Information Sciences 610 (2022) 266–280

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Contrastive autoencoder for anomaly detection in multivariate time series

Hao Zhou [a], Ke Yu [a,*], Xuan Zhang [b], Guanlin Wu [a], Anis Yazidi [c,d,e]

[a] Beijing University of Posts and Telecommunications, Beijing, China
[b] Norwegian Research Centre AS, Grimstad, Norway
[c] Oslo Metropolitan University, Oslo, Norway
[d] Norwegian University of Science and Technology, Trondheim, Norway
[e] Oslo University Hospital, Oslo, Norway

ARTICLE INFO

Article history:
Received 21 April 2022
Received in revised form 24 July 2022
Accepted 28 July 2022
Available online 3 August 2022

Keywords:
Anomaly detection
Multivariate time series
Autoencoder
Contrastive learning
Data augmentation

ABSTRACT

With the proliferation of the Internet of Things, a large amount of multivariate time series (MTS) data is being produced daily by industrial systems, corresponding in many cases to life-critical tasks. The recent anomaly detection researches focus on using deep learning methods to construct a normal profile for MTS. However, without proper constraints, these methods cannot capture the dependencies and dynamics of MTS and thus fail to model the normal pattern, resulting in unsatisfactory performance. This paper proposes CAE-AD, a novel contrastive autoencoder for anomaly detection in MTS, by introducing multi-grained contrasting methods to extract normal data pattern. First, to capture the temporal dependency of series, a projection layer is employed and a novel contextual contrasting method is applied to learn the robust temporal representation. Second, the projected series is transformed into two different views by using time-domain and frequency-domain data augmentation. Last, an instance contrasting method is proposed to learn local invariant characteristics. The experimental results show that CAE-AD achieves an F1-score ranging from 0.9119 to 0.9376 on the three public datasets, outperforming the baseline methods.
© 2022 Elsevier Inc. All rights reserved.

1. Introduction

With the proliferation of the Internet of Things (IoT), a large amount of MTS data is generated constantly by industrial systems [18,15]. Since these data are usually associated with critical missions, monitoring them for unveiling anomalies is crucial. In simple terms, anomaly detection in MTS aims to detect the time stamps of the series where the observations deviate largely from the normal values or where unusual temporal patterns emerge.

Anomaly detection is a well-established research topic that has attracted a lot of research attention over the last decades. In early stage, researchers have focused on methods for detecting anomalies in univariate time series data [19,4,17,41,28]. Within the context of modern industrial systems, a device is usually monitored by a multitude of sensors, each of which measures a distinct variable. Diagnosing MTS data with multiple monitoring indicators is essential to ensure the normal

Abbreviations: MTS, multivariate time series.

* Corresponding author.
E-mail addresses: zhouh@bupt.edu.cn (H. Zhou), yuke@bupt.edu.cn (K. Yu), xuan.z.jiao@gmail.com (X. Zhang), wuguanlin@bupt.edu.cn (G. Wu), anisy@oslomet.no (A. Yazidi).

https://doi.org/10.1016/j.ins.2022.07.179
0020-0255/© 2022 Elsevier Inc. All rights reserved.

operations of industrial systems. Further, abnormal readings are usually difficult to be annotated from a large amount of sensor data, making little or even no labels available in MTS. The supervised methods [33,23,31] are unfeasible in this case. Thus, we focus on identifying anomalies in MTS using unsupervised methods.

Unsupervised anomaly detection methods in MTS are challenging with temporal dependency and dynamic variability. In the unsupervised setting, models are designed to construct a normal profile through a large amount of normal time series data. Those that behave very differently from the normal pattern are detected as anomalies. The classic unsupervised time series anomaly detection methods, such as classification-based models (e.g., one-class SVM [12,36]), distance-based models (e.g., KNN [5]) and clustering-based models, (e.g., k-means [20]), ignore the temporal dependency in MTS, resulting in inferior performance.

Recently, autoencoder-based methods have received a lot of attention, owing to their ability to learn informative representation. The state-of-the-art anomaly detection methods usually apply well-designed autoencoders to capture normal patterns of MTS data, e.g., LSTM-based encoder-decoder model [26], adversarial training based autoencoder [3,27], and variational autoencoder [30,9], etc. Here, point-to-point reconstruction or prediction criteria are usually used to detect anomalies. However, due to the complexity of temporal dependency in MTS, the single point-wise context information cannot comprehensively characterize the temporal pattern of MTS data. Moreover, the normal pattern may change dynamically over time. Autoencoder-based models will suffer from an overfitting problem if no proper regularization is applied. Therefore, these methods fail to construct an accurate profile for normal data and are incapable of learning robust representation, resulting in unsatisfactory performance in practice.

