GENAD: GENERAL REPRESENTATIONS OF MULTIVARIATE TIME SERIES FOR ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Anomaly Detection(AD) for multivariate time series is an active area in machine learning, with critical applications in Information Technology system management, Spacecraft Health monitoring, Multi-Robot Systems detection, etc.. However, due to complex correlations and various temporal patterns of large-scale multivariate time series, a general unsupervised anomaly detection model with higher F1-score and Timeliness remains a challenging task. In this paper, We propose a General representations of multivariate time series for Anomaly Detection(GenAD). First, we apply Time-Series Attention to represent the various temporal patterns of each time series. Second, we employ Multi-Correlation Attention to represent the complex correlations of multivariate time series. With the above innovations, GenAD improves F1-scores of AD by 0.3% to 5% over state-of-the-art model in public datasets, while detecting anomalies more rapidly in anomaly segments. Moreover, we propose a general pre-training algorithm on large-scale multivariate time series, which can be easily transferred to a specific AD tasks with only a few fine-tuning steps. Extensive experiments show that GenAD is able to outperform state-of-the-art model with only 10% of the training data.

1 INTRODUCTION

Anomaly detection, which refers to the recognition of anomalous observations that differ from the general distribution patterns(Zhao et al., 2019), is an active area in machine learning, with critical applications in Information Technology system management (Zhang et al., 2019), Spacecraft Health monitoring (Hundman et al., 2018), Multi-Robot systems detection (Park et al., 2018), etc.. There are two kinds of anomaly detection directions in application that are referred to detecting anomalies at the entity-level using multivariate time series, and detecting anomalies at the metric-level using uni-variate time series. In real-world scenarios, the overall status of an entity such as machine or systems is more concerned about than each time series of the entity(Su et al., 2019), so we focus on detecting anomalies at the entity-level with multivariate time series. At the core of evaluating anomaly detection models are F1-score(precision and recall)(Xu et al., 2018), Timeliness(Zhang et al., 2019; Liu et al., 2019a) and Generalization(Li et al., 2018).

Although fruitful progress, such as DAGMM(Zong et al., 2018), MSCRED(Zhang et al., 2019), Omni(Su et al., 2019), has been made in the last several years, due to the complex Inter-dependencies (i.e., correlations among time series of different variate components) and various Intra-dependencies (i.e., temporal patterns within one time series), unsupervised anomaly detection for multivariate time series remains a challenging task. These existing models either lose the dynamic, higher-order and non-linear Inter-dependencies, or do not learn the Intra-dependencies well. How to represent the complex correlations and various temporal patterns of multivariate time series directly affects the performance of anomaly detection.

In addition, anomalies always occur continuously and form contiguous anomaly segments. Operation engineers care about whether the anomaly detection model can detect a continuous anomaly segment, rather than finding every anomaly time-slot. So the existing models pay more attention to successfully detect a subset of observations in the anomaly segment, which is considered that the entire segment is correctly detected. However, anomalies should be detected as soon as possible in anomaly segment, so that further actions can be taken to avoid serious losses. For instance, 1 minute

downtime of an automotive manufacturing plant may cost up to 20 000 US dollars (Djurdjanovic et al., 2003). Therefore, the rapid detection of anomaly segment is a critical core.

Moreover, due to the node-specific patterns in large-scale multivariate time series (i.e., millions of cells in mobile communication networks behave various patterns according to the surrounding environment), although deep learning models (Aytekin et al., 2018; Zhou & Paffenroth, 2017) perform better than any others after being carefully designed, it is impossible to deploy such models on large-scale multivariate time series in real-world scenarios due to the huge overhead in model training and parameter adjustment. That said, for different multivariate time series, it is required to have enough and different data to train dedicated models. How to optimize models for large-scale anomaly detection on thousands to millions of multivariate time series, is particularly important for practical application.

Based on the above problems, we propose a General multivariate time series representation for Anomaly Detection(GenAD), with the following contributions:

- GenAD employs **Time-Series Attention**, instead of LSTM, GRU,etc.(O'Shea et al., 2016; Chauhan & Vig, 2015; Malhotra et al., 2015; 2016), to represent the various temporal patterns of multivariate time series. Multi-head and Hidden layer are introduced in Time-Series Attention to capture Sequentiality, Trend, Delay, and Periodicity of N-dimensional series simultaneously, which is difficult for LSTM or GRU when N is large. Time-Series Attention also allows for parallelization, which is especially important at longer series lengths. Moreover, GenAD employs **Multi-Correlation Attention**, instead of AE, VAE, CNN(Kingma & Welling, 2013; Liu et al., 2019b; Burgess et al., 2018), to represent the complex correlations of multivariate time series. Attention mechanism is introduced in Multi-Correlation Attention to capture the dynamic correlation among time series that are not identically distributed, which is difficult for AE, VAE,CNN, as correlations of these models remain constant after offline training. Multi-head and Hidden layer are used to capture non-linear, coupling and higher-order correlations among time series. As time series contains noise in real-world applications, coherent accumulation and dropout are also introduced. Through extensive experiments, GenAD increases F1-Score by 0.3% to 5% over the state-of-the-art model in public datasets, demonstrating the benefits of explicitly representing the complex correlations and various temporal patterns of multivariate time series.
- Based on the robust representation of complex correlations and various temporal patterns, as well as the introduction of Ripple Effect(Dolgui et al., 2018) for anomaly detection, GenAD detects anomalies rapidly in anomaly segments. Experiments show that GenAD detects anomalies in a lower latency after the anomalies occurred in anomaly segments than the state-of-the-art model.
- GenAD pre-trains a general model on large-scale multivariate time series with self-supervision. The goal of the pre-training is to empower GenAD to capture the general correlations and temporal patterns of multivariate time series, so that it can be easily transferred to anomaly detection tasks for different multivariate time series models with only a few fine-tuning steps. The robustness to noise can also be enhanced at the same time. To the best of our knowledge, GenAD is the first general pre-training model for representing multivariate time series, which supports large-scale anomaly detection task. Extensive experiments show that GenAD achieves higher F1 scores and better stability on all datasets, while only 10% of the training data are used.

