Unsupervised Time Series Anomaly Prediction with Importance-based Generative Contrastive Learning

Kai Zhao[†], Zhihao Zhuang[‡], Chenjuan Guo[‡], Hao Miao[†], Yunyao Cheng[†], Bin Yang[‡]
[†]*Aalborg University*, Aalborg, Denmark
[‡]*East China Normal University*, Shanghai, China
{kaiz, haom, yunyaoc}@cs.aau.dk, {zhuangzhihao, cjguo, byang}@stu.ecnu.edu.cn

***Abstract*—We study a recent novel problem of time series anomaly prediction, which plays an essential role in many real-world scenarios, such as in-time environmental prevention and prompt maintenance of cyber-physical systems. However, existing time series anomaly prediction methods mainly require supervised training for better accuracy with plenty of manually labeled data, which are difficult to obtain in practice. Besides, during real-time deployment, unseen anomalies can occur which could differ from the labeled training data and make these supervised models fail to predict such new anomalies. In this paper, we provide a theoretical analysis and propose Importance-based Generative Contrastive Learning (IGCL) to address the unsupervised time series anomaly prediction. First, IGCL produces anomaly precursor patterns with our sampling diffusion-based generative module. Then, IGCL efficiently learns the contextual representations to extract temporal dependencies from the pair of normal time series and the anomaly precursors. With contrastive learning, IGCL can predict anomalies by identifying anomaly precursors that are the start of deviating from the normal to more future anomalies. To address the efficiency issues caused by the potential complex anomaly precursor combinations in multiple variables, we propose a memory bank with importance-based scores to adaptively store representative anomaly precursors and generate more complicated anomaly precursors. Extensive experiments on eight benchmark datasets show our method outperforms state-of-the-art baselines on the unsupervised time series anomaly prediction task. The appendix, codes, and data are available at https://github.com/zhkai/IGCL.**

***Index Terms*—Multi-variable Time Series, Anomaly Prediction, Generative Contrastive Learning**

I. Introduction

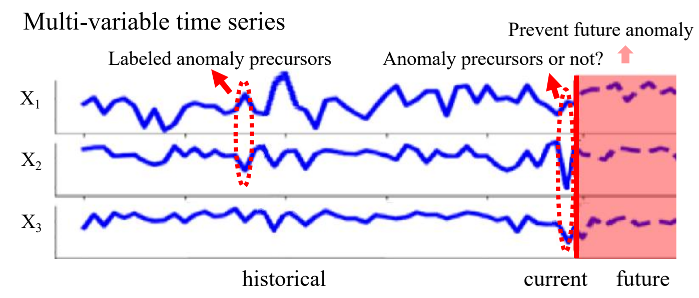
Many data collected from the real-world applications and scenarios can be modeled as the recordings of time-dependent observations, which form multi-variable time series [1]–[5]. For example, in server machine systems, multi-variable time series can record different runtime and monitoring indicators for different machines [6]–[8], and in environmental fields, multi-variable time series can record water or weather observations in different places [9]–[11].

Based on the current time series data, anomaly prediction makes the real-time judgment on whether there is the start of anomaly signals that could indicate more severe anomalies followed after the current time [12]–[15]. Anomaly prediction can help take in-time operations to prevent future severe anomalies, which is illustrated in Figure 1(a). Anomaly prediction plays an important role in many applications and real-world scenarios [12]–[19], such as health care [1], [13], cyber-physical systems maintenance [12], [14]–[18], and environmental prevention [19]. For example, anomaly prediction can help to timely handle pollution for drinking water security [14], [16], prevent more severe faults and failures for systems [15], [17], [18], and warn for earthquakes or extreme weathers [12], [19].

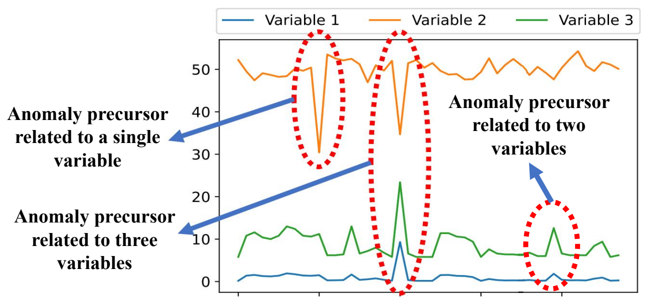
Despite the practical values, there exist only a few classification-based works [12]–[19] related to the time series anomaly prediction problem. These methods predict whether there are future anomalies by classifying whether the current time series contains anomaly signals that are the start of deviating from normal patterns to anomalies [12]–[15]. Such anomaly signals are called *anomaly precursors* [12], [14] of the more future anomalies. However, these existing anomaly prediction methods [12]–[19] heavily rely on supervised machine learning models to get more accurate anomaly prediction results, as anomaly signals in the precursors are just beginning abnormal and are difficult to identify. These supervised machine learning models need to train with manually labeled anomaly precursors and learn to classify anomaly precursors out from the normal time series, where different anomaly precursors are given in the training datasets, which is illuminated in Figure 1.

However, learning supervised models with labeled data is not feasible in many real-world scenarios, because manual labeling is impractical or at high cost [11], [20], [21]. Besides, unexpected anomalies [22]–[24] that never appeared before may occur during real-time deployment, which could not be classified by learning the existing labeled anomaly precursors in the training stage, and thus make the supervised models fail in practice [11], [20]–[22]. Therefore, unsupervised time series anomaly prediction methods become crucial for real-world applications. However, it has not been well studied nor theoretically analyzed due to its challenging nature.

Challenge 1: In the unsupervised time series anomaly prediction problem, there are no labeled anomaly precursor data for the models to learn temporal dependency features. Even though PAD [12] proposes to utilize unsupervised anomaly detection methods [25]–[27] and use specific thresholds on the anomaly scores to identify the anomaly precursors, their accuracy is limited compared to supervised learning with labeled anomaly precursors [13]–[16]. Without labeled data, these methods cannot learn the temporal dependency features from the pair of successive time series sub-sequences where


Multi-variable time series
Prevent future anomaly
Labeled anomaly precursors
Anomaly precursors or not?
X1
X2
X3
historical
current
future

(a) Supervised time series anomaly prediction learns from the training data with labeled anomaly precursors.



(b) Anomaly precursors have different kinds of combinations in correlated variables in the time series.

Fig. 1. Motivations.

data just begin to change from normal to the abnormal [13]–[16], which are crucial to the anomaly prediction tasks [13]–[16]. As a result, these existing methods cannot perform well for the unsupervised anomaly prediction tasks without learning such temporal dependencies, which is also experimentally validated in Section V.

Challenge 2: We are also facing very complex potential anomaly precursors that can have different combinations in the correlated variables in time series, which will result in an explosion of time and space complexity to generate all kinds of potential precursors. As shown in Figure 1(b), for any n-variable time series data, there could be as many as up to $O(2^n)$ potential anomaly precursor combinations [15], where the anomaly precursors could appear in each individual variable or in different combinations of variables [14], [15]. Thus, it is very challenging to learn the comprehensive temporal dependencies from the normal time series and each of these anomaly precursor combinations.

In this paper, we propose a novel approach **IGCL**, **I**mportance-based **G**enerative **C**ontrastive **L**earning for the unsupervised time series anomaly prediction problem to address the above challenges and improve accuracy without manually labeled anomaly precursors. IGCL encompasses three main modules: anomaly precursor pattern generation, positive and negative representations learning, and the memory bank.

For **Challenge 1**, we propose a generative contrastive learning architecture that produces potential anomaly precursors for the model to learn temporal dependency features. First, we proposed the diffusion-based distribution transformation and variance regularization loss for the anomaly precursor pattern generation module, to generate various kinds of potential anomaly precursor patterns as the labeled negative data from Gaussian noises. Second, we propose an overlapping window-based temporal convolution network (TCN) to efficiently extract the temporal dependencies from all pairs of time series sub-sequences and learn the positive and negative contextual representations. More specifically, the pairs of successive normal time series sub-sequences are positive samples, and the pairs of the normal sub-sequence and the anomaly precursor, which contain the start of temporal changes from normal to abnormal, are negative samples. Finally, our model distinguishes normal time series data and identifies the anomaly precursors with contrastive learning.

For **Challenge 2**, we propose a memory bank with importance-based scores to adaptively store representative anomaly precursors, which helps to generate more complex anomaly precursor combinations related to different variables. First, the anomaly precursor patterns can be generated efficiently as we consider generating anomaly precursors related to only one variable during each iteration. Second, the memory bank with a fixed small size K, which could be far smaller than $O(2^n)$, is initialized with the generated anomaly precursors related to one variable. Then, other one-variable anomaly precursor patterns are iteratively generated in following iterations and are injected into the previously generated anomaly precursors stored in the memory bank, to accumulate complex anomaly precursor combinations related to more variables. After the model optimization of each iteration, the memory bank only stores representative anomaly precursors with size K and pops out anomaly precursors that are already easy to distinguish according to our importance-based scores, to reduce the model complexity.

We summarize our contributions as follows.

- We propose generative contrastive learning architecture with diffusion-based distribution transformation and variance regularization loss to generate different anomaly precursors as negative data, and window-based TCN to efficiently learn temporal dependencies for anomaly prediction.
- We propose a memory bank with importance-based scores to store representative negative samples, which helps to generate more complicated anomaly precursor combinations related to different variables efficiently.
- Experiments on eight benchmarks from different domains show our method outperforms the state-of-the-art baselines in terms of both accuracy and efficiency.

II. Related Works

We review the relevant works on time series anomaly prediction, time series anomaly detection, contrastive learning, and generative learning. We also compare our method with the related works in Table I.

TABLE I
COMPARISONS ON REPRESENTATIVE METHODS THAT CAN BE APPLIED TO UNSUPERVISED TIME SERIES ANOMALY PREDICTION.

Methods	For time series	Unsupervised	Precursor Generation	Temporal Dependencies
[28]–[31]	✗	✓	✗	✗
[17]–[19]	✓	✗	✗	✗
[13]–[16]	✓	✗	✗	✓
[12], [20], [21], [25]–[27], [32]–[45]	✓	✓	✗	✗
Our IGCL	✓	✓	✓	✓

A. Time Series Anomaly Prediction

Based on the current time series data, anomaly prediction aims to make the real-time judgment on whether there are starting anomaly precursors where data begin to deviate from normal to future anomalies. Anomaly prediction plays an important role in many real-world scenarios [13]–[16], such as preventing severe faults and failures for systems [17], [18]. There are a few studies [12]–[19] related to this problem. FEP [19] extracts statistical information from manually labeled frequency-based features as the starting anomaly precursors to predict future anomalies. EADS [17], [18] and VFP [15] use graphs to extract correlations among variables to help classify the starting anomaly precursors related to different variables. MCDA [13] proposes multi-instance attention, dCNN [14] proposes deep convolutional neural networks and CHE [16] proposes hierarchical windows, to extract temporal dependency features. They learn temporal dependencies from the pair of the normal time series sub-sequences and the starting anomaly precursors which help better identify starting abnormal changes for anomaly prediction.

