Cluster-Wide Task Slowdown Detection in Cloud System

Feiyi Chen
chenfeiyi@zju.edu.cn
Zhejiang University, Alibaba Group
Hangzhou, China

Yingying Zhang
congrong.zyy@alibaba-inc.com
Alibaba Group
Hangzhou, China

Lunting Fan
lunting.fan@taobao.com
Alibaba Group
Hangzhou, China

Yuxuan Liang
yuxliang@outlook.com
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China

Guansong Pang
gspang@smu.edu.sg
Singapore Management University
Singapore, Singapore

Qingsong Wen
qingsongedu@gmail.com
Squirrel AI
Bellevue, USA

Shuiguang Deng*
dengsg@zju.edu.cn
Zhejiang University
Hangzhou, China

ABSTRACT

Slow task detection is a critical problem in cloud operation and maintenance since it is highly related to user experience and can bring substantial liquidated damages. Most anomaly detection methods detect it from a single-task aspect. However, considering millions of concurrent tasks in large-scale cloud computing clusters, it becomes impractical and inefficient. Moreover, single-task slowdowns are very common and do not necessarily indicate a malfunction of a cluster due to its violent fluctuation nature in a virtual environment. Thus, we shift our attention to cluster-wide task slowdowns by utilizing the duration time distribution of tasks across a cluster, so that the computation complexity is not relevant to the number of tasks. The task duration time distribution often exhibits compound periodicity and local exceptional fluctuations over time. Though transformer-based methods are one of the most powerful methods to capture these time series normal variation patterns, we empirically find and theoretically explain the flaw of the standard attention mechanism in reconstructing subperiods with low amplitude when dealing with compound periodicity. To tackle these challenges, we propose SORN (i.e., Skimming Off subperiods in descending amplitude order and Reconstructing Non-slowing fluctuation), which consists of a Skimming Attention mechanism to reconstruct the compound periodicity and a Neural Optimal Transport module to distinguish cluster-wide slowdowns from other exceptional fluctuations. Furthermore, since anomalies in the training set are inevitable in a practical scenario, we propose a picky loss function, which adaptively assigns higher weights to reliable time slots in the training set. Extensive experiments demonstrate that SORN outperforms state-of-the-art methods on multiple real-world industrial datasets.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**; • **Computer systems organization** → **Cloud computing**.

KEYWORDS

Task slowdown detection, Time series, Unsupervised anomaly detection, AIOps

ACM Reference Format:
Feiyi Chen, Yingying Zhang, Lunting Fan, Yuxuan Liang, Guansong Pang, Qingsong Wen, and Shuiguang Deng. 2024. Cluster-Wide Task Slowdown Detection in Cloud System. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671936

1 INTRODUCTION

Slow task detection is a critical issue in cloud operations and maintenance, as it directly impacts user experience and can lead to significant penalties for service level agreement violations [39]. Most existing anomaly detection methods focus on detecting task slowdowns at the individual task level [23, 34, 44, 46]. However, with millions of tasks running concurrently [23, 49] in large-scale cloud computing clusters, these approaches become impractical and inefficient. Moreover, single-task slowdowns are common and may not indicate a cluster malfunction, given the random and dramatic fluctuations in task duration time within a virtual environment. To address these challenges, we pivot towards detecting slowdowns on a cluster-wide scale, which are more indicative of cluster malfunctions and can be identified without examining each individual task. Furthermore, unlike the random fluctuations observed in single-task duration time, the duration time of cluster-wide tasks exhibits more regular patterns, making slowdown detection more feasible.

**Corresponding authors

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08…$15.00
https://doi.org/10.1145/3637528.3671936

Particularly, we detect cluster-wide task slowdowns using the duration time distribution of a cluster, as illustrated in Fig. 1(a), in which for each time slot we partition the range of task duration time into intervals and calculate the proportion of tasks falling into each interval. This strategic shift not only significantly reduces the computational complexity of our algorithm, making it independent of the number of tasks, but also enhances the accuracy of cluster malfunction detection.

Nonetheless, the distribution of normal task duration time is not stable but varies over time. Hence, there arises a necessity to discern the patterns of distribution variation and differentiate routine slowdowns from anomalies. Among the various methods for extracting normal patterns, transformer-based methods stand out as one of the most powerful and effective unsupervised anomaly detection approaches, resulting in numerous distinguished methods [19, 37, 43, 44]. Despite the abundance of powerful neural networks available for normal pattern extraction, several challenges persist:

- *Compound periodicity*: The distribution of cluster-wide task duration time often exhibits compound periodic variation patterns. Since different tasks exhibit different periodicity, the periodicity of cluster-wide task duration time distribution is compound and complicated. For example, in Fig. 1(b), it shows periodicity on both a weekly and daily basis. As depicted in Fig.1(c), when integrating two periodicities with different amplitudes and frequencies into a unified representation, the attention mechanism shows subpar performance in reconstructing the subperiodicity with relatively low amplitude in the presence of compound periodicity.
- *Non-slowing exceptional fluctuations*: The temporal evolution of task duration time within the cluster manifests periodic characteristics on a global scale, interspersing with localized non-periodic exceptional fluctuations. Within these exceptional fluctuations, only a small fraction corresponds to cluster-wide slowdowns, while others are not the focus of our work (e.g., we are not concerned about exceptional task speedups). However, traditional anomaly detection methods can not reconstruct all of the exceptional fluctuations well and detect all of them as anomalies. To distinguish cluster-wide task slowdowns, it is imperative to accurately reconstruct other exceptional fluctuations while excluding the cluster-wide slowdowns.
- *Anomalies in the training set*: In consideration of the substantial costs linked to manually labeling anomalies, our methodology has been intentionally crafted to function in an unsupervised manner. Nevertheless, it is noteworthy that several unsupervised methods operate on the assumption that anomalies are infrequent within the training set, a premise that tends to be overly optimistic in practical scenarios.

Addressing these challenges is imperative for improving the detection accuracy of compound periodic time series and enhancing model robustness against anomaly contamination in the training set. Therefore, we propose SORN, which Skims Off the subperiodicity with different amplitudes layer by layer and selectively Reconstructs the Non-slowing fluctuations excluding the cluster-wide task slowdowns. It contains three innovative mechanisms to tackle the aforementioned three issues correspondingly: Skimming Attention, Neural Optimal Transport (OT), and Picky Loss.

Specifically, we first theoretically prove that the standard attention mechanism tends to allocate more attention to subperiods with higher amplitudes in compound periodic time series. This bias prevents it from effectively reconstructing subperiods with relatively low amplitudes. Building on this analysis, we introduce a skimming attention mechanism to capture the compound periodicity pattern, where we sequentially skim off subperiods from the original sequence in descending order of amplitudes and reconstruct iteratively from the remaining series. In this way, the subperiods with higher amplitudes are initially well reconstructed and skimmed off from the original time series. After that, the subperiods with low amplitudes in the original series become subperiods with relatively high amplitudes in the remaining series and can be better reconstructed.

Subsequently, we use a Neural OT module to adjust the reconstructed series of skimming attention, where we innovatively transform the traditional optimal transport problem into a neural network, and by intricately designing a transportation cost matrix, we can selectively reconstruct the non-slowing fluctuations.

Furthermore, to mitigate the negative effect of anomaly contamination in the training set, we design a novel picky loss function, which allocates different weights to time slots in the loss function according to their reliability.