2 RELATED WORK

The existing unsupervised anomaly detection models for multivariate time series can be categorized into the following types:

- Anomaly detection can be implemented for each dimension of multivariate time series, and the overall status of an entity is voted or weighed on the outputs of each dimension. Anomaly detection for each dimension can be realized by statistical principles or distance measurements, including 3sigma(Son et al., 2016), boxplot(Moumena & Guessoum, 2015), HBOS(Goldstein & Dengel, 2012), KNN, AvgKNN(Angiulli & Pizzuti,

2002), OCSVM(Das et al., 2010), etc.. These models obtain the distribution or the farthest distance of normal series, while anomaly is detected as outliers. These models require less training data, which are suitable for large-scale series. However, most of these models perform better for short-term abnormalities, while the performance will be attenuated for long-term abnormalities. Anomaly detection for each dimension can also be realized by prediction(ARIMA(Contreras et al., 2003), LSTM, Prophet(Medina et al., 2007)) or reconstruction (AE, VAE). These models learn the Intra-dependencies of series and adapt well to time series. However, these models detect each time series in isolation and lose the correlations among multivariate series, which results in lower performance of an entity. What's more, these models require large and different data for training, which limits the application for large-scale series.

- Anomaly detection can also be implemented at the Entity-level than for each dimension. PCA(Shyu et al., 2003), RPCA(Paffenroth et al., 2018), MCD(Rousseeuw & Driessen, 1999) learn Inter-dependency patterns of multivariate series and detect anomalies based on changes in correlations, which requires less training data. But these models only represent the linear correlations. DEC(Xie et al., 2016), DR+K-means(Yang et al., 2017), DAGMM reduce the dimension of multivariate series by DNN or AE, which solves the problem of representations of non-linear correlation. RSRAE(Lai et al., 2019) combines AE and RSR to learn non-linear correlation, which also exhibits robustness to abnormal points in the training data. However, none of the above models is suitable for time series. MSCRED(Zhang et al., 2019) detected anomalies by calculating the differences between the reconstructed and the original correlation matrix. However, MSCRED only measures the correlation matrix and loses the Intra-dependencies of the series itself. Moreover, the correlation matrix is obtained by a simple inner-product of two time series, which is impossible to find the deep higher-order correlation. Omni(Su et al., 2019) introduces VAE to mine the Inter-dependencies and combines GRU to represent Intra-dependencies of series, which achieves robust results. However, single-layer GRU is difficult to capture the various temporal patterns of N-dimensional series when N is large. Omni also lose dynamic correlation and requires large training data, which also limits the application in large-scale multivariate time series.

Compared with the above approaches, GenAD can not only obtain dynamic, non-linear and deep higher-order correlations, but also represent various temporal patterns of time series. More importantly, GenAD proposes a pre-training algorithm on large-scale multivariate time series, which can outperform state-of-the-art model with only 10% of the training data.

3 General Representations for Anomaly Detect

In this section, we first present the problem statement of multivariate time series for anomaly detection, then introduce the overall architecture of our model GenAD. Besides, we illustrate **TIME-SERIES ATTENTION** and **Multi-Correlation ATTENTION** in detail, which are the key components of GenAD.

3.1 Problem Statement

Given the multivariate time series X of a machine or systems with N-dimensional time series, and T is the length of timestamps for X,

$$X = ({x_1}^T, {x_2}^T, \cdots, {x_N}^T) \in \mathbb{R}^{N \times T} \tag{1}$$

We learn the complex correlations and various temporal patterns of X in $(0, T)$, then determine whether there are anomaly segments of X at certain time steps after T.

3.2 Network Architecture

Figure 1 shows the preprocessing of input series,network architecture of GenAD and model training. **Preprocessing of input series:** input series is N-dimensions multivariate time series with time length T. T is divided into $(0, T_1)$ and (T_1, T), while the length of $(0, T_1)$ is equal to 4 times the

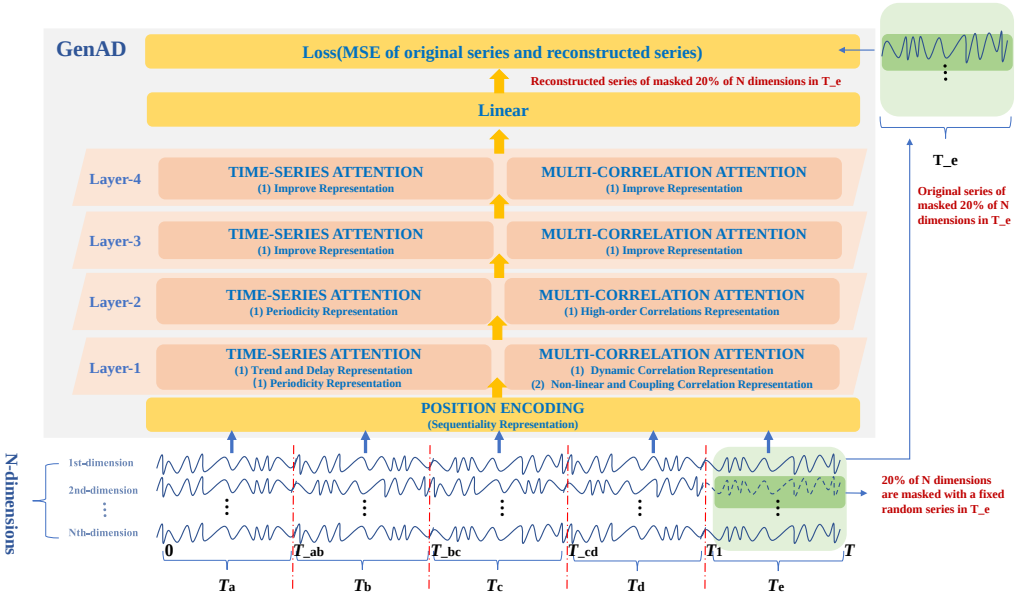


Figure 1: Model Architecture

length of (T_1, T). Given the length of (T_1, T) is T_e, the length of $(0, T_1)$ is sum of T_a, T_b, T_c, and T_d, while length of T_a, T_b, T_c, and T_d are the same and all equal to T_e. 20% of N dimensions are randomly selected to be masked with a fixed random series in T_e. Then, the left 80% original series and 20% masked series in T_e, as well as all series in T_a to T_d, are input series of GenAD. **Network architecture of GenAD:** GenAD consists of four parts, which are position encoding, 4 hidden layers with Time-Series Attention and Multi-Correlation Attention in each layer, linear layer and loss function. Positional encoding is injected into time series in order not to lose the Sequentiality. 4 hidden layers are introduced to represent temporal patterns and correlations of multivariate time series. Layer-4 and Linear output reconstructed series of masked 20% of N dimensions in T_e, while loss is capture by MSE of reconstructed series and 20% original series in T_e. **Model training:** we employ Adam Optimizer to minimize the MSE loss. 20% of N dimensions in preprocessing are randomly selected at each epoch of model training, so GenAD does not know which series of N have been used to be reconstructed or have been selected to be masked during model training. This will force the model to learn temporal patterns and correlations of multivariate time series in order to minimize loss after sufficient number of training epochs.