However, these existing methods heavily rely on supervised machine learning to get accurate anomaly prediction results. They are limited and fall short in the unsupervised time series anomaly prediction task for practical applications.

B. Time Series Anomaly Detection

There has been a lot of research on time series anomaly detection. Early works [22] utilize classification-based methods to detect anomalies in the historical time series data. Recent methods mainly focus on unsupervised settings based on the one-class classification theory [28], [29]. They can be further categorized into three categories [2]–[4]. The clustering-based methods [28], such as K-means [20], Deep-SVDD [29], and ITAD [32], aim to cluster normal samples within a center. THOC [33] uses a hierarchical RNN to learn dynamic clustering centers for time series data. The density estimation-based anomaly detection methods, such as LOF [30], assume abnormal samples lie in a low-density region. IForest [31] uses the tree model and DAGMM [34] uses the encoding probability model to measure the density or distribution. LSTM [21] proposes forecasting-based approaches that forecast values and based on the forecasting errors detect whether the time points are anomalies. Autoencoder [25], [35], [36] (AE) is a classic architecture used in reconstruction. Omni [37], InterFusion [38] and VQRAE [39] utilize the Variational AE and use reconstruction probability to detect anomalies. GANomaly [40] introduces a discriminator to generate normal data and detect anomalies based on reconstruction errors. CAE-Ensemble [26] proposes to ensemble different AEs for reconstruction. PUAD [27] uses meta-learning-oriented AE to reconstruct typical time series.

Generative Learning. The diffusion-based network is a generative model that generates data with denoising steps [46]. ImDiffusion [41], DiffAD [42] and D3R [43] utilize diffusion-based networks to generate normal time series data and detect anomalies based on reconstruction errors. However, these methods only learn the underlying normal data distribution, but cannot generate anomaly precursors.

Contrastive Learning. Contrastive learning learns useful representations by contrasting positive pairs against negative pairs with data augmentation [47]. Recently, contrastive learning has been proposed for anomaly detection, such as DCdetector [44], where two view representations of a normal timestamp should be similar, and contrastive errors are used to detect anomalies. ATransformer [45] combines the reconstruction errors and contrastive errors to detect anomalies.

PAD [12] proposes to utilize unsupervised anomaly detection methods and tune specific thresholds on the anomaly scores to classify the anomaly precursors for anomaly prediction. However, these methods do not have negative samples and only reconstruct or contrast positive samples. They cannot learn the temporal dependencies from the pair of normal time series and the anomaly precursors. Their performance is limited compared to supervised learning with labeled anomaly precursors as negative samples [13]–[16].

III. PRELIMINARIES

A. Problem Definition

We first formalize the unsupervised time series anomaly prediction problem. The notations are summarized in Table II.

Multi-variable Time Series. The multi-variable Time Series is represented as $\mathbf{X} = \langle \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T \rangle \in \mathbb{R}^{N \times T}$, where N is the number of variables collected at each timestamp, T is the total number of historical timestamps, $\mathbf{x}_t \in \mathbb{R}^N$ denotes observations of N-variable at timestamp t. We use $\mathbf{X}_{t+1:t+f} \in \mathbb{R}^{N \times h}$ to indicate the all N-variable data during the time window $[t+1, t+f]$, use $\mathbf{X}^i_{t+1:t+f} \in \mathbb{R}^h$ to indicate the observations of the i-th variable specifically, and use $\mathbf{X}^i_t \in \mathbb{R}^1$ to indicate the observation of the i-th variable at timestamp t specifically. We also use real-time T to denote the current timestamp.

Sub-sequence. A sub-sequence $\mathbf{X}_{t_1:t_2}$ of time series $\mathbf{X}$ is a continuous subset of the data in $\mathbf{X}$ that starts from t_1 and ends at t_2 with sequence length of $(t_2 - t_1 + 1)$. The set of b most recent sub-sequences before the current timestamp T, each with sequence length of $h+1$, is denoted as: $\mathbb{X}^b_{T,h} = \{\mathbf{X}_{T-h:T}, \mathbf{X}_{T-1-h:T-1}, \cdots, \mathbf{X}_{T-b+1-h:T-b+1}\}$.

The Anomaly Precursor. The current sub-sequence $\mathbf{X}_{T-h:T}$ is defined as the anomaly precursor, if $\mathbf{X}_{T-h:T}$ contain anomaly signals that are the start of deviating from normal patterns and are followed by more future anomalies in $\mathbf{X}_{T+1:T+f}$ [12]–[16], where h and f are the hyper-parameters for the look-back and look-forward windows.

Unsupervised Time Series Anomaly Prediction. We first formulate the time series anomaly prediction as follows. At any current real-time T and given the current observations $\mathbf{X}_{T-b+1-h:T}$, which can be also presented as a set of b most recent sub-sequences $\mathbb{X}^b_{T,h} = \{\mathbf{X}_{T-h:T}, \mathbf{X}_{T-1-h:T-1}, \cdots, \mathbf{X}_{T-b+1-h:T-b+1}\}$, the goal is to predict a real-time anomaly score $\hat{p}_T$ to denote whether there are anomaly precursors and will be more future anomalies in $\mathbf{X}_{T+1:T+f}$. The transformation of anomaly scores into binary labels is done by applying a threshold δ [12], *i.e.,* if $\hat{p}_T < \delta$ the model outputs that there are no anomaly precursors and future sub-sequence $\mathbf{X}_{T+1:T+f}$ will still be normal, and if $\hat{p}_T \geq \delta$ the model outputs that there are anomaly precursors and future sub-sequence $\mathbf{X}_{T+1:T+f}$ be more abnormal. Thus, the time series anomaly prediction problem is finding a model with a mapping function $\mathcal{F}$ as

$$\hat{p}_T = \mathcal{F}_\theta(\mathbb{X}^b_{T,h}), \tag{1}$$

where θ is the parameters of the model $\mathcal{F}$. For the unsupervised time series anomaly prediction problem, we do not have the true score p_T for training as there are no labeled data.

B. Theoretical Analysis

We theoretically analyze the unsupervised time series anomaly prediction problem with the information theory [48]. We utilize the Markov Chain [49], which is widely used in time series analytics [50], to model the sequence-based probability. We use random variables $\mathbf{H_{t-h-1}}$ and $\mathbf{H_t}$ to denote the data that the sub-sequences $\mathbf{X}_{t-2h-1:t-h-1}$ and $\mathbf{X}_{t-h:t}$ could be observed during any two successive time-windows $[t-2h-1, t-h-1]$ and $[t-h, t]$, respectively. Then, we define

$\mathcal{N} = P(\cdot|\mathbf{H_{t-h-1}} = \mathbf{X}_{t-2h-1:t-h-1}, \mathbf{H_t} \text{ is normal})$

to denote the Markov conditional distribution of the normal sub-sequence observed during time-window $[t-h, t]$, given that its previous observations during time-window $[t-2h-1, t-h-1]$ are $\mathbf{X}_{t-2h-1:t-h-1}$. Similarly, we define

$\mathcal{A} = P(\cdot|\mathbf{H_{t-h-1}} = \mathbf{X}_{t-2h-1:t-h-1}, \mathbf{H_t} \text{is anomaly precursor})$

to denote the Markov conditional distribution of the anomaly precursor observed during time-window $[t-h, t]$.

Theorem 1: Under the unsupervised setting, the future anomaly is unpredictable if we cannot identify the anomaly precursors in the current observations that follow the same distribution as the previous normal sub-sequences do.

TABLE II
NOTATIONS

Notation	Explanation
$\mathbf{x}_t$	The observed data at timestamp t from the N-variable time series $\mathbf{X}$
$\mathbf{X}_{t+1:t+f}$	The sub-sequence of the time series $\mathbf{X}$ from time $t+1$ to $t+f$, with sequence length of f
$\mathbf{X}^i_{t-h:t}$	The data of the i-th variable starting from $t-h$ to t
$\mathbb{X}^b_{T,h}$	The set of most recent b sub-sequences with length $h+1$ before the current timestamp T $\mathbb{X}^b_{T,h} = \{\mathbf{X}_{T-h:T}, \cdots, \mathbf{X}_{T-b+1-h:T-b+1}\}$
$N(\mu, \sigma^2)$	The multi-variable Gaussian distribution with mean vector μ and covariance matrix σ^2
$\mathbf{R}^i_{t-h:t}$	The anomaly precursor patterns for the i-th variable
S	The max steps for diffusion-based transformation
$\mathbf{v}_t$	The feature vector extracted from time series at timestamp t
z^+_t	The representations from $(\mathbf{X}_{t-2h-1:t-h-1}, \mathbf{X}_{t-h:t})$ to estimate $\mathcal{N} = P(\cdot|\mathbf{H_{t-h}}, \mathbf{H_t}$ is normal)
z^-_t	The representations from $(\mathbf{X}_{t-2h-1:t-h-1}, \mathbf{X}^-_{t-h:t})$ to estimate $\mathcal{A} = P(\cdot|\mathbf{H_{t-h}}, \mathbf{H_t}$ is an anomaly precursor)
K	The max size of the memory bank

Proof 1: As the current observations follow the same conditional distribution as the normal sub-sequence follows, for any model $\mathcal{F}$ we have:

$$\underset{\mathbf{H_T}\sim\mathcal{A}}{P}\left[\mathcal{F}(\mathbb{X}^b_{T,h}) = \hat{p}_T\right] - \underset{\mathbf{H_t}\sim\mathcal{N}}{P}\left[\mathcal{F}(\mathbb{X}^b_{T,h}) = \hat{p}_T\right] = 0. \tag{2}$$

