Switching Gaussian Mixture Variational RNN for Anomaly Detection of Diverse CDN Websites

Liang Dai[†♮♯], Wenchao Chen[‡♯], Yanwei Liu[†*], Antonios Argyriou[§], Chang Liu[†♮], Tao Lin[¶], Penghui Wang[‡], Zhen Xu[†], and Bo Chen[‡]
[†]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[♮]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[‡]National Laboratory of Radar Signal Processing, Xidian University, Xi'an, China
[§]University of Thessaly, Greece [¶]Communication University of China, Beijing, China

***Abstract*—To conduct service quality management of industry devices or Internet infrastructures, various deep learning approaches have been used for extracting the normal patterns of multivariate Key Performance Indicators (KPIs) for unsupervised anomaly detection. However, in the scenario of Content Delivery Networks (CDN), KPIs that belong to diverse websites usually exhibit various structures at different timesteps and show the non-stationary sequential relationship between them, which is extremely difficult for the existing deep learning approaches to characterize and identify anomalies. To address this issue, we propose a switching Gaussian mixture variational recurrent neural network (SGmVRNN) suitable for multivariate CDN KPIs. Specifically, SGmVRNN introduces the variational recurrent structure and assigns its latent variables into a mixture Gaussian distribution to model complex KPI time series and capture the diversely structural and dynamical characteristics within them, while in the next step it incorporates a switching mechanism to characterize these diversities, thus learning richer representations of KPIs. For efficient inference, we develop an upward-downward autoencoding inference method which combines the bottom-up likelihood and up-bottom prior information of the parameters for accurate posterior approximation. Extensive experiments on real-world data show that SGmVRNN significantly outperforms the state-of-the-art approaches according to F1-score on CDN KPIs from diverse websites.**

***Index Terms*—Multivariate Anomaly Detection, CDN, Probabilistic Mixture Model, Variational Recurrent Neural Network, Switching Mechanism.**

I. Introduction

Today's commercial Content Delivery Networks (CDN) typically provide content delivery services for tens of thousands of websites, making it extremely important to monitor and ensure the services of these websites under the constraints specified by the service level agreements (SLA). To this end, CDN operators usually collect various Key Performance Indicators (KPIs) for each website, e.g., traffic volume, delay, and hit ratio, etc., and perform anomaly detection for these multivariate KPIs to detect a service failure or degradation.

Due to its tremendous capability in learning expressive representations of complex data, a recent trend is to utilize deep learning to detect anomalies, called deep anomaly detection, purely from data [1]–[5]. The basic idea of deep anomaly detection is to model the normal patterns of time series, considering that an anomaly or outlier often behaves differently from the normal data. In line with this philosophy, a large number of novel unsupervised methods [2], [3], [5], [6], that use the recurrent neural networks (RNN) for time series feature extraction, have been proposed for multivariate time series anomaly detection.

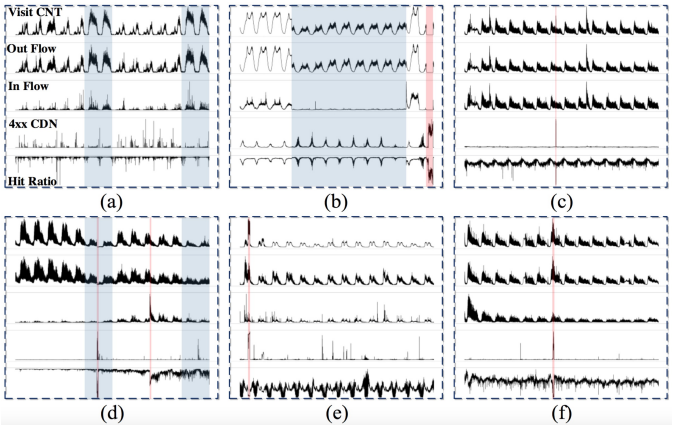


Fig. 1: 2-weeks real world typical multivariate CDN KPIs of 6-websites. Periods in light blue show the change points in KPIs; Regions highlighted in red represent the ground-truth anomaly segments.

Despite the good performance that existing deep anomaly detection methods claim, after conducting an analysis of some typical CDN KPIs collected from a popular ISP-operated (Internet Service Provider) CDN in China, we explored the nature of the KPI data for diverse CDN websites and observed two crucial challenges that the current deep anomaly detection models cannot deal with effectively.

Challenge 1: The non-stationary dependencies across different time periods for one individual website will degrade the performance of the existing deep anomaly detection models. As shown in Fig. 1 (a) and Fig. 1 (d), it is clear that user request behavior on weekdays is different from those on weekends. The former website exhibits a burst of user requests on weekends relative to weekdays, while the latter figure illustrates the opposite. Fig. 1 (b) shows another typical case, i.e., some users are scheduled to a different set of edge

*Yanwei Liu is the corresponding author, ♯equal contribution.

nodes by the scheduling center of the CDN; thus, the moments in time that KPIs change occur during these time periods. In addition, there are even more complex cases in CDNs, that we will examine carefully in our qualitative analysis experiments in Section VI. Hence, the corresponding KPIs usually exhibit non-stationary temporal characteristics due to the normal behaviors of users' or scheduling by CDN, etc., which should not be classified as service failures or degradation. However, these type of expected patterns are difficult to be captured by current methods, which further results in inferior performance of the current methods on anomaly detection for the CDN, as we will illustrate in the quantitative and qualitative analyses in Section VI.

Challenge 2: Websites that are diverse exhibit various characteristics in CDN KPIs, but several of them possess similar characteristics, and this leads to the inability of current deep anomaly detection models to capture this dynamic complexity well, especially in one model. Since commercial CDNs usually provide services for hundreds or even thousands of websites that exhibit varying characteristics due to the service type and the request behaviors of users, variations among the spatial and temporal features of the KPIs that belong to the different websites are observed. For instance, the KPIs of the Video on Demand (VoD) websites in Fig. 1 (a) are usually very different from the Live streaming websites in Fig. 1 (e). To conduct effective anomaly detection for multiple websites, existing deep anomaly detection methods usually train an individual model for each website, thus suffer from the problem of training and maintaining a large number of individual models for each website, which not only consumes huge computing and storage resources, but also raises the costs in model maintenance. In addition, we also observe that KPIs that belong to different websites e.g., in Fig. 1 (c) and Fig. 1 (f) show similar characteristics. In this case, training an individual model for each website is totally wasteful.

Moving beyond the limitations of previous work in managing two challenges above, in this paper, we propose a switching Gaussian mixture variational recurrent neural network (SGmVRNN), which is a powerful probabilistic dynamical model that is able to learn different structural characteristics at different timesteps and capture various temporal dependences between them, and to characterize the complex structural and dynamic characteristics within multivariate KPIs of diverse CDN websites. More specifically, inspired by the probabilistic mixture models [7]–[9], we build a variational recurrent neural network (VRNN) with the assumption that the latent variables of each timestep are drawn from the Gaussian mixture distribution whose parameters for each component consist of two elements: one is the specific prior for the current input, and another is a transformation of the latent state from the last timestep. Based on this, a discrete indicator variable $\boldsymbol{c}_{t,n}$ is introduced to guide the prior selection of the current timestep and how the information transitions between adjacent timesteps, as illustrated in Fig. 3, which is defined as the switching mechanism [10]–[14]. Combining the switching mechanism and the mixture model, SGmVRNN can not only benefit the characterization of current input and the transmission of diverse temporal variation for ample representation capability, tackling *the first challenge* above, with a switching mechanism, but is also able to deal with diverse websites well, addressing *the second challenge* above, with the mixture Gaussian distributed latent variables. Moreover, SGmVRNN can also enable the clustering of current input based on both the structural and temporal characteristics.

Furthermore, to learn the parameters of SGmVRNN, we present an upward-downward autoencoding variational inference method, which combines the bottom-up likelihood and up-bottom prior information of the parameters for accurate posterior approximation, thus to get the rich latent representation for SGmVRNN.

The main contributions of our work are summarized as follows:

- We propose a switching Gaussian mixture variational RNN (SGmVRNN), which incorporates probabilistic mixture and switching mechanism into a variational RNN, thus to efficiently model the non-stationary temporal dependency between adjacent timesteps of multivariate CDN KPIs for one individual website and also the dynamic characteristics of those among different websites.
- To achieve accurate approximation of the posterior of latent variable, we propose an upward-downward autoencoding variational inference method for SGmVRNN.
- We conduct extensive experiments on both a real-world dataset collected from a top CDN provider in China and a public dataset. The quantitative comparison results show that SGmVRNN significantly outperforms the state-of-the-art approaches according to F1-score, and the qualitative analysis shows that it can efficiently characterize CDN KPIs from diverse websites with a single model. For allowing result reproduction, we have released our source code via GitHub at https://github.com/dlagul/SGmVRNN.

