ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230422

DeepDiscord: Dual Contrastive Coding for Transferable Time Series Anomaly Detection

Xin-Yi Li [a], Pei-Nan Zhong[b], Di Chen[a], Zhen-Dong Zhang[b] and Yu-Bin Yang [a;*]

[a]State Key Laboratory for Novel Software Technology, Nanjing University
[b]General Development Dept, Huawei Technologies Co. Ltd.

Abstract. Time series anomaly detection has attracted extensive research attention owing to its real-world applications. Existing deep learning based anomaly detectors usually require a separate training phase for each dataset. However, the long training time restricts their practicality in the industry use. To address this limitation, we propose a novel deep learning based discord search method named *DeepDiscord*, which is a multi-scale anomaly detector capable of directly examining unseen datasets after pre-training. To the best of our knowledge, our study is the first to introduce contrastive learning in the discord search, in order to provide a flexible and effective similarity measure for various kinds of data. We innovatively divide the data into two categories according to their roles in discord search, and combine dual learning with contrastive learning, which improves the efficiency and efficacy of discord search. Furthermore, a novel pretext task is proposed based on our dual contrastive learning setting. We evaluate DeepDiscord comprehensively on five anomaly detection benchmarks. Experimental results show that DeepDiscord achieves the state-of-the-art results on the four out of five benchmarks.

1 Introduction

Time series anomaly detection (TSAD), as a major branch of data science, has gathered extensive research interest [6, 7]. With the aim to spot the regions deviating significantly from the normal data, it has been widely applied for monitoring various systems, such as industrial equipment and cloud services [29, 22], etc. In many practical cases, due to the scale and complexity of these systems, there are usually massive metrics to analyze in order to avoid serious failures [4, 29]. For a typical example, cloud platform monitoring is often required to detect millions of metrics of different nature (e.g. CPU usage, latency, error rate) [33]. Undetected anomalies could seriously affect service availability and reliability, and even result in huge financial losses. Therefore, designing a quick and accurate anomaly detection system is essential in industry.

In recent years, deep learning based TSAD works have demonstrated superior performance [33, 41, 3]. However, their effectiveness commonly depends on the data-specific parameters. For each dataset, model parameters are learned independently and hyperparameters are chosen carefully. Hence, the numbers of required models and metrics are of the same magnitude. With millions of metrics to be monitored in industrial systems, the costs to involve experts to tune the systems are extremely expensive and hence unpractical. Furthermore, for applications in dynamic environments, the performance of existing methods may degrade along time because of the concept drift. And to prevent deterioration, timely model updates are necessary, which will inevitably result in more maintenance burdens. Therefore, anomaly detectors with cross-data transfer ability are strongly demanded in practical use and become an important research topic in the scientific community.

To address this problem, we investigate TSAD from time series *discords* [18] perspective. Time series discords are subsequences of longer time series that maximally different from all other subsequences in the same time series. Accordingly, discords indicate the outlier windows, and discord search is naturally unsupervised window-level anomaly detection. As demonstrated in [18], the most appealing aspect of discord search is that, without a separate training phase for each dataset, anomalies can be discovered by merely examining the test dataset. Secondly, it only require a single intuitive parameter, the length of the subsequence. With these two features, cross-data detection is an intrinsic capability of time series discords.

Current discord search in the literature, the similarity between subsequences is measured via normalized Euclidean distance [18, 42, 28]. However, this distance measure may fail to identify trend outliers [20].

In addition, it is especially vulnerable to data noise, which is common in industrial applications. Therefore, the distance function in discord search should be replaced with a measure that can identify outliers presenting a variety of patterns and is applicable to time series with diverse features.

In this paper, we propose *DeepDiscord*, a novel **Deep** learning based **Discord** search approach using dual contrastive coding for transferable TSAD. To the best of our knowledge, it is the first study that introduces contrastive learning to enhance the efficacy and efficiency of time series discord search. Compared to previous discord studies, DeepDiscord leads to three major advances. First, it employs constrastive learning to provide a flexible distance measure adaptive to various kinds of time series; Second, a dual-encoder network architecture is designed to enable comparison between subsequences of different lengths, which improves the flexibility and efficiency of discord search; Third, based on our dual learning setting, a novel self-supervised task called *Sub-window Matching Task* is proposed to learn multi-scale representations. Consequently, DeepDiscord achieves superlative performance than its traditional counterpart while retaining the attractive cross-data transferability of discords search.

* Corresponding Author. Email: yangyubin@nju.edu.cn. This work is funded by the Natural Science Foundation of China under Grants 62176119.

Although a series of works on time series contrastive representation methods were proposed and showed success in various downstream tasks [43, 36], we believe taking the characteristics of discord search into consideration during the pre-training phase is beneficial for the detection performance. Therefore, we propose our dual contrastive learning approach, in order to train two encoders that yield representations of different semantics. As we shall show, the disparate semantics play a key role in anomaly detection and outperform other downstream agnostic pre-training strategies.

To validate the effectiveness of our method, we first pre-trained our model on a time series archive with rich patterns, then directly transfer and evaluate our model on five time series anomaly detection benchmarks. Comprehensive experimental results show that our pre-trained model achieves competitive performance against state-of-the-art TSAD methods without retraining.

The main contributions of our work are listed as follows:

- We present a novel time series anomaly detection approach based on discord search, which is able to directly transfer to unseen data once finish pre-training.
- To ensure the efficiency and precision of discord search, we divide the time windows into two categories and adopt dual-encoder architecture to learn representations with different semantics.
- We propose a novel pre-text task called Sub-window Matching Task to perform multi-scale contrastive learning. To the best of our knowledge, we are the first to introduce contrastive learning to time series discord problem.
- We directly evaluate our pre-trained model on five TSAD benchmarks and achieve competitive performance against state-of-the-art traditional and deep TSAD methods.

2 Related Work

2.1 Time Series Unsupervised Anomaly Detection

Unsupervised TSAD methods are strongly preferred in the literature since time series are rarely or sparsely labeled in practice. Existing studies can roughly be divided into categories depending on detecting mechanisms such as one-class classification, autoregression, reconstruction, and discord search.

