Imputation-based Time-Series Anomaly Detection with Conditional Weight-Incremental Diffusion Models

Chunjing Xiao
Henan University
China
chunjingxiao@gmail.com

Zehua Gou
Henan University
China
zehuagou@outlook.com

Wenxin Tai*
University of Electronic Science and Technology of China
wxtai@outlook.com

Kunpeng Zhang
University of Maryland, College Park
USA
kpzhang@umd.edu

Fan Zhou*
University of Electronic Science and Technology of China
fan.zhou@uestc.edu.cn

ABSTRACT

Existing anomaly detection models for time series are primarily trained with normal-point-dominant data and would become ineffective when anomalous points intensively occur in certain episodes. To solve this problem, we propose a new approach, called DiffAD, from the perspective of time series imputation. Unlike previous prediction- and reconstruction-based methods that adopt either partial or complete data as observed values for estimation, DiffAD uses a density ratio-based strategy to select normal observations flexibly that can easily adapt to the anomaly concentration scenarios. To alleviate the model bias problem in the presence of anomaly concentration, we design a new denoising diffusion-based imputation method to enhance the imputation performance of missing values with conditional weight-incremental diffusion, which can preserve the information of observed values and substantially improves data generation quality for stable anomaly detection. Besides, we customize a multi-scale state space model to capture the long-term dependencies across episodes with different anomaly patterns. Extensive experimental results on real-world datasets show that DiffAD performs better than state-of-the-art benchmarks.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**; • **Mathematics of computing** → **Time series analysis**.

KEYWORDS

Time series, diffusion models, state space model, data imputation

ACM Reference Format:
Chunjing Xiao, Zehua Gou, Wenxin Tai, Kunpeng Zhang, and Fan Zhou. 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 (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599391

*Corresponding Author.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599391

1 INTRODUCTION

Time series anomaly detection aims to identify unusual samples that significantly deviate from the majority in time series. It can enable warnings and precautions in advance that potentially prevent large malfunctions, which is quite meaningful for a broad variety of applications, such as discovering exceptions of underlying systems [55], monitoring data-failures in large-scale datasets [44], and detecting dramatic changes of KPI in business operations [48].

In practical applications, anomalies are often rare and mixed up with vast normal points, making data labeling difficult. Hence, most studies focus on identifying anomalies using unsupervised methods [4, 37]. For example, density estimation [6, 50] and clustering approaches [38, 40] have been designed for anomaly detection, in particular in the context of time series. Recently, benefiting from the representation learning capability of neural networks, deep learning-based techniques have achieved superior performance for anomaly detection and attracted much attention in both academia and industry. They can primarily be summarized into two categories: prediction-based [12, 51] and reconstruction-based [49, 56]. The former builds a predictive model to infer the subsequent data using the historical data, and then determines anomalies based on the prediction errors between estimated values and real values. The reconstruction-based approaches reconstruct the test data based on training instances and then perform anomaly detection based on the reconstruction errors.

Although great success has been achieved by prior studies, they may still suffer from performance degradation especially when the anomalous points are not uniformly distributed over the whole time series but concentrating at some regions – We call this phenomenon *anomaly concentration*. In this case, both prediction- and reconstruction-based methods may fail to accurately identify anomalous points, because their models are usually trained for regions where normal data are dominant [4, 37]. When anomalous points concentrate in some regions, estimation should be significantly influenced by intensive anomalous points in such a context, making existing methods inappropriate and even invalid. An illustrative example of concentrated anomalies is presented in Figure 1, where blue and red points denote normal and anomalous

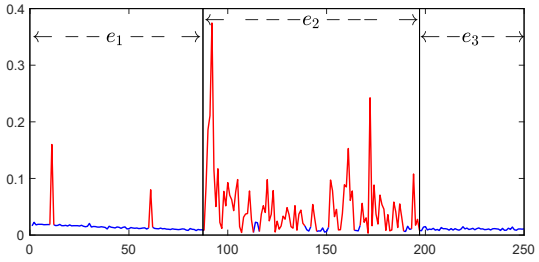

Figure 1: An example of anomaly concentration drawn sampled from the server machine dataset. Blue and red points denote normal and anomalous ones, respectively. Episode e_2 is an anomaly concentration region, and episode e_1 and e_3 are normal data dominant regions.

ones, respectively. There exists an anomaly concentration event in episode e_2. Typical prediction-based models that use past values to forecast the future might incorrectly regard some points in episode e_3 as anomalies. This can result in larger prediction errors for normal points in episode e_3, and correspondingly impair the anomaly detection performance. The same issue also applies to reconstruction-based methods. Our analyses from real data show that numerous anomalous points occur next to each other (referring to Section 2.2). Hence, developing a robust anomaly detection model while accounting for anomaly concentration is desired.

In this work, we introduce a time series imputation solution for anomaly detection while addressing the anomaly concentration problem. The main idea is as follows. We first select a small set of discrete points as observed data (i.e., values are known) and the rest as masked samples (i.e., missing values). Then we impute the missing values based on the observed values. Next, we compute estimation errors between the estimated values and the real values to detect anomalies. In this way, the larger the estimation errors, the higher the probability of being identified as anomalous. Different from prediction- and reconstruction-based methods, the proposed imputation-based approaches can flexibly choose observed values for data estimation. With fewer (even zero) points selected from anomaly-dominant regions as observed ones, our imputation-based method can reduce the impact of intensive anomalous points and relieve the problem caused by anomaly concentration.

In this paper, we propose a diffusion-based imputation framework for time series anomaly detection (called DiffAD), which can effectively mitigate the performance degradation problem in the presence of anomaly concentration. To reduce the influence of anomaly concentration, we present a density ratio-based point selection strategy to choose more normal points as observed ones for data estimation. Selecting normal points is not easy since they are generally unknown. Fortunately, some change point identification methods can help, e.g., the density ratio-based methods such as the SEParation distance change point detection algorithm (SEP) [2]. The rationale behind this is that for time series data, most change points are anomalies [5, 11]. Therefore, for some regions in time series without change points, the longer the region, the higher the probability of all points being normal. In this way, more points can be selected from these regions as observed values.

Besides, we design a conditional weight-incremental diffusion model to estimate missing values based on the observed values. This model takes a Gaussian noise as input and imposes the conditions (i.e., observed values) with varying weights on the reverse diffusion iterations to generate the values of masked points. Different from general imputation tasks that aim to accurately estimate missing values [26, 43], anomaly detection expects a smaller deviation between the estimated values of observed points and the real observed values, in addition to fewer distortions in the estimated values. However, for typical diffusion models, the values of the observed points generated by reverse diffusion iterations might deviate from the true observed values. However, directly exert the condition factor on the generated data (i.e., substituting the generated values with the real observed values at each time step) may make the updated data distorted. To this end, during the exertion procedure, we adjust the weights of the conditions according to the number of iterations, i.e., larger weights for later iterations. As the iteration increases, the generated values become cleaner, and the deviation and distortion diminish gradually. Consequently, imposing the conditions with larger weights will *not* cause serious distortion. Hence, our incremental weight strategy help generate values with less deviation and distortion. The estimated values will be eventually used to calculate estimation errors for final anomaly detection.