II. Preliminaries

In this section, we present the problem and the overall framework of anomaly detection. Then, we give a brief overview of the variational RNN, which is the basis of our work.

A. Problem Definition

Defining the n-th multivariate CDN KPIs as $\boldsymbol{x}_n = \{\boldsymbol{x}_{1,n}, \boldsymbol{x}_{2,n}, ..., \boldsymbol{x}_{T,n}\}$, where $n = 1, ...N$ and N is the number of KPI time series. T is the duration of $\boldsymbol{x}_n$ and the observation at time t, $\boldsymbol{x}_{t,n} \in \mathbb{R}^V$, is a V dimensional vector where V denotes the number of KPIs, thus $\boldsymbol{x}_n \in \mathbb{R}^{T \times V}$. Anomaly detection on multivariate CDN KPIs is defined as a problem that determines whether an observation from a certain website and at a certain time $\boldsymbol{x}_{t,n}$ is anomalous or not. To solve this problem efficiently in an unsupervised way, we need a powerful method for learning the robust representations of the input data.

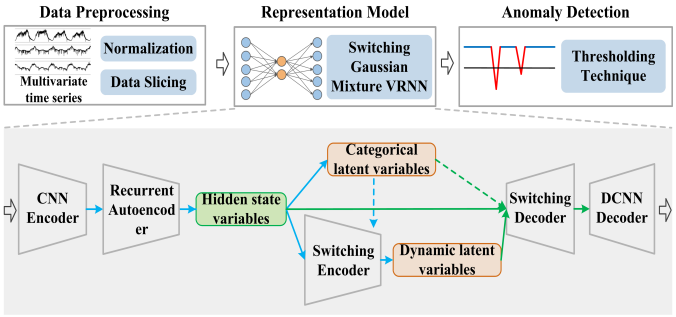


Fig. 2: Framework of the proposed anomaly detection for multivariate CDN KPIs based on SGmVRNN.

B. Overview of the Framework

The complete framework for unsupervised anomaly detection for multivariate CDN KPIs based on SGmVRNN is shown in Fig. 2. The framework contains three key modules. The first module pre-processes the original multivariate CDN KPIs data so that they can be used by the learning model for training. Specifically, the normalization and sliding time window approaches [5] are adopted in this work. In the representation module, we propose a SGmVRNN to learn the complex structural and dynamic characteristics within multivariate CDN KPIs. The detailed description of SGmVRNN will be provided in Section III. Finally, anomalies are detected in terms of the reconstruction probability that is inferred from the representation model. The anomaly detection module will be explained in detail in Section V.

C. Variational RNN (VRNN)

VRNN [15] contains a Variational Autoencoder (VAE) [16] at every timestep and these VAEs are conditioned on the latent states of the last timestep. The prior on latent random variable of the VRNN is no longer a standard Gaussian distribution as standard VAE assumed, but it follows the distribution

$$\begin{aligned} \boldsymbol{z}_t &\sim \mathcal{N}\left(\boldsymbol{\mu}_{0,t}, \operatorname{diag}\left(\boldsymbol{\sigma}_{0,t}^2\right)\right), \\ \left[\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}\right] &= \varphi_\tau^{\text{prior}}\left(\boldsymbol{h}_{t-1}\right) \end{aligned} \tag{1}$$

where $\boldsymbol{\mu}_{0,t}$ and $\boldsymbol{\sigma}_{0,t}$ denote the mean and variance parameters of the conditional prior distribution. The generating distribution of the VRNN is conditioned on both the latent variable $\boldsymbol{z}_t$ and the hidden state variable $\boldsymbol{h}_{t-1}$ as

$$\begin{aligned} \boldsymbol{x}_t \mid \boldsymbol{z}_t &\sim \mathcal{N}\left(\boldsymbol{\mu}_{x,t}, \operatorname{diag}\left(\boldsymbol{\sigma}_{x,t}^2\right)\right), \\ \left[\boldsymbol{\mu}_{x,t}, \boldsymbol{\sigma}_{x,t}\right] &= \varphi_\tau^{\text{dec}}\left(\varphi_\tau^z\left(\boldsymbol{z}_t\right), \boldsymbol{h}_{t-1}\right) \end{aligned} \tag{2}$$

where $\boldsymbol{\mu}_{x,t}$ and $\boldsymbol{\sigma}_{x,t}$ denote the mean and variance parameters of the generating distribution, which transits from $\boldsymbol{z}_t$ and $\boldsymbol{h}_{t-1}$. $\varphi_\tau^{\text{prior}}$, $\varphi_\tau^{\text{dec}}$ and φ_τ^z denote the nonlinear functions respectively, such as a neural network, which is used to extract features. To infer $\boldsymbol{z}_t$, like standard VAE, a Gaussian distributed variational distribution $q(\boldsymbol{z}_t|\boldsymbol{x}_t)$ is used to approximate its true posterior. In a similar fashion with the generative process of VRNN, the variational distribution is a function of both $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$ as

$$\begin{aligned} \boldsymbol{z}_t \mid \boldsymbol{x}_t &\sim \mathcal{N}\left(\boldsymbol{\mu}_{z,t}, \operatorname{diag}\left(\boldsymbol{\sigma}_{z,t}^2\right)\right), \\ \left[\boldsymbol{\mu}_{z,t}, \boldsymbol{\sigma}_{z,t}\right] &= \varphi_\tau^{\text{enc}}\left(\varphi_\tau^x\left(\boldsymbol{x}_t\right), \boldsymbol{h}_{t-1}\right) \end{aligned} \tag{3}$$

where $\boldsymbol{\mu}_{z,t}$ and $\boldsymbol{\sigma}_{z,t}$ are the mean and variance parameters of the variational distribution. In VRNN, the encoding of variational distribution and the decoding of generated distribution are tied together via the hidden state $\boldsymbol{h}_{t-1}$, which enables the temporal information transition through different timesteps. VRNN can be trained in a similar way with VAE.

III. Switching Gaussian Mixture Variational Recurrent Neural Network

In this section, we present the switching Gaussian mixture variational RNN (SGmVRNN), which consists of a novel switching generative model and a powerful upward-downward multiple inference model, for multivariate CDN KPIs.

A. Switching Generative Model of SGmVRNN

As introduced in the subsection II-A, to model the data diversity at different timesteps of multivariate CDN KPIs and solve the non-stationary temporal dependence between them, we extend VRNN into SGmVRNN. Specifically, unlike the standard VRNN, we introduce latent discrete variable $\boldsymbol{c}_{t,n}$ and assign the form of the prior of the latent random variable as

$$\boldsymbol{z}_{t,n}|\boldsymbol{c}_{t,n} \sim \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{z}_{t,n}|\boldsymbol{\mu}_k, \operatorname{diag}\left(\boldsymbol{\sigma}_k\right))^{c_{t,n,k}} \tag{4}$$

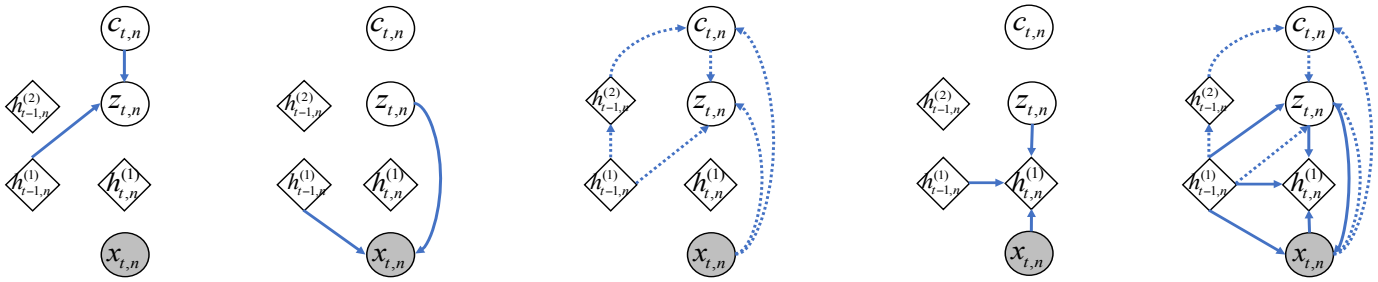
where $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\} = \{\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k\}_{k=1}^K$ denote the parameters of multiple Gaussian distribution and K denotes to the number of components. $\boldsymbol{c}_{t,n} = \left(c_{t,n,1}, \ldots, c_{t,n,K}\right)^T$ is an explicit latent variable associating with each $\boldsymbol{x}_{t,n}$, a one-hot vector which obeys categorical distribution $\boldsymbol{c}_{t,n} \sim \text{Cat}(\boldsymbol{\pi})$ with parameter $\boldsymbol{\pi} \in \mathbb{R}_+^{K\times 1}$. In this way, we assign a different Gaussian distributed prior conditioned on $\boldsymbol{c}_{t,n}$ for latent variable $\boldsymbol{z}_{t,n}$ at different timesteps, which indicates the diverse distribution characteristics of input. Moreover, marginalizing $\boldsymbol{c}_{t,n}$, we can achieve

$$\boldsymbol{z}_{t,n} \sim \sum_{\boldsymbol{c}_{t,n}} p(\boldsymbol{c}_{t,n}|\boldsymbol{\pi}_{t,n}) p(\boldsymbol{z}_{t,n}|\boldsymbol{c}_{t,n}) = \sum_{k=1}^{K} \pi_{t,n,k} \mathcal{N}\left(\boldsymbol{\mu}_k, \operatorname{diag}(\boldsymbol{\sigma}_k)\right)$$

Clearly, it is mixture Gaussian distribution with higher representation power than a Gaussian distribution, and thus is just ideal for characterizing the complex structure and temporal characteristics within multivariate CDN KPIs.

