Applied Intelligence (2024) 54:8968–8981
https://doi.org/10.1007/s10489-024-05341-0

# Unsupervised diffusion based anomaly detection for time series

Haiwei Zuo[1,2] · Aiqun Zhu[1,3] · Yanping Zhu[1] · Yinping Liao[1] · Shiman Li[1] · Yun Chen[1]

Accepted: 14 February 2024 / Published online: 8 July 2024


**Abstract**

Unsupervised anomaly detection aims to construct a model that effectively detects invisible anomalies by training and reconstruct normal data. While a significant amount of reconstruction-based methods has made effective progress for time series anomaly detection, challenges still exist in aspects such as temporal feature extraction and generalization ability. Firstly, temporal features of data are subject to local information interference in reconstruction methods, which limits the long-term signal reconstruction methods. Secondly, the training dataset collector is subject to information nourishment such as collection methods, collection periods and locations, and data patterns are diverse, requiring the model to rebuild normal data according to different patterns. These issues hinder the anomaly detection capability of reconstruction-based methods. We propose an unsupervised anomaly detection model based on a diffusion model, which learns normal data pattern learning through noisy forward diffusion and reverse noise regression. By using a cascaded structure and combining it with a structured state space layer, long-term time series signal feature can be well extracted. Different collection signals are distinguished by introducing collector entity ID embedding. The method proposed in this article significantly improves performance in experimental tests on three public datasets. Innovative aspects: (1) Utilizing the S4 method to capture long-term dependencies; (2) Employing a diffusion model for reconstruction learning; (3) Leveraging embedding techniques to enhance different pattern learning.

**Keywords** Diffusion · State space · Time series · Anomaly

## 1 Introduction

Anomaly detection is a data mining technique that identifies data points or patterns in a dataset that deviate from expected or normal behavior. In the real world, data-intensive applications often encounter the adverse effects of noise, outliers, and anomalies, which are common in large datasets. Anomalous patterns in time-series data often translate into meaningful issues, such as fraudulent activities in bank credit card operations [1]. In terms of network security, anomaly detection can identify abnormal behavior in the network, such as hacker attacks, malware infections. Anomaly detection can also help network administrators understand the normal data distribution or pattern, thus better identifying and preventing future threats. In the field of the Internet of Things, time-series anomaly detection can be used to monitor the operating status of devices, promptly discover abnormal behavior of the devices, and thus avoid device failures [2].

Anomaly detection in time series data faces many challenges. On one hand, time series abnormal detection is a contextual related task. Time series is a kind of data that contains seasonality and trends [3]. Abnormal data seems reasonable if viewed separately, But based on data over a period of time(context), it will show irregularity under trend and periodic characteristics. On the other hand, labeling anomalies in time series data is also a challenge. Many datasets currently lack annotations, and the definition of

Haiwei Zuo, Aiqun Zhu, Yinping Liao, Shiman Li and Yun Chen contributed equally to this work

✉ Yanping Zhu
zhuyanping@szsit.edu.cn

Haiwei Zuo
zuohaiwei@szsit.edu.cn

Aiqun Zhu
zhuaiqun@szsit.edu.cn

1 Department of Big Data Application Technology, School of Information and Communication Technology, City Polytechnic of Shenzhen (Shenzhen Institute of Technology), No.1 Jiangjunmao Road, Shenzhen 518000, Guangdong, China

2 Faculty of Engineering and Science, Curtin University Malaysia, CDT 250, Miri 98009, Sarawak, Malaysia

3 Research Administration Office, City Polytechnic of Shenzhen (Shenzhen Institute of Technology), No.1 Jiangjunmao Road, Shenzhen 518000, Guangdong, China

abnormal data and normal behavior changes over time, making the anomaly data itself unclear [4]. Public datasets used for time series anomaly detection often have problems, such as imbalanced data, containing only a small number of anomalies or error labels [5].

The above challenges exist in real-world application scenarios remaining as practical issues. Therefore, aiming at the characteristics that the abnormal time series data occupies a small portion and is hard to distinguish, we try to focus mainly on the dominant normal time series data. Motivated by general pattern from normal time series data, we proposed a reconstruction based methods to re-build time series curve. Learning data distribution from the major portion of normal data, we expect the net can ignore irregular pattern from anomaly. Denoising diffusion probabilistic models (DDPMs) [6] are a type of generative [7] model that have gained significant interests in recent years due to their ability to produce realistic and diverse samples in field of computer vision. The diffusion process allows the model to capture the underlying structure and complexity of the data distribution, while also allowing for a greater degree of randomness and variability in the generated samples. Further, with the guide of condition from normal data, the net can couple distribution from underlying main portion of data and reconstruct it.

We propose a diffusion based time series anomaly detection methods. The core idea is to learn the general pattern from normal data and reconstruct it. By detecting discrepancy between original and reconstructed curve, one-class classification is applied for anomaly detection. Different curves have different action patterns. We simply utilize Entity-ID, an extra input for embedding, to distinguish different time series pattern. Enhancing time-series feature extraction through structure state space sequence model(S4). Main contributions are as follows:

**a**. A time series anomaly detection algorithm based on diffusion models has been proposed, taking into account the temporal state features of time-series characteristics. By combining the S4 model and diffusion models, the features of normal data distribution are learned, and abnormal data are distinguished in an unsupervised manner through the reconstruction of sequences. By comparing 1D dilated convolution and S4, the results show that the later one performs better.

**b**. In order to enhance the processing of pattern features of different types of curves, the Entity Id feature of the curve collector is introduced, which distinguishes different types of curves and supplements them on the curve in the form of embeddings. By introducing this feature, the generalization ability of the common features of different curves is enhanced.

We have organized this paper according to the following structure. In the Introduction section, we will briefly introduce the background, motivation and main innovations of anomaly detection. In the Related Work section, we will present the related techniques and research progress. In the Methods section, we will focus on the main technical principles involved, including the Diffusion model, Structured State Space, and Framework of our anomaly detection model.In the Experiments section, we will present the training parameters, experimental results, and ablations. Finally, we will discuss and conclude the strengths and weaknesses of method proposed.

## 2 Related work

The breadth and depth of research on anomaly detection has provided numerous definitions for different types of anomalies. Our approach focuses primarily on time series anomaly detection. Currently, the main types of time series anomalies are point anomalies and contextual anomalies [8]. Point anomalies occur when a collected sample deviates from the expected range at a certain point randomly time and it does not match the pattern of data around. Contextual anomalies, on the other hand, refer to a period of time where all data deviates from the normal trajectory.

**Statistical method in anomaly detection** Statistical Methods are very common in anomaly detection, including mean, variance, Z-score, Grubbs' test, and IQR [5]. Clustering-Based Methods group similar data points together and identify anomalies as points that do not belong to any cluster [9]. Hierarchical clustering [5], K-NN based methods [10] and density-based [11] are applied in anomaly detection too. Statistical methods are often used in time series anomaly detection due to their simplicity and effectiveness. The main of advantage is that they rely on means, variance, distribution [12] to identify anomalies. It can be applied to time series data directly without training. And prior information plays a critical role in detection.

**Time series feature extraction and anomaly detection** RNN and LSTM networks have also shown strong capabilities in time series anomaly detection. In OmniAnomaly [13], the authors consider the time dependence and randomness of multivariate time series while learning potential feature representations, and identify anomalies through the Peaks-Over-Threshold method with automatically selected anomaly thresholds. In LSTM-AD [14], it predicts the trend of high-frequency signals to determine anomalous information. BeatGAN [15] regards time series as beats, and uses a discriminator to identify anomalous segments and mark anomalous time points.The space state machine is also applied in time series feature extraction and anomaly detection. Anomaly detection is achieved through unsupervised density generation methods, where the state space combines with generative models [16] or LSTM-based models [17].

