# Multivariate Time Series Anomaly Detection and Interpretation using Hierarchical Inter-Metric and Temporal Embedding

Zhihan Li*
Tsinghua University; BNRist
lizhihan17@mails.tsinghua.edu.cn

Youjian Zhao
Tsinghua University; BNRist

Jiaqi Han
Tsinghua University

Ya Su
Tsinghua University; BNRist

Rui Jiao
Tsinghua University

Xidao Wen
Tsinghua University; BNRist

Dan Pei†
Tsinghua University; BNRist

## ABSTRACT

Anomaly detection is a crucial task for monitoring various status (*i.e.*, metrics) of entities (*e.g.*, manufacturing systems and Internet services), which are often characterized by multivariate time series (MTS). In practice, it's important to precisely detect the anomalies, as well as to interpret the detected anomalies through localizing a group of most anomalous metrics, to further assist the failure troubleshooting. In this paper, we propose *InterFusion*, an unsupervised method that simultaneously models the inter-metric and temporal dependency for MTS. Its core idea is to model the normal patterns inside MTS data through hierarchical Variational Auto-Encoder with two stochastic latent variables, each of which learns low-dimensional inter-metric or temporal embeddings. Furthermore, we propose an MCMC-based method to obtain reasonable embeddings and reconstructions at anomalous parts for MTS anomaly interpretation. Our evaluation experiments are conducted on four real-world datasets from different industrial domains (three existing and one newly published dataset collected through our pilot deployment of *InterFusion*). *InterFusion* achieves an average anomaly detection F1-Score higher than 0.94 and anomaly interpretation performance of 0.87, significantly outperforming recent state-of-the-art MTS anomaly detection methods.

## CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**; *Neural networks*; *Bayesian network models*; *Latent variable models*.

## KEYWORDS

Anomaly Detection; Multivariate Time Series; Hierarchical Structure; Inter-metric and Temporal Embedding

*BNRist: Beijing National Research Center for Information Science and Technology
†Dan Pei is the corresponding author. Email: peidan@tsinghua.edu.cn


*KDD '21, August 14–18, 2021, Virtual Event, Singapore*

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467075

**ACM Reference Format:**
Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. 2021. 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 and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467075

## 1 INTRODUCTION

Anomaly detection has been widely studied in different domains [5] (*e.g.*, images, time series, graphs, etc.), aiming at finding data instances that significantly deviate from the other observations in the same dataset [11]. In this paper, we mainly focus on the anomaly detection for multivariate time series data (MTS for short hereafter), which has been an active research topic in the SIGKDD community these years [2, 14, 30], and is widely used to monitor the status (*i.e.*, metrics) of *entities* (*e.g.*, systems, services) in the application domain of manufacturing industry and Information Technology (IT) systems [14, 19, 20, 22, 30, 34].

Conventionally, domain experts manually establish static thresholds for each monitored metric [2] (*e.g.*, the volume of transactions, CPU utilization) for anomaly detection in industry. However, this process can be labor-intensive for a large number of metrics as the scale and the complexity of data grow exponentially over the years. To tackle this problem, many anomaly detection algorithms have been developed for univariate time series [23, 33], where the anomalies are detected mainly based on **one** specific metric. However, for a complex real-world system, the monitoring metrics are often interacted with each other due to their intrinsic connections (*e.g.*, a group of monitoring metrics for an application server, related sensors in a water treatment plant). Thus, single univariate time series **cannot** well represent the system's overall status and naively combining the anomaly detection results of several univariate time series has performed poorly for MTS anomaly detection [30].

Formally, MTS consists of a group of univariate time series (*i.e.*, *metric*), each of which describes a different part or attribute of a complex entity. Thus, it not only has *intra-metric temporal dependency* (called **temporal dependency** for short, which characterizes the inherent patterns like periodicity within each metric), but also has *inter-metric dependency* within an entity (called **intermetric dependency** for short, which represents the linear or nonlinear relationships among all metrics of an entity at each time period).

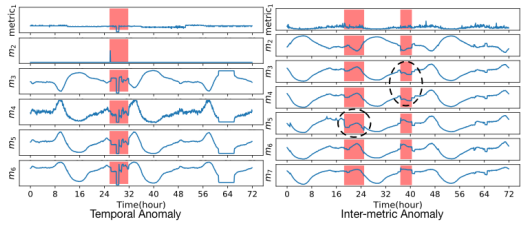


**Figure 1: Illustration of two types of anomalies (anomalous segments highlighted in pink). In a temporal anomaly, several metrics deviate from their corresponding temporal patterns. In an inter-metric anomaly, the relationships among several metrics violate the historical patterns.**

Violating these dependencies would cause different anomalies in MTS, as shown in Fig. 1. To help the system operators monitor the MTS metrics of complex systems, the algorithm should characterize the MTS anomalies from two perspectives. *i.e.*, it should detect *when* the anomalies happen in a system and *interpret* the detected anomalies through localizing a group of most anomalous metrics.

Recent methods for MTS anomaly detection can be roughly divided into two classes: prediction-based and reconstruction-based. Prediction-based methods [13, 14] try to predict the normal values of metrics based on historical data and detect anomalies according to prediction error, but some metrics might be inherently unpredictable in complex real-world systems [20]. Reconstruction-based methods learn low-dimensional representations and reconstruct the "normal patterns" of data and detect anomalies according to reconstruction error. However, existing such methods either use simple deterministic approaches [20, 34], thus are weak in modeling the *intermetric dependency*, or are poor at modeling the *temporal dependency* due to the lack of low-dimensional representations along time dimension [2, 19, 22, 30]. As a result, these methods can have difficulties in detecting intermetric or temporal anomalies.

Based on the observation above, for **MTS anomaly detection**, our core idea is to *explicitly learn the low-dimensional intermetric and temporal representations with properly designed structures* to better capture the normal patterns of MTS. However, this idea faces two main challenges. First, independently learning the intermetric and temporal embeddings can make the feature fusion hard, while using a traditional hierarchical method for images [31] would make the learned intermetric embeddings inconsistent with the temporal order for MTS data. Second, the risk of overfitting to potential anomalies in real-world data brings extra challenges to structure design. To address the first challenge, our method *InterFusion* proposes to use: (1) a hierarchical Variational Auto-Encoder (HVAE) with two stochastic latent variables that learns the low-dimensional intermetric and temporal embeddings, respectively; and (2) two-view embedding in which we compress the MTS along time dimension and metric dimension of the data space by making use of an auxiliary "reconstructed input". To address the second challenge, *InterFusion* proposes a prefiltering strategy where some temporal anomalies are filtered out through an embedding-reconstruction procedure to enable learning flexible and accurate intermetric embeddings.

To make detection results useful to users, it is important to **interpret each detected anomaly through localization**, *i.e.*, to find a group of most anomalous metrics for each MTS anomaly (*i.e.*, entity anomaly) [30, 34]. However, the anomalies can affect the estimation of reconstructions at *all* dimensions (anomalous or not) [15]. Thus using raw reconstruction scores at anomalous points as an interpretation [30, 34] may cause misinterpretations. To obtain accurate interpretation, it's important to obtain reasonable embeddings and reconstructions at anomalous points, which *reflect the normal patterns they should have followed*. Thus, *InterFusion* proposes an MCMC-based anomaly interpretation method, which iteratively applies MCMC imputation [25] to address the above problem. Moreover, instead of using naive point-wise metric [30], we define a new segment-wise metric to better evaluate the anomaly interpretation accuracy for MTS, which is consistent with the preference of the real-world users (*e.g.*, system operators).

The contributions of this paper are summarised as follows:

- To the best of our knowledge, our proposed *InterFusion* is the first MTS anomaly detection algorithm that employs HVAE with explicit low-dimensional inter-metric and temporal embeddings to jointly learn robust MTS representations. We use three designs, hierarchical structure, two-view embedding and prefiltering strategy, to tackle the challenges for learning normal MTS patterns for anomaly detection.
- We propose a novel anomaly interpretation method based on MCMC imputation multivariate time series, and define a new segment-wise metric consistent with the system operators' preferences to quantitatively evaluate the anomaly interpretation results for real-world data.
- We evaluate *InterFusion* on four real-world MTS datasets from different industrial domains (three existing and one newly published dataset collected through our pilot deployment of *InterFusion*). *InterFusion* achieves an overall best F1-Score higher than 0.94 and overall interpretation accuracy of 0.87, outperforming the state-of-the-art methods by at least 0.04 and 0.07, respectively. Ablation studies further demonstrate the effectiveness of our proposed structure design choices for MTS anomaly detection. Our feasibility study with application servers' monitoring data from a large Internet company showed that *InterFusion* meets the company's requirements on MTS anomaly detection and interpretation. We publish our code and data (link in Appendix C.1) for better reproducibility.

## 2 PRELIMINARIES

### 2.1 Problem Statement

MTS contains successive observations with equal-spaced sampling, as shown in Fig. 2. $\mathbf{x} \in R^{M \times N}$, where $M$ and $N$ are the number of metrics and data length of the MTS, respectively. Take a Web application server as an example, the metrics might include CPU utilization, memory utilization, TCP active opens, *etc.*, and the whole entity (*i.e.*, MTS) characterizes the server status. MTS data often has *temporal dependency* within each metric (*e.g.*, the periodicity of CPU utilization), as well as *intermetric dependency* among different metrics (*e.g.*, the positive correlation among packets transmitted per second, TCP active opens, and CPU utilization). To take the contextual information into consideration, we use a sliding window of length $W$ over the MTS to calculate the anomaly results.

