Measurement 191 (2022) 110791

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Variational transformer-based anomaly detection approach for multivariate time series

Xixuan Wang [a], Dechang Pi [a,*], Xiangyan Zhang [b], Hao Liu [a], Chang Guo [a]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[b] Beijing Institute of Spacecraft System Engineering, Beijing, China

ARTICLE INFO

Keywords:
Telemetry data
Transformer
Variational autoencoder
Multivariate time series
Anomaly detection

ABSTRACT

Due to the strategic importance of satellites, the safety and reliability of satellites have become more important. Sensors that monitor satellites generate lots of multivariate time series, and the abnormal patterns in the multivariate time series may imply malfunctions. The existing anomaly detection methods for multivariate time series have poor effects when processing the data with few dimensions or sparse relationships between sequences. This paper proposes an unsupervised anomaly detection model based on the variational Transformer to solve the above problems. The model uses the Transformer's self-attention mechanism to capture the potential correlations between sequences and capture the multi-scale temporal information through the improved positional encoding and up-sampling algorithm. Then, the model comprehensively considers the extracted features through the residual variational autoencoder to perform effective anomaly detection. Experimental results on a real dataset and two public datasets show that the proposed method is superior to the mainstream and state-of-the-art methods.

1. Introduction

The satellite is a complex system composed of many interrelated and coordinated devices, which has important strategic significance to the country [1]. Since satellites have been in a harsh outer space environment for a long time, unforeseen anomalies or failures may occur during their orbiting operation. Therefore, timely and effective anomaly detection can ensure the long-term stable operation of the satellite [2]. Telemetry data is the satellite on-orbit operation data collected by the sensors installed in various satellite components. These data record the information, including the satellite's temperature, power, current, etc., and they are important data for analyzing the operating status of the satellite and detecting anomalous states [4]. Abnormal patterns in telemetry data are likely to suggest defects in satellite components or that the external environment is changing dramatically. If the staff fails to detect and deal with the anomalies in time, it is likely to cause significant economic losses. Therefore, ground personnel analyzes telemetry data to detect satellite anomalies and monitor satellite operation status in real-time [3]. Traditional anomaly detection methods need to rely on expert experience to build complex feature engineering on telemetry data. With the development of technology and increased computing power, anomaly detection methods based on deep learning have begun to receive widespread attention.

The satellite is a complex structure composed of several subsystems, and there is a potential correlation between different time series of satellite telemetry data. If this potential correlation is ignored, the performance of the anomaly detection model will be seriously affected [4]. Therefore, the correlation between time series must be considered in multivariate time-series anomaly detection [538]. At present, multivariate time-series anomaly detection models that use potential correlations between sequences are primarily based on the graph neural network [11173738]. These algorithms treat each time series as a feature node on the graph. It extracts the potential correlation of multivariate time series data by constructing a feature relationship graph. However, the performance of such algorithms depends too much on the size of the feature relationship graph. When the graph is too sparse, or the number of feature nodes in the graph is too small, it will cause the performance bottleneck of the model. As a result, the anomaly detection effect of this type of algorithm is even inferior to the multivariate anomaly detection model that does not consider the potential correlation.

To solve the problems mentioned above, we propose a variational

* Corresponding author.
E-mail addresses: wangxixuan@nuaa.edu.cn (X. Wang), pinuaa@nuaa.edu.cn (D. Pi), 2918266367@qq.com (X. Zhang), liuhaocs@nuaa.edu.cn (H. Liu), gc_nuaa@126.com (C. Guo).

https://doi.org/10.1016/j.measurement.2022.110791
Received 26 September 2021; Received in revised form 9 January 2022; Accepted 23 January 2022
Available online 26 January 2022

Transformer-based model. The model no longer extracts the correlation information by constructing the feature relationship graph but captures the correlation through the self-attention mechanism. It reduces the impact of the dimensionality of the multivariate time series and the sparse correlations between sequences on the model's performance. The main contributions of this paper are summarized as follows:

(1) An unsupervised anomaly detection method based on Transformer is proposed, and experimental verification and analysis are carried out on the public and real datasets. The experimental results show that the method proposed in this paper is superior to the mainstream and state-of-the-art methods.
(2) The positional encoding of the Transformer is improved. We propose a global temporal encoding to add time-series information and period information to the data, which can help the model better capture the long-term dependencies in the sequence.
(3) A multi-scale feature fusion algorithm for time series data is proposed. By fusing the features of multiple time scales, we can make up for the lost detail information during the up-sampling of the data to obtain a more robust feature expression.
(4) A residual variational autoencoder architecture is proposed. It is a generative model, which can learn more robust local features. The special residual structure of this model can alleviate the Kullback-Leibler (KL) divergence vanishing problem and improve the generation ability of the model.

The remaining chapters of this paper are organized as follows. Section 2 introduces the basic principle of the Transformer and the related work of multivariate time-series anomaly detection and variational autoencoder. Section 3 presents the residual variational autoencoder model based on the multi-scale Transformer and points out the functions of each component. Section 4 compares the method proposed in this paper with the existing methods on a real dataset and two public datasets and analyzes the experimental results. Finally, Section 5 summarizes our work and points out future research directions.

2. Related work

In this section, we firstly introduce in detail the concept of anomaly and two classifications of anomalies in time series data. Then, we focus on unsupervised anomaly detection methods for multivariate time-series data and analyze the characteristics and shortcomings of each of the existing methods. Finally, we provide an overview of two deep learning models, named Transformer and variational autoencoder, which serve as major building blocks for our anomaly detection algorithm.

2.1. Multivariate time-series anomaly detection

Anomaly detection aims to detect unusual samples that deviate from most of the data. Typically, anomalous data can be connected to some problems or rare events such as structural defects or malfunctioning equipment, etc. [6]. The anomaly detection for time series is defined as finding abnormal sequences in a series [7]. Anomalies that occur in time series data can be divided into two categories as univariate and multivariate anomalies [3].

Univariate anomalies correspond to an unusual individual behavior affecting one specific parameter. Univariate anomalies can be classified into three main categories: point, context, and collective anomalies. Point anomalies are individual data instances that deviate from the rest of the data. Unlike the other two types of anomalies, point anomalies are the easiest to detect as data points can be treated independently during detection without considering temporal relationships. Contextual anomalies depend on the values of the surrounding data points, so it is necessary to identify contextual anomalies in the time series with the help of local and timing information in the data. Collective anomalies occur when a series of data points exhibit anomalous behavior together. As collective anomalies always occur in sequence over a reasonably long period, it is necessary to identify collective anomalies in the time series with the help of long-term dependencies in the data [22].

Multivariate anomalies are caused by the combined anomalous behavior of one or more parameters. The correlation between sequences makes multivariate anomalies more complex on top of univariate anomalies, which highlights the need for robust and flexible models [23]. As typical multivariate time-series data, satellite telemetry data has complex correlations between their sequences. Besides, this type of data often occurs with a periodic or seasonal mode, so accurate detection of anomalies in telemetry data is a challenging problem [8].

Based on whether the data is labeled or not, the existing anomaly detection algorithms can be classified into three categories as supervised, semi-supervised, and unsupervised [9]. The data with labels is often challenging to obtain in the production process, so most anomaly detection algorithms are based on unsupervised methods [10]. Depending on the modeling approach, the existing time-series anomaly detection algorithms can be classified into univariate and multivariate anomaly detection algorithms. Univariate anomaly detection algorithms strategies process the different sequences independently. And multivariate anomaly detection algorithms model multiple time series as a unified entity [3]. Due to the complex satellite structure and numerous sensors, most telemetry data are multi-dimensional time-series data [11], so the univariate time-series anomaly detection algorithm is difficult to apply to the satellite telemetry data.

According to the principle of the anomaly detection algorithm, the existing anomaly detection algorithms can be broadly classified into two categories, which are based on machine learning, and deep learning approaches. Anomaly detection algorithms based on machine learning include: 1) Methods based on linear models, such as the anomaly detection algorithm based on single-class support vector machines proposed by Ma et al. [12]. 2) Methods based on distance, such as the anomaly detection algorithm based on adaptive Mahalanobis distance proposed by Sarmadi et al. [13]. 3) Methods based on probability and density estimation, such as the anomaly detection method based on empirical cumulative distribution function proposed by Li et al. [14]. 4) Methods based on outlier division, such as the isolated forest algorithm proposed by Liu et al. [34]. Although machine learning-based anomaly detection algorithms have good interpretability, for telemetry data with high sampling frequency and long sampling time, anomaly detection algorithms based on deep learning are often more accurate than machine learning-based algorithms.

Due to the unique network structure of the recurrent neural network (RNN), this type of anomaly detection algorithm can capture the long-term dependence in time series data, so it has been extensively studied. Zhou et al. [15] proposed a variational LSTM (Long Short-Term Memory) learning model based on reconstructed feature representation. Through the compression network of variational reparameterization, the model can effectively avoid the loss of crucial information in dimensionality reduction. Kieu et al. [16] proposed two integration frameworks based on recurrent autoencoders, which improved anomaly detection accuracy through sparsely connected recurrent neural networks and integration methods. Although the algorithm based on the recurrent neural network has achieved excellent results in anomaly detection, it does not consider the correlation between different time series. Therefore, the effect of this type of algorithm in modeling satellite telemetry data with potential sequence correlation is not ideal. In response to this shortcoming, researchers consider using the graph neural network to capture possible correlations between time series. Ruiz et al. [17] proposed a general learning framework based on graph recurrent neural networks, which can consider the temporal dependence and sequence correlation of data at the same time. Xie et al. [11] proposed a dynamic threshold anomaly detection algorithm based on the graph neural network. The algorithm extracts feature correlation through graph building modules and extracts data's spatial and temporal dependence through space and time modules. Compared with the

recurrent neural network, the anomaly detection algorithm based on the graph neural network effectively uses the correlation information between time series and improves anomaly detection accuracy. However, when dealing with multivariable time series with few dimensions or insufficient close relationships between sequences, the feature relationship graph constructed by the graph neural network is too small or too sparse. It makes the information that the graph neural network model can extract from the data limited, leading to the performance bottleneck of the algorithm.

To reduce the impact of the dimension of time series and the sparse correlations between sequences on the model's performance, we no longer extract the correlation between sequences from the feature relationship graph. But we use the self-attention mechanism of the Transformer model to capture the correlation in feature dimensions. The Transformer was first proposed by Vaswani et al. [18], which is an autoencoder model based on the self-attention mechanism. The Transformer model breaks the limitation that the recurrent neural network cannot be calculated parallel. And the number of operations required for the Transformer model to calculate the association between two sequences does not increase with the distance. Therefore, it has received widespread attention. Zhou et al. [19] proposed a Transformer-based long-term sequence prediction model. This model reduces the time and space complexity of the traditional self-attention mechanism through the ProbSparse self-attention mechanism and the self-attention distillation technology. Wu et al. [20] proposed a new decomposition architecture with an autocorrelation mechanism, which embeds the sequence decomposition unit in the Transformer model to achieve progressive prediction and improve the prediction accuracy. The documents mentioned above have proved the great potential of the Transformer model in the field of time-series data. Still, there is little research on anomaly detection algorithms based on Transformer at present. Therefore, this paper makes an in-depth exploration of the application of the Transformer model in the field of multivariate time-series anomaly detection.

2.2. Basic principles of transformer

The Transformer model is based on the autoencoder architecture and consists of the multi-head attention mechanism module and the feedforward neural network module. Each sub-module of the model uses the residual connection and layer normalization to prevent model degradation. The architecture diagram of the model is shown in Fig. 1 (a).

The multi-head attention mechanism of the Transformer model is composed of multiple self-attention mechanism modules. The calculation process of the self-attention mechanism is shown in Fig. 1 (b). We assume that $X = (X_1, X_2, \cdots, X_n)$ represents input data with N features, and linearly transform the input data X to obtain the query vector Q, the key vector K, and the value vector V. The attention score matrix obtained through the self-attention mechanism satisfies Eq. (1).

$$Attention(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_K}})V \tag{1}$$

In Eq. (1), d_K represents the dimension of the key vector sequence. The multi-head attention mechanism is to input the query vector Q, the key vector K, and the value vector V into multiple self-attention mechanism modules after being mapped by a fully connected neural network. Finally, the outputs of multiple self-attention mechanism modules are spliced, and then the output is integrated by a fully connected layer. At this time, the attention score matrix of the module satisfies the Eq. (2).