**Self-supervised anomaly detection** There are one-classification anomaly detection method using self-supervised learning for unknown anomalous patterns [18]. It learns self-supervised deep representations and then builds a generative one-class classifier on learned representations. In [19, 20], authors consider One-Class Time-Series Classification(OCTSC) methods for Anomaly detection based on K-Means within sliding windows. In [19], it proposes distanced based method by calculating the feature vector of local cluster balance (LCB) and local prototype(LP) and distinguishing anomalies. In [20], clustering and linear regression operations was applied to find out anomalies. For nearest neighbor distribution represents, there is a new data descriptor ALP [21] that is proposed to address distances calculation.

**Diffusion-based anomaly detection** Due to the strong performance of image generation, diffusion based models have received considerable attention recently. In the field of medical imaging [22], the core idea is comparing reconstructed healthy data and real data for brain MRI and detecting anomalies. Diffusion models have a strong advantage in learning data distributions. They are widely used in time series forecasting, imputation [23, 24], and generation [25]. In the field of anomaly detection, the ImDiffusion [26] proposed a density ratio-based strategy to select normal observations flexibly that can easily adapt to the anomaly concentration scenarios. Unlike previous prediction or reconstruction-based methods, it built a new denoising diffusion-based imputation method to re-generate performance of missing values with proper condition. For unsupervised methods [27], it explored the use of unsupervised methods in anomalous diffusion data. The result showed that the main diffusion characteristics can be learned without any label.

Similarly, our proposed method utilizes a pipeline of reconstruction of normal data and a distance-based anomaly detection. We use structured state space time series for temporal features extraction and combine each S4 layer through a cascading structure, with forward information passed to the next layer and skip connections retaining global information. They work together for an end-to-end pattern restoring and an intuitive distance-based anomaly detection.

## 3 Theory

Given a vast time series with specific patterns that mainly consist of normal data segments, detecting anomalous time series segments effectively remains a challenge. Existing methods require an excessive amount of labeling and balanced data for classification or rely on less regularized reconstruction, resulting in lower accuracy for anomaly detection [28].

Instead of taking the current two-step [29] approach that first learns new representations and then applies some anomaly measures to the new representations to compute anomaly scores, we take learnable time series association as a structured state space and aim to learn the pattern of normal time series. The core idea of our method is to learn normal pattern of multi-channels time series under the forward and reverse diffusion process. We have approached an unsupervised method, where a large volume of data is utilized to learn the normal data patterns. Then a sampling process is applied to generate data assisted by entity id, a label-like input for embedding. By reconstructing both normal and anomalous data, the distance between normal data and reconstructed data is considerably less and distance between abnormal data and reconstructed data will show another pattern. A threshold for one-classification is then calculated based on statistical methods, which enabling detection of anomalies.

### 3.1 Diffusion

Diffusion model is a generative model that has been gaining popularity in the field of machine learning and artificial intelligence. It is used to generate high-quality sequential images [30], wave audios [31], and other types of data [32] by modeling the data as a diffusion process. Diffusion models acquire knowledge by removing noise in a reverse process, which is added sequentially in a Markovian fashion during a forward process, to create a mapping from a latent space to the original signal space.

In the diffusion forward process, noise is gradually added to the data in a controlled manner. Each forward process is parameterized as in (1):

$$q(x_1, x_2, \ldots, x_T|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$
$$= \prod_{t=1}^{T} N(x_t|\sqrt{1-\beta_t}x_{t-1}, \beta_t I)[x_t]. \tag{1}$$

Where $q(x_t|x_{t-1})$ represents forward-process. We add noise to $x_{t-1}$ and build $x_t$ under the constant noise schedule weight $\beta_t$ which controls the noise level in step $t$. For fast calculation, there is a closed form as $x_t = \sqrt{\alpha_t} \cdot x_0 + \sqrt{1-\alpha_t} \cdot \epsilon$, and $\alpha_t = \sum_{i=1}^{t}(1-\beta_t)$, noise $\epsilon \sim N(0, 1)$.

In the diffusion reverse process, the noise is removed step by step to recover the original data. Each reverse process is parameterized as in (2):

$$p(x_0, x_1, \ldots, x_{T-1}|x_T) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$
$$= p(x_T) \prod_{t=1}^{T} N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2}$$

Where $p_\theta(x_{t-1}|x_t)$ represents reverse-process with learnable parameter $\theta$. According to the theory of the diffusion model, any distribution of data assumes as a Gaussian distribution by adding noise in a forward diffusion process for T rounds, thus $x_T \sim N(0, 1)$.

By the definition showed in (2), the predicted noise should be close to target noise in a way from forward-process where $p_\theta(x_{t-1}|x_t) \sim N(0, 1)$. So we get optimization goal that:

$$L = \min \|\epsilon - \epsilon_\theta(x_t, t)\|^2 = \left\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2. \tag{3}$$

Where $\epsilon_\theta(x_t, t)$ is parameterized using a diffusion network and $\sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\epsilon$ is the way that signal mix noising. $\alpha_t = 1 - \beta_t$ and $\bar{\alpha_t} = \prod_{s=1}^{t} \alpha_s$. And $\beta_t$ represents the noise schedule pre-calculated before training. And Algorithm 1 below shows the details of training process

**Algorithm 1** Training process.

**Require:** $x_0 \sim q(x_0)$
**loop**
  let $\epsilon \sim N(0, 1)$,
  let $t \sim Uniform(\{1, 2, \ldots, T\})$,
  let $x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$,
  calculating gradient: $\nabla\theta \ \|\epsilon - \epsilon_\theta(\epsilon - \epsilon_\theta(x_t, t))\|^2$.
**end loop**

Pertaining to the inference process, the ultimate objective of the network is to generate reconstructed data, restoring major part of data distribution and precluding the possibility of directly outputting noise data. In inference stage, the sampling procedure is crucial for inference. By sampling normal data distribution under condition, signal restoration is effectuated. And Algorithm 2 below shows the details of sampling process.

**Algorithm 2** Sampling process.

**Require:** $x_T \sim N(0, 1)$
**loop** $t = T, T-1, \ldots, 1$
  **if** $t > 1$ **then**
    let $z \sim N(0, 1)$,
  **else if** t == 0 **then**
    let $z = 0$,
  **end if**
  let $\bar{x}_{t-1} = x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t, t)$,
  let $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \bar{x}_{t-1} + \sigma_t z$.
**end loop**

### 3.2 Structured state space for time series

State space models [33] are mathematical models used to describe the behavior of complex systems over time. These models consist of a set of variables that represent the system's state, and a set of recursive steps that describe how the state variables change over time. The structured state-space model (SSM) [34] represents a stable time series modeling paradigm which is able to capture feature among time, especially for long-term time series [24].

According to the SSM theory, the formalism employs a linear state space transition equation, which links a one-dimensional input sequence $\mu(t)$ to a one-dimensional output sequence $y(t)$ through an N-dimensional hidden state $x(t)$. Specifically, this transition equation of fast discrete representations as follows:

$$x'(t) = A \cdot x(t) + B \cdot \mu(t), \quad and \quad y(t) = C \cdot x(t) + D \cdot \mu(t). \tag{4}$$

Where A, B, C, D are transition matrices that map an input signal $\mu(t)$ to output $y(t)$. And this process is SSM block which is to utilize the SSM solely as a black-box representation within a deep sequence model [34].

For purpose of using GPU acceleration, the authors proposed a Structured State Space Sequence model(S4) [35] by stacking multiple copies of the SSM blocks with suitable normalization layers and point-wise fully-connected layers in the style of a transformer layer. And considering the performance of S4, we integrate it with diffusion models for feature extraction in time series. For comparative studies on the performance of S4 can be found in Section 4.4.