One-class classification methods assume that the majority of the data is normal. Classic methods rely on the kernel trick [24, 35], and deep classifiers replace the kernel-induced feature space by learned feature space [30, 32]; Autoregression-based methods identify the outliers by measuring the error between predicted and actual data [16, 34]; Reconstruction-based models detect the anomalies by reconstruction error, which mainly consists of methods based on Variational AutoEncoder [33, 22] and Generative Adversarial Network [23, 21]; Discord search methods measure the similarity between subsequences and regard discords as outliers. MatrixProfile [42] computes minimum Euclidian distances between each subsequence and the others. MERLIN [28] further removes the only required parameter (subsequence length) by finding discords of all lengths.

In addition to the above approaches that follow the popular paradigms, Anomaly Transformer [41] proposes a novel association-based criterion. Besides, there are cold-start methods, such as LOF [9], Twitter-AD [37], Luminol [8] and SR [29]. However, most of the deep learning based methods need to model the given data. Hence we adopt discord search, which requires far less dependency towards data-specific training.

2.2 Time Series Representation Learning

Although self-supervised representation learning has achieved significant progress in computer vision and natural language processing [13, 12], it has been far less studied in the time series domain. As for generative training objectives, auto-encoder based methods such as TimeNet [26] minimizes the input reconstruction error, and forecasting methods such as [44] predicts the future values. However, generating complex time series can be very challenging. Consequently, contrastive learning has been widely studied. T-Loss [15] uses random sub-series from the original time series as positive samples, which is consistent with ours but we adopt dual-encoder architecture for discord search problem. TNC [36] leverages the local smoothness of a signal to define neighbors. TS-TCC [14] forces the consistency between the two augmentations of the input data. TS2Vec [43] hierarchically discriminates positive and negative samples at instance-wise and temporal dimensions.

In addition to the above universal representation works, there are studies that learn representations for specific downstream tasks. For example, CoST [38] learns disentangled trends and seasonal representations for forecasting tasks, and $TS-CP^2$ [11] exploits the local correlation hypothesis for change point detection. In this paper, we propose a novel time series representation learning method towards anomaly detection.

3 Method

The overview of DeepDiscord is shown in Figure 1, which consists of two phases. At dual contrastive learning phase, query and reference encoders are jointly trained by predicting the matching pairs from a set of query-reference examples. At discord search phase, anomaly scores are calculated as the similarity between the embeddings of query and reference.

3.1 Preliminaries and Motivations

A N-dimensional time series of length T can be denoted as

$$\mathbf{X} = [x_1, x_2, \ldots, x_N], \quad x_t \in \mathbb{R}^T \tag{1}$$

As conventional approaches, it could be split into a series of windows with length τ and stride 1 using the sliding window strategy,

$$\mathbf{W} = [w_1, \ldots, w_{T-\tau+1}], \text{ where } w_t = [x_t, \ldots, x_{t+\tau-1}] \tag{2}$$

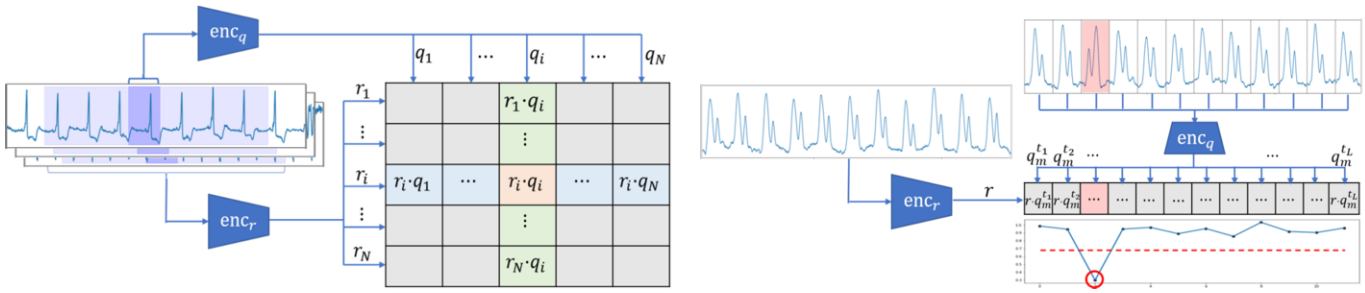
Time series anomaly is a data point or segment that significantly differs from other observations. Since labeled anomalies are usually contiguous segments that can be transformed into window level labels, the goal of TSAD can be formulated as predicting the anomaly label for each window w_t.

Although the proposed framework supports both univariate and multivariate time series training, in this work, we focus on pre-training model for univariate time series. Since it can be applied on multivariate time series dataset with different dimensions, by processing each component $\{x_t^d\}_{t=1,\ldots,T}$ as independent series and then ensembling the results.

Let $\mathbf{X}_q$ denotes a univariate time series of length T_q, which needs to be detected. And another series $\mathbf{X}_r$ of length T_r is also given as normal historical values of $\mathbf{X}_q$. These two series are referred to as query and reference respectively. Let $\mathbf{Q}$ and $\mathbf{R}$ denote the sliding window sequences corresponding to $\mathbf{X}_q$ and $\mathbf{X}_r$ respectively.

$$\mathbf{Q} = [q_t]_{t=1,\ldots,T_q-\tau_q+1}, \quad q_t = \left[x_t^q, \ldots, x_{t+\tau_q-1}^q\right] \tag{3}$$

$$\mathbf{R} = [r_t]_{t=1,\ldots,T_r-\tau_r+1}, \quad r_t = [x_t^r, \ldots, x_{t+\tau_r-1}^r] \tag{4}$$



(a) Dual Contrastive Learning (b) Discord Search

Figure 1: The proposed architecture of our approach. DeepDiscord adopt dual-encoder architecture to learn representations of query and reference respectively. (a) Overview of the dual contrastive learning phase. The training objective is to identify the query-reference pair that sampled from the same time series instance from pairs that sampled from different ones. (b) Overview of the discord search phase. We search the discord according to the similarity between the representations of query and reference.

It should be noticed that the window lengths τ_q, τ_r are not required to be the same, and generally, the reference window is longer than the query window, i.e. $\tau_r > \tau_q$.

The key challenge of discord search is defining a appropriate similarity measure between windows. Traditional discord methods exploit Euclidean metric, which is too restrictive and sensitive to data noise. Therefore, we propose to measure the similarity defined on the learned space. By carefully designing a pretext task, our learned representations could capture the key features and discard noises.

