

Multi-view Deep Markov Models for Time Series Anomaly Detection

Phuong Nguyen

Department of Computer Science
New Mexico State University
Las Cruces, New Mexico, USA
ntphuong@nmsu.edu

Hiep Tran

Department of Computer Science
New Mexico State University
Las Cruces, New Mexico, USA
tuanhiep.tran88@gmail.com

Tuan M. V. Le

Department of Computer Science
New Mexico State University
Las Cruces, New Mexico, USA
tuanle@nmsu.edu

Abstract—We consider the problem of multi-view anomaly detection for multi-view time series data. This task aims to find time steps in time series instances that have inconsistent features across multiple views. To solve the problem, we propose a multi-view deep Markov model that can learn sequential structures in complex high-dimensional multi-view time series data. In our proposed model, each view is modeled by a sequence of latent states and the state transition function is shared across views. Therefore, the inconsistencies in views data will lead to inconsistencies in transitions between latent states across views, which makes the likelihood of abnormal time steps not high. We rely on that property to compute the multi-view anomaly score of each time step in all time series instances. The extensive experiments show that the proposed model is effective in detecting multi-view anomalies in time series data.

Index Terms—multi-view anomaly detection, time series, deep Markov models

I. INTRODUCTION

In multi-view data, an instance is represented by multiple views of distinct features that can be obtained from different sources. For example, an object can be described by cameras from different viewpoints; an article can have versions in various languages. As an example of multi-view time series data, a person's activity can be recorded by several sensors; each gives a view of that person's activity. Here sensors can be placed on a person's chest, right wrist, and left ankle to measure the motion experienced by various body parts. Learning from multi-view data is an emerging direction and it could improve performance in different tasks such as classification [1]–[3], clustering [4]–[6], and anomaly detection [7], [8].

In this paper, we consider the problem of multi-view anomaly detection for multi-view time series data. This task aims to find time steps in time series instances that have inconsistent features across multiple views. As an example, Figure 1 shows the difference between a multi-view time series anomaly and a single-view time series anomaly. In this figure, the three plots on the right show a three-view time series dataset. Single-view anomalies are time steps that are significantly distinct from rest of the data such as time steps at B or C in the figure. In contrast, consider time steps at A, its pattern in view three is inconsistent with the patterns at the same time steps in views one and two. More specifically, the values near A in views one and two are increasing steadily.

However, in view three, its values are decreasing at A. We can observe a similar inconsistency at D. These inconsistencies of features across three views indicate that time steps at A and D are abnormal. Our paper focuses on detecting this type of multi-view anomalies. More formally, we consider the following problem.

Problem definition. We are given a dataset $\mathcal{D}$ of N multi-view multivariate time series instances. Each instance $x \in \mathcal{D}$ is a multi-view time series that is a set of time series collected from multiple views, $x = \{x^{(1)}, \dots, x^{(V)}\}$, where V is the number of views and $x^{(v)} = [x_1^{(v)}, \dots, x_T^{(v)}] \in \mathbb{R}^{d_v \times T}$ denotes the time series observed in the v -th view. Here d_v is the dimension of view v and T is the length of the time series x . For multi-view time series anomaly detection, the objective is to determine whether an observation at time step t of each instance x , x_t , is a multi-view anomaly (i.e., whether it has inconsistent features across multiple views) or not.

To solve the problem, one possible approach is to treat it as a single-view anomaly detection problem. We can merge all views into a single view and apply existing time series anomaly detection methods [9]–[12]. However, since these models do not explicitly model the multiple views of time series, they may not be able to detect the inconsistencies across multiple views, as shown in the experiments. For multi-view anomaly detection, there have been several proposed methods [7], [13], [14]. However, since they are not working with time series data, we can pass time steps as independent data points to these models. Therefore, they may not be able to exploit time-dependency patterns to detect inconsistencies across views.

In this paper, we propose a novel approach that models each view using a latent state sequence via deep Markov models for multi-view time series anomaly detection. We argue that if views are consistent, the model could use a latent state sequence to explain the views because these views are about the same object. When there are inconsistencies between views, the model would need to use different latent state sequences to better explain views. For example, the three plots on the left of Figure 1 show the latent state sequences that explain the corresponding views. As we can see, since there are no inconsistencies between views one and two, they could be modeled using one state sequence. Contrarily, due to the

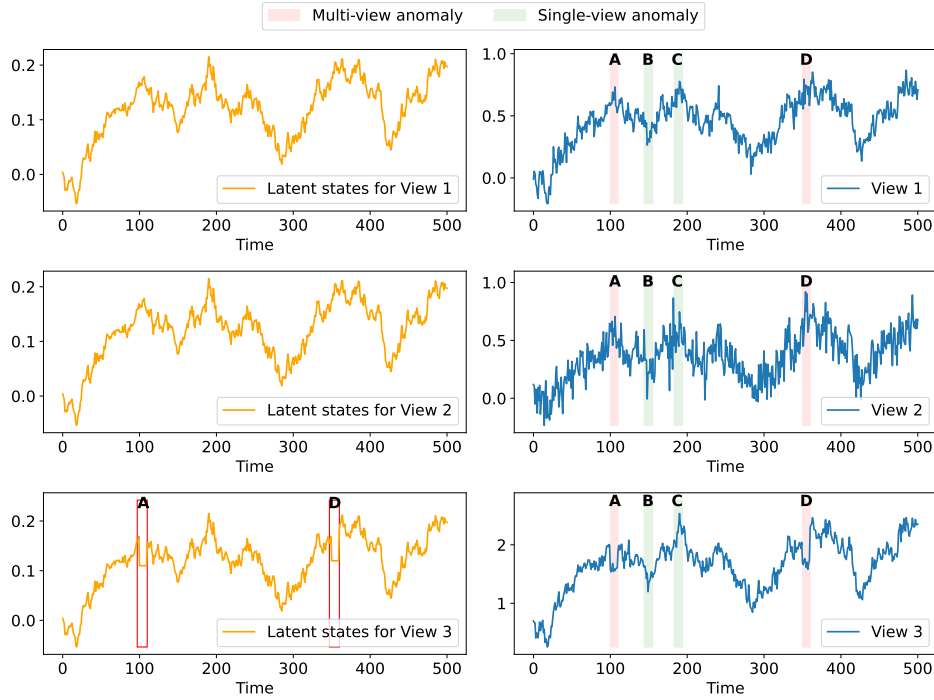


Fig. 1: Illustration of multi-view anomaly on time series.

inconsistencies at A and D in view three, to better explain that part of data, the latent states of view three at A and D (highlighted using red boxes) need to be different from the corresponding latent states of views one and two. Therefore, when we enforce the model to learn one latent state sequence for all views, the likelihood of data at inconsistent time steps will be low because the model cannot use one latent state sequence to explain views that exhibit inconsistencies.