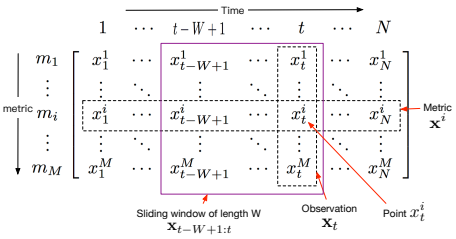


**Figure 2: Data formulation of MTS $\mathbf{x} \in R^{M \times N}$. Each row $\mathbf{x}^i$ is called a metric, while each column $\mathbf{x}_t$ is an observation.**

The goal of MTS anomaly detection is to determine whether an observation $\mathbf{x}_t$ is an anomaly or not, while anomaly interpretation is achieved through localization, *i.e.*, finding a group of metrics $\{\mathbf{x}^1, \ldots, \mathbf{x}^i\}, i \leq M$ that are most anomalous for each detected entity anomaly $\mathbf{x}_t$.

## 2.2 Anomaly Types

For the monitoring metrics (MTS) in a complex system, anomalies can be roughly classified into three types: *temporal anomalies*, *intermetric anomalies*, and *intermetric-temporal anomalies*. Fig. 1 shows the first two types. In a temporal anomaly, several metrics significantly deviate from their corresponding historical normal patterns, which often indicates a system-level failure or rebooting. In an intermetric anomaly, most metrics roughly follow their corresponding normal patterns, but the patterns of the *linear or nonlinear relationships* among the metrics violate the historical patterns. For example, in the right half of Fig. 1, the historical pattern is that metrics $m_3$ and $m_4$ positively correlate with $m_5, m_6, m_7$, but this pattern is violated in the second vertical strip. This often indicates an anomalous behavior in some parts of the system, which caused local fluctuations. In an intermetric-temporal anomaly, both intermetric and temporal dependencies are violated, thus most of them are actually easier to be detected from either a temporal or metric perspective. No matter severe or subtle, each type of anomalies can indicate a potential problem in the system. Thus, precisely detecting such MTS anomalies is urgently needed for system operators.

# 3 DESIGN OF *INTERFUSION*

## 3.1 Motivation and Overview of *InterFusion*

For MTS data in real-world systems, the key to precisely detect and interpret anomalies is to find the *normal patterns* of MTS. As discussed in Section 2.2, violating either intermetric or temporal dependencies could cause anomalies (Fig. 1), which shows the importance of *explicitly modeling both dependencies* for characterizing the normal patterns of MTS. However, previous works [2, 14, 19, 30, 34] mainly model one of the dependencies (*e.g.*, intermetric dependency, through latent metric embedding), *which could limit their capability on learning normal MTS patterns and detecting anomalies*, especially those violating the other kind of dependency (*e.g.*, temporal dependency, see Table 5). Moreover, most of them *lack precise interpretation for detected anomalies*, which is also useful for users (*e.g.*, to accelerate troubleshooting, to explain detection results to users).

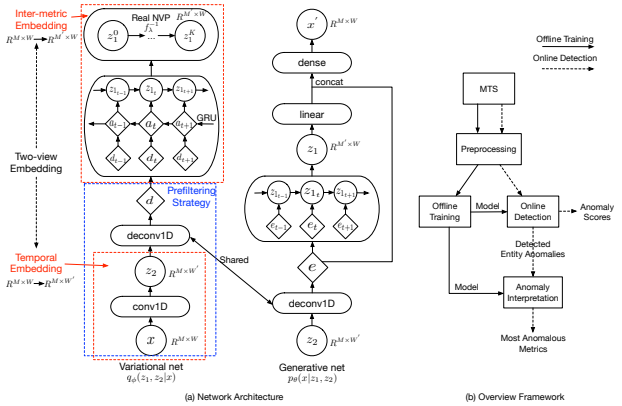


**Figure 3: (a) Network architecture. Circles are stochastic variables, diamonds are deterministic variables, rounded rectangles are layers. (b) Overview Framework. Solid lines are offline training, dashed lines are online detection.**

The observations above motivate us to *jointly* learn both embeddings with novel structures in *latent space*, which enables learning rich representations combing intermetric and temporal information to model the normal patterns of MTS. Furthermore, to enhance *InterFusion*'s applicability on real-world data, the detection method should be robust to the potential anomalies in training data to avoid overfitting to anomalies. A proper anomaly interpretation method should be designed for detection phase to give richer information about detected MTS anomalies for system operators to assist troubleshooting. The overview framework is shown in Fig. 3(b).

## 3.2 Network Architecture

The **core idea** of *InterFusion* is to model the MTS using HVAE with *jointly trained hierarchical stochastic latent variables*, each of which *explicitly learns low-dimensional intermetric or temporal embeddings*. We adopt the HVAE [28, 31] to jointly train the intermetric and temporal latent variables through a **hierarchical structure**, while proposing a **two-view embedding** for MTS data to derive such latent variables characterizing the intermetric and temporal dependencies. Moreover, we propose a **prefiltering strategy** to obtain intermetric embeddings while being robust to potential anomalies in training data. The network architecture is shown in Fig. 3(a).

**Hierarchical structure.** The generative model in Fig. 3(a) can be factorized as: $p_\theta(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2) = p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2)p_\theta(\mathbf{z}_1|\mathbf{z}_2)p_\theta(\mathbf{z}_2)$. The stochastic latent variables, $\mathbf{z}_1$ and $\mathbf{z}_2$, are designed to learn the low-dimensional intermetric and temporal embeddings, respectively. We apply the *hierarchical structure* in HVAE to represent the high-dimensional input $\mathbf{x}$ with hierarchies of low-dimensional $\mathbf{z}$ (*i.e.*, $\mathbf{z}_1|\mathbf{z}_2$ in *InterFusion*), making the intermetric embedding be aware of the learned temporal information, rather than learning the latent variables independently. This can ease the model training [32] to fuse the learned embeddings and capture the "normal patterns" of MTS in latent space.

**Two-view embedding.** To characterize the intermetric and temporal dependencies in MTS, we propose a *two-view embedding*, which compresses the MTS along time and metric dimensions of

the data space to obtain the intermetric and temporal embeddings, by leveraging an auxiliary "reconstructed input" $\mathbf{d}$. Specifically, $\mathbf{z}_2 = f(\mathbf{x}_{1:W}) \in R^{M \times W^{'}}$, where $f$ is several Conv1D [18] layers applied along the time dimension of input window $\mathbf{x}$ to learn the temporal embeddings $\mathbf{z}_2$. $W^{'}$ is the compressed window length. $\mathbf{d}_{1:W} = g(\mathbf{z}_2) \in R^{M \times W}$, and $g$ is the corresponding Deconv1D layers applied on $\mathbf{z}_2$ to reconstruct the input. Intermetric embeddings $\mathbf{z}_1 \in R^{M^{'} \times W}$ is then derived along the metric dimension of $\mathbf{d}$ with an SRNN-like [9] architecture. $M^{'}$ is the number of compressed metric dimension. Our design is different from the traditional HVAEs [28, 31] for images, in which the authors use a larger $\mathbf{z}_2$ to capture low-level features and a smaller $\mathbf{z}_1$ directly derived from $\mathbf{z}_2$ to capture high-level information. Applying the traditional HVAE on MTS anomaly detection would cause the learned time-compressed temporal embedding $\mathbf{z}_{2_t}^{1:M}$ at each time $t$ to misalign with the features at a specific timestamp. This is not suitable, as $\mathbf{z}_1$ should encode the intermetric information (*i.e.*, relationships among all metrics) at each time $t$, but the learned embedding would be inconsistent with the temporal order of MTS if we directly derive a smaller $\mathbf{z}_1$ from the time-compressed $\mathbf{z}_2$. We argue that *two-view embedding* can help *InterFusion* learn better intermetric embeddings that are *aware of the learned temporal information*, while *preserving the time consistency inside intermetric embeddings*.

**Prefiltering strategy.** A flexible intermetric embedding (which is able to capture the complex intermetric dependencies) is often needed to model the MTS. However, real-world raw MTS data used for training often contain anomalies. Thus directly learning a flexible embedding on the raw data may suffer from the risk of overfitting to anomalous patterns [33] and degrade the detection performance. Therefore, inspired by the scheduled sampling for sequence prediction [3], we propose a *prefiltering strategy* to derive the powerful intermetric embeddings $\mathbf{z}_1$. More specifically, $\mathbf{z}_1$ is derived from the "reconstructed input" $\mathbf{d}$ (rather than directly from $\mathbf{x}$), and $\mathbf{d}$ attempts to faithfully reconstruct the raw input $\mathbf{x}$ while filtering out temporal anomalies in $\mathbf{x}$ through the embedding-reconstruction procedure. $\mathbf{d}$ is pretrained with a VAE model to ensure its initial reconstruction capability at the beginning of training *InterFusion*. In this way, we *reduce the risk of the model overfitting to potential anomalies*, while *preserving the flexibility of the intermetric embeddings as well as considering the learned temporal information*.