After assigning the form of the prior of the latent variable into the mixture Gaussian distribution, in a similar way with the VRNN, to make sure our generative framework can model both the diversity structure characteristics of input at different timesteps and various temporal dependence among them, we assign the parameters of the mixture distributed latent variables as

$$\begin{aligned} \boldsymbol{\mu}_{t,n,k} &= \varphi_k^{\text{prior}}(\boldsymbol{h}_{t-1,n}^{(1)}, \boldsymbol{c}_{t,n}), k = 1 \cdots K \\ \boldsymbol{\sigma}_{t,n,k} &= \varphi_k^{\text{prior}}(\boldsymbol{h}_{t-1,n}^{(1)}, \boldsymbol{c}_{t,n}), k = 1 \cdots K \end{aligned} \tag{5}$$



(a) Prior (b) Generation (c) Inference (d) Recurrent (e) Overall

Fig. 3: Graphical illustration of each operation of the SGmVRNN: (a) conditional prior of latent variables $z_{t,n}$ and $c_{t,n}$; (b) generation process of $x_{t,n}$; (c) inference of the variational distribution of $z_{t,n}$ and $c_{t,n}$; (d) updating the hidden units of the RNN recurrently; (e) overall operations of the SGmVRNN. Note that circles denote stochastic variables while diamond-shaped units are used for deterministic variables, and shaded nodes denote observed variables.

where $\boldsymbol{h}_{t-1,n}^{(1)}$ is the determinative latent state variable of the RNN, as shown in Figs. 3 (a) and (d). As we can see, the prior of latent variables can be divided into two parts: one is information transition from the last timesteps, and another is the structure-related prior indicated by $\boldsymbol{c}_{t,n}$, which is associated with current input. φ_k^{prior} can be any highly flexible function, such as neural networks, and K different functions are used here to model different structures and temporal characteristics at different timesteps. More specifically, we use the fully connected network as φ_k^{prior}, which is expressed as

$$\begin{aligned}\boldsymbol{\mu}_{t,n,k} &= f(\boldsymbol{W}_{h\mu}^k \boldsymbol{h}_{t-1,n}^{(1)} + \boldsymbol{W}_{c\mu}(k,:)),\\ \boldsymbol{\sigma}_{t,n,k} &= f(\boldsymbol{W}_{h\sigma}^k \boldsymbol{h}_{t-1,n}^{(1)} + \boldsymbol{W}_{c\sigma}(k.:))\end{aligned} \tag{6}$$

where $f(\cdot)$ is a deterministic non-linear transition function, $\{\boldsymbol{W}_{h\mu}^k, \boldsymbol{W}_{h\sigma}^k\}_{k=1}^K \in \mathbb{R}^{d\times d}$ are the transition matrices that capture various temporal dependences between components. $\{\boldsymbol{W}_{c\mu}, \boldsymbol{W}_{c\sigma}\} \in \mathbb{R}^{K\times d}$ are the connected matrices from categorical variable $\boldsymbol{c}_{t,n}$ to the latent variable, which assigns a specific prior in terms of the structure of the current input, and d is the dimension of latent variable $\boldsymbol{z}_{t,n}$. Based on this, the diversity of both structural and dynamical characteristics within KPIs are considered by our model. In addition, we define the parameter of the prior of $\boldsymbol{c}_{t,n}$ as $\boldsymbol{\pi}_{t,n}^0$

Similar with [15], we assign the generated distribution $p(\tilde{\boldsymbol{x}}_{t,n}|\boldsymbol{z}_{t,n})$ to be the Gaussian distribution and conditioned on both $\boldsymbol{z}_{t,n}$ and $\boldsymbol{h}_{t-1,n}^{(1)}$, which can be expressed as

$$\begin{aligned}\tilde{\boldsymbol{x}}_{t,n}|\boldsymbol{z}_{t,n} &\sim \mathcal{N}\left(\boldsymbol{\mu}_{t,n}^x, \boldsymbol{\sigma}_{t,n}^x\right),\\ \boldsymbol{\mu}_{t,n}^x &= \varphi_\mu^{\text{dec}}(\boldsymbol{z}_{t,n}, \boldsymbol{h}_{t-1,n}^{(1)}),\\ \boldsymbol{\sigma}_{t,n}^x &= \varphi_\sigma^{\text{dec}}(\boldsymbol{z}_{t,n}, \boldsymbol{h}_{t-1,n}^{(1)})\end{aligned} \tag{7}$$

where $\boldsymbol{\mu}_{t,n}^x, \boldsymbol{\sigma}_{t,n}^x$ denote mean and variance parameters of the generating distribution, and φ_μ^{dec} and $\varphi_\sigma^{\text{dec}}$ can also be any nonlinear flexible functions and we also use neural network as

$$\begin{aligned}\boldsymbol{\mu}_{t,n}^x &= f(\boldsymbol{W}_{z\mu}^x \boldsymbol{z}_{t,n} + \boldsymbol{W}_{h\mu}^x \boldsymbol{h}_{t-1,n}),\\ \boldsymbol{\sigma}_{t,n}^x &= f(\boldsymbol{W}_{z\sigma}^x \boldsymbol{z}_{t,n} + \boldsymbol{W}_{h\mu}^x \boldsymbol{h}_{t-1,n})\end{aligned} \tag{8}$$

where $\{\boldsymbol{W}_{z\mu}^x, \boldsymbol{W}_{z\sigma}^x\} \in \mathbb{R}^{d\times\tilde{V}}$ and $\{\boldsymbol{W}_{h\mu}^x, \boldsymbol{W}_{h\mu}^x\} \in \mathbb{R}^{d\times\tilde{V}}$ are learnable parameters of our generative model. Finally, to better generate the structures of data, we further apply a deconvolutional network as $\boldsymbol{x}_{t,n} = \text{DCNN}(\tilde{\boldsymbol{x}}_{t,n})$, whose parameters are defined as $\boldsymbol{D}$. For ease of understanding, the graphical illustration of the whole generation process is listed in Figs. 3 (a) and (b).

As introduced in [17] and [18], feature extraction is crucial for modeling complex sequences and the good fitting performance plays an important role in the reconstruction-based unsupervised anomaly detection. In this work, the latent variables have a mixture distribution prior, and the transition between them is handled by a switching mechanism, which improves their power in modeling complex sequences. We will prove this by visualizing the reconstruction likelihood in the experiment. Besides, the RNN in our model updates its hidden states using recurrent equation

$$\boldsymbol{h}_{t,n}^{(1)} = f_\theta\left(\varphi_\tau^{\text{x}}\left(\boldsymbol{x}_{t,n}\right), \boldsymbol{z}_{t,n}, \boldsymbol{h}_{t-1,n}^{(1)}\right) \tag{9}$$

where $f_\theta(\cdot)$ is a deterministic non-linear transition function, and here we use long short-term memory (LSTM), a gated activation function, with parameter $\boldsymbol{\theta}$. As we can see, the hidden state of the RNN is the function of $\boldsymbol{z}_{t,n}, \boldsymbol{x}_{t,n}$, and $\boldsymbol{h}_{t-1,n}^{(1)}$, indicating the distribution of $\boldsymbol{z}_{t,n}$ and $\boldsymbol{x}_{t,n}$ in Eqs. (4) and (7) can be defined as $p(\boldsymbol{z}_{t,n}|\boldsymbol{x}_{<t,n}, \boldsymbol{z}_{<t,n}, \boldsymbol{c}_{t,n})$ and $p(\boldsymbol{x}_{t,n}|\boldsymbol{x}_{<t,n}, \boldsymbol{z}_{\leq t,n})$.

Note that the generative model of SGmVRNN reduces to VRNN if we remove the switching and mixture structure, which models the diversity of the structure and temporal characteristics within multivariate CDN KPIs, by setting $K = 1$. In addition, SGmVRNN also reduces to GmVAE [19], [20] if we ignore its recurrent structure that models the temporal dependence within multivariate CDN KPIs.