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, \cdots, head_h)W^O \\ where\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2}$$

In Eq. (2), W^Q, W^K, W^V, and W^O represent the parameter matrix of the fully connected layer. The dimension d_K of the key vector satisfies $d_K = d_{model}/h$, where d_{model} represents the unified input dimension of the Transformer model, and h represents the number of self-attention mechanism modules. The calculation process of the multi-head attention mechanism is shown in Fig. 1(c).

Because the Transformer model abandons the use of recurrent neural networks and convolutional neural networks as the basic architecture for building models, it is difficult for the Transformer model to extract the order of the sequence directly. To make the model use the sequential information of the input data, the Transformer model uses the positional encoding function to attach the sequential information to the original data. The positional encoding function satisfies Eq. (3).

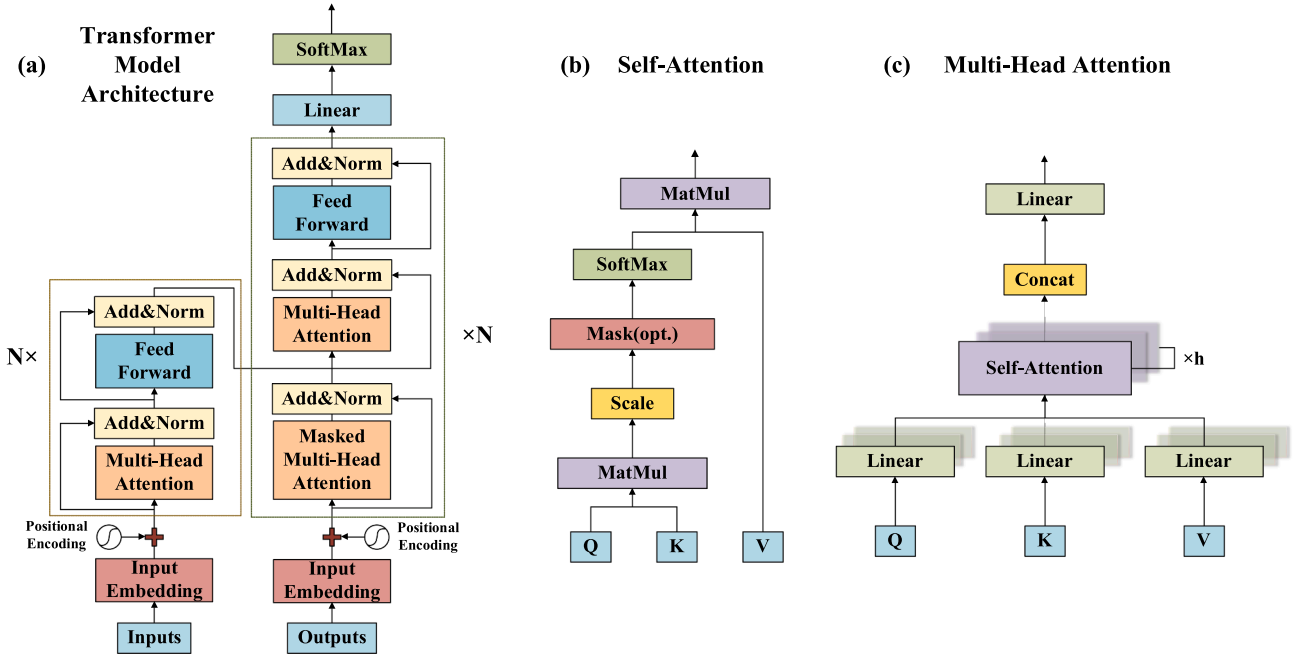


Fig. 1. Fundamental structure diagram of Transformer.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \quad (3)$$

where *pos* represents the position of the data, and *i* represents the dimension of the data. The trigonometric function is selected as the coding function because the position vector PE_{pos+k} at any position can be expressed as a linear function of PE_{pos} and is not limited by the length of the sequence.

2.3. Variational autoencoder

Although the Transformer model obtains strong feature extraction ability by relying on the self-attention mechanism, it still has some inherent defects of the autoencoder architecture model. For anomaly detection, we can use the autoencoder architecture model to learn the characteristics of normal data. But the autoencoder is only trained with as little reconstruction loss as possible, regardless of how the hidden space encoded by the encoder is organized. It causes the autoencoder to use any possibility of overfitting to complete the task as much as possible, resulting in a lack of regularity in the hidden space. The lack of regular hidden space will cause the autoencoder to give meaningless content when decoding normal data other than the training data, resulting in a more significant reconstruction error and judging the normal data as an abnormality. To solve this misjudgment problem, we introduce the variational autoencoder to make the hidden space regular.

Variational autoencoder (VAE) was first proposed by Kingma et al. [21] and received widespread attention once it was proposed. Lin et al. [22] proposed an anomaly detection algorithm based on the VAE-LSTM hybrid model, which forms robust local features through the VAE module, and uses the LSTM module to capture the long-term correlation in the sequence on top of the features inferred by the VAE module. Li et al. [23] proposed a smoothly induced sequence variational autoencoder. The model has effectively trained through the designed stochastic gradient variational Bayesian estimation and then performs the robust assessment and anomaly detection on multi-dimensional time series. Liu et al. [24] proposed an adversarial variational autoencoder based on sparse dictionary learning. The intrinsic features are extracted from the data through GAN (Generative Adversarial Networks) and VAE models. And the influence of random noise is eliminated through sparse dictionary learning, which improves the fault recognition performance. The wide application of the variational autoencoder in the field of anomaly detection proves the effectiveness of this model.

3. Multiscale transformer-based residual variational autoencoder

Compared with other models, the Transformer model can extract the correlation between different features, so the Transformer model has significant advantages in dealing with sequence problems. However, the traditional Transformer model can only extract the local sequential information in the data with the help of the original positional encoding and cannot extract the global time-series information. Therefore, the traditional Transformer model is not effective in handling time series data. Besides, before inputting the time-series data to the Transformer model, it needs to be mapped into a high-dimensional vector that matches the model input by up-sampling. But in the process of up-sampling, the detailed information is lost because the mapping relationship constructed is not accurate enough. In view of the above shortcomings, we propose a multiscale Transformer-based residual variational autoencoder model (MT-RVAE). The model structure is shown in Fig. 2.

The model comprises four parts: positional encoding module, multi-scale feature fusion module, feature-learning module, and data reconstruction module. The positional encoding module provides the model local sequential information and global time-series information. The multi-scale feature fusion module is used to make up for the missing details of high-level features to obtain a more robust feature expression. The feature-learning module learns the complex dependencies of multivariate time-series in both temporal and feature dimensions through the Transformer, and it learns more robust local features through the residual variational autoencoder. The data reconstruction module adjusts the output format of the model through the one-dimensional convolution layer and the fully connected layer. This section will introduce the four parts of the model in detail.

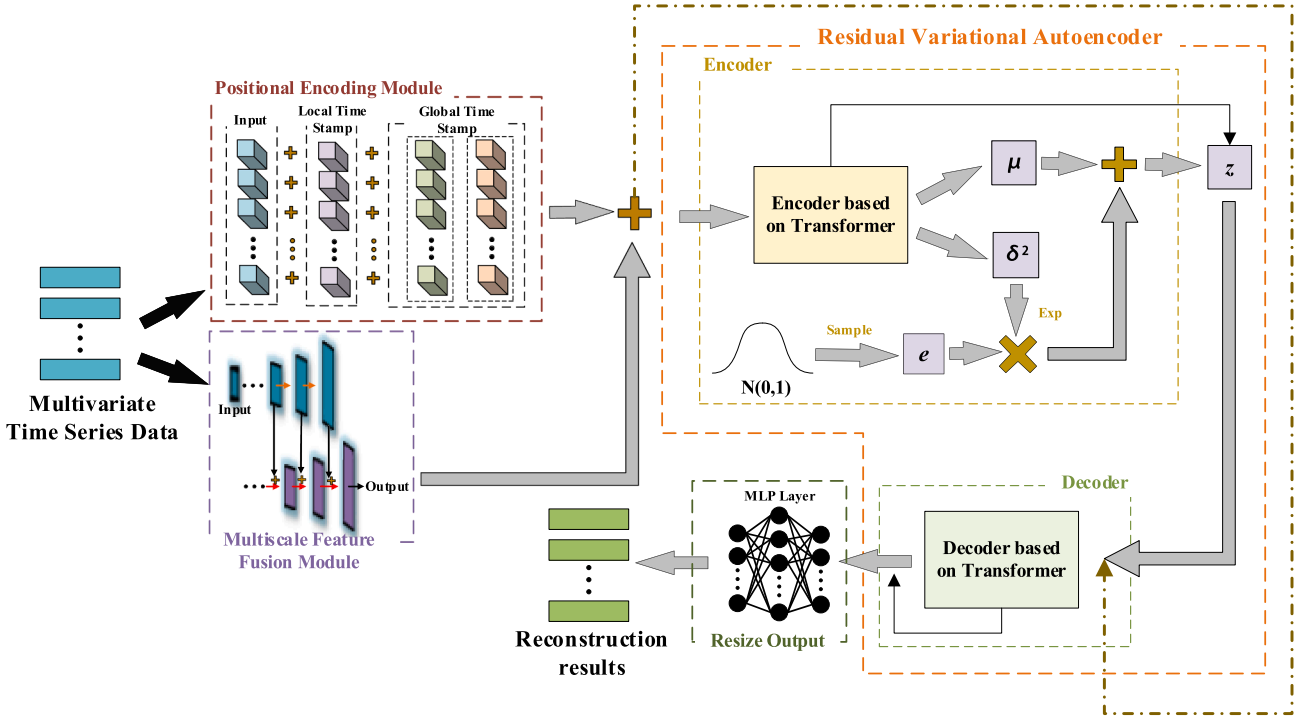


Fig. 2. Model structure diagram of residual variational autoencoder based on multiscale Transformer.

3.1. Positional encoding

Through positional encoding, the Transformer model learns the order of the sequence while learning the characteristics of the data. But the original positional encoding algorithm can only add the sequential information to the data in a single time window. In different time windows, the additional sequential information is the same, which cannot reflect the temporal and long-term dependence of the time series data. Therefore, if we only use this local temporal encoding algorithm to attach sequential information to the input data, the model will fail to learn the data's time pattern and long-term dependence. This makes the Transformer model unable to be used directly for time series anomaly detection.

In order to make up for the shortcomings of local temporal encoding, we proposed a global temporal encoding with periodic information and time-series information based on Zhou et al. [19]. Global temporal encoding and local temporal encoding together form the positional encoding module of this paper, as shown in Fig. 3. The global temporal encoding includes time-series encoding and periodic encoding. Time-series encoding can decompose the time stamp information and encode it into a vector. Because of the periodic characteristics of complex equipment such as satellites, the periodic encoding can use Fourier transform to analyze the periodic information of the data and encode it.

The timestamp of time series data is a character string in date format. We need to decompose the date format string into specific numeric information, such as hours, minutes, and seconds. Then this digital information is encoded and normalized by Eqs. (4) and (5) according to their characteristics.

Assuming that the timestamp at time i is $date_i$, the $date_i$ is input into the coding function and converted into a time-series encoding vector TPE_i, as shown in Eq. (4).

$$TPE_i = F(date_i) \tag{4}$$

$$F(date_i) = f_1(month) + f_2(day) + f_3(hour) + \cdots + f_n(second)$$

$$\begin{cases} f_1(month) = \dfrac{month - 1}{11} - 0.5 \\ f_2(day) = \dfrac{day - 1}{30} - 0.5 \\ f_3(hour) = \dfrac{hour}{23} - 0.5 \\ \vdots \\ f_n(second) = \dfrac{second}{59} - 0.5 \\ (month, day, hour, \cdots, second) \in date_i \end{cases} \tag{5}$$

In Eq. (5), $(month, day, hour, \cdots, second)$ is the numerical information such as hour, minute, and second, obtained from the time stamp $date_i$, and $(f_1, f_2, f_3, \cdots, f_n)$ is the corresponding coding formula for different date data. In order to normalize all types of date data to the interval of $[-0.5, 0.5]$, the date data whose value starts from one needs to be subtracted by one.