The *three design choices* above enable *InterFusion* to jointly learn low-dimensional intermetric and temporal embeddings in the latent space, which are then used for characterizing the normal patterns of MTS data and detecting different types of anomalies.

Finally, as shown in Fig. 3(a), the variational posterior can be factorized as: $q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{z}_2)q_\phi(\mathbf{z}_2|\mathbf{x})$. Specifically, to deduce powerful intermetric embeddings that are aware of temporal information, an SRNN-like [9] architecture is adopted. That is, at each time $t$, $q_\phi(\mathbf{z}_{1_t}|\mathbf{z}_{1_{t-1}}, \mathbf{a}_t = h_{\phi_a}(\mathbf{a}_{t+1}, \mathbf{d}_t))$, where $\mathbf{a}_t$ is the deterministic state derived by a backward-recurrent GRU network [7] with $\mathbf{d}$ as its input, which is used to capture the future dependency $q_\phi(\mathbf{z}_{1_t}|\mathbf{d}_{t+1:W})$ in the input sequence, as suggested by [9]. A Real NVP flow [8] is applied on Gaussian $\mathbf{z}_{1_t}$ to get a more powerful representation. In the generative net, $p_\theta(\mathbf{z}_1|\mathbf{z}_2)$ is modeled by a non-linear state space model [26] via $p_\theta(\mathbf{z}_{1_t}|\mathbf{z}_{1_{t-1}}, \mathbf{e}_t)$, similar to $\mathbf{z}_1$ in the variational net. Note that, the parameters of the Deconv1D layers (to derive $\mathbf{d}$ and $\mathbf{e}$) are shared between the generative net and variational net to share knowledge about the current "reconstructed input" and the learned temporal information to improve training.

## 3.3 Model Training and Inference

**Training**. The VAE-based model can be trained by optimizing the ELBO $\mathcal{L}(\mathbf{x}, \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{\mathrm{KL}}\big(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\big)$, using SGVB estimator [17], where $D_{\mathrm{KL}}$ is the Kullback-Leibler divergence. In *InterFusion*, we take the *auxiliary deterministic variables* $\mathbf{d}$, $\mathbf{e}$ into consideration, thus rewrite the training objective as:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \theta, \phi) &= \mathbb{E}_{q_\phi(\mathbf{z}_1, \mathbf{z}_2, \mathbf{d}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, \mathbf{e})\right] \\ &\quad - D_{\mathrm{KL}}\big(q_\phi(\mathbf{z}_1, \mathbf{z}_2, \mathbf{d}|\mathbf{x})||p_\theta(\mathbf{z}_1, \mathbf{z}_2, \mathbf{e})\big) \qquad (1) \\ &= \mathbb{E}_{q_\phi}\left[\log p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, \mathbf{e}) + \log p_\theta(\mathbf{z}_1, \mathbf{e}|\mathbf{z}_2)\right. \\ &\quad \left. + \log p_\theta(\mathbf{z}_2) - \log q_\phi(\mathbf{z}_1, \mathbf{d}|\mathbf{z}_2, \mathbf{x}) - \log q_\phi(\mathbf{z}_2|\mathbf{x})\right] \qquad (2)\end{aligned}$$

Note that, $\mathbf{d}_{1:W}$ is the deterministic "reconstructed input" derived by applying DeconvNets on the time-compressed $\mathbf{z}_2$, as shown in Fig. 3(a), thus $\mathbf{d}_{1:W} \sim q_\phi(\mathbf{d}_{1:W}|\mathbf{z}_2, \mathbf{x}) = q_\phi(\mathbf{d}_{1:W}|\mathbf{z}_2) = \delta(\mathbf{d}_{1:W} - g(\mathbf{z}_2))$, which follows a delta distribution. Similarly, $p_\theta(\mathbf{e}_{1:W}|\mathbf{z}_2) = \delta(\mathbf{e}_{1:W} - g(\mathbf{z}_2))$ also follows a delta distribution. We let the delta distributions $q(\mathbf{d}_{1:W}|\mathbf{z}_2) = p(\mathbf{e}_{1:W}|\mathbf{z}_2)$ through *sharing parameters* of DeconvNets in the qnet and pnet, thus they can be canceled out when calculating ELBO. According to the dependencies in Fig. 3(a), we have:

$$\begin{aligned}&\iiint q_\phi(\mathbf{z}_1, \mathbf{z}_2, \mathbf{d}_{1:W}|\mathbf{x})\mathrm{d}\mathbf{z}_1\mathrm{d}\mathbf{z}_2\mathrm{d}\mathbf{d}_{1:W} \\ &= \iint q_\phi(\mathbf{z}_1|\mathbf{d}_{1:W} = g(\mathbf{z}_2))q_\phi(\mathbf{z}_2|\mathbf{x})\mathrm{d}\mathbf{z}_1\mathrm{d}\mathbf{z}_2 \qquad (3)\end{aligned}$$

Thus the expectation with $q_\phi$ in Eq. (2) can be evaluated by taking $L$ $(\mathbf{z}_1, \mathbf{z}_2)$ samples from $q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})$ (where $\mathbf{z}_1$ is sampled with a sequential manner) and using Monte Carlo integration [10]. The first, third and fifth terms in Eq. (2) are all diagonal Gaussians, whose log probability can be calculated analytically. For the other two terms, remark $q(\mathbf{d}_{1:W}|\mathbf{z}_2) = p(\mathbf{e}_{1:W}|\mathbf{z}_2)$, we have:

$$\begin{aligned}&\log p_\theta(\mathbf{z}_1, \mathbf{e}|\mathbf{z}_2) - \log q_\phi(\mathbf{z}_1, \mathbf{d}|\mathbf{z}_2, \mathbf{x}) \\ &= \log p_\theta(\mathbf{z}_{1_{1:W}}|\mathbf{e}_{1:W}) - \log q_\phi(\mathbf{z}_{1_{1:W}}|\mathbf{d}_{1:W}) \qquad (4) \\ &= \sum_{t=1}^{W}\left[\log p_\theta(\mathbf{z}_{1_t}|\mathbf{z}_{1_{t-1}}, \mathbf{e}_t) - \log q_\phi(\mathbf{z}_{1_t}|\mathbf{z}_{1_{t-1}}, \mathbf{a}_t = h_{\phi_a}(\mathbf{a}_{t+1}, \mathbf{d}_t))\right]\end{aligned}$$

Moreover, a Real NVP transformation is applied on the posterior to obtain a more powerful one through an invertible mapping $\mathbf{z}^K = f_\lambda^{-1}(\mathbf{z}^0)$, and $\log q_\phi(\mathbf{z}_{1_t}^K|\mathbf{z}_{1_{t-1}}, \mathbf{a}_t) = \log q_\phi(\mathbf{z}_{1_t}^0|\mathbf{z}_{1_{t-1}}, \mathbf{a}_t) + \log\left|\det\left(\frac{\partial f_\lambda(\mathbf{z}_{1_t}^K)}{\partial \mathbf{z}_{1_t}^K}\right)\right|$. $\det(\partial f_\lambda(\mathbf{z}^K)/\partial \mathbf{z}^K)$ is the Jacobian determinant of $f_\lambda$. $f_\lambda$ is composed of $K$ invertible mappings modeled with *affine coupling layers* [8]. The final training objective can be obtained by substituting it and Eq. (4) into Eq. (2), which can be optimized using SGVB estimator and *reparameterization trick* [17].

**Inference**. During online detection, we use the reconstruction probability as the anomaly score (*i.e.*, $\mathbb{E}_{q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2)]$), which has been widely used in anomaly detection literature [1, 30, 33]. We choose the sliding window $(\mathbf{x}_{t-W+1}, \ldots, \mathbf{x}_t)$ as the input for detecting anomaly at time $t$, and use the score for the *last data* $\mathbf{x}_t$ in the window as the anomaly score (following [14, 23, 30, 33]).