B. Upward-downward Switching Inference Network

In a similar way with VRNN, we develop an inference network to map the inputs directly to their latent variables. Specifically, we use a Concrete distribution [21] to approximate the categorical distributed indicator variables $\boldsymbol{c}_{t,n}$, and a Gaussian distribution based inference network for latent representation $\boldsymbol{z}_{t,n}$. Besides, to achieve accurate inference, we further develop an upward-downward switching inference network, as shown in Fig. 3 (c).

*1) **Gumbel-softmax-based variational inference for $\boldsymbol{c}_{t,n}$:*** The categorical variable $\boldsymbol{c}_{t,n}$ controls the structural prior of latent state variable $\boldsymbol{z}_{t,n}$ and its change from $t-1$ to t. To learn $\boldsymbol{c}_{t,n}$, we apply variational inference [22], [23]. However, it is difficult to directly optimize the discrete variable $\boldsymbol{c}_{t,n}$ since the back-propagation algorithm cannot be applied to non-differentiable layers. Inspired by [22], [23], we introduce a differentiable sample from Gumble-softmax distribution to approximate the sample from the categorical distribution. Specifically, we assign the variational distribution as $q(\boldsymbol{c}_{t,n}) =$ Gumble-softmax$(\bar{\boldsymbol{\pi}}_{t,n})$, where $\bar{\boldsymbol{\pi}}_{t,n} \in \mathbb{R}^{K\times 1}$ is the parameter for $q(\boldsymbol{c}_{t,n})$ and it draws samples via

$$\begin{aligned} c_{t,n,k} &= \frac{\exp\left(\left(\log \bar{\pi}_{t,n,k} + g_{t,n,k}\right)/\lambda\right)}{\sum_{k=1}^{K}\exp\left(\left(\log \bar{\pi}_{t,n,k} + g_{t,n,k}\right)/\lambda\right)} \\ &\text{for } k = 1,\ldots,K, \\ g_{t,n,k} &\sim \text{Gumbel}(0,1) = -\log\left(-\log\left(\epsilon_{t,n,k}\right)\right) \end{aligned} \tag{10}$$

where λ denotes the softmax temperature and $\epsilon_{t,n,k}$ refers to the standard uniform variable. As λ approaches 0, samples from the Gumbel-softmax distribution become one-hot and the Gumbel-softmax distribution $q(\boldsymbol{c}_{t,n})$ becomes identical to the categorical distributed prior of $\boldsymbol{c}_{t,n}$, which is denoted as $p(\boldsymbol{c}_{t,n})$.

*2) **Gaussian-based variational inference for $\boldsymbol{z}_{t,n}$:*** As the usual strategy of VAE, we use a Gaussian variational distribution $q(\boldsymbol{z}_{t,n})$ to infer the posterior distribution of $\boldsymbol{z}_{t,n}$. We assign $q(\boldsymbol{z}_{t,n})$ as

$$q(\boldsymbol{z}_{t,n}) = \mathcal{N}(\bar{\boldsymbol{\mu}}_{t,n}, \text{diag}(\bar{\boldsymbol{\sigma}}_{t,n})) \tag{11}$$

where $\bar{\boldsymbol{\mu}}_{t,n}$ and $\bar{\boldsymbol{\lambda}}_{t,n}$ are the parameters determinately transformed from the inference network, which will be introduced below. Sampling from the latent variable can be achieved by

$$\boldsymbol{z}_{t,n} = \bar{\boldsymbol{\mu}}_{t,n} + \bar{\boldsymbol{\sigma}}_{t,n}\boldsymbol{\epsilon}_{t,n}, \boldsymbol{\epsilon}_{t,n} \sim \text{Uniform}(0,1), \tag{12}$$

which ensures the application of back-propagation algorithm.

*3) **Upward-downward information propagation:*** As a deep VAE, we define the joint variational distribution $q(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n}|\boldsymbol{x}_{t,n})$, that needs to be flexible enough to approximate the true posterior distribution $p(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n}|\boldsymbol{x}_{t,n})$ as closely as possible, to be factorized with a bottom-up structure [24]–[26] as

$$q(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n}|\boldsymbol{x}_{t,n}) = q(\boldsymbol{z}_{t,n}|\boldsymbol{x}_{t,n})q(\boldsymbol{c}_{t,n}|\boldsymbol{z}_{t,n}, \boldsymbol{x}_{t,n}) \tag{13}$$

To achieve the accurate optimization of the hierarchically structured variational distribution, inspired by ladder VAE of [27], [28], we construct an upward-downward inference network by combing the bottom-up likelihood information and up-bottom prior information from the generative distribution as

$$q(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n}|\boldsymbol{x}_{t,n}) = q(\boldsymbol{c}_{t,n}|\boldsymbol{x}_{t,n})q(\boldsymbol{z}_{t,n}|\boldsymbol{x}_{t,n}, \boldsymbol{c}_{t,n}) \tag{14}$$

Specifically, for capturing the structural characteristic of input, we first conduct a one-dimensional convolutional network as $\bar{\boldsymbol{x}}_{t,n} = \text{CNN}(\boldsymbol{x}_{t,n})$, where CNN denotes the convolutional operation and we define its parameters as $\bar{\boldsymbol{D}}$, $\bar{\boldsymbol{x}}_{t,n} \in \mathbb{R}^{\bar{V}\times 1}$. Then, as shown in Fig. 3 (c), for upward information transition, we get the latent state variable $\boldsymbol{h}_{t,n}^{(1)}$ of the recurrent network with Eq. (9). Based on this, we further apply fully connected network to get the latent state variable $\boldsymbol{h}_{t,n}^{(2)}$ for $\boldsymbol{c}_{t,n}$ as

$$\boldsymbol{h}_{t,n}^{(2)} = f(\bar{\boldsymbol{W}}_{hy}\boldsymbol{h}_{t,n}^{(1)} + \bar{\boldsymbol{b}}_{hy}) \tag{15}$$

where $\bar{\boldsymbol{W}}_{hy} \in \mathbb{R}^{d_1\times d_2}$ denotes the hidden-to-hidden weight matrix, d_2 refers to the dimension of $h_{t,n}^{(2)}$. With the assumption that the real-word changes in dynamics at timestep t are causal [13], [29], which is also appropriate for multivariate CDN KPIs, we assign $\boldsymbol{c}_{t,n}$ to non-linearly depends on the history input. Specifically, in inference network for $\boldsymbol{\pi}_{t,n}$, to make sure the information transition from both the current and history inputs, we parameterize two deterministic upward paths to obtain the indicator variable as

$$\bar{\boldsymbol{\pi}}_{t,n} = \text{softmax}(\bar{\boldsymbol{W}}_{xc}\bar{\boldsymbol{x}}_{t,n} + \bar{\boldsymbol{W}}_{hc}\boldsymbol{h}_{t-1,n}^{(2)} + \bar{\boldsymbol{b}}_{hc}) \tag{16}$$

where $\bar{\boldsymbol{W}}_{xc} \in \mathbb{R}^{\bar{V}\times K}$, $\bar{\boldsymbol{W}}_{hc} \in \mathbb{R}^{d_2\times K}$ and $\bar{\boldsymbol{b}}_{hc} \in \mathbb{R}^{K}$ denote the learnable parameters of inference model from $\bar{\boldsymbol{x}}_{t,n}$ and $\boldsymbol{h}_{t-1,n}^{(2)}$ to $\bar{\boldsymbol{\pi}}_{t,n}$. After getting $\bar{\boldsymbol{\pi}}_{t,n}$, we can sample $\boldsymbol{c}_{t,n}$ via Eq. (10). $\bar{\boldsymbol{W}}_{xc}\boldsymbol{x}_{h,t}$ in Eq. (16) ensures that $\boldsymbol{c}_{t,n}$ directly reflects the structural characteristics of current input, while $\bar{\boldsymbol{W}}_{hc}\boldsymbol{h}_{t-1,n}^{(2)}$ reflects the temporal relations of latent states. SGmVRNN fuses the obtained indicator variable $\boldsymbol{c}_{t,n}$ with the prior to guide the inference of $\boldsymbol{z}_{t,n}$ and construct the switching inference network as

$$\begin{aligned} q(\boldsymbol{z}_{t,n}|\boldsymbol{x}_{t,n}, \boldsymbol{h}_{t-1,n}^{(1)}, \boldsymbol{c}_{t,n}) &= \mathcal{N}(\bar{\boldsymbol{\mu}}_{t,n}, \text{diag}(\bar{\boldsymbol{\sigma}}_{t,n})) \\ \bar{\boldsymbol{\mu}}_{t,n} &= f(\prod_{k=1}^{K}(\bar{\boldsymbol{W}}_{h\mu}^{k}\boldsymbol{h}_{t-1,n}^{(1)} + \bar{\boldsymbol{W}}_{x\mu}^{k}\bar{\boldsymbol{x}}_{t,n} + \bar{\boldsymbol{b}}_{h\mu}^{k})^{c_{t,n,k}}) \\ \bar{\boldsymbol{\sigma}}_{t,n} &= \text{softplus}(\prod_{k=1}^{K}(\bar{\boldsymbol{W}}_{h\lambda}^{k}\boldsymbol{h}_{t-1,n}^{(1)} + \bar{\boldsymbol{W}}_{x\lambda}^{k}\bar{\boldsymbol{x}}_{t,n} + \bar{\boldsymbol{b}}_{h\lambda}^{k})^{c_{t,n,k}}) \end{aligned}$$