Further, we exploit long-term interactions beyond the range of the anomaly concentration in the U-Net of the diffusion model to mitigate the problem caused by anomaly concentration. For general diffusion models such as denoising diffusion probabilistic models (DDPM) [19], the U-Net [35] is utilized as the denoising neural network to remove noises for generating clean samples. However, widely used convolution and pooling operations in U-Net are not able to efficiently capture long-range dependencies [17, 18]. The structured state-space sequence (S4) [17] model has achieved successes in many time series applications, such as audio waveform, movie clip, and neural language processing [17, 18, 21]. In DiffAD, we incorporate S4 into the U-Net and design a *multi-scale* S4-based U-Net, which consolidates information from different tiers at multiple resolutions and exploits longer-range interactions to alleviate the impact of anomaly concentration.

The contributions of this study are four-fold:

- To our knowledge, DiffAD is the first denoising diffusion probabilistic model for time series anomaly detection. We formulate anomaly detection as a generative time series imputation process, which initiates the attempt to explore the DDPM and imputation paradigm for anomaly detection in time series.
- We propose a new solution for adapting DDPM to the task of anomaly detection and ensuring consistency between observed values and generated ones, accompanied by a novel conditional weight-incremental diffusion model with a specially designed multi-scale state space model.
- We provide a density ratio-based selection strategy that can flexibly choose normal values as observed points. This strategy makes it easier for the detection model to identify anomalous points in situations where anomalies are densely concentrated.
- Extensive experiments conducted on five datasets demonstrate the superiority of DiffAD over state-of-the-art time series anomaly detection benchmarks. The code is available online[1].

[1] https://github.com/ChunjingXiao/DiffAD

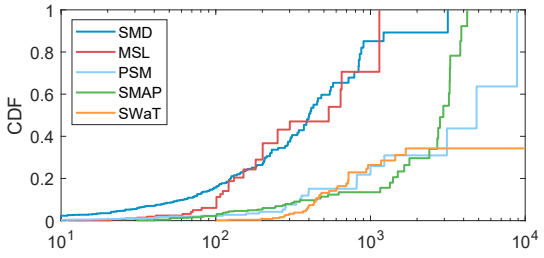

Figure 2: The CDFs of the consecutive anomalous points.

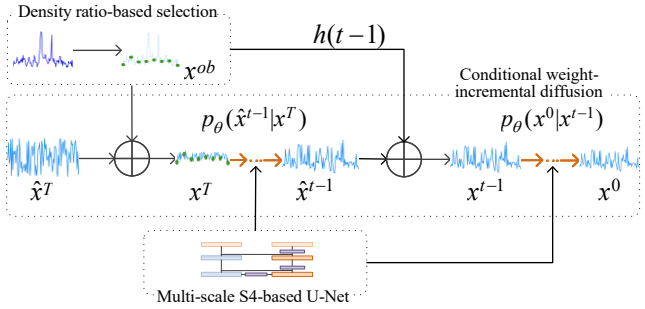

Figure 3: Overview of the DiffAD framework. During the estimation process, the observed values x^{ob} (green dots) are first extracted via the density ratio-based selection strategy. Then, missing values are estimated by the conditional weight-incremental diffusion model. For each reverse diffusion iteration, the conditions x^{ob} are imposed on the generated data $\hat{x}^{t-1}$ with the weight $h(t-1)$ increasing along with the iterations. In addition, a multi-scale S4-based U-Net is designed to learn long-term dependencies. Finally, the estimation errors between estimated values and real values are used to identify anomalies.

2 PRELIMINARIES

In this section, we present a preliminary overview of the time series anomaly detection problem, anomaly concentration and denoising diffusion probabilistic models that are the basic background of studied problem and the motivations of the proposed solution in this work.

2.1 Problem Formulation

In this paper, we focus on unsupervised anomaly detection for multivariate time series. In the training set, each sample $\mathcal{X}$ is represented as a set of time points $\{c_1, c_2, ..., c_N\}$, where $c_i \in \mathbb{R}^d$ represents the observation at time i. The unsupervised time series anomaly detection problem is to determine whether c_i is anomalous or not.

We assume that a few time points are selected as known (observed), and the rest as masked (missing). To detect anomalies, we first adopt time series imputation techniques to estimate the values of masked points. Then, we identify anomalous points by comparing the estimation errors between the real values and estimated values. The higher the estimation errors, the more likely the point are anomalous.

2.2 Degree of Anomaly Concentration

Anomaly concentration refers to the phenomenon where numerous anomalous points consecutively occur at some regions in the time series. To understand the degree of anomaly concentration, we explore the number of consecutive anomalous points in five real datasets: SMD [41], PSM [1], SWaT [29], MSL and SMAP [20]). The distribution of the number of consecutive anomalies is illustrated in Figure 2, where the x-axis represents the number of anomalies that are adjacent.

From the data statistics, we can see that anomalies occur continuously. Even for the dataset with scattered anomalies, e.g., SMD, there are more than 80% anomalous points falling into the regions where the number of consecutive anomalous points is larger than 100. Most data such as PSM, SMAP and SWaT demonstrate a higher degree of anomaly concentration, i.e., more than 70% anomalous points fall into the consecutive anomalous sequences with a length of 1, 000+. These results suggest that anomaly concentration widely exists across the data, which should be taken into account in the anomaly detection models.

2.3 Denoising Diffusion Probabilistic Models

A T-step denoising diffusion probabilistic model (DDPM) [19] consists of two processes: the *diffusion* process with steps $t \in \{0, 1, \cdots, T\}$ and the *reverse* process $t \in \{T, T-1, \cdots, 0\}$. The diffusion process, $q(x^t|x^{t-1})$, gradually corrupts data from some target distribution $q_{\text{data}}(x^0)$ into a Gaussian noise, and the reverse process, $p_(x^{t-1}|x^t)$, generates samples by turning noise into samples from x^T.

Given the data x^0, the diffusion process in the first step ($t = 0$), $q_{data}(x^0)$, is defined as the data distribution x^0 on $\mathbb{R}^L$, where L is the signal length in samples. The *diffusion* process from data x^0 to the variable x^T can be formulated based on a fixed Markov chain. The *reverse* process converts the latent variable $x^T \sim \mathcal{N}(0, I)$ to x^0, which is also based on a Markov chain. The mathematical fundamentals of the typical DDPM can be found in [19].

3 METHOD

DiffAD is composed of three components: the density ratio-based point selection strategy, the conditional weight-incremental diffusion model and the multi-scale S4-based U-Net. An overview of these components is illustrated in Figure 3. To reduce the negative impact of anomaly concentration, the density ratio-based point selection strategy selects more normal points as observed points for data estimation. Then, the conditional weight-incremental diffusion model tries to estimate missing values based on the observed values. In the diffusion model, a multi-scale S4-based U-Net is involved to capture long-range dependencies for alleviating the issue caused by anomaly concentration. The differences between the estimated values and the ground truths are used to identify anomalies – i.e., the larger the estimation errors, the higher the probability of being anomalous. To obtain the estimation errors of observed points, we first select a set of adjacent points of the observed points as new observed points. Then we use these new observed points to estimate values of the original observed points to determine whether the original observed points are anomalous or not.

