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

Weiqi Zhang¹,², Chen Zhang³ *, Fugee Tsung¹,²
¹The Hong Kong University of Science and Technology (Guangzhou)
²The Hong Kong University of Science and Technology
³Tsinghua 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.

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-
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 spatiotemporal 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 dependence and temporal dependence while predicting. Well designed modules could achieve more accurate prediction, and consequently be helpful to detect abnormal ones. Recent work [Abdulaal et al., 2021] combines feature-oriented Graph Attention Network (GAT) and time-oriented GAT to handle spatial dependence and temporal dependence while predicting.

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.

[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 \( \mathbf{X} \in \mathbb{R}^{N \times T} \). We denote the data collected at each sensor \( i \) and each timestamp \( t \) by using subscript and superscript respectively. That is, \( X_i^t = X_{i:t} \). In our unsupervised assumption, the system operates in normal condition during the first \( T_{\text{train}} \) timestamps, which could be regarded as training data. Our task here is to identify potential anomaly in the following timestamps \( t > T_{\text{train}} \). All the data coming after \( T_{\text{train}} \) 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: \( y^t = f(S^t), y_i^t \in \{0, 1\}, 1 \leq t \leq T_{\text{test}} \). 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.

(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 $q_\phi$ that the latent dependence relationship $Z^t|S^t$ follows. In particular, we can denote that $q_\phi$ is parameterized by $\Theta^t = \{\theta^t_{i,j}, i, j = 1, \ldots, N\}$.

We first use a linear layer to extract temporal feature $H^t$ from the raw input series, where $W_i \in R^{T \times S_i}$ is the learnable parameters and $h_1$ is the dimension of high-level extracted temporal features: $H^t = S^t W_i$.

Self-attention-based mechanism would be applied to calculate $\Theta^t$ based on $H^t = \{h^t_i, i = 1, \ldots, N\}$ and $Q^t K^t = [\theta^t_{i,j}]_{1 \leq i,j \leq N}$.

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 $\Theta^t$ as follows:

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

Here, each pair-wise parameter $\theta^t_{i,j}$ 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 $\theta^t_{i,j}$ 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.

### Table 1: Notations

<table>
<thead>
<tr>
<th>Indices</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N, T$</td>
<td>Number of sensors and timestamps in collected data</td>
</tr>
<tr>
<td>$X_i, X^t$</td>
<td>Data collected at sensor $i$ and timestamp $t$</td>
</tr>
<tr>
<td>$w$</td>
<td>Length of sliding window to generate sample</td>
</tr>
<tr>
<td>$S^t$</td>
<td>Sliding windowed sample data at timestamp $t$</td>
</tr>
<tr>
<td>$h_0$</td>
<td>Number of types of dependence relationship we hope to learn</td>
</tr>
<tr>
<td>$\Theta^t$</td>
<td>Probability parameters for dependence relationship</td>
</tr>
<tr>
<td>$Z^t$</td>
<td>Practical dependence relationship</td>
</tr>
<tr>
<td>$\mu_i, \sigma$</td>
<td>Parameters of Gaussian distribution while reconstructing</td>
</tr>
<tr>
<td>$\tilde{p}_k, \hat{q}_k$</td>
<td>Learned and prior probability to fall into dependence type $k$</td>
</tr>
</tbody>
</table>

(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.

---

*Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*

2392
3.5 Sampling

The above encoder formulates the distribution \( q_\psi \) to generate latent variable \( z_{ij}^t \in R^{h_0} \), which is supposed to be a one-hot vector indicating which type of dependence it belongs to. However, the discretization of \( z_{ij}^t \) 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

\[
z_{ij}^t = \text{softmax} \left( \frac{\theta_{ij}^t + g}{\tau} \right)
\]

where \( g \) 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 \( z_{ij}^t \) is an \( h_0 \) dimension vector. Furthermore, \( z^t \in R^{h_0 \times N^2} \) 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:

\[
p_\psi \left( X_{t+1}^t | X_t, \ldots, X_{t-w+1}, Z^t \right) = \mathcal{N} \left( \mu_{t+1}^t, \sigma^2 I \right)
\]