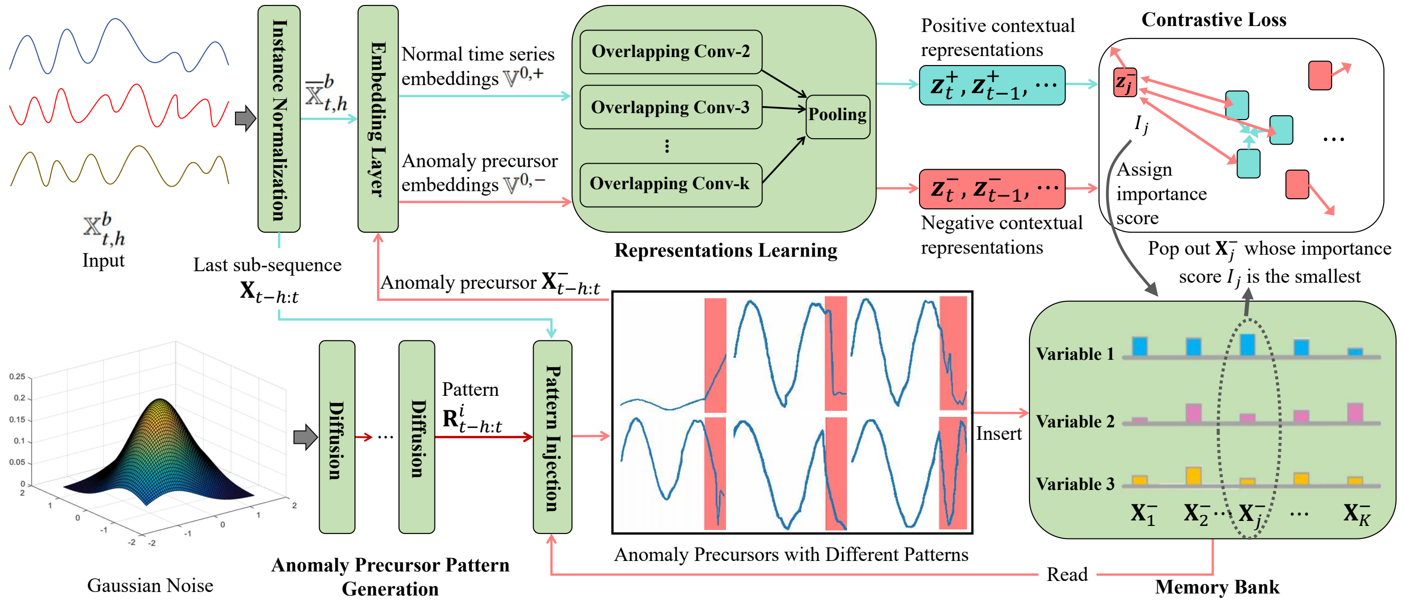
Thus, the predicted probability score for the anomaly precursor in the current observations is equal to that for the normal sub-sequence. Therefore, no unsupervised model can predict the future anomalies if we cannot identify the anomaly precursors in the current observations.

Theorem 2: Under the unsupervised setting, the future anomaly is predictable if we can identify its anomaly precursors in the current observations that follow a distribution different from the normal sub-sequence.

Proof 2: As the anomaly precursors follow a data distribution different from the normal sub-sequence, we have $\mathcal{N} \neq \mathcal{A}$. According to the Maximum Mean Discrepancy (MMD) theory [48], there exists at least one mapping function $\mathcal{F}$ satisfying:

$$\underset{\mathbf{H_t}\sim\mathcal{A}}{\mathbb{E}}\left[\mathcal{F}(\mathbb{X}^b_{t,h})\right] - \underset{\mathbf{H_t}\sim\mathcal{N}}{\mathbb{E}}\left[\mathcal{F}(\mathbb{X}^b_{t,h})\right] > 0 \tag{3}$$

Thus, the predicted probability score for the anomaly precursor is larger than that for the normal sub-sequence. Therefore, there exists at least one model $\mathcal{F}$ that can output a larger probability score to identify there are starting anomaly signals and the future time series is more like an anomaly if the anomaly precursors follow the distribution different from the normal sub-sequence. □



Instance Normalization
Embedding Layer
Normal time series embeddings $\mathbb{V}^{0,+}$
Anomaly precursor embeddings $\mathbb{V}^{0,-}$
Overlapping Conv-2
Overlapping Conv-3
Overlapping Conv-k
Pooling
Representations Learning
Positive contextual representations
$z_t^+, z_{t-1}^+, \cdots$
$z_t^-, z_{t-1}^-, \cdots$
Negative contextual representations
Contrastive Loss
z_j^-
I_j
Assign importance score
Pop out $\mathbf{X}_j^-$ whose importance score I_j is the smallest
$\mathbb{X}_{t,h}^b$
Input
Last sub-sequence $\mathbf{X}_{t-h:t}$
Anomaly precursor $\mathbf{X}_{t-h:t}^-$
Gaussian Noise
Diffusion
Pattern $\mathbf{R}_{t-h:t}^i$
Pattern Injection
Anomaly Precursor Pattern Generation
Anomaly Precursors with Different Patterns
Insert
Read
Variable 1
Variable 2
Variable 3
$\mathbf{X}_1^-$
$\mathbf{X}_2^-$
$\mathbf{X}_j^-$
$\mathbf{X}_K^-$
Memory Bank

Fig. 2. The overall architecture.

Based on the above analysis, we can only predict the future time series anomalies whose anomaly precursors just begin to deviate from the nearest previous normal sub-sequences under the unsupervised setting. However, we do not have labeled anomaly precursor data for the model to learn the temporal dependencies to estimate the Markov conditional distributions for $\mathcal{A} = P(\cdot|\mathbf{H_{t-h-1}}, \mathbf{H_t}$ is anomaly precursor) and $\mathcal{N} = P(\cdot|\mathbf{H_{t-h-1}}, \mathbf{H_t}$ is normal). Thus, the best potential model is that we generate different kinds of potential anomaly precursors to simulate the beginning of deviation from normal and then learn the temporal dependency features.

To address this problem, in Section IV-A, we propose the anomaly precursor pattern generation module to generate different kinds of potential anomaly precursors, *e.g.,* $\mathbf{X}_{t-h:t}^-$, from the Gaussian noises. In Section IV-B, we propose an overlapping window-based TCN to efficiently learn the contextual representations, which estimate the conditional distribution of normal sub-sequences $\mathcal{N}$ by learning the temporal dependency features from all pairs of successive normal sub-sequences, and estimate the conditional distribution of anomaly precursors $\mathcal{A}$ by learning the temporal dependency features from all pairs of successive normal sub-sequence and anomaly precursor. Last, the model distinguishes the normal data and anomaly precursors with our contrastive strategy with the learned contextual representations, which is shown in Section IV-C.

IV. Methodology

We present the overall architecture in Figure 2, which mainly consists of anomaly precursor pattern generation, positive and negative representations learning, contrastive loss, and the memory bank.

For any timestamp t for training or real-time inference, the model takes as input the set of b most recent sub-sequences $\mathbb{X}_{t,h}^b = \{\mathbf{X}_{t-h:t}, \mathbf{X}_{t-1-h:t-1}, \cdots\cdots, \mathbf{X}_{t-b+1-h:t-b+1}\}$. We first apply Instance Normalization [51] on the observation of each time series variable, *i.e.,* $\mathbf{X}_{t'}^i$ in the set $\mathbb{X}_{t,h}^b$, and output $\overline{\mathbb{X}}_{t,h}^b$, to learn stable features [52] from the time series data, as shown below:

$$\mathbf{X}_{t'}^i \leftarrow \gamma^i \left(\frac{\mathbf{X}_{t'}^i - \mathbb{E}^i(\mathbb{X}_{t,h}^b)}{\sqrt{Var^i(\mathbb{X}_{t,h}^b) + \epsilon}} \right) + \beta^i, \text{ for } 1 \le i \le N \quad (4)$$

where $Var^i(\mathbb{X}_{t,h}^b)$ is the variance value of the i-th variable, $\mathbb{E}^i(\mathbb{X}_{t,h}^b)$ is the expectation value of the i-th variable, ϵ is to avoid dividing by zero, and β^i and γ^i are learnable affine parameters to control the normalized mean value and variance value to for i-th variable differently.

Embedding layer takes as input the normalized $\overline{\mathbb{X}}_{t,h}^b$, maps the observations at each timestamp t', *i.e.,* $\mathbf{X}_{t'} \in \mathbb{R}^N$, to a d-dimensional embedding $\mathbf{v}_{t'} \in \mathbb{R}^d$, which extracts the dense feature from the N-variable normal time series data, and outputs a set of normal time series embeddings, *i.e.,* $\mathbb{V}^{0,+}$.

Anomaly precursor pattern generation takes as input a randomly generated Gaussian noise $x \sim N(0, I)$, and iteratively outputs different potential anomaly precursor patterns, *e.g.,* $\mathbf{R}_{t-h:t}^i$, for a randomly selected i-th time series variable.

Pattern injection module injects precursor patterns $\mathbf{R}_{t-h:t}^i$ into the last sub-sequence $\mathbf{X}_{t-h:t}$ in the selected i-th time series variable, and outputs a time series anomaly precursor, *i.e.,* $\mathbf{X}_{t-h:t}^- = \mathbf{X}_{t-h:t}^i + \mathbf{R}_{t-h:t}^i$, which is fed into the embedding layer to get anomaly precursor embeddings $\mathbb{V}^{0,-}$.

Positive and negative representations learning takes as input all possible pairs of successive sub-sequences from $\mathbb{V}^{0,+}$

and $\mathbb{V}^{0,-}$. For example, this module takes as input a pair of normal sub-sequences $\mathbf{X}_{t-2h-1:t-h-1}$ and $\mathbf{X}_{t-h:t}$ from $\mathbb{V}^{0,+}$, and outputs a positive contextual representation z_t^+ that learns their normal temporal dependencies. This module also takes as input a pair of a normal sub-sequence $\mathbf{X}_{t-2h-1:t-h-1}$ from $\mathbb{V}^{0,+}$ and an anomaly precursor $\mathbf{X}_{t-h:t}^-$ from $\mathbb{V}^{0,-}$, and outputs a negative contextual representation z_t^- that learns their abnormal temporal dependencies.

With our **contrastive loss**, the model distinguishes $\mathcal{N} = P(\cdot|\mathbf{H_{t-h-1}},\ \mathbf{H_t}$ is normal) and $\mathcal{A} = P(\cdot|\mathbf{H_{t-h-1}}, \mathbf{H_t}$ is anomaly precursor) by pushing z_t^- away from z_t^+. Intuitively, it clusters positive representations, *e.g.,* z_t^+, that belong to $\mathcal{N}$ together and dissociates negative representations, *e.g.,* z_t^-, that belong to $\mathcal{A}$ away.

The memory bank, with a fixed size of K, stores the representative anomaly precursors. The generated anomaly precursor patterns are iteratively injected into existing anomaly precursors in the memory bank, to accumulate more complex anomaly precursor combinations. The importance score I_j for the anomaly precursor $\mathbf{X}_j^-$ is calculated by its similarity with the normal samples. Finally, the memory bank pops out the anomaly precursor with the smallest importance score, which the model can already distinguish.

In the next sub-sections, we will introduce each module in more detail.