3.1 Density Ratio-based Point Selection Strategy

Different from prediction-based and reconstruction-based methods, the observed values in our anomaly detection method can be flexibly selected. To mitigate the negative impact of anomaly concentration and enhance detection performance, more normal points should be designated as observed ones. The reason is intuitive: if there are more normal values in the observed data, the estimated values (e.g., inferred based on observed ones) are more likely to approach normal values. This can lead to larger estimation errors for anomalies, yielding superior detection performance. Therefore, selecting more normal points as observed ones can substantially improve the detection performance.

Towards this goal, we design a density ratio-based point selection strategy to select more normal values as observed ones for estimating the masked values. The density ratio is an effective measure for outlier and change point detection [2, 30, 47, 53], and the change point is a time point at which the behavior of a time series changes abruptly [45]. For time series data, most change points can be regarded as a kind of anomaly [5, 11]. We employ the density ratio-based sensitive change scores to determine change points by comparing these scores to a threshold. The main idea of our strategy is that for a piece of data without change points, the longer their length is, the higher the probability that they are normal points. Correspondingly, more points in this piece of data should be selected as observations.

In particular, we identify change points based on the density ratios of two consecutive windows, which can reflect the degree of changes between windows. To compute the change scores, we first divide time series data into h disjoint windows, and compute the density ratio of window k as follows:

$$g_k(x) = \frac{f_{k-1}(x)}{f_k(x)}, \tag{1}$$

where $f_{k-1}(x)$ and $f_k(x)$ correspond to estimated probability densities of the two consecutive windows respectively. Then, the change score is calculated:

$$\hat{\text{CHG}} = \text{Max}\left(0, \quad \frac{1}{2} - \frac{1}{s}\sum_{i=1}^{s} g_k(x^i),\right), \tag{2}$$

where s is the window length. A larger CHG score suggests a higher probability of having change points [2]. Hence, the points with CHG scores higher than a threshold are considered as change points [2]. As change points are unknown to us, the threshold value can be set based on the distribution of all the CHG scores (e.g., the 30-th percentile value). Note that we mainly focus on decreasing the false negative rate of changing point detection, which can be enhanced by using a lower threshold. Based on CHG scores, we are able to extract all regions without notable change points.

To obtain appropriate observed values, the selection strategy must meet two constraints: (i) The selected points should have smaller CHG scores, indicating that they are not change points, and therefore not anomalies; (ii) The intervals between two adjacent selected points should not be too large, since large intervals might increase the difficulty of estimating missing values.

Therefore, for each window without change points, we traverse every point to compute its probability of being selected as observed values. For the i-th point, its probability is:

$$P_i = \frac{d_i - (\hat{\text{CHG}}_i - \hat{\text{CHG}}_{\text{avg}}) * d_i}{\hat{\text{CHG}}_i}, \tag{3}$$

where d_i refers to the distance between point i and the last selected observed point, $\hat{\text{CHG}}_i$ is the CHG score of the window with the i-th point, and $\hat{\text{CHG}}_{\text{avg}}$ denotes the average CHG scores of all windows:

$$\hat{\text{CHG}}_{\text{avg}} = \frac{1}{H-1}\sum_{j=1}^{H-1} \hat{\text{CHG}}_j, \tag{4}$$

where H is the number of all windows in regions. P_i indicates the probability that point i is chosen as observed. We can select a given proportion of points as observed ones, such as the top 10% points with the highest probability. This probability is positively correlated with the distance between this point and the last selected point and is negative with the CHG score. In other words, it conforms to the two rules mentioned above. We add $(\hat{\text{CHG}}_i - \hat{\text{CHG}}_{\text{avg}}) * h_i$ in the numerator to further enhance the role of CHG scores. In this way, a point with a lower CHG score and longer interval tends to be normal and will be selected as the observation for estimating the masked values.

3.2 Conditional Weight-Incremental Diffusion

Due to the superiority of diffusion models in data generation and imputation [26, 33, 43], we exploit DDPM as the basic imputation framework. However, anomaly detection is different from imputation, i.e., the former pursues larger estimation errors for anomalies so as to distinguish the anomalous data, while the latter aims to accurately estimate missing values. To bridge this gap, we design a *conditional weight-incremental diffusion model* to generate estimations based on the observed points. The goal is to generate data with fewer distortions and preserve the information of the observed values by adjusting weights of observed values , which are particularly informative for anomaly detection.

Specifically, the observed values are regarded as the conditions to be exerted during the reverse diffusion iterations. To keep the dimensions of conditions x^{ob} and the initial state of $\hat{x}_T \in \mathcal{N}(0, I)$ consistent, we adopt the bicubic interpolation to resize x^{ob} to the same as $\hat{x}^T$. We then combine the resized x^{ob} and $\hat{x}^T$ to produce the input of the diffusion model. The combination is as follows:

$$x^T = s \odot x^{ob} + (1-s) \odot \left(g(x^{ob})\gamma + \hat{x}^T(1-\gamma)\right), \tag{5}$$

where s is a binary sequence representing which point in data is observed (e.g., 011000 indicates the second and the third are observed points and the rest are masked points). $(1-s)\odot(\cdot)$ means the values of masked points. $g(\cdot)$ is the bicubic interpolation function, and γ is the weight parameter for adjusting the relative importance of two terms.

Then, we impose the condition on the reverse diffusion iteration. According to the Markov chain, the conditional reverse diffusion process aims to predict $\hat{x}^{t-1}$ based on x^t and x^{ob}. After adding the condition x^{ob}, the reverse process of DDPM becomes:

$$p_\theta(\hat{x}^{t-1}|x^t, x^{ob}) = \mathcal{N}(\hat{x}^{t-1}; \mu_\theta(x^t, x^{ob}, t), \tilde{\beta}_t \mathbf{I}), \tag{6}$$

where $\mu_\theta(x^t, x^{ob}, t)$ is the estimated mean of the conditional reverse process, and $\tilde{\beta}_t$ is a fixed constant. The reverse process starts with

a Gaussian noise $\hat{x}^T$, and generates a clean sample x^0 by sampling reverse steps $p_\theta(\hat{x}^{t-1}|x^t, x^{ob})$.

To parameterize $\mu_\theta(x^t, x^{ob}, t)$, we train a neural denoising model $f_\theta(x^t, x^{ob}, t)$ to predict the noise vector ϵ. The objective is defined as follows:

$$\mathbb{E}_{x^{ob}}\mathbb{E}_{(\epsilon,t)}\left[\|f_\theta(x^t, x^{ob}, t) - \epsilon\|_2^2\right], \tag{7}$$

and $\mu_\theta(x^t, x^{ob}, t)$ can be derived from $f_\theta(x^t, x^{ob}, t)$:

$$\mu_\theta(x^t, x^{ob}, t) = \frac{1}{\sqrt{a_t}}\left(x^t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}} f_\theta(x^t, x^{ob}, t)\right). \tag{8}$$

Consequently, the generative (reverse diffusion) process is:

$$\hat{x}^{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x^t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} f_\theta(x^t, x^{ob}, t)\right) + \sqrt{\tilde{\beta}_t}\mathbf{z}, \tag{9}$$

where $\mathbf{z} \sim N(0, \mathbf{I})$, implying that each generation step is stochastic. This generative process iteratively refines the distribution until reaching a clean sample x^0.