3.2.1 TIME-SERIES ATTENTION

GenAD employs Time-Series Attention, instead of RNN, GRU, LSTM, etc., to represent the various temporal patterns of multivariate time series, including Sequentiality, Trend, Delay, and Periodicity. Given one of the time series $x_i(n)$ in $X(n)$ as an example, Time-Series Attention is implemented by Eq.(2)

$$\text{TimeSeriesAttention}[{x_i}^{T_e}(n)] = \sum_{t=(T_a,T_b,T_c,T_d)} soft\max(\frac{({Q_i}^{T_e})({K_i}^{t})^{Transpose}}{\sqrt{d}}){V_i}^{t} \tag{2}$$

Where ${Q_i}^{T_e}$ is the transformation of $x_i(n)$ in time T_e; ${K_i}^{T_e}$ and ${V_i}^{T_e}$ are the transformation of $x_i(n)$ in time T_a, T_b, T_c, T_d; *Transpose* is the transpose of a matrix; d is the length of $x_i(n)$ in time T_e. Eq.(2) means that GenAD learns the various temporal patterns of $x_i(n)$ in T_a, T_b, T_c, T_d and reconstructs $x_i(n)$ in time T_e.

Sequentiality Representation:

Since Time-Series Attention does not contain convolution or recurrence, in order not to lose the Sequentiality, positional encoding is injected into the time series data. The encoding algorithm can choose the sin and cos functions mentioned in Transformer(Devlin et al., 2018), or use learned positional encoding variables. The two versions produce nearly identical results, and the latter is chosen here.

Trend Representation:

Assuming there are trends in $x_i(n)$, we can simplify that only linearly increasing trend exists:

$$x_i(n)=A_i t + bias, \text{s.t. } A_i > 0 \tag{3}$$

A_i is the linearly increasing coefficient of $x_i(n)$, $bias$ is $x_i(n)$ when t = 0. According to the network architecture in 3.2,

$$x_i{}^{T_e}(n) = A_i(T_1 + \Delta t, T) + bias \tag{4}$$

Δt is sample interval of $x_i(n)$. Assuming that T_e can be reconstructed by $x_i(n)$ in T_a, T_b and T_d, then setting Eq(2) that

$$\begin{aligned} &(\frac{(Q_i{}^{T_e})(K_i{}^t)^{Transpose}}{\sqrt{d}}) = 1 \\ &V_i{}^t = x_i{}^t \times V\_Array_i{}^t, t = (T_a, T_b, T_c, T_d) \end{aligned} \tag{5}$$

We let $V\_Array_i{}^{T_a} = I \times (-3), V\_Array_i{}^{T_b} = I \times 3, V\_Array_i{}^{T_c} = I \times 0, V\_Array_i{}^{T_d} = I \times 3$, where I is identity matrix, so

$$\begin{aligned} &\text{TimeSeriesAttention}[x_i{}^{T_e}(n) = \sum_{t=(T_a,T_b,T_c,T_d)} soft\max(\frac{(Q_i{}^{T_e})(K_i{}^t)^{Transpose}}{\sqrt{d}})V_i{}^t \\ &\text{=-}\frac{1}{3} \times (3 \times V_i{}^{T_a})\text{+}\frac{1}{3} \times (3 \times V_i{}^{T_b}) + \frac{1}{3} \times (3 \times V_i{}^{T_d}) \\ &= -(A_i(0 + \Delta t, T_{\_ab}) + bias) + (A_i(T_{\_ab} + \Delta t, T_{\_bc}) + bias) + (A_i(T_{\_cd} + \Delta t, T_1) + bias) \\ &= A_i T_a + (A_i(T_{\_cd} + \Delta t, T_1) + bias) = (A_i(T_1 + \Delta t, T) + bias) = x_i{}^{T_e}(n) \end{aligned} \tag{6}$$

$x_i(n)$ in T_e can be reconstructed by $x_i(n)$ in T_a, T_b and T_d. With the complexity of trends, the correlation between $x_i(n)$ in T_e and $x_i(n)$ in other times also changed, which can be learned by module training.

Delay Representation:

As there may exist delay between $x_i(n)$ in time T_e and $x_i(n)$ in T_a, T_b, T_c, T_d, GenAD captures delay by $V\_Array_i{}^t$ mentioned in Eq(5). Simplify Eq(2) that only correlation between $x_i(n)$ in T_e and $x_i(n)$ in T_a exists, and assuming $x_i(n)$ in T_e is $[a, b, c]$ and $x_i(n)$ in T_a is $[0, a, b]$, then setting $(\frac{(Q_i{}^{T_e})(K_i{}^t)^{Transpose}}{\sqrt{d}}) = 1$ and $V\_Array_i{}^{T_a} = \begin{bmatrix} 0,1,0 \\ 0,0,1 \\ 0,0,0 \end{bmatrix}$, so

$$\begin{aligned} &\text{TimeSeriesAttention}[x_i{}^{T_e}(n)] = \sum_{t=(T_a,T_b,T_c,T_d)} soft\max(\frac{(Q_i{}^{T_e})(K_i{}^t)^{Transpose}}{\sqrt{d}})V_i{}^t \\ &\text{=}x_i{}^{T_a}(n) \times \begin{bmatrix} 0,1,0 \\ 0,0,1 \\ 0,0,0 \end{bmatrix} \text{=[a,b,c]} \times \begin{bmatrix} 0,1,0 \\ 0,0,1 \\ 0,0,0 \end{bmatrix} \text{=[0,a,b]=}x_i{}^{T_e}(n) \end{aligned} \tag{7}$$

which delay $x_i(n)$ in T_a by one timestamp for $x_i(n)$ in T_e.

