Variational Autoencoder based Automatic Clustering for Multivariate Time Series Anomaly Detection

Li Yan[1], Hailin Hu[1], Kun Yang[1], Gaozhou Wang[1], Ti Guan[2], and Yanwei Zheng[3⋆]

[1] State Grid Shandong Electric Power Company Information and Communication Company, China
yanli@sd.sgcc.com.cn, huhailin@sd.sgcc.com.cn, yangkun@sd.sgcc.com.cn, wanggaozhou@sd.sgcc.com.cn
[2] State Grid Shandong Electric Power Company, China
guanti@sd.sgcc.com.cn
[3] School of Computer Science and Technology, Shandong University, China
zhengyw@sdu.edu.cn

Abstract. Multivariate time series anomaly detection is of great significance in monitoring and ensuring the stable operation of complex systems. The multivariate time series generated in real scenes often have complex dependency patterns, summarized as *temporal dynamics* and *spatial dynamics*. Specifically, temporal dynamics manifested as complex dependencies between values at different timestamps, while spatial dynamics refer to uncertain similarity relationships between different sequences. In order to simultaneously model temporal and spatial dynamics, we propose a variational autoencoder based automatic clustering method for multivariate time series anomaly detection(ACVAE), which maps input sequences to latent representations using VAE and reconstructs input sequences based on the latent representations, while detecting anomalies based on reconstruction errors. Specifically, we design an encoder network that combines TCN and GRU to learn multi-scale long short-term temporal dependencies, and introduce the Dirichlet prior to automatically capture the similarity between sequences. Finally, we conduct extensive experiments on two publicly available datasets, and the results show that the ACVAE is superior to other baseline methods.

Keywords: Multivariate time series · Anomaly detection · Dirichlet process · Variational autoencoder.

1 Introduction

With the rapid development of IoT technology, more and more sensors are deployed in industrial environments, generating a large amount of real-time monitoring data, making multivariate time series anomaly detection an indispensable

⋆ Corresponding author.

part of intelligent industrial systems[1]. Considering the uncertainty of anomalous patterns and the difficulty in obtaining labels, the recently popular idea of anomaly detection approaches is to train the model on normal data to learn the normal patterns, and then identify the data deviating from the normal patterns as anomalies on the testing set [2]. In order to learn normal patterns, we need to focus on the complex *temporal dynamics* and *spatial dynamics* in multivariate time series, which are the two major challenges of multivariate time series modeling.

Temporal dynamics refers to the complex nonlinear long short-term temporal dependencies in time series. In addition, time series have stochasticity, namely the presence of noise and uncertainty fluctuations. The CNN based methods, such as DeepAnt[3], can effectively extract short-term features. Some TCN based works such as HS-TCN[4] and TCN-GMM[5] can model long-term dependencies. In addition, LSTM based EncDec-AD[6], Telemanom[7] and GRU based OmniAnomaly[8], GGM-VAE[9] have been successively proposed. The attention mechanism based MTAD-GAT[10] and Transformer based GTA[11] also have emerged. Moreover, to model stochasticity and learn robust latent representations, some VAE based models have been proposed to simultaneously learn temporal dependence and stochasticity, such as LSTM-VAE[12], and OmniAnomaly[8].

Spatial dynamics refers to the similarity relationships between sequences, and the number of clusters is uncertain. It is difficult for us to cluster sequences based on manual experience. Some existing works have designed specific graph learning strategies to model the inter sequence graph relationships. The GDN[13], FuSAGNet[14] and StackVAE-G[15] employ the top-k strategy to construct sensor relationship graphs, while GTA[11] proposes a graph structure learning strategy based on Gumbel-Softmax Sampling to learn the global topology among all nodes. The MTAD-GAT[10] designs a feature-oriented graph attention network to model inter sequence relationships.