We use Fourier transform to analyze the period of time series data from the frequency domain perspective. The Fourier transform is derived from the Fourier series and Euler formula and is a generalized form of the Fourier series. The Fourier transform formula is shown in Eq. (6).

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \tag{6}$$

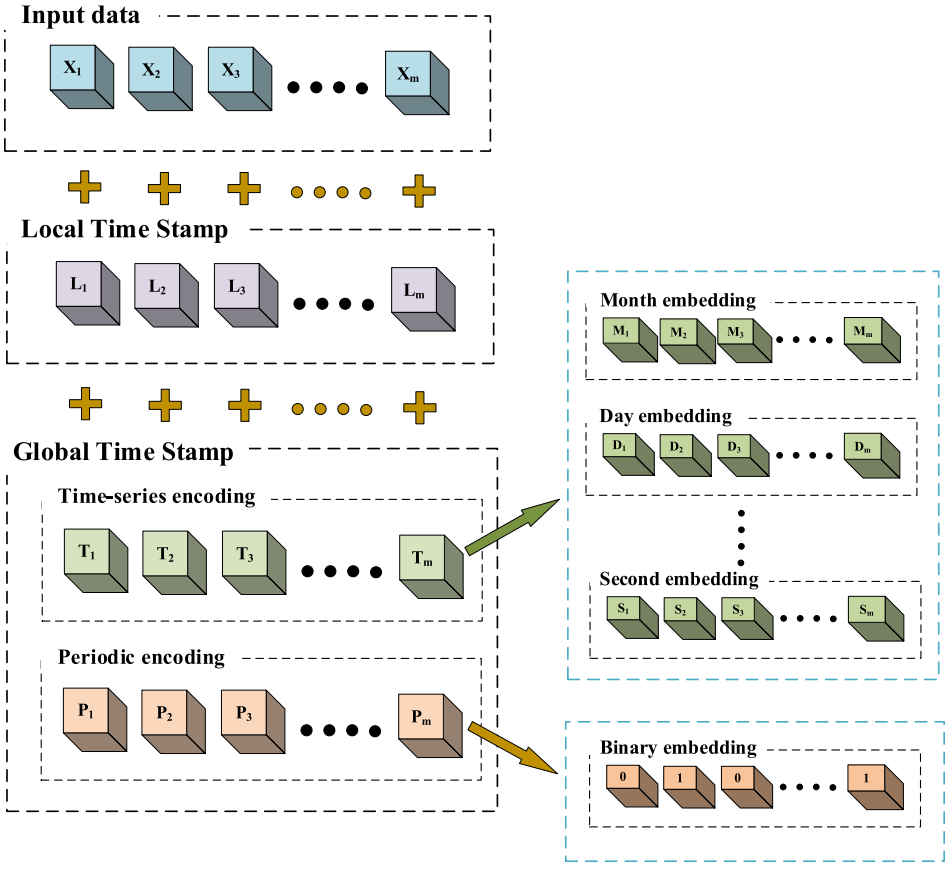


Fig. 3. Positional encoding module structure diagram.

where e is a natural constant, i is an imaginary unit, $F(\omega)$ represents the image function of $f(t)$, and $f(t)$ represents the original function of $F(\omega)$. The Fourier transform can decompose non-periodic time series data into a set of harmonics. With the help of the frequency spectrum of the harmonics, we can find the harmonics that have the most significant impact on the time-series data, as shown in Fig. 4. Through the harmonic frequency, the harmonic period can be calculated, that is, the main period of the time series data.

The periodic encoding module uses the obtained period information to encode the time series data. The periodic encoding vector PPE_t at time t satisfies Eq. (7).

$$PPE_t = [\frac{t}{T}] \mod 2 \tag{7}$$

In Eq. (7), t represents the timestamp after digitization, T represents the period of the data, [] is the rounding function, and mod is the remainder operator. If the time series data is a group of sine waves with tiny amplitude after Fourier transform, then the DC component is the most influential to the time series data. The DC component is a sine wave with an infinite period. That is, the frequency approaches zero. At this time, the main period of the time series data approaches infinity. Its periodic encoding formula is shown in Eq. (8).

$$PPE_t = [\frac{t}{T \to \infty}] \mod 2 \tag{8}$$

From Eq. (8), we can conclude that when the period T approaches infinity, no matter the value of t, the result of $[t/T]$ is always equal to zero. Therefore, periodic encoding does not add extra information to non-periodic time series data, and the periodic encoding does not affect it. This proves the versatility of the periodic encoding formula proposed in this paper for periodic data and non-periodic data.

Finally, the periodic encoding vector and the time-series encoding vector are embedded and projected through an Embedding layer. The embedding layer maps the vectors to the high-dimensional vectors that match the model input. Then the result is appended to the input data together with the local temporal encoding vector. Since the Embedding layer is also a learnable neural network, the Embedding layer will learn the mapping relationship in the continuous training process.

3.2. Multiscale feature fusion

Before the time series data is input to the Transformer model, it needs to pass through a fully-connected layer or a one-dimensional convolutional layer to map it into a high-dimensional vector that matches the model input. However, a fully-connected layer or a one-dimensional convolutional layer alone cannot fit an accurate mapping relationship, which results in a large amount of detailed information being lost in the up-sampling of the input data. As a result, the model's sensitivity to faulty data is reduced, leading to the inability of the model to detect some faulty data in the time series data. To obtain a more robust feature representation, we introduce the feature pyramid structure [25] to make up for the missing detail information in the up-sampling process. However, the existing feature pyramid structure can only handle image data [2627]. When the existing feature pyramid structure is applied to time-series data, it fails to compensate for the missing detail information in the mapping process of the data from low-dimensional vectors to high-dimensional vectors. Moreover, the structure even adds noise to the data, which reduces the anomaly detection performance of the model. This is contrary to our intention of using the method to enrich the detailed information of higher-level features and improve the anomaly detection performance of the model. Therefore, we propose a multi-scale feature fusion algorithm for time series data. The algorithm process is shown in Fig. 5.

Unlike the traditional feature pyramid structure, the multi-scale feature fusion module only convolves and up-samples the data in the time dimension. We first use multiple one-dimensional convolutional layers $F_{conv1d} = (F^1_{conv1d}, F^2_{conv1d}, \cdots, F^m_{conv1d})$ to iteratively map the data to features $C = (C_0, C_1, \cdots, C_m)$ of multiple time scales, as shown in Eq. (9).

$$\begin{cases} C_0 = Src \\ C_1 = F^1_{conv1d}(C_0) \\ C_2 = F^2_{conv1d}(C_1) \\ \vdots \\ C_m = F^m_{conv1d}(C_{m-1}) \end{cases} \tag{9}$$

In Eq. (9), Src is the input data of the module, and m is the number of convolutional layers, which is a hyper parameter that needs to be set manually.

Up-sampling $F_{unsample}$ is performed from the initial time scale feature C_0. Add the up-sampling results and the feature data of the next scale to obtain the new feature data P_1. Repeat the above operation to get the data P_m integrating all time scale features, as shown in Eq. (10).

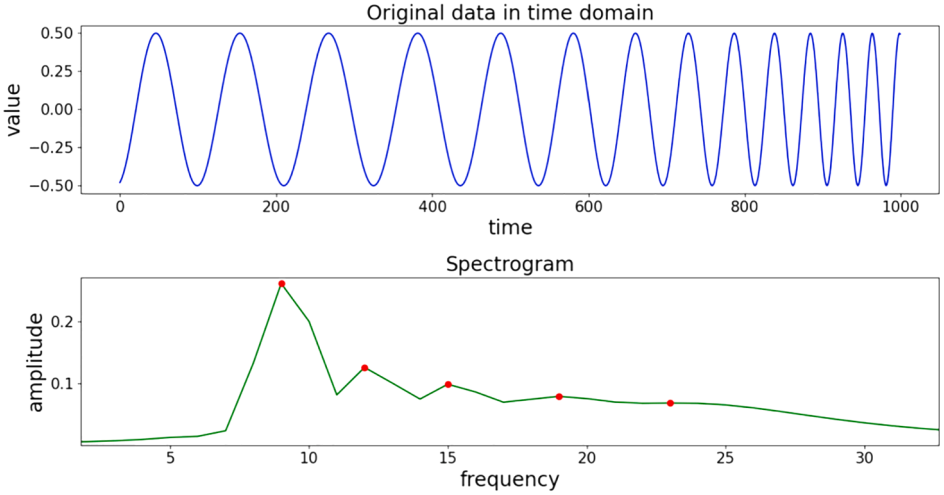


Fig. 4. Spectrogram after Fourier Transform.

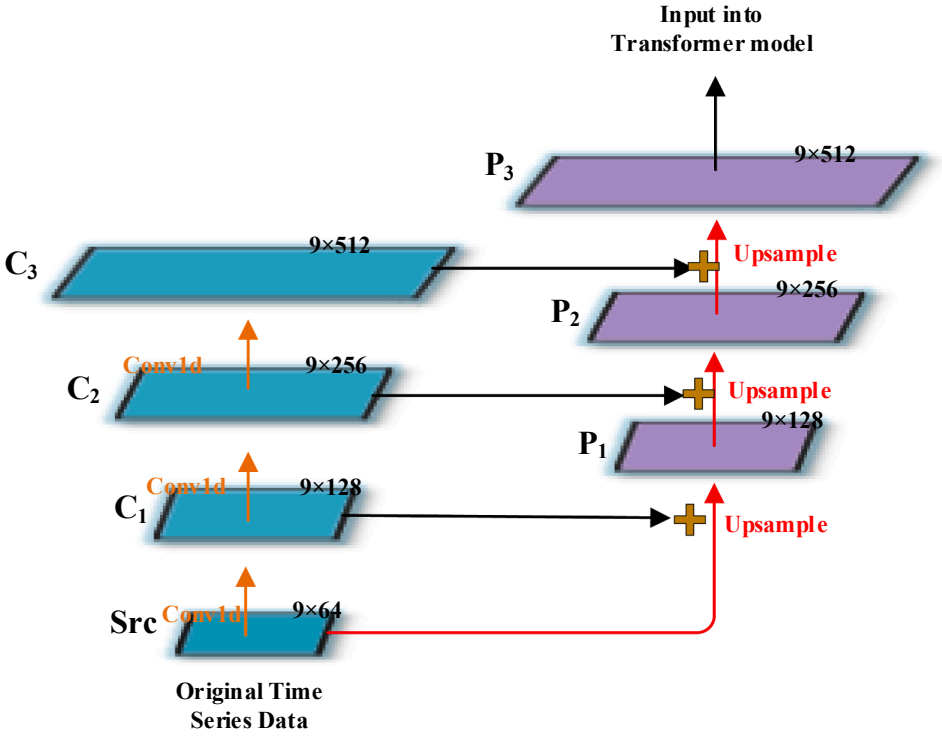


Fig. 5. Schematic diagram of the multi-scale feature fusion.

$$\begin{cases} P_1 = F^1_{unsample}(C_0) + C_1 \\ P_2 = F^2_{unsample}(P_1) + C_2 \\ P_3 = F^3_{unsample}(P_2) + C_3 \\ \vdots \\ P_m = F^m_{unsample}(P_{m-1}) + C_m \end{cases} \tag{10}$$

The data P_m that integrates all scale features can combine the detailed information of the original data and the rich semantic information of high-level features to obtain a more robust feature expression.

We use the one-dimensional transposed convolutional network as the up-sampling method. Other up-sampling methods need to select an interpolation algorithm manually, and the parameters of the interpolation algorithm are fixed during the model training process. Therefore, the gradient descent algorithm cannot be used to learn the parameters of the up-sampling module, and the optimal solution cannot be obtained. The transposed convolutional network does not require pre-defined interpolation methods and has learnable parameters, which can be optimally up-sampled with the training of the entire model. It should be noted that the convolution step stride of the transposed convolution network needs to divide the size of the transposed convolution kernel. If it cannot be divided, it will cause the data after transposed convolution to have checkerboard noise, which will affect the effect of feature fusion, as shown in Fig. 6 (b).

3.3. Residual variational autoencoder

In order to make the Transformer model more suitable for anomaly detection problems, we modify the Transformer model from an autoencoder architecture to a variational autoencoder architecture. The variational autoencoder architecture can ensure that the hidden space encoded by the Transformer encoder is sufficiently regular. It prevents the Transformer model from having significant reconstruction errors when decoding normal data other than the training data, thereby effectively improving the performance of the anomaly detection algorithm.