To the best of our knowledge, our proposed model is the first to explicitly deal with multi-view anomaly detection on time series data. We propose a multi-view deep Markov model, called *ITime*, that can learn sequential structures in complex high-dimensional multi-view time series data. To realize the above idea and to link the views together for detecting multi-view anomalies, we use the same neural networks to calculate the means and variances for Gaussians in transitions of the deep Markov models for all views. In other words, views all share the same neural networks that generate their latent states, which encourages the model to learn similar latent state sequences for all views. Therefore, if there is a view that is inconsistent with other views at some time steps, the likelihood will be low at those inconsistent time steps, which we rely on to calculate the multi-view anomaly score. We summarize our contributions as follows:

- We propose a multi-view deep Markov model that can learn sequential structures in complex high-dimensional multi-view time series data for detecting abnormal time steps that exhibit inconsistencies across multiple views.
- To estimate the model parameters, we derive an algorithm based on the variational inference approach.

- We conduct extensive experiments several datasets. The results show that the proposed model is effective in detecting multi-view anomalies in time series data.

II. PRELIMINARIES

A. Gaussian State Space Models

Gaussian State Space Models (GSSMs) are widely used for modeling time series data [15], [16]. We denote a time series as $x = [x_1, \dots, x_T]$, $x_t \in \mathbb{R}^{d_x}$. Each observed x_t is associated with a latent variable z_t that generates/emits that observation at t . Denote the sequence of latent variables as $z = [z_1, \dots, z_T]$, where $z_t \in \mathbb{R}^{d_z}$. The generative process of GSSM is defined as follows:

$$z_t \sim \mathcal{N}(\mu_\alpha(z_{t-1}, u_t), \Sigma_\kappa(z_{t-1}, u_t)) \quad (\text{Transition}) \quad (1)$$

$$x_t \sim \Pi(\mu_\beta(z_t)) \quad (\text{Emission}) \quad (2)$$

where Π is the emission distribution whose parameters are determined by a possibly non-linear function μ_β of z_t and latent variable z_t follows a Gaussian distribution that is conditioned on previous z_{t-1} and input variable u_t . The mean and the covariance matrix of the Gaussian distribution are parameterized by possibly non-linear functions $\mu_\alpha(z_{t-1}, u_t)$ and $\Sigma_\kappa(z_{t-1}, u_t)$ respectively. When $\mu_\alpha(z_{t-1}, u_t)$, $\Sigma_\kappa(z_{t-1}, u_t)$, and μ_β of z_t are parameterized with deep neural networks, we will have deep Markov models. The parameters of the generative model are $\phi = (\alpha, \kappa, \beta)$.

B. Variational Inference

For learning parameters of GSSMs, we can optimize the following variational lower bound on the data marginal log likelihood:

$$\log p_\phi(x) \geq \mathbb{E}_{q_\theta(z|x)} [\log p_\phi(x|z)] - \text{KL}[q_\theta(z|x) \| p_\phi(z)] = \mathcal{L}(x; \phi, \theta) \quad (3)$$

where the variational distribution $q_\theta(z|x)$ is used to approximate the intractable posterior distribution $p(z|x)$. Since x_t depends only on z_t in Eq. 2, the expectation w.r.t $q_\theta(z|x)$, $\mathbb{E}_{q_\theta(z|x)} [\log p_\phi(x|z)]$, in Eq. 3 can be computed as follows:

$$\mathbb{E}_{q_\theta(z|x)} [\log p_\phi(x|z)] = \sum_{t=1}^T \mathbb{E}_{q_\theta(z_t|x)} [\log p_\phi(x_t|z_t)] \quad (4)$$

As shown in [17], the KL divergence term, $\text{KL}[q_\theta(z|x) \| p_\phi(z)]$, in Eq. 3 can be factorized as follows for time series data:

$$\text{KL}[q_\theta(z|x) \| p_\phi(z)] = \text{KL}[q_\theta(z_1|x) \| p_\phi(z_1)] + \sum_{t=2}^T \mathbb{E}_{q_\theta(z_{t-1}|x)} [\text{KL}[q_\theta(z_t|z_{t-1}, x) \| p_\phi(z_t|z_{t-1})]] \quad (5)$$

Since $p_\phi(z_t|z_{t-1})$ is a Gaussian distribution, the approximated posterior $q_\theta(z_t|z_{t-1}, x)$ should also be a Gaussian distribution, $q_\theta(z_t|z_{t-1}, x) = \mathcal{N}(\mu_\theta(z_{t-1}, x), \Sigma_\theta(z_{t-1}, x))$. Here, following the variational inference approach [18], the functions to compute the mean $\mu_\theta(z_{t-1}, x)$ and the variance $\Sigma_\theta(z_{t-1}, x)$ can be approximated by neural networks. By these parameterizations, the expectations in Eq. 4 and Eq. 5 can be approximated by Monte Carlo estimates [18].

III. PROPOSED MODEL

In this section, we present our proposed multi-view deep Markov model for multi-view anomaly detection on time series data, the inference algorithm, and how anomaly score is measured.

A. Generative Model of ITime

To model multiple views of time series, we assume that each view of the time series is generated by a sequence of latent states and the transition function parameters are shared across views. More specifically, following GSSM, we model a latent state at time step t in view v of time series instance n using a Gaussian distribution with a diagonal covariance matrix:

$$z_t^{(n,v)} \sim \mathcal{N}(\mu_z(z_{t-1}^{(n,v)}), \sigma_z^2(z_{t-1}^{(n,v)})) \quad (6)$$

here $\mu_z(z_{t-1}^{(n,v)})$ and $\sigma_z^2(z_{t-1}^{(n,v)})$ are non-linear functions to compute the mean and variance of the Gaussian distribution. We do not use the input variable u_t as in Eq. 1 because we assume that $z_t^{(n,v)}$ depends only on the latent state of the previous time step, $z_{t-1}^{(n,v)}$. Note that all latent states in multiple

TABLE I: Notations

Notation	Description
$x^{(n,v)} \in \mathbb{R}^{d_v \times T}$	the v^{th} view of the n^{th} instance
$\bar{x}^{(n,v)} \in \mathbb{R}^{d_v \times T}$	reconstruction of $x^{(n,v)}$
V	the number of views
N	the number of instances
d_v	the dimension of the v^{th} view
$z^{(n,v)} \in \mathbb{R}^{d_z \times T}$	the sequence of latent states
d_z	the dimension of latent state
$p_\phi(x_t^{(n,v)} z_t^{(n,v)})$	the emission
$p_\phi(z_t^{(n,v)} z_{t-1}^{(n,v)})$	the transition
$q_\theta(z_t^{(n,v)} z_{t-1}^{(n,v)}, x^{(n)})$	the variational distribution
ϕ	the generative parameters
θ	the variational parameters of q_θ
K	the latent state delay
$\oplus$	the concatenation
$\odot$	the element-wise multiplication

views are in the same latent space, i.e., $z_t^{(n,v)} \in \mathbb{R}^{d_z}, \forall v$. To link the views together, we let all views depend on the same $\mu_z(\cdot)$ and $\sigma_z(\cdot)$.

The observation $x_t^{(n,v)}$ at time step t in view v of time series instance n is then generated as follows:

$$x_t^{(n,v)} \sim \mathcal{N}(\mu_x^{(v)}(z_t^{(n,v)}), \sigma_x^{(v)}(z_t^{(n,v)})) \quad (7)$$

where we assume that the emission distribution Π is a Gaussian distribution with a diagonal covariance matrix. Its mean and variance are parameterized by two functions $\mu_x^{(v)}(z_t^{(n,v)})$ and $\sigma_x^{(v)}(z_t^{(n,v)})$ respectively.