In summary, previous works either utilize deep neural networks to model temporal dependencies or design graph learning strategies to model the inter sequence graph relationships. However, there lacks a method to simultaneously model multi-scale long short-term temporal dependencies and stochasticity, while considering the uncertain similarity relationship between sequences and achieving automatic clustering. Therefore, we propose a variational autoencoder based automatic clustering method to address the *temporal dynamics* and *spatial dynamics* simultaneously. Overall, this is a VAE framework that maps input sequences to latent spaces and reconstructs inputs based on latent representations, while using reconstruction errors as anomaly scores to determine anomalies. Specifically, we design a TCN-GRU network as the encoder for VAE to learn multi-scale features and long short-term dependencies of time series. In order to model spatial dynamics and automatically cluster similarity sequences, we introduce the Dirichlet prior, which allows the model to dynamically adjust the number of categories based on data without specifying the number of categories K in advance. Our main contributions can be summarized as follows:

– We propose a variational autoencoder based automatic clustering method for multivariate time series anomaly detection to address the *temporal dynamics* and *spatial dynamics* simultaneously.
– For each sequence, we use a variational autoencoder to map the input sequence to a latent representation and reconstruct the input accordingly. Specifically, we design a TCN-GRU as the encoder network to model multi-scale long short-term temporal dependencies patterns.
– We apply a Dirichlet process prior on the latent representations of all sequences, which aims to model the similarity relationship between sequences and achieve clustering automatically.
– We conduct extensive experiments on two publicly available datasets to evaluate our proposed model, and the results show that the performance of our model is significantly better than the baseline methods.

2 Related Work

In existing deep learning based multivariate time series anomaly detection algorithms, modeling temporal dependencies and spatial dependencies are the two focuses of modeling multivariate time series patterns.

To model temporal dependence and stochasticity, researchers have proposed some deep learning methods based on CNN, TCN, LSTM, GRU, or Transformer[16]. DeepAnt[3] uses a two-layer CNN to model temporal dependencies within input time window. HS-TCN[4] proposes a semi-supervised layered stacking TCN to handle the anomaly detection problem. EncDec-AD[6], Telemanom[7], S-RNNs[17] and LSTM-AD[18] are all prediction frameworks based on LSTM, especially LSTM-AD which designs stacked LSTM to learn temporal dependencies. The GGM-VAE[9] employs GRU within the VAE to discover the dependencies and stochasticity among the time series data. GTA[11] proposes a new multi-branch attention mechanism to simultaneously consider local and global context temporal dependencies.

To model spatial dependencies between sequences, some works have designed specific graph learning strategies. GDN[13] is a graph deviation network that defines an embedding for each sensor sequence and uses the similarity between embeddings as the basis for selecting top-k neighbors to construct a sensor relationship graph. FuSAGNet[14] also uses the top-k strategy to learn graph structures and combines sparse autoencoders and GNN to achieve both reconstruction and forecasting. MTAD-GAT[10] proposes a feature oriented graph attention method to learn inter sequence dependencies.

3 Model Architecture

3.1 Problem Statement

Given a dataset $\mathcal{D} = \{\{x_n^t\}|_{n=1}^N\}|_{t=1}^T$, where N represents the number of sequences and T is the total length of time series. We define $x_n = \{x_n^{t_1}, \ldots, x_n^{t_s}\}$ as an input sequence of length s, and $x^{t_{s^*}} = \{x_1^{t_{s^*}}, \ldots, x_N^{t_{s^*}}\}$ represents the feature vector at timestamp t_{s^*}. In reconstruction based models, the training objective is to learn a function that can correctly reconstruct inputs on the normal dataset D_{train}, denoted as $f(x_n) \to \hat{x}_n$. The aim of anomaly detection is to find out whether an anomaly has occurred at a certain moment according to whether the reconstruction error surpasses the threshold on the dataset D_{test}.