In this paper, we propose a contrastive autoencoder for anomaly detection in MTS, namely CAE-AD. Contrastive learning has been proved successful to learn transformation-invariant representation of data in various domains, such as image classification [6], audio [29] and language understanding [13]. Combining autoencoder with the contrastive loss for the window-segmented MTS, we find that it is capable of capturing invariant information from the dynamically changing time series, which improves the robustness of the representation for autoencoder. However, the window-wise method is insufficient for obtaining fine-grained temporal representation. Therefore, we expand this method with multi-contrasting to explore point-wise and window-wise temporal information, which helps construct a normal profile in MTS.

Going beyond previous autoencoder-based methods, the proposed CAE-AD is able to model the temporal and dynamic features of time series. More precisely, we first utilize the attention mechanism to capture the temporal dependency. Unlike commonly used positional coding methods, we adopt a contextual contrasting method to learn the position information. Subsequently, we explore data augmentation in the time and frequency domain, which helps obtain different views of the same segment. After that, a shared-weight encoder is developed to encode the augmented data and an instance contrasting method is proposed to capture the local invariant characteristics of latent variables. With the proposed multi-grained contrasting (the contextual contrasting and instance contrasting), CAE-AD can learn the robust representation of MTS. Finally, an LSTM decoder is devised and the reconstruction errors are further utilized to detect anomalies. Extensive experiments are conducted to validate the effectiveness of the proposed method.

The main contributions of the paper are summarized as follows:

- We propose a novel approach to detect anomalies in MTS, namely CAE-AD. Contrastive learning method is utilized to enhance the ability of the autoencoder for constructing a robust profile for normal data, making abnormal data more distinguishable to be detected.
- Simple but efficient data augmentation methods are designed for MTS in the CAE-AD framework. We explore augmentation for MTS in both the time and frequency domains. This helps select positive pairs for contrasting and encourages learning dynamic behaviours of MTS data.
- Multi-grained contrasting methods are proposed to learn the robust representations of MTS. First, we propose contextual contrasting to extract the temporal dependency of MTS. Second, we propose instance contrasting to further capture local invariant characteristics of MTS. With multi-grained contrasting methods, CAE-AD can comprehensively learn multi-scale contextual information.
- We conduct extensive experiments to evaluate the performance of CAE-AD on three public datasets. The experiment results demonstrate that the performance of CAE-AD outperforms state-of-the-art baselines.

The remainder of the paper is organized as follows: In Section 2, we provide an overview of the related work. In Section 3, we introduce the CAE-AD framework and anomaly detection in detail. The experimental results and analyses are presented in Section 4 before concluding the paper in the last section.

2. Related Work

2.1. Anomaly Detection

MTS anomaly detection has recently become a popular research topic. There is an abundance of literature on MTS anomaly detection. Traditional methods of anomaly detection in MTS mostly resort to statistical methods. In [5], Chaovalitwongse et al. suggested a distance-based model based on KNN to classify abnormal MTS data. In [21], an improved variant of the

distance-based model was proposed, and the authors took the angular relationship among the data points into account. Isolation Forest was utilized in [24], where the authors isolated outliers by dividing the data set. Although these methods are highly efficient, they do not consider the long-term temporal dependency of MTS data.

Methods based on deep learning have recently achieved significant performance improvement for anomaly detection in MTS data. LSTM-NDT [18] applied LSTM to model the temporal dependence and detected anomalies based on prediction errors. The transformer-based methods have also gained a lot of attention. Anomaly Transformer [39] explored the self-attention weight for anomalies and proposed a minimax training strategy to amplify the difference between normal and abnormal data. GTA [7] used a transformer-based structure to learn a graph structure, which helped capture temporal dependency in MTS. TranAD [35] proposed a transformer-based model for anomaly detection and used adversarial training and self-conditioning technologies to improve the performance.