softplus denotes the operation $\log(1+\exp(\cdot))$ to ensure the nonlinearly and positively transition from the latent states to $\boldsymbol{\sigma}$. $\{\bar{\boldsymbol{W}}_{h\mu}^{k}, \bar{\boldsymbol{W}}_{h\lambda}^{k}\}_{k=1}^{K} \in \mathbb{R}^{d_1\times d}$ and $\{\bar{\boldsymbol{W}}_{x\mu}^{k}, \bar{\boldsymbol{W}}_{x\lambda}^{k}\}_{k=1}^{K} \in \mathbb{R}^{\bar{V}\times d}$ denote the learnable parameters of the inference model from $\boldsymbol{x}_{t,n}$ and $\boldsymbol{h}_{t-1,n}^{(1)}$ to $\boldsymbol{z}_{t,n}$. In this way, the mean and variance parameters of the latent variable $\boldsymbol{z}_{t,n}$ are both inferred by combining the bottom-up likelihood information and the prior information from the generative distribution using the inference network. By integrating the indicator variable into the inference network, the diversity of the structural and temporal information within the inputs are fused, enabling the proposed upward-downward multiple variational inference network to learn the rich latent representation for SGmVRNN.

C. Model Properties

SGmVRNN inherits the good properties of both the probabilistic mixture model and variable RNN, as described bellow.

*1) **Clustering based on both structural and temporal characteristics:*** As introduced in [7] and [9], the probabilistic mixture model is a wildly used clustering method and it separates the data into several groups according to their

structure. According to Eq. (7), the posterior of the cluster indicator variable $\boldsymbol{c}_{t,n}$ can be obtained as

$$\begin{aligned} p(\boldsymbol{c}_{t,n}|-) &\propto p(\boldsymbol{z}_{t,n}|\boldsymbol{c}_{t,n})p(\boldsymbol{c}_{t,n}|\pi_{t,n}) \\ &= \prod_{k=1}^{K} \left(\pi_{t,n,k}\mathcal{N}(\boldsymbol{\mu}_{t,n,k}, \mathrm{diag}(\boldsymbol{\sigma}_{t,n,k}))\right)^{c_{t,n,k}} \\ &= \prod_{k=1}^{K} \left(\pi_{t,n,k}\mathcal{N}(\boldsymbol{W}_{h\mu}^{k} h_{t-1,n}^{(1)} + \boldsymbol{W}_{c\mu}(k,:),\right. \\ &\quad \left.\mathrm{diag}(\boldsymbol{W}_{h\sigma}^{k} h_{t-1,n}^{(1)} + \boldsymbol{W}_{c\sigma}(k.:))\right)^{c_{t,n,k}} \end{aligned} \tag{17}$$

As we can see, the parameters of the posterior of $c_{t,n}$ are the function of both $\boldsymbol{W}_{c\mu}(1:K,:), \boldsymbol{W}_{c\sigma}(1:K,:)$, which reflects the structural characteristics of input at current timestep as usual mixture model did, and the $\{\boldsymbol{W}_{h\mu}^{k}, \boldsymbol{W}_{h\sigma}^{k}\}_{k=1}^{K}$, which are the transition matrices that model different temporal dependence of adjacent timesteps. In this way, indicator variable can reflect both the structural and the dynamic characteristics of multivariate CDN KPIs.

*2) **Switching mechanism**:* As mentioned in [13] and [14], the switching mechanism is defined as follows. There are multiple different connection parameters between two variables, and one parameter within them will be selected so that it corresponds to current inputs, thus to increase the representation ability of the model. As mentioned before, no matter the generation or the inference process of SGmVRNN, there exists the switching mechanism within them. For the generation process, $\{\boldsymbol{W}_{h\mu}^{k}, \boldsymbol{W}_{h\sigma}^{k}\}_{k=1}^{K}$ are switched under the guidance of $\boldsymbol{c}_{t,n}$ to model different temporal dependences between adjacent timesteps. For the inference process, with our upward-downward inference method, $\{\bar{\boldsymbol{W}}_{h\mu}^{k}, \bar{\boldsymbol{W}}_{h\sigma}^{k}\}_{k=1}^{K}$ are also switched to enable the inference network to learn rich a latent representation.

*3) **Beneficial to model Multivariate CDN KPIs**:* The properties of the SGmVRNN make it more suitable at modeling of multivariate CDN KPIs, especially for tackling the two challenges introduced in Section I. For the CDN KPIs of a single website, SGmVRNN can model the diversity structure and non-stationary temporal dependence within them. Besides, for the CDN KPIs from different websites, which exhibit varying characteristics, previous methods always need to train different specific groups of parameters to model them. On the contrary, SGmVRNN can model them with only one group of parameters for being able to capture the various characteristics within them by the mixture distributed latent variables, and then modeling them with the switching mechanism.

IV. Model Training

The training objective of SGmVRNN is to minimize the distance between the variational distribution of latent variables $q(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n})$ and their true posterior distribution $p(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n}|-)$, which can be quantified by the Kullback-Leibler (KL) divergence, $\mathrm{KL}(q(x)||p(x)) = \int q(x)\log\frac{q(x)}{p(x)}dx$. So, during the inference of SGmVRNN, we aim to minimize $\mathrm{KL}(q(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n})||p(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n}|-))$. Following the usual strategy of VAE, we can transfer this KL divergence as

$$\begin{aligned} &\mathrm{KL}(q(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n})||p(\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n}|-)) \\ &= \log p(\boldsymbol{x}_{t,n}) - \mathbb{E}_{q(\boldsymbol{z}_{t,n})q(\boldsymbol{c}_{t,n})}\left(\log p(\boldsymbol{x}_{t,n}|\boldsymbol{z}_{t,n}, \boldsymbol{c}_{t,n})\right. \\ &\left. - \log\frac{q(\boldsymbol{z}_{t,n})q(\boldsymbol{c}_{t,n})}{p(\boldsymbol{z}_{t,n}|\boldsymbol{c}_{t,n})p(\boldsymbol{c}_{t,n})}\right) \\ &= \log p(\boldsymbol{x}_{t,n}) - \mathrm{ELBO} \end{aligned}$$

where ELBO refers to the evidence lower bound. As illustrated in Eq. (18), minimizing the KL divergence can be transformed to maximize the ELBO, and we can re-express it as

$$\begin{aligned} \mathrm{ELBO} = \mathbb{E}_{q(\boldsymbol{z}_{t,n})q(\boldsymbol{c}_{t,n})}(\log(p(\boldsymbol{x}_{t,n}|\boldsymbol{z}_{t,n}) - \\ \mathrm{KL}\left(q(\boldsymbol{z}_{t,n})||p(\boldsymbol{z}_{t,n}|\boldsymbol{c}_{t,n})\right) - \mathrm{KL}\left(q(\boldsymbol{c}_{t,n})||p(\boldsymbol{c}_{t,n})\right) \end{aligned} \tag{18}$$

where the first term is the expected log-likelihood that guarantees the reconstruction capacity of the generative model, while the second and the third terms are the KL divergence that constrains variational distributions to be close to their prior in the generative model. Specifically, because $\boldsymbol{c}_{t,n}$ is a discrete variable, the KL divergence between $q(\boldsymbol{c}_{t,n})$ and $p(\boldsymbol{c}_{t,n})$ can be calculated as

$$\begin{aligned} &\mathrm{KL}(q(\boldsymbol{c}_{t,n})||p(\boldsymbol{c}_{t,n})) \\ &= \sum_{k=1}^{K}\left[q_k\log q_k\right] - \sum_{i=1}^{k}\left[q_k\log p\left(c_{t,n,k}=1\right)\right] \\ &= \sum_{k=1}^{K}\left[q_k\log\left(q_k\right)\right] - \log\left(\frac{1}{k}\right) \end{aligned} \tag{19}$$

where q_k denotes $q\left(c_{t,n,k}=1\right)$. As we assign the prior of latent variables as mixture Gaussian distribution, the KL divergence between the variational distribution $q(z_{t,n})$ and the conditional prior $p(z_{t,n}|c_{t,n})$ can be derived as

$$\begin{aligned} &\mathrm{KL}(q(\boldsymbol{z}_{t,n})\|p(\boldsymbol{z}_{t,n} \mid \boldsymbol{c}_{t,n})) \\ &= \int\left(\sum_{k=1}^{K} c_{t,n,k}q(\boldsymbol{z}_{t,n})\log\frac{q(\boldsymbol{z}_{t,n})}{\mathcal{N}\left(\boldsymbol{\mu}_{t,n,k}, \boldsymbol{\sigma}_{t,n,k}\right)}\right) \\ &= \sum_{k=1}^{K} c_{t,n,k}\mathrm{KL}\left(q(\boldsymbol{z}_{t,n})\|\mathcal{N}\left(\boldsymbol{\mu}_{t,n,k}, \boldsymbol{\sigma}_{t,n,k}\right)\right) \end{aligned} \tag{20}$$