As in deep Markov models, we represent the functions $\mu_z, \sigma_z^2, \mu_x, \sigma_x^2$ using neural networks. More specifically, for μ_z and σ_z^2 , we parameterize them using a Gated Transition Function [17] as follows :

$$g_t = \text{Sigmoid}(\text{MLP}_g(z_{t-1}^{(n,v)})) \quad (\text{Gating unit}) \quad (8)$$

$$h_t = \text{Linear}(\text{MLP}_h(z_{t-1}^{(n,v)})) \quad (\text{Proposed mean}) \quad (9)$$

where $\text{MLP}(\cdot)$ is a multilayer perceptron with ReLU activation function.

$$\mu_z(z_{t-1}^{(n,v)}) = (1 - g_t) \odot \text{Linear}(z_{t-1}^{(n,v)}) + g_t \odot h_t \quad (10)$$

$$\sigma_z^2(z_{t-1}^{(n,v)}) = \text{Softplus}(\text{Linear}(\text{ReLU}(h_t))) \quad (11)$$

where $\odot$ denotes element-wise multiplication. By using the Gated Transition Function, it allows some dimensions to use a linear transition while the other dimensions use a non-linear transition [17].

Since we want to connect multiple views through latent states in different views, we assume that the latent states in different views will have the same dimension. Therefore, we can let the μ_z and σ_z^2 are shared across views. Here as in [13], we assume that if the views are consistent, the latent states of views calculated from the shared transition function should be similar to each other. In other words, the inconsistencies in views will make the model learn inconsistent or not similar latent states among views. When maximizing the likelihood, these inconsistent latent states are likely not generated by

the same transition function. Therefore, the likelihood of the instance that has inconsistent views will be not high. We rely on this assumption to compute the multi-view anomaly score presented in Section III-D.

Similarly, for μ_x and σ_x^2 , we parameterize them as follows:

$$\mu_x^{(v)}(z_t^{(n,v)}) = \text{Linear}(\text{MLP}(z_t^{(n,v)})) \quad (12)$$

$$\sigma_x^{(v)2}(z_t^{(n,v)}) = \text{Softplus}(\text{MLP}(z_t^{(n,v)})), \quad (13)$$

Note that since the dimensions of views can be different, we have separate functions $\mu_x^{(v)}$ and $\sigma_x^{(v)}$ for each view v .

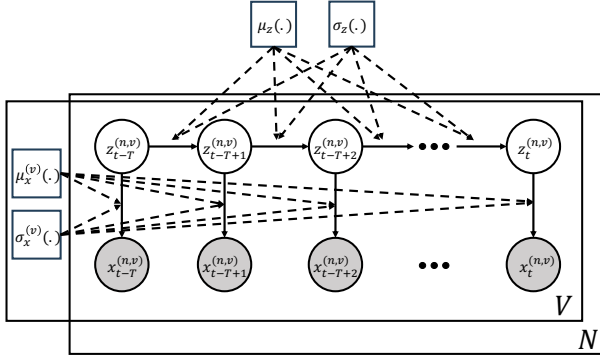


Fig. 2: The graphical model of *ITime*.

Given a dataset $\mathcal{D}$ of N multi-view multivariate time series instances, $\mathcal{D} = \{x^{(n)} | n = 1 \dots N\}$, the complete process of our proposed multi-view deep Markov model to generate $\mathcal{D}$ is described below. Its corresponding graphical model is shown in Figure 2.

For each instance: $n = 1 \dots N$

For each view: $v = 1 \dots V$

1) Draw sequence of latent states:

- a) $z_0^{(n,v)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- b) For each time step: $t = 1 \dots T$

$$z_t^{(n,v)} \sim \mathcal{N}(\mu_z(z_{t-1}^{(n,v)}), \sigma_z^2(z_{t-1}^{(n,v)}))$$

2) Draw the v^{th} view of the instance: $t = 1 \dots T$

$$x_t^{(n,v)} \sim \mathcal{N}(\mu_x^{(v)}(z_t^{(n,v)}), \sigma_x^{(v)2}(z_t^{(n,v)}))$$

B. Variational Inference

In this section, we present an algorithm to learn our proposed model based on variational inference. For the dataset $\mathcal{D}$ containing N multi-view multivariate time series instances, we aim to maximize the following variational lower bound on its marginal log likelihood:

$$\mathcal{L}(\mathcal{D}; \phi, \theta) = \sum_{n=1}^N \sum_{v=1}^V \mathcal{L}(x^{(n,v)}; \phi, \theta) \quad (14)$$

where $\mathcal{L}(x^{(n,v)}; \phi, \theta)$ is the variational lower bound on the marginal log likelihood of one instance n and its view v .

$$\begin{aligned} \mathcal{L}(x^{(n,v)}; \phi, \theta) &= \mathbb{E}_{q_\theta(z^{(n,v)} | x^{(n)})} [\log p_\phi(x^{(n,v)} | z^{(n,v)})] \\ &\quad - \text{KL}[q_\theta(z^{(n,v)} | x^{(n)}) \| p_\phi(z^{(n,v)})] \\ &= \sum_{t=1}^T \mathbb{E}_{q_\theta(z_t^{(n,v)} | x^{(n)})} [\log p_\phi(x_t^{(n,v)} | z_t^{(n,v)})] \\ &\quad - \text{KL}[q_\theta(z_1^{(n,v)} | x^{(n)}) \| p_\phi(z_1^{(n,v)})] \\ &\quad - \sum_{t=2}^T \mathbb{E}_{q_\theta} [\text{KL}[q_\theta(z_t^{(n,v)} | z_{t-1}^{(n,v)}, x^{(n)}) \| p_\phi(z_t^{(n,v)} | z_{t-1}^{(n,v)})]] \end{aligned} \quad (15)$$

here $q_\theta(z_t^{(n,v)} | z_{t-1}^{(n,v)}, x^{(n)})$ is parameterized using a neural network. Since latent states z_t form a sequence, we use a structured inference network for learning latent states [17]. For each time step t , the parameters of the variational distribution $q_\theta(z_t^{(n,v)} | z_{t-1}^{(n,v)}, x^{(n)})$ (i.e., mean $\mu_t^{\text{posterior}}$ and variance $\sigma_t^{2\text{posterior}}$) are calculated from the structured inference network. More specifically, we use a Bidirectional Recurrent Neural Network (BRNN) as shown in Figure 4. At each time step, we combine hidden states in the BRNN, h_t^{left} and h_t^{right} , into h_{combined} as follows:

$$h_{\text{combined}} = \frac{1}{3}(\tanh(\text{Linear}(z_{t-1}^{(n,v)})) + h_t^{\text{left}} + h_t^{\text{right}}) \quad (16)$$

The mean $\mu_t^{\text{posterior}}$ and variance $\sigma_t^{2\text{posterior}}$ are then computed as:

$$\mu_t^{\text{posterior}} = \text{Linear}(h_{\text{combined}}) \quad (17)$$

$$\sigma_t^{2\text{posterior}} = \text{Softplus}(\text{Linear}(h_{\text{combined}})) \quad (18)$$