3.2 Dual Contrastive Learning

Intuitively, contrastive representation learning can be considered as learning by comparing. The representations of similar samples will be mapped close together, while those of dissimilar samples will be pulled away in the embedding space. Therefore, a pretext task should be designed to define the distributions for positive samples $P(\mathbf{x}^+ \mid \mathbf{x})$ and negative samples $P(\mathbf{x}^- \mid \mathbf{x})$ with respect to a given input x.

3.2.1 Sub-window Matching Task

A novel pretext task named Sub-window Matching Task is discussed in this part. Similar to other self-supervised tasks, the notion of similarity is defined according to the pre-training data via the designed sampling process.

Given a pre-training dataset consisting of time series of different lengths

$$\mathcal{D} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots\}.$$

The inputs of self-supervised training are constructed by the following three steps.

Firstly, a time series $\mathbf{X} \sim P_{\mathcal{D}}(\cdot)$ is randomly drawn from $\mathcal{D}$. Then a subsequence r_i of length τ_r is randomly sampled from $\mathbf{X}$, which could be viewed as an analogy to the historical window at detection. After N times of sampling, i.e. $r \sim P(r)$, a batch of references is collected

$$\mathcal{R} = \{r_1, \ldots, r_N\}.$$

Secondly, a subsequence q_i of length $\tau_q < \tau_r$ is randomly sampled from each reference r_i, i.e. $q_i \sim P(q|r_i)$, which is regarded as the query window.

Thirdly, a random window shift is applied to the query window q_i in order to encourage the model to focus on the pattern modeling and learn translation-invariant representations. After sampling and shifting, the batch of query windows corresponding to $\mathcal{R}$ is collected

$$\mathcal{Q} = \{q_1, \ldots, q_N\}.$$

It should be noticed that in order to support multi-scale detection, the query window length τ_q is set randomly from a range of lengths smaller than τ_r for each batch iteration.

Obviously, with respect to r_i, the subsequence $q_i \subset r_i$ could be regarded as the normal pattern, while queries $q_j \not\subset r_i$ sampled from other series should be considered as the anomalous data. Formally, the positive query-reference pairs are sampled from the joint distribution $(q_i, r_i) \sim P(q, r) = P(q \mid r)P(r)$, and the negative pairs are sampled from the product of marginals $(q_i, r_j) \sim P(q)P(r)$. More detailed discussion can be found in [31].

3.2.2 Dual-encoder architecture

Traditional discord search methods require the similarity to be defined on time windows of equal length. For discords exhibiting abnormal patterns at different length resolutions, this restriction inevitably requires a full scan of all possible lengths to discover them, which will result in significant increase of computation time.

To improve the efficiency of search, we propose to learn representations with different semantics for query and reference windows. Therefore, we perform our contrastive coding in a dual learning setting to extract features of different granularities. Specifically, learned reference representations are prompted to characterize the large-scale "slow features" of normal behaviors spanning multiple query window lengths, while query representations are encouraged to describe their own detailed features. Taking account of the semantic disparity, the dual-encoder architecture is adopted to encode the query and reference windows separately.

Formally, two encoders g_θ and h_ϕ are employed to extract the features of query and reference window respectively, where θ and ϕ are their corresponding parameters. The latent representations are encoded as

$$z = g_\theta(q), \quad w = h_\phi(r).$$

The similarity between a query-reference pair (q, r) can be calculated via the cosine similarity of their embedding z, w:

$$\mathrm{sim}_{(\theta,\phi)}(q, r) = \cos\left(g_\theta(q), h_\phi(r)\right) \tag{5}$$

This dual-encoder architecture is of great benefit for efficient discord search. For a given query-reference window pair (q, r), with $\tau_q \ll \tau_r$, the similarity via embedding only takes $\mathcal{O}(1)$ to calculate. However, traditional approaches require $\mathcal{O}(\tau_r)$ to compute the similarity by scanning all sliding windows in r with length τ_q. Futhermore, enlarging the reference window length τ_r can reduce the number of encoded windows, thereby expediting the encoding for the en-

tire reference series and the discord search. Additionally, for multi-scale detection, all query windows with length scale τ_q smaller than τ_r can be compared with r via their embeddings, without updating reference representations.

3.2.3 Learning objective

The basic idea of contrastive learning is to learn a embedding that the similarity function (5) will achieve high values for positive query-reference pairs (q_i, r_i) and low values for negative pairs (q_i, r_j), with $i \neq j$. Similar to recent setups for contrastive learning, the training processing is driven by minimizing the InfoNCE loss [11].

Let $f_{(\theta,\phi)}$ denote the exponential of the similarity function (5)

$$f_{(\theta,\phi)}(q, r) \equiv \exp\left[\cos\left(g_\theta(q), h_\phi(r)\right)\right] \tag{6}$$

For a query batch $\mathcal{Q}$ and a reference batch $\mathcal{R}$ sampled as discussed above, by treating a given reference $r_i \in \mathcal{R}$ as an anchor, the query batch $\mathcal{Q}$ contains one positive sample and $N-1$ negative samples. Accordingly, the InfoNCE loss can be written into the form:

$$\mathcal{L}_r = -\frac{1}{N}\mathbb{E}_{(\mathcal{Q},\mathcal{R})}\left[\sum_{r^i\in\mathcal{R}} \log \frac{f_{(\theta,\phi)}(q^i, r^i)}{\sum_{q^j\in\mathcal{Q}} f_{(\theta,\phi)}(q^j, r^i)}\right] \tag{7}$$

Symmetrically, by treating a given query as the anchor and enumerating over $\mathcal{R}$, the InfoNCE loss takes the form

$$\mathcal{L}_q = -\frac{1}{N}\mathbb{E}_{(\mathcal{Q},\mathcal{R})}\left[\sum_{q^i\in\mathcal{Q}} \log \frac{f_{(\theta,\phi)}(q^i, r^i)}{\sum_{r^j\in\mathcal{R}} f_{(\theta,\phi)}(q^i, r^j)}\right] \tag{8}$$

Finally, the training objective is defined as

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_q \tag{9}$$

3.3 Discord Search

3.3.1 Anomaly Score and Probability Estimation

Once the model has been trained, the similarity (5) will achieve high values for positive pairs and low values for negative pairs. It will be shown in this part that this similarity gives a reasonable anomaly score definition and the meaning of its values can be explained from the probability perspective.