A. Anomaly Precursor Pattern Generation

Given a randomly selected i-th variable in the time series, *i.e.,* $\mathbf{X}_{t-h:t}^i$, the anomaly precursor pattern generation module takes as input a randomly generated Gaussian noise $x \in \mathbb{R}^h$, where $x \sim N(0, I)$ and h denotes the look-back window size, and outputs an anomaly precursor pattern $\mathbf{R}_{t-h:t}^i \in \mathbb{R}^h$. Pattern injection module then injects $\mathbf{R}_{t-h:t}^i$ into the last sub-sequence $\mathbf{X}_{t-h:t}$ in the selected i-th time series variable, and outputs a time series anomaly precursor that involves one variable, *i.e.,* $\mathbf{X}_{t-h:t}^- = \mathbf{X}_{t-h:t}^i + \mathbf{R}_{t-h:t}^i$.

Specifically, we propose sampling diffusion distribution transformation and a variance regularization loss to generate diverse kinds of potential anomaly precursors to simulate the beginning of deviation from normal with Gaussian noises.

The denoising diffusion probabilistic model consists of two processes, *i.e.,* the noise pollution process with S steps and its reverse denoising diffusion processes where S is the max diffusion steps. To be more specific, we use $x^0 = \mathbf{R}_{t-h:t}^i \in \mathbb{R}^h$ to denote a potential anomaly precursor pattern that can be more complicated than the Gaussian noise patterns.

The noise pollution process $q(x^s|x^{s-1})$ is shown as

$$x^s = \sqrt{1-\beta^s}x^{s-1} + \sqrt{\beta^s}\epsilon^s,\ \forall s \in \{1, 2, \cdots, S\}, \tag{5}$$

where $\epsilon^s \sim N(0, I)$ are the independent Gaussian pollutions to be added in different steps, I is the identity matrix, $0 < \beta^s < 1$ are the hyper-parameters control the amount of pollution added in each step. Thus, we have:

$$\begin{aligned} x^s &= \sqrt{1-\beta^s}x^{s-1} + \sqrt{\beta^s}\epsilon^s \\ &= \sqrt{1-\beta^s}(\sqrt{1-\beta^{s-1}}x^{s-2} + \sqrt{\beta^{s-1}}\epsilon^{s-1}) + \sqrt{\beta^s}\epsilon^s \\ &= \cdots \\ &= (\sqrt{1-\beta^s}\sqrt{1-\beta^{s-1}}\cdots\sqrt{1-\beta^1})x^0 + \\ &\quad (\sqrt{1-\beta^s}\sqrt{1-\beta^{s-1}}\cdots\sqrt{1-\beta^2})\sqrt{\beta^1}\epsilon^1 + \\ &\quad (\sqrt{1-\beta^s}\sqrt{1-\beta^{s-1}}\cdots\sqrt{1-\beta^3})\sqrt{\beta^2}\epsilon^2\ + \cdots + \\ &\quad \sqrt{1-\beta^s}\sqrt{\beta^{s-1}}\epsilon^{s-1} + \sqrt{\beta^s}\epsilon^s \end{aligned} \tag{6}$$

For any independent Gaussian distributions $\epsilon_1 \sim N(\mu_1, \sigma_1^2)$ and $\epsilon_2 \sim N(\mu_2, \sigma_2^2)$, we have the following properties:

$$a\epsilon_1 + b \sim N(a\mu_1 + b, a^2\sigma_1^2), \tag{7}$$

$$\epsilon_1 + \epsilon_2 \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2). \tag{8}$$

We also have:

$$\begin{aligned} 1 = &\left(\sqrt{1-\beta^s}\sqrt{1-\beta^{s-1}}\cdots\sqrt{1-\beta^1}\right)^2 + \\ &\left(\sqrt{1-\beta^s}\sqrt{1-\beta^{s-1}}\cdots\sqrt{1-\beta^2}\right)^2\beta^1 + \\ &\left(\sqrt{1-\beta^s}\sqrt{1-\beta^{s-1}}\cdots\sqrt{1-\beta^3}\right)^2\beta^2\ + \cdots + \\ &(1-\beta^s)\beta^{s-1} + \beta^s, \end{aligned} \tag{9}$$

Thus, we have:

$$x^s = \sqrt{\widetilde{\alpha}^s}x^0 + \sqrt{1-\widetilde{\alpha}^s}\epsilon, \tag{10}$$

where $\epsilon \sim N(0, I)$ and $\sqrt{\widetilde{\alpha}^s} = \sqrt{1-\beta^s}\ \sqrt{1-\beta^{s-1}}\ \cdots \sqrt{1-\beta^1}$. Then, for enough S steps and proper $0 < \beta^s < 1$, $\sqrt{\widetilde{\alpha}^s}$ will be close to 0, and the output from the noise pollution process, *i.e.,* x^S, will be gradually corrupted into random Gaussian noises.

Our anomaly precursor pattern generation module is based on the denoising diffusion processes $p_\theta(x^{s-1}|x^s)$, by reversing random Gaussian noise x^S step by step to produce more complicated potential anomaly precursor patterns:

$$x^{s-1} \sim p_\theta(x^{s-1}|x^s) = \frac{1}{\sqrt{1-\beta^s}}\left(x^s - \sqrt{\beta^s}\varepsilon_\theta(x^s)\right), \tag{11}$$

where $\varepsilon_\theta(x^s) \sim N(\mu_s, {\sigma_s}^2)$, and μ_s is the mean value parameter and σ_s is the variance value parameter to be learned. We use Multi-layer Perceptron (MLP) layers to model with μ^s and σ^s:

$$\mu^s, \sigma^s = MLP\big(concat(x^s, s, \mathbf{X}_{t-h:t}^i)\big),\ \forall s \in \{S, \cdots, 1\}, \tag{12}$$

where x^S is a random Gaussian noise, x^s is the denoised state after each step, and $\mathbf{X}_{t-h:t}^i$ denotes this process aiming to generate anomaly precursor patterns for the i-th variable. We have $\sigma^s\varepsilon + \mu^s \sim N(\mu_s, {\sigma_s}^2)$, where $\varepsilon \sim N(0, I)$, and $\sigma^s\varepsilon + \mu^s$ is derivable with respective to μ^s and σ^s which are the outputs from the MLP layers. Thus, the final denoising diffusion process is

$$x^{s-1} = \frac{1}{\sqrt{1-\beta^s}}\left(x^s - \sqrt{\beta^s}(\sigma^s\varepsilon + \mu^s)\right), \tag{13}$$

where we randomly sample the Gaussian noise x^S, and apply the denoising diffusion process step by step to get the anomaly precursor pattern $\mathbf{R}_{t-h:t}^i = x^0$.

We also propose a regularization loss for parameters $N(\mu_s, {\sigma_s}^2)$, where the variance ${\sigma_s}^2$ controls the diversity of generated anomaly precursor patterns and the mean value μ_s and the variance ${\sigma_s}^2$ both control the degree of deviation from normal. First, the variance ${\sigma_s}^2$ should not be too large, as a large variance will produce severe anomalies rather than anomaly precursors with starting deviation from normal. Second, ${\sigma_s}^2$ should not be too small, as a small variance will generate less diverse [46] anomaly precursor patterns. Therefore, we propose to use the Kullback-Leibler divergence as a regularization loss function to make $N(\mu_s, {\sigma_s}^2)$ be close to $N(\mu_s, I)$ where the variance ${\sigma_s}^2$ should be close to the unit normal variance I which is neither too large nor too small [53]. For any $N(\mu, \sigma^2)$, $KL\Big(N(\mu,\sigma^2)\Big\|N(\mu,I)\Big) = \frac{1}{2}\Big(-\log\sigma^2+\sigma^2-1\Big)$. The proof is as follows.

$$\begin{aligned}
&KL\Big(N(\mu,\sigma^2)\Big\|N(\mu,I)\Big)\\
&=\int \frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2}\left(\log\frac{e^{-(x-\mu)^2/2\sigma^2}/\sqrt{2\pi\sigma^2}}{e^{-(x-\mu)^2/2}/\sqrt{2\pi}}\right)dx\\
&=\int \frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2}\log\\
&\qquad\qquad\left\{\frac{1}{\sqrt{\sigma^2}}\exp\left(\frac{1}{2}(x-\mu)^2(1-\frac{1}{\sigma^2})\right)\right\}dx\\
&=\frac{1}{2}\int \frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/2\sigma^2}\Big[-\log\sigma^2+(x-\mu)^2(1-\frac{1}{\sigma^2})\Big]dx\\
&=\frac{1}{2}\Big(-\log\sigma^2+\sigma^2(1-\frac{1}{\sigma^2})\Big)=\frac{1}{2}\Big(-\log\sigma^2+\sigma^2-1\Big).
\end{aligned}\tag{14}$$

Thus, our analytical solution of the regularization loss is:

$$\mathcal{L}_r=\sum_{s=1}^{S}\frac{1}{2}\Big(-\log\sigma_s^2+\sigma_s^2-1\Big)\tag{15}$$

B. Positive and Negative Representations

According to Section III-B, we propose positive and negative representations learning to learn contextual representations from the pairs of successive sub-sequences in the Markov Chain to extract their temporal dependencies. However, there will be a lot of calculations when considering all possible pairs of successive sub-sequences using traditional neural networks. Taking the Transformer [54] as an example:

$$\begin{aligned}
z_t^+ &= Transformer(\mathbf{X}_{t-2h-1:t-h-1},\mathbf{X}_{t-h:t}),\\
z_{t-1}^+ &= Transformer(\mathbf{X}_{t-2h-2:t-h-2},\mathbf{X}_{t-h-1:t-1}),\\
&\cdots\\
z_{t-1}^- &= Transformer(\mathbf{X}_{t-2h-2:t-h-2},\mathbf{X}^-_{t-h-1:t-1}),\\
z_t^- &= Transformer(\mathbf{X}_{t-2h-1:t-h-1},\mathbf{X}^-_{t-h:t}),
\end{aligned}\tag{16}$$

where the $Transformer$ layer learns the temporal features from the time series data across different timestamps. As a result, each $z_{t'}^+$ with $t'\in[t-b+1-h,t]$ is learned as the contextual representation for the pair of the successive normal sub-sequences $\mathbf{X}_{t'-2h-1:t'-h-1}$ and $\mathbf{X}_{t'-h:t'}$ to represent for $\mathcal{N}$. Each $z_{t'}^-$ is learned as the contextual representation for the pair of a normal sub-sequence $\mathbf{X}_{t'-2h-1:t'-h-1}$ and an anomaly precursor $\mathbf{X}^-_{t'-h:t'}$ to represent for $\mathcal{A}$. The computational complexity using Transformer to get each $z_{t'}^+$ is $O(h^2)$, and the total complexity is $O(bh^2)$, where h denotes the size of the look-back time-window and b is the total number of most recent sub-sequences used as input.