3.2 Model Overview

To model the temporal and spatial dynamics simultaneously in multivariate time series and further achieve accurate anomaly detection performance, we propose a variational autoencoder based automatic clustering framework. Overall, the proposed framework is a VAE based reconstruction framework and detects anomalies through the reconstruction errors. Specifically, our model mainly consists of three parts: encoder, clustering part, and decoder. The TCN-GRU based encoder embeds the input x_n of each sequence into a latent representation z_n. The clustering part uses a Dirichlet prior to guide latent representations to achieve automatic clustering and model the similarity between sequences. In addition,

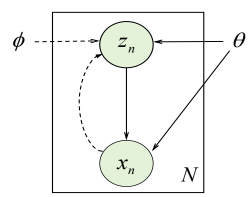

Fig. 1. Graphical model representations. The dashed lines mean the encoder whose parameter is ϕ. The solid lines denote the decoder whose parameter is θ.

the graph model representation of our ACVAE is shown in Fig. 1. According to VAE theory [19], the ELBO function of the n-th sequence is:

$$\begin{aligned} &\log p_\theta(x_n) \geq E_{q_\phi(z_n|x_n)}[\log\{\tfrac{p_\theta(x_n,z_n)}{q_\phi(z_n|x_n)}\}] \\ &= E_{q_\phi(z_n|x_n)}[\log p_\theta(x_n|z_n)] - D_{KL}(q_\phi(z_n|x_n)||p_\theta(z_n)) = \mathcal{L}(\theta,\phi) \end{aligned} \tag{1}$$

where the first item characterizes the model's ability to reconstruct the input series. The second item is the Kullback-Leibler divergence between the approximate posterior $q_\phi(z_n|x_n)$ and prior distributions $p_\theta(z_n)$. The parameters of $q_\phi(z_n|x_n)$ and $p_\theta(x_n|z_n)$ are learned with the encoder and decoder.

3.3 Encoder

To model the multi-scale long short-term temporal dependencies in the input time series, we propose a TCN-GRU based encoder. We assume that $q_\phi(z_n|x_n)$

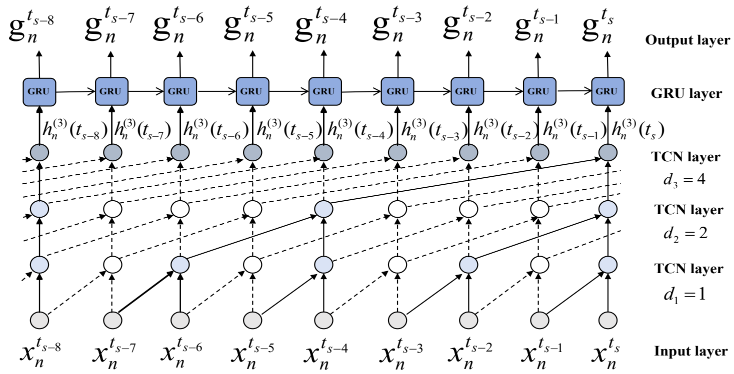

Fig. 2. The details of TCN-GRU network.

follows a Gaussian distribution and use the encoder to learn the parameters $[\mu_n, \sigma_n]$. As shown in Fig. 2, we adopt a three-layer TCN structure with dilation of 1, 2, 4, respectively. The output of time t_{s^*} in the l-th layer is:

$$h_n^{(l)}(t_{s^*}) = f_{d_l}^{(l)}(h_n^{(l-1)}) = \sum_{i=0}^{K_c-1} w_i^{(l)} \cdot h_n^{(l-1)}(t_{s^*} - d_l \cdot i) \tag{2}$$

where $l = 1, 2, ..., L$ and we set $L = 3$. The $f_{d_l}^{(l)}$ represents the dilation convolution operation of layer l, with a dilation rate of d_l. The K_c is the size of the convolution kernel. Then, the outputs of the third layer TCN are fed into the GRU unit to further model the sequential long-term dynamics within time series data. And we pass the output of the last timestamp in GRU through two linear layers to obtain μ_n and σ_n, respectively:

$$\left[\mu_n, \log \sigma_n^2\right] = Linear(\mathcal{F}_{\text{GRU}}(h_n^{(3)}(t_s), g_n^{t_{s-1}}) \tag{3}$$

where $\mathcal{F}_{\text{GRU}}$ is the GRU cell, and $g_n^{t_{s-1}}$ is the output of previous GRU cell. Finally, the latent representation z_n was obtained through reparameterization techniques[19]: $z_n = \mu_n + \sigma_n \odot \rho, \rho \sim N(0, 1)$.