### 3.3 Framework for anomaly detection

For a time series anomaly detection problem, let $X = \{x_1, x_2, \ldots, x_n | x_i \in \mathbb{R}^{L \times K}\}$ denote a collection of K channels and L samples time series. let $Y\{y_1, y_2, \ldots, y_n | y_i \in \{0, 1\}^{L \times 1}\}$ denote a collection of labels where 0 represents normal data and 1 represents abnormal data and k is the size of Y which is much smaller than n. A entity ID, a label-like input for embedding is introduced. The ID of curves are represented by $E = (E_1, E_2, \ldots, E_N)$. Each element of $x_i$ is a segment of time series. The target of anomaly detection is to classify whether part of $x_i$ is abnormal without the prior of normal data or abnormal data. Inspired by both GAN-Based method and cascade structure from diffwave [31] that reconstruct normal data from a real sample, we proposed a method that using diffusion process to sample normal pattern and distance was used to distinguish anomaly data which is calculated between real data and reconstruction result in a way like one-classification and ignoring Y-Label.

The diffusion model demonstrates a very powerful ability in generative tasks. Considering time series reconstruction, we take cascade structure and propose a framework structure for time series reconstruction net.

In Fig. 1, the Left side shows the core layer's structure. This layer of the proposed approach is used to analyze signals and extract time-series temporal features. The forward part utilizes bidirectional dilated Conv1D to extract local features of the time-series signal. The framework adopted here is similar to DiffWave [31]. The condition works like a supplementary information that contains original time series curve, controls signals and entity id. Long-term features are extracted by S4 block and it has been experimentally evaluated. The overall framework adopts a cascading structure, such that the output part, i.e. layer out, is passed into the next layer. The skip connection of each S4Layer is preserved. The Right side shows the entire structure of the proposed approach. The entire structure's input consists of three parts, namely noise $x_t$, forward diffusion step t, and conditions. The skip connection from each layer will be add together and pass through several Conv1D, a simulation of ELBO-style remainder term. Besides, adding additional paths in the network, skip connections allow gradients to flow directly to earlier layers and enable accelerating training process [36].

After sample process discussed in Algorithm 2, the smoothed distance can be calculated as follow:

$$distance(x_i) = \int_{i-w}^{i+w} c(i-t) \cdot \|x_t - p(q(x_t))\|_{L_1} \, dt. \quad (5)$$

where in equation (5), $q(x_i)$ is forward diffusion process and $p(q(x_i))$ represents sampling process and $c(t)$ represents discrete convolutional kernel. The goal of (5) is to find a smooth distance between origin and reconstructed time series for one classification. After calculating distance, we statistics method is applied for one-classification.

## 4 Experiment

In the experimental section, we will introduce the data form, training parameters, performance evaluation, and ablations.

### 4.1 Dataset

Consider using SMAP, MSL and SMD as the datasets for our experimental purposes and ignore missing or incomplete data. Table 1 shows the statistics of these datasets [37]. The SMAP represents Soil Moisture Active Passive satellite and MSL represents Mars Science Laboratory rover. SMD is a Server Machine Dataset from a large Internet company and this dataset contains 3 groups of entities.

SMAP and MSL contain telemetry values and command information as showed in Fig. 2(Left). Telemetry values are numeric data while command information is boolean data. During a specific time interval, each telemetry sample is collected under the control of a set of specific command from different time points. Some methods utilize historical telemetry data and employ command control information as an auxiliary condition to predict the telemetry information for the next time step. Our method focused on reconstruction from noised telemetry data with condition of control commands. It is clear that the pattern of telemetry data is associated with sensors and environments. In in

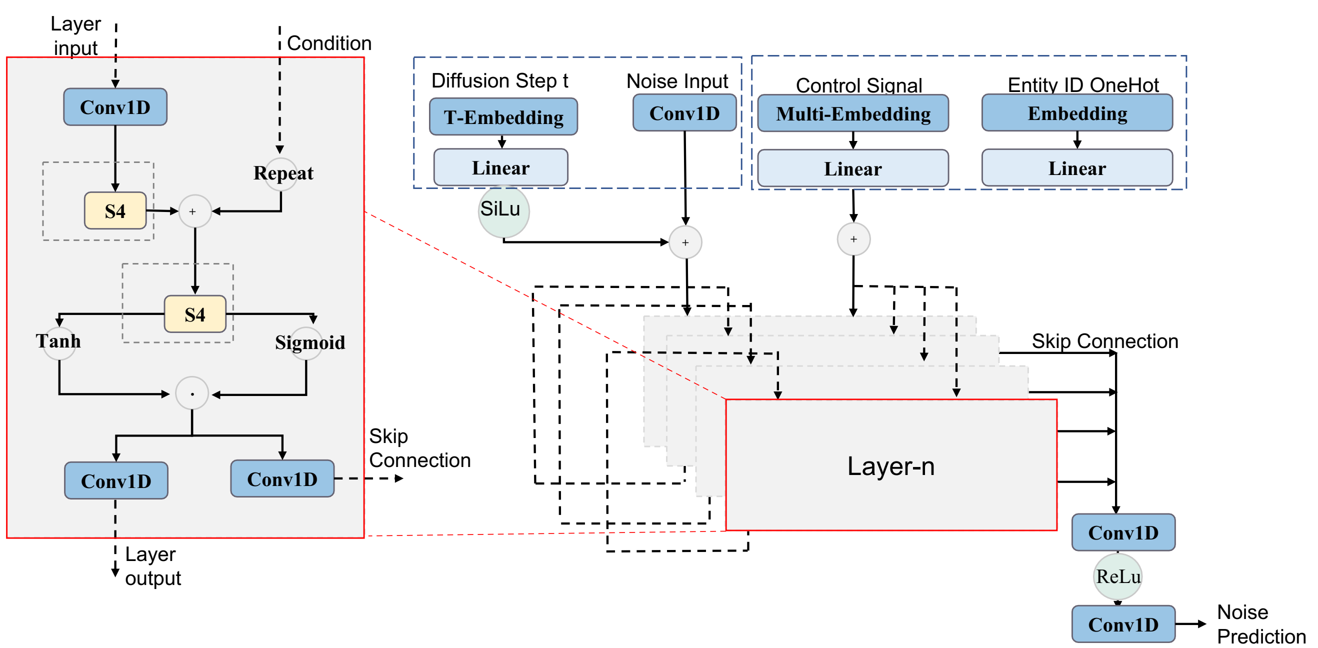


**Fig. 1** Proposed diffusion structure for time series anomaly detection (left: layer structure; right : framework for anomaly detection)

**Table 1** Statistics of dataset

| Data | N entity | Channels | Size | Anomaly class | Anomaly ratio (%) |
|---|---|---|---|---|---|
| SMAP[1] | 55 | 25 | 562,800 | 1 | 13.13 |
| MSL[1] | 27 | 55 | 132,046 | 1 | 10.72 |
| SMD[2] | 28 | 38 | 1,416,825 | 1 | 4.16 |

Data Source: All data is experimented under anonymous processing
[1]SMAP and MSL: Soil Moisture Active Passive satellite and Mars Science Laboratory rover
[2]SMD [13] : Server Machine Dataset

Fig. 2(Middle), normal data exhibits different patterns, some of which are gradient-like signals while others are serrated signals. Anomalous telemetry data also comes in different types.in Fig. 2(Right), one type of anomalies is scattered around the regular data in the form of discrete points. Another type of abnormal data requires the combination of contextual information for identification.