Different from the autoencoder model, the loss function of the variational autoencoder model can be divided into two parts, namely the reconstruction error term $Loss_{res}$ and the information divergence term $Loss_{kl}$, as shown in Eq. (11). The reconstruction error term represents the error between the reconstruction result of the decoder and the original data. The information divergence term represents the Kullback-Leibler (KL) divergence of the posterior distribution in the hidden space and the standard normal distribution.

$$Total\_loss = Loss_{res} + Loss_{kl}$$

$$\begin{cases} Loss_{res} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \widetilde{X}_i) \\ Loss_{kl} = \sum_{i=1}^{n} KL(p(Z_i|X_i)||p(Z_i)) = \sum_{i=1}^{n} KL(N(\mu_i, \theta_i^2)||N(0,1)) \\ \theta_i^2 = exp(\delta_i^2) \end{cases} \tag{11}$$

In the Eq. (11), X_i represents the original data at time i, $\widetilde{X}_i$ represents the reconstruction result of the decoder at time i, n represents the total duration of the time series data, and Z_i represents the hidden space vector at time i. $p(Z_i|X_i)$ represents the posterior distribution of the hidden space data under the condition that the original data X_i is known at time i, and the posterior distribution satisfies the Gaussian distribution $N(\mu_i, \theta_i^2)$ with mean μ_i and variance θ_i^2. The variance θ_i^2 satisfies the formula $\theta_i^2 = exp(\delta_i^2)$, exp is an exponential function based on the natural constant e. δ_i and μ_i are the parameters learned by the model. The purpose of using the exponential function exp is to control the value range of the variance θ_i^2 to $[1, +\infty]$ and prevent the value of the information divergence item $Loss_{kl}$ from appearing negative.

$p(Z_i)$ represents the prior distribution of the hidden space vector Z_i at time i, and the prior distribution is a standard normal distribution. In addition, the hidden space vector Z_i at time i satisfies Eq. (12).

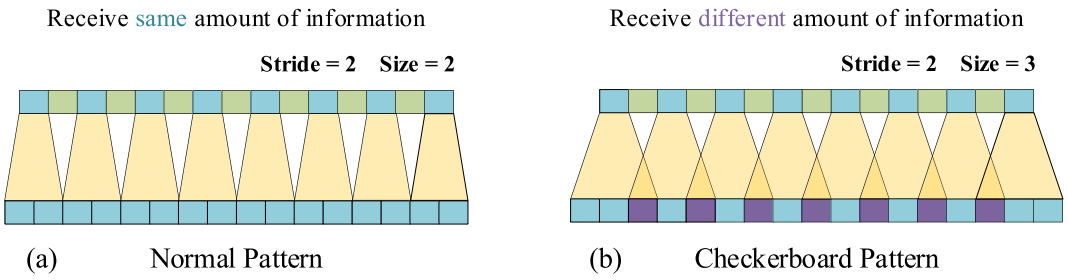


Fig. 6. Checkerboard effect of transposed convolution.

$$\widetilde{X}_i = g(Z_i) = g(\mu_i + exp(\delta_i^2)*\xi), \xi \sim N(0,1) \tag{12}$$

The function g in Eq. (12) represents the mapping relationship fitted by the Transformer decoder, and ξ represents the noise data sampled from the standard normal distribution $N(0,1)$.

When the decoder of the model is too powerful, or the weight of the $Loss_{kl}$ term is too high, the model encodes all the input data into a standard normal distribution in order to obtain the minimum value of the loss function, which causes the hidden space not to contain any information of the original data. As a result, the decoder only relies on the noise data sampled from the standard normal distribution for decoding, which makes the variational autoencoder architecture invalid, as shown in Eq. (13).

$$\begin{cases} p(Z_i|X_i) \sim N(0,1) \\ \widetilde{X}_i = g(Z_i) = g(\xi), \xi \sim N(0,1) \end{cases} \tag{13}$$

The phenomenon that causes the posterior distribution $p(Z_i|X_i)$ of hidden space data to degenerate to the standard normal distribution is called the KL divergence vanishing problem. Due to Transformer's powerful feature extraction capabilities, the Transformer-based variational autoencoder is more prone to the disappearance of divergence.

The problem of the KL divergence vanishing can be roughly divided into two solutions. One is to set dynamic coefficients for the $Loss_{kl}$ term, which reduces the contribution of the $Loss_{kl}$ term to the loss function at the beginning of training [29]. However, this method requires a long process of finding the suitable coefficient, which increases the labor cost. The other is to reduce the decoder's performance, thereby increasing the contribution of the reconstruction error term to the loss function [30]. However, a decoder with low performance will reduce the upper limit of the model generation ability.

In response to this problem, we propose a residual variational autoencoder structure, which uses residuals to alleviate the disappearance of divergence. Different from the ordinary residual structure, the residual structure proposed in this paper does not connect the encoder and the decoder but combines the residuals separately. This special residual structure prevents the encoder information from leaking to the decoder too much to avoid the failure of the variational autoencoder structure. In addition, we also added a time window-based attention mechanism to the encoder of the model. Combined with the Transformer, the model can better learn the distribution of data in the hidden space. The above structures together constitute the feature-learning module of this model, and the network structure is shown in Fig. 7.

The residual structure adds a constant term $\widehat{\mu}_i$ to the posterior distribution $p(Z_i|X_i)$. At this time, the posterior distribution satisfies Eq. (14).

$$p(Z_i|X_i) \sim N(\mu_i + \widehat{\mu}_i, exp(\delta_i^2)) \tag{14}$$

The mean μ_i and variance δ_i^2 are the parameters learned by the model. The hidden space vector Z_i at time i satisfies Eq. (15).

$$Z_i = \widehat{\mu}_i + \mu_i + exp(\delta_i^2)*\xi = \sum_{j=1}^{k} f_j(X_i) + \mu_i + exp(\delta_i^2)*\xi \tag{15}$$

where k represents the number of network layers of the encoder, f_j represents the mapping function fitted by the encoder at layer j, X_i represents the original data at time i, $\widehat{\mu}_i$ represents the output sum of the encoders of each layer, and ξ represents the noise data sampled from the standard normal distribution $N(0,1)$.

At this time, the encoder learns the residual between the approximate posterior distribution and the actual posterior distribution, not the posterior distribution itself. When the problem of the disappearance of divergence occurs again, that is, $\mu_i = 0$, $\delta_i^2 = 0$, the posterior distribution $p(Z_i|X_i)$ of the hidden space satisfies the Eq. (16).

$$p(Z_i|X_i) \sim N(\widehat{\mu}_i, 1) \tag{16}$$

The hidden space vector Z_i at time i satisfies Eq. (17).

$$Z_i = \widehat{\mu}_i + \xi = \sum_{j=1}^{k} f_j(X_i) + \xi \tag{17}$$

It can be seen from Eq. (16) and Eq. (17) that the existence of the constant term $\widehat{\mu}_i$ prevents the posterior distribution $p(Z_i|X_i)$ of the hidden space from degenerating to the meaningless standard normal distribution. For the hidden space vector Z_i, if there is no constant term $\widehat{\mu}_i$, there will be no information about the input data in the hidden space, only the noise data that meets the standard normal distribution. Therefore, the constant term $\widehat{\mu}_i$ also prevents the decoder of the model from only reconstructing from the noise data and improves the lower limit of the model generation ability.

The residual structure ensures that the output of the encoder can provide helpful information for the decoder, thus making the model more dependent on the hidden vector of the encoder. Indirectly, it increases the weight of the $Loss_{res}$ term in the loss function and prevents the disappearance of the KL divergence. At the same time, the Gaussian noise ξ prevents the addition of the residual structure from degrading the model to the form of an autoencoder, which ensures the generation ability of the model.

In order to further improve the feature extraction ability of the model, we add a time dimension attention mechanism to the residual variational autoencoder. The Transformer-based variational autoencoder can capture the correlation of data in feature dimensions through the self-attention mechanism. However, the self-attention mechanism does not consider the autocorrelation of data features in the time dimension in extracting feature information. For time-series data, the features themselves change over time. The data at different moments have different effects on the data at the current moment. In order to extract the importance of data in different time steps, we introduce a time-dimensional attention mechanism, which adds weight information to the data in the same time window. The time attention module is shown in Fig. 8.

Assuming that $\lambda^m = (\lambda_1^m, \lambda_2^m, \cdots, \lambda_l^m)$ is a feature of input data λ in a time window, where l is the width of the time window. After λ^m is input to the time attention module, the MLP network calculates the score S^m at each moment of λ^m. Then the Softmax function calculates the weight W^m at each moment of λ^m according to the score S^m. Finally, the weight information W^m is added to the original data λ^m to obtain the time

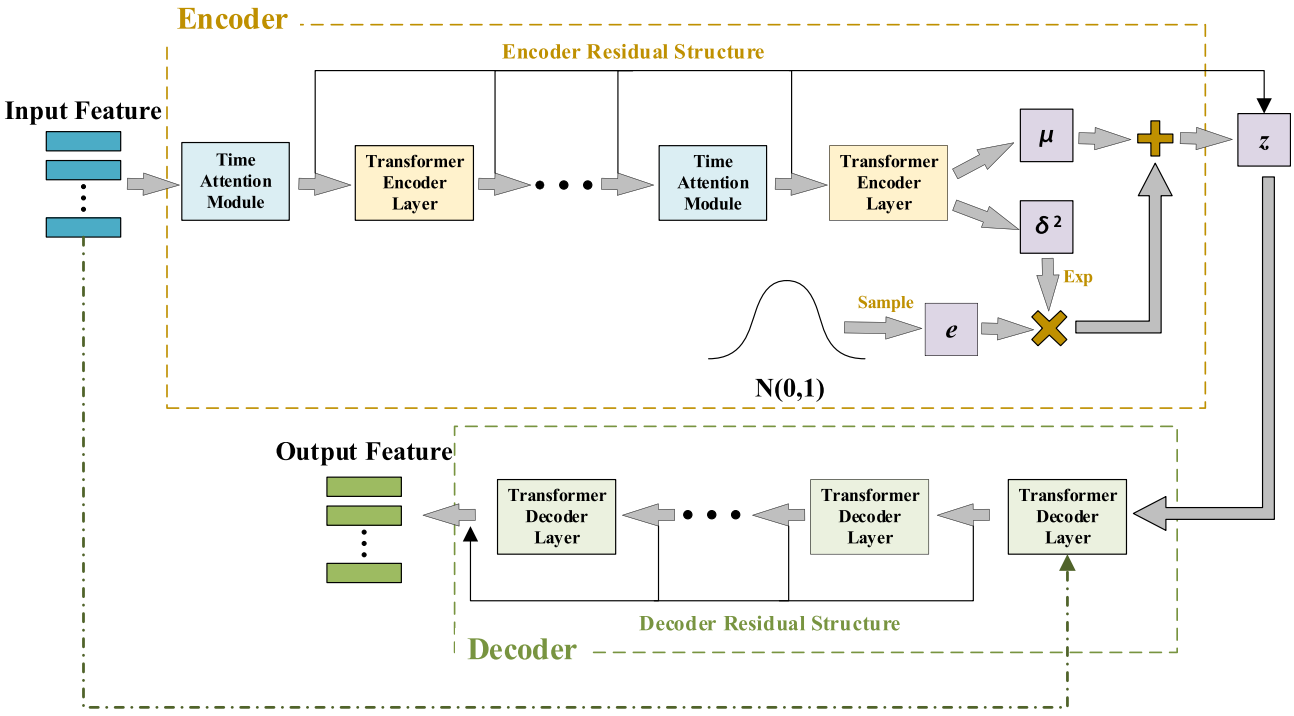


Fig. 7. Network structure diagram of the residual variational autoencoder.