The data input to the BRNN at time step t will include the data $x_t^{(n,v)}$ of the current view as well as the $x_t^{(n,v')}$ from other views. We combine the two inputs using F_1 and F_2 layers. Each $F(\cdot)$ is a multilayer perceptron (MLP). For F_1 , it will receive in the data of of current view $x_t^{(n,v)}$. For F_2 , it will receive in the flatten data of other views. The input to the BRNN at time step t , $e_t^{(n,v)}$, is then computed based on F_1 and F_2 as follows:

$$e_t^{(n,v)} = F_1^{(v)}(x_t^{(n,v)}) \oplus F_2^{(v)}\left(\left\{x_t^{(n,v')} \mid v' \in \{1 \dots V\} \setminus v\right\}\right). \quad (19)$$

After having $\mu_t^{\text{posterior}}$ and $\sigma_t^{2\text{posterior}}$, we sample the latent vectors for each view to calculate the Monte Carlo estimates of the expectations in Eq. 15¹. Given that the transition follows Gaussian distribution, the latent states are sampled as follows:

$$\begin{aligned} z_t^{(n,v)} &= \mu_t^{\text{posterior}}(z_{t-1}^{(n,v)}) + \sigma_t^{\text{posterior}}(z_{t-1}^{(n,v)}) \odot \epsilon_z \\ \epsilon_z &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \end{aligned} \quad (20)$$

The complete inference network architecture is shown in Figures 3 and 4. The final inference algorithm is presented in Algorithm 1.

¹We use a single sample when sampling the latent states to estimate the expectations in Eq. 15.

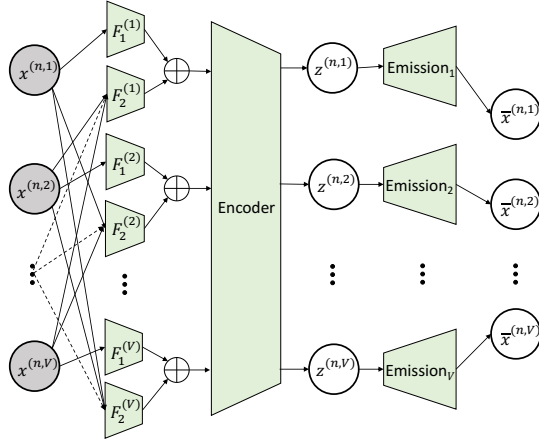


Fig. 3: Inference Network Structure of *ITime*.

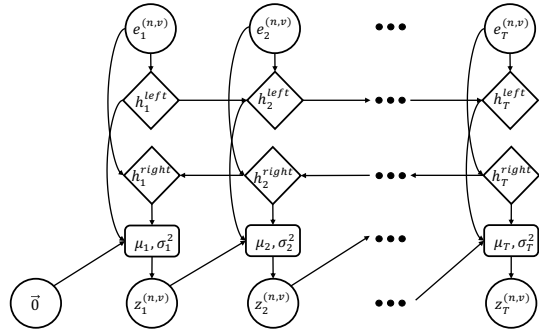


Fig. 4: Encoder Structure.

C. Complexity Analysis

In *ITime* inference, we parameterize all distributions by using neural networks. Hence, the heavy computations are essentially matrix multiplications. For analyzing the computational complexity, we analyze the complexity of matrix multiplication of each component at each time step of each view. We estimate the computational complexity per epoch of both KL-divergence and log likelihood in Eq. 15 as $\mathcal{O}(V \cdot T \cdot (N \cdot \max\{d_V, d_z, d_h\} \cdot \max\{d_V, d_z, d_h\}))$, where $d_V = \max\{d_v\}_{v=1}^V$ is the largest dimension of all input views, d_h is the largest dimension of all hidden layers of the neural networks. The part $(N \cdot \max\{d_V, d_z, d_h\} \cdot \max\{d_V, d_z, d_h\})$ is the upper bound of matrix multiplications of all components.

D. Anomaly Score

Based on our assumption that each view is derived from a sequence of latent states and normal time steps should have similar sequences of latent states across multiple views, for each time step of instance n , we define its multi-view anomaly score based on the conditional negative log likelihood of the time step as follows:

$$\text{score}(x_t^{(n)}) = - \sum_{v=1}^V \log p_\phi(x_t^{(n,v)} | z_t^{(n,v)}) \quad (21)$$

Algorithm 1 Learning a multi-view deep Markov model with stochastic gradient ascent.

Input: Data $\mathcal{D}$

Output: Parameters ϕ, θ

```

1: while not converged do
2:   Sample instance  $x^{(n)} \in \mathcal{D}$ 
3:   for  $v = 1 \dots V$  do
4:     Calculate embeddings  $e^{(n,v)}$ 
5:     Estimate posterior parameters  $\mu_z^{\text{posterior}}$  and  $\sigma_z^{2\text{posterior}}$ 
      (Eqs. 17, 18)
6:     Sample  $z^{(n,v)} \sim q_\theta(z^{(n,v)} | x^{(n)})$  (Eq. 20)
7:     Estimate prior transition  $\mu_z^{\text{prior}}$  and  $\sigma_z^{\text{prior}}$ 
8:     Estimate  $\mathbb{E}_{q_\theta} [\text{KL} [q_\theta(z^{(n,v)} | x^{(n)}) || p_\phi(z^{(n,v)})]]$ 
9:     Estimate prior emission  $\mu_x$  and  $\sigma_x$ 
10:    Estimate  $\mathbb{E}_{q_\theta} [\log p_\phi(x^{(n,v)} | z^{(n,v)})]$ 
11:    Evaluate  $\mathcal{L}(x^{(n,v)}; \phi, \theta)$  (Eq. 15)
12:  end for
13:  Evaluate ELBO  $\mathcal{L}(x^{(n)}; \phi, \theta)$ 
14:  Estimate gradient  $\Delta_\phi \mathcal{L}$  and  $\Delta_\theta \mathcal{L}$ 
15:  Update  $\phi$  and  $\theta$  using ADAM optimizer [19]
16: end while

```

here the likelihood of $x_t^{(n,v)}$ depends only on the latent state of that time step $z_t^{(n,v)}$. We argue that if the current time step is normal, it can also be explained by the latent states of the previous time steps because the near time steps should be consistent with each other. Therefore, we also compute Eq.21 with $z_{t-1}^{(n,v)}, z_{t-2}^{(n,v)}, \dots, z_{t-K+1}^{(n,v)}$ and sum up all results to obtain the score. Here K is called the latent state delay. More specifically,

$$\text{score}_K(x_t^{(n)}) = - \sum_{k=0}^{K-1} \sum_{v=1}^V \log p_\phi(x_t^{(n,v)} | z_{t-k}^{(n,v)}) \quad (22)$$

Intuitively, the higher the score is, the more abnormal that time step is. In Section IV-D2, we show the effects of K on the performance of our model.

IV. EXPERIMENTS

A. Datasets

For quantitatively evaluating the proposed model, we use three real time series datasets:

- Daily and Sport Activities (DSA)² contains motion sensor data of 19 daily and sports activities such as sitting, standing, walking, and running. Each activity is performed by 8 subjects. In total, there are 152 time series instances with length 7500. There are sensors on five body parts that are torso, right arm, left arm, right leg, and left leg. For each body part, there will 9 recorded motion features (e.g., accelerometers, gyroscopes, and magnetometers).