Accordingly, this work presents several novel and distinctive contributions to the field of cluster-wide slow task detection:

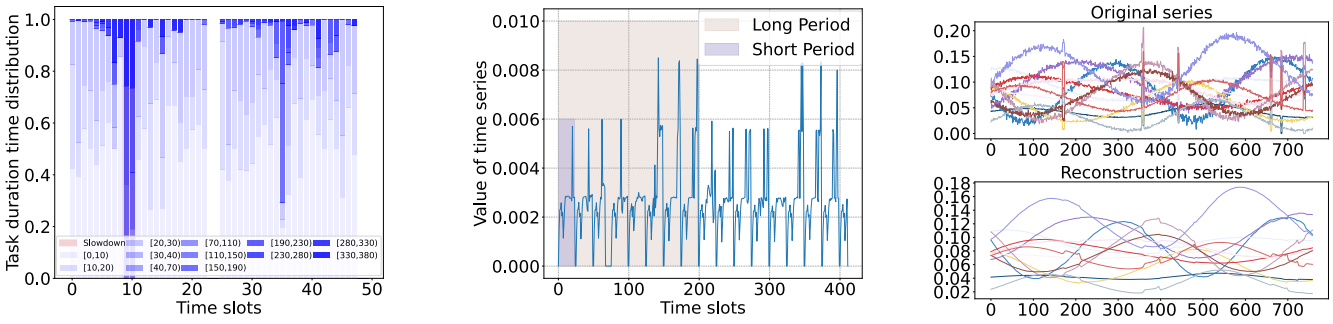
- We present the first attempt to formalize the cluster-wide slowdown problem with the identification of the problem specifications and relevant challenges.
- We provide a theoretical explanation for the limitations of the standard attention mechanism in effectively reconstructing subperiods with low amplitude in compound periodicity. Moreover, we introduce a novel skimming attention mechanism designed to extract subperiodic components with varying amplitudes and aggregate them to ensure accurate reconstruction of both high and low-amplitude subperiods.
- We introduce a novel Neural OT module tailored to reconstruct the normal non-periodic fluctuations observed in the duration time distribution, while effectively filtering out the cluster-wide slow-down anomalies.
- We propose a picky loss function that assigns higher weights to reliable time slots within the loss function.

Besides, we conducted extensive experiments and demonstrated that our method outperforms the state-of-the-art (SOTA) methods in F1 score on real-world industrial datasets.

2 PRELIMINARY

2.1 Optimal Transport (OT)

It is given a set of value intervals $I = \{(s_1, s_2], (s_2, s_3], \ldots, (s_{n-1}, s_n]\}$ and two distributions $\mathbf{a} \in R^N$ and $\mathbf{b} \in R^N$, where $\mathbf{a}_i = P(s_i < x \leq s_{i+1}), x \sim \mathbf{a}$. Similarly, $\mathbf{b}_i = P(s_i < x \leq s_{i+1}), x \sim \mathbf{b}$. The Optimal Transport problem aims at transforming distribution $\mathbf{a}$ to $\mathbf{b}$ by moving a fraction of the amount in each interval of $\mathbf{a}$ to another interval. Moving a unit from j^{th} interval to i^{th} interval costs a price $C_{i,j}$. The Optimal Transport problem gropes for an optimal transport strategy P costing the lowest price, where $P_{i,j}$ denotes the amount



(a) The task duration time distribution (b) The compound periodicity (c) The reconstruction series of attention

Figure 1: (a) At each time slot, we use a stacked histogram bar to plot the frequency distribution of the duration time at that slot. We use a darker color to denote the interval requiring more duration time. The stacked histogram bar is ordered in time order. (b) The compound periodicity of task duration time. (c) The original series and series reconstructed by standard attention are plotted in one figure, where the subperiod with low amplitude can not be well reconstructed.

of unit moving from j^{th} interval to the i^{th}, as shown in Eq.1, where $< P, C >$ denotes the Frobenius dot-product.

$$\min_{P} < P, C >, \quad s.t.\ P \cdot \vec{1} = \mathbf{b}, P^T \cdot \vec{1} = \mathbf{a}. \tag{1}$$

2.2 Problem Setup

Definition 1. f_t and f_t^* are used to denote the real-time distribution and expected distribution at time slot t. $f_t(\alpha)$ and $f_t^*(\alpha)$ are used to denote the α-quantile of distribution f_t and f_t^*. $\mathcal{T}$ is used to denote the threshold for tolerable fluctuation range of duration time distribution.

Definition 2. If there is a slowdown at time slot t, then $\max_\alpha f_t(\alpha) - f_t^*(\alpha) > \mathcal{T}, \forall \alpha$.

Definition 3. (Input data & output data) Given a set of intervals $I = \{[s_1, s_2), [s_2, s_3), \ldots, [s_D, s_{D+1})\}$, the input data is a $T-$length and D dimensional multivariate time series $x \in R^{T \times D}$, where $x[t, d]$ is the number of tasks whose duration time falls into the d^{th} interval $[s_d, s_{d+1})$. The reconstruction series of SORN is denoted by $\dot{\tilde{x}}$.

Problem Formalization. We use a SORN to obtain a reconstruction series $\dot{\tilde{x}}$ from the original input data x. Subsequently, we use an anomaly score function AnomalyScore$(\dot{\tilde{x}}, x, I)$. We aim to maximize the anomaly score gap between the slow-down time slots and the others.

3 METHODOLOGY

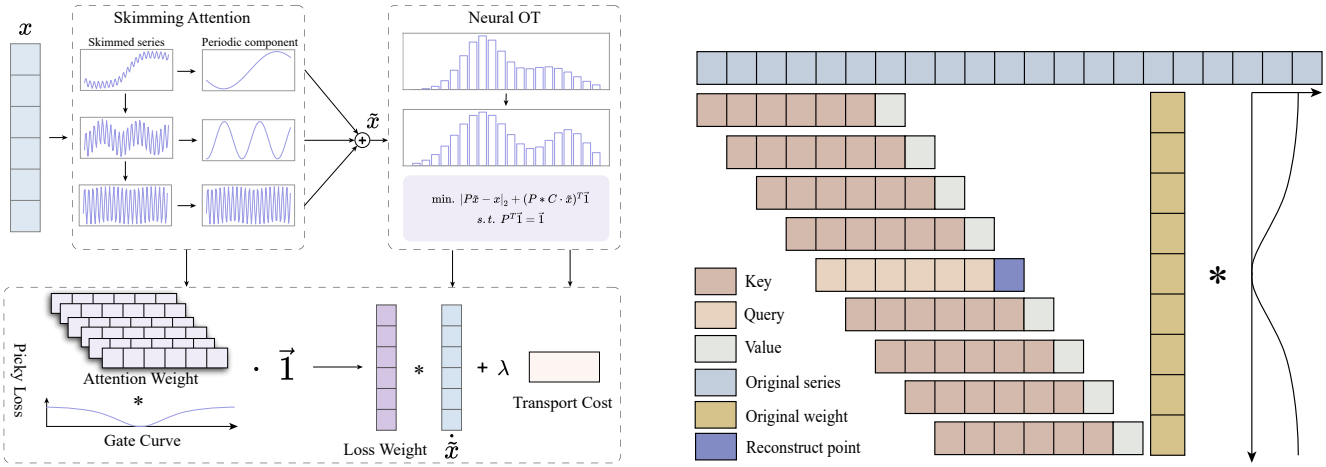
The overview of SORN is depicted in Fig. 2(a). We sequentially mask each time slot in x and employ a multi-layer Skimming Attention mechanism to reconstruct the time slot by leveraging compound periodic information. Subsequently, we utilize Neural OT to fine-tune the reconstructed series obtained from Skimming Attention, capturing aperiodic but typical fluctuations in the time series. Finally, we apply the picky loss function to assign higher weights to normal time slots while assigning lower weights to occasional anomalous slots in the loss function.