The time series can be described as follow that it contains n points with time point T. Let $X = x_1, x_2, .., x_n$ represent signals from telemetry values in a format of time series and $x_i \in \mathbb{R}^{L \times K_1}$ where $L$ is the length of signals and $K_1$ is the number of numeric channels. In SMAP and MSL, $K_1$ equals 1. Let $C = c_1, c_2, .., c_n$ is a segment of time series and $c_i \in \{0, 1\}^{L \times K2}$ where $c_i$ is a command control signal. The $L$ is max length in this sample and $K_2$ is the number of boolean command channels. For example, the P-1 marks curve a control information in $c_i$ is 1 in P-1 and 0 in other places. The target of anomaly detection is to learn normal data pattern and reconstruct data between normal data and anomalies. Let $E = (E_1, E_2, \ldots, E_N)$, named collectors' entity ID, like middle part in Fig. 2. It is an auxiliary label for pattern classification like zigzag or serrate periodic curve..

All of the data has been anonymized and numeric data have been pre-scaled between $(-1, 1)$ by min-max normalization from values obtained in each channel. All of the data removes any identifying information related to sensor or other commands. The data of signal is divided into short frames at which window length equals $L$ and hop length equals $\alpha \cdot L$. The i-th block locates $(t_{str}, t_{end})$ around $t_i$ where $t_i = i \cdot \alpha \cdot L$, $t_{str} = t_i - (1 - \alpha) \cdot L$, $t_{end} = t_i + \alpha \cdot L$. $\alpha \in (0, 1)$is ratio in which each segment of time series will overlaps partly with adjacent blocks. In Right fig, it shows point anomaly and contextual anomaly in orange-yellow. The anomaly in former one is clearly a point anomaly that few points deviated from average points. The anomaly in later picture is some kind a contextual anomaly with pattern out of expectation.

## 4.2 Training and anomaly detection

The training process is as same as normal diffusion model. A step is selected randomly under total diffusion step length. The signal is mixed with noise to generate input data as showed in Section 3.1. The second part is the condition, and we take the control information and entity ID as the conditions. The control information is transformed into a continuous dense space through an encoding process and entity embedding is added. Then the noise is regressed through reverse diffusion process with input data and conditions under certain step. And the loss function is MSELoss.

The setting of hyperparameters comes from grid search. And some major parameters can be set based on experience. Usually, the net will run under a group of hyperparameters. Then according to the results, we can make adjustments

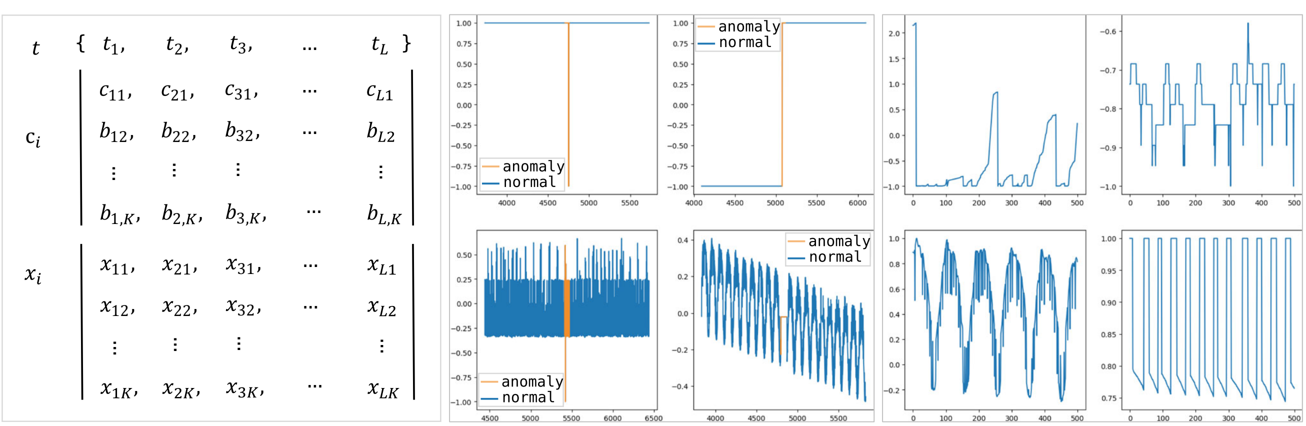


**Fig. 2** Data format (left); data pattern (middle); anomalous pattern (right)

correspondingly. Specific training parameter is showed in Table 2.

In terms of the model architecture, the entire structure adopts a cascade structure. The cascade length determines the degree of information extraction. The data is processed through 1D dilated convolution to extract temporal features, and then passed to S4Layer for state space feature learning. The skip connection not only avoids gradient disappearance but also fuses information at specific output temporal points by combining low-level and high-level features [36]. This enables the network to retain more temporal information and refine the reconstruction effect of time series data. The data time frame length is fixed to 2000, which is relatively long and may lead to some performance loss. However, due to the small and continuous appearance of abnormal data, it is not possible to use a shorter length of time frame. According to the distribution of the data, 2000 is a reasonable size. Besides, a warmup strategy is used for training acceleration.

After the sampling process of the diffusion model, anomaly detection works by comparing the convolutional distance between reconstruction data and the original signals. Once the sampling process is completed, the convolutional distance between the reconstructed data and the signal data is calculated in time-point wise. Simply, the mean $\mu_d$ and variance $\sigma_d^2$ of convolution distance in (5) is calculated. Assume that noise follows a normal distribution and distance between reconstructed data and normal data is follow such a distribution. With three-sigma rule, $P(\mu - 3\sigma, \mu + 3\sigma)$ will cover normal 99.73% of normal data and can used as a threshold.

### 4.3 Evaluation

**Visualization** Anomaly detection employs noise-added signals as input, with control information as conditions, progressively denoising and reconstructing samples. The detection of anomalies is achieved through the discrepancies between reconstructed outcomes and original signals. The results are showed in Figs. 3 and 4. In the illustration, the blue curve represents the test set sample data, which, when noise is added, serves as input for the network. The orange curve depicts the result of reconstruction after sampling. According to the curve's trajectory and patterns, it is evident that the reconstructed signal can restore the anticipated data pattern. The red curve delineates the convolution distance between the test set sample data and the reconstructed data. It is discernible that in the anomalous data segment, the convolution distance exhibits substantial energy. Figures 3 and 4 correspond to the reconstruction results without and with collectors' entity ID embedding, respectively. The disparity in their performance is quite conspicuous. In terms of data reconstruction, one exhibits superior stability with entity id embedding. Regarding continuity, both tend to generate noticeable oscillations.

Abnormal data is small-scale data with a low proportion in the dataset discussed in Table 1. In statistics, when dealing with imbalanced data, major metrics such as recall, precision, and F1 are taking into consideration. Besides, we introduced a frame-level coverage rate that treat time series a data frame by frame. And IOU is applied to calculate the real anomaly frames and predicted anomaly frames for contextual anomaly. In Table 3, it shows the comparison of mean precision, recall, F1 and accuracy of our method and other baselines. Out methods achieve slightly outperforms than other methods.

**Quantization** The metrics indicate that the reconstructed results essentially conform to the normal data pattern. The precision rate and recall rate demonstrate the ability to reconstruct normal data in Table 3. The precision rate represents the effectiveness of time-series data reconstruction based on the diffusion scheme. In the experiment of SMAP, precision rate represents the probability of correctly predicting actual anomalies among all data predicted as anomalies. In other words, it measures the confidence in predicting correctly among the anomalies. The experimental result reached about 91%, indicating that the convolution error with the normal data is within a certain range. Due to the small portion of anomalies, the recall rate test shows that the probability of detecting actual targets in all anomalies is around 86%. For the indicators of accuracy is not a useful guideline. By comparing Diffusion with Entity Embedding (DiffAD-E) and Diffusion without Entity Embedding (DiffAD), it shows that different pattern can be recognized and generalized when net know what type of sensor it is.