Let the latent random variable C denotes whether a pair (q, r) was drawn from the joint ($C = 1$), or the product of marginals ($C = 0$),

$$P(q, r \mid C = 1) = P(q, r), \quad P(q, r \mid C = 0) = P(q)P(r) \tag{10}$$

The distribution $P(q, r)$ is defined in the sampling process described in section 3.2.1.

Recall that for a pair (q, r) drawn from the joint distribution, the query window q is considered as a normal pattern with respect to the reference window r, since q is sampled from the subsequence r. Therefore, the posterior $P(C = 1 \mid q, r)$ gives the probability of q being normal for reference r. According to Bayes' theorem, this posterior can be expressed as

$$\begin{aligned}
&P(C = 1 \mid q, r)\\
=&\frac{P(q, r \mid C = 1)P(C = 1)}{P(q, r \mid C = 0)P(C = 0) + P(q, r \mid C = 1)P(C = 1)}\\
=&\frac{P(q, r \mid C = 1)P(C = 1)}{P(q, r)P(C = 1) + P(q)P(r)P(C = 0)}\\
=&\frac{1}{1 + \frac{P(q)P(r)}{P(q,r)}\frac{P(C=0)}{P(C=1)}}
\end{aligned} \tag{11}$$

Since the optimal value for $f_{(\theta,\phi)}$ is proportional to $\frac{P(q,r)}{P(q)(r)}$ [11], the posterior can be simplified into

$$P(C = 1 \mid q, r) = \frac{1}{1 + \beta \cdot \exp(-\mathrm{sim}_{(\theta,\phi)}(q, r))} \tag{12}$$

where β is a constant. It is obvious that when the similarity value of a query-reference pair (q, r) increases, there is more chance that q is a normal window, and vice versa.

Let $\mathbf{X}$ denotes a time series which we want to detect, and associated with X, there is a vector set $\mathcal{R}_\mathbf{X}$ containing historical time windows of length τ_r. Samples $r \in \mathcal{R}_\mathbf{X}$ are regarded as normal patterns happened before. For a subsequence $q^t = \mathbf{X}[t : t + \tau_q]$ of length $\tau_q < \tau_r$ beginning at position t, its anomaly score is defined as

$$S_a\left(q^t;\, \mathcal{R}_\mathbf{X}\right) = 1 - \max_{r\in\mathcal{R}_\mathbf{X}} \mathrm{sim}_{(\theta,\phi)}(q^t, r). \tag{13}$$

It should be noticed that the computation of $\max_{r\in\mathcal{R}_\mathbf{X}}$ can be accelerated by leveraging the off-the-shelf methods for approximate nearest neighbor search [17].

We use a sliding window sequence $\mathbf{Q}_\mathbf{X}$ to extract all possible subsequences of length τ_q from $\mathbf{X}$. If q^t has the largest anomaly score among $\mathbf{Q}_\mathbf{X}$, it is said to be the *discord*. If q^t has the K^{th} largest anomaly score, it is the K^{th} discord. The ranking of discords gives the severity of an anomaly.

3.3.2 Multi-scale Detection

Recall that our method is compatible with multi-scale detection to discover outliers exhibiting at different length resolutions. For a reference window set $\mathcal{R}_\mathbf{X}$ with fixed window size τ_r, we could use query windows beginning at the same position t and with monotonically increasing lengths to calculate the anomaly scores

$$\left\{q_m^t\right\}_{m=1,\ldots,M}, \quad \tau_{q_m^t} < \tau_{q_{m+1}^t} < \tau_r \quad \text{for all } m \tag{14}$$

where M is the number of length scales required to analyze.

After enumerating all scores $S_a(q_k^t; \mathcal{R}_\mathbf{X})$ at different scales, the multi-scale score is calculated by averaging

$$\tilde{S}_a(t) = \frac{1}{M}\sum_{m=1}^{M} S_a(q_m^t;\, \mathcal{R}_\mathbf{X}) \tag{15}$$

As we shall show, the multi-scale score $\tilde{S}_a$ could obtain more robust and better predictive performance than could be obtained from any S_a of the constituent scores alone.

3.4 Encoder Architecture

Since the learning framework of DeepDiscord is agnostic to the network architecture, the encoder could be any parametric model that meets the two requirements, which are vital to the multi-scale detection: 1. it allows sequence modeling with variable lengths τ, i.e. $f : \mathbf{X}[t : t + \tau] \mapsto \mathbf{Z}[t : t + \tau]$; 2. it satisfies the causal constrain. Sequence modeling allows the encoder to take query windows of different lengths as its inputs. Causal modeling helps the encoder to obtain multi-scale representations starting at the same position at one time.

In this work, query encoder g_θ and reference encoder h_ϕ are designed to share a same network architecture satisfying the two requirements discussed above, which is illustrated in Figure 2. Firstly, we leverage a linear network to map the input time series value into a

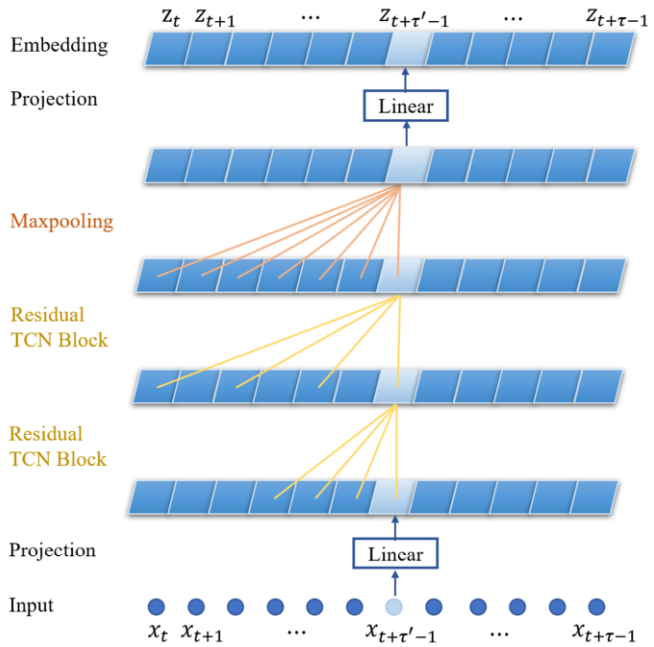


Figure 2: The encoder architecture for DeepDiscord. A linear layer is employed to project the input into vector space. We use two stacks of TCN residual blocks followed by a causal maxpooling layer to encode temporal features. The final linear layer is appended as a projection head.

dense vector. Next, two stacks of residual TCN [5] blocks are adopted to extract temporal features in a sequential manner. Each TCN block consists of two layers dilated causal convolution and non-linearity. Directly after the convolution layers, a masked maxpooling is applied without future input dependencies. Finally, a linear layer is added as the projection head.