3.1 Skimming Attention

The duration time distribution usually exhibits compound periodic fluctuations, as shown in Fig. 1(b). In a compound periodic series, different subperiods usually have different amplitudes (i.e., variation range) [40], as shown in Fig. 3(a). When dealing with this kind of compound periodicity, the standard attention mechanism falls short in reconstructing the subperiod with low amplitude, as shown in Fig.1(c), where we fuse two periodicities with different amplitudes and frequency, the standard attention only reconstructs the one with higher amplitude well. We theoretically explain this phenomenon in Theorem 1 and Theorem 2, where we prove that a self-attention mechanism pays more attention to the subperiod with relatively higher amplitude in compound periodic series, which degrades the performance of reconstructing the subperiods with lower amplitudes in compound periodic series. Thus, we propose a skimming attention that masks each time slot alternatively and aims at reconstructing it by compound periodic information. There are two challenges to achieving this. On the one hand, we need to prevent it from reconstructing time slots only by leveraging the similarity of adjacent time slots in each layer but neglecting the periodic information. On the other hand, we need to reconstruct every subperiod well rather than just those with high amplitudes.

We deduce Theorems 1-2 using the same setting as the self-attention mechanism in a patching transformer [25], where a time series is split into a set of p-length patches, which constitute the query, key, and value vectors of a self-attention mechanism. We start with a simple case and generalize it to a general situation. In the derivation, we omit the final step of applying softmax to the attention weights, as softmax does not alter the relative order of the attention weights assigned to different time slots in the sequence and will not affect the conclusion.

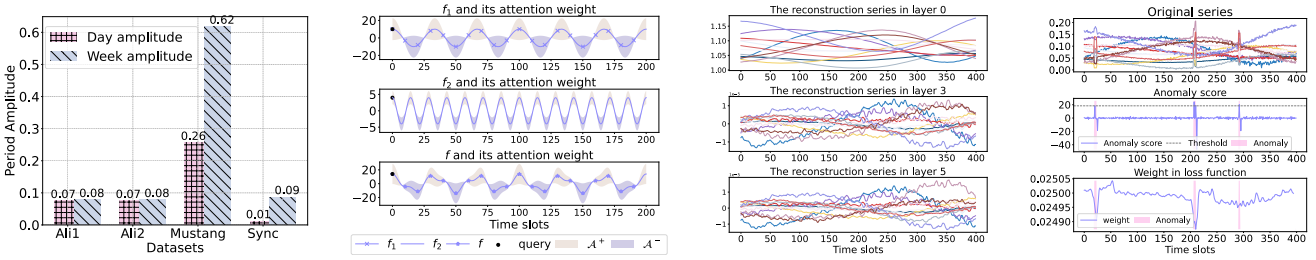
Definition 4. Given $a, b \in Z, a \neq b$, we set the patch length p to lcm(a, b), where lcm(a, b) denotes the least common multiple of a and b. It is given a series with compound periodicity $f(t) = c_1 \cos(\omega_1 t) + c_2 \sin(\omega_2 t)$, where $\omega_1 = \frac{2a\pi}{p}$, $\omega_2 = \frac{2b\pi}{p}$ and $c_1, c_2 \in R, c_1 > c_2$. There are two subperiod component in $f(t)$: $f_1(t) = c_1 \cos(\omega_1 t)$ and $f_2(t) = c_2 \sin(\omega_2 t)$. We use T_1 and T_2 to denote the



(a) The model architecture of SORN

(b) A layer of Skimming Attention

Figure 2: The model architecture of the proposed SORN algorithm.



(a) The amplitude of different periods

(b) Attention weights along different subperiods

(c) The series from different Skimming Attention layers

(d) The reconstruction and loss weight of SORN

Figure 3: (a) The figure shows different amplitudes of different subperiods; (b) The figure shows attention weight along different subperiods in $f(t)$. The width of the shadow is the value of the attention weight divided by 100 at the corresponding time slot. To distinguish the positive attention weight and negative attention weight we plot them in different colors and denote them by $\mathcal{A}^+$ and $\mathcal{A}^-$ respectively. (c) & (d) The visualization of SORN.

period length of f_1 and f_2 respectively.

Theorem 1. In $f(t)$, when taking the patch starting from t_1^{th} time slot as the query, the attention weight of the patch starting from t_2^{th} is $\frac{p}{2}[c_1^2 \cos \omega_1 \Delta t + c_2^2 \cos \omega_2 \Delta t]$, where $\Delta t = (t_2 - t_1)$.

Proof. Please refer to Appendix A for more details.

Taking a further look at the attention weight $\frac{p}{2}[c_1^2 \cos \omega_1 \Delta t + c_2^2 \cos \omega_2 \Delta t]$, it is a linear combination of $\cos \omega_1 \Delta t$ and $\cos \omega_2 \Delta t$. The first one distributes attention weight according to the periodicity of f_1: it assigns the highest attention weight to the time slot that is nT_1-slots apart from the query time slot, where $n \in Z$ (i.e. when $\Delta t = nT_1$, $\cos \omega_1 \Delta t$ reaches its maximum value). Similarly, the second one distributes attention weight according to the periodicity of f_2 and assigns the highest attention weight to the time slot that is nT_2 apart from the query time slot. Moreover, their impact on the attention weight is decided by the amplitudes of their corresponding subperiod. Since $c_1 > c_2$, $\cos \omega_1 \Delta t$ contributes more to the attention weight. Thus, the periodic information of f_1 can obtain higher attention weight and f_1 will be reconstructed better. As shown in Fig. 3(b), the highest attention weights show up at the time slot that nT_1-slots apart from the query slot without concerning the subperiod with period length of T_2.

To generalize Theorem 1 to a general situation, given a time series $\tilde{f}(t)$ with compound periodicity, we use trigonometric series to decompose it as defined in Definition 5.

Definition 5. Given a compound periodic time series $\tilde{f}(t)$ with period length p, we set the patch length to p. We decompose $\tilde{f}(t)$ to a linear combination of trigonometric series as: $\tilde{f}(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos \omega_n t + b_n \sin \omega_n t)$, where $\omega_n = \frac{2n\pi}{p}$ and a_n and b_n are coefficients for triangonometric series.

Theorem 2. In $\tilde{f}(t)$, when taking the patch starting from t_1^{th} time slot as the query, the attention weight of the patch starting from t_2^{th} is $\frac{a_0^2 p}{4} + \frac{p}{2}\sum_{n=1}^{\infty}(a_n^2 + b_n^2) \cos \omega_n \Delta t$, where $\Delta t = (t_2 - t_1)$.

Proof. Please refer to Appendix B for more details.

Similar to the analysis of $f(t)$, the attention weight of $\tilde{f}(t)$ is a linear combination of $\cos \omega_n \Delta t$. The subperiods with the higher amplitudes are more decisive to the attention weight distribution

and the periodic information of these subperiods can obtain higher attention weights. Thus, the subperiods with higher amplitudes are more likely to reconstruct better, while the subperiods with low amplitudes can be poorly reconstructed.