However, simply injecting the condition into the denoising network and expecting the network to automatically exploit condition information to generate the expected data might suffer from the deviation problem. In other words, after multiple iterations, the generated values of the observed points might deviate from their real values. An alternative way is to forcibly exert the condition factor on the generated data (i.e., replacing the generated values with the real observed values at each time step), which, however, may cause the distortion problem.

To this end, we propose a novel weight-incremental condition injecting approach to efficiently address this problem. In other words, we introduce a monotonic function $h(\cdot)$ to adjust the weights of the conditions (e.g., observed values x^{ob}). $h(\cdot)$ has a co-domain from 0 to 1 and can be obtained based on the exponential decay equation:

$$h(t-1) = N_0 e^{-\lambda(t-1)}, \tag{10}$$

where λ is a hyper-parameter representing the exponential decay constant, and N_0 is the initial quantity. The motivation is that, as timestamp t decreases and reverse sampling continues, the generated values of observed points should get closer to real observed values – accordingly, the weight $h(\cdot)$ should become larger to guarantee consistency between the generated values and observed ones.

Having the weight function $h(\cdot)$, we combine the observed values into the generative ones to produce x^{t-1}:

$$x^{t-1} = s \odot \left(\Big(1 - h(t-1)\Big)\hat{x}^{t-1} + h(t-1)x^{ob}\right) + (1-s) \odot \hat{x}^{t-1}, \tag{11}$$

where $s \odot (\cdot)$ and $(1-s) \odot \hat{x}^{t-1}$ refer to the generated values of observed points and masked points, respectively. Note that the proposed conditional weight-incremental diffusion learning has some appealing properties: (*i*) When $t \to T$, $h(t)$ limits to 0, thus the condition factor will have (almost) no effect on the initial state of DDPM; (*ii*) Since $h(t)$ is a monotonically decreasing function, it satisfies the motivation mentioned above (the smaller timestamp, the tighter condition constraint). Due to the continuous property of the exponential function, it can correct deviation gently and progressively. (*iii*) When $t \to 0$, $h(t)$ get the largest value: $h(t) = N_0$. And, according to Eq.(11) and our experimental setting, we can guarantee that the positions of observed points will be completely substituted by the ground true observed values.

As shown in Figure 3, the estimation process starts with a Gaussian noise $\hat{x}^T$ and generates a clean sample x^0 through iterative refinement, i.e., $\hat{x}^T \to x^T \cdots \to \hat{x}^{t-1} \to x^{t-1} \to \cdots \to x^0$. At each reverse step $t \in \{T, T-1, \cdots, 0\}$, we first generate the noisy candidate $\hat{x}^{t-1}$ according to Eq.(9). Then, we apply Eq.(11) to explicitly exert conditions on $\hat{x}^{t-1}$ to get x^{t-1}. Finally, the masked values can be estimated by converting the Gaussian noise $\hat{x}^T$ into a clean sample x^0, conditioned on observed points x^{ob}.

3.3 Multi-scale S4-based U-Net

In diffusion models, the U-Net [35] based on a Wide ResNet [54] is generally used as the denoising neural network [19]. However, the components such as convolution and pooling operations in this U-Net backbone cannot efficiently handle data with long-range dependencies [17, 18]. In the presence of anomaly concentrations, exploiting long-term interactions is particularly important in capturing time series patterns beyond the anomaly concentration episodes. Here we introduce the structured state-space sequence model (S4) [17] for U-Net to handle such long-term dependencies. S4 is recently proposed to model long-range dependencies for sequence modeling tasks [17] and has been successfully applied to Autoregressive and Transformer models for various applications, such as speech classification [18], audio generation [16], and movie classification [21].

Specifically, we customize a *multi-scale* S4-based U-Net backbone to consolidate information from different tiers at multiple resolutions. Our proposed network architecture consists of multiple tiers with each containing a stack of residual S4 blocks. The top tier processes the raw time series data at its original sampling rate, while the lower tiers process downsampled versions of the input signal. The output of lower tiers is upsampled and combined with the input to the above tier in order to provide a stronger conditioning signal. Owing to the multi-scale strategy, different blocks can learn features at different scales, which helps the model to learn complex temporal dependencies over long time series data. This multi-scale S4-based U-Net will be used as the denoising neural network $f_\theta(\cdot)$ in the conditional weight-incremental diffusion model.

3.4 Detection Criterion

For anomaly detection, we compare the anomaly score with a given threshold to determine anomalies. For a testing point, its anomaly score is computed based on the estimation error:

$$AS(c_i) = \sum_{k=1}^{d} \|c_i^k - \hat{c}_i^k\|^2, \tag{12}$$

where c_i and $\hat{c}_i$ are the real value and estimated value, respectively, and d refers to the dimension of multivariate time series.

Similar to the previous works [57], we obtain the threshold based on the training data. Given the training data $\mathcal{X} = \{c_1, c_2, ..., c_N\}$, the corresponding decision threshold $\mathcal{T}$ is:

$$\mathcal{T} = \frac{1}{N}\sum_{i=1}^{N} \ell(c_i) + \sqrt{\frac{1}{N}\sum_{k=1}^{N}\big(\ell(c_i) - \ell_{\text{avg}}\big)^2}, \tag{13}$$

Table 1: Descriptive Statistics of Datasets

Datasets	Applications	# Dimension	# point	anomaly(%)
MSL	Space	55	132,046	10.5%
SWaT	Water	51	944,919	12.1%
PSM	Server	25	220,322	27.8%
SMAP	Space	25	562,800	12.8%
SMD	Server	38	1,416,825	4.2%

where $\ell(c_i)$ refers to the sum of loss function for c_i, and ℓ_{avg} denotes the average value. A testing sample is considered to be abnormal if $AS(c_i) > \mathcal{T}$, or normal otherwise.

4 EXPERIMENTAL EVALUATION

In this section, we evaluate the efficacy of our model in terms of anomaly detection performance, ablation study, and the role of point selection strategies and observation numbers.

4.1 Experimental Settings

Datasets. We conducted experiments on the benchmarks for time series anomaly detection, as shown in Table 1. (1) SWaT (Secure Water Treatment) is obtained from 51 sensors of the critical infrastructure system under continuous operations [29]. (2) PSM (Pooled Server Metrics) is collected internally from multiple application server nodes at eBay [1]. (3) SMD (Server Machine Dataset) is a 5-week-long dataset collected from a large Internet company [41]. (4) Both MSL (Mars Science Laboratory rover) and SMAP (Soil Moisture Active Passive satellite) are public datasets from NASA, which contain the telemetry anomaly data derived from the Incident Surprise Anomaly reports of spacecraft monitoring systems [20].