²<https://archive.ics.uci.edu/dataset/256/daily+and+sports+activities>

TABLE II: Dataset Statistics

Dataset	#Views	#Features	#Instances	Length
DSA	5	[9, 9, 9, 9, 9]	152	7500
MEx	2	[3, 3]	180	2924
MHealth	3	[5, 9, 9]	110	2096

Therefore, we have 5 views. Each view has 9 features corresponding to one body part.

- Multi-modal Exercise (MEx)³ contains data of different physiotherapy exercises performed by 30 subjects. We use 6 activities. There are two accelerometers on the wrist and the thigh. Each accelerometer records 3 features. Therefore, there are two views in this dataset, each has 3 features. In total, there are 180 time series instances with length 2924.
- Mobile Health (MHealth)⁴ contains body motion and vital signs recordings of 10 subjects while performing several physical activities such as climbing stairs, cycling, and jogging. We use 11 activities. Therefore, there are 110 time series instances with length 2096. Sensors are placed on the subject's chest, right wrist and left ankle to measure the motion experienced by diverse body parts. The sensor on the chest also provides 2-lead ECG measurements. Therefore, there are 3 views with 5, 9, and 9 features respectively.

Table II shows the size of each dataset after preprocessing. Following the experimental setting in [9], we divide each time series into two subsets of equal size. The first half is the training set and the second half is the testing set. Since we do not have the ground truth for multi-view anomalies, following [7], [13], [20], we add multi-view anomalies to the test set. More specifically, for each time series instance A corresponding to an activity performed by a subject, we randomly select another time series instance B for another different activity performed by the same subject. For an anomaly rate r (e.g., $r = 5\%, \dots, 20\%$), we select randomly r time steps from A . For each selected time step in A , we randomly choose a view, and replace the values at that time step in the selected view by values of the same time step from B . By this way, we generate multi-view anomalies because the injected time step in the injected view belongs to a different cluster (activity) as compared to the same time step in other views. We compute the anomaly scores for all time steps in all instances in the testing set. For the evaluation measurement, we use AUC (Area under the ROC curve) that is one of the most widely used performance metrics for anomaly detection problems. A higher AUC indicates a higher anomaly detection performance. We generate 10 samples with 4 anomaly rates for each dataset and report the averaged results.

B. Baselines

We compare the following state-of-the-art methods:

- LOF [21], OCSVM [22]: These are methods for single-view outlier detection. They are not designed for time series data. To run these methods, we merge all views into one single view and treat time steps as independent data points. Therefore, we lose the information on time dependencies between time steps. We use the implementations from scikit-learn⁵.
- Bayesian-MVAD [7]⁶, SRLSP [23]⁷: These are methods for multi-view anomaly detection. However, it cannot run with time series data. As above, we pass time steps as independent data points to these models.
- Omni [9]⁸, MTAD-GAT [10]⁹, TFAD [24]¹⁰, GANF [25]¹¹, TranAD [26]¹², Anomaly-Transformer [27]¹³, InterFusion [28]¹⁴: These models are proposed for single-view time series anomaly detection. As above, we merge features in all views into one single view.
- *ITime*¹⁵: This is our proposed model. To the best of our knowledge, our model is the first attempt to detect multi-view anomalies on time series data.

In our experiments, for *ITime* model, latent dimension is set to 20. For other methods, we set the latent dimension as default (i.e., 150 for MTAD-GAT, 3 for OmniAnomaly, and $\min(\{d_v | v \in V\}) - 1$ for Bayesian-MVAD). The number of epochs is set 100. Other hyperparameters are set as default for all baselines. We employ Adam optimizer with learning rate 10^{-3} for the training of our model. We run the methods on a system with 64GB memory, an Intel(R) Xeon(R) CPU E5-2623v3, 16 cores at 3.00GHz, and a GPU NVIDIA Quadro P2000 GPU with 5 GB GDDR5.

C. Multi-View Anomaly Detection

Figure 5 shows the average AUCs with different anomaly rates on DSA, MEx, and MHealth. In general, the performance of all methods decreases when the anomaly rate increases. *ITime* consistently outperforms the baselines in all settings across all datasets, which demonstrates the effectiveness of *ITime* in detecting multi-view anomalies on time series data. Among the single-view outlier detection methods for non-time series data, LOF has the best performance. Since LOF is a density-based method, it is able to detect the multi-view anomalies that deviate from main clusters when merging views into one. However, LOF's performance is lower than time series methods' because they can model the time dependencies in time series. Moreover, MVAD-GAT is able to detect multi-view anomalies more effectively than others single-view time series methods due to the explicit modeling

⁵<https://scikit-learn.org/stable>

⁶https://github.com/zwang-datascience/MVAD_Bayesian

⁷<https://github.com/wy54224/SRLSP>

⁸<https://github.com/NetManAI/Ops/OmniAnomaly>

⁹<https://github.com/ML4ITS/mtad-gat-pytorch>

¹⁰<https://github.com/DAMO-DI-ML/CIKM22-TFAD>

¹¹<https://github.com/EnyanDai/GANF>

¹²<https://github.com/imperial-qore/TranAD>

¹³<https://github.com/thuml/Anomaly-Transformer/tree/main>

¹⁴<https://github.com/zhhlee/InterFusion>

¹⁵<https://github.com/thanhphuong163/ITime>

³<https://archive.ics.uci.edu/dataset/500/mex>

⁴<https://archive.ics.uci.edu/dataset/319/mhealth+dataset>

TABLE III: Average AUCs with anomaly rate 5% on the three datasets. A higher AUC is better.

Method	DSA	MEx	MHealth
ITime	0.97008 $\pm$ 0.004	0.97156 $\pm$ 0.005	0.95469 $\pm$ 0.006
MVAD-GAT	0.92964 $\pm$ 0.006	0.95711 $\pm$ 0.003	0.93503 $\pm$ 0.006
Omni	0.88018 $\pm$ 0.007	0.89715 $\pm$ 0.011	0.86930 $\pm$ 0.010
TranAD	0.83492 $\pm$ 0.003	0.77981 $\pm$ 0.008	0.82937 $\pm$ 0.004
Anomaly-Transformer	0.90756 $\pm$ 0.005	0.94962 $\pm$ 0.004	0.88212 $\pm$ 0.010
InterFusion	0.90772 $\pm$ 0.009	0.90291 $\pm$ 0.011	0.87715 $\pm$ 0.010
GANF	0.80971 $\pm$ 0.007	0.85370 $\pm$ 0.008	0.82598 $\pm$ 0.007
TFAD	0.67346 $\pm$ 0.006	0.86997 $\pm$ 0.005	0.69058 $\pm$ 0.004
Bayesian-MVAD	0.67460 $\pm$ 0.007	0.53727 $\pm$ 0.015	0.62969 $\pm$ 0.011
SRLSP	0.94664 $\pm$ 0.003	0.87223 $\pm$ 0.008	0.92689 $\pm$ 0.003
LOF	0.80628 $\pm$ 0.010	0.61323 $\pm$ 0.005	0.84213 $\pm$ 0.010
OCSVM	0.55022 $\pm$ 0.006	0.53064 $\pm$ 0.007	0.56935 $\pm$ 0.013

of correlations between time series features. For SRLSP and Bayesian-MVAD, they are designed for multi-view non-time series data. Therefore, they are not as effective as some of time-series methods because SRLSP and Bayesian-MVAD do not model the time dependencies. However, since SRLSP is a multi-view anomaly detection method, SRLSP can produce better results in some settings when compared to some single-view time series methods. In contrast, since *ITime* is the first method designed for detecting multi-view anomalies on time series data, it consistently outperforms both single-view anomaly detection methods for time series data and multi-view anomaly detection methods for non-time series data. Table III shows detailed numbers of average AUCs of all methods with anomaly rate 5% on the three datasets.