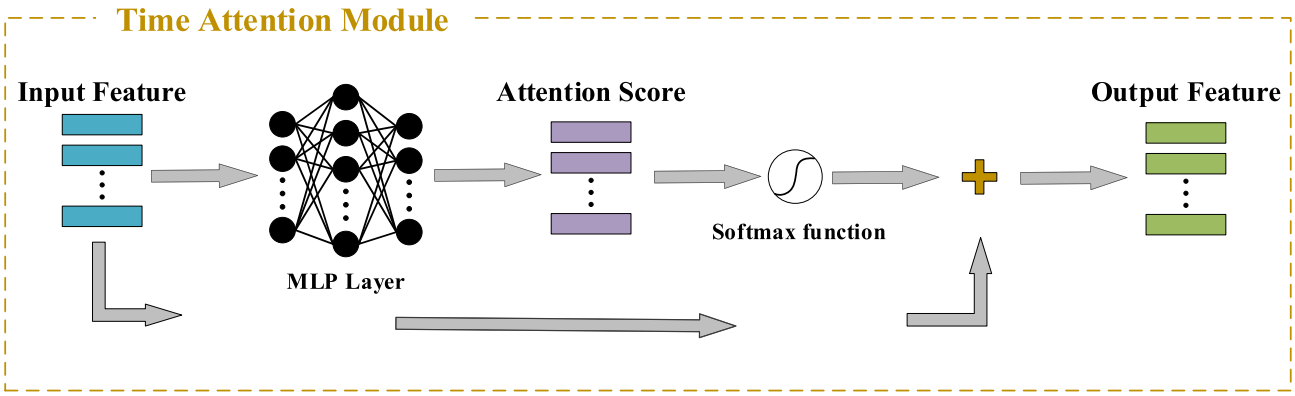


Fig. 8. Network structure diagram of the time attention module.

dimension attention data $\widehat{\lambda}^m$, as shown in Eq. (18).

$$\widehat{\lambda}^m = \mathrm{Softmax}(F_{mlp}(\lambda^m)) + \lambda^m \tag{18}$$

In addition, to further prevent the disappearance of divergence in the variational Transformer, we only add the time attention module to the encoder. Suppose the time attention module is added to the decoder. In that case, the decoding ability of the variational autoencoder will be too strong, and the risk of the KL divergence vanishing problem will increase.

3.4. Data reconstruction and anomaly detection

The data reconstruction module is composed of the one-dimensional convolutional layer and the fully connected layer, which is mainly used to adjust the output size of the model. In the training phase, we use unlabeled normal data to train the model. At this time, the model learns the characteristics of normal data and can reconstruct the input data according to the learned characteristics. We set an upper threshold limit based on the reconstruction loss value of the model under normal data. When the model performs anomaly detection, due to the significant difference between the characteristics of the fault data and the normal data, the reconstruction error at the fault location increases abnormally and exceeds the set upper threshold. Therefore, we judge the time point at which the reconstruction error exceeds the upper limit of the threshold as abnormal. However, due to the changeable operating environment of complex equipment and various working conditions, it is difficult to apply the same upper threshold for different types of faults. In response to this problem, we use the dynamic threshold method [28] to adaptively find the most suitable threshold range for the current data.

We record the reconstruction error of the detection data at each moment as the vector $R = (R_1, R_2, \cdots, R_h)$, where h is the total time of the detection data. In order to reduce the influence caused by the normal fluctuation of the data, we use the exponentially weighted moving average algorithm (EWMA) to smooth the reconstruction error, as shown in Eq. (19).

$$V_t = \eta V_{t-1} + (1-\eta) R_t \tag{19}$$

where V_t represents the smoothed reconstruction error at time t, V_{t-1}

represents the smoothed reconstruction error at the previous time, R_t represents the actual reconstruction error at the current time, and η represents the weight coefficient. The smaller the weight coefficient η, the smaller the influence of the old reconstruction error on the smoothing result at the current moment. For the smoothed error vector V, the upper and lower error thresholds at time t satisfy Eq. (20).

$$\begin{cases} MAX_t = \mu_i + Z^*\delta_i^2 \\ MIN_t = \mu_i - Z^*\delta_i^2 \\ i = [\frac{t}{n}] \end{cases} \tag{20}$$

In Eq. (20), Z is a hyper parameter manually set, n represents the size of the sliding window, and [] represents a round-down symbol. μ_i represents the mean value of the error vector V in the sliding window of group i, δ_i^2 represents the variance of the error vector V in the sliding window of group i, and both satisfy Eq. (21).

$$\begin{cases} \mu_i = \dfrac{\sum_{j=i^*n}^{(i+1)^*n} V_j}{n} \\ \delta_i^2 = \dfrac{\sum_{j=i^*n}^{(i+1)^*n} (V_j - \mu_i)^2}{n} \end{cases} \tag{21}$$

The multiscale Transformer-based residual variational autoencoder can learn deeper data features, and the dynamic threshold algorithm can help the model find the best error threshold. The combination of the two is the anomaly detection algorithm in this paper. The algorithm steps are as follows:

(1) Use the trained model MT-RVAE to reconstruct the detection data and calculate the reconstruction error of the data.
(2) Use the exponentially weighted moving average algorithm to smooth the reconstruction error vector.
(3) Calculate the error threshold at each moment through the dynamic threshold algorithm.
(4) Traverse the reconstruction error value at all times, judge whether there is an abnormal time point beyond the threshold range, and record it.

The specific process of the anomaly detection algorithm is shown in Algorithm 1.

Algorithm 1. *Anomaly detection algorithm based on MT-RVAE*

Input: Multivariate time series data set $X = [x_1, x_2, \cdots, x_m]$ sampled from mechanical equipment such as the satellite, data set size m, trained MT-RVAE model $F(\bullet)$
Output: Anomaly detection result RES
1: For $t = 1$ **to** m **Do:**
2: Calculate the reconstruction error of the data $R[t] = (X[t] - F(X[t]))^2$
3: End for
4: For $t = 1$ **to** m **Do:**
5: Use Eq. (19) to smooth the reconstruction error at time t to obtain $V[t]$
6: End for
7: For $t = 1$ **to** m **Do:**
8: Use Eq. (20) to calculate the sliding window i at time t
9: Use Eq. (21) to calculate the mean value $\mu[t]$ of the smoothing error V in the sliding window i
10: Use Eq. (21) to calculate the variance $\delta^2[t]$ of the smoothing error V in the sliding window i
11: Use Eq. (20) to calculate the upper limit of error $MAX[t]$ at time t
12: Use Eq. (20) to calculate the lower limit of error $MIN[t]$ at time t
13: End for
14: For $t = 1$ **to** m **Do:**
15: **IF** $V[t] > MAX[t]$ **OR** $V[t] < MIN[t]$ **THEN:**
16: $RES[t] = 1$
17: **ELSE:**
18: $RES[t] = 0$
19: **End IF**
20: End for
21: Return RES

Lines 1–3 of the algorithm are used to calculate the reconstruction error of the model on the test data. Lines 4–6 of the algorithm are used to smooth the reconstruction error by the exponentially weighted moving average algorithm. Lines 7–13 of the algorithm are used to calculate the error threshold at each moment through the dynamic threshold algorithm. Lines 14–20 of the algorithm are used to judge whether the reconstruction error at each time exceeds the threshold range and obtain the detection result.

4. Experiments

In this section, we firstly describe the experimental datasets and performance metrics. Then, the effectiveness of our method is proved compared with four traditional and six state-of-the-art anomaly detection methods. In addition, we also compare the time complexity of four different types of anomaly detection models and summarize how to choose methods in different detection scenarios. Finally, we demonstrate the necessity of each component of our model through ablation experiments.

4.1. Datasets and performance metrics

In order to prove the effectiveness of the method in this paper, we conducted experiments on two public datasets and a real dataset. The statistics of the data sets are shown in Table 1.

The public dataset SKAB (Skoltech Anomaly Benchmark) is a multivariate time-series dataset proposed by Iurii et al. [31] for evaluating time-series anomaly detection algorithms. The dataset contains 35 labeled data files, each of which represents a multivariate time series collected from sensors installed on the experimental platform. The industrial Internet of Things (IIOT) experimental system is located at the Skolkovo Institute of Science and Technology (Skoltech). As shown in Fig. 9, it consists of five parts: water circulation system, water-circulation control system, data Storage and visualization system, time-sensitive network system, and monitoring system. The multivariable time series collected by the system has nine dimensions, including time stamp information, amperage of the motor, the fluid temperature in the water circulation loop, and other indicators to monitor whether the system fails.

The real dataset SAT comes from the telemetry data of a satellite control system and power supply system. Due to the high sampling frequency and long sampling time, the telemetry dataset is often massive. In order to improve the detection efficiency, it is necessary to screen the parameters based on the experience of experts in related fields. Finally, nine parameters, including timestamps, are retained as indicators for evaluating the state of the satellite system. Besides, in order to analyze the satellite status only from the data-driven perspective, we changed the original feature name of telemetry data. We choose the telemetry data training and test model from November 2018 to March 2020, 80% of which is used as training data, and the remaining 20% is used as test data. The training data contains 400,000 records, all of which are under normal conditions. The test data contains 91,990 records, including 21,990 records at abnormal time points.

The public dataset NAB (The Numenta Anomaly Benchmark) is a

Table 1
Detailed statistics and experimental settings of the three datasets.

	Real Dataset SAT	Public Dataset SKAB	Public Dataset NAB-MT
Number of sequences	9	9	2
Training set size	400,000	9401	10,000
Testing set size	91,990	37,459	22,695
Anomaly Rate	4.47%	28.26%	9.98%

Fig. 9. Physical Picture of SKAB Industrial Internet of Things System [31].

univariate time-series dataset proposed by Lavin et al. [32] for evaluating algorithms for anomaly detection in streaming. The dataset consists of more than 50 labeled data files. These data files can be roughly divided into seven categories, including artificial and real-world time series. Each data file has only two columns as timestamp and data value, and the number of data points in the data file is between 1000 and 22,000. This paper selects the temperature sensor data of an internal component of a large industrial machine in the NAB dataset (NAB-MT) as the benchmark data for evaluating the performance of anomaly detection algorithms.

Examples of the anomaly detection results on the three data sets are shown in Fig. 10.

The anomaly detection algorithm divides the detected data into two categories, normal data and abnormal data. Therefore, the anomaly detection problem can be regarded as a binary classification problem, where the anomaly is true, and the normal is false. We use TP to indicate the number of true-positive samples, FP to indicate the number of false-positive samples, FN to indicate the number of false-negative samples, and TN to indicate the number of true-negative samples. Due to the long operating life and short failure time of complex mechanical equipment, the number of normal data is far greater than the abnormal data, which leads to an imbalance between the datasets. Therefore, we use five metrics: Precision, Recall, Missing Alarm Rate, False Alarm Rate, and F1 score to measure the performance of the anomaly detection algorithm for multivariate time series data. The calculation formula of each metric is shown in Eq. (22) and Eq. (23).

$$\begin{aligned} Prec &= \frac{TP}{TP+FP}, Rec = \frac{TP}{TP+FN} \\ F1 &= \frac{2\times Prec\times Rec}{Prec+Rec} = \frac{2\times TP}{2\times TP+FP+FN} \end{aligned} \tag{22}$$

In Eq. (22), *Prec* represents the precision rate, *Rec* represents the recall rate, and *F1* represents the F1 score. The larger the above three metrics, the higher the performance of the anomaly detection algorithm.

$$FAR = \frac{FP}{FP+TN}, MAR = \frac{FN}{FN+TP} \tag{23}$$

In Eq. (23), *FAR* represents the false alarm rate, and *MAR* represents the missing alarm rate. The smaller the above two metrics, the higher the performance of the anomaly detection algorithm.

4.2. Baseline methods and model settings

We compared four classical methods and six state-of-the-art methods to prove the superiority and effectiveness of the anomaly detection method proposed in this paper.

The four classical anomaly detection methods selected include local outlier factor (LOF) algorithm [33], isolation forest algorithm [34], anomaly detection algorithm based on LSTM autoencoder (LSTM-AE) [35], and fault prediction algorithm based on LSTM [36].