However, as discussed in [33], anomalies in test data may introduce bias to the learned embeddings and affect the estimation of reconstructions. Since we do not know whether a new coming data $\mathbf{x}_t$ is an anomaly or not, we assume it is an "anomaly" beforehand and use MCMC imputation [25] to get a more reasonable reconstruction. Specifically, MCMC imputation is proposed by [25] to impute missing points in images. For an input $\mathbf{x} = (\mathbf{x}_o, \mathbf{x}_m)$, $\mathbf{x}_o$ is the observed part and $\mathbf{x}_m$ is the missing part. The latent embedding $\mathbf{z}$ is sampled from $q_\phi(\mathbf{z}|\mathbf{x}_o, \mathbf{x}_m)$, and then the observation is reconstructed given $\mathbf{z}$ to get the missing imputation $\mathbf{x}_m^{'}$. $(\mathbf{x}_o, \mathbf{x}_m)$ is then replaced by $(\mathbf{x}_o, \mathbf{x}_m^{'})$. Iterating the procedures above makes the imputation $\mathbf{x}_m^{'}$ getting closer to its normal pattern, from the correct marginal $p(\mathbf{x}_m|\mathbf{x}_o)$. During detection, we assume the last data $\mathbf{x}_t$ is an "anomaly" and regard it as $\mathbf{x}_m$, other points in the window are $\mathbf{x}_o$. We iterate the imputation for sufficiently large $S$ times to eliminate the bias and obtain a more reasonable estimation of the reconstructed $\bar{\mathbf{x}} = (\mathbf{x}_o, \mathbf{x}_m^{'})$. The revised reconstruction probability can be calculated by taking $L(\mathbf{z}_1, \mathbf{z}_2)$ samples for Monte Carlo integration (Eq. (5)), where $(\mathbf{z}_1^{(l)}, \mathbf{z}_2^{(l)})$ are sampled from $q_\phi(\mathbf{z}_1, \mathbf{z}_2|\bar{\mathbf{x}})$. The anomaly score is *negative* revised reconstruction probability. An observation with a higher score is more likely to be an anomaly.

$$\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\bar{\mathbf{x}})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1,\mathbf{z}_2)] = \frac{1}{L}\sum_{l=1}^{L}[\log p_\theta(\mathbf{x}|\mathbf{z}_1^{(l)},\mathbf{z}_2^{(l)})] \quad (5)$$

### 3.4 Anomaly Interpretation

For MTS data, we interpret the detected anomalies by finding a group of most anomalous metrics for each detected anomaly. However, The anomalies may bring bias to the learned embeddings, and affect the estimation of reconstruction at all dimensions (*i.e.*, some normal metrics may get poor reconstructions due to the effect of other anomalous metrics). Therefore, we propose an MCMC-based method to approximate the normal patterns and then interpret the anomalies based on the revised reconstruction probability.

The **core idea** is to estimate a group of "most anomalous points" $\mathbf{x}_m$ according to the original reconstruction probability for each detected entity anomaly, then apply MCMC imputation to get a more reasonable latent embedding and reconstruction. Note that the first estimation of $\mathbf{x}_m$ may not cover all potential anomalies, thus an iterative process is needed to make the revised reconstruction "normal enough". Specifically, for a detected anomaly $\mathbf{x}_t$, the original reconstruction probability $\mathbf{r}^0 = \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1,\mathbf{z}_2)] \sim R^{M\times W}$, which is estimated with input sequence $\mathbf{x} = \{\mathbf{x}_{t-W+1}, \ldots, \mathbf{x}_t\}$. For simplicity, we estimate the "normal window" according to the average training statistics, *i.e.*, denote baseline $b = \frac{1}{Q}\sum_Q[\sum_{m,t}\mathbf{r}^0_{m,t}]$ roughly as the reconstruction probability for normal input windows, where $Q$ is the number of input sliding windows in training data and $\mathbf{r}^0_{m,t}$ is the value of the $m$-th metric at time $t$ in $\mathbf{r}^0$. Our proposed anomaly interpretation method is shown in Algorithm 1.

Note that, *although the supposedly normal set $\mathbf{x}_o$ doesn't change during MCMC imputation, the supposed "anomaly" set $\mathbf{x}_m$ has been imputed, which leads to a more accurate estimation of latent embeddings and reconstruction probability.* If $r^{\mathrm{a}} \geq b$, we have obtained the latent embedding corresponds to normal patterns, after imputing the potential anomalous points in $\mathbf{x}$. The anomalous dimensions of MTS can be interpreted using the value for each dimension in AS in Algorithm 1, while dimensions with higher values are more likely to be anomalous. This is extremely useful for interpreting the detected entity anomalies in MTS, as it can show the system operators about the most anomalous metrics (which could be used to accelerate root cause analysis) and the corresponding normal patterns the algorithm expects (to explain the users about the detection results).

**Algorithm 1:** *InterFusion* Anomaly Interpretation

**Input:** input sequence $\mathbf{x} \in R^{M\times W}$, original reconstruction probability $\mathbf{r}^0$, normal baseline $b$, window length $W$, number of metrics $M$, small constant ratio $\beta_{init}$, $\beta_{inc}$
**Output:** revised anomaly score $AS \sim R^{M\times W}$ for interpretation
$n_p \leftarrow$ number of points $(\mathbf{x}_{m,t})$ where $\mathbf{r}^0_{m,t} < \frac{b}{M*W}$;
$n_{init} \leftarrow \beta_{init} n_p$, $n_{inc} \leftarrow \beta_{inc} n_p$, $n \leftarrow n_{init}$, $r^{\mathrm{a}} = \sum_{m,t}\mathbf{r}^0_{m,t}$;
**while** *not* $(r^{\mathrm{a}} \geq b$ *or* $n > n_p)$ **do**
  $\mathbf{x}_m \leftarrow$ top $n$ points in $\mathbf{x}$ that have the lowest $\mathbf{r}^0_{m,t}$;
  $\mathbf{x}_o \leftarrow$ other points in $\mathbf{x}$ but not in $\mathbf{x}_m$;
  Denote $\mathbf{x}^{'} = \mathbf{x} = (\mathbf{x}_o, \mathbf{x}_m)$;
  **for** $s \leftarrow 1$ *to* $S$ **do** `// MCMC imputation for S times`
    sample $(\mathbf{z}_1, \mathbf{z}_2)$ from $q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}_o, \mathbf{x}_m)$;
    reconstruct $(\mathbf{x}_o^{'}, \mathbf{x}_m^{'})$ from $p_\theta(\mathbf{x}_o, \mathbf{x}_m|\mathbf{z}_1, \mathbf{z}_2)$;
    update $\mathbf{x}^{'} \leftarrow (\mathbf{x}_o, \mathbf{x}_m^{'})$;
  **end**
  `/* Approximate the true reconstruction prob of the input window using revised` $\mathbf{x}^{'}$ `*/`
  $r^{\mathrm{a}} = \frac{M*W}{M*W-n}\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x}')}[\sum_{\mathbf{x}_i\in\mathbf{x}_o}\log p_\theta(\mathbf{x}_i|\mathbf{z}_1,\mathbf{z}_2)]$;
  add $r^{\mathrm{a}}$ to rlist, $n \leftarrow n + n_{inc}$;
**end**
$\tilde{\mathbf{x}} \leftarrow \mathbf{x}^{'}$ that achieves the highest $r^{\mathrm{a}}$ in rlist;
$\mathbf{r}^{\mathrm{f}} = \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\tilde{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1,\mathbf{z}_2)]$, $AS = -\mathbf{r}^{\mathrm{f}}$;

## 4 EXPERIMENTS AND ANALYSIS

First, we introduce the datasets and metrics for evaluation. Then we design experiments to answer the following research questions:
**RQ1:** How does *InterFusion* perform on MTS anomaly detection and interpretation, in comparison with the state-of-the-art methods?
**RQ2:** How effective is each design choice in *InterFusion*?
**RQ3:** Is *InterFusion* feasible to be deployed in production?

### 4.1 Datasets and Evaluation Metrics

We use four real-world MTS datasets to evaluate the anomaly detection and interpretation performance of *InterFusion*: three public datasets, SWaT [19] (Secure Water Treatment), WADI [19] (Water Distribution), SMD [30] (Server Machine Dataset), and a new dataset ASD (Application Server Dataset, which is collected and published by this paper). More details are in Appendix A.

We compute an anomaly score for each observation $\mathbf{x}_t$, where observations with higher scores are more anomalous. In practice, anomalies often last for some time and form a contiguous anomaly segment. Therefore, it is acceptable for a model to trigger an alert for any observation within the anomaly segment. Thus, we adopt the *point-adjust* approach, which is proposed by [33] and

widely used in the evaluation of detection tasks [2, 23, 27, 30]. More specifically, if at least one observation in a contiguous anomaly segment from the ground-truth is correctly identified, the segment is detected correctly; thus, all observations in the same anomaly segment are considered to have been correctly detected. The observations outside the ground-truth anomaly segment are treated as usual [33]. Following previous works [2, 14, 30] published on SIGKDD, we **mainly** use **F1-score** (F1 for short hereafter) to evaluate the anomaly detection performance. We enumerate and find the optimal global threshold for anomaly scores to calculate best-F1s.