D. Parameter Analysis

1) *Analysis of Latent Dimension*: We study the influence of the latent dimension on our proposed model. Figure 6 shows the average AUCs by our method with different latent dimensions. As we can see in the figure, the latent dimension has little effect on the performance of our method. The performance of *ITime* is a little bit better with latent dimensions 15 and 20 on MHealth dataset.

2) *Analysis of Latent State Delay on Anomaly Score*: In addition, we conduct an experiment to show the effects of the latent state delay K on the performance of our model. Figure 7 shows the AUC performance of *ITime* when varying K . As we can see from the figure, increasing the latent state delay from 0 to 3 can help improve the AUC significantly. The AUC is decreasing a little bit after that in DSA and MHealth datasets.

E. Running Time Analysis

Table IV shows running time by some of the strongest methods for time series data across all datasets. Compared to anomaly detection methods for time series data such as OmniAnomaly, and MTAD-GAT, *ITime* has a reasonable running time on large datasets, as theoretically shown in the complexity analysis in Section III-C.

F. A Use Case with Real-World Multi-View Time Series Data

In this section, we show an application of multi-view anomaly detection by *ITime*. Figure 8 shows S&P 500 index data of 5 economic sectors of the U.S. economy from January

TABLE IV: Training time (hh:mm:ss).

Method	DSA	MEx	Mhealth
ITime	15:17:37	1:26:39	1:00:11
MTAD-GAT	4:34:53	0:15:33	0:13:24
OmniAnomaly	17:00:20	1:44:50	1:11:39

1st, 2021 to December 31th, 2022¹⁶. The data measures the performance of the U.S. economy in 5 sectors including *Information Technology*, *Health Care*, *Consumer Staples*, *Energy*, and *Real Estate*. The data is collected from S&P 500 via Yahoo! Finance¹⁷. We treat each sector as a view to the U.S. economy and run *ITime* to detect whether there are any inconsistencies in the performance of the 5 views/sectors. Furthermore, we also run single-view MTAD-GAT model on this data by merging features in all views into one single view for comparing single-view and multi-view methods. In Figure 8, we plot anomaly scores by *ITime* and MTAD-GAT (the 2 plots on the top), and the time series of 5 views (the 5 plots at the bottom). We highlight top 10 (2%) time steps that have the highest anomaly scores detected by *ITime* and MTAD-GAT. For *ITime*, these top 10 anomalies concentrate at the two segments highlighted in red. For MTAD-GAT, its top 10 detected anomalies are highlighted in green.

As detected by *ITime*, the first abnormal segment is from March 2nd to March 8th, 2021. We can see an inconsistent performance between the *Energy* sector and the rest. There was a short drop, then an increase in the indices of *Information Technology*, *Health Care*, *Consumer Staples*, and *Real Estate* sectors. In contrast, *Energy* sector was increasing steadily. This could be partially explained by the start of the recovery after the COVID-19 pandemic. Around this time, the mass COVID-19 vaccinations and the economic stimulus packages¹⁸ boosted the U.S. consumer confidence and spending. However, the energy index kept increasing because the cold weather in February 2021 caused the natural gas price increase throughout the U.S.¹⁹. The second inconsistent segment is from December 29th, 2021 to January 5th, 2022. This period of time witnessed a rising trend losing its momentum in *Information Technology*, *Health Care*, *Consumer Staples*, and *Real Estate* sectors. The root of this situation could be because COVID-19 cases were surging due to the Omicron variant in the U.S.²⁰. Meanwhile, *Energy* sector still went up. This could partially be due to the energy crisis between Europe and Russia affecting the U.S. energy market. For MTAD-GAT model, it does not rank these inconsistencies high because it is a single-view method that aims to detect abnormal changes in the indices instead of pointing out the inconsistencies between views.

¹⁶Data is scaled by using StandardScaler

¹⁷<https://pypi.org/project/yfinance/>

¹⁸<https://www.cnn.com/2021/03/12/economy/march-consumer-sentiment-vaccines/index.html>

¹⁹<https://www.eia.gov/todayinenergy/detail.php?id=50798>

²⁰<http://bit.ly/cnbcmmicron>

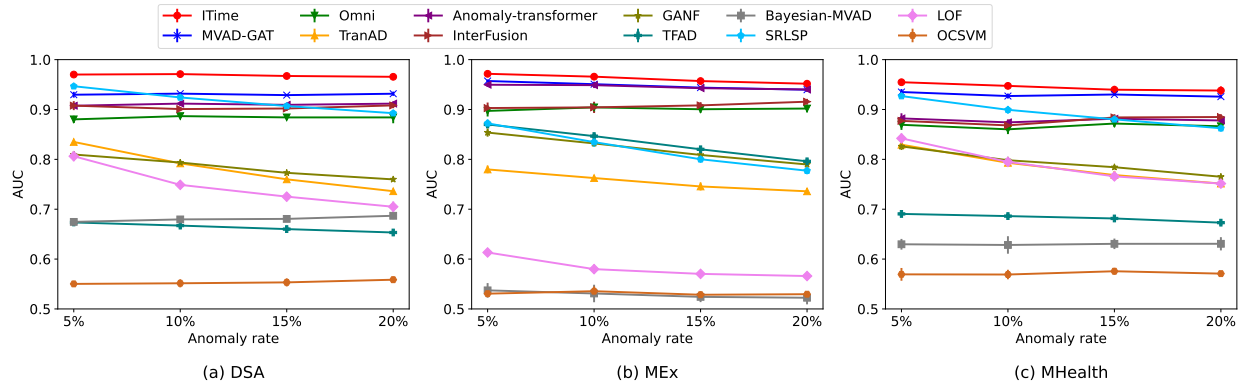


Fig. 5: Average AUCs with different anomaly rates on the three datasets. A higher AUC is better.

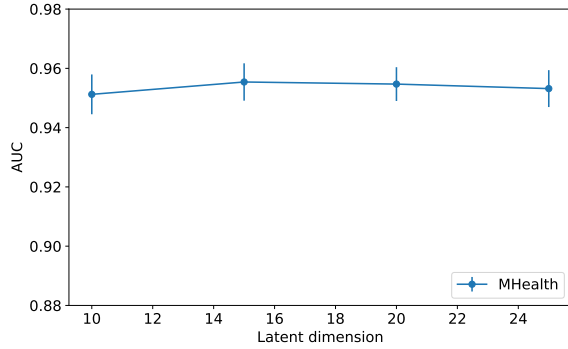


Fig. 6: Average AUCs by *ITime* with different latent dimensions.