Owing to the causal nature of the encoder, an element in the output sequence at timestep t cannot depend on any elements in the input sequence at the future timesteps $t' > t$. Let $\mathbf{X}[t : t + \tau]$ denotes an input sequence of length τ beginning at t. For any smaller length $\tau' < \tau$, the element $\mathbf{Z}_{t+\tau'-1}$ in the output sequence $\mathbf{Z}$ could be taken as the embedding of the window $\mathbf{X}[t : t + \tau']$. Therefore, the embeddings of all the smaller scales $\{\mathbf{X}[t : t + \tau']\}_{\tau'=1,\ldots,\tau}$ are directly obtained from the output sequence $\mathbf{Z}[t : t + \tau]$ only through a single neural network calculation. Consequently, the multi-scale detection processing is effectively accelerated.

4 Experiments

In this section, we extensively evaluate DeepDiscord on five commonly-used TSAD benchmarks.

4.1 Datasets and Evaluation Metrics

The five datasets that we use to evaluate the anomaly detection performance of DeepDiscord are listed as following: 1. UCR [39] is a univariate dataset consists of 250 sub-datasets with a single anomaly. 2. PSM [1] (Pooled Server Metrics) is collected from multiple application server nodes at eBay with 26 dimensions. 3. ASD [22] (Application Server Dataset) is collected from an Internet company and contains the metrics of 12 servers, where each server corresponds to 45 days of data with 19 metrics. 4. SWaT [27] (Secure Water Treatment) is obtained from 51 sensors of a water treatment testbed over 11 days and 36 attacks were launched in the last 4 days. 5. WADI [2] (Water Distribution) is a multivariate dataset collected from a water distribution testbed, where 14 days were collected under normal status and 2 days with attack scenarios.

According to the characteristics of the datasets, we adopt two sets of evaluation measures. Since time series in UCR only contains one anomaly, detectors are required to return the top-k discords. Top-k accuracy computes whether the labeled segment is contained in the top-k discords, which is a fair evaluation for UCR, as it is independent of threshold selection methods. In this paper, we use top-1, top-2, and top-3 accuracy for comprehensive evaluation. For the rest four real-world datasets that do not have any assumption on the number of anomalies, we adopt the point-adjust approach [40] that has been widely used in TSAD [22, 41]. With this strategy, we can eliminate the influence introduced by the labeling noises for the metrics [25]. Following previous works, we use precision, recall and F1-score to evaluate the performance.

4.2 Baselines

Our proposed model DeepDiscord is compared with the following two groups of anomaly detection algorithms. The first group consists of three state-of-the-art anomaly detection methods. USAD [3] uses an adversarially trained autoencoder to model the inter-metric dependency; InterFusion [22] simultaneously models the inter-metric and temporal dependency for multivariate time series; Anomaly Transformer [41] identifies anomalies by distinguishing abnormal association discrepancy. In order to achieve fair comparisons, the second group consists of univariate cold-start or zero-shot anomaly detectors that do not require extra training and disregard inter-metric dependency. LOF [9] calculates the local density; Twitter-AD [37] adopts statistical methods; Luminol [8] utilizes the frequency of similar time series chunks; SR [29] based on Spectral Residual and Convolutional Neural Network; MatrixProfile [42] and MERLIN [28] are representative discord detection methods. As for TS2Vec [43], we use its transferred version, which is a universal time series representation model pre-trained on the FordA dataset in the UCR archive.

4.3 Implementation Details

At the pre-training phase, our model is trained on the training data of UCR Time Series Anomaly and Classification Archive [39, 10], which provide many time series instances of different nature for our encoders to learn pattern modeling. As for multi-scale learning, the reference window size is fixed to 512 and the query window size is sampled from the range $[20, 256]$. Our reference and query encoder share the same model structure, where the kernel sizes of TCN blocks are 16 and the dimensions of the hidden states and final representations are 64. We use the ADAM [19] optimizer with an initial learning rate of 10^{-3}. All the training experiments are performed with a single NVIDIA V100 GPU.

At the detection phase, we also set the reference window size as 512 and enumerate the query window size from 64 to 512 with interval 16. After obtaining the anomaly scores, we first find the peaks over the threshold and then label the timestamps within the peaks' widths as anomalies. The threshold is searched by moving the initial threshold from top to down until the next move will cause the number of the detected consecutive abnormal segments exceeding a predefined value. Both the moving step and the maximal amount of the anomaly segments are slightly varied from different datasets.

Table 1: Performance of DeepDiscord on five datasets. We use top-k ($k = 1, 2, 3$) accuracy (Acc@1, Acc@2, Acc@3) (as %) to evaluate the results on UCR, and report precision (Prec), recall (Rec) and F1 score results (as %) on the rest four datasets.