3.4 Clustering

To capture the similarity between sequences and achieve automatic clustering, we introduce a stick-breaking Dirichlet process on the prior distribution of z. The details are shown in Fig. 3.

First, we draw c_k from the Beta distribution $Beta(1, \eta)$, and calculate $\pi_k = c_k \prod_{i=1}^{k-1} (1 - c_i)$. And we sample $\tilde{\mu}$ from the Gaussian distribution $N(\overline{\mu}, \overline{\sigma})$, whose parameters are constant. Then, we sample ξ_k from $N(\tilde{\mu}, \tilde{\sigma})$, and all clusters share the $\tilde{\sigma}$. For each sequence, we use an indicator vector y_n to indicate which cluster

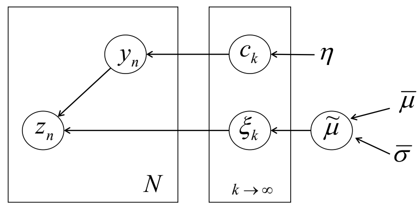

Fig. 3. The generative process of prior distribution of z.

it belongs to and sample y_n from $Mult(1; \pi_1, \cdots \pi_\infty)$. Finally, we draw the z_n from $N(\xi_{y_n}, \sigma_c^2 I)$, where σ_c is the variance parameter shared by all clusters.

We estimate the stick-breaking Dirichlet process prior[20] parameters with the variational inference[21]. Consistent with literature [22, 23], we consider the stick-breaking process to have a truncation level of K. And we have $q(c, \xi, y) = \prod_{k=1}^{K} q(c_k) \prod_{k=1}^{K} q(\xi_k) \prod_{n=1}^{N} q(y_n)$, where $q(c_k) \sim Beta(\gamma_{1k}, \gamma_{2k})$. According to literature[20], the $(\gamma_{1k}, \gamma_{2k})$ can be calculated by:

$$\gamma_{1k} = 1 + \sum_{n=1}^{N} \widetilde{y_n^k}, \quad \gamma_{2k} = \eta + \sum_{n=1}^{N} \sum_{j=k+1}^{K} \widetilde{y_n^j}. \tag{4}$$

For each cluster k, we have $q(\xi_k) \sim N(\mu_k, \sigma_k^2 I)$ and the (μ_k, σ_k) can be calculated by:

$$\frac{\mu_k}{{\sigma_k}^2} = \frac{\tilde{\mu}}{\tilde{\sigma}^2} + \frac{\sum_{n=1}^{N} \widetilde{y_n^k} z_n}{\sigma_c^2}, \quad \frac{1}{{\sigma_k}^2} = \frac{1}{\tilde{\sigma}^2} + \frac{\sum_{n=1}^{N} \widetilde{y_n^k}}{\tilde{\sigma}^2}. \tag{5}$$

For latent representation z_n, the cluster indicator variable y_n can be distributed as $q(y_n) \sim Mult(\widetilde{y_n})$ and $\widetilde{y_n}$ can be calculated by:

$$\begin{aligned} \widetilde{y_n^i} = \exp\{&\Psi(\gamma_{1i}) - \Psi(\gamma_{1i} + \gamma_{2i}) + \sum_{j=1}^{i-1} [\Psi(\gamma_{2j}) - \Psi(\gamma_{2j} + \gamma_{2j})] \\ &- \frac{1}{2\sigma_c^2} [(z_n - \mu_i)^T (z_n - \mu_i) + \sigma_i^2]\}, \end{aligned} \tag{6}$$

where $\Psi(\cdot)$ is the digmma function.