Periodicity Representation

GenAD captures the Periodicity of time series in T_a, T_b, T_c, T_d, and reconstructs the series in T_e according to the captured Periodicity. Simplify Eq(2) that only correlation between $x_i(n)$ in T_e and $x_i(n)$ in T_a exists, then $soft\max$=1. Assuming that $Q_i{}^{T_e} = \cos((2\pi/N)kn)$, $K_i{}^{T_a} = 2 \times x_i{}^{T_a}(n) \times \sqrt{d}$, $V_i{}^{T_e} = 1$, where $0 \le k \le (Num - 1)/2$, $0 \le n \le Num - 1$, $Num = \frac{T_a}{\Delta t}$,

then

$$\begin{aligned}
&\text{TimeSeriesAttention}[{x_i}^{T_e}(n)] = (\frac{({Q_i}^{T_e})({K_i}^{T_a})^{Transpose}}{\sqrt{d}}){V_i}^t \\
&= \cos\left((-2\pi/N)kn\right) \times (\frac{(2 \times {x_i}^{T_a}(n) \times \sqrt{d})}{\sqrt{d}})^{Transpose} \times 1 \\
&= \sum_{n=0}^{Num-1} \cos((-2\pi/N)kn) \times \frac{(2 \times {x_i}^{T_a}(n) \times \sqrt{d})}{\sqrt{d}} \\
&= FFT_k({x_i}^{T_a}(n))
\end{aligned} \tag{8}$$

The output of Time-Series Attention of *the first hidden layer* is the kth frequency point of $x_i(n)$ in T_a, and Multi-head Attention is employed to get other k frequency points of $x_i(n)$, where the number of Multi-head is $Num/2$. The formula of IFFT is similar to FFT, which can also be realized by Time-Series Attention in *the second hidden layer*. So $x_i(n)$ in T_e can be better reconstructed by the Periodicity of $x_i(n)$ in T_a.

After model training for Time-Series Attention, GenAD can represent the various temporal patterns of multivariate time series, including Sequentiality, Trend, Delay, and Periodicity.

3.2.2 Multi-Correlation Attention

Existing deep models use CNN, AE, VAE, etc. to represent correlations of multivariate time series. Once model training has been completed offline, the correlations will not change during online inference. However, the N-dimensions of the multivariable time series are not identically distributed in practice. For example, there may exist strong correlation between $x_i(n)$ and $x_j(n)$ of $X(n)$ in $(0, T_1)$, but when distribution of $x_i(n)$ and $x_j(n)$ has changed, there will be weak or no correlation between $x_i(n)$ and $x_j(n)$ in (T_1, T). GenAD introduces Multi-Correlation Attention, instead of CNN, AE, VAE, etc., to capture the dynamic correlation, as well as non-linear, coupling, and high-order correlations. Assuming the original series in (T_1, T) is X^{T_e}, and the reconstructed series is $\tilde{X}^{T_e}$. Given one of the reconstructed series $\tilde{x}_i^{T_e}$ in $\tilde{X}^{T_e}$ as an example, Multi-Correlation Attention is implemented by Eq(9):

$$\begin{aligned}
&\tilde{x}_i^{T_e} = \text{MultiCorrelationAttention}[{x_i}^{T_e}(n)] \\
&= \sum_{j=(0,1,\ldots,i-1,i+1,\ldots,N)} soft\max(\frac{({Q_i}^{T_e})({K_j}^{T_e})^{Transpose}}{\sqrt{d}}){V_j}^{T_e}
\end{aligned} \tag{9}$$

where ${Q_i}^{T_e}$ is obtained through transformation of $x_i(n)$ in (T_1, T), ${K_i}^{T_e}$ and ${V_i}^{T_e}$ are obtained through transforming of surrounding $x_i(n)$ in (T_1, T). Multi-Correlation Attention first obtains the correlations between ${Q_i}^{T_e}$ and all ${K_j}^{T_e}$ by $\frac{({Q_i}^{T_e})({K_j}^{T_e})^{Transpose}}{\sqrt{d}}$, then reconstructes ${x_i}^{T_e}(n)$ by a weighted sum of ${V_j}^{T_e}$:

$$\tilde{x}_i^{T_e} = \sum_{j=(0,1,\ldots,i-1,i+1,\ldots,N)} sim_j {V_j}^{T_e} \tag{10}$$

sim is correlations and obtained through the real-time Dot-Product of ${Q_i}^{T_e}$ and ${K_j}^{T_e}$. Although transformation parameters of Q and K remain constant after offline training, as $x_i(n)$ and surrounding $x_i(n)$ keep changing online, the result of Q and K also keep changing, with the dynamic correlations(sim) between $x_i(n)$ and all series being captured, as well as non-linear and coupling correlations from ReLU activation function and Dot-Product operation. The ability to represent correlation can be increased by introducing Multi-head. Multi-Correlation Attention also captures the higher-order correlation by stacking multiple layers. Adding coherent accumulation (Appendix A) and dropout to improve robustness to noise. In summary, Multi-Correlation Attention consists of two layers at least. The first layer learns the dynamic, non-linear and coupling correlation, and the second layer learns high-order correlations.

3.2.3 Summary of Network Architecture

GenAD introduces Time-Series Attention and Multi-Correlation Attention to capture the temporal patterns and correlations of multivariate time series. The four layers model is constructed for

GenAD with independent Time-Series Attention and Multi-Correlation Attention. We can set the first and third layers of GenAD are Time-Series Attention, while the second and fourth layers are Multi-Correlation Attention. However, such an independent architecture will artificially reduce the representation of GenAD, so we adopt the fusion of Time-Series Attention and Multi-Correlation Attention. Each layer captures both the temporal patterns of series and the correlations among series. The fusion can be implemented by concatenating two kinds of attention or adopt the general attention representations which can support two kinds of attention at the same time. GenAD chooses the latter, and automatically learns the representation parameters.

3.3 METHOD FOR ANOMALY DETETION

GenAD evaluate the reconstruction error between the original signal and the reconstructed signal to detect anomalies. We set a two-level dynamic threshold (denoted as metric-level threshold $Gate_{metric}$ and entity-level threshold $Gate_{entity}$), which is derived from the anomaly rate. For the metric-level, if reconstruction error of one time series at time t is greater then $Gate_{metric}$, the time series is declared as anomalous. For an N-dimensional entity at time t', if there are M time series ($M > Gate_{metric}$) that are anomalous, then the entity is declared as anomalous. How to get anomaly threshold by anomaly rate is shown in Appendix B.

4 EXPERIMENTS

In this section, we first introduce the experimental datasets, comparison methods and evaluation metrics. We further conduct experiments to verify the effectiveness of our model for anomaly detection in multivariate time series. In addition, we also show the feasibility of our model in large-scale anomaly detection scenarios.