Dataset	UCR			PSM			ASD			SWaT			WADI		
Metrics	Acc@1	Acc@2	**Acc@3**	Prec	Rec	**F1**	Prec	Rec	**F1**	Prec	Rec	**F1**	Prec	Rec	**F1**
USAD	17.60	25.20	30.40	63.32	68.82	65.96	45.63	49.25	47.37	57.48	72.81	64.24	48.61	36.45	41.66
InterFusion	11.20	16.40	22.80	97.62	91.81	94.62	79.16	93.06	85.54	86.08	87.21	86.64	96.80	41.40	58.00
Anomaly Transformer	10.00	17.60	22.00	97.28	97.46	**97.37**	66.75	61.87	64.22	93.74	92.92	93.33	62.89	31.44	41.92
LOF	14.80	18.40	20.40	98.18	92.27	95.13	51.38	72.15	60.02	96.12	85.70	90.61	62.86	88.28	73.43
Twitter-AD	6.80	9.60	10.40	98.06	96.13	97.09	59.37	71.11	64.71	78.72	83.72	81.14	40.90	36.91	38.80
Luminol	14.80	20.00	23.20	79.96	95.50	87.04	31.86	29.97	30.89	71.27	74.96	73.07	15.00	54.10	23.49
SR	28.00	32.80	35.20	97.67	93.30	95.44	68.62	94.73	79.59	59.51	99.76	74.55	30.05	97.85	45.98
MatrixProfile	52.40	60.00	63.20	99.10	78.71	87.74	65.02	61.41	63.16	96.54	84.52	90.13	49.17	93.16	64.37
MERLIN	28.80	34.80	36.80	91.92	96.20	94.01	56.04	72.49	63.21	94.97	78.17	85.75	29.79	87.69	44.47
TS2Vec	18.80	20.80	21.20	98.63	93.91	96.21	81.40	91.42	86.12	71.20	99.26	82.92	36.84	96.33	53.29
DeepDiscord (ours)	70.80	78.00	**80.80**	98.24	95.31	96.76	87.79	94.55	**90.83**	94.25	97.35	**95.78**	84.04	90.85	**87.31**

Table 2: Ablation results (Top-3 Accuracy for UCR, F1 Score for PSM, ASD, SWaT, and WADI) of DeepDiscord.

	UCR	PSM	ASD	SWaT	WADI	Avg.
DeepDiscord	**80.80**	**96.76**	**90.83**	95.78	87.31	**90.29**
w/o Dual Learning	75.60	91.34	83.95	92.65	80.22	84.75
w/o Multi-scale Contrast	65.20	91.72	90.67	83.02	87.19	83.56
Augmentations						
Shift						
Original	76.00	89.26	88.33	90.99	80.19	84.95
Jitter	74.80	93.34	88.11	93.54	89.28	87.81
Shift and Jitter	78.00	92.27	85.52	**96.45**	82.35	86.91
Original or Shift	76.00	91.72	83.39	89.82	81.93	84.57
Original or Jitter	72.00	92.09	80.98	89.82	82.43	83.46
Shift or Jitter	76.80	91.25	88.88	90.78	88.93	87.32
Training Objective						
$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_q$						
$\rightarrow \mathcal{L}_r$	77.60	92.79	89.71	93.99	**90.29**	88.87
$\rightarrow \mathcal{L}_q$	77.20	92.80	89.27	94.69	85.43	87.87

4.4 Main Results

We extensively evaluate our method on five datasets with ten competitive baselines. Table 1 illustrates that DeepDiscord achieves state-of-the-art on four out of five benchmarks without training on their training datasets, demonstrating the transferibility and efficacy of our method. The explanation for our slightly lower F1 score on PSM is that the truth abnormal proportion of PSM is 27%, which is notably larger than other datasets. Therefore, the methods report more anomalies are likely to obtain better results.

UCR is a very challenging dataset that covers various real-world scenarios. As shown in Table 1, existing methods perform extremely poorly on UCR and DeepDiscord outperforms all baselines by a substantial margin. Specifically, compared to previous state-of-the-art aprroach MatrixProfile, DeepDiscord gives 17.6% improvement on top-3 accuracy. Furthermore, DeepDiscord shows its robustness by obtaining stable and strong results on the rest four datasets compared to other works. This superior performance suggests the DeepDiscord's capability of detecting different types of anomalies on data with various patterns.

4.5 Ablation Studies

We conduct ablation studies on the five benchmarks using several variants of DeepDiscord to further demonstrate the effectiveness of the designs for the learning and detection mentioned in Section 3. For simplicity, we only report the Top-3 Accuracy for UCR and the F1 Score for the rest four datasets.

4.5.1 Contrastive Learning Ablations

The ablation results towards the contrastive learning components are illustrated in Table 2.

Firstly, we investigate our proposed pre-text task, where **w/o Dual Learning** encodes the query and reference windows with the same encoder, **w/o Multi-scale Contrast** fixes the query window length to 64 during training. Results demonstrate the superiority of our Subwindow Matching Task, as metrics decrease by at least 5% on three datasets when the key components of the task are ablated.

Secondly, we study the augmentation strategies. Recall that DeepDiscord performs window shift to enhance the encoders' pattern modeling capability. To explore the impacts of the data augmentation, we first remove window shift (**Original**), and then attempt **Jitter** that introduces random noises. Additionally, we investigate the performance when applying multiple augmentations individually and jointly. Concretely, **Shift and Jitter** refers to the sequential application of shift and jitter to each query, whereas **Shift or Jitter** refers to the random application of shift or jitter. Notice that scores drop significantly when the original queries are used for contrasting. Furthermore, overall performance decreases when shift is replaced with other augmentation policies. We assume that this is because in some cases jitter could impact the original pattern of the query and thus the encoder is encouraged to ignore these changes, which is unfavorable for anomaly detection.

Lastly, we investigate the training objective. Our proposed loss function (9) consists of $\mathcal{L}_r$ (7) and $\mathcal{L}_q$ (8). Table 2 shows that results degrade marginally when we optimize them independently, which suggests that our defined loss is superior.