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 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:

\[
R_t = \text{sigmoid} \left( W_R \ast A \left[ X_t \| H_g^{t-1} \right] + b_R \right);
\]

\[
C_t = \tanh \left( W_C \ast A \left[ X_t \| H_g^{t-1} \right] + b_C \right);
\]

\[
U_t = \text{sigmoid} \left( W_U \ast A \left[ X_t \| H_g^{t-1} \right] + b_U \right);
\]

\[
H_g^t = U_t \circ H_g^{t-1} + \left( 1 - U_t \right) \circ C_t
\]

where the graph convolution operation \( \ast A \) is defined as:

\[
W_Q \ast A Y = \sum_{k=0}^K \left( w_{Q,k}^O \left( D_O^{-1} A \right)^k + w_{Q,k}^Q \left( D_I^{-1} A^T \right)^k \right) Y
\]

Here, \( D_O \) and \( D_I \) are the out-degree and in-degree matrix of graph. \( \| \) denotes concatenation. For \( Q = R, U, C \), \( w_{Q,1}^O, w_{Q,2}^Q \) are learnable parameters and \( K \) is the hyperparameter representing 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:

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

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

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

\[
p_\psi \left( S^t | Z^t \right) = \prod_{t'=t-w+1}^{t-1} p_\psi \left( X_{t'+1}^t | X_{t'}, \ldots, X_{t'-w+1}, Z^t \right)
\]

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

\[
L_1 = -\sum_{i=1}^N \sum_{t'=t-w+1}^{t-1} \frac{|| x_{i,t'}^t - \mu_{i,t'}^t ||^2}{2\sigma^2}
\]

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 \( \tilde{q} = (q_1, q_2, \ldots, q_{h_0}) \). Mathematically, the KL divergence loss could be computed by

\[
L_2 = \text{KL} \left( q_\phi \left( Z^t | S^t \right) \| p_\psi \left( Z^t \right) \right) = \sum_{k=1}^{h_0} \tilde{p}_k \log \left( \tilde{p}_k / q_k \right)
\]

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

4 Experiments

4.1 Datasets

<table>
<thead>
<tr>
<th>Datasets Name</th>
<th>SWaT</th>
<th>WADI</th>
<th>SMD</th>
<th>PSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training size</td>
<td>49990</td>
<td>1084371</td>
<td>708397</td>
<td>139764</td>
</tr>
<tr>
<td>Testing size</td>
<td>499919</td>
<td>172801</td>
<td>708393</td>
<td>87851</td>
</tr>
<tr>
<td>Number of Sensors</td>
<td>51</td>
<td>112</td>
<td>38</td>
<td>26</td>
</tr>
<tr>
<td>Number of Attacks</td>
<td>41 (36)</td>
<td>15</td>
<td>327</td>
<td>72</td>
</tr>
<tr>
<td>Anomaly Durations</td>
<td>100 ~ 34208</td>
<td>87 ~ 1740</td>
<td>2~3161</td>
<td>1~8861</td>
</tr>
<tr>
<td>Anomaly rate(%)</td>
<td>11.97</td>
<td>5.99</td>
<td>4.16</td>
<td>27.76</td>
</tr>
</tbody>
</table>

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

We conduct our proposed model on four real-world datasets: SWaT (Secure Water Treatment Testbed) [Goh et
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]; **OmnimAnomaly** [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 $1 \times 10^{-3}$. 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 $L_2$ here (denoted by $KL_{Loss}$). By experiments we find that $L_2$ is more stable and comprehensive than reconstruction loss $L_1$.