The autoencoder-based model has become popular recently, where the encoder is employed to reduce the dimensionality of MTS data while the decoder aims to reconstruct the MTS. By minimizing the reconstruction errors, the autoencoder-based model can capture the normal pattern of the data, and reconstruction errors are further used to detect anomalies. For example, EncDec-AD [26] utilized LSTM as the basic cell of the encoder and decoder. LSTM-VAE [30] combined LSTM and VAE, but it ignored the dependence of stochastic variables. An approach reckoned as OmniAnomaly was proposed in [34] and it is based on a stochastic recurrent neural network that learnt normal data patterns by modelling robust representations of MTS data. MAD-GAN [22] adopted the GAN framework to capture the time and space information of the data.

2.2. Contrastive Learning

Invariant information plays a crucial role in inter-domain or intra-domain tasks with noise perturbations. For example, CMCH was proposed by [2] to learn the consistent representation of hash codes for multi-modal data, which utilized the invariant information in multi-modal data to construct the informative latent space. MIAN [42] proposed modality-invariant asymmetric networks to preserve the semantic similarity, and proposed a conditional variational information bottleneck network to learn modality-invariant information.

Unlike these supervised methods, contrastive learning methods learn the transformation-invariant representation of data in a self-supervised setting. The main idea of contrastive learning is to reduce the distance between similar samples in the feature space and maximize the distance between different samples in the same space. CPC [29] used autoregressive models to predict future values in latent space and proposed InfoNCE loss to train the network for obtaining robust data representations. MoCo [16] further considered the limitation of obtaining negative samples and built a dynamic dictionary to look up sample pairs for contrasting. SimCLR [6] discussed the significant role of data augmentation in contrastive learning and proposed a learnable nonlinear transformation module to improve the quality of contrastive representation vectors.

Applying contrastive learning to time series data, TS-TCC [11] proposed a self-supervised framework with multiple contrasting methods to learn informative time-series representation. TS2Vec [40] proposed a hierarchical contrasting method to learn multi-grained contextual information of time series. In this work, we develop a contrastive autoencoder representation learning and anomaly detection of MTS.

3. Proposed Model

We first present the problem description of anomaly detection in MTS data. Then, we give an overview of the CAE-AD framework and explain its main modules. Finally, we describe in detail the proposed model and anomaly detection.

3.1. Problem Description

In this paper, we focus on detecting anomalies in MTS data. Let $\boldsymbol{x_t} \in \mathbb{R}^m$ denotes the vector of dimension m at time step t, where $x_{t,i} \in \mathbb{R}$ denotes the value of i^{th} variable at time step t. Multivariate time series can be expressed as $\mathcal{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}\}, \mathcal{X} \in \mathbb{R}^{N \times m}$, where N is the length of the observed sequence. A anomaly score S_t is calculated at timestep t. The point is detected as an anomaly if the anomaly score is greater than a specified threshold.

3.2. Overview of the Framework

The CAE-AD framework is shown in Fig. 1, which consists of three parts: data preprocessing, learning representation, and anomaly detection.

First, we normalize the data and divide them via windows during data preprocessing. Second, we propose a Contrastive Autoencoder model to learn the representation of MTS. As demonstrated in the bottom of Fig. 1, a projection layer is implemented to learn the embedding, and then we design two types of data augmentation methods in the time domain and frequency domain respectively, which is helpful to generate different views of the sequence data. The multi-grained contrasting method is proposed to capture the temporal dependency and local-invariant characteristics of the data. Finally, the anomaly score of each timestep is calculated based on the reconstruction error.

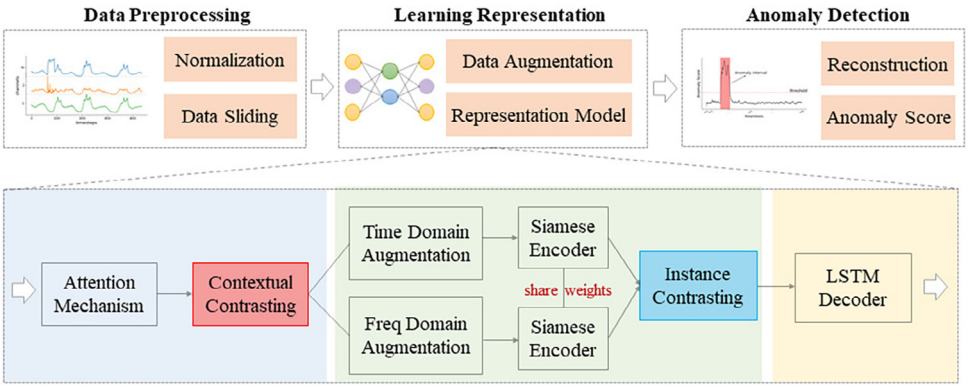


Fig. 1. The overall framework of CAE-AD. We first conduct data normalization and segment the MTS using a sliding window. Then contrastive autoencoder is adopted to the learned robust representation of time series data. Finally, reconstruction errors are utilized to detect anomalies.