To handle this problem, we propose an overlapping window-based TCN to learn contextual representations from the all possible pairs of successive sub-sequences and extract their temporal dependencies at once.

To be more specific, our embedding layer firstly maps the normalized time series data, *i.e.,* $\mathbf{X}_{t'}\in\mathbb{R}^N$ with $t'\in[t-b+1-h,t]$ from the set $\mathbb{X}^b_{t,h}$, into the d-dimensional feature vectors $\mathbf{v}_{t'}\in\mathbb{R}^d$, which aim to extract the dense features from time series for each timestamp. Further, the input to the embedding layer can also be coupled with other auxiliary attributes, such as the encoding of timestamps. The auxiliary attributes at timestamp t' are represented as $\mathbf{a}_{t'}\in\mathbb{R}^F$, where F is the total dimensions of the auxiliary attributes. Thus, the embedding layer concatenates the original time series data $\mathbf{X}_{t'}$ and the auxiliary attributes

$$\mathbf{f}_{t'}=concat(\mathbf{X}_{t'},\mathbf{a}_{t'})\ \forall t'\in[t-b+1-h,t],\tag{17}$$

as its input, and maps it to the feature vector by:

$$\mathbf{v}_{t'}=\sigma(W_e\,\mathbf{f}_{t'}),\ \forall t'\in[t-b+1-h,t],\tag{18}$$

where $\mathbf{v}_{t'}\in\mathbb{R}^d$ is the learned feature vector, σ is an activation function, and $W_e\in\mathbb{R}^{d\times(N+F)}$ is the learnable neural network parameters which extract the feature from the time series data and the auxiliary attributes at each timestamp separately. Following the same way, we can also get the feature vector $\mathbf{v}^-_{t'}\in\mathbb{R}^d$, where $t'\in[t-h,t]$, from the generated anomaly precursor $\mathbf{X}^-_{t-h:t}$.

Then, our overlapping window-based TCN uses the feature vectors $\mathbf{v}_{t'}$ and $\mathbf{v}^-_{t'}$ to learn the temporal dependencies for all possible pairs of successive sub-sequences at once. The temporal convolution operation slides over timestamps by skipping feature vectors with a certain dilation factor in different layers, as illustrated in Figure 3. More specifically, this module takes as input all the feature vectors, *i.e.,* $\mathbf{v}_{t'}$ with $t'\in[t-b+1-h,t]$ and $\mathbf{v}^-_{t'}$ with $t'\in[t-h,t]$:

$$\begin{aligned}
\mathbb{V}^{0,+}&=\big\langle\mathbf{v}_{t-b+1-h},\cdots,\mathbf{v}_{t-h-1},\mathbf{v}_{t-h},\cdots,\mathbf{v}_t\big\rangle,\\
\mathbb{V}^{0,-}&=\big\langle\mathbf{v}_{t-b+1-h},\cdots,\mathbf{v}_{t-h-1},\mathbf{v}^-_{t-h},\cdots,\mathbf{v}^-_t\big\rangle,
\end{aligned}\tag{19}$$

and uses the learnable convolution $filter\in\mathbb{R}^k$ with the kernel size of k to extract features and learn the temporal dependencies by:

$$\begin{aligned}
\mathbb{V}^{l,+}(t')&=\sum_{i=0}^{k-1}filter(i)\mathbb{V}^{l-1,+}(t'-k^{l-1}\times i),\ \text{for } 1\le l\le L,\\
\mathbb{V}^{l,-}(t')&=\sum_{i=0}^{k-1}filter(i)\mathbb{V}^{l-1,-}(t'-k^{l-1}\times i),\ \text{for } 1\le l\le L,
\end{aligned}\tag{20}$$

where k^{l-1} is the dilation factor in the layer $l-1$, $\mathbb{V}^{l,+}(t')$ and $\mathbb{V}^{l,-}(t')$ is the learned dependency features in the layer

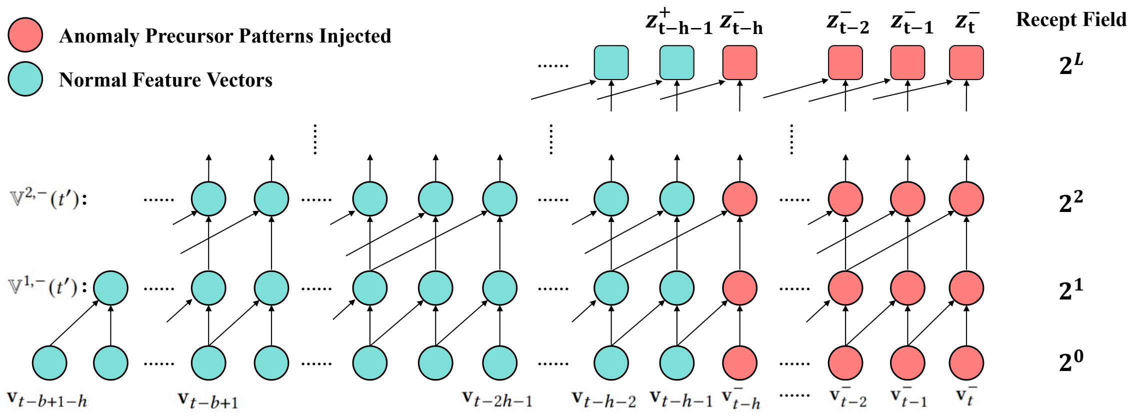

Anomaly Precursor Patterns Injected
Normal Feature Vectors
Recept Field
2^L
2^2
2^1
2^0
$\mathbb{V}^{2,-}(t')$:
$\mathbb{V}^{1,-}(t')$:
z^+_{t-h-1} z^-_{t-h} z^-_{t-2} z^-_{t-1} z^-_t
$v_{t-b+1-h}$ v_{t-b+1} v_{t-2h-1} v_{t-h-2} v_{t-h-1} v^-_{t-h} v^-_{t-2} v^-_{t-1} v^-_t

Fig. 3. The overlapping window-based temporal convolutional network with a kernel size k of 2.

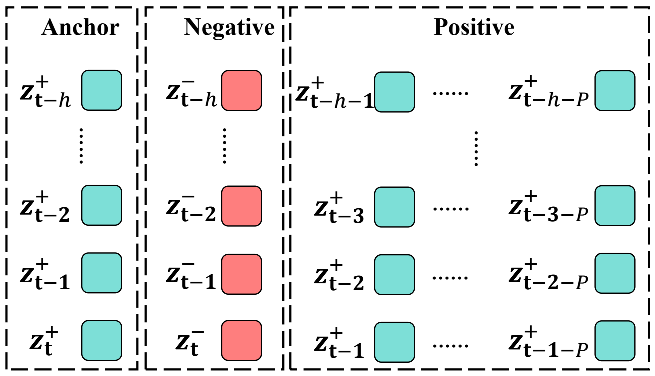

Anchor
Negative
Positive
z^+_{t-h} z^+_{t-2} z^+_{t-1} z^+_t
z^-_{t-h} z^-_{t-2} z^-_{t-1} z^-_t
z^+_{t-h-1} z^+_{t-h-P} z^+_{t-3} z^+_{t-3-P} z^+_{t-2} z^+_{t-2-P} z^+_{t-1} z^+_{t-1-P}

Fig. 4. The construction of contrastive pairs.

l for timestamp t', and $L \sim O(log(h))$ is the max number of TCN layers whose recept field is large than the whole window size of the pair of successive sub-sequences. We use different kernel sizes to extract features in parallel and pool them together, to capture different kinds of temporal dependencies with different kernel scales. Finally, it outputs all contextual representations for all pairs of successive sub-sequences at once, *e.g.,* $z_t^+ = \mathbb{V}^{L,+}(t)$, $z_{t-1}^+ = \mathbb{V}^{L,+}(t-1)$, $z_t^- = \mathbb{V}^{L,-}(t)$ and $z_{t-1}^- = \mathbb{V}^{L,-}(t-1)$.

In this way, our total complexity is reduced to $O\big((b+h)log(h)\big)$. This is because our TCN learns dependency features $\mathbb{V}^{l,+}(t')$ and $\mathbb{V}^{l,-}(t')$ at each timestamp t' only once in each l-th layer, and then $\mathbb{V}^{l,+}(t')$ and $\mathbb{V}^{l,-}(t')$ are used for different overlapping sub-sequences in the next layer.

C. The Objective Function

We propose a contrastive strategy to distinguishes $\mathcal{N}$ and $\mathcal{A}$ by pushing z_t^- away from z_t^+. Intuitively, it clusters positive representations, *e.g.,* z_t^+, that belong to $\mathcal{N}$ together and dissociates negative representations, *e.g.,* z_t^-, that belong to $\mathcal{A}$ away. The construction of contrastive pairs is illustrated in Figure 4. Each anchor z_t^+ is the contextual representation from the pair of the successive normal sub-sequences before timestamp t, which represents for $\mathcal{N}$. Thus, we make the anchor z_t^+ similar to the positive representations z_{t-j}^+ with $1 \le j \le P$, which are from the most nearby successive normal sub-sequences before timestamp t and they follow the normal data distribution. In order to avoid data leakage and be consistent with the inference stage, for each anchor z_t^+ the positive samples are always from the previous timestamps. Meanwhile, the anchor z_t^+ should be different from the negative representation z_t^-, which learns the temporal dependencies from normal to the generated anomaly precursors and follows the anomaly precursors distribution. Thus, our contrastive loss function is:

$$\mathcal{L}_c = \sum_{i=t}^{t-h} -log \Big[\frac{\sum_{j=1}^{P} exp\big(Sim(z_i^+, z_{i-j}^+)/\tau\big)}{exp\big(Sim(z_i^+, z_i^-)/\tau\big) + \sum_{j=1}^{P} exp\big(Sim(z_i^+, z_{i-j}^+)/\tau\big)}\Big], \tag{21}$$

where τ is the temperature parameter to control the softmax strength, P is the hyper-parameter for the number of positive samples used, and Sim is the similarity measurement, *e.g.,* Euclidean Distance or Cosine Similarity, to distinguish whether the contextual representations belong to the same distributions.

Finally, together with the regularization loss, we train the model using gradient descent [55] with the loss function by:

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_r, \tag{22}$$

where λ is to control the strength of regularization.

D. Memory Bank

We propose to use a memory bank to store previously generated anomaly precursors and combine them with the current anomaly precursor pattern $\mathbf{R}_{t-h:t}^i$, to accumulate more complex anomaly precursor combinations related to more than one variable.

To be more specific, the memory bank M has a fixed size of K and $K \ll O(2^n)$:

$$\mathbf{M} = \{\mathbf{X}_1^-, \mathbf{X}_2^-, \cdots, \mathbf{X}_K^-\}, \tag{23}$$

where each $\mathbf{X}_j^-$ denotes a stored anomaly precursor. The currently generated anomaly precursor pattern $\mathbf{R}_{t-h:t}^i$ which

TABLE III
DETAILS OF BENCHMARK DATASETS.