Furthermore, by analyzing the average L1 Loss in the reconstructed signals, which is approximately 0.03, it demonstrates that the signals reconstruction has been achieved for normal data. Employing convolution averaging has mitigated vibration of generated data, thereby effectuating a smoothing action. For point-based anomaly data, the likelihood of identifying surrounding anomalies increases, thereby

**Table 2** Diffusion model parameters

| Hyper parameter | Value |
|---|---|
| Layer num[1] | 18 |
| Feature dim[1] | 512 |
| Diffusion noise step length | 200 |
| Time series max length | 2000 |
| Batch size | 32 |
| Optimizer | AdamW |
| Warm-up[2] epochs | 200 |
| Total epochs | 800 |
| Learning rate | 1e-3 |

Net structure was showed in Fig. 1
[1]Feature dim of channel and embedding keep same
[2]Warm-up strategy is going to discuss in Section 4.4

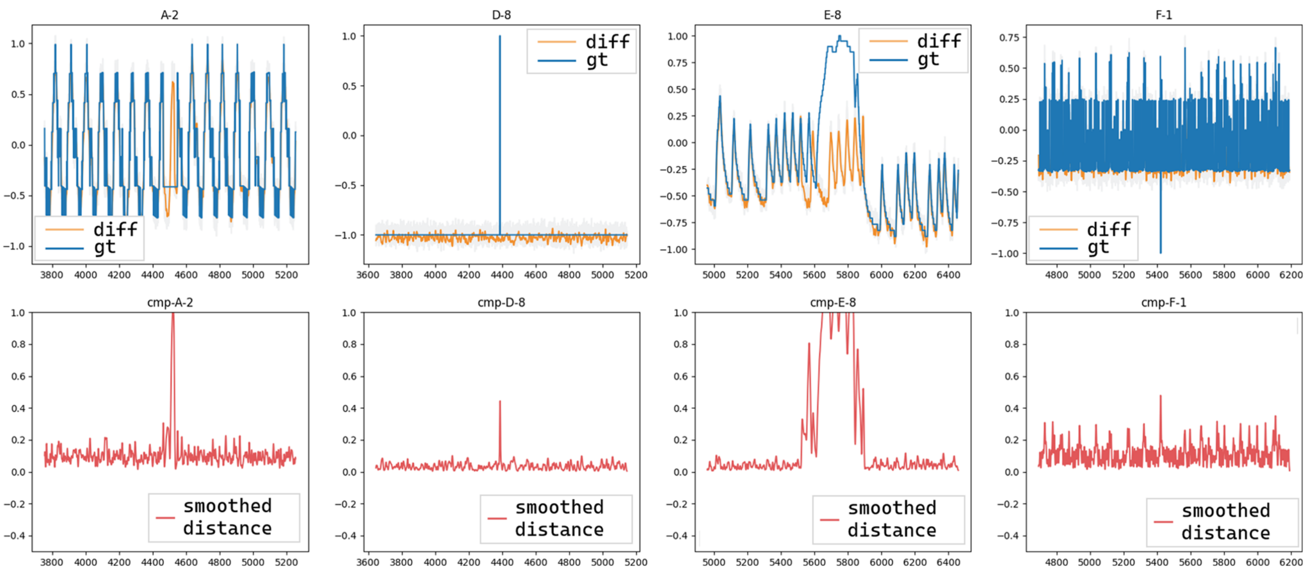


**Fig. 3** Without entity-ID reconstruction

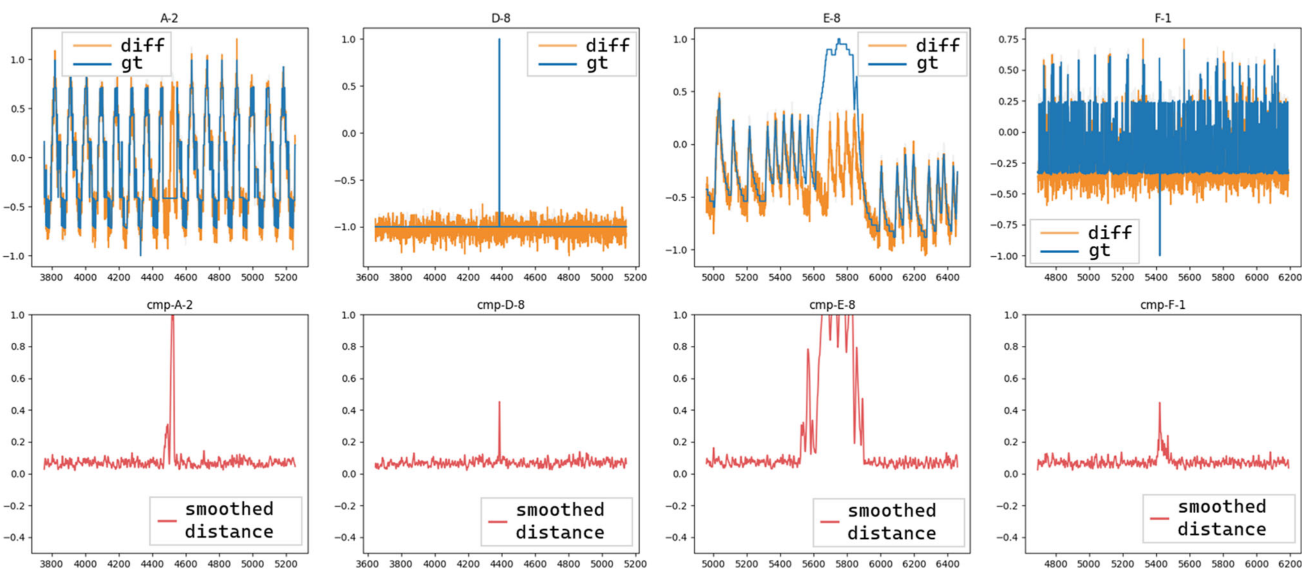


**Fig. 4** With entity-ID reconstruction

**Table 3** Metrics of different methods

| Methods | SMAP dataset | | | MSL dataset | | | SMD dataset | | |
|---|---|---|---|---|---|---|---|---|---|
| | mPre | mRec | mF1 | mPre | mRec | mF1 | mPre | mRec | mF1 |
| Kernel PCA [38] | 0.8747 | 0.6075 | 0.7170 | 0.7779 | 0.6437 | 0.7045 | 0.9231 | 0.5497 | 0.6890 |
| OC-SVM [39] | 0.6916 | 0.6202 | 0.6540 | 0.8030 | 0.7085 | 0.7528 | 0.8940 | 0.6050 | 0.7217 |
| OC-ALP [21] | 0.9002 | 0.7899 | 0.8044 | 0.8413 | 0.8550 | 0.7998 | 0.9506 | 0.8291 | 0.9010 |
| OC-LCB [19] | 0.8440 | 0.8669 | 0.7940 | 0.8883 | 0.8550 | 0.8033 | 0.9101 | 0.8308 | 0.8505 |
| LASTM-AE [40] | 0.8550 | 0.8550 | 0.7100 | 0.9260 | 0.6940 | 0.6900 | 0.8039 | 0.7714 | 0.7878 |
| BeatGan [15] | 0.9494 | 0.8060 | 0.8718 | 0.9323 | 0.8108 | 0.8672 | 0.7680 | 0.8236 | 0.7948 |
| DiffADT[1] | 0.9538 | 0.8260 | 0.8853 | 0.9454 | 0.8656 | 0.9037 | 0.7874 | 0.8611 | 0.8226 |
| DiffADT(embeding)[2] | 0.9633 | 0.8660 | 0.9120 | 0.9521 | 0.8727 | 0.9107 | 0.7987 | 0.8627 | 0.8295 |

Note: All the result are calculated under same data group. mPre, mRec and mF1 corresponding to mean precision, recall and F1 metrics. [1]DiffADT represents net that do not contain entity-id embedding. [2]DiffADT(Embeding) represents net with entity-id embedding expanding the actual range of anomaly detection. In the case of contextual anomaly data, approximately 90% of the intersection-over-union (IOU) ratio can be covered showed in Table 4.