3.3. Preprocessing

For the input MTS data $\mathcal{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}\}$, normalization is first applied to scale the data. The observations at the current timestep depend on historical observations of the data. To capture the temporal dependency, we introduce a sliding window in the CAE-AD framework. As shown in Fig. 2, we obtain continuous data segments as the input of CAE-AD using a sliding window with the interval of l. For example, a window of observations can be expressed as $\boldsymbol{x_{1:w}} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_w}], \boldsymbol{x_{1:w}} \in \mathbb{R}^{w \times m}$, where w is the window size and m is the dimension of the data.

3.4. Representation Model

In the CAE-AD, we first employ a projection layer and propose contextual contrasting to learn the temporal embedding of MTS data. Then we explore data augmentation methods for time series and propose instance contrasting to encourage robustness of representation. Last, an LSTM decoder is adopted to reconstruct the MTS data.

3.4.1. Projection Layer

Contrastive learning aims at maximizing the similarity between the same samples of different views while minimizing the similarity between different samples to learn invariant characteristics. In order to select the positive pairs for contrasting, we explore different data augmentation methods to obtain various views of the data, which is considered critical in contrastive learning [6].

Simple transformations are commonly used for time series augmentation, such as cropping, flipping, and jittering, etc. However, it is not always appropriate to directly transform the original time series data. For example, the raw data may have an upward trend. If the cropping operation removes this trend, the augmented time series may represent a different mode while the previous contrastive learning strategy will deem both time series, original and augmented, similar. To overcome this problem, we employ a projection layer. We first project the MTS data, and its embedding after projection layer can be used for data augmentation to supplement domain-specific knowledge [10].

In order to capture the temporal dependency of the data, we adopt the attention mechanism [38] as the projection layer. A mask is used to prevent future data information leakage, and all attention scores of the future data are assigned zeros. For a

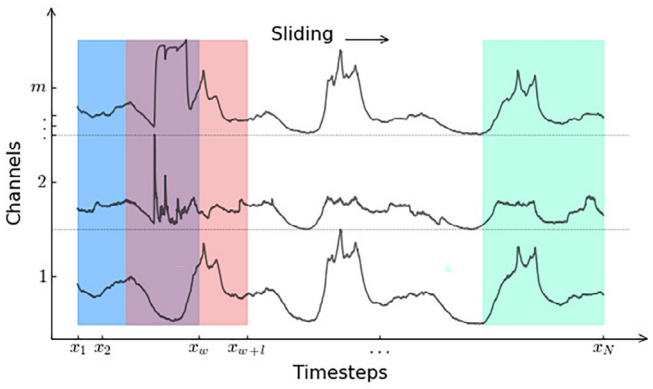


Fig. 2. Data preprocessing. A sliding window is adopted to slice the sequences.

window of the data $\boldsymbol{x_{1:w}} = [\boldsymbol{x_1}, \boldsymbol{x_2} \ldots, \boldsymbol{x_w}]$, where w is the window size, linear transformations are utilized to obtain the query matrix Q, the key matrix K and the value matrix V respectively, where $Q = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_w}]^T W^Q$ and W^Q is a learnable matrix. The embedding after attention mechanism can be expressed by the following formula.

$$\boldsymbol{a_{1:w}} = \text{softmax}\left(\frac{QK^T}{\sqrt{m}} \odot Mask\right)V, \tag{1}$$

where $\boldsymbol{a_{1:w}} = [\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_w}], \boldsymbol{a_{1:w}} \in \mathbb{R}^{w \times m}$ denotes the sequence of embeddings $\boldsymbol{a_t} \in \mathbb{R}^m$. Symbol $\odot$ denotes element-wise operation and K^T is the transpose of K. *Mask* is a lower triangular matrix, which means that the data at timestep t only focus on current and historical information. For example, the embedding $\boldsymbol{a_t}$ only pays attention to the information of $\boldsymbol{x_{1:t}}$.