We show the architecture of each skimming attention layer in Fig.2(b), which aims at preventing the attention mechanism from directly reconstructing time slots by making use of the similarity of adjacent time slots. As shown in Fig.2(b), we first use a sliding window with padding to extend the input data $x \in R^{T\times D}$ to $\bar{x} \in R^{T\times(p+1)\times D}$, where $p+1$ denotes the window length of the sliding window. Subsequently, we take each dimension separately (taking the d^{th} dimension as an example) and use the first p-length series in each window as the keys and the final time slot in each window as the queries and values. This process is shown in Eq.2-Eq.5, where $[:p+1]$ denotes the time slices from beginning to the p^{th} one.

$$\bar{x} = \text{SlidingWindow}(x, p+1, 1), \tag{2}$$

$$q_d = \bar{x}[:, :p+1, d], \tag{3}$$

$$k_d = \bar{x}[:, :p+1, d], \tag{4}$$

$$v_d = \bar{x}[:, p+1, d]. \tag{5}$$

Afterward, as shown in Eq. 6, we apply a standard attention mechanism to the queries, keys, and values and obtain a set of attention weight $\mathcal{A} \in R^{T\times T}$, where $\mathcal{A}_{i,j}$ denotes the j^{th} attention weight for the i^{th} query. Then, in Eq. 7, we multiply a gate curve $G \in R^{T\times T}$ to the $\mathcal{A}$, where $G[i,j] = 1 - \exp^{-\frac{(i-j)^2}{\sigma^2}}$, σ is a learnable parameter and $*$ denotes an element-wise multiplication. In this way, the attention weights of time slots that are closer to the query are harder to pass through the gate, while the further one can easily get passed. Consequently, we can force the attention mechanism to put more weight on the hopping time slots. Finally, we obtain the reconstruction series in this layer as in Eq. 8, where $\tilde{x}_l$ denotes the reconstruction series of the l^{th} skimming attention layer:

$$\mathcal{A} = q_d k_d^T, \tag{6}$$

$$\tilde{\mathcal{A}} = \text{softmax}(\mathcal{A} * G), \tag{7}$$

$$\tilde{x}_l[:, d] = \tilde{\mathcal{A}} v_d. \tag{8}$$

We organize different layers of skimming attention as follows to deal with compound periodic information:

$$\begin{aligned} \tilde{x}_l &= \text{Skimming AttentionLayer}(x_l), \\ x_{l+1} &= x_l - \tilde{x}_l, \end{aligned} \tag{9}$$

where x_l is the l^{th} layer input data and $x_0 = x$. There are two benefits to organizing the skimming attention layers in this way. On the one hand, each skimming attention layer skims off the subperiod with the highest amplitude in x_l and the next layer can pay more attention to the subperiod with relatively low amplitude in the remaining series. Consequently, the subperiods with different amplitudes can be reconstructed well. We show the reconstruction series of different Skimming Attention layers in Fig. 3(c), where it reconstructs subperiods in descending amplitude order. On the other hand, it can also prevent the problem of vanishing gradient like ResNet does, since the input of every layer can be also reduced to $x_l = x - \sum_{k=0}^{l-1} \tilde{x}_l$.

3.2 Neural OT

Besides the periodic patterns, there are still aperiodic but normal fluctuations in task duration time distribution. Since we only pay attention to slow-down anomalies but not others (e.g., the execution speed of homework has significantly increased), we target modeling these non-periodic fluctuations but only hinder the reconstruction of slow-down anomalies. Inspired by the Optimal Transport (OT) algorithm, we transform a standard OT problem into a neural network and embed it into our model so that our model becomes end-to-end.

We first establish an OT problem and then transform it into a neural network. For each time slot t, we take its reconstruction duration time distribution $\tilde{x}[t] \in R^{1\times d}$ as a source distribution and take its original duration time distribution $x[t] \in R^{1\times d}$ as a target distribution. The OT problem gropes for a transport strategy P to transform the source distribution to the target distribution with a minimum cost $< P * \tilde{x}[t], C >$, where $P[d,s]$ denotes the ratio of $\tilde{x}[t,s]$ transporting to $\tilde{x}[t,d]$, $C[d,s]$ denotes the cost of transporting a unit from $\tilde{x}[t,s]$ to $\tilde{x}[t,d]$ and $*$ denotes element-wise multiplication. According to the definition of P, $P\tilde{x}[t]$ denotes the distribution after applying the transport strategy P to the source distribution $\tilde{x}[t]$, which should approach the target distribution $x[t]$, and the sum of each column of P should be 1. Thus, we formulate $|P\tilde{x}[t] - x[t]|$ as an optimization goal and the $P^T\vec{1} = \vec{1}$ as a constraint in our OT problem. To reconstruct anomalies except the slow ones, we set C as follows:

$$C_{i,j} = \begin{cases} M[i] - M[j], & i > j \\ 0, & else, \end{cases} \tag{10}$$

where $M[i]$ is the midpoint of i^{th} interval in I (I is defined in Definition 3). In this way, only the slow-down distribution shift is penalized by the transporting cost. Based on the setting above, we formulate an OT problem as:

$$\begin{aligned} \min . \lambda < P * \tilde{x}[t], C > + |P\tilde{x}[t] - x[t]|_2, \\ s.t. P^T\vec{1} = \vec{1}, \end{aligned} \tag{11}$$

where λ is a hyperparameter belonging to $[0,1]$.

Furthermore, we transform it into a neural network. We take P as a trainable parameter. To meet its constraint in the OT problem, we manipulate P as $\text{softmax}(P^T)^T$, and the neural layer is specified as:

$$\dot{\tilde{x}} = \text{softmax}(P^T)^T \tilde{x} \tag{12}$$

Besides, we also introduce the optimization objective of the OT problem to the loss function.

3.3 Picky Loss Function

The reconstruction-based methods assume that there are no anomalies in the training set. However, it is inevitable to have some anomalies in the training set in the scenario of unsupervised learning. Thus, we propose a picky loss function, which adaptively attributes a weight $\mathcal{W} \in R^T$ according to trustworthiness to the loss of each time slot. The more trustful a time slot is, the higher its weight is. Inspiring by AnomalyTransformer [43], which points out that the normal points can establish wide-broad informative association along the whole series in attention mechanism whereas the anomalies can only concentrate on adjacent time slots, we utilize

the attention weight $\mathcal{A}$ in subsection. 3.1 to obtain the weight $\mathcal{W}$. We use a trainable gate curve $\hat{G} \in R^{T\times T}$ to filter out the attention weight in the adjacent part, where $\hat{G}[i,j] = 1 - \exp^{-\frac{(i-j)^2}{\hat{\sigma}^2}}$ and $\hat{\sigma}$ is a trainable parameter and obtain $\mathcal{W}$ via:

$$\mathcal{W} = \text{softmax}[(\mathcal{A} * \hat{G})\vec{1}]. \tag{13}$$

We obtain the final loss function by attributing the weight $\mathcal{W}$ to each time slot and fusing the optimizing objective in Section 3.2, resulting in the final loss function as follows:

$$\mathcal{L} = \sum_{t=0}^{T} \mathcal{W}[t](\left|\dot{\tilde{x}}[t] - x[t]\right|_2 + \lambda \langle P * \tilde{x}[t], C\rangle). \tag{14}$$

As shown in Fig. 3(d), the picky loss function renders lower weights to the anomaly time slots.

3.4 Anomaly Score

Since the duration time distribution of different tasks does not distribute uniformly, we split the distribution intervals I according to the distribution density of task duration time. This leads to the heterogeneous importance of the reconstruction errors for different intervals. However, the trivial anomaly score, which adds the reconstruction errors for different intervals together, ignores this heterogeneity. Thus, we use the difference between the task duration time expectations of the original distribution and reconstruction one as the anomaly score:

$$\begin{aligned}\text{AnomalyScore}[t] &= \mathbb{E}(\bar{T}(x[t])) - \mathbb{E}(\bar{T}(\dot{\tilde{x}}[t])) \\ &= \sum_{d=0}^{D} x[t,d] * M[d] - \sum_{d=0}^{D} \dot{\tilde{x}}[t,d] * M[d],\end{aligned} \tag{15}$$

where $\text{AnomalyScore}[t]$ denotes the anomaly score of t^{th} time slot, and $\bar{T}(x[t])$ and $\bar{T}(\dot{\tilde{x}}[t])$ denote two variables: the task duration time from distributions $x[t]$ and $\dot{\tilde{x}}[t]$ respectively.

4 EXPERIMENT

We have made extensive experiments on four datasets to verify the following conclusions:

- SORN can achieve the best performances on the four datasets, compared with the SOTA methods.
- SORN consumes tolerable time and memory overhead.
- SORN is parameter insensitive.
- SORN is resistant to noise and lax periodicity.
- Each module in SORN has contributed to the performance.