Baselines. We compare our model DiffAD with various baselines from five major categories: (1) The density estimation methods (e.g., DAGMM [60], MPPCACD [50], and LOF [6]) which evaluate the density of time series data to detect anomalies; (2) The clustering-based approaches (e.g., ITAD [40], THOC [39], and Deep-SVDD [36]) divide data sequences into clusters and identify anomalies by comparing their distance from clusters; (3) The time series imputation techniques (e.g., CSDI [43], and STING [31]) are designed for imputing missing values based on observed values. Note that they are primarily designed for data imputation rather than anomaly detection. Here we adapt them to time series anomaly detection using their imputation errors; (4) The prediction-based models (e.g., CL-MPPCA [42], and LSTM [20]) learn a predictive model to forecast values in a number of time steps based on values in the current context window and identify anomalies according to the prediction errors; (5) The reconstruction-based methods (e.g., LSTM-VAE [32], BeatGAN [59], OmniAnomaly [41], InterFusion [24], and ATransformer [49]) build models to understand the normal behavior by encoding subsequences of a normal training time series in a latent space, and determine anomalies according to reconstruction errors.

Experiment Setup. We use the Adam optimizer [22] with an initial learning rate of $3 * 10^{-6}$, and set the batch size to 16 for SMD and 32 for the other four datasets based on the validation set. We use 100 diffusion steps for the diffusion model. The values of different dimensions at a point are simultaneously estimated. We empirically set $\gamma = 0.9$ in Eq.(5), $\lambda = 25$ and $N_0 = 1$ in Eq.(10). We adopt the widely-used adjustment strategy [39, 41, 48]: if a time point in a certain successive abnormal segment is detected, all anomalies in this abnormal segment are considered to be correctly detected. This strategy is justified by the observation that an abnormal time point will cause an alert and further make the whole segment noticed in real-world applications [39].

As for the number of observed points, we select the top 6% of points with the highest selection probability computed using Eq.(3) as the observed points for MSL, and 12% for the other four datasets. The observed points are used as the conditions of our designed diffusion model to estimate the values of masked points.

4.2 Anomaly Detection Results

We report the anomaly detection results of DiffAD and the baselines in Table 2, where the P, R and F1 represent the Precision, Recall and F1 score, respectively. From the table, we have the following observations. First, our proposed DiffAD method achieves better performance than the baseline approaches across datasets in most evaluation metrics. Although the recall of CL-MPPCA is slightly higher than our model, its F1 score and precision are significantly lower than ours. Notably, our model exceeds the best baseline by a large margin on SWaT, because the anomalies on SWaT concentrate more heavily. This result verifies the superiority of our method in improving detection performance in the presence of a higher degree of anomaly concentration.

Second, the time series imputation methods, i.e., CSDI and STING, achieve relatively good performance, compared with density estimation and clustering-based approaches. This indicates that the imputation models are potential techniques for time series anomaly detection. Surprisingly, the two communities are very close but rarely intersect. However, our model outperforms the two state-of-the-art imputation models, which validates the effect of the proposed multi-scale learning network and observed point selection strategy in time series anomaly detection.

Third, ATransfomer, a specially designed method with Transformers for anomaly detection, surpasses other baselines, suggesting advanced deep learning techniques such as Transformers can promote the performance of time series anomaly detection. However, our DiffAD outperforms ATransfomer by introducing the diffusion model-based imputation technique, which verifies the effectiveness of the proposed weight-incremental diffusion model in anomaly detection.

4.3 Ablation Study

Next, we evaluate the role of the important components in DiffAD: the conditional weight-incremental diffusion model, multi-scale S4-based U-Net and density ratio-based point selection strategy. We mainly investigate the following variants: (1) *DiffAD-Base* is a basic diffusion model without having three components (i.e., the increasing condition weights, multi-scale S4-based U-Net and density ratio-based selection strategy). (2) *DiffAD-Wei* is a model with conditional weight-incremental diffusion. (3) *DiffAD-MS4* is a model equipped with our designed multi-scale S4-based U-Net. (4) *DiffAD-CHG* is a model with the density ratio-based selection strategy.

The anomaly detection performance on the five datasets is presented in Table 3. As shown, compared with *DiffAD-Base*, *DiffAD-Weight* obtain a better detection result, suggesting that our weight-varying for the conditional diffusion indeed enhances anomaly

Table 2: Performance comparison between DiffAD and baselines on the five datasets.

Method	MSL			SWaT			PSM			SMAP			SMD		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DAGMM	89.60	63.93	74.62	89.92	57.84	70.40	93.49	70.03	80.08	86.45	56.73	68.51	67.30	49.89	57.30
MPPCACD	81.42	61.31	69.95	82.52	68.29	74.73	76.26	78.35	77.29	88.61	75.84	81.73	71.20	79.28	75.02
LOF	47.72	85.25	61.18	72.15	65.43	68.62	57.89	90.49	70.61	58.93	56.33	57.60	56.34	39.86	46.68
ITAD	69.44	84.09	76.07	63.13	52.08	57.08	72.80	64.02	68.13	82.42	66.89	73.85	86.22	73.71	79.48
THOC	88.45	90.97	89.69	83.94	86.36	85.13	88.14	90.99	89.54	92.06	89.34	90.68	79.76	90.95	84.99
Deep-SVDD	91.92	76.63	83.58	80.42	84.45	82.39	95.41	86.49	90.73	89.93	56.02	69.04	78.54	79.67	79.10
CSDI	90.46	90.92	90.69	91.66	91.98	91.82	94.30	95.23	94.76	94.23	93.85	94.04	88.32	89.03	88.67
STING	88.25	89.15	88.70	87.28	87.69	87.48	92.35	93.47	92.91	88.97	89.85	89.41	85.14	86.49	85.81
CL-MPPCA	73.71	88.54	80.44	76.78	81.50	79.07	56.02	**99.93**	71.80	86.13	63.16	72.88	82.36	76.07	79.09
LSTM	85.45	82.50	83.95	86.15	83.27	84.69	76.93	89.64	82.80	89.41	78.13	83.39	78.55	85.28	81.78
LSTM-VAE	85.49	79.94	82.62	76.00	89.50	82.20	73.62	89.92	80.96	92.20	67.75	78.10	75.76	90.08	82.30
BeatGAN	89.75	85.42	87.53	64.01	87.46	73.92	90.30	93.84	92.04	92.38	55.85	69.61	72.90	84.09	78.10
OmniAnomaly	89.02	86.37	87.67	81.42	84.30	82.83	88.39	74.46	80.83	92.49	81.99	86.92	83.68	86.82	85.22
InterFusion	81.28	92.70	86.62	80.59	85.58	83.01	83.61	83.45	83.52	89.77	88.52	89.14	87.02	85.43	86.22
ATransformer	92.09	95.15	93.59	91.55	96.73	94.07	96.91	98.90	97.89	94.13	**99.40**	96.69	89.40	95.45	92.33
DiffAD	**92.97**	**95.44**	**94.19**	**98.44**	**96.90**	**97.66**	**97.00**	98.92	**97.95**	**96.52**	97.38	**96.95**	**90.01**	**95.67**	**92.75**

detection performance. Moreover, by considering the multi-scale S4-based U-Net, *DiffAD-MS4* achieves a great improvement in detection accuracy compared with *DiffAD-Base*. This indicates the multi-scale network can capture long-term dependencies. Besides, *DiffAD-Change* outperforms *DiffAD-Base* by a certain margin, implying that it is essential to select proper points as the observations.