3.4.2. Contextual Contrasting

The attention mechanism ignores the position information. To include position information, we propose contextual contrasting to learn the relative position of the MTS data.

The embedding $\boldsymbol{a_t}$ has a higher similarity to the representation of neighbor timesteps $\boldsymbol{a_{t-1}}, \boldsymbol{a_{t+1}}$, while has a lower similarity to representation of distant timesteps. Contextual contrasting aims at reducing the distance between neighbouring timesteps in the embedding space while increasing the distance between different timesteps in the embedding space, which can be expressed as follows:

$$l_{cont}^{(i,t)} = -\log \frac{\exp(\cos(\boldsymbol{a_t^{(i)}}, \boldsymbol{a_{t-1}^{(i)}})/\tau) + exp(\cos(\boldsymbol{a_t^{(i)}}, \boldsymbol{a_{t+1}^{(i)}})/\tau)}{\sum_{k=1}^{w} I_{[k \neq t]} \exp(\cos(\boldsymbol{a_t^{(i)}}, \boldsymbol{a_k^{(i)}})/\tau)},$$

$$\mathcal{L}_{cont} = \frac{1}{N_w} \sum_{i=1}^{N_w} \sum_{t=1}^{w} l_{cont}^{(i,t)},$$

where $l_{cont}^{(i,t)}$ denotes the loss function at timestep t in the i-th window. N_w is the total number of windows. $I_{[k \neq t]} \in \{0, 1\}$ is an indicator function, and it equals to 1 if $k \neq t$, and 0 otherwise. τ is the temperature parameter. Cosine function is utilized to measure the similarity among the embeddings.

The embeddings obtained by the attention mechanism do not include position information. There are certain articles [38,32,25] that have studied positional coding. However, those methods directly add the position information to the data, so that the data information may interfere with the position information. Alternatively, we adopt contextual contrasting to learn the position information in the embedding space, which provides a new view for the positional coding of the attention mechanism.

3.4.3. Data Augmentation

To make full use of the properties of the MTS, we explore data augmentation methods for the embeddings of the projection layer in both time and frequency domains respectively.

In the time domain, we add Gaussian noise $N(0, \Sigma_1)$ for the embeddings to generate similar samples, where $\Sigma_1 = diag(\sigma_1)$ and σ_1 is the deviation. In frequency main, we perform a two-dimensional Discrete Fourier Transform on the embeddings $\boldsymbol{a_{1:w}} = [\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_w}]$ to obtain the spectrum $F(u, v)$.

$$\begin{aligned} F(u, v) &= \tfrac{1}{wm} \sum_{t=0}^{w-1} \sum_{k=0}^{m-1} a_{t,k} \exp(-j2\pi(ut/w + vk/m)) \\ &= A(u, v) \exp(j\theta(u, v)), \end{aligned} \tag{3}$$

where $u = 0, 1 \ldots, w-1$ and $v = 0, 1, \ldots, m-1$ denote the indices of frequencies, w and m are window size and dimension of the embeddings $\boldsymbol{a_t}$ respectively. Furthermore, $F(u, v)$ can also be expressed as the combination of amplitude spectrum $A(u, v)$ and phase spectrum $\theta(u, v)$.

We add Gaussian noise $N(0, \Sigma_2)$ for the amplitude spectrum $A(u, v)$ and phase spectrum $\theta(u, v)$, respectively, where $\Sigma_2 = diag(\sigma_2)$ and σ_2 control the deviation of noise. Then, Inverse Discrete Fourier Transform (IDFT) is conducted to convert the frequency domain data into time-domain data. Finally, we take the real part as the augmented data, as shown in the equation below:

$$\begin{aligned} &f(t, k) = \text{Real}[IDFT(F(u, v))] \\ &IDFT = \tfrac{1}{wm} \sum_{u=0}^{w-1} \sum_{v=0}^{m-1} F(u, v) \exp(j2\pi(ut/m + vk/w)), \end{aligned} \tag{4}$$

where $t = 0, 1, \ldots, w-1$, and $k = 0, 1, \ldots, m-1$ denote the timestep and the dimension of the augmented data $f(t, k)$ respectively.

