	62.3
RCoders	90.1±0.32	76.8±0.71	82.9±0.53	64.5±0.48	33.5±0.53	44.1±0.53	81.2±0.67	80.0±0.82	80.6±0.61	98.9±0.40	87.3±0.32	92.7±0.25	75.1
GReLeN_Loss	77.8±0.35	78.0±0.27	77.9±0.08	80.8±0.97	37.4±0.74	51.1±0.82	79.4±0.85	79.1±1.03	79.2±0.36	58.2±0.37	96.5±0.41	72.6±0.21	70.3
GReLeN_Topk	91.0±0.32	80.9±0.14	85.7±0.12	79.3±0.52	48.2±0.91	59.9±0.70	88.2±0.61	86.3±0.64	87.2±0.21	95.8±0.98	77.3±0.49	85.6±0.49	79.6
GReLeN_Degree	95.6±1.04	83.5±0.51	**89.1±0.21**	77.3±1.43	61.3±0.34	**68.2±0.74**	88.2±1.03	95.1±0.90	91.5±0.91	94.2±1.22	92.1±1.12	**93.1±0.63**	**85.5**

Table 3: Performance comparison on real-world datasets with 5 runs. Best F1 score is highlighted by bold style.

al., 2016], WADI (Water Distribution Testbed) [Ahmed *et al.*, 2017], SMD (Server Machine Dataset) [Su *et al.*, 2019], and PSM (Pooled Server Metrics) [Abdulaal *et al.*, 2021]. Normal data would be divided into training data (80%) and validation data (20%). Training data is used for model training, and validation helps model selection. Anomaly only lies in testing data. More descriptions about datasets are shown in Table 2.

In particular, SWaT, WADI and PSM have only one entity to be monitored. SMD has 28 different entities, where each entity has 38 sensors. Different entities need to be trained and tested independently.

4.2 Baselines and Evaluation Metrics

To illustrate the superiority and effectiveness of our proposed model, we compare our model with the following traditional methods **IF: Isolated Forest** [Liu *et al.*, 2008]; **AAD: Active Anomaly Discovery** [Das *et al.*, 2016], and deep-learning-based state-of-the-art baselines: **LSTM-VAE** [Park *et al.*, 2018]; **MAD-GAN** [Li *et al.*, 2019]; **OmniAnomaly** [Su *et al.*, 2019]; **USAD** [Audibert *et al.*, 2020]; **GDN** [Deng and Hooi, 2021]; **RCoders** [Abdulaal *et al.*, 2021].

The evaluation metrics we consider here include Precision, Recall and F1 Score (F1). A commonly used point adjustment strategy [Su *et al.*, 2019; Deng and Hooi, 2021; Audibert *et al.*, 2020] is involved to assure that if at any timestamp of an anomaly's occurring period the anomaly is detected, we regard it as an accurate detection.

4.3 Experimental Setup

In our experiments, we set $c_1 = 64, h_0 = 4$ and all the hidden dimension in DCGRU cell equal to 64. The number of DCRNN layers is 2 in SWaT and 1 in other datasets. The batch size is 32. The method is implemented by Pytorch using the Adam [Kingma and Ba, 2015] optimizer with the learning rate 1×10^{-3}. All the samples are generated with a sliding window $w = 30$. 100 epochs are used for training. Our prior multinomial distribution parameters setting is $\tilde{q} = [0.91, 0.03, 0.03, 0.03]$.

4.4 Anomaly Score Establishment

We compare two ways of anomaly score establishment in this work. First we utilize the loss function, a common type of anomaly score regarding reconstruction-based methods. More specifically, we use KL divergence loss $\mathcal{L}_2$ here (denoted by *GReLeN_Loss*). By experiments we find that $\mathcal{L}_2$ is more stable and comprehensive than reconstruction loss $\mathcal{L}_1$.

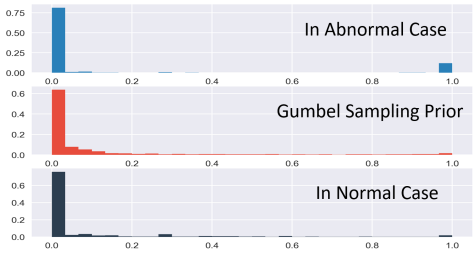

Figure 4: Distribution for dependence relationship in abnormal case, expected prior and normal case (experiments for SWAT).