Table 3: The role of different components

Dataset	Metric	DiffAD-Base	DiffAD-Weight	DiffAD-MS4	DiffAD-CHG	DiffAD (Full)
MSL	P	85.64	87.51	89.84	90.42	**92.97**
	R	86.55	87.46	87.35	92.57	**95.44**
	F1	86.09	87.48	88.58	91.48	**94.19**
SWaT	P	90.12	91.22	94.37	95.55	**98.44**
	R	91.24	91.94	93.65	96.16	**96.90**
	F1	90.68	91.58	94.01	95.85	**97.66**
PSM	P	91.61	92.49	94.05	95.98	**97.00**
	R	93.64	94.54	95.87	96.13	**98.92**
	F1	92.61	93.50	94.95	96.05	**97.95**
SMAP	P	89.45	90.36	92.62	94.28	**96.52**
	R	90.44	91.03	92.99	95.67	**97.38**
	F1	89.94	90.69	92.80	94.97	**96.95**
SMD	P	81.87	83.21	85.64	88.33	**90.01**
	R	85.42	86.89	89.33	92.75	**95.67**
	F1	83.61	85.01	87.45	90.49	**92.75**

4.4 Role of Selection Strategy

One of the main differences between our work and typical imputation tasks is that the observed points in this work can be flexibly chosen. Therefore, we investigate how the selected points impact the detection performance. We compare our designed selection strategy with the following: (*i*) *Random strategy*: It randomly selects a certain percentage of points as the observed ones. (*ii*) *Fixed-interval strategy*: The observed points are selected with a fixed interval. (*iii*) *Forecasting strategy*: It simulates the time series forecasting task, which iteratively selects the past values as the observed ones to forecast future values. Except for the *forecasting strategy*, the other strategies including ours select the same percentage of data as the observed points, i.e., 6% for MSL and 12% for SWaT, PSM, SMAP and SMD.

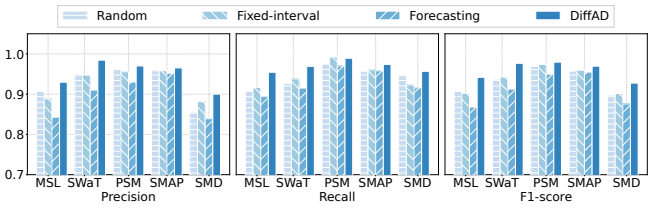

Figure 4: Performance with different selection strategies.

Figure 4 illustrates the detection results on the five datasets. First, our density ratio-based selection strategy achieves the best performance in terms of all three metrics, meaning that our strategy can efficiently choose more normal values as observations, and further assist downstream anomaly detection. Second, *forecasting strategy* performs the worst among all strategies. When both historical values and future values contain many normal values, the future values can be accurately estimated, i.e., the model is effective in anomaly detection. However, the future normal points might be incorrectly labeled as anomalies when anomalies are dominant in the past values, which, unfortunately, may deteriorate the model performance greatly. Third, *random strategy* and *fixed-interval strategy* obtain similar performance, because the ratios of normal points among the observed points selected by the two methods are quite close. As other settings are the same, the selected observed points determine the detection performance and, as a result, make the two strategies similar.

Figure 5 further plots examples of the imputation results from SMD and the observed points selected using different strategies. The visual results support the above analyses that our method

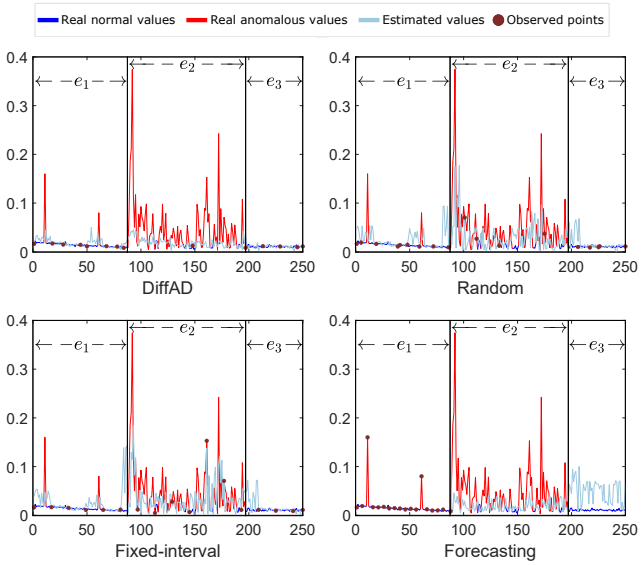

Figure 5: Visual examples for different selection strategies.

can select more normal points as observed ones, and obtain larger estimation errors for anomalies, yielding better anomaly detection performance. In addition, *forecasting strategy* might incorrectly annotate normal points (e.g., those in e_3) as anomalous ones when the past points (e.g., those in e_2) are dominated by anomalies.

4.5 Influence of Observed Points

The number of observed points is critical to missing value estimation, which determines estimation errors, and, subsequently, impacts the anomaly detection results. Hence, we inspect the performance by varying the number of observed points. In this experiment, we compare our DiffAD with two time series imputation baselines, i.e., STING and CSDI.

The F1 and Recall scores on five datasets are illustrated in Figure 6. Clearly, DiffAD obtains relatively lower performance when the number of the observed points is small (e.g., 3%) and large (e.g., 24%). On one hand, a few points may not represent the overall distribution of all normal points in the dataset, especially for MSL where there is a large fluctuation at different time points. A small number of points are inadequate for the models to estimate the values of masked points, resulting in low detection performance. On the other hand, when selecting too many observed points, a few anomalies might be selected as observed ones for estimating values of the masked points, which may lead to small estimation errors for anomalies – and therefore poor detection performance. Besides, our model DiffAD always outperforms the two imputation baselines, regardless of the number of observed points. This result further validates the effectiveness of DiffAD in selecting meaningful observations for anomaly detection.

4.6 Case Study

To provide more intuitive cases of DiffAD, we present some examples in Figure 7. It shows the input data and results learned by DiffAD on five datasets. Here, the raw input data is plotted in the upper row (blue lines), and their corresponding estimation results are shown in the lower row. The point-wise anomalies are bounded

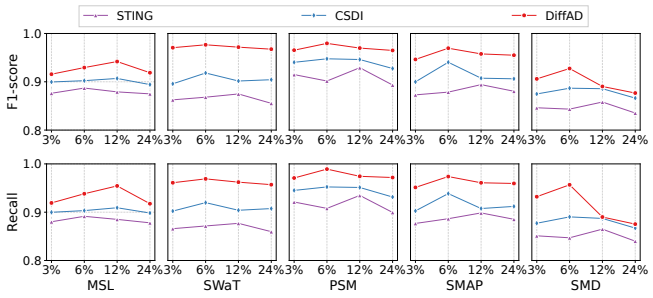

Figure 6: Influence of the number of observed points.