The proposed data augmentation methods help generate different views of the observed sequence and obtain positive pairs for instance constrasting.

3.4.4. Instance Contrasting

We obtain two views of the data after time-domain and frequency-domain data augmentation. Then the augmented data is sent to LSTM siamese encoders to obtain low-dimensional latent variables, where LSTM siamese encoders are composed of two shared-weights LSTM neural networks. Finally, instance contrasting is applied to learn local invariant information of latent variables.

The two views of MTS data in the same window are positive pairs, while the MTS data in different windows are considered as negative pairs. Intuitively, Instance contrasting encourages maximizing the similarity of the different augmented data from the same window while minimizing the similarity of data from different windows, which aims to learn invariant information of the two types of augmented data. As shown in Fig. 3, The positive pair of latent variables can be expressed as $(\boldsymbol{z_t}, \boldsymbol{z_t^+})$, and the instance contrastive loss is shown in the equation below:

$$
\begin{aligned}
l_{inst}^{(i,t)} &= -\log \frac{\exp(\cos(\boldsymbol{z_t^{(i)}}, \boldsymbol{z_t^{(i)+}})/\tau)}{\sum_{k=1}^{B}[\exp(\cos(\boldsymbol{z_t^{(i)}}, \boldsymbol{z_k^{(i)+}})/\tau) + I_{[k\neq t]} \exp(\cos(\boldsymbol{z_t^{(i)}}, \boldsymbol{z_k^{(i)}})/\tau)]}, \\
\mathcal{L}_{inst} &= \frac{1}{N_b} \sum_{i=1}^{N_b} \sum_{t=1}^{B} l_{inst}^{(i,t)},
\end{aligned}
\tag{5}
$$

where B is the minibatch size, and N_b is the number of the minibatches. Given the positive pair $(\boldsymbol{z_t^{(i)}}, \boldsymbol{z_{i,t}^{(i)+}})$ of the t-th window in the i-th minibatch, we calculate the contrastive loss $l_{inst}^{(i,t)}$ using cosin similarity. τ is the temperature parameter.

Instance contrasting works in window level, and contextual contrasting works in timestep level. The multi-grained contrasting methods can learn the fine-grained and coarse-grained information for time series. By combining the two types of contrastive losses, CAE-AD framework is capable of learning the representation from multiple scales. Furthermore, contextual contrasting and instance contrasting permit the framework to learn complementary information so that the representation can better reconstruct the normal MTS.

3.4.5. LSTM Decoder

The main goal of the decoder is to reconstruct the data. We conduct a recurrent decoding method as shown in Fig. 3. Specifically, we first concatenate the latent variables $[\boldsymbol{z}, \boldsymbol{z}^+]$ as the initial information, and an LSTM cell is applied to reconstruct the data at the first timestep, namely, $\hat{\boldsymbol{x}}_1$. Then, the reconstruction data $\hat{\boldsymbol{x}}_1$ is concatenated with the hidden state of the decoder for the reconstruction at the next timestep, namely $\hat{\boldsymbol{x}}_2$. The reconstruction method can be formulated as:

$$
\hat{x}_t = \begin{cases} W^r g(concat[\boldsymbol{z}, \boldsymbol{z}^+]), & if\ t = 0 \\ W^r g(W^z concat[h_t, \widehat{x_{t-1}}]), & if\ t = 1, 2, \ldots w-1, \end{cases}
\tag{6}
$$

where W^z, W^r are the learnable weight matrices. h_t and $\hat{x}_t$ denote the t-th hidden state of the decoder and the t-th output of the decoder respectively. We use LSTM network as the decoder $g(.)$, as shown in Fig. 3.