Then, the KL-divergence between two Gaussian distribution has an analytic expression as

$$\begin{aligned} &\mathrm{KL}(\mathcal{N}(\boldsymbol{\mu}_1, \mathrm{diag}(\boldsymbol{\sigma}_1))||\mathcal{N}(\boldsymbol{\mu}_2, \mathrm{diag}(\boldsymbol{\sigma}_2))) = \\ &\frac{1}{2}\left[\log\frac{\boldsymbol{\sigma}_1}{\boldsymbol{\sigma}_2} - d + \boldsymbol{\sigma}_2^{-1}\boldsymbol{\sigma}_1 + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\mathrm{diag}(\boldsymbol{\sigma}_2^{-1})(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right] \end{aligned} \tag{21}$$

Thanks to the analytic KL expression in Eqs. (19), (20) and (21), and also easy reparameterization of the Gumble-softmax and Gaussian distribution, the gradient of ELBO with respect to the parameters in the inference network can be accurately evaluated. We list the details of the upward-downward autoencoding variational inference for SGmVRNN in Algorithm 1. As described in Algorithm 1, the encoder parameters, defined as $\boldsymbol{\Omega}$, and decoder parameters, defined as $\boldsymbol{\Psi}$, in SGmVRNN are jointly updated with stochastic gradient descent (SGD).

Algorithm 1 Upward-Downward Autoencoding Variational Inference for SGmVRNN

Input: The pre-processed KPI training dataset $\mathcal{D}\left(x_{1:T}\right)$;
Output: The encoder parameters of SGmVRNN: $\boldsymbol{\Omega}=\left\{\boldsymbol{D}, \boldsymbol{W}_{hy}, \{\boldsymbol{W}_{h\mu}^{k}, \boldsymbol{W}_{h\lambda}^{k}\}_{k=1}^{K}\right\}$;
the decoder parameters of SGmVRNN: $\boldsymbol{\Psi}=\left\{\bar{\boldsymbol{D}}, \{\bar{\boldsymbol{W}}_{h\mu}^{k}, \bar{\boldsymbol{W}}_{h\lambda}^{k}, \bar{\boldsymbol{W}}_{x\mu}^{k}, \bar{\boldsymbol{W}}_{x\lambda}^{k}\}_{k=1}^{K}, \bar{\boldsymbol{W}}_{xh}, \bar{\boldsymbol{W}}_{hh}, \bar{\boldsymbol{W}}_{hy}\right\}$;
and the parameter of recurrent network $\boldsymbol{\theta}$;
Set mini-batch size as M, the number of convolutional filters K and hyperparameters;
Initialize the encoder parameters $\boldsymbol{\Omega}$, decoder parameters $\boldsymbol{\Psi}$ and recurrent parameters $\boldsymbol{\theta}$;
repeat
  Randomly select a mini-batch of M multivariate CDN KPIs consist of T subsequences to form a subset $\{\boldsymbol{x}_{1:T,i}\}_{i=1}^{M}$;
  Draw random noise $\left\{\epsilon_{t,n}^{c}\right\}_{t=1,n=1}^{T,M}$ and $\left\{\epsilon_{t,n}^{z}\right\}_{t=1,n=1}^{T,M}$ from uniform distribution for sampling latent states $\{c_{t,n}\}_{t=1,n=1}^{T,M}$ and $\{z_{t,n}\}_{t=1,n=1}^{T,M}$;
  Calculate $\nabla_{\Omega} L\left(\boldsymbol{\Omega}, \Psi; X, \epsilon_{t,i}^{c}, \epsilon_{t,i}^{c}\right)$ according to Equation (18), (19), (20) and (21), and update encoder parameters $\boldsymbol{\Omega}$ and decoder parameters $\boldsymbol{\Psi}$ jointly;
until convergence
return global parameters $\{\boldsymbol{\Omega}, \boldsymbol{\Psi}, \boldsymbol{\theta}\}$.

TABLE I: Basic statistics of datasets

Statistics	KPIs of CDN	KPIs of SMD
Dimensions	31*36	28*38
Granularity (sec)	60	60
Training set size	1,227,249	708,405
Testing set size	1,227,250	708,420
Anomaly ratio (%)	3.68	4.16

V. Anomaly Detection

Since the model is usually trained to learn the normal patterns of multivariate CDN KPIs, the more an observation follows normal patterns, the more likely it can be reconstructed with higher confidence. Hence, we apply the reconstruction probability of x_t as the anomaly score to determine whether an observed variable is anomalous or not [3], [30]–[33], and it is computed as

$$\mathcal{S}_{t,n} = \log p(\boldsymbol{x}_{t,n}|\boldsymbol{z}_{t,n}) \tag{22}$$

An observation x_t will be classified as anomalous if $\mathcal{S}_t$ is below a specific threshold. From a practical point of view, we use the Peaks-Over-Threshold (POT) [34] approach to help select such threshold. In our case, the lower anomaly scores are more likely considered to be extreme values, because the lower anomaly score, the greater the probability of outlier. Therefore, similar to [3], we adopt the lower-bound thresholds.

VI. Experiment

In this section, we first introduce the experiment setup and then evaluate our model via various experiments.

A. Experiment Setup

1) ***Dataset****:* We make extensive experiments on two categories of real-world datasets: a CDN multivariate KPI dataset and a public dataset named SMD that was released by the work [3]. CDN multivariate KPI dataset is collected from a popular ISP-operated CDN in China, and the dataset contains 31 websites that are monitored with 36 KPIs individually. These websites are different from each other in types of services, etc. In our experiments, for each website, the first half of the KPIs are used for training, while the second half are used for testing. Note that for the experiment "one model fits all websites", we aggregate all of the training set to train the model, while in the experiment "one model for one website", the models are trained on each website dataset individually. The basic statistical information of datasets is reported in Table I, and the ground-truth anomalies in the test set of CDN have been confirmed by human operators. Please refer to [3] for more details of the public dataset SMD.

2) ***Evaluation metrics****:* Three metrics, including Precision, Recall and F1 score, are employed as the performance indicators [2]–[5], [32], [35]. Among them, F1 is deemed as a comprehensive indicator since it balances the precision and recall. Note that, in practice, if any point in a ground-truth anomaly segment is correctly detected, all points in the ground-truth anomaly segment will be identified as true positive [3], [5].

3) ***Compared baselines****:* We employ the state-of-the-art deep anomaly models as the baseline methods, including *1)* VRNN [15], a probabilistic model that extends the VAE into a recurrent framework for modelling high-dimensional sequences; *2)* DOMI [31], a deep model that combines Gaussian mixture VAE (GmVAE) with 1D-CNN to detect outlier machine instances; *3)* OmniAnomaly [3], a stochastic RNN-based model; *4)* SDFVAE [5], a static and dynamic factorized VAE-based framework to conduct anomaly detection for each CDN website individually.

4) ***Hyper-parameters****:* In our experiments, we implement SGmVRNN based on Pytorch. Both the CNN encoder and DCNN decoder are with 3 of 1-dimension convolutional layers, whose filters and strides are set to (3,3), (2,2), (2,2) successively. The dimensions of the hidden states of LSTM-Cell are 20. Besides, we set z-space and categorical-space dimensions to 10 and 5 empirically. The Adam optimizer is employed with a learning rate of 0.0002, and the batch size is set to 256. Besides, we set the initial temperature λ in Gumbel-Softmax to 5.0, and anneal to a small but non-zero temperature, e.g., 0.1, with rate 0.1 per epoch. And the probability p associated with the initial threshold used in POT is set to 0.003 empirically.

5) ***Hardware platform****:* Our experiments are conducted on servers with Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz accelerated by two NVIDIA RTX 8000, with 48GB VRAM of each graphics card.

B. Quantitative Comparison

1) ***The influence of model parameters****:* In this part, we discuss the sensitivity of model parameters, which includes the the number K of clusters and dimension d of latent variables.

Fig. 5 (left) presents the changing curve of the F1-score with the increasing clustering number, which determines the number of switching parameters of SGmVRNN. As shown in

TABLE II: Performance of "one model for one website".