Similarly, in practice, a group of metrics are often used to interpret a contiguous *anomaly segment*, since it's hard for operators to determine what are the most anomalous metrics for each *anomaly point*. More specifically, different metrics might show anomalous behaviors at different time within the **same** anomaly segment, due to the inherent correlations among metrics (*e.g.*, in the same anomaly segment, the network-related metrics drop at time $t$, while the CPU related metrics drop at $t + t_0$, due to the decreased requests received from the network. Both of them are regarded as the most anomalous metrics for this anomaly segment). Therefore, inspired by the *top-k hit ratio* (*i.e.*, recall) for recommendations [29], we propose a new metric, namely "InterPretation Score" (IPS), to evaluate the anomaly interpretation accuracy at the **segment level**. $A$ is the total number of detected anomaly segments. Denote the ground-truth anomalous metric set for segment $\Phi_a$ as $G_{\Phi_a}$. For each detected anomaly $\mathbf{x}_t$, $\mathrm{AS}_t^i$ denotes the anomaly score of the $i$-th metric at time $t$. For anomaly segment $\Phi_a$, metric i's segment score $\mathrm{AS}_{\Phi_a}^i = \max_{\mathbf{x}_t \in \Phi_a} \mathrm{AS}_t^i$. $I_{\Phi_a}$ is the top $\left|G_{\Phi_a}\right|$ metrics with the highest scores out of $\mathrm{AS}_{\Phi_a}^i$. $N_{\phi_a}$ is the number of detected anomaly observations in $\phi_a$, representing the anomaly segment $\phi_a$'s importance. The IPS is defined as Eq. (6). Intuitively, IPS is the weighted sum of top $\left|G_{\Phi_a}\right|$ hit ratio evaluated at the segment level.

$$\mathrm{IPS} = \sum_{a=1}^{A} \frac{w_a \left|G_{\Phi_a} \cap I_{\Phi_a}\right|}{\left|G_{\Phi_a}\right|}, \quad w_a = \frac{N_{\phi_a}}{\sum_{a=1}^{A} N_{\phi_a}} \tag{6}$$

## 4.2 RQ1. Performance and Analysis

**Anomaly Detection.** We compare *InterFusion*'s performance with recent state-of-the-art unsupervised MTS anomaly detection methods: LSTM-NDT [14], MSCRED [34], MAD-GAN [19], DSANet [13], OmniAnomaly [30], USAD [2] and VAEpro [15]. Although all the compared methods use a sliding window input (except VAEpro), most of them mainly model one of the dependencies in MTS. *i.e.*, LSTM-NDT and MSCRED mainly model the *temporal dependency*, while MAD-GAN, OmniAnomaly, DSANet and USAD pay attention to the *intermetric dependency*. More details are described in Section 5. The overall F1-score is shown in Table 1.

Overall, *InterFusion* outperforms all baselines. We observe that most methods achieve high detection performance on SWaT and SMD dataset since their anomalies are easier to be detected (*e.g.*, large spikes co-occurrence on several metrics), but *InterFusion*'s best-F1 still outperforms them by 0.0187-0.2130. In practice, the system operators also need the algorithm to detect subtle anomalies, like local fluctuations or anomalous correlations among several metrics (*e.g.*, the right half of Fig. 1). These anomalies may happen only in part of the system and affect several monitoring metrics, the negligence of which may lead to more severe failures later. Therefore, on the more complex MTS datasets, WADI (which contains 118 metrics) and ASD (which contains different kinds of intermetric and temporal anomalies that only affect parts of the system), most existing methods have shown inferior results, and *InterFusion*'s best-F1 significantly outperforms them by 0.0364-0.5470.

**Table 1: Average best-F1 for *InterFusion* and baselines.**

| Methods | SWaT | WADI | SMD | ASD | Avg. |
|---|---|---|---|---|---|
| LSTM-NDT | 0.8133 | 0.5067 | 0.7687 | 0.4061 | 0.6237 |
| MSCRED | 0.8346 | 0.5469 | 0.8252 | 0.5948 | 0.7004 |
| MAD-GAN | 0.8431 | 0.7085 | 0.8966 | 0.6325 | 0.7702 |
| OmniAnomaly | 0.7344 | 0.7927 | 0.9628 | 0.8344 | 0.8311 |
| DSANet | 0.8924 | 0.8739 | 0.9630 | 0.8740 | 0.9008 |
| USAD | 0.8227 | 0.4275 | 0.9024 | 0.7987 | 0.7378 |
| VAEpro | 0.8369 | 0.8200 | 0.8693 | 0.8522 | 0.8446 |
| ***InterFusion*** | **0.9280** | **0.9103** | **0.9817** | **0.9531** | **0.9433** |

Detailedly, (1) LSTM-NDT and MSCRED mainly model the *temporal dependency* but are weak at modeling *intermetric dependency*. LSTM-NDT makes predictions for each metric, which ignores the intermetric correlation. MSCRED models the MTS using signature matrices, which is poor at detecting subtle anomalies and fails to capture the intermetric dependency when the interactions among metrics are complex and nonlinear (*e.g.*, on WADI and ASD).

(2) On the contrary, MAD-GAN, OmniAnomaly, DSANet and USAD mainly models the *intermetric dependency* through embeddings along metric dimension using stochastic methods or adversarial training. However, they ignore learning low-dimensional representations along time dimension for each metric, thus are poor at modeling the *intra-metric temporal dependency*. DSANet performs the best among these four methods. Although it lacks the explicit low-dimensional temporal embeddings, we conjecture that DSANet uses the global and local convolutions to capture part of the temporal information, which ultimately improves the detection performance. However, as a prediction-based method, DSANet fails to predict the normal patterns accurately on some inherent unpredictable metrics [20], which downgrades its performance.

(3) Different from the models above, VAEpro takes each point (not sliding window) as its input, and detects the outliers without considering the contextual information. Therefore, it only detects *severe point anomalies* and fails to find contextual anomalies.

To further support the observations above, we divide the anomalies in ASD into three anomaly types mentioned in Section 2.2 and evaluate the performance of baseline methods for each type. The detailed results (shown in Appendix D) show that LSTM-NDT and MSCRED achieve better performance on detecting temporal anomalies than detecting intermetric anomalies, while MAD-GAN, OmniAnomaly, USAD and DSANet are the opposite as expected. VAEpro performs similarly for each type of anomalies, since severe point anomalies may occur in each type. *InterFusion* achieves recall higher than 0.98 and approximated F1 higher than 0.95 for all three types of anomalies, significantly outperforming all baselines.

**Takeaways:** Simultaneously learning low-dimensional intermetric and temporal embeddings improves the anomaly detection performance for each type of anomalies than just learning a single

**Table 2: Interpretation IPS for *InterFusion* and baselines.**

| Methods | SMD | ASD | Avg. |
|---|---|---|---|
| LSTM-NDT | 0.5751 | 0.8619 | 0.7185 |
| MSCRED | 0.6421 | 0.7652 | 0.7037 |
| OmniAnomaly | 0.8008 | 0.8029 | 0.8019 |
| DSANet | 0.6713 | 0.8123 | 0.7418 |
| VAEpro | 0.5681 | 0.8236 | 0.6959 |
| VAEpro* | 0.7433 | 0.8916 | 0.8175 |
| InterFusion-nI | 0.7752 | 0.8881 | 0.8317 |
| ***InterFusion*** | **0.8340** | **0.9107** | **0.8724** |

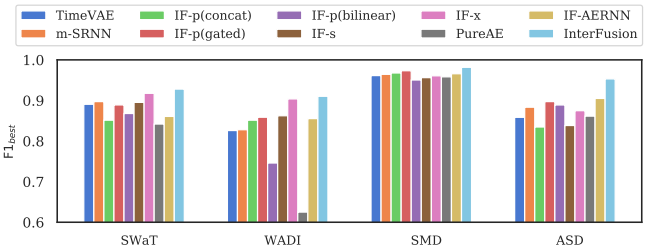


**Figure 4: Average anomaly detection best-F1 for *InterFusion* and its variants. 'IF' denotes *InterFusion* for short.**

type representation for MTS. A two-view embedding with hierarchical stochastic latent variables has been shown as an effective way to jointly learn robust MTS representations, which helps *InterFusion* outperform all competitors in MTS anomaly detection.

**Anomaly Interpretation.** We evaluate the anomaly interpretation on SMD and ASD, which provided ground-truth interpretation labels for evaluation (SWaT and WADI do not have interpretation labels, thus are not included). For the methods that do NOT propose a specific interpretation approach (LSTM-NDT and DSANet), we calculate **IPS** according to their own detection results and anomaly scores. MAD-GAN and USAD do not provide metric-wise anomaly scores, thus are not included for the comparison. Moreover, since *InterFusion* achieves the best detection results among these methods, we also apply other interpretation approaches to *InterFusion*'s detection results for a more fair comparison. InterFusion-nI denotes the *InterFusion* variant using the original reconstruction probability for interpretation (without the imputation and interpretation method). VAEpro* denotes the approach applying VAEpro's interpretation method on *InterFusion*'s detection results.

As shown in Table 2, *InterFusion* achieves higher IPS than all baselines and InterFusion-nI, demonstrating the effectiveness of our proposed interpretation method. VAEpro enhances interpretability by directly optimizing the learned distribution in latent space, which may cause the revised reconstructions to deviate from the original input, thus making the interpretation irrelevant to the input data. VAEpro* still falls behind InterFusion-nI on IPS, which indicates that VAEpro approach sometimes may harm the interpretation. DSANet, which achieves the best detection performance among baseline methods, fails to obtain reasonable anomaly interpretation due to its inaccurate predictions on some metrics in MTS. In general, the learned embeddings are more likely to deviate from the normal ones on datasets where anomalies' intensity is higher (*e.g.*, SMD). In this case, our proposed interpretation method can help approximate the normal patterns and achieve further improvement in interpretation.

**Takeaways:** The MCMC-based interpretation method can obtain reasonable embeddings and reconstructions at detected entity anomalies, which helps *InterFusion* outperform all baselines in MTS anomaly interpretation. *InterFusion* is able to find the most anomalous metrics and their corresponding normal patterns, which can further convince the system operators about the detection results.

## 4.3 RQ2. Ablation Studies

We conduct ablation studies using several variants of *InterFusion* to further demonstrate the effectiveness of the designs described in Section 3.2. The results are shown in Fig. 4.