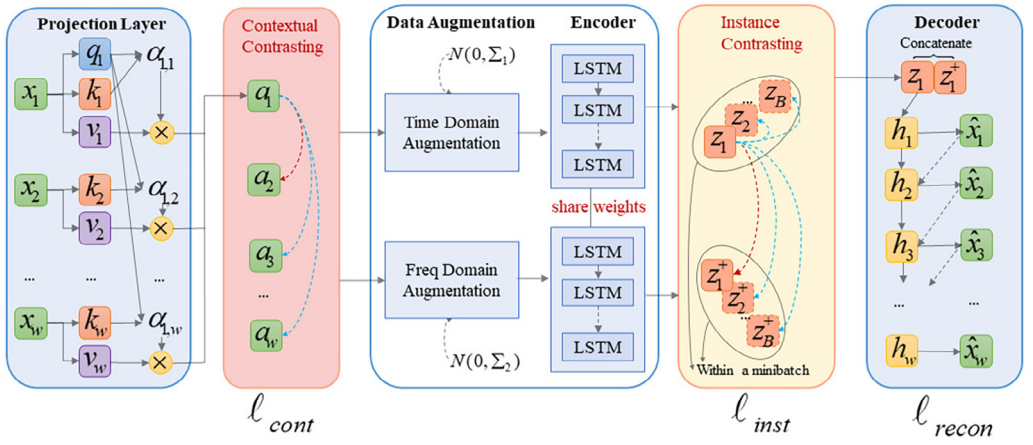


Fig. 3. Detailed Neural Network of CAE-AD. In the detailed framework, q, k, v denote the query, key and value vector of attention mechanism, respectively. The green blocks represent the raw data $\{x_1, x_2, \ldots, x_w\}$, the embeddings $\{a_1, a_2, \ldots, a_w\}$ and the reconstructed data $\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_w\}$, respectively. The latent variables (z_1, z_1^+) is a positive pair in the same window, and $\{(z_1, z_2), \ldots, (z_1, z_B), (z_1, z_2^+), \ldots, (z_1, z_B^+)\}$ are negative pairs in diffient windows of the minibatch. $\{h_1, h_2, \ldots, h_w\}$ are hidden states of the LSTM decoder.

The reconstruction loss can be expressed by:

$$\mathcal{L}_{recon} = \frac{1}{wN_w}\sum_{i=1}^{N_w}\sum_{t=1}^{w}(\hat{x}_t^{(i)} - x_t^{(i)})^2, \tag{7}$$

where w, N_w are the window size and the number of windows, respectively. $\hat{x}_t^{(i)}, x_t^{(i)}$ are the reconstruction and raw data at timestep t in the i-th window, respectively.

The overall loss function can be expressed as the summation of reconstruction loss, contextual contrastive loss, and instance contrastive loss, as shown in Eq. (8). Our goal is to minimize the overall loss so as to learn the normal pattern of the data.

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda_1\mathcal{L}_{cont} + \lambda_2\mathcal{L}_{inst}, \tag{8}$$

where $\mathcal{L}_{recon}, \mathcal{L}_{cont}, \mathcal{L}_{inst}$ represent reconstruction loss, contextual contrastive loss, and instance contrastive loss respectively. λ_1, and λ_2 are hyperparameters. The training procedure is summarized in Algorithm 1.

Algorithm 1: CAE-AD training

Input: The preprocessed training dataset $\mathcal{X}$
Output: The trained network with parameters θ
1: Initialize the network parameters θ
2: **repeat**
3: /* Samples for model inputs */
$x_{1:w} \sim \mathcal{X}$
4: /* Contextual contrastive loss using Eq. (2) */
$\mathcal{L}_{cont} \leftarrow \frac{1}{N_w}\sum_{i=1}^{N_w}\sum_{t=1}^{w} l_{cont}^{(i,t)}$
5: /* Instance contrastive loss using Eq. (5) */
$\mathcal{L}_{inst} \leftarrow \frac{1}{N_b}\sum_{i=1}^{N_b}\sum_{t=1}^{B} l_{inst}^{(i,t)}$
6: /* Reconstruction loss using Eq. (7) */
$\mathcal{L}_{recon} \leftarrow \frac{1}{wN_w}\sum_{i=1}^{N_w}\sum_{t=1}^{w}(\hat{x}_t^{(i)} - x_t^{(i)})^2$
7: /* Overall loss using Eq. (8) */
$\mathcal{L}(x_{1:w};\theta) \leftarrow \mathcal{L}_{recon} + \lambda_1\mathcal{L}_{cont} + \lambda_2\mathcal{L}_{inst}$
8: /* Optimize the parameters */
$\theta \leftarrow Adam(\mathcal{L}(x_{1:w};\theta))$
9: **until** convergence

3.5. Anomaly Detection

Historical information is essential for MTS data, so we utilize the previous reconstruction errors to calculate the anomaly score S_t of x_t at the current timestep t, as:

$$S_t = \frac{1}{w}\sum_{i=t-w+1}^{t}(\hat{x}_i - x_i)^2, \tag{9}$$

where $\hat{x}_i$ is the reconstructed MTS data, and the window size is w.