Dataset	Domain	N	#Training (Unlabeled)	#Validation (Unlabeled)	#Test (Labeled)	Anomaly Rate(%)
SMD	server machine	38	566,724	141,681	708,420	4.16
PSM	application server	25	105,885	26,398	87,841	27.8
MSL	NASA space sensor	55	46,653	11,663	73,729	10.72
SMAP	NASA space sensor	25	108,146	27,036	427,617	13.13
SWAN	space solar weather	38	48,000	12,000	60,000	32.6
SWaT	water treatment	51	396,000	99,000	449,919	12.14
GECCO	water quality	9	55,408	13,852	69,261	1.1
UCR	various domains	1	1,790,679	447,670	6,143,541	0.6

is only related to the i-th variable, is also injected into the anomaly precursors in the memory bank:

$$\mathbf{X}_j^- \leftarrow \mathbf{X}_j^- + \mathbf{R}_{t-h:t}^i, \tag{24}$$

and thus we can get anomaly precursors that involve more than one variable. Then, similar to Equation (19), we can get more negative contextual representations, *e.g.,* $z_{t,j}^-$ and $z_{t-1,j}^-$, from the anomaly precursor $\mathbf{X}_j^-$. The contrastive loss $\mathcal{L}_c$ is updated to

$$\sum_{i=t}^{t-h} -log \frac{\sum_{j=1}^{P} exp\big(Sim(z_i^+, z_{i-j}^+)/\tau\big)}{\sum_{j=0}^{K} exp\big(Sim(z_i^+, z_{i,j}^-)/\tau\big) + \sum_{j=1}^{P} exp\big(Sim(z_i^+, z_{i-j}^+)/\tau\big)} \tag{25}$$

where $z_{i,0}^- = z_i^-$ is the negative contextual representations from the current anomaly precursor $\mathbf{X}_{t-h:t}^-$, and $z_{i,j}^-$ is the negative contextual representations from the j-th anomaly precursor $\mathbf{X}_j^-$ stored in the memory bank.

Finally, the current anomaly precursor $\mathbf{X}_{t-h:t}^-$ is inserted into the memory bank which will have a size of $K+1$, *i.e.,* $\mathbf{X}_0^- = \mathbf{X}_{t-h:t}^-$. Instead of using a first-in-first-out strategy to maintain the memory bank with a fixed size of K, we propose to pop out the anomaly precursor based on an importance score. The intuition is that the memory bank should store the hard and important negative samples for further model training, *i.e.,* the anomaly precursors that are not yet distinguished by the model, and should pop out not important samples, *i.e.,* the anomaly precursors that are already dissociated away from normal. The importance score I_j for the anomaly precursor $\mathbf{X}_j^-$ is calculated by:

$$I_j = \sum_{i=t}^{t-h} Sim(z_i^+, z_{i,j}^-). \tag{26}$$

Thus, we pop out the anomaly precursor with the smallest importance score after model optimization.

E. Inference

For real-time T in the inference stage, we output $\hat{p}_T$ to predict how likely there are anomaly precursors currently and will be more future anomalies in $\mathbf{X}_{T+1,T+f}$:

$$\hat{p}_T = \sum_{j=1}^{K} Sim(z_t^+, z_{t,j}^-) - \sum_{j=1}^{P} Sim(z_t^+, z_{t-j}^+), \tag{27}$$

where K is the size of the memory bank storing anomaly precursors as negative samples, and P is the hyper-parameter for the number of positive samples.

V. EXPERIMENTS

In this section, we conduct extensive experiments to empirically evaluate the unsupervised time series anomaly prediction problem on eight real-world benchmark datasets, to justify the effectiveness and efficiency of our model and compare it with the state-of-the-art baseline methods. We present all the experimental results in terms of overall comparison, ablation studies, parameter sensitivity, overall runtime, and total parameters used, and analyze our model in more detail. We also visualize the anomaly prediction with real-world data.

A. Experimental Settings

1) Datasets: Following existing works [12], [26], [56], we validate the performance of our method with eight real-world benchmark datasets:

- SMD [37] (Server Machine Dataset) is collected from a computing cluster, where different server machines stack accessed traces of resource utilization with 38 dimensions.
- PSM [6] (Pooled Server Metrics) is a 25-dimension dataset collected by eBay that shows the performance of multiple application servers.
- MSL [57] (Mars Science Laboratory Rover) is from the spacecraft monitoring systems and collected by NASA. It shows the health check-up data of the Mars rover sensors.
- SMAP [57] (Soil Moisture Active Passive) is a 25-dimension dataset collected by NASA, which contains soil samples and telemetry data.
- SWAN [9] is extracted from solar photospheric vector magnetograms in the Spaceweather, which is the benchmark dataset used in NeurIPS Competition Track.
- SWaT [45] (Secure Water Treatment) is collected from sensors of the critical infrastructure systems under continuous operations for secure water treatment.

TABLE IV
ANOMALY PREDICTION RESULTS OF DIFFERENT METHODS. WE REPORT PRECISION, RECALL, AND F1-SCORE, PRESENTED IN PERCENTAGES.

Dataset	PSM			SMAP			SWAN			SWaT			GECCO		
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DAGMM	39.27	36.14	37.64	11.91	22.00	15.51	52.73	39.46	45.14	82.14	61.72	70.48	47.50	27.64	34.95
IForest	39.39	38.82	39.10	11.79	18.67	14.45	53.85	39.07	45.28	75.39	62.22	68.17	42.75	29.89	35.18
K-means	38.89	39.04	38.96	13.05	21.37	16.20	54.04	42.30	47.45	82.13	62.50	70.98	47.94	29.18	36.28
Deep-SVDD	36.63	36.30	36.46	11.77	17.09	13.94	52.39	39.18	44.83	81.29	61.95	70.31	46.09	28.45	35.18
THOC	37.64	38.93	38.27	12.61	21.89	15.99	54.93	40.02	46.30	82.42	62.45	71.06	46.41	29.49	36.09
DCdetector	28.97	12.05	17.02	8.04	11.30	9.40	21.04	10.94	14.40	46.45	36.36	40.79	1.76	3.46	2.33
PAD	31.23	33.05	32.11	11.01	22.37	14.76	50.13	37.61	42.98	74.83	59.09	66.04	44.38	28.17	34.46
ATransformer	30.56	34.64	32.47	11.40	22.80	15.20	52.20	38.75	44.48	75.00	60.50	66.97	43.62	27.92	34.05
Omni	39.45	40.88	40.15	11.90	23.39	15.77	54.83	41.00	46.92	85.73	58.57	69.59	50.55	29.65	37.38
GANomaly	37.02	43.06	39.81	12.18	23.41	16.02	54.44	40.91	46.72	83.99	59.77	69.84	50.06	29.23	36.91
PUAD	39.91	42.38	41.11	15.20	27.20	19.50	54.91	41.19	47.07	85.68	58.05	69.21	50.76	29.58	37.38
LSTM	39.20	40.31	40.02	11.68	22.65	15.41	54.42	40.88	46.69	66.76	68.36	67.55	46.67	30.92	37.20
CAE-Ensemble	40.18	41.99	41.07	15.88	26.68	19.91	55.98	41.52	47.68	88.61	59.90	71.48	51.67	29.92	37.90
D3R	40.73	42.21	41.46	15.73	26.89	19.85	55.32	42.64	48.16	84.13	61.85	71.29	51.24	30.91	38.56
IGCL	42.31	46.97	**44.52**	17.48	28.02	**21.53**	59.20	44.94	**51.09**	84.98	63.54	**72.71**	52.10	32.77	**40.23**

- GECCO [9] is a 9-dimension dataset collected from the cyber-physical systems showing the drinking water quality.
- UCR [56] is a 1-dimension dataset containing 250 sub-datasets from various domains.

These datasets are representative open benchmarks for multi-variable and uni-variable time series anomaly tasks, and thus we evaluate the unsupervised time series anomaly prediction problem on these datasets. More statistical information and details can be seen in Table III, where N is the number of variables observed at each timestamp. We follow the same training-validation-test splits as in the original papers [6], [9], [37], [45], [57].

2) Baselines: We compare our method with the unsupervised anomaly prediction method PAD [12]. We also modify more state-of-the-art unsupervised anomaly detection methods for the real-time anomaly prediction task following PAD:

- Density estimation-based methods [30]: IForest [31] and DAGMM [34].
- Clustering-based methods: K-means [20], Deep-SVDD [29] and THOC [33].
- Contrastive-based methods: DCdetector [44] and PAD [12].
- Forecasting or reconstruction-based methods: D3R [43], Omni [37], LSTM [21], ATransformer [45], GANomaly [40], CAE-Ensemble [26] and PUAD [27].

We introduce the details of all baselines in Section II. We employ the official open-source implementations of baseline methods and have carefully tuned their hyper-parameters based on the recommendations from the original papers. We do not compare [17], [18] as we cannot access their codes. The experiments are conducted on an Ubuntu 18.04.5 LTS system server with Intel(R) Xeon(R) Gold 5215 CPU and NVIDIA Quadro RTX 8000 GPU. All deep learning models are executed with Pytorch 1.2.0. The size of the default look-forward window f is 4, and we also evaluate with larger look-forward windows of $\{4, 8, 12\}$. We use Adam optimizer with the default parameter and the learning rate is 0.0001. The look-back window size h and the number of sub-sequences b are tuned from $\{8, 16, 32, 64\}$. The memory bank size K and the number of positive samples P are tuned from $\{12, 16, 20, 24\}$. The λ which controls the strength of regularization is tuned from $\{0.5, 1\}$. We use three different kernel sizes, *i.e.,* $\{2, 3, 5\}$, for our overlapping TCN. Following [21], we use the same strategy to choose the threshold for all methods.

TABLE V
ANOMALY PREDICTION RESULTS ON THE REST DATASETS. ALL RESULTS ARE PRESENTED IN PERCENTAGES.