Methods	CDN			SMD		
	P	R	F1	P	R	F1
VRNN	0.9578	0.9012	0.9286	0.9139	0.9346	0.9241
DOMI	0.9376	0.8781	0.9069	0.9425	0.9125	0.9273
OmniAnomaly	0.9832	0.8755	0.9262	0.8290	0.9681	0.8932
SDFVAE	0.9428	0.9385	0.9407	0.9717	0.9035	0.9364
SGmVRNN	0.9595	0.9448	**0.9521**	0.9514	0.9290	**0.9400**

TABLE III: Performance of "one model fits all websites".

Methods	CDN			SMD		
	P	R	F1	P	R	F1
VRNN	0.9814	0.8317	0.9003	0.9825	0.8383	0.9047
DOMI	0.9665	0.8348	0.8958	0.9770	0.8036	0.8819
OmniAnomaly	0.8385	0.8757	0.8567	0.9801	0.7843	0.8713
SDFVAE	0.9675	0.8615	0.9115	0.9810	0.8498	0.9107
SGmVRNN	0.9667	0.9204	**0.9430**	0.9607	0.9123	**0.9356**

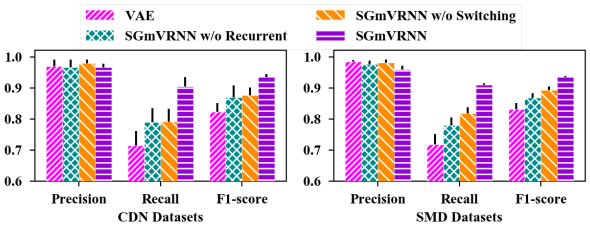


Fig. 4: Ablation study of SGmVRNN

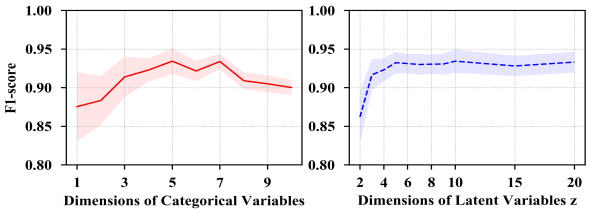


Fig. 5: Varying Dimensions of Latent Variables

Fig. 5 (left), the F1-score of the SGmVRNN first increases and then decreases a little bit with the number of clusters ranging from 1 to 10. The main reason for this phenomenon is that a very small number of clusters cannot characterize all the structural patterns within CDN KPIs. Similarly, a sub-optional high number of clusters is likely to produce redundant parameters in the SGmVRNN, which will increase memory burden and computational complexity, thus resulting in worse performance.

Second, we evaluate the effect of the dimension d of latent variables and list the F1 scores of the proposed model with different values of d in Fig. 5 (right). The anomaly detection performance is improved along with the increase of the hidden state dimensions, for the promotion of representation power. However, the higher dimension of the hidden states will result in higher computational cost, even though at a certain point it does not help the score improve.

*2) **Anomaly Detection Performance**:* To evaluate the performance of SGmVRNN, the experiments of "one model for one website" and "one model fits all websites" are performed and the results are reported in Table II and Table III respectively. The best F1-score on each dataset is highlighted in boldface.

In the experiments of "one model for one website", SGmVRNN outperforms all the baselines, verifying the efficiency of the proposed switching mechanism in modeling the non-stationary temporal dependence in multivariate KPIs. Similarly, in the experiments of "one model fits all websites", as we can see, by considering both structural and dynamical diversity within CDN KPIs from different websites, and incorporating mixture distributed variable and switching mechanism for characterizing them, SGmVRNN outperforms all the baselines on all test datasets, showing its efficiency in anomaly detection of diverse CDN websites. Moreover, by comparing the results in Table II and Table III, we obtain an interesting observation: SGmVRNN only decreases less performance from the experiment of "one model for one website" to "one model fits all websites", while the other baseline methods show the significant decrease of performance, e.g., the results of OnmiAnomaly on CDN dataset. This indicates that the mixture distributed variables and switching mechanism play an important role in the performance improvement of SGmVRNN.

*3) **Ablation Study**:* We conduct ablation study to analyze the importance of switching and recurrent structure in our model by comparing the results of SGmVRNN, SGmVRNN w/o switching by setting $K = 1$, SGmVRNN w/o recurrent by setting $T = 1$ and the basic VAE. Observations can be drawn from the experimental results that are shown in Fig. 4. As we can see, every structures we incorporate into SGmVRNN can bring improvement on performance, illustrating the effectiveness of each component in it.

C. Qualitative Analysis

In addition to quantitative analysis, we also make some qualitative analyses in this subsection.

First, to show intuitively the efficiency of our mixture distributed latent variable and switching mechanism, we visualize an case study of SGmVRNN in Fig. 6. Fig. 6 (a) shows a non-stationary multivariate time series with diverse structural and temporal characteristics at different timesteps, which is the challenge 1 introduced in Section I. To tackle this challenge, as a probabilistic mixture model, SGmVRNN clusters the multivariate time series in Fig. 6 (a) into several groups and the clustering indexes are shown in Fig. 6 (b). As we can see, the clustering groups reflect the structural and temporal characteristics at different tiemsteps. Besides, as shown in Figs. 6 (c) and (d), by the aid of the switching mechanism and guided by the clustering index in Fig. 6 (b), SGmVRNN can model the input at different timesteps assigned into different cluster groups with different parameters, thus leading to a more stable anomaly score when compared to VRNN, while it exhibits considerable spikes in the regions of anomalies. It further demonstrates the capability of SGmVRNN to learn normal patterns of complex KPIs.

We also visualize the 2D embedding of the latent variables on SGmVRNN on four example multivariate CDN KPIs from

TABLE IV: Training and testing time of SGmVRNN

Datasets	# Training samples	Training times per epoch (min)	Testing times per sample (sec)
CDN	1,227,228	7.3	0.093
SMD	708,399	4.67	0.098

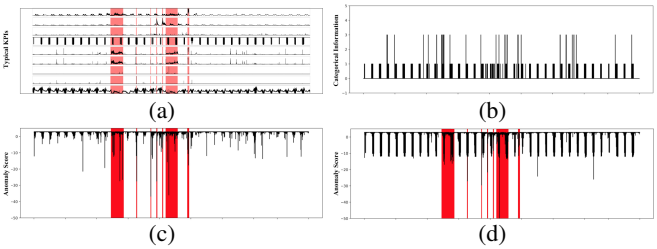


Fig. 6: Case study of (a) multivariate CDN KPIs; (b) clustering index; (c) anomaly score by SGmVRNN; (d) anomaly score by VRNN.

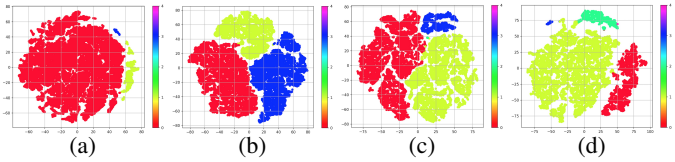


Fig. 7: The visualization of latent variables $z_{t,n}$

different websites using t-distributed stochastic neighborhood embedding (t-SNE) [36] methods in Fig. 7. Each dot in Fig. 7 represents the latent variable of the observation at a certain timestep, and each color represents a clustering group. From Fig. 7, we can observe the following properties on the latent variable $z_{t,n}$: *1)* There are different structural characteristics at different timesteps of KPIs from one website and our model can separate them into different clustering groups; *2)* There are some similar characteristics shared among all websites, such as cluster group $0, 1, 3$ in Fig. 7; *3)* Different websites may also exhibit some specific structural characteristics, such as cluster group $2, 4$ in Fig. 7 (d). Note that Fig. 7 (a) shows the corresponding visualized latent variables of Fig. 6 (a). All these results conform with the two challenges listed in Section I and indicate that our model can represent these challenged features with the mixture distributed latent variables, thus to verify the effectiveness of SGmVRNN.

D. Time Efficiency

Table IV shows the time efficiency of SGmVRMM in term of its training and testing time on the hardware platform introduced in the subsection VI-A. It can be seen from Table IV that SGmVRMM can perform anomaly detection for a sample within one-tenth second versus the data collecting interval of 60 seconds. Hence, SGmVRMM can be deployed in the manner of offline training and online detection [3], [5].