**Intermetric-temporal Embeddings.** *InterFusion* outperforms m-SRNN (only intermetric embedding) and TimeVAE (only temporal embedding), which demonstrates the importance of *jointly learning low-dimensional intermetric-temporal embeddings* for MTS anomaly detection. Moreover, as shown in Appendix D, *InterFusion* outperforms both methods even in detecting the anomaly types (temporal or intermetric) that the methods are specifically designed for, which indicates that *InterFusion* does not simply combine the results of intermetric and temporal methods, but *benefits from the complementary characteristic information from both perspectives (intermetric and temporal) to learn better representations for normal data.*

**Latent Variables Dependency and Auxiliary Fusion Method.** InterFusion-p derives independent latent variables in the variational model without a hierarchical structure. Instead, it uses different fusion methods (concatenate, gated [6], or bilinear [16] fusion) in the generative model to combine the learned embeddings. *InterFusion* outperforms these methods, demonstrating the effectiveness of *using hierarchical structure and information sharing in latent space for jointly learning latent embeddings from different perspectives.*

**Two-view Embedding.** InterFusion-s uses a hierarchical structure similar to the HVAE for images [31], which makes the intermetric embedding for MTS inconsistent with the temporal order of the data, thus achieves worse performance than *InterFusion*, demonstrating that *two-view embedding is more suitable for modeling MTS data.*

**Prefiltering Strategy** can *prevent the model from overfitting to the potential anomalies in training data.* On SMD and ASD, anomalies exist in training data. Without using prefiltering, IF-x performs worse than *InterFusion* since it overfits the anomalous patterns in training data with the learned flexible intermetric embeddings. On SWaT and WADI, whose training data include only *normal* patterns but no anomalies [19], IF-x achieves similar results with *InterFusion.*

**Generalization.** IF-AERNN applies the three design choices (Sec 3.2) in *InterFusion* on autoencoder structure and significantly outperforms Pure autoencoder, demonstrating the *generalizability of our designs.* Moreover, incorporating VAE and SRNN in *InterFusion* enables it to model the stochasticity and complex patterns inside MTS, which further improves the detection performance.

## 4.4 RQ3. Feasibility Study

The deployment of *InterFusion* can be divided into 4 stages, as discussed in Fig. 3(b). The preprocessing step is common for training and detection, where MTS data is normalized and split into sliding

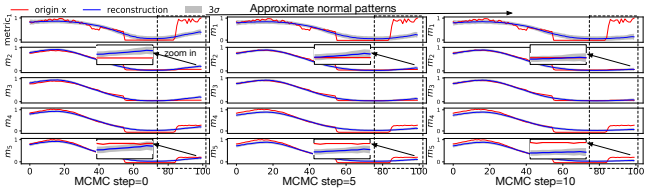


**Figure 5: MCMC imputation approximates the normal patterns of anomalies and finds the true anomalous metrics.**

windows of length $W$. Then *InterFusion* model is trained offline to learn the normal patterns of MTS. After training, the stored model can be used for online anomaly detection. *i.e.*, as a new observation $\mathbf{x}_t$ arrives, the model is used to obtain an entity anomaly score. If the score is higher than a pre-defined threshold, then $\mathbf{x}_t$ is declared as an entity anomaly. Finally, the anomaly interpretation is applied on each detected entity anomaly to show the most anomalous metrics.

**Detection.** We conducted a feasibility study using our ASD dataset collected from 12 application servers in a large Internet company. On each server, after the offline training with 3-week data, *InterFusion* was able to detect nearly *all* anomalies over 15 days length of test data (with an overall precision of 0.93 and recall of 0.99), including severe anomalies (*e.g.*, several metrics concurrently dip due to severe network failure) and subtle anomalies (*e.g.*, TCP metrics fluctuate subtly due to burst congestion), which demonstrates its feasibility on detecting MTS anomalies in production.

**Interpretation.** Fig. 5 shows five out of nineteen metrics within a sliding window in an entity of ASD dataset. Metrics $m_1$, $m_4$ and $m_5$ are anomalous in the area highlighted by the dashed ractangles, while other metrics are normal (more normal metrics are omitted here), according to the ground-truth labels. The anomaly in $m_1$ is much more severe than others, which can bring bias to the learned embedding and affect the estimation of reconstruction at other dimensions. Thus, in the left figure (before MCMC imputation), $m_4$ and $m_5$ in the dashed rectangle are detected as normal metrics. After applying MCMC-based anomaly interpretation, as shown in the right figure, we gradually approximate the normal patterns that $m_4$ and $m_5$ should have followed, which are generated from the correct marginal $p(\mathbf{x}_m|\mathbf{x}_o)$. This helps the algorithm find the true anomalous metrics ($m_4$, $m_5$) with subtle anomalies, which is essential for interpreting anomalies in MTS data. Moreover, the normal metrics (*e.g.*, $m_2$ and $m_3$) also get better reconstructions, since the effect of severe anomalies has been alleviated after the MCMC-based interpretation process. This can tell the operators about the most affected metrics and the extent to which the metrics deviate from their normal patterns, which can help the operators decide if they should resolve the problem immediately.

**Computation time.** The experiments are conducted on an NVIDIA GeForce GTX 1080 Ti. With 3-week training data for each entity in ASD, *InterFusion* only takes about 6 minutes for offline training (including pretrain phase), which is among the most efficient methods. For the online detection, it takes *less than 1 second* for each point, which is much smaller than the data collection intervals (one point per 1 or 5 minutes). For each detected entity anomaly, the interpretation takes 2-15 seconds. Thus, *InterFusion* is able to do anomaly detection and interpretation in a real-time manner.

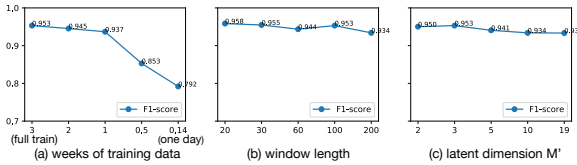


**Figure 6: Parameter sensitivity of *InterFusion* on ASD.**

**Parameter sensitivity.** We have shown that with the prefiltering strategy, *InterFusion* is robust to potential anomalies in training data. Therefore, we do not need to enforce the training data to be "anomaly-free". In Fig. 6, we show *InterFusion*'s sensitivity to other key parameters that are of concern in deployment. *InterFusion* can achieve high performance even with one-week training data (Fig. 6a) since the monitoring metrics often present weekly patterns. Even though the training data is not enough, *InterFusion* is able to well approximate the normal patterns from existing data and present fine results. Operators can do further training to enhance the model performance once the subsequent data is ready. Besides, *InterFusion* is not sensitive to the sliding window length (Fig. 6b) and the dimensions of the latent variables (Fig. 6c). In general, the sliding window length depends on whether the temporal dependencies are long-term or short-term in data. The latent dimension can be set to a much smaller value than the number of metrics, preventing the model from learning identical reconstructions with the input.

Overall, we have shown the feasibility of *InterFusion* through its pilot deployment on real-world data ASD. It's important to note that, in this paper, we focus on the MTS metrics of a stable real-world system that does not experience severe and frequent service changes. Thus, we assume that the training and testing data roughly follow the same distribution, which makes it possible to learn normal patterns from data to detect anomalies. In case of a large service change, a retrain step should be triggered to make *InterFusion* adapt to the *new* normal patterns. We want to note that *InterFusion*'s efficiency in training time and low demand for training data make it possible to continuously safe-guard the monitored servers with minimal retrain overhead.

## 5 RELATED WORK

**Anomaly Detection**. LSTM-NDT [14] and MSCRED [34] mainly models the *intra-metric temporal dependency*. Specifically, LSTM-NDT used LSTM [12] for MTS prediction in each metric and then detected anomalies according to prediction errors, but ignored the intermetric correlations. MSCRED used signature matrices to characterize MTS, and then applied an Encoder-Decoder structure to learn the reconstructions for anomaly detection. However, it relies on the *covariance* among different metrics, and cannot well learn the complex and nonlinear interactions among metrics [27].

Another group of methods mainly models the *intermetric dependency*. LSTM-VAE [22] combined VAE and LSTM by replacing the feed-forward networks in VAE with an LSTM. Similarly, MAD-GAN [19] combined GAN and LSTMs. OmniAnomaly [30] proposed an SRNN model with a Planar Normalizing Flow [24] posterior to enhance the capability of intermetric embeddings. Although these three models used stochastic variables for better intermetric embeddings, they did not learn specific representations along time

dimension for each metric, which is crucial for MTS anomaly detection, and thus are poor at modeling the *temporal dependency*. The RNNs [7, 12] used in these works actually act as a feature extraction layer for the whole entity, aiming at learning better intermetric embeddings at each timestamp. Similarly, USAD [2] proposed an adversarially trained autoencoder to model the *intermetric dependency*, while DSANet [13] used multi-head attention networks to make predictions. Both of them lack of low-dimensional *temporal embeddings*, thus are weak at modeling the *temporal dependency*.