3.5 Decoder

To reconstruct the input series based on the learned latent representation z_n, we use a GRU based decoder to model $p(x_n|z_n)$. To address the cumulative errors problem, we use the scheduled sampling strategy[24] to introduce ground-truth values as the input for the next step during the training phase. Specifically, the input of each timestamp t_{s^*} is the connection of $(z_n, \tilde{v}_n^{t_{s^*}})$, where $\tilde{v}_n^{t_{s^*}}$ can be represented as:

$$\tilde{v}_n^{t_{s^*}} = \begin{cases} x_n^{t_{s^*-1}}, & p(m) > rand(0,1) \\ \hat{x}_n^{t_{s^*-1}}, & otherwise \end{cases} \tag{7}$$

where $\hat{x}_n^{t_{s^*-1}}$ is the reconstructed output , and $x_n^{t_{s^*-1}}$ is the ground-truth value. $p(m) = \frac{1}{1+\alpha e^{\alpha m}}$ is a inverse sigmoid decay function, in which α is the hyper-parameter and m represents the number of iterations. As the training rounds increases, the value of $p(m)$ decreases. Finally, the output of GRU is processed through a linear layer to obtain the final reconstructed output $\{\hat{x}_n^{t_1}, \hat{x}_n^{t_2}, \cdots, \hat{x}_n^{t_s}\}$.

3.6 Objective Function and Training

For the ACVAE framework, our training goal is to maximize the ELBO function. Considering all sequences, the objective function can be represented as:

$$\mathcal{L}(\theta, \phi) = \sum_{n=1}^{N} E_{q_\phi(z_n|x_n)}[\log p_\theta(x_n|z_n)] - \sum_{n=1}^{N} D_{KL}(q_\phi(z_n|x_n)||p_\theta(z_n)), \tag{8}$$

where ϕ is the parameters of the encoder TCN-GRU network, while θ includes both the variational parameters in Sec. 3.4 and the parameters of the decoder GRU network. We use back-propagation algorithm to update the neural network parameters and use equations Eq. 4, Eq. 6, Eq. 5 to update the stick-breaking variational parameters. And we adopt alternating optimization strategies[25] to jointly learn the above parameters.

3.7 Anomaly Detection

When detecting anomalies, we calculate the mean reconstruction error of all N sequences at timestamp t_{s^*} as the anomaly score:

$$S_a(t_{s^*}) = \frac{1}{N} \sum_{n=1}^{N} \sqrt{\left(\hat{x}_n^{t_{s^*}} - x_n^{t_{s^*}}\right)^2} \tag{9}$$

If $S_a(t_{s^*}) > \tau$, we assume an anomaly has occurred at timestamp t_{s^*}. Consistent with highly relevant works GRELEN[2] and GTA[11], we use the grid search algorithm to find the threshold τ and report the optimal F1.

4 Experiments

4.1 Experimental Setup

Datasets Description We conduct extensive experiments on two public multivariate time series datasets: SWaT, SMD. The SWaT dataset is obtained from 51 sensors and actuators within 11 days at a frequency of 1 second. The SMD dataset is gathered from 28 different server equipment, each of which has 38

Table 1. Experimental results of different approaches on SWaT and SMD using Precision, Recall, F1 and ROC-AUC as metrics. The best performances are highlighted in bold and the second best are underlined.