VII. Related Work

Anomaly detection for multivariate time series has been an active topic in data science. More recent studies have shifted from the traditional statistical-based anomaly detection [37], [38] to machine learning-based ones [39] that can be classified into two primary categories, i.e., supervised [40]–[42] or unsupervised [1]–[6], [31], [35], [43] methods. Due to the labor-intensive data labeling and lack of anomaly instances in real-world scenarios, supervised methods tend to become impractical. Hence, unsupervised deep anomaly detection has been widely investigated in recent years. Among them, one line of the research mainly focus on learning of the spatial characteristics in the multivariate metrics but ignore the temporal dependency across the varying timesteps [4], [6], [31]. Another kind of algorithm is seminal RNN-based anomaly detection, modelling the temporal characteristics via recurrent network structure [1]–[3], [5], [43]. Typically, MAD-GAN [1] employs an LSTM-RNN structured generative adversarial network framework to capture the normal temporal and spatial patterns, and OmniAnomaly [3] designs a stochastic-RNN (SRNN) model to help learn the robust representation. Recently, SDFVAE [5] introduced a static and dynamic factorized VAE-based framework to explicitly learn the time-invariant as well as time-varying characteristics.

Although the existing RNN-based anomaly detections have shown the effectiveness in some real-world scenarios, however, they cannot deal well with the anomaly detection for various CDN websites, especially in one model, due to the non-stationary temporal dependencies in the KPIs for an individual website as well as the varying characteristics in the KPIs for diverse websites. Compared with previous studies, SGmVRNN is the first anomaly detection method that can potentially capture the non-stationary temporal patterns and various characteristics of diverse websites simultaneously by designing a switching mechanism.

VIII. Conclusion

In this paper, we propose a switching Gaussian mixture variational RNN (SGmVRNN) to cope with the anomaly detection challenges that brought by the natural characteristics of multivariate CDN KPIs of diverse websites. SGmVRNN brings in a variational recurrent structure and assigns its latent variables into a mixture Gaussian distribution and thus it can model complex KPI time series and capture the diversely structural and dynamical characteristics within them. With a switching mechanism, SGmVRNN learns richer representations of KPIs. Furthermore, an upward-downward autoencoding inference method which combines the bottom-up likelihood and up-bottom prior information of the parameters for accurate posterior approximation is developed. The extensive experimental results demonstrate that SGmVRNN outperforms the state-of-the-art approaches in terms of F1-score and show the great superiority in one model that fits multiple websites.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China under Grants 61771469 and 61971382, the Industrial Internet Innovation and Development Project (TC200H030) and the Cooperation project between Chongqing Municipal undergraduate universities and institutes affiliated to CAS (HZ2021015).

REFERENCES

[1] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. Ng, "MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks," in *Artificial Neural Networks and Machine Learning - ICANN*, 2019, pp. 703–716.

[2] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *AAAI '19*, 2019, pp. 1409–1416.

[3] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *ACM SIGKDD*, 2019, pp. 2828–2837.

[4] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: unsupervised anomaly detection on multivariate time series," in *ACM SIGKDD '20*, 2020, pp. 3395–3404.

[5] L. Dai, T. Lin, C. Liu, B. Jiang, Y. Liu, Z. Xu, and Z.-L. Zhang, "SDFVAE: Static and dynamic factorized vae for anomaly detection of multivariate cdn kpis," in *WWW '21: The Web Conference*, 2021.

[6] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *ICLR '18*, 2018.

[7] H. Yan, J. Zhou, and C. K. Pang, "Gaussian mixture model using semisupervised learning for probabilistic fault diagnosis under new data categories," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 4, pp. 723–733, 2017.

[8] K. Honda and H. Ichihashi, "Regularized linear fuzzy clustering and probabilistic PCA mixture models," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 508–516, 2005.

[9] W. Chen, B. Chen, X. Peng, Y. Liu, Jiaqiand Yang, H. Zhang, and H. Liu, "Tensor RNN with bayesian nonparametric mixture for radar HRRP modeling and target recognition," *IEEE Trans. Signal Process.*, vol. 69, pp. 1995–2009, 2021.

[10] C. Chang and Athans, "State estimation for discrete systems with switching parameters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 14(3), no. 3, pp. 418–425, 1978.

[11] M. O. Sang, J. M. Rehg, T. Balch, and F. Dellaert, "Learning and inference in parametric switching linear dynamic systems," *in ICCV*, pp. 1161–1168, 2005.

[12] S. W. Linderman, A. C. Miller, and R. P. Adams, "Recurrent switching linear dynamical systems," *in AISTATS*, 2016.

[13] P. Becker Ehmck, J. Peters, and v. d. S. Patrick, "Switching linear dynamics for variational bayes filtering," in *ICML*, 2019, pp. 553–562.

[14] W. Chen, B. Chen, Y. Liu, Q. Zhao, and M. Zhou, "Switching poisson gamma dynamical systems," *in IJCAI*, pp. 2029–2036, 2020.

[15] C. Junyoung, K. Kyle, D. Laurent, G. Kratarth, C. C. Aaron, and B. Yoshua, "A recurrent latent variable model for sequential data," *in NeurIPS*, pp. 2980–2988, 2015.

[16] P. K. Diederik and W. Max, "Auto-encoding variational bayes," *in ICLR*, 2014.

[17] S. Aaron, M. W. Hanna, and M. Zhou, "Poisson gamma dynamical systems," *in Annual Conference on Neural Information Processing Systems*, pp. 5006–5014, 2016.

[18] D. Guo, B. Chen, H. Zhang, and M. Zhou, "Deep poisson gamma dynamical systems," *in NeurIPS*, pp. 8451–8461, 2018.

[19] L. Yang, N. Cheung, J. Li, and J. Fang, "Deep clustering by gaussian mixture variational autoencoders with graph embedding," in *ICCV*, 2019, pp. 6439–6448.

[20] C. Alexander, L. Yuan, and K. Diego, "Open-set recognition with gaussian mixture variational autoencoders," in *AAAI*, 2021, pp. 6877–6884.

[21] J. M. Chris, M. Andriy, and W. T. Yee, "The concrete distribution: A continuous relaxation of discrete random variables," in *ICLR*, 2017.

[22] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *in ICLR*, 2014.

[23] D. M. Blei and M. I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Analysis*, pp. 121–144, 2006.

[24] H. Zhang, B. Chen, D. Guo, and M. Zhou, "Whai: Weibull hybrid autoencoding inference for deep topic modeling," *in ICLR*, 2018.

[25] D. Guo, B. Chen, W. Chen, C. Wang, and M. Zhou, "Variational temporal deep generative model for radar hrrp target recognition," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5795–5809, 2020.

[26] Z. Duan, D. Wang, B. Chen, C. Wang, W. Chen, Y. Li, J. Ren, and M. Zhou, "Sawtooth factorial topic embeddings guided gamma belief network," in *ICML*, 2021, pp. 2903–2913.

[27] K. S. Casper, R. Tapani, M. Lars, K. S. Søren, and W. Ole, "Ladder variational autoencoders," in *NeurIPS*, 2016, pp. 3738–3746.

[28] W. Matthew, R. Stephen, and H. Chris, "Disentangling to cluster: Gaussian mixture variational ladder autoencoders," *in NeurIPS 2019 Workshop on Bayesian Deep Learning*, 2019.

[29] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther, "A disentangled recognition and nonlinear dynamics model for unsupervised learning," *in NeurIPS*, pp. 3601–3610, 2017.

[30] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Technical Report. SNU Data Mining Center*, pp. 1–8, 2015.

[31] Y. Su, Y. Zhao, M. Sun, S. Zhang, X. Wen, Y. Zhang, X. Liu, X. Liu, J. Tang, W. Wu, and D. Pei, "Detecting outlier machine instances through gaussian mixture variational autoencoder with one dimensional cnn," *IEEE Transactions on Computers*, pp. 1–1, 2021.

[32] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *WWW '18*, 2018, pp. 187–196.

[33] W. Chen, H. Xu, Z. Li, D. Pei, J. Chen, H. Qiao, Y. Feng, and Z. Wang, "Unsupervised anomaly detection for intricate kpis via adversarial training of VAE," in *IEEE INFOCOM*, 2019, pp. 1891–1899.

[34] A. Siffer, P. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *ACM SIGKDD '17*, 2017, pp. 1067–1075.

[35] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Söderström, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *ACM SIGKDD '18*, 2018, pp. 387–395.

[36] L. van der Maaten, "Accelerating t-sne using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, 2014.

[37] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[38] Y. Chen, R. Mahajan, B. Sridharan, and Z. Zhang, "A provider-side view of web search response time," in *ACM SIGCOMM '13*, 2013.

[39] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," 2020.

[40] D. Liu, Y. Zhao, H. Xu, Y. Sun, D. Pei, J. Luo, X. Jing, and M. Feng, "Opprentice: Towards practical and automatic anomaly detection through machine learning," in *ACM IMC '15*, 2015, pp. 211–224.

[41] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Inf. Sci.*, vol. 177, no. 18, pp. 3799–3821, 2007.

[42] M. Yamada, A. Kimura, F. Naya, and H. Sawada, "Change-point detection with feature selection in high-dimensional time-series data," in *IJCAI '13*, 2013, pp. 1827–1833.

[43] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," in *ICML 2016 Anomaly Detection Workshop*, 2016.