4.5.2 Multi-scale Detection Ablations

We have ablated multi-scale contrast in Section 4.5.1 to verify its necessarity for learning robust representations. To further suggest the importance of multi-scale detection, we explore the performance of DeepDiscord under the single-scale detection setting. Figure 3 depicts the results under different detection scales, ranging from 64 to 512 window sizes. We could observe that the multi-scale setting surpasses all single-scale settings for UCR, ASD, and SWaT, which indicates that the optimal detection scale differs for each time series. Despite the fact that a few single-scale settings outperform the multi-scale setting on PSM and WADI, it is infeasible to set a optimal

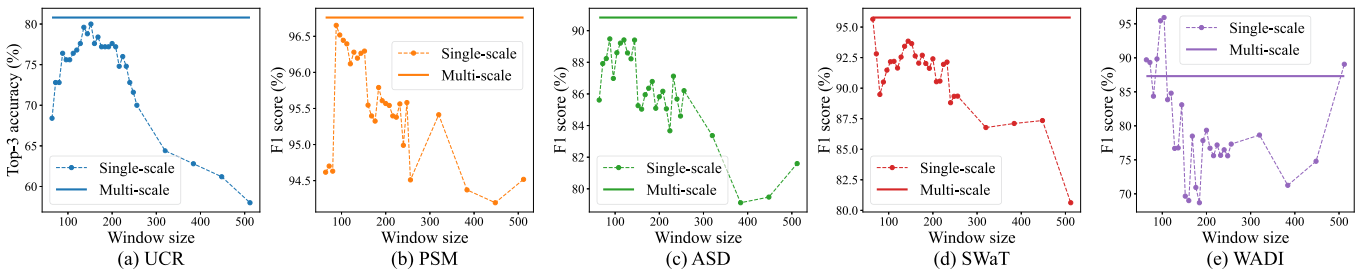


Figure 3: Performance comparison of multi-scale and single-scale detection across different window sizes on five benchmarks.

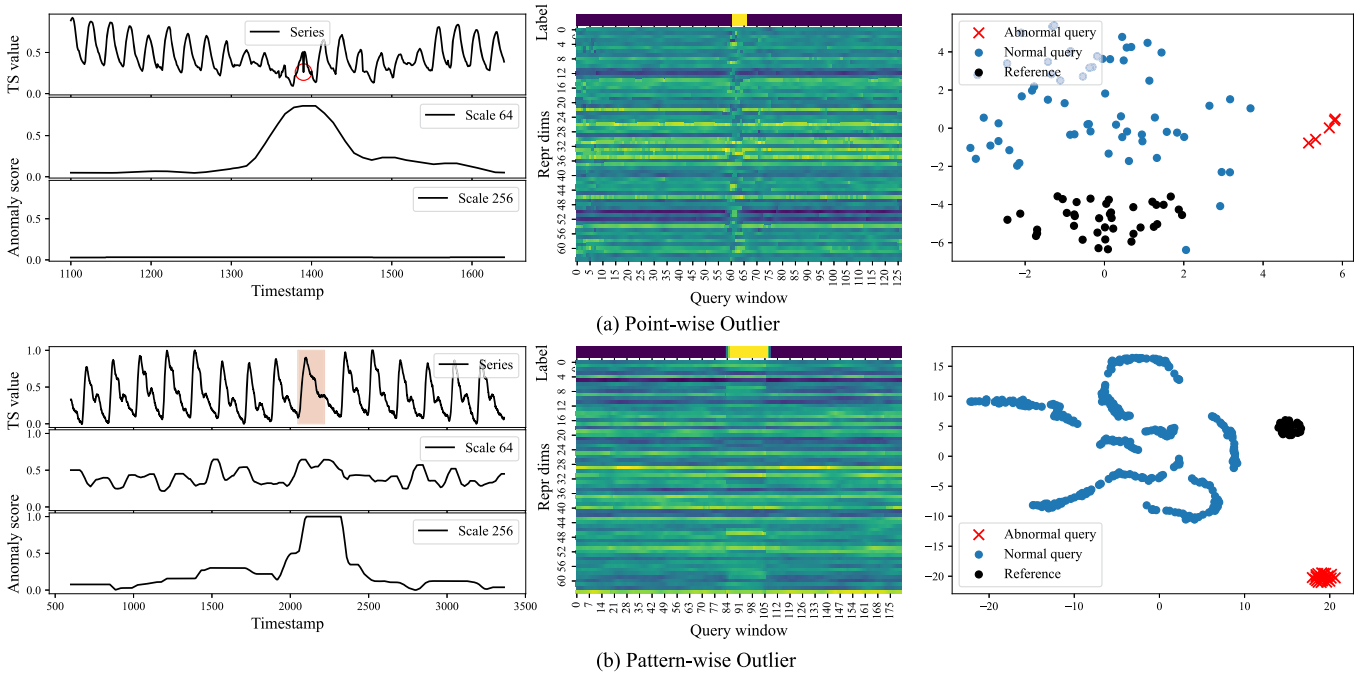


Figure 4: Examples of two types of anomalies. We plot the series values and anomaly scores (first column), display the query embeddings' heatmaps (second column), and display the reference and query embeddings' t-SNE visualizations (third column).

scale in advance during practice. In summary, the results presented in Figure 3 illustrate that the multi-scale score (15) could obtain more robust performance than any single-scale alone.

4.6 Model Analysis

To illustrate the mechanism of our detection algorithm intuitively, we visualize the anomaly scores and the representations in Figure 4. We use two examples from UCR to demonstrate that DeepDiscord is able to identify both point- and pattern-wise outliers.

As for the point-wise case, the anomaly scores calculated under scale 64 clearly reveal the ground-truth anomaly, whereas the scores under scale 256 cannot indicate any anomalies. This is because the query encoder tends to prioritize the pattern-wise feature and ignore the fine-grained information when encoding long windows. Consequently, short queries are better suited for detecting point-wise anomalies. In contrast, the scores under scale 256 are much more distinguishable for identifying the pattern-wise anomaly. These two examples show that multi-scale detection is crucial for DeepDiscord to cover various kinds of anomalies.

We also visualize the representations under the suitable scales for the two samples. The heatmaps in the second column show that query representations clearly distinguish the abnormal windows from the normal ones. Moreover, normal queries are closer to the references compared to the anomalous ones, as depicted by the t-SNE visualizations in the third column. Thus, our learnt feature space is a qualified metric for anomaly detection.

5 Conclusion

This paper studies the unsupervised time series anomaly detection problem. Unlike existing deep learning-based methods, we investigate the problem from the time series discords perspective. To overcome the limitations of the distance functions adopted by the classic time series discord method, we propose DeepDiscord, which introduces dual contrastive learning to the time series discord problem. Different from previous time series representation learning methods, we perform contrastive learning in a dual learning setting to ensure the model efficiency and effectiveness under multi-scale detection. Based on our dual learning setting, we further propose a novel self-supervised task called Sub-window Matching Task to learn multi-scale representations. Extensive experimental results show that DeepDiscord achieves competitive results on five time series anomaly detection benchmarks.