4.1 EXPERIMENTAL SETUP

Datasets. We use **SMD** (Server Machine Dataset) (Su et al., 2019) and **MSL** (Mars Science Laboratory rover) for empirical studies. **(i)** SMD is a 5-week-long dataset collected from a large Internet company, where each observation is equally spaced by 1 minute. SMD contains three groups of machines (denoted as **SMD-1**, **SMD-2** and **SMD-3** respectively), a total of 28 machines, and each of which contains 38-dimensional metrics. Each machine subset contains approximately 28000 time steps, and is divided into two parts of the same length as a training set and a testing set. **(ii)** MSL has 132,046 time steps, of which the training set size is 58317 and the testing set size is 73729. Compared with SMD, MSL contains more metrics, a higher anomaly rate and more types of anomalies.

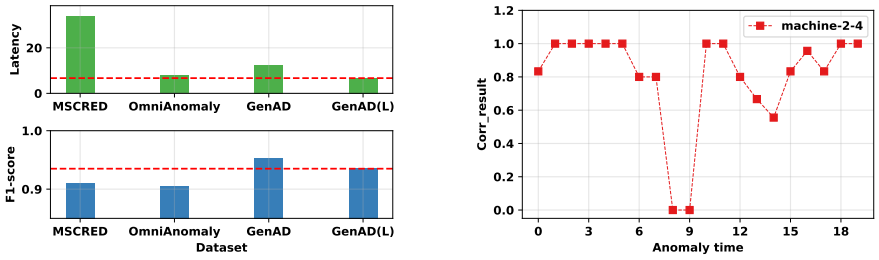
Baseline methods. We compare GenAD with the following baseline methods: MSCRED, LSTM-NDT(Hundman et al., 2018) and the state-of-the-art unsupervised method OmniAnomaly. Of these baselines, LSTM-NDT applies LSTM for multivariate time series prediction, MSCRED detects anomalies based on reconstruction errors, and OmniAnomaly is based on reconstruction probability.

Evaluation metrics. We use 3 metrics of **Precision**, **Recall**, and **F1-Score** to evaluate the anomaly detection performance of GenAD and baseline methods. In practice, abnormal observations usually appear continuously to form an anomaly segment. Generally, operation personnel care more about whether the anomaly detection model can detect a continuous anomaly segment, rather than finding every anomaly in the segment. Following the suggestion of (Su et al., 2019), we adopt a point-adjust approach (Xu et al., 2018)to calculate the evaluation metrics, that is, if any observation in the anomaly segment is detected, it is considered that the entire segment is correctly detected. In addition, we set a two-level dynamic threshold (denoted as single-dimensional threshold and multi-dimensional threshold), which is derived from the distribution of anomaly scores in the testing set.

Model parameters and thresholds. All models in the experiment use the same architecture and parameters, details are in Appendix C. We also show the thresholds used for each dataset. In addition, we explore the impact of important model parameters and the results are in Appendix D.

Table 1: Results of GenAD and baselines

Method	SMD-1			SMD-2			SMD-3			MSL		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
LSTM-NDT	0.773	0.298	0.431	0.767	0.282	0.412	0.819	0.428	0.557	0.452	0.389	0.418
MSCRED	0.869	0.853	0.861	0.922	0.898	0.910	0.839	**0.908**	0.872	0.746	**0.881**	0.808
OmniAnomaly	0.891	0.926	0.908	0.829	**0.994**	0.904	0.858	0.875	0.866	0.847	0.873	0.860
GenAD(Ours)	**0.924**	**0.941**	**0.933**	**0.941**	0.964	**0.952**	**0.884**	0.903	**0.893**	**0.855**	0.871	**0.863**



(a) (b)

Figure 2: Evaluation of the latency of anomaly detection. (a) F1 and latency on SMD-2. (b) The correlation between abnormal metrics when anomalies occur, taking machine-2-4 as an example.

4.2 Results compared with baselines

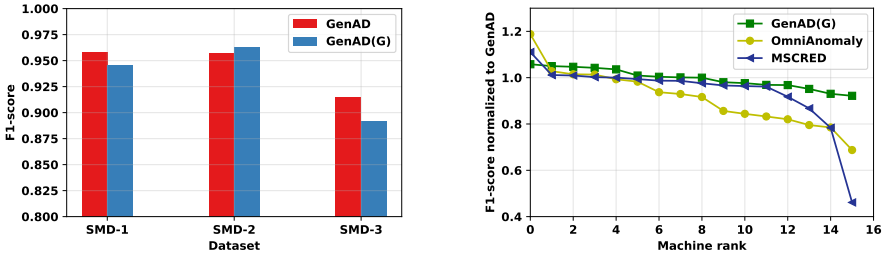
We evaluate GenAD and baselines on 4 datasets: SMD-1, SMD-2, SMD-3 and MSL. Table 1 reports the precision, recall, and F1-score of various anomaly detection methods, in which the best score is highlighted in bold. Note that the precision and recall are the average values of the datasets, and F1 is derived from the precision and recall. Although each of these methods provides an algorithm for calculating the anomaly threshold, they all need a parameter (e.g. the parameter "level" in OmniAnomaly) that quantifies the degree of anomaly as input. Therefore, we conduct multiple experiments to choose parameters to get the best results for all the methods.

Overall, GenAD performs better than all baseline methods, with F1-score increasing by 0.3% to 5% over the best baseline. We observe that reconstruction-based models perform better than prediction-based models(e.g. LSTM-NDT). This is because the prediction-based method is more sensitive to noise, and some time series are less predictable due to some uncontrollable factors (such as changes in the network environment). Compared with MSCRED and OmniAnomaly, GenAD performs better on representations of complex correlations and various temporal patterns. Moreover, introducing coherent accumulation and dropout techniques make GenAD robust to noise. Therefore, GenAD has the highest precision and F1-score on the 4 datasets.

4.3 Rapid detection of anomalies

As mentioned earlier, abnormal observations usually appear continuously, and an abnormal segment with a long duration is likely to cause a system failure. Therefore, it is essential to detect the anomaly of the abnormal segment as early as possible. In this section, we conduct experiments to evaluate the latency of anomaly detection, where latency is defined as the average value of the time points that have passed when anomalies are detected in the abnormal segment. We select the SMD-2 dataset with the best average performance of all anomaly detection models for the experiment, and the results are shown in Figure 2a. Compared to MSCRED and GenAD, OmniAnomaly has a smaller latency. This is due to a trade-off between a higher F1-score and a smaller latency, and the higher Precision of the threshold set by GenAD also leads to longer latency. To illustrate this, we slightly lower the threshold, denoted as GenAD(L). Compared with GenAD, GenAD(L) reduces the latency by 46.79% at the cost of a 1.79% drop in F1-score, and is better than MSCRED and OmniAnomaly in terms of latency and F1-score. A case of latency of anomaly detection is shown in Appendix E.