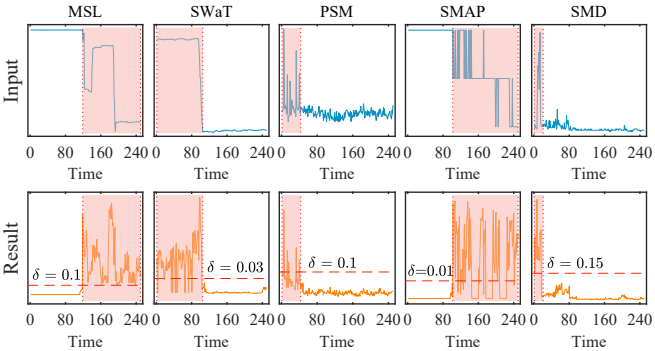

Figure 7: Examples of anomaly detection using DiffAD. Blue lines (upper row) are inputs. Orange lines are estimation errors and red dashed lines are thresholds used to detect anomalies (i.e., anomalies when above the red lines).

by red boxes. We can find that our proposed DiffAD is able to distinguish anomalies in the data. For the data pieces with normal points, the estimation errors (orange lines) keep stable. However, they have great fluctuations in the presence of anomalous points. This feature provides an easy way to select a proper criterion. It also verifies that our criterion can highlight the anomalies and provide distinct values for both normal and abnormal points, making the detection precise and reducing the false-positive rate.

5 RELATED WORK

5.1 Time Series Anomaly Detection

Anomaly detection is a widely studied subject due to its diverse applications. Recently, inspired by the success of deep learning, many deep anomaly detection models have been proposed and achieved remarkable success on time series anomaly detection. Existing models generally fall into two categories: reconstruction-based and prediction-based methods.

Reconstruction-based methods aim to learn the latent representation for the entire time series for data reconstruction and use the reconstruction errors for anomaly detection. Different deep generative techniques have been implemented to build such reconstruction models, including Variational Auto-Encoder (VAE) and Generative Adversarial Network (GAN). For example, Donut [48] introduced VAE to reconstruct the normal points and computed the reconstruction probability at each time point for anomaly judgment. Further, LSTM was incorporated into VAE to detect anomalies in the robot-assisted feeding system [32]. A regularized encoder-decoder architecture was proposed to detect sequence anomalies in ECG

time series [8]. Transformers and graph relational learning have also been explored to distinguish anomalies for multivariate time series data [46, 49, 56].

Inspired by the advances of GAN, MAD-GAN enhances anomaly detection in multivariate time series with adversarial learning, where the anomaly score was defined as the combination of reconstruction loss and discriminating loss [23]. TadGAN [15], a reconstruction-GAN-based architecture, uses cycle-consistent GAN architecture to prevent the contradiction between the encoder and decoder. Another model called BeatGAN [59] was designed to detect anomalous rhythms in time series. It consists of an autoencoder with adversarial regularization in training, and a discriminator for distinguishing real sequences from reconstructed ones. Omni-Anomaly [41] uses stochastic Recurrent Neural Network to find robust representations for multivariate time series. USAD [3] presents an autoencoder architecture whose adversarial-style learning is also inspired by GAN.

Prediction-based methods utilize advanced deep learning to forecast future ones based on past values, and then use the prediction errors between the predicted values and real observations to determine anomalies. For instance, an LSTM model was designed to predict future telemetry data based on historical telemetry data [20], towards identifying spacecraft anomalies in multi-channel telemetry data. MTAD-GAT [58] combines feature-oriented Graph Attention Network (GAT) and time-oriented GAT to handle spatial dependence and temporal dependence for forecasting. KfreqGAN [51] adopts an adversarially trained sequence predictor to predict future sequences. GDN [12] proposes a GNN-based method to aggregate the information between sensors.

Difference: Our imputation-based approach adopts different data to infer the values of test data. In comparison, reconstruction-based methods use all the test data to reconstruct their values, while prediction-based models utilize historical values to forecast the future. Since our imputation-based approaches use discrete observed values, which allows it to flexibly select observations for the masked value estimation. The masked values are among observed values, which can provide more valuable information for data estimation and provide expected results. Besides, we design a density ratio-based point selection strategy to choose appropriate observed points and present a conditional weight-incremental diffusion model with the multi-scale S4-based network to enhance the detection performance.

5.2 Time Series Imputation

Time series imputation aims to estimate missing data based on the observed values. In general, deep learning models can capture the temporal dependencies in time series and achieve more accurate imputation than statistical methods [13]. Among them, *RNN-based methods* use various RNNs, including LSTMs and GRUs, to model the time series. In this line, GRU-D [9] addresses the missing data problem in the time series classification problem. M-RNN [52] and BRITS [7] interpolate missing values according to the hidden states from bidirectional RNN.

GAN-based and *VAE-based methods* utilize generative models to produce imputation values. For example, VAE architecture has been exploited for time series imputation in [14] where a Gaussian process prior is considered in the latent space. L-VAE [34] uses an additive multi-output Gaussian process prior to accommodate auxiliary covariate information. GRUI [27] (GRU for Imputation) enhances the generator and discriminator of the GAN model to learn temporal patterns for incomplete time series imputation. Luo et al. [28] propose an end-to-end method called E^2GAN which adopts an auto-encoder based on GRUI to form the generator for alleviating the difficulty of model training. NAOMI [25] is a non-autoregressive model for spatiotemporal sequence imputation, which consists of a bidirectional encoder and a multiresolution decoder. Recently, diffusion models and graph neural networks are also introduced for time series imputation [10, 43].

Difference: There are two major differences between data imputation and anomaly detection tasks. (*i*)The anomaly detection task aims to enlarge the estimation errors for anomalies instead while the goal of the imputation task is to accurately estimate the missing values; (*ii*) The anomaly detection task can flexibly choose observed points, but data imputation usually uses adjacent data for missing value interpolation. With the differences in mind, we design a conditional weight-incremental diffusion model for anomaly detection and a density ratio-based point selection strategy to obtain appropriate observed points.

6 CONCLUSIONS

In this paper, we proposed a diffusion-based imputation framework for time series anomaly detection, DiffAD. It can effectively mitigate the performance decline issue in the context of anomaly concentration. We designed a conditional weight-incremental diffusion model, which exerts the observed values with incremental weights on the reverse diffusion iterations to gradually refine the estimated results. Further, we implemented a multi-scale S4-based U-Net for the diffusion model to capture long-term dependencies in time series data, and a density ratio-based selection strategy to choose more normal values as observed ones to estimate missing values for accurate anomaly detection. Extensive experiments demonstrate the superiority of our proposed approach over state-of-the-art baselines on time series benchmarks.