Dataset	SMD			MSL		
Metric	P	R	F1	P	R	F1
DAGMM	14.74	15.37	15.05	13.94	17.04	15.33
IForest	15.25	18.28	16.63	13.82	17.11	15.29
Deep-SVDD	14.99	15.78	15.37	12.94	16.04	14.32
THOC	16.90	16.79	16.84	14.37	16.22	15.24
DCdetector	3.71	9.03	5.26	2.35	3.46	2.80
PAD	11.23	18.76	14.05	14.37	13.22	13.77
ATransformer	10.52	19.42	13.65	14.62	12.42	13.43
LSTM	14.21	16.42	15.24	16.18	15.28	15.72
Omni	16.22	18.19	17.15	13.66	21.66	16.75
GANomaly	15.06	21.39	17.68	15.36	17.30	16.27
PUAD	16.37	21.85	18.72	20.04	17.60	18.74
CAE-Ensemble	18.41	19.51	18.94	20.35	18.27	19.26
D3R	15.78	19.33	17.38	19.99	20.32	20.15
IGCL	17.39	25.90	**20.81**	21.97	22.81	**22.38**

3) Evaluation Metrics: Following the evaluation methodology in existing works [12], [26], we validated the performance of time series anomaly prediction models with Precision (P), Recall (R), F1-score (F1), where larger values indicate higher model performance. We do not use the point adjustment (PA) metrics, as recent works have demonstrated that PA can lead to faulty performance evaluations [58], [59]. We also use the Area Under the Curve of the Receiver Operating Characteristic (AUC-ROC) [60] as a metric that enables evaluation without choosing the threshold.

TABLE VI
ANOMALY PREDICTION RESULTS OF DIFFERENT METHODS WITH MORE METRICS, PRESENTED IN PERCENTAGES.

Dataset	PSM	SMAP	SWAN	SWaT	GECCO	SMD	MSL	UCR			
Metrics	AUC-ROC							P	R	F1	AUC-ROC
DCdetector	50.91	43.98	51.00	65.57	50.79	50.44	30.34	12.59	10.90	11.68	48.65
PAD	51.02	50.13	50.85	61.35	49.71	49.66	48.28	29.92	20.35	24.22	51.67
ATransformer	54.91	53.70	51.95	75.63	50.39	59.30	50.21	32.07	20.86	25.28	51.68
CAE-Ensemble	63.68	58.76	65.41	81.71	52.94	67.99	51.26	34.15	21.47	26.37	58.26
D3R	63.93	59.61	66.13	80.62	53.09	66.83	51.83	39.10	20.77	27.13	60.11
IGCL	**65.75**	**61.92**	**68.92**	**82.39**	**54.02**	**69.97**	**55.89**	36.43	31.04	**33.52**	**69.82**

TABLE VII
ABLATION STUDY.

Dataset	PSM			SWAN			SWaT			GECCO		
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1
w/o APPG	41.58	40.51	41.04	55.82	41.03	47.30	84.19	60.28	70.26	51.88	29.03	37.23
w/o regularization loss	41.39	43.70	42.51	57.14	41.81	48.29	84.45	63.01	72.17	52.03	31.90	39.55
w/o memory bank	41.42	42.86	42.13	56.24	41.50	47.76	84.59	61.25	71.05	52.27	31.24	39.11
w Transformer	41.07	44.23	42.59	57.08	42.63	48.80	84.26	62.86	72.00	52.20	32.12	39.77
IGCL	42.31	46.97	**44.52**	59.20	44.94	**51.09**	84.98	63.54	**72.71**	52.10	32.77	**40.23**

B. Overall Comparison and Analysis

Tables IV, V and VI show the anomaly prediction results of different models on all datasets. We randomly repeat each method 3 times and report the average result, where the best F1-score and AUC-ROC are highlighted in bold and significantly outperform the underlined second-best ones.

Key observations are followed. First, IGCL consistently outperforms the state-of-the-art baseline methods on all datasets. It demonstrates that IGCL is able to learn the normal temporal dependencies from a pair of successive normal time series sub-sequences, as well as identify the abnormal temporal dependencies that change from the normal to the abnormal with our generated anomaly precursors, and thus finally improve the accuracy of anomaly prediction.

Second, we observe that the accuracy of the naive contrastive learning-based baseline methods, *i.e.,* DCdetector and PAD, is unsatisfactory on most datasets. This is because they only focus on learning normal patterns from the non-labeled time series data and cannot generate anomaly precursors as negative samples to learn the abnormal temporal dependencies that change from the normal to the abnormal, which results in limited accuracy for unsupervised anomaly prediction tasks that require distinguishing both normal temporal dependencies and abnormal temporal dependencies.

Finally, IGCL achieves the best accuracy compared to the forecasting or reconstruction-based baselines. This suggests that learning the temporal dependencies from a pair of successive time series sub-sequences that change from the normal to the anomaly precursor is crucial to the anomaly prediction tasks. However, the forecasting or reconstruction-based baseline methods only focus on learning the normal patterns from the non-labeled time series data using the sliding time windows. They cannot explicitly learn the abnormal temporal dependencies from the normal to the anomaly precursors and thus have limited accuracy for the unsupervised anomaly prediction task.

C. Ablation Studies

We conduct ablation studies to validate the effectiveness of our key components that contribute to the improvements. In particular, we compare IGCL with the following variants:

- w/o anomaly precursor pattern generation (APPG): This variant does not use the anomaly precursor pattern generation module and thus the model cannot explicitly learn abnormal temporal dependencies.
- w/o regularization loss: This variant does not use the regularization loss to control the diffusion process.
- w/o memory bank: This variant does not use the memory bank and thus the anomaly precursors could only appear in each individual variable.
- w Transformer: This variant does not use the overlapping window-based TCN. It uses Transformer as shown in Eq.(16) to learn the contextual representations.

Table VII shows the accuracy of different variants. For the other datasets, the results show similar trends. From Table VII we observe that: (1) Our anomaly precursor pattern generation and regularization loss help generate diverse and complicated anomaly precursors that just start to deviate from the normal, which enhances the model to identify different kinds of potential anomaly precursors for more accurate time series anomaly prediction. (2) Without the anomaly precursor pattern generation, the model cannot learn the abnormal temporal dependencies that start to change from the normal to the abnormal and thus has much worse accuracy for the unsupervised anomaly prediction. (3) The memory bank is crucial for multi-variable anomaly prediction, which can help generate complicated anomaly precursors related to different

TABLE VIII
RUNTIME AND TOTAL PARAMETERS USED

Dataset	SWaT		MSL	
Method	Runtime (s)	Parameters (K)	Runtime (s)	Parameters (K)
PAD	2376	204	248	204
GANomaly	993	1693	81	1695
CAE-Ensemble	787	236	62	236
D3R	1443	380	133	382
w Transformer	1286	471	108	472
IGCL	760	172	61	173

TABLE IX
PARAMETER SENSITIVITY

Dataset	SWaT			MSL		
K	P	R	F1	P	R	F1
4	84.67	61.93	71.54	20.01	20.17	20.09
8	84.23	62.48	71.74	20.38	21.42	20.89
12	84.52	62.97	72.17	20.50	21.51	20.99
16	84.21	63.19	72.20	20.43	21.69	21.04
20	84.09	63.25	72.19	20.33	21.65	20.97
24	84.98	63.54	**72.71**	21.97	22.81	**22.38**

variables efficiently. (4) Our overlapping window-based TCN can effectively learn the temporal dependencies from the pairs of time series sub-sequences, and it is also more efficient than Transformer as shown in Table VIII.

D. Runtime and Total Parameters Used

We show the overall runtime and the number of total parameters used for different methods having state-of-the-art accuracy in Table VIII. It can be observed that our IGCL model is better than these baseline methods regarding the runtime and the number of parameters used. This is because our memory bank can help efficiently generate complicated anomaly precursors related to different variables, and our overlapping window-based TCN can help efficiently learn the contextual representations for all possible pairs of successive sub-sequences at once. Besides, we can see that CAE-Ensemble which uses convolutional network ensembles also has good performance regarding accuracy and complexity compared to the other baselines. However, CAE-Ensemble is still worse than our IGCL model, which validates the efficiency and effectiveness of our design of overlapping TCN.

E. Parameter Sensitivity

We evaluate the impact of the main hyperparameter, *i.e.,* K the size of the memory bank which controls how many anomaly precursors can be stored. The experimental results are shown in Table IX. For the other datasets, the results show similar trends. If we set a bigger size for the memory bank to store more representative anomaly precursors, our IGCL model can have better anomaly prediction results. When K increases to 24, the overall anomaly prediction results are already significantly better than the baselines. Besides, our model can spare the space complexity by selecting a small size, *i.e.,* 24, for the memory bank that is far smaller than $O(2^n)$, where n, *i.e.,* the total number of variables, is 51 and 55 for SWaT and MSL datasets, respectively. The impact of the size of the look-back window h can be seen in the Appendix at https://github.com/zhkai/IGCL.

TABLE X
LARGER LOOK-FORWARD WINDOWS

Dataset	SWaT					
f	4		8		12	
Metric	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC
CAE-Ensemble	71.48	81.71	68.62	74.92	62.70	73.32
D3R	71.29	80.62	68.51	75.94	62.43	71.76
IGCL	**72.71**	**82.39**	**70.02**	**77.75**	**64.56**	**74.88**

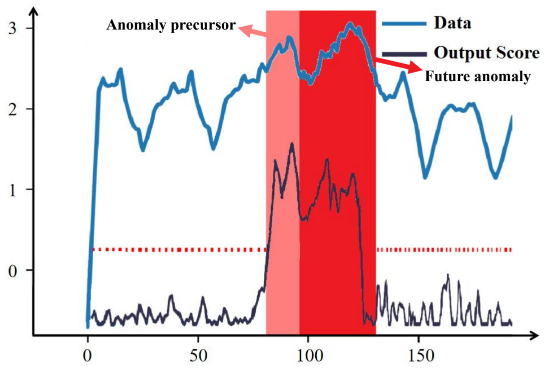


Fig. 5. IGCL can output larger scores for the precursors that start to differ from the normal ahead of the future severer anomalies.

We also evaluate the anomaly prediction task with larger look-forward windows of {4, 8, 12} in Table X. We can see that IGCL still can perform better than state-of-the-art baselines, even though the farther future anomalies are more difficult to predict.

F. Case Study and Visualization

We visualize the anomaly prediction of our IGCL method on the SMAP dataset, as shown in Figure 5. By identifying the anomaly precursor that just begins to deviate from the normal, our model can report a warning at timestamp 80, ahead of the future severe anomalies after timestamp 100.

VI. CONCLUSION

We present the importance-based generative contrastive learning model for a recent novel problem of unsupervised time series anomaly prediction. We propose anomaly precursor pattern generation, with diffusion-based distribution transformation and a variance regularization loss, to generate different kinds of potential anomaly precursors as labeled negative data. Then, we propose an overlapping window-based contrastive learning to distinguish anomaly precursors from the normal efficiently. Last, we propose a memory bank with importance-based scores to store representative anomaly precursors to generate complicated precursors related to more variables efficiently. Experiments on eight benchmark datasets demonstrate the superiority of our method. In future works, it is worth studying efficient pre-training on large-scale time series data with IGCL.

References

[1] G. Huang, J. He, J. Cao, Z. Qiao, M. Steyn, and K. Taraporewalla, "A real-time abnormality detection system for intensive care management," in *ICDE*. IEEE Computer Society, 2013, pp. 1376–1379.