Datasets	SWaT				SMD			
Method/Metric	**Pre.**	**Rec.**	**F1**	**AUC**	**Pre.**	**Rec.**	**F1**	**AUC**
Telemanom	<u>0.9575</u>	0.6422	0.7687	0.7814	0.6242	0.6534	0.5781	0.7563
LSTM-AD	0.8716	0.7329	0.7963	0.8186	0.8482	0.8547	0.8307	0.7351
GDN	**0.9721**	0.6907	0.8076	0.8266	<u>0.8503</u>	<u>0.9104</u>	<u>0.8757</u>	**0.8655**
MTAD-GAT	0.8721	<u>0.7984</u>	<u>0.8337</u>	<u>0.8346</u>	0.7873	0.8851	0.7736	0.8021
ACVAE(Ours)	0.9223	**0.8058**	**0.8601**	**0.8576**	**0.8545**	**0.9556**	**0.8881**	<u>0.8112</u>

indicators. It is worth noting that we perform max-min normalization preprocessing on all datasets. And we use the normal data portion as the training set and the data containing anomalies as the testing set. In addition, we randomly divide 20% of the training set as the validation set for model selection.

Baselines. We compare the anomaly detection performance of our ACVAE with four popular anomaly detection methods:

- **Telemanom[7]:** The Telemanom utilizes LSTM for one-step prediction and uses prediction error to detect anomalies.
- **LSTM-AD[18]:** The LSTM-AD is a forecasting based anomaly detection approach that designs stacked LSTMs to learn the temporal dependence.
- **GDN [13]:** The GDN selects K sequences with the highest similarity between embeddings as neighbors to represent the relationships among sensors.
- **MTAD-GAT [10]:** The MTAD-GAT designs time oriented and feature oriented GAT to model temporal dependencies and inter sequence relationships.

Metrics. We take the Precision, Recall, F1 and ROC-AUC as evaluation metrics. The higher the F1 and AUC, the better the model performance.

Training Settings. We conduct all experiments on six NVIDIA GeForce RTX 3090 GPUs and use Pytorch version 1.10.0 with CUDA 11.1. We use the Adam optimizer and set the learning rate to 0.001. In addition, an early stop method with patience 5 is adopted. The batch size is 128 and 512 on the SWaT and SMD datasets, respectively. The size of the input window is 30, and the size of the latent variable z_n is 5. We use a three-layer TCN network with expansion rates of 1, 2, and 4, and set the convolutional kernel size to 3. The hidden layer dimension of GRU network is 64.

4.2 Performance and Analysis

We use precision, recall, F1, and AUC as metrics to compare the performance of ACVAE with the other four baselines on the SWaT and SMD datasets. The results are shown in Table 1. It can be seen that our model performs the best in recall, F1, and AUC metrics. Compared with the suboptimal model, ACVAE improves recall, F1, and AUC by 0.93%, 3.17%, 2.76% on the SWaT dataset and improves precision, recall, F1 by 0.49%, 4.96%, 1.42% on the SMD dataset,

Table 2. Experimental results of ablation studies on SWaT and SMD using Precision, Recall, F1 and ROC-AUC as metrics. The ACVAE without removal of any components achieves the best performance.

Datasets	SWaT				SMD			
Method/Metric	**Pre.**	**Rec.**	**F1**	**AUC**	**Pre.**	**Rec.**	**F1**	**AUC**
ACVAE(Ours)	0.9223	**0.8058**	**0.8601**	**0.8576**	0.8545	**0.9556**	**0.8881**	**0.8112**
ACVAE_*w/o DP prior*	0.9347	0.7795	0.8501	0.8502	0.8636	0.9392	0.8768	0.8070
ACVAE_*w/o TCN*	0.9070	0.8028	0.8517	0.8506	0.8713	0.9002	0.8705	0.8016

respectively. The superior performance of ACVAE demonstrates the necessity of modeling both temporal and spatial dynamics simultaneously. In addition, models that consider both inter sequence relationships and temporal dependencies, such as ACVAE, MTAD-GAT and GDN perform significantly better than models that only consider temporal dependencies such as LSTM-AD and Telemanom.