4.1 Experiment Setup

Baseline Methods. We compare SORN with the SOTA anomaly detection methods: DCdetector [44], TranAD [37], AnomalyTransformer [43], VQRAE [17], OmniAnomaly [34], MSCRED [46]. Furthermore, we compare SORN with a method specifically designed for slow-down detection: IASO [26] and a method designed for distribution shift detection, feature-shift detection [18].

Datasets. We perform our experiments on four datasets. Two of them (Ali1, Ali2) are monitoring data of industrial cloud clusters from Alibaba. One of them (Mustang) is disclosed by Carnegie Mellon Parallel Data Laboratory, and we label the slow-down anomalies in it manually. To further verify the impact of different factors on the performance, such as noise, periodicity, slow tasks ratio, and the average task slow-down time, we also introduce a synthetic dataset (Sync) so that we can keep every factor under control. We summarize key statistics of different datasets in Table 1.

Table 1: Statistics of different datasets.

	Ali1	Ali2	Mustang	Sync
Dimension	14	14	17	14
Anomaly ratio (%)	3.71	6.06	7.75	1
Subsets	25	25	1	10

Table 2: The hyperparameters.

Hyperparameter	Value	Hyperparameter	Value
Batch Size	100	Learning Rate	0.001
Skimming Layer of Ali1	10	Patch Size of Ali1	2
Skimming Layer of Ali2	6	Patch Size of Ali2	2
Skimming Layer of Mut	6	Patch Size of Mut	2
Skimming Layer of Sync	6	Patch Size of Sync	10

Besides, different datasets exhibit different periodicity strictness. To measure the periodicity strictness of each dataset, for each subset, if it exhibits periodicity we calculate its autocorrelation coefficient at intervals of its period length as its periodicity strictness level, otherwise, we set its periodicity strictness as 0. We show the periodicity strictness level distribution of subsets in each dataset in Fig. 4(a). Ali1 and Ali2 show relatively strict periodicity. Mustang shows lax periodicity. Some subsets of Sync show strict periodicity, while others are aperiodic.

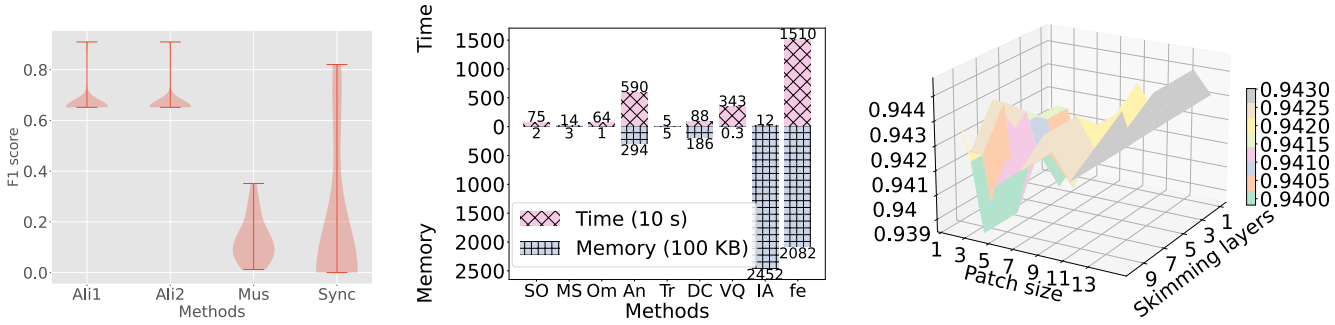
For more data preprocessing details, please refer to Appendix. C.

Hyperparameters. We show some important hyperparameters in Table 2, where we use Mut to stand for Mustang.

Evaluation Metrics. We choose three of the most widely-used metrics to measure the performance of our method as many marvelous methods did [7, 9, 19]: the precision, recall, and F1 score.

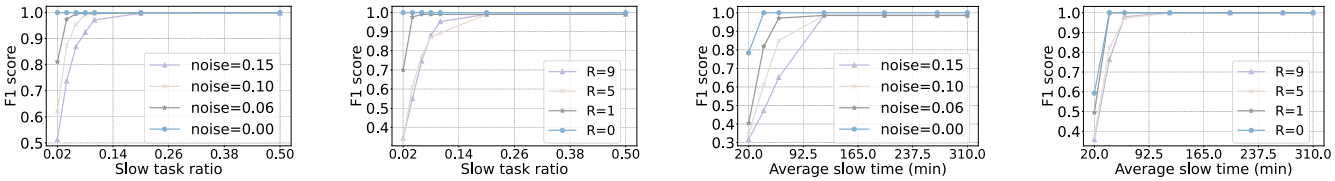
4.2 Prediction Accuracy

We take 70% of each subset as the training set and take the remaining 30% as the testing set. For each subset, we train a unique model. This training strategy is also adopted by other marvelous works, such as [7, 34, 46]. We show the performance of SORN and baselines in Tab. 3, where we use "Pre" and "Rec" to stand for precision and recall respectively, and highlight the best performance as the boldfaced. When SORN achieves the best performance, we underline the best performance among baselines. Otherwise, we underline the second-best performance among all methods. SORN achieves the best F1 scores on all datasets compared with the state-of-the-art methods. Comparing the performance of our method on four datasets, we observe that it performs best on the Ali1 and Ali2 datasets, followed by Mustang and Sync. It can be seen that the effectiveness of our method is positively correlated to the strictness of periodicity in the datasets. It achieves impressive performance on datasets with strict periodicity, while also demonstrating competitive results on datasets with relaxed periodicity or non-periodic



(a) Periodicity strictness of each dataset (b) Time and memory overhead (c) Parameter sensitivity

Figure 4: (a) We show the autocorrelation coefficient distribution at the interval of period length for subsets in every dataset. (b) The time and memory overhead of SORN and baselines on Sync dataset. We use the first two characters to stand for each method; (c) The hyperparameter sensitivity of the number of skimming layers and patch size on Sync dataset.



(a) The impact of noise and slow task ratio on performance (b) The impact of periodicity and slow task ratio on performance (c) The impact of noise and average slow-down time on performance (d) The impact of periodicity and average slow-down time on performance

Figure 5: (a) We add noise to the original synthetic time series, whose standard deviation is the maximum amplitude of the original time series multiplied by the "noise" shown in the legend. Then, we test the performance of SORN for different slow task ratios. (b) For each period in a periodic time series, we extend it by a scaler which is randomly sampled from $(1, 1+R]$. In this way, the original time series will have a lax periodicity. Then, we test the performance of SORN for different slow task ratios. (c) Using the same noise setting as (a), we test the performance of SORN for different average slow-down time. (d) Using the same period setting as (b), we test the performance of SORN for different average slow-down time.

characteristics. We will further discuss the impact of periodicity strictness in subsection. 4.5.

4.3 Time and Memory Overhead

We evaluated both time and memory overhead on a server equipped with a configuration comprising 32 Intel(R) Xeon(R) CPU E5-2620 @ 2.10GHz CPUs and 2 K80 GPUs. We use the checkpoint sizes to stand for the neural network memory overhead and use the time of training model for one epoch to stand for the time overhead. As for the non-neural network methods, IASO and feature-shift detection, we use the maximum memory consumption during its inferring process as its memory overhead. We show the time and memory overhead of different methods in Fig. 4(b), where SORN only introduces marginal time and memory overhead compared with some light methods, such as OmniAnomaly, but can achieve better performance on all the datasets. Compared with some transformer-based methods, such as AnomalyTransformer and DCdetector, we use less memory overhead yet achieve better accuracy. In this way, SORN can better meet the real-time requirements of the cloud center.