The selected six state-of-the-art methods of anomaly detection are as follows:

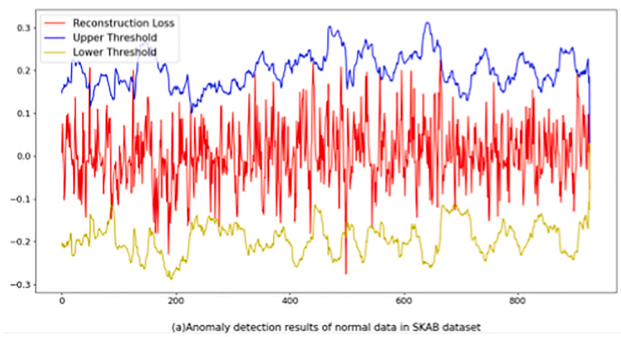
(1) GDN: GDN is a multivariate anomaly detection algorithm based on the graph deviation network proposed by Deng et al. [37]. The algorithm regards each time series as a node on the graph and adaptively learns the connection relationship between the nodes. The algorithm scores by detecting the deviation of the pattern in the node relationship graph and then detects abnormal time points.
(2) MTAD-GAT: MTAD-GAT is a multivariate anomaly detection algorithm based on the graph attention (GAT) network proposed by Zhao et al. [38]. The algorithm uses each time series as a feature and then uses two parallel GAT network layers to learn the complex dependencies of multivariate time-series in both temporal and feature dimensions. In addition, MTAD-GAT has also jointly optimized the prediction-based model and the reconstruction-based model to obtain a better time series representation.
(3) LSCP: LSCP is a novel integrated anomaly detection framework proposed by Zhao et al. [39]. This algorithm proposes a new parallel integration method, which defines the local area around the test instance by using the consistency of the nearest neighbors in randomly selected feature subspaces. The basic detector with the best performance in this local area is selected and combined as the final output of the model.
(4) OmniAnomaly: OmniAnomaly is a random recurrent neural network proposed by Su et al. [40]. This anomaly detection algorithm combines VAE and GRU (Gate Recurrent Unit) through random variable connection and plane normalized flow technology. This allows the model to learn robust latent representations while considering time dependence and the randomness of multivariate sequences.
(5) CPA-TCN: CPA-TCN is a unified model for detecting collective and point anomalies based on stacked temporal convolution networks (TCN) proposed by Li et al. [41]. This anomaly detection algorithm reconstructs sequential features with current inputs and historical features and is only trained on normal datasets. The algorithm's two-part anomaly detection module can significantly improve the accuracy of point anomaly detection.
(6) TCN-AE: TCN-AE is a temporal convolutional network autoencoder based on dilated convolutions [42]. This anomaly detection model consists of an encoder and a decoder, which are both trained simultaneously and learn to find a compressed representation of the input time series (encoder) and reconstruct the original input again (decoder). The model can also use so-called dilated convolutional layers to naturally create a large receptive field and process a time series signal at different time scales.

In order to ensure the fairness of the experiment and the validity of the experimental results, we debug the network parameters of the above method as much as possible to obtain the best results. For the model proposed in this paper, we give detailed parameter settings in Table 2.

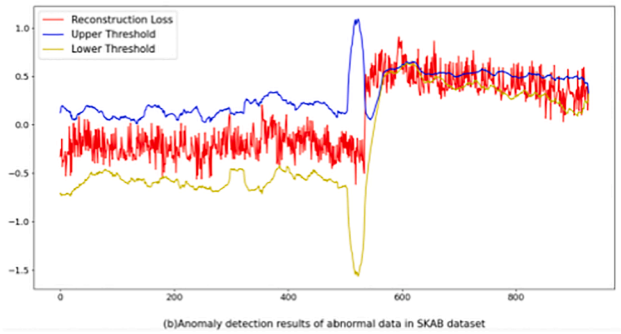
All the experimental processes aforementioned in this paper run in the same hardware configuration environment. The relevant hardware configuration is NVIDIA GTX 1660 Ti GPU, AMD R5 4600H CPU, and 16G memory.

4.3. Method comparison and result analysis

Since there exist complex temporal patterns in time-series data, accurate detection of anomalies in multivariate time-series data requires capturing temporal dependencies. Besides, the correlation between sequences makes multivariate time-series anomaly detection more

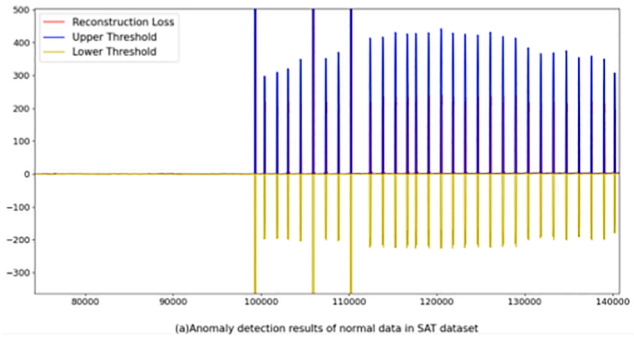


(a)Anomaly detection results of normal data in SKAB dataset

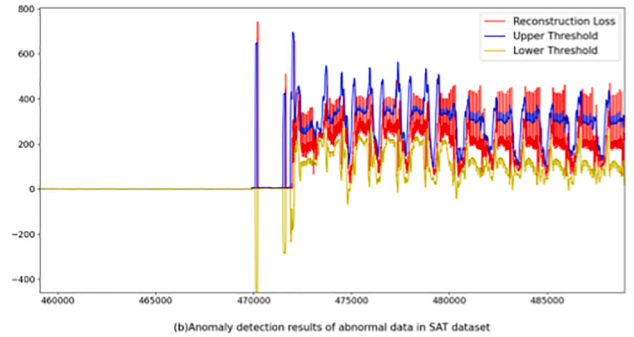


(b)Anomaly detection results of abnormal data in SKAB dataset

(1) Example image of anomaly detection results of the SKAB data set

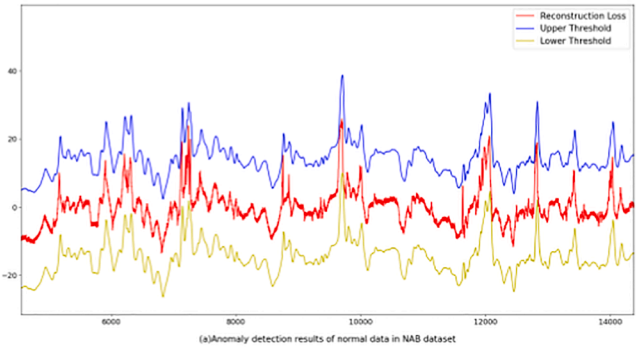


(a)Anomaly detection results of normal data in SAT dataset

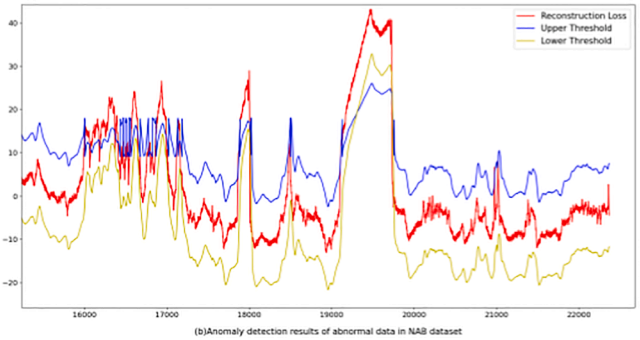


(b)Anomaly detection results of abnormal data in SAT dataset

(2) Example image of anomaly detection results of the SAT data set



(a)Anomaly detection results of normal data in NAB dataset



(b)Anomaly detection results of abnormal data in NAB dataset

(3) Example image of anomaly detection results of the NAB-MT data set

Fig. 10. Example images of anomaly detection results of three data sets.

complex on top of univariate time-series anomaly detection. Therefore, the ability to capture the sequence correlation in the data is another important factor to improve the performance of anomaly detection algorithms for multivariate time-series data.

To demonstrate that the MT-RVAE model proposed in this paper can effectively capture temporal dependence and sequence correlation in the data, we conducted comparative experiments using a univariate time-series dataset and two multivariate time-series datasets. Table 3, Table 4, and Table 5 show the five metrics of the MT-RVAE method and the comparison method. Fig. 11 shows the detrended cross-correlation analysis results of the two multivariate time-series datasets. Table 6 compares the time complexity of four different models.

From Table 3, we can get the following conclusions. Since the NAB-MT dataset has only two columns for timestamps and data values, there is no potential correlation between the data sequences. Therefore, whether the temporal dependence of data can be captured has a more significant impact on improving the detection performance of the model. So, machine learning algorithms that cannot capture temporal dependencies generally perform poorly in the NAB-MT dataset. In contrast, anomaly detection algorithms such as temporal convolutional

Table 2
Model parameter settings.

Parameters	Setting
Window size	64
Batch size	50
Dropout rate	0.25
Optimizer	AdamW
Learning rate	0.0002
Epochs	200
Transformer block number	3
Multi-head number	8
D_model	512

networks, recurrent neural networks, and MT-RVAE that can consider temporal dependencies in the data show a more significant advantage in the NAB-MT dataset. The performance of the MT-RVAE is similar to the performance of anomaly detection algorithms based on recurrent neural networks and temporal convolutional networks, demonstrating that the MT-RVAE model can effectively capture the temporal dependence in time series. However, the NAB-MT dataset has only one attribute in addition to the timestamp. Therefore, the performance of the graph neural network model and MT-RVAE, which focus on the multidimensional time-series data, is not as good as the TCN-AE method, which focuses on the one-dimensional time-series data.

The performance of the MT-RVAE model on the univariate time-series NAB-MT dataset demonstrates that the model can effectively capture the long-term dependence in time-series data. However, many real-world systems involve large numbers of interconnected sensors which generate substantial amounts of potentially correlated multivariate time-series data. Therefore, the ability to capture the sequence correlation in the data is another important factor to improve the performance of anomaly detection algorithms for multivariate time-series data.

From Fig. 11 and Table 4, we can get the following conclusions. The relationship between the different attributes of the public dataset SKAB is relatively sparse, resulting in less information in the feature

Table 3
Anomaly detection results on the univariate dataset NAB-MT.

Categories	Method	NAB-MT Dataset				
		F1	Prec(%)	Rec(%)	FAR(%)	MAR(%)
Machine Learning Model	LOF [33]	0.3623	31.83	42.05	9.98	57.95
	Isolation Forest [34]	0.4377	31.51	71.64	17.26	28.36
	LSCP [39]	0.3923	27.51	68.33	19.95	31.67
Temporal Convolutional Network Model	CPA-TCN [41]	0.5242	61.12	45.89	3.73	54.11
	TCN-AE [42]	**0.5999**	86.58	45.89	0.86	54.12
Recurrent Neural Network Model	LSTM [36]	0.4067	77.88	27.52	0.94	72.48
	LSTM-AE [35]	0.4259	69.50	30.70	1.63	69.30
	OmniAnomaly [40]	0.4388	31.01	**75.00**	18.58	**25.00**
Graph Neural Network Model	GDN [37]	0.3408	23.06	65.28	24.14	34.72
	MTAD-GAT [38]	0.3885	**87.08**	25.00	**0.41**	75.00
Our Model	**MT-RVAE**	0.5750	55.51	59.63	5.37	40.37

Table 4
Anomaly detection results on the multivariate dataset SKAB.

Categories	Method	SKAB Dataset				
		F1	Prec (%)	Rec (%)	FAR (%)	MAR (%)
Machine Learning Model	LOF [33]	0.6462	68.71	61.01	15.19	38.99
	Isolation Forest [34]	0.3000	42.33	23.23	17.31	76.77
	LSCP [39]	0.6570	66.69	64.73	17.68	35.27
Temporal Convolutional Network Model	CPA-TCN [41]	0.5475	48.78	62.38	37.50	37.62
	TCN-AE [42]	0.5759	53.41	62.48	31.28	37.52
Recurrent Neural Network Model	LSTM [36]	0.6366	68.25	59.65	15.28	40.35
	LSTM-AE [35]	0.6769	71.28	64.44	14.20	35.56
	OmniAnomaly[40]	0.7338	**91.13**	61.42	**3.28**	38.58
Graph Neural Network Model	GDN [37]	0.5234	40.92	72.62	57.35	27.38
	MTAD-GAT [38]	0.5840	50.20	69.80	38.01	30.20
Our Model	**MT-RVAE**	**0.7906**	78.94	**79.19**	13.35	**20.81**

Table 5
Anomaly detection results on the multivariate dataset SAT.