4.3 Ablation Studies

To verify the effectiveness of each module of the ACVAE, we evaluate the performance of the following variants:

- ACVAE_*w/o DP prior*. To study the superiority of the Dirichlet process prior, we replace the prior with the standard normal distribution commonly used in traditional VAE frameworks.
- ACVAE_*w/o TCN*. To study the effect of proposed TCN-GRU, we replace TCN-GRU with traditional GRU in the encoder.

As shown in Table 2, the F1 and AUC of ACVAE_*w/o DP prior* decreases 1.16%, 0.86% on SWaT dataset and 1.27%, 0.52% on SMD dataset, which demonstrates the superiority of the Dirichlet prior over the ordinary Gaussian prior in the anomaly detection task, as well as the necessity of modeling the similarity between sequences. The F1 and AUC of ACVAE_*w/o DP TCN* decreases 0.98%, 0.82% on SWaT dataset and 1.98%, 1.18% on SMD dataset, which indicates that our proposed TCN-GRU is superior in modeling multi-scale long short-term temporal dependencies.

4.4 Case Study

To demonstrate the superiority of our ACVAE in anomaly detection, we visualize the anomaly labels of the total SWaT dataset, as well as the anomaly scores and thresholds of our model and other baseline methods. As shown in Fig. 4, we mainly select three representative regions for analysis. The $R1$ and $R2$ are regions with anomalies to verify the model's anomaly detection performance, while $R3$ is an area without anomalies to study the model's robustness. From the Fig. 4, in region $R1$, all methods except ACVAE missed detecting anomalies, which proves the superior anomaly detection performance of our model. In region

$R2$, only ACVAE and GDN correctly detect anomalies, further demonstrating the importance of considering spatial dependencies. In region $R3$, MTAD-GAT, GDN, and LSTM-AD all experience false positives, while our model did not experience any false positives. This proves that our ACVAE can better distinguish normal and anomalous data.

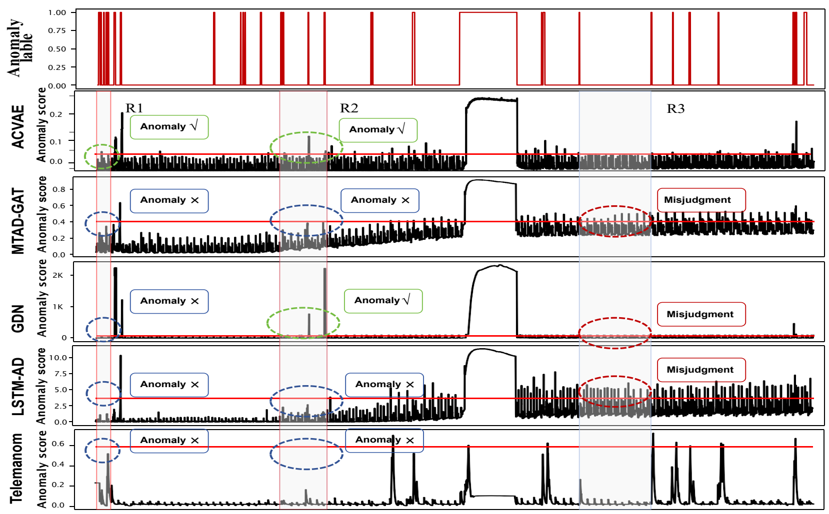

Fig. 4. Case study on SWaT dataset. The green elliptical box represents correctly detected anomalies, blue represents missed detections, and red represents false detections.

5 Conclusion

In this paper, we proposed a variational autoencoder based automatic clustering method for multivariate time series anomaly detection to address the temporal dynamics and spatial dynamics simultaneously. In the ACVAE framwork, we designed TCN-GRU as the encoder network to model multi-scale long short-term time-dependent and mapped the input time series to latent representations. In the decoder, we used GRU to reconstruct the input and adopted the scheduled sampling strategy to address cumulative errors. Specifically, in order to model the similarity between sequences, we applied a Dirichlet process prior on the latent representations of all sequences, which can achieve automatic clustering. The experimental results on two public datasets show the superiority of our ACVAE over other baseline methods.