4.4 Hyperparameter Sensitivity

We test the performance of SORN when setting the number of skimming layers and patch size as the Cartesian product of {1,3,5,7,9} for skimming layers and {1,3,5,7,9,11,13} for patch size. We exhibit the result in Fig. 4(c). Overall, the performance of SORN is parameter insensitive. As the number of skimming attention layers and patch size increase, the performance of SORN increases in fluctuations.

4.5 The Impact of Dataset Property

We investigate the impact of four factors on the performance of SORN on the Sync dataset: the noise, periodicity strictness, slow task ratio, and average slow-down time in slow-down anomalies. The noise introduced into the Sync data is a random variable with a mean of 0 and standard deviation of $noise * \mathcal{A}$, where $\mathcal{A}$ is the amplitude of the original time series. To manipulate the periodicity strictness, we distort each period of the original series by using a scalar randomly sampled from a distribution $(1, 1+R]$ to extend it. When we test the impact of the noise, we make the time series strictly periodic before introducing noise and vice versa. The results are displayed in Fig. 5(a)-Fig. 5(d). Generally, when the time series is strictly periodic without any noise, SORN can achieve excellent performance on the Sync dataset. When the noise becomes more variable and the periodicity is more severely distorted, the performance degrades but SORN is still sensitive and accurate: SORN can achieve an F1 score over 0.9 as long as the slow task ratio overpasses 10% in all conditions of the noise and periodicity strictness explored in our experiment; SORN can achieve an F1 score over 0.9 as long as the average slow-down time overpasses 60 minutes in

Table 3: Average performance of SORN and baselines on subsets of four datasets.

	Ali1			Ali2			Mustang			Sync		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
MSCRED	0.841	0.981	0.878	0.928	0.988	0.954	0.871	0.960	0.896	0.717	0.874	0.779
Omni	0.681	0.981	0.782	0.814	0.987	0.890	0.812	0.968	0.878	0.655	**0.997**	0.787
AnomalyTr	**1.000**	0.870	0.923	**0.999**	0.763	0.857	**1.000**	0.891	0.935	**1.000**	0.680	0.809
TranAD	0.784	0.989	0.864	0.827	0.968	0.877	0.865	0.918	0.867	0.247	0.568	0.313
DCdetector	0.984	0.728	0.806	0.994	0.723	0.818	0.968	0.718	0.799	0.936	0.406	0.567
VRGAE	0.811	0.981	0.853	0.966	0.992	0.978	0.871	0.959	0.905	0.624	0.794	0.648
IASO	0.492	0.943	0.618	0.611	0.907	0.708	0.420	0.899	0.524	0.389	0.910	0.533
feature-shift	0.533	**1.000**	0.647	0.744	0.953	0.790	0.511	**1.000**	0.629	0.594	0.081	0.142
SORN†	0.891	0.989	0.897	0.955	0.997	0.968	0.895	0.996	0.916	0.963	0.893	0.919
SORN‡	0.944	0.969	0.939	0.960	0.967	0.955	0.912	0.971	0.919	0.939	0.832	0.874
SORN§	0.878	**1.000**	0.891	0.950	0.997	0.965	0.925	0.996	0.938	0.935	0.763	0.826
SORN	**1.000**	0.966	**0.979**	0.980	**1.000**	**0.989**	0.952	0.974	**0.958**	0.956	0.926	**0.932**

all conditions of the periodicity strictness and most of conditions of the noise. It is worth noting that 60 minutes is slightly over the maximum interval length in I (50 minutes). Since the maximum interval length is 50 minutes, the slow task with slow-down time less than that may not bring change to x. Thus, our model can not distinguish them. If there is a need to improve the sensitivity of SORN to the average slow-down time, we can make it by just substituting the interval division I with a fine-grained one.

4.6 Ablation Study

To evaluate the contribution of each module in SORN, we alternatively remove each submodule and test the performance of the remaining model. Specifically, we denote SORN removing skimming attention as SORN†, denote SORN removing neural OT as SORN‡ and denote SORN replacing picky loss with MSE as SORN§. When removing the skimming attention mechanism, we replace it with a standard attention. When removing the picky loss, we substitute it with MSE. As shown in Table 3, the completed SORN achieves the best performance. Thus, each submodule of SORN does contribute to the performance.

5 RELATED WORK

To the best of our knowledge, we are the first to investigate the issue of cluster-wide task slowdowns. While numerous works delve into slow query detection [22, 51] and disk fail-slow detection [20, 21], they primarily focus on detecting slowdowns at the level of individual SQL queries or disks rather than considering the overall aspect. However, detecting slow tasks at the individual level can be unreliable in cloud virtual environments, where task duration time fluctuates randomly and significantly. Single-task slowdowns are common and do not necessarily indicate a cluster malfunction.

Moreover, time series anomaly detection is another relevant area, as we need to capture the normal variation pattern and time dependencies of time series [16, 48]. Time series anomaly detection methods can be broadly categorized into three classes: classical methods [3, 11, 24, 27], signal-processing-based methods [1, 23, 50], and deep learning-based methods [6, 14, 30, 35, 41, 42, 47, 52]. Classical methods typically rely on statistical approaches and have relatively low computational overhead. However, they often make specific assumptions that limit their robustness in detecting anomalies in cloud environments [23]. Signal-processing-based methods leverage the sparsity inherent in the frequency domain to reduce computational overhead. However, they may overlook local subtle features [1] or struggle to handle heavy traffic loads in real-time scenarios [23]. Deep learning-based anomaly detection methods have reported promising performance and diversified into various approaches, including prediction-based [5, 14, 30, 52], reconstruction-based [6, 8, 10, 13, 15, 32, 36, 45], classification-based [12, 29, 31, 35, 42], and perturbation-based methods [4, 33]. Among them, reconstruction-based methods have shown strong advantages over others [17], in which the transformer-based methods have demonstrated good performance recently [28, 38, 43]. However, as we mentioned earlier, the standard attention mechanism may struggle to reconstruct compound periodic time series effectively.

6 CONCLUSION

In this study, we introduce SORN as a method for detecting cluster-wide task slowdowns in cloud clusters, offering three distinctive features: 1) Skimming Attention, where we provide a theoretical explanation for the limitations of standard attention mechanisms in reconstructing compound periodicity and propose a method to separately reconstruct subperiodic components to ensure accurate reconstruction of both high and low amplitude subperiods; 2) Neural OT, which selectively reconstructs non-slowing exceptional fluctuations; 3) Picky Loss, which assigns weights to time slots in the loss function based on their reliability. Additionally, extensive experiments demonstrate that SORN outperforms state-of-the-art methods in real-world datasets. In the future, we will use large language models for further analysis of the causes of slow-down tasks based on this foundation and employ multi-agent systems for automatic recovery.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation of China under Grants 62125206 and U20A20173, and in part by Alibaba Group through Alibaba Research Intern Program.

REFERENCES

[1] Vicente Alarcon-Aquino and Javier A Barria. 2001. Anomaly detection in communication networks using wavelets. *IEEE Proceedings-Communications* 148, 6 (2001), 355–362.

[2] George Amvrosiadis, Jun Woo Park, Gregory R Ganger, Garth A Gibson, Elisabeth Baseman, and Nathan DeBardeleben. 2018. On the diversity of cluster workloads and its impact on research results. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. 533–546.

[3] Björn Barz, Erik Rodner, Yanira Guanche Garcia, and Joachim Denzler. 2018. Detecting regions of maximal divergence for spatio-temporal anomaly detection. *IEEE transactions on pattern analysis and machine intelligence* 41, 5 (2018), 1088–1101.

[4] Jinyu Cai and Jicong Fan. 2022. Perturbation learning based anomaly detection. *Advances in Neural Information Processing Systems* 35 (2022).

[5] Chengwei Chen, Yuan Xie, Shaohui Lin, Angela Yao, Guannan Jiang, Wei Zhang, Yanyun Qu, Ruizhi Qiao, Bo Ren, and Lizhuang Ma. 2022. Comprehensive regularization in a bi-directional predictive network for video anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 230–238.