Limitations & Future work. Compared with prediction- and reconstruction-based methods, our proposed DiffAD has to select observed points. This increases the computation cost in anomaly detection. There are alternative strategies, such as random, fixed-interval, and forecasting strategies that can balance the detection efficiency and effectiveness as studied in Section 4.4. Meanwhile, the anomaly detection in DiffAD is actually performed twice: one for the masked value imputation and the other for the selected observed values. To compute the estimation errors of the observed points, our model would select a batch of new observed points. Generally, the adjacent points of the originally observed points are selected as the new observed points for data estimation. This process also slightly increases the workload, although it can be performed in parallel using GPUs.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 62176043 and 62072077) and Natural Science Foundation of Sichuan Province (Grant No. 2022NSFSC0505).

REFERENCES

[1] Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. 2021. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *KDD*. 2485–2494.

[2] Samaneh Aminikhanghahi, Tinghui Wang, and Diane J. Cook. 2019. Real-Time Change Point Detection with Application to Smart Home Time Series Data. *IEEE Transactions on Knowledge and Data Engineering* 31, 5 (2019), 1010–1023.

[3] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. 2020. Usad: Unsupervised anomaly detection on multivariate time series. In *KDD*. 3395–3404.

[4] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. 2021. A review on outlier/anomaly detection in time series data. *Comput. Surveys* 54, 3 (2021), 1–33.

[5] Mohammad Braei and Sebastian Wagner. 2020. Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv* (2020).

[6] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *SIGMOD*. 93–104.

[7] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. Brits: Bidirectional recurrent imputation for time series. In *NIPS*.

[8] Ashutosh Chandra and Rahul Kala. 2019. Regularised encoder-decoder architecture for anomaly detection in ECG time signals. In *ICTC*. 1–6.

[9] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8, 1 (2018), 1–12.

[10] Andrea Cini, Ivan Marisca, and Cesare Alippi. 2022. Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks. In *ICLR*.

[11] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. 2019. Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal* 7, 7 (2019), 6481–6494.

[12] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *AAAI*, Vol. 35. 4027–4035.

[13] Chenguang Fang and Chen Wang. 2020. Time series data imputation: A survey on deep learning approaches. *arXiv* (2020).

[14] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. 2020. Gp-vae: Deep probabilistic time series imputation. In *AISTATS*. 1651–1661.

[15] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2020. TadGAN: Time series anomaly detection using generative adversarial networks. In *Big Data*. 33–43.

[16] Karan Goel, Albert Gu, Chris Donahue, and Christopher Re. 2022. Its Raw! Audio Generation with State-Space Models. In *ICML*. 7616–7633.

[17] Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. In *ICLR*.

[18] Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. 2022. On the Parameterization and Initialization of Diagonal State Space Models. In *NIPS*.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NIPS*. 6840–6851.

[20] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *KDD*. 387–395.

[21] Md Mohaiminul Islam and Gedas Bertasius. 2022. Long movie clip classification with state-space video models. In *ECCV*.

[22] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

[23] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *ICANN*. 703–716.

[24] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. 2021. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *KDD*. 3220–3230.

[25] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. 2019. Naomi: Non-autoregressive multiresolution sequence imputation. In *NIPS*.

[26] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*. 11461–11471.

[27] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. 2018. Multivariate time series imputation with generative adversarial networks. In *NIPS*.

[28] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. 2019. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *IJCAI*. 3094–3100.

[29] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *CySWater*. 31–36.

[30] Gyoung S Na, Donghyun Kim, and Hwanjo Yu. 2018. Dilof: Effective and memory efficient local outlier detection in data streams. In *KDD*. 1993–2002.

[31] Eunkyu Oh, Taehun Kim, Yunhu Ji, and Sushil Khyalia. 2021. STING: Self-attention based Time-series Imputation Networks using GAN. In *ICDM*.

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

[33] Tal Peer, Simon Welker, and Timo Gerkmann. 2023. DiffPhase: Generative Diffusion-based STFT Phase Retrieval. In *ICASSP*. 7402–7406.

[34] Siddharth Ramchandran, Gleb Tikhonov, Kalle Kujanpää, Miika Koskinen, and Harri Lähdesmäki. 2021. Longitudinal variational autoencoder. In *AISTATS*. 3898–3906.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. 234–241.

[36] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *ICML*. 4393–4402.

[37] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *VLDB* 15, 9 (2022), 1779–1797.

[38] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13, 7 (2001), 1443–1471.

[39] Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. In *NIPS*, Vol. 33. 13016–13026.

[40] Youjin Shin, Sangyup Lee, Shahroz Tariq, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S Woo. 2020. Itad: integrative tensor-based anomaly detection system for reducing false positives of satellite systems. In *CIKM*. 2733–2740.

[41] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *KDD*. 2828–2837.

[42] Shahroz Tariq, Sangyup Lee, Youjin Shin, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S Woo. 2019. Detecting anomalies in space using multivariate convolutional LSTM with mixtures of probabilistic PCA. In *KDD*. 2123–2133.

[43] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *NIPS*.

[44] Luan Tran, Min Y Mun, and Cyrus Shahabi. 2020. Real-time distance-based outlier detection in data streams. *VLDB* 14, 2 (2020), 141–153.

[45] Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing* 167 (2020), 107299.

[46] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. 2022. TranAD: deep transformer networks for anomaly detection in multivariate time series data. *VLDB* 15, 6 (2022), 1201–1214.

[47] Chunjing Xiao, Shiming Chen, Fan Zhou, and Jie Wu. 2022. Self-Supervised Few-Shot Time-series Segmentation for Activity Recognition. *IEEE Transactions on Mobile Computing* (2022).

[48] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *WWW*. 187–196.

[49] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *ICLR*.

[50] Takehisa Yairi, Naoya Takeishi, Tetsuo Oda, Yuta Nakajima, Naoki Nishimura, and Noboru Takata. 2017. A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction. *IEEE Trans. Aerospace Electron. Systems* 53, 3 (2017), 1384–1401.

[51] Yueyue Yao, Jianghong Ma, and Yunming Ye. 2022. KfreqGAN: Unsupervised detection of sequence anomaly with adversarial learning and frequency domain information. *Knowledge-Based Systems* 236 (2022), 107757.

[52] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. 2018. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering* 66, 5 (2018), 1477–1490.

[53] Susik Yoon, Jae-Gil Lee, and Byung Suk Lee. 2020. Ultrafast local outlier detection from a data stream with stationary region skipping. In *KDD*. 1181–1191.

[54] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. In *British Machine Vision Conference*.

[55] Aoqian Zhang, Shaoxu Song, and Jianmin Wang. 2016. Sequential data cleaning: A statistical approach. In *SIGMOD*. 909–924.

[56] Weiqi Zhang, Chen Zhang, and Fugee Tsung. 2022. GRELEN: Multivariate Time Series Anomaly Detection from the Perspective of Graph Relational Learning. In *IJCAI*. 2390–2397.

[57] Yuxin Zhang, Yiqiang Chen, Jindong Wang, and Zhiwen Pan. 2023. Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2023), 2118–2132.

[58] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Multivariate time-series anomaly detection via graph attention network. In *ICDM*. 841–850.

[59] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. 2019. BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series.. In *IJCAI*. 4433–4439.

[60] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*.