References

[1] Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki, 'Practical approach to asynchronous multivariate time series anomaly detection and localization', in *ACM SIGKDD*, p. 2485–2494, (2021).

[2] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur, 'Wadi: A water distribution testbed for research in the design of secure cyber physical systems', in *CySWATER*, p. 25–28, (2017).

[3] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga, 'Usad: Unsupervised anomaly detection on multivariate time series', in *ACM SIGKDD*, p. 3395–3404, (2020).

[4] Fadhel Ayed, Lorenzo Stella, Tim Januschowski, and Jan Gasthaus, 'Anomaly detection at scale: The case for deep distributional time series models', *CoRR*, **abs/2007.15541**, (2020).

[5] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun, 'An empirical evaluation of generic convolutional and recurrent networks for sequence modeling', *arXiv:1803.01271*, (2018).

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

[7] Mohammad Braei and Sebastian Wagner, 'Anomaly detection in univariate time-series: A survey on the state-of-the-art', *CoRR*, **abs/2004.00433**, (2020).

[8] V Brennan and M Ritesh. Luminol. https://github.com/linkedin/luminol.

[9] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander, 'Lof: Identifying density-based local outliers', in *ACM SIGMOD*, p. 93–104, (2000).

[10] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh, 'The ucr time series archive', *IEEE/CAA Journal of Automatica Sinica*, **6**(6), 1293–1305, (2019).

[11] Shohreh Deldari, Daniel V. Smith, Hao Xue, and Flora D. Salim, 'Time series change point detection with self-supervised contrastive predictive coding', in *WWW*, p. 3124–3135, (2021).

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'BERT: Pre-training of deep bidirectional transformers for language understanding', in *NAACL*, pp. 4171–4186, (2019).

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, 'An image is worth 16x16 words: Transformers for image recognition at scale', *ICLR*, (2021).

[14] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan, 'Time-series representation learning via temporal and contextual contrasting', in *IJCAI*, pp. 2352–2359, (2021).

[15] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi, 'Unsupervised Scalable Representation Learning for Multivariate Time Series', in *NeurIPS*, (2019).

[16] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom, 'Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding', in *ACM SIGKDD*, p. 387–395, (2018).

[17] Jeff Johnson, Matthijs Douze, and Hervé Jégou, 'Billion-scale similarity search with GPUs', *IEEE Transactions on Big Data*, **7**(3), 535–547, (2019).

[18] Eamonn Keogh, Jessica Lin, Sang-Hee Lee, and Helga Van Herle, 'Finding the most unusual time series subsequence: algorithms and applications', *Knowledge and Information Systems*, **11**(1), 1–27, (2007).

[19] Diederik P. Kingma and Jimmy Ba, 'Adam: A method for stochastic optimization', in *ICLR*, eds., Yoshua Bengio and Yann LeCun, (2015).

[20] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu, 'Revisiting time series outlier detection: Definitions and benchmarks', in *NeurIPS*, (2021).

[21] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng, 'Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks', in *ICANN*, pp. 703–716. Springer, (2019).

[22] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei, 'Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding', in *ACM SIGKDD*, (2021).

[23] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He, 'Generative adversarial active learning for unsupervised outlier detection', *IEEE Transactions on Knowledge and Data Engineering*, **32**(8), 1517–1528, (2019).

[24] J. Ma and S. Perkins, 'Time-series novelty detection using one-class support vector machines', in *IJCNN*, volume 3, pp. 1741–1745 vol.3, (2003).

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

[26] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff, 'Timenet: Pre-trained deep recurrent neural network for time series classification', *arXiv preprint arXiv:1706.08838*, (2017).

[27] Aditya P. Mathur and Nils Ole Tippenhauer, 'Swat: a water treatment testbed for research and training on ics security', in *CySWater*, pp. 31–36, (2016).

[28] Takaaki Nakamura, Makoto Imamura, Ryan Mercer, and Eamonn Keogh, 'Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives', in *ICDM*, pp. 1190–1195, (2020).

[29] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang, 'Time-series anomaly detection service at microsoft', in *ACM SIGKDD*, p. 3009–3017, (2019).

[30] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft, 'Deep one-class classification', in *ICML*, (2018).

[31] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar, 'A theoretical analysis of contrastive unsupervised representation learning', in *ICML*, (2019).

[32] Lifeng Shen, Zhuocong Li, and James Kwok, 'Timeseries Anomaly Detection using Temporal Hierarchical One-Class Network', in *NeurIPS*, volume 33, pp. 13016–13026, (2020).

[33] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei, 'Robust anomaly detection for multivariate time series through stochastic recurrent neural network', in *ACM SIGKDD*, p. 2828–2837, (2019).

[34] Shahroz Tariq, Sangyup Lee, Youjin Shin, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S. Woo, 'Detecting anomalies in space using multivariate convolutional lstm with mixtures of probabilistic pca', in *ACM SIGKDD*, p. 2123–2133, (2019).

[35] David MJ Tax and Robert PW Duin, 'Support vector data description', *Machine learning*, **54**(1), 45–66, (2004).

[36] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg, 'Unsupervised representation learning for time series with temporal neighborhood coding', in *ICLR*, (2021).

[37] Owen Vallis, Jordan Hochenbaum, and Arun Kejariwal, 'A novel technique for long-term anomaly detection in the cloud', p. 15. USENIX Association, (2014).

[38] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi, 'CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting', in *ICLR*, (2022).

[39] Renjie Wu and Eamonn Keogh, 'Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress', *IEEE Transactions on Knowledge and Data Engineering*, 1–1, (2021).

[40] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jie Chen, Zhaogang Wang, and Honglin Qiao, 'Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications', in *WWW*, p. 187–196, (2018).

[41] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long, 'Anomaly transformer: Time series anomaly detection with association discrepancy', *arXiv preprint arXiv:2110.02642*, (2021).

[42] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh, 'Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets', in *ICDM*, pp. 1317–1322, (2016).

[43] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu, 'Ts2vec: Towards universal representation of time series', in *AAAI*, pp. 8980–8987, (2022).

[44] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff, 'A transformer-based framework for multivariate time series representation learning', in *ACM SIGKDD*, p. 2114–2124, (2021).