[6] Feiyi Chen, Zhen Qin, Mengchu Zhou, Yingying Zhang, Shuiguang Deng, Lunting Fan, Guansong Pang, and Qingsong Wen. 2024. LARA: A Light and Anti-overfitting Retraining Approach for Unsupervised Time Series Anomaly Detection. In *Proceedings of the ACM on Web Conference 2024*. 4138–4149.

[7] Wenchao Chen, Long Tian, Bo Chen, Liang Dai, Zhibin Duan, and Mingyuan Zhou. 2022. Deep variational graph convolutional recurrent network for multivariate time series anomaly detection. In *International Conference on Machine Learning*. PMLR, 3621–3633.

[8] Wenchao Chen, Long Tian, Bo Chen, Liang Dai, Zhibin Duan, and Mingyuan Zhou. 2022. Deep Variational Graph Convolutional Recurrent Network for Multivariate Time Series Anomaly Detection. In *International Conference on Machine Learning, ICML 2022 (Proceedings of Machine Learning Research, Vol. 162)*. 3621–3633.

[9] Xuanhao Chen, Liwei Deng, Feiteng Huang, Chengwei Zhang, Zongquan Zhang, Yan Zhao, and Kai Zheng. 2021. Daemon: Unsupervised anomaly detection and interpretation for multivariate time series. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2225–2230.

[10] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4027–4035.

[11] Jingkun Gao, Xiaomin Song, Qingsong Wen, Pichao Wang, Liang Sun, and Huan Xu. 2020. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. *arXiv preprint arXiv:2002.09545* (2020).

[12] Will Grathwohl, Kuan-Chieh Wang, Jorn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. 2020. Your classifier is secretly an energy based model and you should treat it like one. In *8th International Conference on Learning Representations, ICLR 2020*.

[13] Thi Kieu Khanh Ho and Narges Armanfard. 2023. Self-supervised learning for anomalous channel detection in EEG graphs: application to seizure analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 7866–7874.

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

[15] Xi Jiang, Jianlin Liu, Jinbao Wang, Qiang Nie, Kai Wu, Yong Liu, Chengjie Wang, and Feng Zheng. 2022. Softpatch: Unsupervised anomaly detection with noisy data. *Advances in Neural Information Processing Systems* 35 (2022), 15433–15445.

[16] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2024. Time-llm: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*.

[17] Tung Kieu, Bin Yang, Chenjuan Guo, Razvan-Gabriel Cirstea, Yan Zhao, Yale Song, and Christian S Jensen. 2022. Anomaly detection in time series with robust variational quasi-recurrent autoencoders. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1342–1354.

[18] Sean Kulinski, Saurabh Bagchi, and David I Inouye. 2020. Feature shift detection: Localizing which features have shifted via conditional distribution tests. *Advances in neural information processing systems* 33 (2020), 19523–19533.

[19] Yuxin Li, Wenchao Chen, Bo Chen, Dongsheng Wang, Long Tian, and Mingyuan Zhou. 2023. Prototype-oriented unsupervised anomaly detection for multivariate time series. In *International Conference on Machine Learning*. PMLR, 19407–19424.

[20] Ruiming Lu, Erci Xu, Yiming Zhang, Fengyi Zhu, Zhaosheng Zhu, Mengtian Wang, Zongpeng Zhu, Guangtao Xue, Jiwu Shu, Minglu Li, et al. 2023. Perseus: A Fail-Slow Detection Framework for Cloud Storage Systems. In *21st USENIX Conference on File and Storage Technologies (FAST 23)*. 49–64.

[21] Ruiming Lu, Erci Xu, Yiming Zhang, Zhaosheng Zhu, Mengtian Wang, Zongpeng Zhu, Guangtao Xue, Minglu Li, and Jiesheng Wu. 2022. NVMeSSD failures in the field: the Fail-Stop and the Fail-Slow. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*. 1005–1020.

[22] Minghua Ma, Zheng Yin, Shenglin Zhang, Sheng Wang, Christopher Zheng, Xinhao Jiang, Hanwen Hu, Cheng Luo, Yilin Li, Nengjun Qiu, et al. 2020. Diagnosing root causes of intermittent slow queries in cloud databases. *Proceedings of the VLDB Endowment* 13, 8 (2020), 1176–1189.

[23] Minghua Ma, Shenglin Zhang, Junjie Chen, Jim Xu, Haozhe Li, Yongliang Lin, Xiaohui Nie, Bo Zhou, Yong Wang, and Dan Pei. 2021. Jump-Starting multivariate time series anomaly detection for online service systems. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*. 413–426.

[24] Takaaki Nakamura, Makoto Imamura, Ryan Mercer, and Eamonn Keogh. 2020. Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives. In *2020 IEEE international conference on data mining (ICDM)*. IEEE, 1190–1195.

[25] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023*.

[26] Biswaranjan Panda, Deepthi Srinivasan, Huan Ke, Karan Gupta, Vinayak Khot, and Haryadi S Gunawi. 2019. IASO: A Fail-Slow Detection and Mitigation Framework for Distributed Storage Services. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*. 47–62.

[27] Guansong Pang, Kai Ming Ting, and David Albrecht. 2015. LeSiNN: Detecting anomalies by identifying least similar nearest neighbours. In *2015 IEEE international conference on data mining workshop (ICDMW)*. IEEE, 623–630.

[28] İlkay Yıldız Potter, George Zerveas, Carsten Eickhoff, and Dominique Duncan. 2022. Unsupervised Multivariate Time-Series Transformers for Seizure Identification on EEG. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 1304–1311.

[29] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert A. Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep One-Class Classification. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018 (Proceedings of Machine Learning Research, Vol. 80)*. 4390–4399.

[30] Lena Sasal, Tanujit Chakraborty, and Abdenour Hadid. 2022. W-Transformers: A Wavelet-based Transformer Framework for Univariate Time Series Forecasting. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 671–676.

[31] Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems* 33 (2020), 13016–13026.

[32] Lifeng Shen, Zhongzhong Yu, Qianli Ma, and James T Kwok. 2021. Time series anomaly detection with multiresolution ensemble decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9567–9575.

[33] Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. 2021. Graph posterior network: Bayesian predictive uncertainty for node classification. *Advances in Neural Information Processing Systems* 34 (2021), 18033–18048.

[34] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2828–2837.

[35] Yuting Sun, Guansong Pang, Guanhua Ye, Tong Chen, Xia Hu, and Hongzhi Yin. 2023. Unraveling the Anomaly in Time Series Anomaly Detection: A Self-supervised Tri-domain Solution. *arXiv preprint arXiv:2311.11235* (2023).

[36] Kai Tian, Shuigeng Zhou, Jianping Fan, and Jihong Guan. 2019. Learning competitive and discriminative reconstructions for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5167–5174.

[37] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. 2022. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284* (2022).

[38] Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. 2022. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proc. VLDB Endow.* 15, 6 (2022), 1201–1214.

[39] Utsav Upadhyay and Geeta Sikka. 2020. STDADS: an efficient slow task detection algorithm for deadline schedulers. *Big Data* 8, 1 (2020), 62–69.

[40] Qingsong Wen, Zhe Zhang, Yan Li, and Liang Sun. 2020. Fast RobustSTL: Efficient and robust seasonal-trend decomposition for time series with complex patterns. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2203–2213.

[41] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. 2023. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[42] Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang, and Guansong Pang. 2024. Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[43] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

[44] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. 2023. DCdetector: Dual Attention Contrastive Representation Learning for Time Series Anomaly Detection. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2203–2213.

[45] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. 2022. A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems* 35 (2022), 4571–4584.

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

[47] Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. 2022. TFAD: A decomposition time series anomaly detection architecture with time-frequency analysis. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2497–2507.