[2] Q. Liu, P. Boniol, T. Palpanas, and J. Paparrizos, "Time-series anomaly detection: Overview and new trends," *Proc. VLDB*, 2024.

[3] P. Boniol, J. Paparrizos, and T. Palpanas, "An interactive dive into time-series anomaly detection," in *ICDE*, 2024.

[4] A. Zhang, S. Deng, D. Cui, Y. Yuan, and G. Wang, "An experimental evaluation of anomaly detection in time series," *Proc. VLDB*, 2023.

[5] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *SDM*, 2001, pp. 1–11.

[6] A. Abdulaal, Z. Liu, and T. Lancewicki, "Practical approach to asynchronous multivariate time series anomaly detection and localization," in *SIGKDD*, 2021, pp. 2485–2494.

[7] X. Wu, D. Zhang, M. Zhang, C. Guo, B. Yang, and C. S. Jensen, "AutoCTS+: Joint neural architecture and hyperparameter search for correlated time series forecasting," *Proc. ACM Manag. Data*, vol. 1, no. 1, 2023.

[8] D. Campos, M. Zhang, B. Yang, T. Kieu, C. Guo, and C. S. Jensen, "LightTS: Lightweight time series classification with adaptive ensemble distillation," *Proc. ACM Manag. Data*, vol. 1, no. 2, pp. 171:1–171:27, 2023.

[9] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu, "Revisiting time series outlier detection: Definitions and benchmarks," in *NeurIPS*, 2021.

[10] R.-G. Cirstea, B. Yang, and C. Guo, "Graph attention recurrent neural networks for correlated time series forecasting," in *KDD*, 2019.

[11] F. Chen, Y. Zhang, Z. Qin, L. Fan, R. Jiang, Y. Liang, Q. Wen, and S. Deng, "Learning multi-pattern normalities in the frequency domain for efficient time series anomaly detection," in *ICDE*, 2024.

[12] S. Y. Jhin, J. Lee, and N. Park, "Precursor-of-anomaly detection for irregular time series," in *SIGKDD*. ACM, 2023, pp. 917–929.

[13] V. Cerqueira, L. Torgo, and C. Soares, "Early anomaly detection in time series: a hierarchical approach for predicting critical health episodes," *Mach. Learn.*, vol. 112, no. 11, pp. 4409–4430, 2023.

[14] P. Boniol, M. Meftah, E. Remy, B. Didier, and T. Palpanas, "dcnn/dcam: anomaly precursors discovery in multivariate time series with deep convolutional neural networks," *Data-Centric Engineering*, vol. 4, 2023.

[15] H. Hojjati, M. Sadeghi, and N. Armanfard, "Multivariate time-series anomaly detection with temporal self-supervision and graphs: Application to vehicle failure prediction," in *PKDD*, vol. 14175, 2023.

[16] D. Xu, W. Cheng, J. Ni, D. Luo, M. Natsumeda, D. Song, B. Zong, H. Chen, and X. Zhang, "Deep multi-instance contrastive learning with dual attention for anomaly precursor detection," in *SIAM SDM*, 2021.

[17] Y. Ang, Q. Huang, A. Tung, and Z. Huang, "A stitch in time saves nine: Enabling early anomaly detection with correlation analysis," in *ICDE*. IEEE, 2023, pp. 1832–145.

[18] Y. Ang, Q. Huang, A. K. Tung, and Z. Huang, "Eads: An early anomaly detection system for sensor-based multivariate time series," in *ICDE*. IEEE, 2024, pp. 5433–5436.

[19] A. Nina, "Analysis of VLF signal noise changes in the time domain and excitations/attenuations of short-period waves in the frequency domain as potential earthquake precursors," *Remote. Sens.*, vol. 16, no. 2, 2024.

[20] S. Schmidl, F. Naumann, and T. Papenbrock, "Autotsad: Unsupervised holistic anomaly detection for time series data," *Proc. VLDB*, 2024.

[21] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proc. VLDB*, 2022.

[22] Y. Zhao, L. Deng, X. Chen, C. Guo, B. Yang, T. Kieu, F. Huang, T. B. Pedersen, K. Zheng, and C. S. Jensen, "A comparative study on unsupervised anomaly detection for time series: Experiments and analysis," *CoRR*, vol. abs/2209.04635, 2022.

[23] K. M. Ting, Z. Liu, L. Gong, H. Zhang, and Y. Zhu, "A new distributional treatment for time series anomaly detection," *VLDB J.*, 2024.

[24] X. He, Y. Li, J. Tan, B. Wu, and F. Li, "Oneshotstl: One-shot seasonal-trend decomposition for online time series anomaly detection and forecasting," *Proc. VLDB*, 2023.

[25] T. Kieu, B. Yang, C. Guo, C. S. Jensen, Y. Zhao, F. Huang, and K. Zheng, "Robust and explainable autoencoders for unsupervised time series outlier detection," in *ICDE*, 2022, pp. 3038–3050.

[26] D. Campos, T. Kieu, C. Guo, F. Huang, K. Zheng, B. Yang, and C. S. Jensen, "Unsupervised time series outlier detection with diversity-driven convolutional ensembles," *PVLDB*, vol. 15, no. 3, 2022.

[27] Y. Li, W. Chen, B. Chen, D. Wang, L. Tian, and M. Zhou, "Prototype-oriented unsupervised anomaly detection for multivariate time series," in *ICML*, vol. 202. PMLR, 2023, pp. 19 407–19 424.

[28] B. Sch"olkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[29] L. Ruff, R. A. Vandermeulen, N. G"ornitz, L. Deecke, S. A. Siddiqui, A. Binder, E. M"uller, and M. Kloft, "Deep one-class classification," in *ICML*, vol. 80, 2018, pp. 4393–4402.

[30] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *SIGMOD*, 2000.

[31] P. Karczmarek, A. Kiersztyn, W. Pedrycz, and E. Al, "K-means-based isolation forest," *Knowledge-based systems*, vol. 195, p. 105659, 2020.

[32] Y. Shin, S. Lee, S. Tariq, M. S. Lee, O. Jung, D. Chung, and S. S. Woo, "ITAD: integrative tensor-based anomaly detection system for reducing false positives of satellite systems," in *CIKM*, 2020, pp. 2733–2740.

[33] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," *NeurIPS*, 2020.

[34] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *ICLR*. OpenReview.net, 2018.

[35] Y. Fang, J. Xie, Y. Zhao, L. Chen, Y. Gao, and K. Zheng, "Temporal-frequency masked autoencoders for time series anomaly detection," in *ICDE*, 2024.

[36] X. Hao, Y. Chen, C. Yang, Z. Du, C. Ma, C. Wu, and X. Meng, "From chaos to clarity: Time series anomaly detection in astronomical observations," in *ICDE*, 2024.

[37] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *SIGKDD*, 2019, pp. 2828–2837.

[38] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, and D. Pei, "Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding," in *SIGKDD*, 2021.

[39] T. Kieu, B. Yang, C. Guo, R. Cirstea, Y. Zhao, Y. Song, and C. S. Jensen, "Anomaly detection in time series with robust variational quasi-recurrent autoencoders," in *ICDE*, 2022, pp. 1342–1354.

[40] B. Du, X. Sun, J. Ye, K. Cheng, J. Wang, and L. Sun, "Gan-based anomaly detection for multivariate time series using polluted training set," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, 2023.

[41] Y. Chen, C. Zhang, M. Ma, Y. Liu, R. Ding, B. Li, S. He, S. Rajmohan, Q. Lin, and D. Zhang, "Imdiffusion: Imputed diffusion models for multivariate time series anomaly detection," *Proc. VLDB*, 2023.

[42] C. Xiao, Z. Gou, W. Tai, K. Zhang, and F. Zhou, "Imputation-based time-series anomaly detection with conditional weight-incremental diffusion models," in *SIGKDD*, 2023, pp. 2742–2751.

[43] C. Wang, Z. Zhuang, Q. Qi, J. Wang, X. Wang, H. Sun, and J. Liao, "Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection," in *NeurIPS*, 2023.

[44] Y. Yang, C. Zhang, T. Zhou, Q. Wen, and L. Sun, "Dcdetector: Dual attention contrastive representation learning for time series anomaly detection," in *SIGKDD*. ACM, 2023.

[45] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *ICLR*, 2022.

[46] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020.

[47] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, vol. 119, 2020.

[48] A. S. Iyer, J. S. Nath, and S. Sarawagi, "Maximum mean discrepancy for class ratio estimation: Convergence bounds and kernel selection," in *ICML*, vol. 32. JMLR.org, 2014, pp. 530–538.

[49] L. Kohs, B. Alt, and H. Koeppl, "Markov chain monte carlo for continuous-time switching dynamical systems," in *ICML*, 2022.

[50] N. Engelmann and H. Koeppl, "Forward-backward latent state inference for hidden continuous-time semi-markov chains," in *NeurIPS*, 2022.

[51] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv/1607.08022*, 2016.

[52] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *ICLR*. OpenReview.net, 2021.

[53] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[56] R. Wu and E. J. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress (extended abstract)," in *ICDE*, 2022.

[57] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *SIGKDD*, 2018, pp. 387–395.

[58] A. Huet, J. M. Navarro, and D. Rossi, "Local evaluation of time series anomaly detection algorithms," in *SIGKDD*, 2022.

[59] Y. Sun, G. Pang, G. Ye, T. Chen, X. Hu, and H. Yin, "Unraveling the 'anomaly' in time series anomaly detection: A self-supervised tri-domain solution," in *ICDE*, 2024.

[60] J. Paparrizos, P. Boniol, T. Palpanas, R. Tsay, A. J. Elmore, and M. J. Franklin, "Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection," *Proc. VLDB*, 2022.

TABLE XI
PARAMETER SENSITIVITY

Dataset	SWaT				PSM				SWAN			
h	P	R	F1	AUC-ROC	P	R	F1	AUC-ROC	P	R	F1	AUC-ROC
8	83.13	62.75	71.52	81.33	41.56	46.83	44.04	65.67	58.13	43.11	49.51	65.99
16	83.59	62.86	71.76	81.72	42.31	46.97	44.52	65.75	59.20	44.94	51.09	68.92
32	84.98	63.54	72.71	82.39	42.14	46.61	44.26	65.70	59.31	44.07	50.57	66.91
64	87.67	61.36	72.19	82.33	42.59	45.10	43.81	65.18	59.85	44.03	50.73	68.57

Appendix

The impact of the size of the look-back window h can be seen in Table XI. From the table we can see that our IGCL model is relatively stable to the hyperparameter of the look-back window size h and data from different domains require different look-back windows to better identify the anomaly precursors.