Categories	Method	SAT Dataset				
		F1	Prec (%)	Rec (%)	FAR (%)	MAR (%)
Machine Learning Model	LOF [33]	0.4053	25.68	96.13	13.02	3.87
	Isolation Fores t[34]	0.4643	31.20	90.73	9.36	9.27
	LSCP [39]	0.4379	28.49	94.51	11.10	5.49
Temporal Convolutional Network Model	CPA-TCN [41]	0.7712	66.91	91.00	1.91	9.00
	TCN-AE [42]	0.8184	75.74	89.00	1.21	11.00
Recurrent Neural Network Model	LSTM [36]	0.7696	66.67	91.00	1.94	9.00
	LSTM-AE [35]	0.8404	79.20	89.50	1.00	10.50
	OmniAnomaly [40]	0.9155	87.19	**96.36**	0.66	**3.64**
Graph Neural Network Model	GDN [37]	0.8842	93.62	83.78	0.27	16.22
	MTAD-GAT [38]	0.8468	76.36	95.04	1.38	4.96
Our Model	**MT-RVAE**	**0.9414**	**97.10**	91.36	**0.13**	8.64

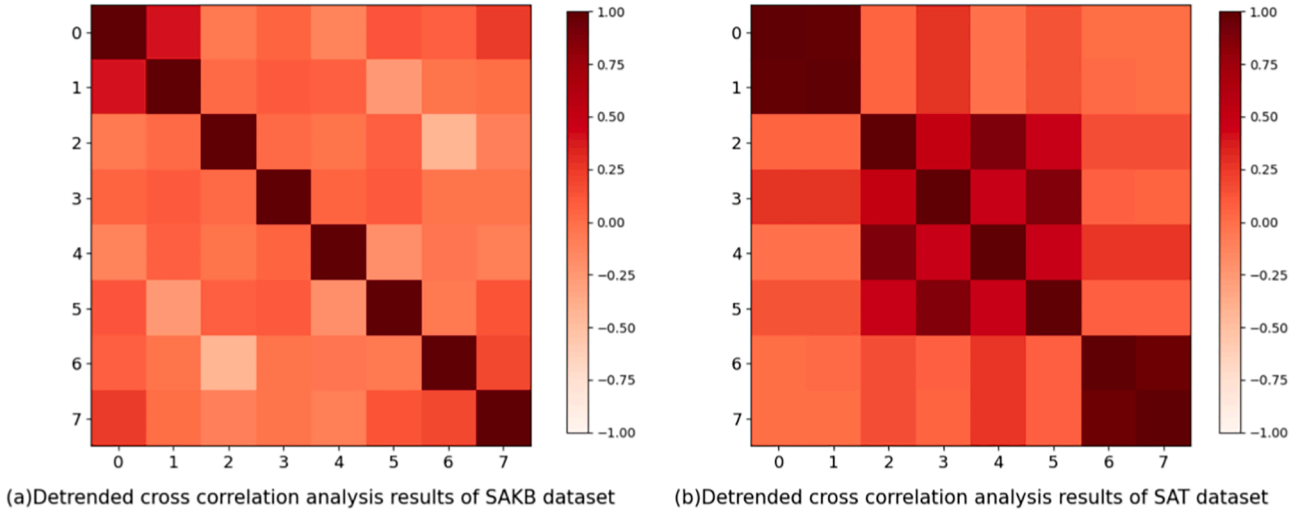


Fig. 11. Data sets de-trend cross-correlation analysis result graph.

Table 6
The time complexity of different models.

Layer Type	Complexity per Layer
Transformer	$O(L^2 \cdot m)$
Graph Neural Network	$O(L^2 \cdot m)$
Recurrent Neural Network	$O(L \cdot m^2)$
Temporal Convolutional Network	$O(k \cdot L \cdot m^2)$

relationship graph constructed by the graph neural network. Therefore, the performance indicators of anomaly detection algorithms based on the graph neural network are lower than the other types of anomaly detection algorithms except for the temporal convolutional network. Because the recurrent neural network can capture the temporal dependence in the data, the performance of the anomaly detection algorithm based on the recurrent neural network is better than the anomaly detection algorithm based on machine learning. However, due to the small size of SKAB dataset, the performance gap between the two anomaly detection algorithms is not significant. As for the temporal convolutional network, since the number of attributes in the SKAB dataset has increased compared to the previous one, the anomaly detection performance of the model, which relies only on the temporal dependence in the data, does not improve further. And the performance of multivariate time-series anomaly detection algorithms that can capture potential correlations between sequences, such as graph neural networks and MT-RVAE, is starting to come to the fore.

From Fig. 11 and Table 5, we can get the following conclusions. For the satellite telemetry dataset SAT with a large amount of data, the superiority of the neural network model has begun to highlight, and the performance of anomaly detection algorithms based on the neural network is generally higher than that based on machine learning. Due to the close relationship between different attributes in the satellite telemetry dataset, there is more information in the feature relationship graph than the feature relationship graph composed of the public dataset SKAB. Therefore, the performance metrics of the graph neural network model on the real dataset SAT are higher than that on the public dataset SKAB. And the performance metrics are also higher than the recurrent neural network model and the temporal convolutional network model because of the consideration of the data's sequence correlation and temporal dependence. However, since the SAT dataset has only nine dimensions, the upper limit of the information of the feature relationship graph is low, which causes a bottleneck in the graph neural network model's performance. Therefore, the performance metric of this type of algorithm is still lower than the OmniAnomaly algorithm that combines GRU and VAE. But, since the OmniAnomaly algorithm does not explicitly consider the correlation between sequences, the anomaly detection performance of this algorithm is consistently lower than that of the MT-RVAE model.

The MT-RVAE model proposed in this paper can adaptively capture the correlation between different sequences through Transformer's self-attention mechanism. Therefore, the algorithm does not need to extract information through the feature relationship graph and avoids the information bottleneck caused by node sparseness. This makes the model perform optimally even in the SKAB dataset, where the relationship between attributes is relatively sparse. In addition, the model also combines global temporal encoding and residual variational autoencoder, which can effectively extract the temporal dependence and local features of the data. Therefore, the MT-RVAE model proposed in this paper performs best in both multivariate time-series datasets.

In summary, from Table 3, Table 4 and Table 5, we can get the following conclusions.

(1) The MT-RVAE model proposed in this paper performs best in both multivariate time-series datasets, which proves this method's effectiveness and superiority in the multivariate time-series anomaly detection problem.
(2) The excellent performance of the anomaly detection algorithm based on the recurrent neural network in all data sets proves the importance of capturing the temporal dependence to improve the algorithm's performance.
(3) The significant performance difference of the temporal convolutional network-based anomaly detection algorithm on the univariate time-series dataset and the multivariate time-series dataset demonstrates that the model, which relies only on the temporal dependence in the data, will not be able to further improve the accuracy of the anomaly detection for multivariate time series.
(4) The performance metric of the anomaly detection algorithm based on the graph neural network in the SAT dataset is higher than the recurrent neural network model and the temporal convolutional network model, which proves the importance of capturing the sequence correlation to improve the performance of the algorithm.
(5) The significant performance difference of the anomaly detection algorithm based on the graph neural network in different datasets proves that the tightness of data feature relationships will affect the performance of such algorithms. And the performance

metrics of this type of algorithm in the two multivariate time-series datasets are always lower than the OmniAnomaly algorithm, which proves that the number of features of the data will cause the performance bottleneck of this type of algorithm.

(6) The performance metrics of the MT-RVAE model in the two multivariate time-series datasets are higher than the metrics of the recurrent neural network model, the temporal convolutional network model, and the graph neural network model, which proves that the model in this paper can effectively capture the temporal dependence and sequence correlation of the data. Besides, the number of data features and the closeness of feature relationships have little effect on the method in this paper.

The time complexity of the algorithm is an important indicator to measure the pros and cons of an algorithm. Many studies have used time complexity to compare the performance of algorithms [4344]. The time complexity of the four different models [184546] is shown in Table 6. Where L represents the length of the time series, m represents the dimensionality of the time series, and k represents the size of the convolution kernel.

From Table 6 and the above experimental results, we can get the following conclusions.

(1) Temporal convolutional networks and recurrent neural networks have lower time complexity. Therefore, for time-series data that are unidimensional or have no correlation between sequences, anomaly detection algorithms based on temporal convolutional networks or recurrent neural networks are more suitable.
(2) For time-series data that are multidimensional or have the correlation between sequences, although the time complexity of anomaly detection algorithms based on graph neural networks or Transformer is higher, these types of algorithms can obtain more accurate detection results. Therefore, they are more suitable.

4.4. Ablation experiment

In order to prove the necessity of each component in the model proposed in this paper, we remove the components one by one and observe how the performance of the model changes. We first remove the residual structure of the model and study the effect of the residual structure proposed in this paper on the performance of the variational autoencoder. Secondly, we replace the model's variational autoencoder structure with the autoencoder structure to study whether the regular hidden space can improve the anomaly detection performance of the model. Next, we delete the multi-scale feature fusion module to study whether the multi-scale feature information can improve the model's performance. Finally, to study the influence of the global temporal encoding module on the model's performance, we remove this module, only use the local temporal encoding to add sequential information to the data. Table 7 shows the performance metrics of the model after subtracting each module.

From Table 7 we can draw the following conclusions.

(1) As the components of the model are removed one by one, the performance metrics are gradually decreasing, which proves the necessity of each component.
(2) The removal of the residual structure leads to an increase in the false detection rate and a decrease in the accuracy rate of the model, which proves that the residual structure proposed in this paper can effectively alleviate the disappearance of the divergence of the variational autoencoder.
(3) The removal of the structure of the variational autoencoder causes the model's performance to decrease, which proves that the regular hidden space encoded by the variational autoencoder can improve the performance of the anomaly detection algorithm.

Table 7
Results of the ablation experiment.

Method	SKAB Dataset				
	F1	Prec (%)	Rec (%)	FAR (%)	MAR (%)
MT-RVAE	**0.7906**	**78.94**	79.19	**13.35**	20.81
- Residual structure	0.7708	73.68	80.82	18.42	19.18
- VAE	0.7621	71.90	**81.07**	20.22	**18.93**
- Multiscale feature fusion	0.7497	72.40	77.74	18.92	22.26
- Global temporal encoding	0.6426	55.23	76.81	39.74	23.19

(4) The removal of the multi-scale feature fusion module has led to an increase in the missed detection rate of the model, which proves that the multi-scale feature information can help the model better distinguish abnormal time points.
(5) The removal of the global temporal encoding module causes a severe decline in the model's performance, which proves that the attached time-series information of the module could help the model better capture the long-range dependencies of the data.

In order to further prove the effectiveness of the multi-scale feature fusion algorithm and residual variational autoencoder proposed in this paper, we use the public data set SKAB to compare various variants of the model.

In order to prove that the multi-scale feature fusion algorithm can improve the performance of the anomaly detection algorithm, we use the one-dimensional convolutional layer to replace the multi-scale feature fusion process to obtain the model CT-RVAE. In addition, in order to prove the superiority of the transposed convolutional network in the up-sampling process, we also use the bilinear interpolation algorithm to replace the transposed convolutional network for up-sampling and obtain the model LMT-RVAE. Table 8 shows the performance metrics of these three models.

From Table 8 we can draw the following conclusions.

(1) The LMT-RVAE model that uses the bilinear interpolation algorithm for up-sampling not only fails to extract the multi-scale feature information of the data but also increases the noise in the data, resulting in a severe decline in the performance of the model. It proves the superiority of using the transposed convolutional network for up-sampling.
(2) Although the CT-RVAE model without multi-scale feature information has achieved good performance, its performance indicators are lower than the MT-RVAE model that combines multi-scale feature information. It proves that the multi-scale feature fusion algorithm improves the performance of the model.

In order to prove that the residual structure proposed in this paper can alleviate the KL divergence vanishing problem, we use the ordinary residual structure to replace the residual structure proposed in this paper to obtain the model MT-NVAE. In addition, to prove that the regular hidden space encoded by the variational autoencoder can improve the

Table 8
Comparison of variant models of the multi-scale feature fusion algorithm.

Categories	Method	SKAB Dataset				
		F1	Prec (%)	Rec (%)	FAR (%)	MAR (%)
Our Model	**MT-RVAE**	**0.7906**	**78.94**	79.19	**13.35**	20.81
Variant model	CT-RVAE	0.7803	75.70	**80.52**	16.49	**19.48**
	LMT-RVAE	0.6173	50.11	80.34	51.04	19.66

Table 9
Comparison of variant models of the residual variational autoencoder.