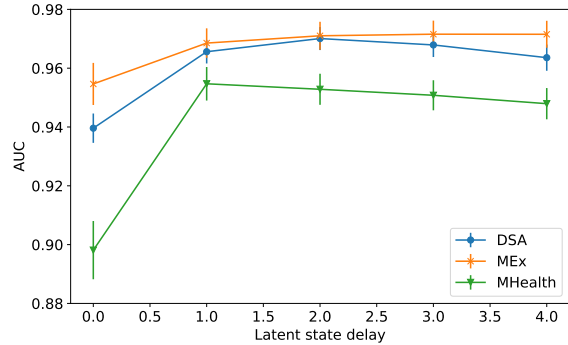


Fig. 7: Average AUCs by *ITime* with different latent state delays of anomaly score.

V. RELATED WORK

A. Non-time Series Anomaly Detection

There have been several methods for detecting anomalies in non-time series data. Most of them are single-view methods such as density-based methods [21], [29], clustering-based methods [30], [31], projection-based methods [32], [33], and deep learning based approaches [34]–[36]. Recently, there is an interest in anomaly detection for multi-view data whose objective is to detect the inconsistencies across multiple views [7], [8], [13], [20], [37]–[39]. [38] uses consensus clustering to detect multi-view anomalies. According to this method, data

points that do not belong to any consensus cluster are considered abnormal. [40] is another clustering-based method for detecting multi-view outliers. By comparing affinity vectors in different views derived from the clustering results, this method can identify anomalies on multi-view data. Another approach based on spectral clustering is HOrizontal Anomaly Detection (HOAD) [37]. This method computes the key eigenvectors from a combined similarity graph based on the similarity matrices. Anomaly score is obtained by calculating cosine distances of those eigenvectors. Multi-view low-rank analysis [39] is another approach to tackle this problem. The proposed method creates a cross-view low-rank coding to capture the intrinsic structures of the data. The multi-view anomaly score is calculated by the coefficients from the low-rank matrix.

There are other methods that rely on the assumption that all views of a normal instance should be generated from a single latent vector [6], [13]. Based on that assumption, [13] designs a probabilistic latent variable model that generates the data and calculates the probability that a multi-view instance is generated from more than one latent vector. This probability is used to determine if a data point is a multi-view anomaly or not. [6] also addresses this task based on this paradigm. A hierarchical Bayesian model is proposed to link the views of instances through a single, reduced-dimensionality latent space. Multi-view anomalies are identified by the negative Student's *t* density calculated from the learned parameters.

B. Time Series Anomaly Detection

Most of proposed anomaly detection methods for multi-variate time series are reconstruction-based models [9], [10], [27], [28], [41]–[45]. A reconstruction-based model learns the latent representation of the input time series and reconstructs that original input based on some of its latent variables. [41] proposes an LSTM-based Encoder-Decoder for multi-sensor anomaly detection. This method uses an LSTM-based encoder to encode the input time series into a vector representation, and an LSTM-based decoder to reconstruct the input from that vector. [42] detects anomalous instances by using a collective of autoencoders to differentiate between normal and abnormal instances. However, the previous deterministic

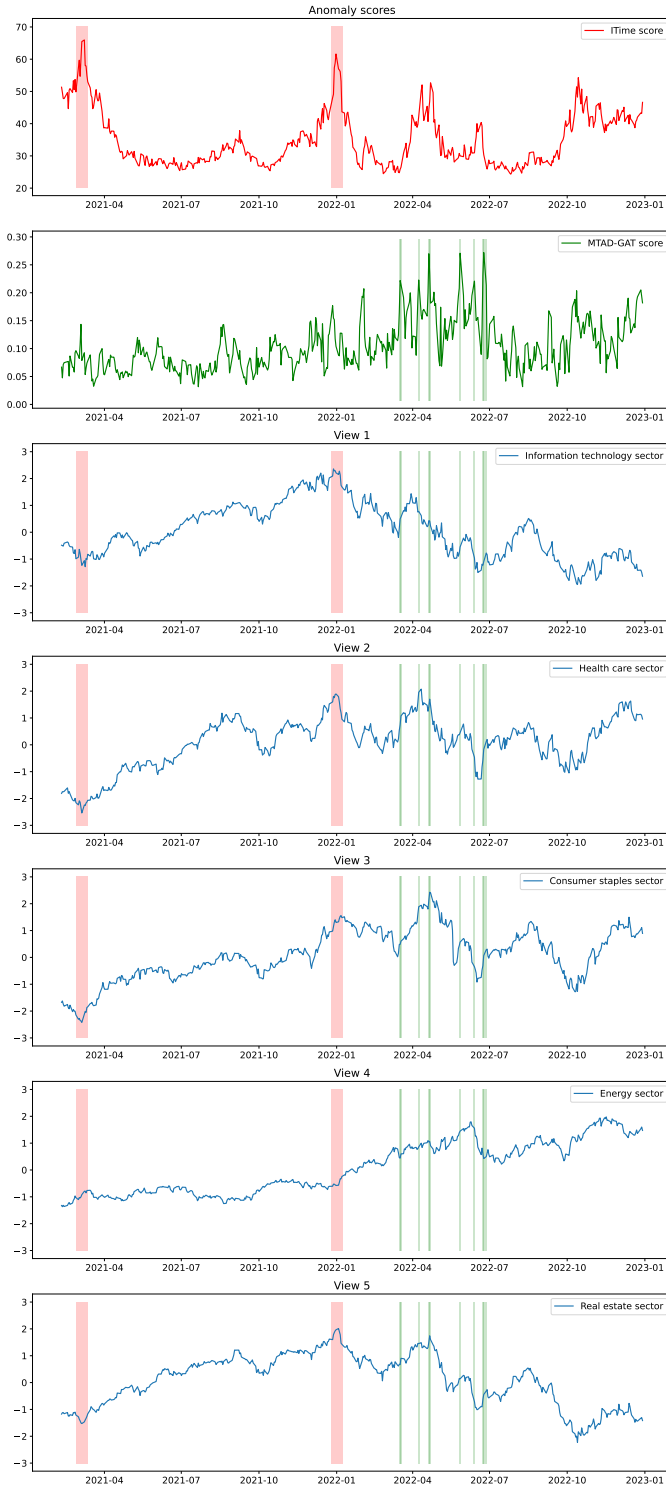


Fig. 8: An example of predicted multi-view anomalies on S&P 500 dataset by *ITime*.

methods are sensitive with unpredictable instances. Therefore, [9] integrates LSTM architecture to variational autoencoders for a robust learning representation. By introducing stochastic variables and planar normalizing flow, the model can capture the normal patterns of time series without being misled by

unpredictable samples. The model produces low reconstruction probability for anomalies. Anomaly-Transformer from [27] introduces Association Discrepancy which is a distinguishment between normal and abnormal time steps. By minimizing prior-association and maximizing series-association, the model can find the most informative association for association discrepancy, and combine it with reconstruction error to improve detection performance.

Furthermore, none of the above methods explicitly models the correlation between univariate time series features. [10] introduces the graph attention layer in the proposed MTAD-GAT framework for modeling the relationship between time series attributes and the temporal dependencies within each univariate time series attribute. In addition, MTAD-GAT model uses the latent representation from a GRU layer to reconstruct the original input and forecast its future values. This approach calculates anomaly score via a weighted sum of both the reconstruction probability from its reconstruction model and the forecasting error from its forecasting model. InterFusion from [28] renovates hierarchical Variational Auto-Encoder with two latent variables that learn low-dimensional inter-metric and temporal dependency among multivariate time series simultaneously.