In contextual anomaly data, a further analysis of the coverage of contextual anomalies in the reconstructed sampling is conducted by employ the IOU metric. This study utilizes convolution to average the distance information between real signals data and reconstructed data, thereby differentiating anomalies within continuous segments. Regarding the contextual reconstruction disparity, the distinctions among several filtering kernel algorithms are minimal, indicating that the reconstruction methodology essentially restores the normal data pattern and effectively distinguishes anomalous data.

### 4.4 Ablations

We conduct ablation studies to demonstrate the effectiveness of our proposed methods and some hyper-parameters studies to seek the best model configurations.

**Warm-up strategy** A warm up strategy is used that is a technique gradually increasing the learning rate during the initial training phase of a model. This allows the model to warm up and start learning slowly before the learning rate is increased to a higher level. With large batch size, the goal of a warm-up strategy is to help the model learning normal pattern and converge faster. In experience, we set a gradual warm up learning rate that reaching the highest point of the learning rate at 1/4 of the total epochs, and then it will gradually decrease.

The Warm-up acceleration strategy is employed to enhance the training speed. The warm-up mechanism allows the learning rate to first rise, then gradually decline. During the actual training process showed in Fig. 5(Left), before the epoch is less than 400, the fixed learning rate training process exhibits a relatively steady decline in loss. Due to the accelerated training process facilitated by preheating, the learning rate gradually increases, and the loss reduction rate progressively escalates, reaching the 1e-2 level earlier. When the epoch exceeds 400, the learning rate of the preheated acceleration begins to decrease, resulting in a smoother, slower decline in training loss. Conversely, the fixed learning rate training process is accompanied by a conspicuous oscillation in the loss reduction process. As the loss reaches the 1e-3 level, the jitter issue becomes increasingly apparent. Using the first loss accuracy of 1e-2 as the evaluation criterion, the warm-up mechanism can reduce training time by 60%, considerably augmenting training efficiency.

At the initial state, Conv1D starts from zero weights. At the beginning of training, Diffusion has no understanding of the data distribution and considers all data equally important. As the diffusion process progresses, the large loss at the initial stage can decrease quickly, and the model will promptly correct the data distribution and learn the main distribution characteristics of the data. The model has already gained prior information about the data, and with the increase of learning rate, the model will be more prone to learning the distribution characteristics of regular data, which is equivalent to preheating the model with data.

The learning rate curve under the Warm-up mechanism is a good way for acceleration, yet it also presents certain issues. Firstly, the inflection point in the learning rate increase process requires empirical estimation, typically situated at one-quarter of the complete training position. If the inflection point is reached too early, the subsequent low learning rate will result in a reduced loss decline speed, thereby prolonging the training duration. Conversely, if the inflection point is reached too late, the training may encounter instances where an excessively high learning rate leads to local solutions. Designing the learning rate peak at the inflection point location necessitates specific skills; during training, setting the peak too high can cause initial instability, as demonstrated in the graph where several instances of loss oscillation occur between epochs 200 and 400.

**Layers setting** The layer configuration is determined through experimentation. As diffusion involves regressing noise from the data distribution, the noise must reach a specific value range before it can be deemed that the training has achieved the desired objectives. During testing, once the loss falls below 1e-3, the reconstructed curve of the sampled data essentially stabilizes. We have examined the effects of cascading S4 layers at 6, 12, 18, and 24 layers, maintaining consistent model parameters and training parameters, with the exception of layer discrepancies.

In the Table 5, the second column represents the iteration count when the initial noise regression loss falls below 1e-3. Owing to the presence of fluctuations, the loss must consistently reach 1e-3 multiple times to be considered stable. The third column pertains to the reconstruction loss of

**Table 4** IOU for different smoothed distance

| Convolution kernel | Convolution size | IOU metric in average |
|---|---|---|
| Hanning | 15/25/35 | 89.7%, 92.4%, 90.4% |
| Hamming | 15/25/35 | 89.1%, 92.3%, 90.3% |
| Barlett | 15/25/35 | 89.9%, 91.8%, 90.4% |

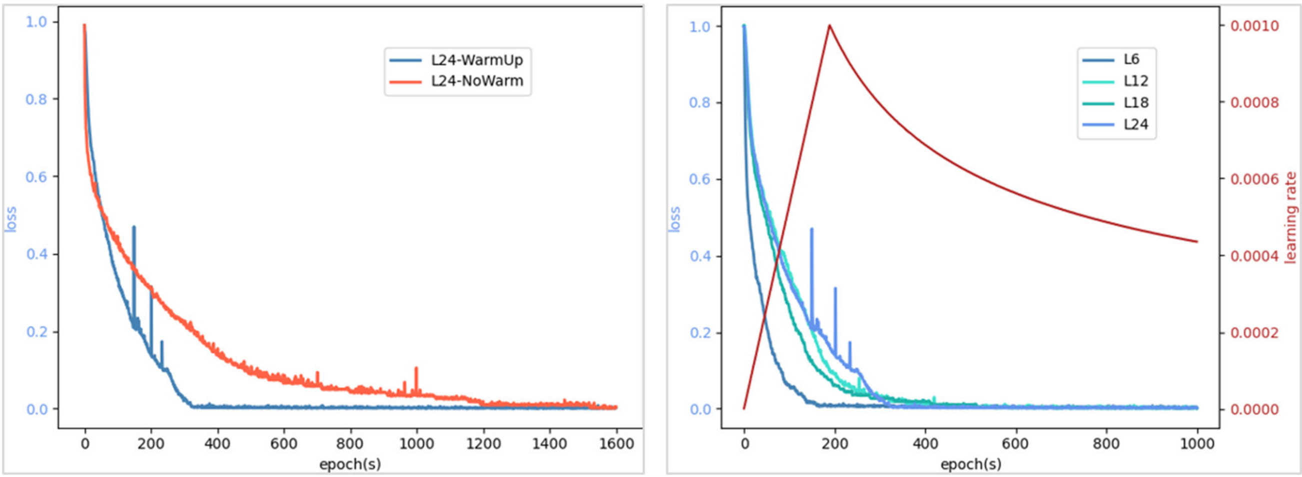


**Fig. 5** Loss curve with or without warmup (left); loss curve with different layers (right)

normal data, with the loss function being L1, the sampling count being Diffusion Step=20, the maximum noise schedule being 0.02, and the minimum noise schedule being 1e-4. The fourth column denotes the time consumption, where the testing environment is Nvidia Tesla P40, the sampling steps are consistent, and the average time taken for multiple data samples' reconstruction is calculated. Ultimately, the loss can reach below 1e-3. Among these, L6 exhibits the fastest convergence speed under noise regression, while L18 has the slowest. In terms of normal data loss during data reconstruction, L24 achieves the smallest loss, with the most significant variation observed in L6. As for time consumption, the fewer the layers, the shorter the duration.

In the reconstruction of time-series curves, the L1 Loss of S4 L24 showed the best performance. In ablation study, the results between using 1d dilated convolution [41] and using the S4 layer were compared. We compared items such as L1 Loss, L2 Loss and Time Consume. With all other hyperparameters being the same, we compared the fluctuation of loss in the last 500 epochs. In Table 6, under the same parameters, S4 performed better than the Res layer. Under same layer type, it can be seen that as the number of layer increases, the stability of loss gradually increases.

**Performance comparing to entity embedding** The variety of data patterns is substantial. As discernible from Fig. 2, different data collectors and varying conditions yield diverse data patterns. Utilizing control information solely as the denoising condition is, in actuality, insufficient. The entity object serves as a style embedding, concatenated with the control condition to form a novel condition. Testing demonstrates that the model's noise regression loss exhibits minimal discrepancies. In data sampling, incorporating entity-style samples decouples entity information from data patterns, facilitating improved learning of the relationship between control information and data distribution. Under L24, the reconstructed data model presents relatively smooth reconstructed data. In performance comparison, the anomaly detection efficacy has been enhanced, as detailed in the Table 3.