Acknowledgments. This paper is supported by the Technology project of State Grid Shandong Electric Power Company (No. 2023A-038).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Leon, R.A., Vittal, V., Manimaran, G.: Application of sensor network for secure electric energy infrastructure. IEEE Transactions on Power Delivery **22**(2), 1021–1028 (2007)
2. Zhang, W., Zhang, C., Tsung, F.: 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. pp. 2390–2397
3. Munir, M., Siddiqui, S.A., Dengel, A., Ahmed, S.: Deepant: A deep learning approach for unsupervised anomaly detection in time series. Ieee Access **7**, 1991–2005 (2018)
4. Cheng, Y., Xu, Y., Zhong, H., Liu, Y.: Hs-tcn: A semi-supervised hierarchical stacking temporal convolutional network for anomaly detection in iot. In: 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC). pp. 1–7. IEEE (2019)
5. Liu, J., Zhu, H., Liu, Y., Wu, H., Lan, Y., Zhang, X.: Anomaly detection for time series using temporal convolutional networks and gaussian mixture model. In: Journal of Physics: Conference Series. vol. 1187, p. 042111. IOP Publishing (2019)
6. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: Lstm-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148 (2016)
7. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 387–395 (2018)
8. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2828–2837 (2019)
9. Guo, Y., Liao, W., Wang, Q., Yu, L., Ji, T., Li, P.: Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach. In: Asian Conference on Machine Learning. pp. 97–112. PMLR (2018)
10. Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., Zhang, Q.: Multivariate time-series anomaly detection via graph attention network. In: 2020 IEEE International Conference on Data Mining (ICDM). pp. 841–850. IEEE (2020)
11. Chen, Z., Chen, D., Zhang, X., Yuan, Z., Cheng, X.: Learning graph structures with transformer for multivariate time-series anomaly detection in iot. IEEE Internet of Things Journal **9**(12), 9179–9189 (2021)
12. Park, D., Hoshi, Y., Kemp, C.C.: A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. IEEE Robotics and Automation Letters **3**(3), 1544–1551 (2018)
13. Deng, A., Hooi, B.: Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4027–4035 (2021)

14. Han, S., Woo, S.S.: 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. pp. 2977–2986 (2022)
15. Li, W., Hu, W., Chen, T., Chen, N., Feng, C.: Stackvae-g: An efficient and interpretable model for time series anomaly detection. AI Open **3**, 101–110 (2022)
16. Choi, K., Yi, J., Park, C., Yoon, S.: Deep learning for anomaly detection in timeseries data: Review, analysis, and guidelines. IEEE Access **PP**(99), 1–1 (2021)
17. Kieu, T., Yang, B., Guo, C., Jensen, C.S.: Outlier detection for time series with recurrent autoencoder ensembles. In: IJCAI. pp. 2725–2732 (2019)
18. Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al.: Long short term memory networks for anomaly detection in time series. In: Esann. vol. 2015, p. 89 (2015)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Zhang, X., Li, W., Nguyen, V., Zhuang, F., Xiong, H., Lu, S.: Label-sensitive task grouping by bayesian nonparametric approach for multi-task multi-label learning. In: IJCAI. pp. 3125–3131 (2018)
21. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning **37**(2), 105–161 (1999)
22. Blei, D.M., Jordan, M.I., et al.: Variational inference for dirichlet process mixtures. Bayesian Analysis **1**(1), 121–143 (2006)
23. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with dirichlet process priors. JMLR **8**(Jan), 35–63 (2007)
24. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. Advances in neural information processing systems **28** (2015)
25. Goyal, P., Hu, Z., Liang, X., Wang, C., Xing, E.P.: Nonparametric variational auto-encoders for hierarchical representation learning. In: ICCV. pp. 5094–5102 (2017)