However, the KL divergence loss mainly focuses on global deviation of the dependence structure from its prior distribution. As mentioned before, the specific dependence relationship between sensors also reveals the anomaly and includes more local information. Therefore, in dependence graphs, we use the sum of all the sensors' in/out degree as another anomaly score (denoted by *GReLeN_Degree*). Consider only sudden changes should be detected as anomaly, while the gradual shift is acceptable for normal operation, we use a moving average filter strategy for noise smoothing. By experiments, we find that the sudden, anomalous changes could be enlarged and the gradual, noisy shift could be removed via this strategy.

Mathematically we have, *GReLeN_Loss* = $\mathcal{L}_2$, *GReLeN_Degree* = $\sum_{i=1}^{N} d_i^{in} + d_i^{out}$. Here, d_i^{in}, d_i^{out} are moving filtered in/out degree of node i in the learned graph respectively. In our composite score, we consider the bi-direction dependence changes for all the time series. We believe that our anomaly score could provide a comprehensive metric to reveal the changes in the state of system.

4.5 Results

We evaluate our proposed model with 5 runs. Consistent with [Audibert *et al.*, 2020; Abdulaal *et al.*, 2021; Su *et al.*, 2019], we use grid search to get possible anomaly thresholds for every model and report the results with the highest F1 score.

As shown in Table 3, our proposed GReLeN model achieves the best F1 performance on average, with the highest F1 on three datasets and comparable results on SMD. Comparing with other SOTA baselines, GReLeN has better and more balanced performance between Precision and Recall. Furthermore, WADI has low anomaly rate and consequently bring large challenges to other works. Yet our GReLeN could still achieve a high Recall performance with low