**Indicator of smoothed distance** In paper [37], the author distinguishes anomalies into point anomalies and contextual anomalies. It is undeniable that discrepancies exist between the reconstruction results and the authentic data. Diffusion is incapable of achieving a wholly congruent fit for normal data, particularly in data with high-frequency fluctuations, where some reconstruction loss persistently occurs between the reconstructed normal data and the genuine data. By smoothing the reconstruction discrepancies, the quality of individual data reconstruction is mitigated, optimizing the differences between the reconstructed and normal data.

Our method of anomaly detection relies on distance between real data and reconstructed data. Essentially, it

**Table 5** L1 loss and time consume by different layers (SMAP)

| Layer number | Epoch that first reach(1e-3) | L1 loss for sampling | Time consume(second) |
|---|---|---|---|
| S4-L6 | 310 | 0.0310 | 0.2520 |
| S4-L12 | 520 | 0.0242 | 0.3440 |
| S4-L18 | 590 | 0.0154 | 0.5088 |
| S4-L24 | 480 | 0.0122 | 0.7121 |

**Table 6** Result from S4 layer and res layer (SMAP in sampling)

| Layer number | L1 loss | L2 loss (last 500 epochs) | Time consume (second) |
|---|---|---|---|
| S4-L6 | 0.0310 | 0.0069 ± .0030 | 0.2520 |
| S4-L12 | 0.0242 | 0.0057 ± .0025 | 0.3440 |
| S4-L18 | 0.0155 | 0.0033 ± .0012 | 0.5088 |
| S4-L24 | 0.0123 | 0.0040 ± .0005 | 0.7121 |
| RES-L6 | 0.0608 | 0.0131 ± .0040 | 0.1021 |
| RES-L12 | 0.0457 | 0.0084 ± .0024 | 0.1439 |
| RES-L18 | 0.0355 | 0.0045 ± .0010 | 0.3055 |
| RES-L24 | 0.0310 | 0.0032 ± .0009 | 0.4101 |

remains a one-class classification problem. We introduce the ROC(Receiver Operating Characteristic) curve for performance evaluation. The distance between real data and reconstructed data is transformed into prediction scores range from 0 to 1 by maximum distance. The process of ROC calculation relies on sklearn.metrics [42]. Through the ROC curve Fig. 6, we can clearly see the significant distinction between the reconstructed data and the real data. With an AUC (Area Under the Curve) of 92.14%, we think indicator of smooth distance is reasonable.

## 5 Discussion

It is feasible for anomaly detection in time series by reconstruction from diffusion model. But it has several limitations. First, the anomaly should be limited in low level to make sure that it does not impact regular data. Second, the entity id is designed based on the dataset of the curve, it does not have universality, and it is an ID added in advance manually. Thirdly, the net is still weak in handling real-world problem when there may be a large amount of abnormal data or noisy data collected due to device failures. Performance can still be affected by the magnitude and frequency of changes in data distribution. Finally, there is a vital problem. If the data distribution changes significantly, it may result in the net not being applicable with the new distribution. The net will need to be fine-tuned or re-trained with the new data.

Additionally, the method proposed in this paper still has some directions for improvement. Combining Fourier Transform and Trend could be utilized to enhance feature extraction in time series. Replacing Entity ID, the type of curve features can be automatically obtained through a small part of curve by using some prompt idea. As for the discrepancies between the reconstructed data and the real data, optimizing anomalies detection through dynamic thresholds is also a possible direction for improvement. The design of the threshold lacks consideration of the trend of the time series itself. In fact, the threshold should vary with trends and seasonality.

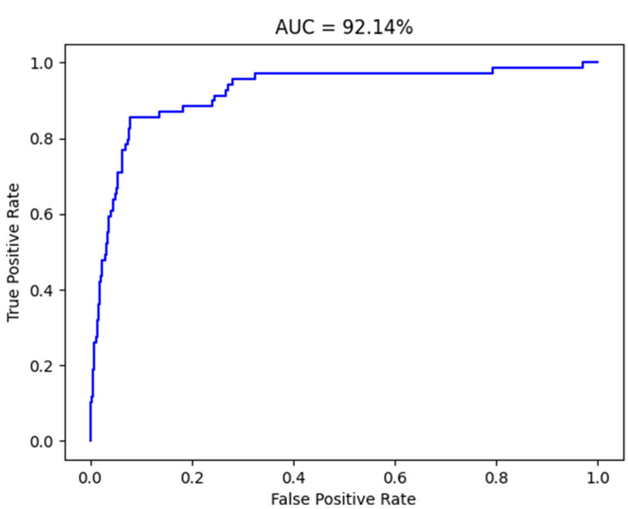


**Fig. 6** ROC of smoothed distance based anomaly detection

## 6 Conclusion

In this paper, we propose an unsupervised anomaly detection network based on Diffusion and one-class classification. After reconstructing multi-channel time series signals, a smooth distance is applied to detect anomalies. In order to enhance the model's generalization ability to time series anomalies, we combine diffusion with S4 layer to achieve rich feature representation of both global and local timing characteristics. Experiments on 3 public datasets demonstrate that our methods show better result on metric like accuracy, precision.

However, the method we propose is currently not a universal method. We need to learn the distribution of major data and combine it with a pre-labeled Entity ID to enhance the reconstruction goal. Additionally, we plan to develop more efficient training algorithms and conduct theoretical analysis of our method.