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, $GReLeN\_Loss = L_2$, $GReLeN\_Degree = \sum_{i=1}^{N} d_{i}^{in} + d_{i}^{out}$. Here, $d_{i}^{in}$ and $d_{i}^{out}$ 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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>SWAT</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>SMD</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>PSM</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>98.0±0.11</td>
<td>61.3±0.24</td>
<td>77.5±0.19</td>
<td>90.1±0.34</td>
<td>40.1±0.44</td>
<td>55.3±0.47</td>
<td>99.8±0.91</td>
<td>40.9±0.36</td>
<td>58.1±0.66</td>
<td>92.1±0.104</td>
<td>41.5±0.50</td>
<td>55.5±0.40</td>
</tr>
<tr>
<td>IF</td>
<td>80.3±0.17</td>
<td>35.6±0.27</td>
<td>49.4±0.25</td>
<td>93.0±0.24</td>
<td>24.1±0.19</td>
<td>38.0±0.18</td>
<td>97.7±0.36</td>
<td>47.6±0.63</td>
<td>64.5±0.59</td>
<td>75.9±0.81</td>
<td>57.3±0.86</td>
<td>65.3±0.85</td>
</tr>
<tr>
<td>LSTM-VAE</td>
<td>96.1±0.83</td>
<td>59.3±0.91</td>
<td>73.4±0.74</td>
<td>87.4±0.63</td>
<td>34.2±0.71</td>
<td>23.5±0.08</td>
<td>87.0±0.94</td>
<td>79.4±0.74</td>
<td>83.0±0.82</td>
<td>81.0±0.58</td>
<td>58.2±0.88</td>
<td>67.7±0.59</td>
</tr>
<tr>
<td>MAD-GAN</td>
<td>97.3±0.56</td>
<td>64.6±0.49</td>
<td>77.7±0.43</td>
<td>41.0±0.60</td>
<td>34.2±0.67</td>
<td>37.3±0.46</td>
<td>17.5±0.43</td>
<td>92.9±0.80</td>
<td>29.5±0.59</td>
<td>43.4±0.83</td>
<td>63.7±0.35</td>
<td>51.6±0.57</td>
</tr>
<tr>
<td>OmniAnomaly</td>
<td>71.7±0.78</td>
<td>96.3±0.76</td>
<td>82.2±0.40</td>
<td>26.9±0.73</td>
<td>98.2±0.86</td>
<td>42.2±0.91</td>
<td>97.8±1.07</td>
<td>94.3±0.70</td>
<td>96.0±0.64</td>
<td>96.0±0.39</td>
<td>80.9±0.68</td>
<td>87.8±0.48</td>
</tr>
<tr>
<td>USAD</td>
<td>97.9±0.20</td>
<td>72.7±0.67</td>
<td>83.4±0.43</td>
<td>64.4±0.68</td>
<td>31.6±0.69</td>
<td>42.3±0.63</td>
<td>93.6±0.46</td>
<td>95.5±0.70</td>
<td>94.6±0.57</td>
<td>92.1±1.07</td>
<td>76.1±1.00</td>
<td>70.9±0.81</td>
</tr>
<tr>
<td>GDN</td>
<td>98.2±0.16</td>
<td>67.3±0.49</td>
<td>79.8±0.27</td>
<td>98.2±0.32</td>
<td>39.9±1.11</td>
<td>56.7±1.10</td>
<td>58.1±0.93</td>
<td>56.6±0.41</td>
<td>57.3±0.54</td>
<td>43.4±1.04</td>
<td>76.0±0.61</td>
<td>55.2±0.81</td>
</tr>
<tr>
<td>RCoders</td>
<td>90.1±0.32</td>
<td>73.8±0.71</td>
<td>80.9±0.53</td>
<td>64.5±0.48</td>
<td>35.5±0.53</td>
<td>44.1±0.53</td>
<td>81.2±0.67</td>
<td>80.0±0.82</td>
<td>80.6±0.61</td>
<td>89.9±0.40</td>
<td>87.3±0.32</td>
<td>92.7±0.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>SWAT</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>SMD</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>PSM</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>GReLeN_Loss</td>
<td>91.0±0.32</td>
<td>80.0±0.14</td>
<td>85.7±0.12</td>
<td>79.3±0.52</td>
<td>48.2±0.91</td>
<td>59.0±0.70</td>
<td>88.2±0.61</td>
<td>83.6±0.64</td>
<td>87.2±0.21</td>
<td>95.8±0.98</td>
<td>77.3±0.49</td>
<td>85.6±0.49</td>
</tr>
<tr>
<td>GReLeN_Topk</td>
<td>95.6±0.10</td>
<td>83.5±0.01</td>
<td>89.1±0.21</td>
<td>77.3±1.43</td>
<td>61.3±0.34</td>
<td>68.2±0.74</td>
<td>92.2±1.01</td>
<td>96.0±0.41</td>
<td>92.6±0.31</td>
<td>91.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

2394
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 (GReLeNDegree) 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 been 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 GReLeNDegree and GReLeN_Topk, a better F1 and Recall performance comes from our GReLeNDegree 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 abnormal 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-FYTR03B, in part by the NSFC Grant 71901131, 71932006 and in part by the BNSF Grant 9222014.

References