**Anomaly Interpretation**. MSCRED [34] and OmniAnomaly [30] directly used the raw reconstruction score for each metric as an interpretation, which ignored the fact that the anomalies can affect the estimation of reconstructions at all dimensions [15] and cause misinterpretation. VAEpro [15] proposed an approximative probabilistic model to find better latent distribution for anomalous input. However, directly optimizing the encoded distribution in latent space may cause the revised reconstructions to deviate from the original input, which also causes misinterpretation.

## 6 CONCLUSION

Anomaly detection and interpretation for MTS are essential tasks for system monitoring. In this paper, we propose *InterFusion*, a novel unsupervised anomaly detection method that simultaneously models the intermetric and temporal dependency in MTS using HVAE with specifically designed structures. *InterFusion* outperforms the SOTA methods on four real-world datasets, demonstrating the effectiveness of explicitly learning low-dimensional intermetric and temporal embeddings with our design choices (hierarchical structure, two-view embedding, and prefiltering strategy) for MTS anomaly detection. Moreover, we propose a novel MTS anomaly interpretation method based on MCMC imputation, and a new quantitative evaluation metric consistent with the system operators' preferences. The feasibility study shows that *InterFusion* successfully meets the requirements on MTS anomaly detection and interpretation in real-world application server monitoring data, and provides suggestions to help deploy *InterFusion* and apply it on other industrial domains.

## ACKNOWLEDGEMENTS

We thank Wenxiao Chen and Haowen Xu for their helpful discussions on this work. We also thank iTrust for sharing us SWaT and WADI datasets. This work has been supported by the National Key R&D Program of China under Grant No.2019YFE0105500, the State Key Program of National Natural Science of China under Grant 62072264, and the Beijing National Research Center for Information Science and Technology (BNRist) key projects.

## REFERENCES

[1] Jinwon An and Sungzoon Cho. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE* 2, 1 (2015).
[2] Julien Audibert, Pietro Michiardi, et al. 2020. USAD: UnSupervised Anomaly Detection on Multivariate Time Series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3395–3404.
[3] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*. 1171–1179.
[4] Guilherme O Campos, Arthur Zimek, Jörg Sander, et al. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery* 30, 4 (2016), 891–927.
[5] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).
[6] Yanhua Cheng, Rui Cai, et al. 2017. Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3029–3037.
[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using real nvp. In *International Conference on Learning Representations, ICLR*.
[9] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. 2016. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*. 2199–2207.
[10] John Geweke. 1989. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society* (1989), 1317–1339.
[11] Douglas M Hawkins. 1980. *Identification of outliers*. Vol. 11. Springer.
[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[13] Siteng Huang, Donglin Wang, Xuehan Wu, and Ao Tang. 2019. DSANet: Dual Self-Attention Network for Multivariate Time Series Forecasting. In *28th ACM International Conference on Information and Knowledge Management*. 2129–2132.
[14] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 387–395.
[15] Yasuhiro Ikeda, Kengo Tajiri, Yuusuke Nakano, Keishiro Watanabe, and Keisuke Ishibashi. 2019. Estimation of dimensions contributing to detected anomalies with variational autoencoders. *Proceedings of the AAAI Conference on Artificial Intelligence, Workshop on Network Interpretability for Deep Learning* (2019).
[16] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Advances in Neural Information Processing Systems*. 1564–1574.
[17] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. *2nd International Conference on Learning Representations (ICLR)* (2014).
[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
[19] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, et al. 2019. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*. Springer, 703–716.
[20] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148* (2016).
[21] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. IEEE, 31–36.
[22] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. 2018. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1544–1551.
[23] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, et al. 2019. Time-Series Anomaly Detection Service at Microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3009–3017.
[24] Danilo Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning*. 1530–1538.
[25] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of the 31st International Conference on Machine Learning* 32 (2014), II–1278.
[26] Sam Roweis and Zoubin Ghahramani. 1999. A unifying review of linear Gaussian models. *Neural computation* 11, 2 (1999), 305–345.
[27] Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries Anomaly Detection using Temporal Hierarchical One-Class Network. *Advances in Neural Information Processing Systems* 33 (2020).
[28] Casper Kaae Sønderby, Tapani Raiko, et al. 2016. Ladder variational autoencoders. In *Advances in neural information processing systems*. 3738–3746.
[29] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 713–722.
[30] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. 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*. 2828–2837.
[31] Jakub M Tomczak and Max Welling. 2018. VAE with a vampprior. In *21st International Conference on Artificial Intelligence and Statistics, AISTATS*. 1214–1223.
[32] Harri Valpola. 2015. From neural PCA to deep unsupervised learning. In *Advances in independent component analysis and learning machines*. Elsevier, 143–171.
[33] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, et al. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*. 187–196.
[34] Chuxu Zhang, Dongjin Song, et al. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1409–1416.

# A DATASET DETAILS

SWaT [21] (Secure Water Treatment) and WADI [19] (Water Distribution) are two datasets about water treatment plants, which have been used for MTS anomaly detection in [19]. Both datasets collect the normal sensor and actuator data of the plants as the training set, while several attacks (which caused anomalies) are launched to the system in the testing set (including normal and anomalous data).

SMD is a server machine dataset collected by [30]. Some machines in SMD experienced service change during the data collection period, which leads to severe concept drift in training and testing data (*i.e.*, the training and testing data follow different normal patterns, which is inconsistent with the assumptions in this paper). Thus, we only use part of the SMD dataset, containing 12 entities (machines) that do not suffer concept drift, for evaluating *all* the algorithms in this paper. The SMD dataset provides anomaly detection and interpretation labels on the test set for evaluation.

Moreover, we published a new ASD dataset (Application Server Dataset) collected from a large Internet company. A group of stable services is run on the entities (servers) in the dataset, thus no entity in ASD experiences service changes or concept drifts during the time period included in this dataset. More detailedly, ASD contains 12 entities, each of which characterizes the status of a server, containing 45-day-long MTS data with 19 metrics characterizing the status of the server (including CPU-related metrics, memory-related metrics, network metrics, virtual machine metrics, *etc.*). The observations in ASD are equally-spaced 5 minutes apart. The first 30-day data are used for training, while the last 15-day data are used for testing. Anomalies and their most anomalous dimensions in the ASD testing set have been labeled by system operators based on incident reports and domain knowledge, including severe system failures and subtle anomalies that affect parts of the system. Moreover, the anomalies in ASD are roughly classified into three types: *temporal anomalies*, *intermetric anomalies*, and *intermetric-temporal anomalies*, as specified in Section 2.2.

**Table 3: Dataset Statistics.**

| Dataset | # entities | # metrics | Train | Test | Anomaly (%) |
|---|---|---|---|---|---|
| SWaT | 1 | 51 | 475200 | 449919 | 12.13 |
| WADI | 1 | 118 | 789371 | 172801 | 5.85 |
| SMD | 12 | 38 | 304168 | 304174 | 5.84 |
| ASD | 12 | 19 | 102331 | 51840 | 4.61 |

Note that, in training data, there might be a small number of anomalies in SMD and ASD, while SWaT and WADI only contain normal data. For detailed dataset statistics shown in Table 3, SWaT and WADI only contain one entity, while SMD and ASD both have 12 entities. Thus we calculate the average performance metrics of each entity for SMD and ASD when reporting the performance in Table 1 and Table 2.

The SWaT and WADI datasets can be acquired following the instructions in their original paper [19]. The SMD dataset can be acquired from [30], and our ASD dataset is released in https://github.com/zhhlee/InterFusion.

**Table 4: Average AUROC and AP (± std) over all datasets.**

| Methods | AUROC | AP |
|---|---|---|
| LSTM-NDT | 0.8462 (0.0528) | 0.5624 (0.2008) |
| MSCRED | 0.8495 (0.1074) | 0.6326 (0.1429) |
| MAD-GAN | 0.9007 (0.0669) | 0.7162 (0.1538) |
| OmniAnomaly | 0.9674 (0.0293) | 0.8090 (0.1215) |
| DSANet | 0.9817 (0.0221) | 0.9140 (0.0410) |
| USAD | 0.9213 (0.0620) | 0.7271 (0.1820) |
| VAEpro | 0.9438 (0.0440) | 0.8287 (0.0383) |
| ***InterFusion*** | **0.9904 (0.0123)** | **0.9582 (0.0199)** |

# B EVALUATION METRICS

As discussed in Section 4.1, we adopt the *point-adjust* approach [33] to calculate the performance metrics for anomaly detection. Therefore, given a specific threshold, we can calculate the TP (True Positives), FP (False Positives), TN (True Negatives) and FN (False Negatives) according to *point-adjust*. Thus we have $\text{F1} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$, while $\text{precision} = \frac{TP}{TP+FP}$, $\text{recall} = \frac{TP}{TP+FN}$. The best-F1 can be found using the optimal global threshold. Moreover, there are two other metrics that do not rely on the best threshold selection, AP and AUROC [4]. Given all possible thresholds, we can get a precision-recall curve (with recall as the x-axis) and calculate AP (average precision) as: $\text{AP} = \sum_n [(R_n - R_{n-1})P_n]$, where $R_n$ and $P_n$ are recall and precision at the nth threshold. Similarly, $\text{TPR} = \text{recall} = \frac{TP}{TP+FN}$, $\text{FPR} = \frac{FP}{FP+TN}$, then we have the ROC curve with FPR as the x-axis and TPR as the y-axis. AUROC is the area under the ROC curve. In practice, a feasible way is to select a small part of data as a validation set to evaluate a threshold that can achieve the best F-score, then it can be used for online detection as long as the distribution of normal data does not significantly change.