In addition, we explore why our method can quickly detect anomalies. According to the analysis in Appendix F, anomalies in multivariate time series have Ripple Effect, and GenAD detect anoma-



(a) (b)

Figure 3: Evaluation of the generality of GenAD. (a) Performance of GenAD and GenAD(G). (b) F1-score on each machine, compared to GenAD result.

lies based on the precise representation of correlation among time series, so as to infers anomalies quickly. We randomly select a machine, denoted as "machine-2-4", which has 20 anomaly segments and 38-dimensional metrics such as CPU load, memory usage, etc.. We further analyze whether there are correlations among the abnormal metrics. Specifically, for each anomaly segment $T_{anomaly}$, Given N is the number of abnormal metrics, and the Pearson Correlation is calculated within $T_{anomaly}$. According to Benesty et al. (2009), when Pearson Correlation Coefficient is greater than 0.4, correlation is considered to be moderate or strong. Given the correlation degree $Corr\_result$ in $T_{anomaly}$, which is formulated as $Corr\_result$=N_{corr}/N, where N_{corr} represents the metrics with Pearson Coefficient greater than 0.4. As shown in Figure 2b, except for two anomaly segments of 8 and 9, the $Corr\_result$ of the other anomaly segments is high, and the overall average correlation reaches 80.4%. It is worth noting that Pearson Coefficient only represents linear correlation, and there are also complex correlations among multivariate metrics such as dynamic, non-linear, coupling, and higher-order, so the actual $Corr\_result$ is much higher than 80.4%.

4.4 Generality for anomaly detection

It is impossible to deploy existing deep learning models on large-scale multivariate time series in real-world scenarios, due to the huge overhead in model training and parameter adjustment for each node-specific pattern. Unlike these models, GenAD pre-trains a general model for the representation of large-scale multivariate time series with self-supervision. Furthermore, for each node-specific pattern, only a small amount of data is required to fine-tune the model for the anomaly detection tasks.To learn the general representation of time series for anomaly detection, We use the data of the first 4 machines in each SMD-1, SMD-2 and SMD-3 for pre-training, and fine-tune the general model based on 10% of the training data in each test machine. As shown in Figure 3a, despite the reduction in training data, the performance of the generic model of GenAD (denoted as GenAD(G)) is not significantly reduced. Interestingly, GenAD (G) performs better than GenAD on SMD-2. This is understandable because GenAD may be over-fitted on some specific training sets and thus sensitive to noise. On the contrary, GenAD(G) obtains fewer details of specific data, which can prevent overfitting and get a better F1-score. Figure 3b shows the F1-scores of different anomaly detection algorithms normalized to GenAD on each tested machine. We observe that GenAD and GenAD(G) have higher F1 scores and better stability on all tested machines compared with the baseline method.

5 Conclution

In this paper, we propose a novel model named GenAD, a General representations of multivariate time series for Anomaly Detection. In GenAD we employ Time-Series Attention to represent the various temporal patterns of each time series and adopts Multi-Correlation Attention to represent the complex correlations among the multivariate time series. We also propose a general pre-training algorithm on large-scale multivariate time series, which can be easily transferred to a specific AD tasks with only a few fine-tuning steps. Through extensive experiments, GenAD outperforms the state-of-the-art approaches on four public datasets, while the generic model of GenAD achieves excellent performance with only 10% of the training data.

REFERENCES

Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, pp. 15–27. Springer, 2002.

Caglar Aytekin, Xingyang Ni, Francesco Cricri, and Emre Aksu. Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE, 2018.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pp. 1–4. Springer, 2009.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Sucheta Chauhan and Lovekesh Vig. Anomaly detection in ecg time signals via deep long short-term memory networks. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–7. IEEE, 2015.

Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. Arima models to predict next-day electricity prices. *IEEE transactions on power systems*, 18(3):1014–1020, 2003.

Santanu Das, Bryan L Matthews, Ashok N Srivastava, and Nikunj C Oza. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 47–56, 2010.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

Dragan Djurdjanovic, Jay Lee, and Jun Ni. Watchdog agent—an infotronics-based prognostics approach for product performance degradation assessment and prediction. *Advanced Engineering Informatics*, 17(3-4):109–125, 2003.

Alexandre Dolgui, Dmitry Ivanov, and Boris Sokolov. Ripple effect in the supply chain: an analysis and recent literature. *International Journal of Production Research*, 56(1-2):414–430, 2018.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, pp. 59–63, 2012.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. Robust subspace recovery layer for unsupervised anomaly detection. *arXiv preprint arXiv:1904.00152*, 2019.

Zhihan Li, Youjian Zhao, Rong Liu, and Dan Pei. Robust and rapid clustering of kpis for large-scale anomaly detection. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018.

Ping Liu, Yu Chen, Xiaohui Nie, Jing Zhu, and Dan Pei. Fluxrank: A widely-deployable framework to automatically localizing root cause machines for software service failure mitigation. In *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, 2019a.

Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 2019b.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89, pp. 89–94. Presses universitaires de Louvain, 2015.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

Ignacio Medina, David Montaner, Joaquín Tárraga, and Joaquín Dopazo. Prophet, a web-based tool for class prediction using microarray data. *Bioinformatics*, 23(3):390–391, 2007.

Ahmed Moumena and Abdelrezzak Guessoum. Fast anomaly detection using boxplot rule for multivariate data in cooperative wideband cognitive radio in the presence of jammer. *Security and Communication Networks*, 8(2):212–219, 2015.

Timothy J O'Shea, T Charles Clancy, and Robert W McGwier. Recurrent neural radio anomaly detection. *arXiv preprint arXiv:1611.00301*, 2016.

Randy Paffenroth, Kathleen Kay, and Les Servi. Robust pca for anomaly detection in cyber networks. *arXiv preprint arXiv:1801.01571*, 2018.

Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING, 2003.

Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1067–1075, 2017.

Siwoon Son, Myeong-Seon Gil, Yang-Sae Moon, and Hee-Sun Won. Anomaly detection of hadoop log data using moving average and 3-sigma. *KIPS Transactions on Software and Data Engineering*, 5(6):283–288, 2016.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan 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*, pp. 2828–2837, 2019.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487, 2016.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pp. 187–196, 2018.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pp. 3861–3870. PMLR, 2017.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1409–1416, 2019.

Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *arXiv preprint arXiv:1901.01588*, 2019.

Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674, 2017.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

A IMPROVE ROBUSTNESS BY COHERENT ACCUMULATION

There are noises in multivariate time series, which affect series reconstruction and abnormal detection. GenAD introduces coherent accumulation to obtain Information Gain, which is denoted as $Information\_Gain = 10 \times \log_{10} N$, where N is the number of accumulations. GenAD choose 32 times accumulation to improve robustness to noise.

B GET ANOMALY THRESHOLD BY ANOMALY RATE

Assuming that the anomaly rate is A_R, the anomaly threshold is $Gate$, and the single-dimensional or multi-dimensional data is E, then

$$P(E \geq Gate) = A_R \tag{11}$$

Assume that the probability density function of E is $f(e)$, and the probability distribution function $F(e) = \int_{-\infty}^{e} f(t)dt$, then Eq.(11) becomes

$$F(Gate) = \int_{-\infty}^{Gate} f(t)dt = A_R \tag{12}$$

We need to get the probability distribution function $F(e)$ of E, and then calculate the anomaly threshold $Gate$. A simple idea is to obtain $F(e)$ by analyzing the characteristics of E, for example, E obeys Gaussian-distribution or t-distribution, etc. However, this method is not available in large-scale service or equipment scenarios. Futhermore, (Siffer et al., 2017; Su et al., 2019) applies Extreme Value Theory (EVT) to estimate the parameters of the distribution of E. However, the complexity of this method is high, which makes it difficult to quickly obtain the anomaly threshold $Gate$. Different from the above methods, we estimate the probability density function $f(e)$ of E based on the sampled data of E,

$$f(e) = \sum y \, , y = \begin{cases} 1, if(e - \Delta e \leq E \leq e) \\ 0, else \end{cases} \tag{13}$$

Then we integrate $f(e)$ to get the probability distribution function $F(e) = \sum_{\min(E)}^{e} f(t)$, and get the dynamic threshold $Gate$ by Eq.(12). It is worth noting that in order to reduce the error of parameter estimation through sample data, we set $A'_R = \eta + A_R$ and Eq.(12) becomes $F(Gate) = \int_{-\infty}^{Gate} f(t)dt = A'_R$, where $\eta \in [-0.01, 0.01]$ is set to maximize the F1-score during the validation period. Subsequent experimental results show that this threshold selection method is simple but effective.

C MODEL PARAMETERS AND THRESHOLDS

All models in the experiment use the same architecture and parameters, as shown in Table 2. In addition, we also show the thresholds used for each dataset in Table 3, where threshold 1 represents the single-dimensional threshold, and threshold 2 represents the multi-dimensional threshold.

Table 2: Model parameters

attention head numbers	12
hidden layer numbers	4
training iterations	100000
dropout	0.1
training batch size	16
testing batch size	8
mask ratio	0.2

Table 3: The reference threshold of GenAD for the four datasets

	SMD-1										
	1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8			
threshold 1	0.999	0.9965	0.9995	0.9985	0.9995	0.999	0.9985	0.999			
threshold 2	0.997	0.991	0.99	0.99	0.997	0.99	0.99	0.99			
	SMD-2										
	2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8	2-9		
threshold 1	0.9995	0.9985	0.999	0.997	0.999	0.997	0.999	0.995	0.9985		
threshold 2	0.99	0.99	0.99	0.99	0.99	0.99	0.996	0.997	0.99		
	SMD-3										
	3-1	3-2	3-3	3-4	3-5	3-6	3-7	3-8	3-9	3-10	3-11
threshold 1	0.999	0.9985	0.9985	0.9965	0.9965	0.998	0.9975	0.999	0.997	0.9985	0.996
threshold 2	0.996	0.998	0.992	0.99	0.998	0.99	0.99	0.99	0.99	0.99	0.998
	MSL										
threshold 1	0.999										
threshold 2	0.99										

Table 4: F1-score of GenAD with different number of attention heads and hidden layers

(Attention heads, Hidden layers)	(12, 2)	(12, 6)	**(12, 4)**	(8, 4)	(16, 4)
F1-score	0.913	0.927	**0.933**	0.923	0.902

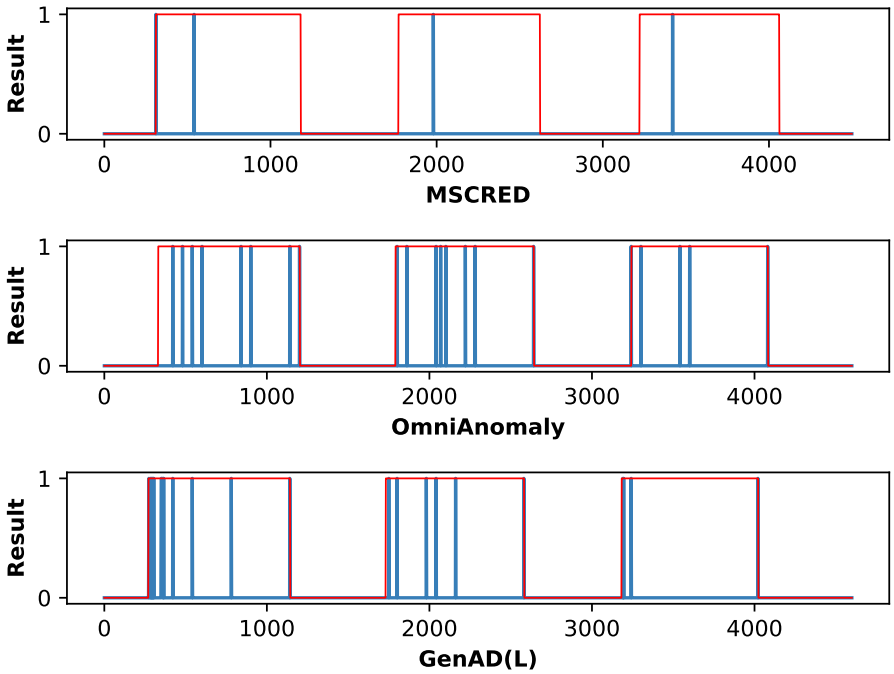


Figure 4: Case of latency of anomaly detection

D IMPACT OF NUMBER OF ATTENTION HEADS AND HIDDEN LAYERS

The number of attention heads and hidden layers is important for GenAD. Table 4 shows F1-score of GenAD on SMD-1 by varying different attention head numbers and hidden layer numbers. We observe that keeping the number of heads constant and changing the number of layers, the 4-layers perform best. The 2-layer Multi-Correlation Attention is weak in represent deep and high-order correlations, which leads to a decrease in F1-score; the 6-layer requires a higher amount of training data, and the model convergence is not as good as the 4-layer. In addition, we try the 12-layer attention, and the model can not converge, which further verifies the analysis results. Similarly, keeping 4-layers unchanged, and changing the number of heads, 12-head attention performs best. The 8-head Time-Series Attention has a decline in the ability to capture the number of periodic frequencies and trends, while the Multi-Correlation Attention has weak ability to capture dynamic, non-linear and coupling relationships, resulting in a decline in F1; 16 heads are consistent with the 6-layer analysis. Overall, we empirically set the number of heads and layers of all datasets to 12 and 4, respectively.