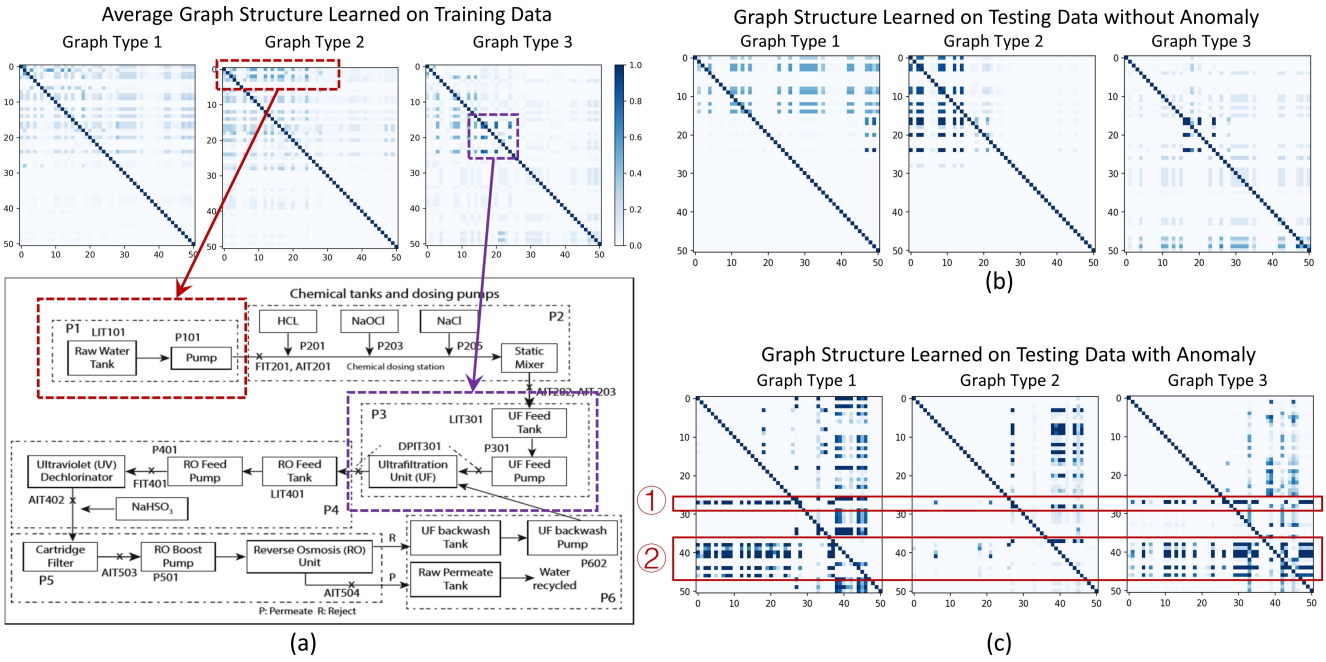

Figure 5: Adjacency matrix for graph relational learning results (SWAT). (a). Average learned structure by using training data and SWaT process diagram. (b). Learned graph on testing data without anomaly. (c). Learned graph on testing data with anomaly.

missing alarm. Also, our model achieves balanced performance on all these datasets, which illustrates the model robustness.

Unsatisfactory performance in other reconstructed-based models (*LSTM-VAE, MAD-GAN, OmniAnomaly, USAD*) highlights our superiority to involve explicit dependence relationship inference in data reconstruction. As to prediction-based *GDN*, it uses node embedding and GAT to model node correlation, and improves the prediction performance. Yet it has intrinsic limitation due to only using naive prediction error to be anomaly detection metric. For our GReLeN models, though *GReLeN_Loss* could detect potential anomaly, its global loss based anomaly score is too coarse to detect minor anomaly. In contrast, our designed dependence relationship structure based anomaly score (*GReLeN_Degree*) is more powerful to describe and capture system anomaly.

4.6 Effect of Graph Relational Learning

We further demonstrate the superiority of our dependence relationship learning strategy compared with previous similar work *GDN* [Deng and Hooi, 2021]. They claim to use explicit graph to capture spatial correlation, but only "top-k" neighbors have be retained. Consequently its GAT can only provide dynamic weights to the retained neighbors. In this way, if two nodes' dependence relationship is not that intense and get discarded while training, they cannot provide further information in testing phase.

In contrast, our proposed GReLeN will retain all the potential neighbors, and use a proper prior to help control the sparsity of learned graph in a data-driven way. Hence it can avoid the information loss occurred in "top-k" sparsity.

We validate the above discussions by a comparison, where we retain only "top-k" neighbors for GReLeN. The sparsity parameter k is $15, 30, 10, 8$, which is consistent with the settings in *GDN* experiments. As shown in Table 3, comparing *GReLeN_Degree* and *GReLeN_Topk*, a better F1 and Recall performance comes from our *GReLeN_Degree* model.

Meanwhile, as mentioned in Section 3.5, we use the Gumbel-Softmax categorical reparameterization trick to generate the discrete latent variables. Fig.4 shows the reparameterized probability of the dependence relationship not belonging to "null structure". The middle figure of Fig.4 shows the expected results of the prior, while the other two plots show the latent variables learned in abnormal and normal cases. We find that the anomaly could be detected by comparing the latent variables. First, since in the normal case, we assume all the dependence relationships of nodes are sparse, hence we can see besides the "null structure", all the other learnt graph structures are quite sparse. However, in the abnormal case, we can observe that the dependence structure of nodes is no longer sparse due to the right tail in the top figure, which means the system with anomaly tends to have more intensive dependencies.

4.7 Model Interpretability and Anomaly Diagnosis

The dependence structure we learned by latent variable makes our proposed model have better interpretability. In this subsection, we discuss the relational learning and anomaly diagnosis ability.

Fig.5(a) shows the relational graph adjacency matrix learned on SWaT training data, which could be partly interpreted by human knowledge. Three types of graphs are involved to capture different views of dependence relationships. According to the second type of graph, one may con-

clude the first four sensors have strong dependence with other sensors. Referring to the process diagram of this water treatment system [Goh *et al.*, 2016], the first four sensors consist of the first treatment process phase **P1** and should be considered as the fundamental elements of system, which leads to much closer relationship with other sensors. The third type of graph indicates that a small group of sensors have strong dependence among each other. In fact, the sensors circled in purple belong to the same treatment process phase **P3**. It is also acceptable that these sensors have strong dependence because of their same process phase. In Fig.5(b), the system in normal condition shows similar graph pattern to what we have learned in the training phase, while anomaly may lead to a totally different structure (Fig.5(c)).