# C EXPERIMENT DETAILS

## C.1 Hyperparameter Selection

For *InterFusion* and its variants, we set the window length $W = 100$ on SMD and ASD, $W = 30$ on SWaT and WADI, as suggested in their original papers [19, 30]. For intermetric embedding, $M'$ is 2 for SWaT, 3 for SMD and ASD, 4 for WADI, considering their different metric numbers and data complexities. For temporal embedding, the (filter, strides) for each Conv1D layer is: (M,2), (M,1), (M,2), (M,1),(M,2), with kernel size = 5, M is the number of metrics, for SMD and ASD. SWaT and WADI only use the first three layers since the window length is shorter. The posterior Real NVP [8] flow layers are set to 20. ReLU is used as the activation function for layers other than linear layers. L2 regularization with a coefficient of $1e^{-4}$ is applied on non-linear hidden layers. The log standard deviations of latent distributions and posteriors are clipped within $[-5, 2]$ to avoid numerical problems. For training, the training set is preprocessed by MinMax Scaler within each metric. The min and max values in the training set are further used to preprocess data in the validation and testing sets. We apply Adam optimizer to optimize our model. The batch size is set to 100. For SMD and ASD datasets, the last 30% of data in the training set is used as the validation set, while for larger SWaT and WADI datasets, the validation portion is 10%.

An early stopping strategy is taken according to validation loss in each epoch. The number of $\mathbf{z}$ samples $L = 100$ for Monte Carlo integration during testing. Each MCMC imputation procedure is executed for $S = 10$ times, while 10 MCMC chains are used to evaluate the results and eliminate the potential bias. As suggested by [19], due to the cold start of the system, we use the training data of SWaT starting from point 21600 and point 259200 for WADI; while other datasets use the whole training/testing set. To obtain a more accurate "reconstructed input" at the early step of training and make the training easier, we use a vanilla VAE with only the temporal embedding latent variable $\mathbf{z}_2$ as a pretrain model. The structures of 1D ConvNets and DeconvNets are the same in the pretrain and main models, while their parameters and the derived $\mathbf{z}_2$ distributions are used to initialize the same parts in the main model. **Our code and data** are released at https://github.com/zhhlee/InterFusion.

## C.2 Baseline implementation

LSTM-NDT [14] comes from the authors' implementation on https://github.com/khundman/telemanom. MSCRED [34] comes from the implementation on https://github.com/wxdang/MSCRED. MAD-GAN [19] comes from the authors' implementation on https://github.com/LiDan456/MAD-GANs. OmniAnomaly [30] comes from the authors' implementation on https://github.com/NetManAIOps/OmniAnomaly. DSANet [13] comes from the authors' implementation on https://github.com/bighuang624/DSANet. USAD [2] comes from the authors' implementation on https://github.com/robustml-eurecom/usad. VAEpro [15] is implemented by us following their paper.

## C.3 *InterFusion*'s Variants

In Section 4.3 we proposed several variants of *InterFusion* for ablation study. Here we describe each variant model in detail.

**TimeVAE** removes intermetric latent variable $\mathbf{z}_1$ from *InterFusion*, while preserving the temporal embeddings using 1D ConvNets.

**Modified-SRNN** (m-SRNN) removes the temporal embedding latent variable $\mathbf{z}_2$ from *InterFusion*, and the remaining model looks similar to an SRNN model [9]. The additional inputs $\mathbf{u}$ in SRNN are always set to zeros in modified-SRNN. The Real NVP [8] is also applied to enrich the posterior, as done in *InterFusion*.

**InterFusion-p** does not use hierarchical structure and two-view embedding, and derives independent latent variables $\mathbf{z}_1$ and $\mathbf{z}_2$ in the variational net. It acts as a parallel combination of TimeVAE and modified-SRNN, where $q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{x})$, $p_\theta(\mathbf{z}_1, \mathbf{z}_2) = p_\theta(\mathbf{z}_1)p_\theta(\mathbf{z}_2)$. Three different fusion methods (*i.e.*, concatenate, gated fusion [6], bilinear fusion [16]) are used to combine the learned features in the generative net.

**InterFusion-s** does not adopt two-view embedding, but uses a hierarchical structure similar to the HVAE for images [31]. *i.e.*, first learn the temporal embedding $\mathbf{z}_2 \in R^{M \times W'}$ via $q(\mathbf{z}_2|\mathbf{x})$, then derive $q(\mathbf{z}_1|\mathbf{z}_2)$ through embedding $\mathbf{z}_2$ along the metric dimension. Thus, the intermetric embedding $\mathbf{z}_1 \in R^{M' \times W'}$ is inconsistent with the temporal order of the MTS data.

**InterFusion-x** does not apply the prefiltering strategy, which lets $\mathbf{z}_1$ also directly depend on the original input $\mathbf{x}$, rather than only depend on $\mathbf{d}$, in the variational net. In this way, it may overfit to the potential anomalies in training set, which can downgrade its detection performance (*e.g.*, on SMD and ASD).

**InterFusion-AERNN** applies the design choices (hierarchical structure, two-view embedding and prefiltering strategy) of *InterFusion* on an autoencoder structure, and replaces the SRNN with GRU [7]. It significantly outperforms PureAE, which demonstrates the generalizability of our proposed designs for MTS anomaly detection.

**Table 5: Anomaly detection performance on three type of anomalies in ASD dataset.**

| Methods | Intermetric | | | Temporal | | |
|---|---|---|---|---|---|---|
| | aPr | R | aF1 | aPr | R | aF1 |
| LSTM-NDT | 0.245 | 0.251 | 0.248 | 0.277 | 0.295 | 0.286 |
| MSCRED | 0.358 | 0.397 | 0.376 | 0.452 | 0.587 | 0.511 |
| MAD-GAN | 0.712 | 0.594 | 0.648 | 0.653 | 0.451 | 0.534 |
| OmniAnomaly | 0.850 | 0.835 | 0.842 | 0.842 | 0.785 | 0.813 |
| DSANet | 0.885 | 0.917 | 0.901 | 0.880 | 0.872 | 0.876 |
| USAD | 0.825 | 0.959 | 0.887 | 0.761 | 0.650 | 0.701 |
| VAEpro | 0.801 | 0.886 | 0.841 | 0.798 | 0.867 | 0.831 |
| TimeVAE | 0.721 | 0.463 | 0.564 | 0.843 | 0.965 | 0.900 |
| modified-SRNN | 0.865 | 0.978 | 0.918 | 0.822 | 0.705 | 0.759 |
| *InterFusion* | 0.928 | 0.981 | **0.954** | 0.928 | 0.982 | **0.954** |
| Methods | Intermetric-Temporal | | | Total | | |
| | aPr | R | aF1 | aPr | R | aF1 |
| LSTM-NDT | 0.477 | 0.705 | 0.569 | 0.411 | 0.538 | 0.466 |
| MSCRED | 0.534 | 0.814 | 0.645 | 0.496 | 0.700 | 0.581 |
| MAD-GAN | 0.747 | 0.709 | 0.728 | 0.723 | 0.627 | 0.672 |
| OmniAnomaly | 0.866 | 0.954 | 0.908 | 0.859 | 0.894 | 0.876 |
| DSANet | 0.888 | 0.942 | 0.914 | 0.886 | 0.920 | 0.903 |
| USAD | 0.825 | 0.963 | 0.889 | 0.812 | 0.881 | 0.845 |
| VAEpro | 0.819 | 0.994 | 0.898 | 0.812 | 0.947 | 0.874 |
| TimeVAE | 0.842 | 0.953 | 0.894 | 0.833 | 0.892 | 0.861 |
| modified-SRNN | 0.864 | 0.975 | 0.916 | 0.855 | 0.905 | 0.879 |
| *InterFusion* | 0.929 | 0.988 | **0.958** | 0.929 | 0.986 | **0.957** |

# D RESULTS FOR ANOMALY TYPES

We take ASD dataset as an example to show each methods' anomaly detection performance on different types of anomalies, as discussed in Section 4.2. Among the 2392 anomaly observations in 12 entities (each entity corresponds to one server), 60.79% are intermetric-temporal anomalies, 26.04% are temporal anomalies and the rest 13.17% are intermetric anomalies. We calculate the recall, approximated precision (aPr) and approximated F1-score (aF1) for each method, as shown in Table 5. The detected anomalous points are evaluated with *point-adjust* approach and optimal threshold. For each type of anomalies, TP and FN can be directly obtained, thus the precise recall can be calculated. However, since normal data *cannot* be categorized into different types, we can only obtain the number of FP for the whole dataset. Therefore, we divide the number of FP in the whole dataset into three parts according to the anomaly ratio of each type of anomalies. *i.e.*, the FP points for intermetric, temporal and intermetric-temporal anomalies are 13.17%, 26.04% and 60.79% of the total FP points, respectively. The "total" column is calculated using TP, FN and FP for *whole* ASD dataset (including all 12 entities, rather than the average F1 for each entity as in Table 1).