E STUDY OF LATENCY OF ANOMALY DETECTION

We randomly select a machine in SMD-2, denoted as "machine-2-2". The size of the testing set is approximately 28,000. Figure 4 shows the three longest duration anomaly segments of the machine. The red box represents the anomaly label, and the blue vertical line represents the anomalies detected by the algorithm. We observe that GenAD (L) has the highest timeliness in detecting abnormalities.

F RIPPLE EFFECT FOR RAPID DETECTION OF ANOMALY SEGMENT

GenAD finds anomalies by measuring the relationship, including the internal relationship of the series itself, and the relationship between the series and the surroundings in the current period. Here, we analyze the above-mentioned second relationship, and the other relationship are the same.

$$x_i{}^{T_1+\Delta t^{\frown}T} = \beta \times f_2(x_0{}^{T_1+\Delta t^{\frown}T}, x_1{}^{T_1+\Delta t^{\frown}T}, \cdots, x_{i-1}{}^{T_1+\Delta t^{\frown}T}, x_{i+1}{}^{T_1+\Delta t^{\frown}T}, \cdots, x_N{}^{T_1+\Delta t^{\frown}T})$$

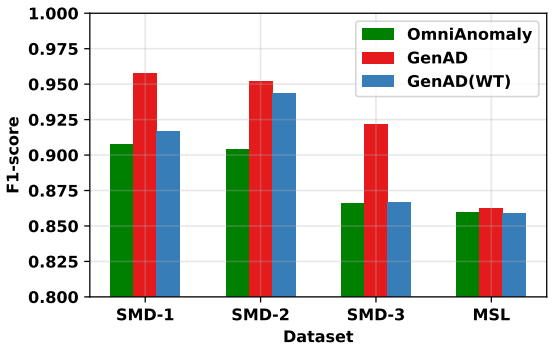


Figure 5: Model without Time-Series Attention

For the convenience of analysis, assuming there are only 3 time series, then

$$
\begin{aligned}
{x_0}^{T_1+\Delta t^{\sim}T} &= \beta_{0\_0} \times f_{12\_0}({x_1}^{T_1+\Delta t^{\sim}T}, {x_2}^{T_1+\Delta t^{\sim}T} + \beta_{0\_1} \times f_{12\_1}({x_1}^{T_1+\Delta t^{\sim}T} \times {x_2}^{T_1+\Delta t^{\sim}T})) \\
{x_1}^{T_1+\Delta t^{\sim}T} &= \beta_{1\_0} \times f_{02\_0}({x_0}^{T_1+\Delta t^{\sim}T}, {x_2}^{T_1+\Delta t^{\sim}T} + \beta_{1\_1} \times f_{02\_1}({x_0}^{T_1+\Delta t^{\sim}T} \times {x_2}^{T_1+\Delta t^{\sim}T})) \\
{x_2}^{T_1+\Delta t^{\sim}T} &= \beta_{2\_0} \times f_{01\_0}({x_0}^{T_1+\Delta t^{\sim}T}, {x_1}^{T_1+\Delta t^{\sim}T} + \beta_{2\_1} \times f_{01\_1}({x_0}^{T_1+\Delta t^{\sim}T} \times {x_1}^{T_1+\Delta t^{\sim}T}))
\end{aligned}
$$

The correlations here have dynamic, non-linear and high-order properties($f_{12\_0}(\cdot)$,$f_{02\_0}(\cdot)$,$f_{01\_0}(\cdot)$), and also has coupling properties ($f_{12\_1}(\cdot)$,$f_{02\_1}(\cdot)$,$f_{01\_1}(\cdot)$). β represents the weight value of each function. Assuming that ${x_0}^{T_1+\Delta t^{\sim}T}$ has changed, $\tilde{x}_0^{T_1+\Delta t^{\sim}T}$=${x_0}^{T_1+\Delta t^{\sim}T}$+$\Delta x_0$, the change of each time series is

$$
\begin{aligned}
\Delta x_0 &= \Delta x_0 \\
\Delta x_1 &= \beta_{1\_0} \times \frac{d(f_{02\_0}({x_0}^{T_1+\Delta t^{\sim}T}, {x_2}^{T_1+\Delta t^{\sim}T})}{dx_0}\Delta x_0 + \beta_{1\_1} \times \frac{d(f_{02\_1}({x_0}^{T_1+\Delta t^{\sim}T} \times {x_2}^{T_1+\Delta t^{\sim}T}))}{dx_0}\Delta x_0 \\
\Delta x_2 &= \beta_{2\_0} \times \frac{d(f_{01\_0}({x_0}^{T_1+\Delta t^{\sim}T}, {x_1}^{T_1+\Delta t^{\sim}T})}{dx_0}\Delta x_0 + \beta_{2\_1} \times \frac{d(f_{01\_1}({x_0}^{T_1+\Delta t^{\sim}T} \times {x_1}^{T_1+\Delta t^{\sim}T}))}{dx_0}\Delta x_0
\end{aligned}
$$

It can be seen from the formula that after ${x_0}^{T_1+\Delta t^{\sim}T}$ has changed, due to the relevance, other series have also changed. We can first measure the reconstruction error of each single series and then vote on all series (for example, meeting two series are abnormal at a time point) to find out whether multiple series are abnormal. The analysis method of the internal relationship of the series itself is similar. After an series is abnormal, the abnormal series and other related series have large reconstruction errors at the abnormal time, and the abnormality can be found as soon as possible through this method.

G Model without Time-Series Attention

We further evaluate the performance of the model without Time-Series Attention (denoted as GenAD (WT)), and the results are shown in Figure 5. We observe that GenAD performs best overall. Due to the loss of some temporal information, the performance of GenAD (WT) decreases, but it is still better than OmniAnomaly.