The above models are not designed for multi-view time series data. The most relevant work to our proposed method is MTHL [46] that aims to find anomalous patterns from dynamic network systems. MTHL uses multi-view learning framework where edge and node properties are considered as two distinct views. The model projects these two views into a shared latent space and learns a hypersphere boundary containing all latent embeddings of normal multi-view instances. If the distances of those projections are greater than the boundary's radius, it is considered as anomaly. Although MTHL can model multi-view time series based on edge and node properties, it is designed for detecting anomalies in dynamic network systems and only work with two-view time series data. Moreover, MTHL aims to detect consistent irregular patterns in multiple views which is the second type of multi-view anomaly discussed above.

VI. CONCLUSION

We propose *ITime*, a multi-view deep Markov model, for detecting multi-view anomalies in multi-view time series data. To link multiple views of time series, we assume that each view of the time series is generated by a sequence of latent states and the transition function parameters are shared across views. By learning the latent state sequences of views that are generated by the shared neural networks, our model can detect abnormal time steps that exhibit inconsistencies across multiple views of time series data. We derive a variational inference algorithm to estimate the parameters of our proposed model. The extensive experiments on several datasets show that the *ITime* model outperforms state-of-the-art baselines significantly in detecting multi-view anomalies.

ACKNOWLEDGMENT

This research is sponsored by NSF #1757207 and NSF #1914635.

REFERENCES

- [1] Z. Han, C. Zhang, H. Fu, and J. T. Zhou, "Trusted multi-view classification," in *International Conference on Learning Representations*, 2020.
- [2] S. Li, Y. Li, and Y. Fu, "Multi-view time series classification: A discriminative bilinear projection approach," in *Proceedings of the 25th ACM international conference on information and knowledge management*, 2016, pp. 989–998.
- [3] Y. Bai, L. Wang, Z. Tao, S. Li, and Y. Fu, "Correlative channel-aware fusion for multi-view time series classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6714–6722.
- [4] S. Huang, Y. Ren, and Z. Xu, "Robust multi-view data clustering with multi-view capped-norm k-means," *Neurocomputing*, vol. 311, 2018.
- [5] H. Wang, Y. Yang, and B. Liu, "Gmc: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2019.
- [6] M.-S. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 3513–3520.
- [7] Z. Wang and C. Lan, "Towards a hierarchical bayesian model of multi-view anomaly detection," pp. 2420–2426, 7 2020, main track. [Online]. Available: <https://doi.org/10.24963/ijcai.2020/335>
- [8] X.-R. Sheng, D.-C. Zhan, S. Lu, and Y. Jiang, "Multi-view anomaly detection: Neighborhood in locality matters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019.
- [9] 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 *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019.
- [10] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Multivariate time-series anomaly detection via graph attention network," in *2020 IEEE International Conference on Data Mining (ICDM)*, 2020, pp. 841–850.
- [11] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, 2018.
- [12] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3395–3404.
- [13] T. Iwata and M. Yamada, "Multi-view anomaly detection via robust probabilistic latent variable models," *Advances in neural information processing systems*, vol. 29, 2016.
- [14] J. Gao, W. Fan, D. Turaga, S. Parthasarathy, and J. Han, "A spectral framework for detecting inconsistency across multi-source object relationships," in *2011 IEEE 11th International Conference on Data Mining*, IEEE, 2011, pp. 1050–1055.
- [15] D. Xu, "Learning nonlinear state space models with hamiltonian sequential monte carlo sampler," *arXiv preprint arXiv:1901.00862*, 2019.
- [16] T. Raiko and M. Tornio, "Variational bayesian learning of nonlinear hidden state-space models for model predictive control," *Neurocomputing*, vol. 72, no. 16–18, pp. 3704–3712, 2009.
- [17] R. Krishnan, U. Shalit, and D. Sontag, "Structured inference networks for nonlinear state space models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 09 2016.
- [18] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR*, 2014.
- [19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [20] K. Li, S. Li, Z. Ding, W. Zhang, and Y. Fu, "Latent discriminant subspace representations for multi-view outlier detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [21] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.
- [22] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [23] Y. Wang, C. Chen, J. Lai, L. Fu, Y. Zhou, and Z. Zheng, "A self-representation method with local similarity preserving for fast multi-view outlier detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 1, pp. 1–20, 2023.
- [24] C. Zhang, T. Zhou, Q. Wen, and L. Sun, "Tfad: A decomposition time series anomaly detection architecture with time-frequency analysis," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2497–2507.
- [25] E. Dai and J. Chen, "Graph-augmented normalizing flows for anomaly detection of multiple time series," *arXiv preprint arXiv:2202.07857*, 2022.
- [26] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *arXiv preprint arXiv:2201.07284*, 2022.
- [27] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *International Conference on Learning Representations*, 2022.
- [28] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, and D. Pei, "Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021.
- [29] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, 2017.
- [30] M.-F. Jiang, S.-S. Tseng, and C.-M. Su, "Two-phase clustering process for outliers detection," *Pattern recognition letters*, vol. 22, no. 6–7, pp. 691–700, 2001.
- [31] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern recognition letters*, vol. 24, no. 9–10, pp. 1641–1650, 2003.
- [32] T. Pevný, "Loda: Lightweight on-line detector of anomalies," *Machine Learning*, vol. 102, no. 2, pp. 275–304, 2016.
- [33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth IEEE international conference on data mining*. IEEE, 2008, pp. 413–422.
- [34] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, pp. 90–98.
- [35] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 665–674.
- [36] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International conference on data mining (ICDM)*. IEEE, 2018, pp. 727–736.
- [37] J. Gao, W. Fan, D. Turaga, S. Parthasarathy, and J. Han, "A spectral framework for detecting inconsistency across multi-source object relationships," in *2011 IEEE 11th International Conference on Data Mining*, 2011, pp. 1050–1055.
- [38] A. Y. Liu and D. N. Lam, "Using consensus clustering for multi-view anomaly detection," in *2012 IEEE Symposium on Security and Privacy Workshops*, 2012, pp. 117–124.
- [39] S. Li, M. Shao, and Y. Fu, "Multi-view low-rank analysis with applications to outlier detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 3, pp. 1–22, 2018.
- [40] A. Alvarez, M. Yamada, A. Kimura, and T. Iwata, "Clustering-based anomaly detection in multi-view data," 10 2013, pp. 1545–1548.
- [41] T. SHROFF, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.
- [42] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.
- [43] D. Li, D. Chen, J. Goh, and S.-k. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," *arXiv preprint arXiv:1809.04758*, 2018.
- [44] H. Xu, Y. Wang, S. Jian, Q. Liao, Y. Wang, and G. Pang, "Calibrated one-class classification for unsupervised time series anomaly detection," *arXiv preprint arXiv:2207.12201*, 2022.
- [45] R. Wang, C. Liu, X. Mou, K. Gao, X. Guo, P. Liu, T. Wo, and X. Liu, "Deep contrastive one-class time series anomaly detection," in *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 2023, pp. 694–702.
- [46] X. Teng, Y.-R. Lin, and X. Wen, "Anomaly detection in dynamic networks using multi-view time-series hypersphere learning," in *The 26th ACM International Conference on Information and Knowledge Management (CIKM 2017)*. ACM, November 2017.