Categories	Method	SKAB Dataset				
		F1	Prec (%)	Rec (%)	FAR (%)	MAR (%)
Our Model	**MT-RVAE**	**0.7906**	**78.94**	79.19	**13.35**	20.81
Variant model	MT-NVAE	0.6284	52.29	78.74	45.85	21.26
	MT-RAE	0.7763	74.04	**81.58**	18.26	**18.42**

model's performance, we use the residual autoencoder to replace the residual variational autoencoder to obtain the model MT-RAE. Table 9 shows the performance metrics of these three models.

From Table 9 we can draw the following conclusions.

(1) The MT-NVAE model that uses the ordinary residual structure not only does not alleviate the disappearance of the divergence of the variational autoencoder but also aggravates the process, resulting in a severe decline in the performance of the model. It proves that the residual structure proposed in this paper is effective in solving the problem of the KL divergence vanishing.
(2) The MT-RAE model using the residual autoencoder structure has lower performance metrics than the original model MT-RVAE, which proves the superiority of the variational autoencoder architecture in the anomaly detection problem.

5. Summary and future work

This paper proposes an unsupervised anomaly detection model based on the variational Transformer for multivariable time series data such as telemetry data. The model captures the correlation between time series through the self-attention mechanism, which reduces the impact of the number of data features and the closeness of feature relationships on the algorithm's performance. The global temporal encoding module of the model adds time-series and periodic information to the data to capture the long-term dependence. The multi-scale feature fusion module of the model adds feature information of multiple time scales to the data, enabling it to obtain a more robust feature expression. The residual variational autoencoder module of the model captures robust local features so that the hidden space encoded by it has regularity. The module's residual structure alleviates the vanishing of KL divergence and improves the lower limit of model generation ability. We compared the method in this paper with the existing methods on a real dataset and two public datasets and proved the method's effectiveness in this paper.

In the future, we will continue to conduct in-depth research on the following issues.

(1) In order to allow the model to capture better the autocorrelation of the data in the time dimension, we added the time attention mechanism before each Transformer coding layer. However, the structure of this attention mechanism is too simple and still has room for improvement.
(2) The dot product operation of the self-attention mechanism makes the model's time and space complexity high. The subsequent consideration is to improve the self-attention mechanism to reduce the model's time and space complexity.

CRediT authorship contribution statement

Xixuan Wang: Methodology, Software, Writing – original draft, Visualization. **Dechang Pi:** Conceptualization, W, riting – review & editing, Resources, Funding acquisition. **Xiangyan Zhang:** Investigation. **Hao Liu:** Formal analysis, Supervision. **Chang Guo:** Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Briskman, R. Akturan, Interference into radio broadcast satellite uplinks[J], Acta Astronaut. 166 (2020) 413–416, https://doi.org/10.1016/j.actaastro.2019.07.040.
[2] J. Chen, D. Pi, Z. Wu, X. Zhao, Y. Pan, Q. Zhang, Imbalanced satellite telemetry data anomaly detection model based on Bayesian LSTM[J], Acta Astronaut. 180 (2021) 232–242, https://doi.org/10.1016/j.actaastro.2020.12.012.
[3] B. Pilastre, L. Boussouf, S. D'Escrivan, J.-Y. Tourneret, Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning[J], Signal Process. 168 (2020) 107320, https://doi.org/10.1016/j.sigpro.2019.107320.
[4] S. Chen, G. Jin, X. Ma, Detection and analysis of real-time anomalies in large-scale complex system[J], Measurement 184 (2021) 109929, https://doi.org/10.1016/j.measurement.2021.109929.
[5] C. Zhang, D. Song, Y. Chen, et al, A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 1409-1416, https://doi.org/10.1609/aaai.v33i01.33011409.
[6] Z. Ghrib, R. Jaziri, R. Romdhane. Hybrid approach for anomaly detection in time series data[C]// Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020: 1-7, https://doi.org/10.1109/IJCNN48605.2020.9207013.
[7] C. Yin, S. Zhang, J. Wang, N.N. Xiong, Anomaly detection based on convolutional recurrent autoencoder for IoT time series[J], IEEE Trans. Syst. Man Cybernet.: Syst. 52 (1) (2022) 112–122, https://doi.org/10.1109/TSMC.2020.2968516.
[8] M. Yu, S. Sun, Policy-based reinforcement learning for time series anomaly detection[J], Eng. Appl. Artif. Intell. 95 (2020) 103919, https://doi.org/10.1016/j.engappai.2020.103919.
[9] Y.i. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, M.S. Hossain, Deep anomaly detection for time-series data in industrial iot: a communication-efficient on-device federated learning approach[J], IEEE Internet Things J. 8 (8) (2021) 6348–6358, https://doi.org/10.1109/JIOT.2020.3011726.
[10] D. Li, D. Chen, B. Jin, et al, MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks[C]//Proceedings of the International Conference on Artificial Neural Networks. Springer, Cham, 2019: 703-716, https://doi.org/10.1007/978-3-030-30490-4_56.
[11] L. Xie, D. Pi, X. Zhang, J. Chen, Y.i. Luo, W. Yu, Graph neural network approach for anomaly detection[J], Measurement 180 (2021) 109546, https://doi.org/10.1016/j.measurement.2021.109546.
[12] J. Ma, S. Perkins, Time-series novelty detection using one-class support vector machines[C]//Proceedings of the International Joint Conference on Neural Networks, 2003. IEEE, 2003, 3: 1741-1745, https://doi.org/10.1109/IJCNN.2003.1223670.
[13] H. Sarmadi, A. Karamodin, A novel anomaly detection method based on adaptive Mahalanobis-squared distance and one-class kNN rule for structural health monitoring under environmental effects[J], Mech. Syst. Sig. Process. 140 (2020) 106495, https://doi.org/10.1016/j.ymssp.2019.106495.
[14] Z. Li, Y. Zhao, N. Botta, et al, COPOD: copula-based outlier detection[C]// Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020: 1118-1123, https://doi.org/10.1109/ICDM50108.2020.00135.
[15] X. Zhou, Y. Hu, W. Liang, J. Ma, Q. Jin, Variational LSTM enhanced anomaly detection for industrial big data[J], IEEE Trans. Ind. Inf. 17 (5) (2021) 3469–3477, https://doi.org/10.1109/TII.2020.3022432.
[16] T. Kieu, B. Yang, C. Guo, et al, Outlier Detection for Time Series with Recurrent Autoencoder Ensembles[C]//Proceedings of the 28th International Joint Conference on Artificial Intelligence. IJCAI, 2019: 2725-2732, https://doi.org/10.24963/ijcai.2019/378.
[17] L. Ruiz, F. Gama, A. Ribeiro, Gated graph recurrent neural networks[J], IEEE Trans. Signal Process. 68 (2020) 6303–6318, https://doi.org/10.1109/TSP.2020.3033962.
[18] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need[C], in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.
[19] H. Zhou, S. Zhang, J. Peng, et al, Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(12): 11106-11115, https://ojs.aaai.org/index.php/AAAI/article/view/17325.
[20] H. Wu, J. Xu, J. Wang, et al, Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting[J]. arXiv preprint arXiv:2106.13008, 2021.
[21] D. P. Kingma, M. Welling, Auto-Encoding Variational Bayes[J]. stat, 2014, 1050: 1.
[22] S. Lin, R. Clark, R. Birke, et al, Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model[C]//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 4322-4326, https://doi.org/10.1109/ICASSP40776.2020.9053558.

[23] L. Li, J. Yan, H. Wang, Y. Jin, Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder[J], IEEE Trans. Neural Networks Learn. Syst. 32 (3) (2021) 1177–1191, https://doi.org/10.1109/TNNLS.2020.2980749.
[24] X. Liu, W. Teng, S. Wu, X. Wu, Y. Liu, Z. Ma, Sparse dictionary learning based adversarial variational auto-encoders for fault identification of wind turbines[J], Measurement 183 (2021) 109810, https://doi.org/10.1016/j.measurement.2021.109810.
[25] T.Y. Lin, P. Dollár, R. Girshick, et al., Feature pyramid networks for object detection[C], Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition IEEE (2017) 2117–2125, https://doi.org/10.1109/CVPR.2017.106.
[26] S. Qiao, L. C. Chen, A. Yuille, Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 10213-10224.
[27] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790, https://doi.org/10.1109/CVPR42600.2020.01079.
[28] K. Hundman, V. Constantinou, C. Laporte, et al., Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding[C], Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining KDD (2018) 387–395, https://doi.org/10.1145/3219819.3219845.
[29] H. Fu, C. Li, X. Liu, et al, Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019: 240-250, https://doi.org/10.18653/v1/N19-1021.
[30] T. Zhao, R. Zhao, M. Eskenazi, Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017: 654-664, https://doi.org/10.18653/v1/P17-1061.
[31] D.K. Iurii, O.K. Vyacheslav, Skoltech Anomaly Benchmark (SKAB). Kaggle, 2020, https://doi.org/10.34740/KAGGLE/DSV/1693952.
[32] A. Lavin, S. Ahmad. Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark[C]// Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015: 38-44, https://doi.org/10.1109/ICMLA.2015.141.
[33] M. M. Breunig, H. P. Kriegel, R. T. Ng, et al, LOF: identifying density-based local outliers[C]//Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000: 93-104, https://doi.org/10.1145/342009.335388.
[34] F.T. Liu, K.M. Ting, Z.H. Zhou, Isolation Forest[C], Proceedings of the Eighth IEEE International Conference on Data Mining. 2008 (2008) 413–422, https://doi.org/10.1109/ICDM.2008.17.
[35] H. Homayouni, S. Ghosh, I. Ray, et al, An autocorrelation-based LSTM-Autoencoder for anomaly detection on time-series data[C]//Proceedings of the 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020: 5068-5077, https://doi.org/10.1109/BigData50022.2020.9378192.
[36] S. Chauhan, L. Vig, Anomaly detection in ECG time signals via deep long short-term memory networks[C]//Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2015: 1-7, https://doi.org/10.1109/DSAA.2015.7344872.
[37] A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(5): 4027-4035, https://ojs.aaai.org/index.php/AAAI/article/view/16523.
[38] H. Zhao, Y. Wang, J. Duan, et al, Multivariate time-series anomaly detection via graph attention network[C]//Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020: 841-850, https://doi.org/10.1109/ICDM50108.2020.00093.
[39] Y. Zhao, Z. Nasrullah, M. K. Hryniewicki, et al, LSCP: Locally selective combination in parallel outlier ensembles[C]//Proceedings of the 2019 SIAM International Conference on Data Mining. SIAM, 2019: 585-593, https://doi.org/10.1137/1.9781611975673.66.
[40] Y. Su, Y. Zhao, C. Niu, et al, Robust anomaly detection for multivariate time series through stochastic recurrent neural network[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 2828-2837, https://doi.org/10.1145/3292500.3330672.
[41] Z. Li, Z. Xiang, W. Gong, et al., Unified model for collective and point anomaly detection using stacked temporal convolution networks[J], Applied Intelligence (2021) 1–14, https://doi.org/10.1007/s10489-021-02559-0.
[42] M. Thill, W. Konen, H. Wang, T. Bäck, Temporal convolutional autoencoder for unsupervised anomaly detection in time series[J], Appl. Soft Comput. 112 (2021) 107751, https://doi.org/10.1016/j.asoc.2021.107751.
[43] Z. He, P. Chen, X. Li, et al. A Spatiotemporal Deep Learning Approach for Unsupervised Anomaly Detection in Cloud Systems[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, https://doi.org/10.1109/TNNLS.2020.3027736.
[44] P. Gupta, A. Jindal, Jayadeva, D. Sengupta, Linear time identification of local and global outliers[J], Neurocomputing 429 (2021) 141–150, https://doi.org/10.1016/j.neucom.2020.11.059.
[45] P. Veličković, G. Cucurull, A. Casanova, et al. Graph Attention Networks[C]//Proceedings of the 2018 International Conference on Learning Representations (ICLR). 2018.
[46] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks[J], IEEE Trans. Neural Networks Learn. Syst. 32 (1) (2021) 4–24, https://doi.org/10.1109/TNNLS.2020.2978386.