[48] Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. 2023. Self-Supervised Learning for Time Series Analysis: Taxonomy, Progress, and Prospects. *arXiv preprint arXiv:2306.10125* (2023).

[49] Yingying Zhang, Zhengxiong Guan, Huajie Qian, Leili Xu, Hengbo Liu, Qingsong Wen, Liang Sun, Junwei Jiang, Lunting Fan, and Min Ke. 2021. CloudRCA: A root cause analysis framework for cloud computing platforms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4373–4382.

[50] Nengwen Zhao, Jing Zhu, Yao Wang, Minghua Ma, Wenchi Zhang, Dapeng Liu, Ming Zhang, and Dan Pei. 2019. Automatic and generic periodicity adaptation for kpi anomaly detection. *IEEE Transactions on Network and Service Management* 16, 3 (2019), 1170–1183.

[51] Xuanhe Zhou, Lianyuan Jin, Ji Sun, Xinyang Zhao, Xiang Yu, Jianhua Feng, Shifu Li, Tianqing Wang, Kun Li, and Luyang Liu. 2021. Dbmind: A self-driving platform in opengauss. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2743–2746.

[52] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae-ki Cho, and Haifeng Chen. 2018. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In *6th International Conference on Learning Representations, ICLR 2018*.

A PROOF OF THEOREM 1

In the following, we use AttentionWeight$[\mathrm{t}_1, \mathrm{t}_2]$ to denote the attention weight of the patch starting from t_2^{th} time slot, when using the patch starting from t_1^{th} time slot as the query. We use the orthogonality of trigonometric functions when deriving Eq. 16 to Eq.17. Since $\cos\omega_1 t \cos\omega_1(t+\Delta t) = \frac{1}{2}\cos(\omega_1 t+\omega_1(t+\Delta t))+\cos(\omega_1 t-\omega_1(t+\Delta t))$, $\sin\omega_2 t \sin\omega_2(t+\Delta t) = -\frac{1}{2}(\cos(\omega_2 t+\omega_2(t+\Delta t))-\cos(\omega_2 t-\omega_2(t+\Delta t)))$, and $\int_{t_1}^{t_1+p}\cos(2\omega_1 t+\omega_1\Delta t)\,dt = 0$ (because p is integer multiple of the period length of $\cos(2\omega_1 t+\omega_1\Delta t)$), we derive Eq. 17 to Eq. 18. Since Δt is a constant without relevance to t, we derive Eq. 18 to Eq. 19.

$$\begin{aligned}
\text{AttentionWeight}[\mathrm{t}_1, \mathrm{t}_2] &= \int_{t_1}^{t_1+p} (c_1\cos\omega_1 t + c_2\sin\omega_2 t)[c_1\cos\omega_1(t+\Delta t) + c_2\sin\omega_2(t+\Delta t)]\,dt && (16)\\
&= \int_{t_1}^{t_1+p} c_1^2\cos(\omega_1 t)\cos\omega_1(t+\Delta t) + c_2^2\sin(\omega_2 t)\sin\omega_2(t+\Delta t)\,dt && (17)\\
&= \int_{t_1}^{t_1+p} \frac{1}{2}c_1^2\cos(\omega_1\Delta t) + \frac{1}{2}c_2^2\cos(\omega_2\Delta t)\,dt && (18)\\
&= \frac{p}{2}(c_1^2\cos\omega_1\Delta t + c_2^2\cos\omega_2\Delta t) && (19)
\end{aligned}$$

B PROOF OF THEOREM 2

We prove Theorem 2 in a similar way as in Theorem 1.

$$\begin{aligned}
\text{AttentionWeight}[t_1, t_2] &= \int_{t_1}^{t_1+p} (\frac{a_0}{2} + \sum_{n=0}^{\infty} a_n\cos\omega_n t + b_n\sin\omega_n t)\cdot[\frac{a_0}{2} + \sum_{n=0}^{\infty} a_n\cos\omega_n(t+\Delta t) + b_n\sin\omega_n(t+\Delta t)]\,dt\\
&= \frac{a_0^2 p}{4} + \sum_{n=0}^{\infty}\int_{t_1}^{t_1+p} a_n^2\cos\omega_n t\cos\omega_n(t+\Delta t) + b_n^2\sin\omega_n t\sin\omega_n(t+\Delta t)\,dt && (20)\\
&= \frac{a_0^2 p}{4} + \frac{p}{2}\sum_{n=0}^{\infty}(a_n^2+b_n^2)\cos\omega_n\Delta t
\end{aligned}$$

C DATA PREPROCESSING

The code and some datasets are available at https://github.com/gyhswtxnc/SORN.

- **Ali1 & Ali2** (periodic): We collect these datasets by tracing 25 industrial cloud clusters from Alibaba for 15 days. Most of the labels in these two datasets are assigned manually according to the experience of our engineers. Some of the labels are assigned according to our customer's feedback. These two datasets were collected on server clusters in different regions, and there is a significant difference in the anomaly proportion between them. Each subset in Ali1 and Ali2 stands for a cluster.
- **Mustang** (lax periodic) [2]: Mustang is a dataset that records task duration time for 5 years. We preprocess the original dataset as shown in Appendix. C and label the slow-down anomalies manually. Then, we equally divide the five years of tracing data into 35 intervals and constitute 35 subsets.
- **Sync** (mixture of periodic and aperiodic): We synthesize this dataset by combining cosine waves with different frequencies and amplitudes. Then, we manually insert noise, distorted period and slow-down anomalies.

For every dataset, we count a task duration time distribution I at each time slot and divide the intervals in I according to the distribution density of the execution time. We show the interval division for every dataset in Tab. 4.

Table 4: The interval division for each dataset.

Dataset	Edges of I
Ali1	{0, 10, 20, 30, 40, 70, 110, 150, 190, 230, 280, 330, 380, 430}
Ali2	{0, 10, 20, 30, 40, 70, 110, 150, 190, 230, 280, 330, 380, 430}
Mut	{0, 5, 10, 20, 30, 40, 70, 110, 150, 190, 230, 280, 330, 380, 430, 900, 1200, 9000}
Sync	{0, 10, 20, 30, 40, 70, 110, 150, 190, 230, 280, 330, 380, 430}

D HYPERPARAMETER SEARCHING SPACE

We use grid-search to figure out the optimal hyperparameter settings. We list the ranges for important hyperparameters in Tab.5.

Table 5: The searching ranges for important hyperparameters.

Hyperparameter	Searching Range
Skimming layers	{1,3,5,7}
Patch size	{2,3,4,5,7,9,11,15}
Window length	{10,20,30,40,50,80}
Learning rate	{0.0001,0.001,0.01}

E BASELINES INTRODUCTION

- **DCdetector**: DCdetector is one of the most SOTA anomaly detection methods, which assembles a novel dual attention asymmetric design and a pure contrastive loss.
- **TranAD**: TranAD is an influential and novel anomaly detection method, which is assisted by meta-learning and shows the high accuracy of anomaly detection.
- **AnomalyTransformer**: AnomalyTransformer is one of the founders who introduced the deep transformer into the area of anomaly detection, which is verified with strong performance.
- **VQRAE**: VQRAE is a novel and sharp anomaly detection method, which also delves into the problem that there are anomalies in the training set. Thus, we also include this method in our baseline.
- **OmniAnomaly**: OmniAnomaly is one of the most widely-recognized and widely-used anomaly detection methods with small time and memory overhead.
- **MSCRED**: MSCRED is an anomaly detection method garnering widespread attention with strong efficacy, which not only considers the temporal correlation but also takes the interdependency between features into account.
- **IASO**: IASO is a method specifically designed to detect the slow-down data retrieval of disks.
- **Feature-shift detection**: The feature-shift detection method is designed to detect whether the distribution of features has shifted.