Our model is also helpful for model diagnosis. A closer look at Fig.5(c) finds two significant abnormal hubs as highlighted by rectangles. The abnormal hub in *block1* shows an actuator has an anomalous setting, which has been recorded in the operation log. The abnormal hubs in *block2* lies in process stage **P5**, i.e., the downstream process of the attacked point in *block1*. It indicates that our method can not only detect the recorded anomaly, but could also report the relative anomalies happening on downstream process beyond the operation log.

Fig.6 shows a case study for WADI. The WADI system has three sub-processes. The sub-process that each sensor belongs to is marked by the first number of the sensor's name. The two most suspicious sensors, whose anomaly scores are exhibited in Fig.6(c)(d), are consistent with the ground-truth operation log. The dependence graph learned in normal condition (Fig.6(a)) has sparse connections for sensors. But in the anomalous state, the attacked sensors(shown in orange color) have much intenser connections with other sensors in the same sub-process, which is consistent with our domain knowledge.

The above case studies show our model is powerful to diagnose anomaly, and can provide crucial information and guidance for daily operation.

5 Conclusion

In our work, we proposed to detect anomaly in multivariate time series from the perspective of graph relational learning. Our Graph Relational Learning Network (GReLeN) model combines VAE structure as well as a graph dependence structure learning strategy for anomaly detection in multivariate time series in a reconstructed way. The latent variable in VAE captures the dependence relationships between sensors explicitly, based on which a well-designed anomaly score is constructed. The experiments show our superiority over both prediction-based and reconstruction-based methods. The interpretability helps us with anomaly diagnosis and provides in-time guidance to daily operation.

Acknowledgements

This work was supported by the RGC GRF 16216119, Foshan HKUST Projects FSUST20-FYTRI03B, in part by the NSFC Grant 71901131, 71932006 and in part by the BNSF Grant 9222014.

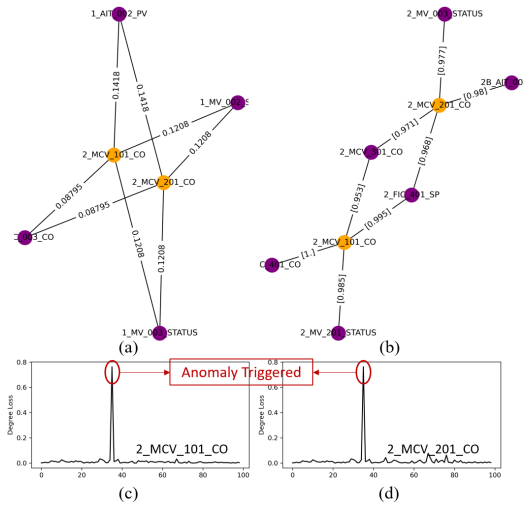

Figure 6: Case study. (a)(b): The graph neighboring structure of the attacked sensors. (c)(d): The anomaly score of attacked sensors.

References

[Abdulaal *et al.*, 2021] 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*, KDD '21, page 2485–2494, New York, NY, USA, 2021. Association for Computing Machinery.

[Ahmed *et al.*, 2017] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, pages 25–28, 2017.

[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.

[Das *et al.*, 2016] Shubhomoy Das, Weng-Keen Wong, Thomas Dietterich, Alan Fern, and Andrew Emmott. Incorporating expert feedback into active anomaly discovery. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 853–858. IEEE, 2016.

[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.

[Goh *et al.*, 2016] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In *International conference on critical information infrastructures security*, pages 88–99. Springer, 2016.

[Jang *et al.*, 2017] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Kipf *et al.*, 2018] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018.

[Li *et al.*, 2018] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[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.

[Liu *et al.*, 2008] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[Park *et al.*, 2018] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

[Shang *et al.*, 2020] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2020.

[Su *et al.*, 2019] 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*, pages 2828–2837, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[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.

[Zhao *et al.*, 2020] 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)*, pages 841–850. IEEE, 2020.