## References


1. Pimentel MA, Clifton DA, Clifton L, Tarassenko L (2014) A review of novelty detection. Signal Process 99:215–249
2. Schmidl S, Wenig P, Papenbrock T (2022) Anomaly detection in time series: a comprehensive evaluation. Proceedings of the VLDB Endowment 15(9):1779–1797
3. Shaukat K, Alam TM, Luo S, Shabbir S, Hameed IA, Li J, Abbas SK, Javed U (2021) A review of time-series anomaly detection techniques: a step to future perspectives. In: Advances in information and communication: proceedings of the 2021 future of information and communication conference (FICC), Springer, vol 1, pp 865–877 (2021)
4. Blázquez-García A, Conde A, Mori U, Lozano JA (2021) A review on outlier/anomaly detection in time series data. ACM Comput. Surveys (CSUR) 54(3):1–33
5. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41:15
6. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Adv Neural Inform Process Syst 33:6840–6851
7. Li Y, Lu X, Wang Y, Dou D (2022) Generative time series forecasting with diffusion, denoise, and disentanglement. Adv Neural Inform Process Syst 35:23009–23022
8. Hayes MA, Capretz MA (2015) Contextual anomaly detection framework for big sensor data. J Big Data 2(1):1–22
9. Kumari R, Singh M, Jha R, Singh N et al (2016) Anomaly detection in network traffic using k-mean clustering. In: 2016 3rd International conference on recent advances in information technology (RAIT), IEEE, pp 387–393
10. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data, pp 427–438
11. Breunig MM, Kriegel H-P, Ng RT, Sander J (2000) Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data, pp 93–104 (2000)
12. Perer A, Sun J (2012) Matrixflow: temporal network visual analytics to track symptom evolution during disease progression. In: AMIA annual symposium proceedings, American Medical Informatics Association, vol 2012, p 716
13. Su Y, Zhao Y, Niu C, Liu R, Sun W, Pei D (2019) Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining, pp 2828–2837
14. Malhotra P, Vig L, Shroff G, Agarwal P et al (2015) Long short term memory networks for anomaly detection in time series. In: ESANN, vol 2015, p 89
15. Zhou B, Liu S, Hooi B, Cheng X, Ye J (2019) Beatgan: anomalous rhythm detection using adversarially generated time series. IJCAI 2019:4433–4439
16. Li L, Yan J, Wen Q, Jin Y, Yang X (2022) Learning robust deep state space for unsupervised anomaly detection in contaminated time-series. IEEE Trans Knowl Data Eng
17. Wang F, Wang K, Yao B (2023) Time series anomaly detection with reconstruction-based state-space models. In: International conference on artificial neural networks, Springer, pp 74–86
18. Li C-L, Sohn K, Yoon J, Pfister T (2021) Cutpaste: self-supervised learning for anomaly detection and localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9664–9674
19. Hayashi T, Cimr D, Studnička F, Fujita H, Bušovskỳ D, Cimler R, Selamat A (2024) Distance-based one-class time-series classification approach using local cluster balance. Expert Syst Appl 235:121201
20. Hayashi T, Cimr D, Studnička F, Fujita H, Bušovskỳ D, Cimler R (2024) Patient deterioration detection using one-class classification via cluster period estimation subtask. Inform Sci 657:119975
21. Lenz OU, Peralta D, Cornelis C (2021) Average localised proximity: a new data descriptor with good default one-class classification performance. Pattern Recognit 118:107991
22. Pinaya WH, Graham MS, Gray R, Da Costa PF, Tudosiu P-D, Wright P, Mah YH, MacKinnon AD, Teo JT, Jager R et al (2022) Fast unsupervised brain anomaly detection and segmentation with diffusion models. In: Part VIII (ed) Medical image computing and computer assisted intervention-MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings. Springer, pp 705–714
23. Tashiro Y, Song J, Song Y, Ermon S (2021) Csdi: conditional score-based diffusion models for probabilistic time series imputation. Adv Neural Inform Process Syst 34:24804–24816
24. Alcaraz JML, Strodthoff N (2022) Diffusion-based time series imputation and forecasting with structured state space models. arXiv:2208.09399
25. Harvey W, Naderiparizi S, Masrani V, Weilbach C, Wood F (2022) Flexible diffusion modeling of long videos. Adv Neural Inform Process Syst 35:27953–27965
26. Xiao C, Gou Z, Tai W, Zhang K, Zhou F (2023) Imputation-based time-series anomaly detection with conditional weight-incremental diffusion models. In: Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining, pp 2742–2751
27. Muñoz-Gil G, Corominas GG, Lewenstein M (2021) Unsupervised learning of anomalous diffusion data: an anomaly detection approach. J Phys A: Mathematical Theoretical 54(50):504001
28. Sperl RE, Chung SM (2019) Two-step anomaly detection for time series data. In: 2019 International conference on data and software engineering (ICoDSE), IEEE, pp 1–5
29. Li J, Di S, Shen Y, Chen L (2021) Fluxev: a fast and effective unsupervised framework for time-series anomaly detection. In: Proceedings of the 14th ACM international conference on web search and data mining, pp 824–832
30. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2022) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10684–10695
31. Kong Z, Ping W, Huang J, Zhao K, Catanzaro B (2020) Diffwave: a versatile diffusion model for audio synthesis. In: International conference on learning representations
32. Hoogeboom E, Nielsen D, Jaini P, Forré P, Welling M (2021) Argmax flows and multinomial diffusion: learning categorical distributions. Adv Neural Inform Process Syst 34:12454–12465
33. Kitagawa G (1998) A self-organizing state-space model. J Am Stat Assoc 1203–1215
34. Krishnan R, Shalit U, Sontag D (2017) Structured inference networks for nonlinear state space models. In: Proceedings of the AAAI conference on artificial intelligence, vol 31
35. Smith JT, Warrington A, Linderman S (2022) Simplified state space layers for sequence modeling. In: The eleventh international conference on learning representations
36. Oyedotun OK, Al Ismaeil K, Aouada D (2021) Training very deep neural networks: rethinking the role of skip connections. Neurocomputing 441:105–117
37. Hundman K, Constantinou V, Laporte C, Colwell I, Soderstrom T (2018) Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp 387–395
38. Simoliuniene R, Krisciukaitis A, Macas A, Baksyte G, Saferis V, Zaliunas R (2008) Principal component analysis based method for

detection and evaluation of ecg t-wave alternans. In: 2008 Computers in cardiology, IEEE, pp 757–760
39. Zhang R, Zhang S, Muthuraman S, Jiang J (2007) One class support vector machine for anomaly detection in the communication network performance data. In: Proceedings of the 5th conference on applied electromagnetics, wireless and optical communications, Citeseer, pp 31–37
40. Provotar OI, Linder YM, Veres MM (2019) Unsupervised anomaly detection in time series using lstm-based autoencoders. In: 2019 IEEE International conference on advanced trends in information theory (ATIT), IEEE, pp 513–517
41. Li Y, Li K, Chen C, Zhou X, Zeng Z, Li K (2021) Modeling temporal patterns with dilated convolutions for time-series forecasting. ACM Trans Knowl Discovery Data (TKDD) 16(1):1–22
42. De Vazelhes W, Carey C, Tang Y, Vauquier N, Bellet A (2020) metric-learn: metric learning algorithms in python. J Mach Learn Res 21(1):5447–5452
43. Harrou F, Kadri F, Chaabane S, Tahon C, Sun Y (2015) Improved principal component analysis for anomaly detection: application to an emergency department. Comput Indust Eng 88:63–77
44. Měkota O (2019) Anomaly detection using generative adversarial networks
45. Schlegl T, Seeböck P, Waldstein SM, Schmidt-Erfurth U, Langs G (2017) Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Information processing in medical imaging: 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings, Springer, pp 146–157



**Haiwei Zuo** obtained his Master's degree in Computer Science in 2017 from University College London, UK. He is currently pursuing his Ph.D. at the Department of Electrical and Computer Engineering at Curtin University Malaysia. Additionally, he holds a position as a professional lecturer at the School of Information and Communication, City Polytechnic of Shenzhen (Shenzhen Institute of Technology). His research interests include virtual reality, large language models, and serious gaming.

**Aiqun Zhu** obtained her Ph.D. from Wuhan University. Currently serves as the Director of the Office of Scientific Research and Development at City Polytechnic of Shenzhen. Has long been focused on and engaged in research in the fields of information resource management, vocational education, and academic research management, serving as the chief editor for 2 professional textbooks and authoring numerous journal articles and conference papers.

**Yanping Zhu** a female with a bachelor's degree from Central China Normal University, currently serves as the director of the professional teaching and research department of big data application technology at the School of Information and Communication, City Polytechnic of Shenzhen (Shenzhen Institute of Technology). Her research interests include big data and artificial intelligence, as well as the efficient adaptation of large language models. She has published numerous journal articles and conference papers in these areas of research.

**Yinping Liao** obtained her Master's degree from Huazhong University of Science and Technology, China. She currently serves as the dean of the School of Information and Communication at City Polytechnic of Shenzhen (Shenzhen Institute of Technology). She is also recognized as an expert in the Shenzhen professional capacity-building expert database and is a professional leader in her field. Her research interests encompass big data, artificial intelligence, and intelligent electronics. She has published numerous journal articles and conference papers in these areas of research.

**Shiman Li** obtained her Master's degree from East China Normal University, China. She is currently serving as a Teaching Supervisor at City Polytechnic of Shenzhen (Shenzhen Institute of Technology). Her research interests encompass big data, artificial intelligence, and intelligent electronics. She has published numerous journal articles and conference papers in these areas of research.

**Yun Chen** the Associate Dean of the School of Information and Communication, holds a graduate degree and graduated from South China University of Technology. He is a senior lecturer and a national referee for computer maintenance. His research interests encompass big data and intelligent electronics.