We hypothesis that anomalies appear in low-density regions. The CAE-AD is optimized using clean dataset without anomalies to learn the normal pattern. In test process, the test data will be mapped to the normal latent space and the latent vectors z will be close to each other due to the effect of multi-grained contrasting. If an anomaly point $x\prime$ does not conform the normal pattern, then it will have a low probability $p(x\prime|z)$ to perform reconstruction, which leads to a high reconstruction error. Thus, a higher anomaly score means that the data at timestep t is more likely to be an anomalous value. Otherwise, it is considered as normal data. The choice of the threshold depends on the application scenario, and many studies [18,34] configure the threshold dynamically based on anomaly scores. In this paper, we focus on designing the framework for learning robust representation and performing better reconstruction. We enumerate all thresholds and select the one with the best F1 score as the previous works [3,9] done.

4. Experiments

In this section, extensive experiments are conducted to evaluate the performance of CAE-AD. First, we compare CAE-AD with the state-of-the-art models. Subsequently, three variants of the CAE-AD framework are constructed to validate the effectiveness of two proposed contrastive losses. Furthermore, parameter sensitivity and ablation study experiments are also

conducted to learn the response of CAE-AD in different settings. Last, visualization experiments are carried out to demonstrate the representation of latent variables.

4.1. Datasets

Three public datasets are adopted in our experiments. The properties of the datasets are summarized in Table 1.

Server Machine Dataset (SMD)[1]. SMD is a new 5-week-long dataset from a large Internet company collected and publicly published by [34]. SMD dataset is divided into 28 subsets and each subset is collected from a server machine with 38 monitoring indicators.

Soil Moisture Active Passive (SMAP)[2] *satellite Datasets and Mars Science Laboratory (MSL)*[2] *rover Datasets.* SMAP and MSL datasets are real-world and expert-labeled telemetry data from NASA [18]. The SMAP/MSL datasets contain 55/27 subsets, each of which has 25/55 channels.

4.2. Evaluation Metrics

Precision (Pre), Recall (Rec), and F1 score are used to evaluate the performance of CAE-AD and baselines. In our experiments, we enumerate all thresholds and use the best F1-score to evaluate the performance, which is also called F1-best [26,34,9].

Anomalous observations usually appear in consecutive segments. Therefore, we utilize the point-adjust approach to detect anomalies. Specifically, if any observation in the ground truth abnormal segment is detected correctly, all observations in the segment are considered to be detected correctly. Otherwise, all the observations in the abnormal segment are not identified.

4.3. Baseline Models

We compare our model with nine unsupervised methods for MTS data anomaly detection as follows. Note that only our proposed framework CAE-AD considers the local invariant characteristics of MTS data.

iForest [24]. An ensemble model to isolate anomalies by randomly selecting a feature and randomly splitting the observations. However, temporal information is not considered in iForest.

EncDec-AD [26]. A Seq2seq model to detection anomaly for multivariate time series data. LSTM network is used to conduct an encoder and decoder, which can capture the temporal dependence of time series data.

LSTM-NDT [18]. A prediction-based model, which uses the LSTM network to predict the telemetry data and the prediction errors are the measures of anomaly scores.

OmniAnomaly [34]. A stochastic model to learn the robust representation of the MTS data. The reconstruction probabilities are utilized to calculate the anomaly scores.

Transformer-AD [38]. A prediction-based model using transformer encoder to detect anomalies. First, we construct a transformer encoder with multi-head attention mechanism and positional encoding following the literature. And then a mask is used to prevent future information leakage. Finally, a fully-connected layer is constructed to predict the MTS data, and the prediction errors are employed to calculate anomaly scores.

USAD [3]. A anomaly detection method based on two autoencoders, which are trained in an adversarial way to reconstruct data. The reconstruction errors of two autoencoders are utilized to calculate anomaly scores.

DROCC [14]. A one-class based anomaly detection method, which utilizes adversarial training to learn robust representation of data.

RANSynCoders [1]. A anomaly detection approach based on bootstrapped autoencoders, which learns synchronized representation of data.

GANF [8]. A density-based anomaly detection method. GANF uses a graph-based encoder to model relationships of different dimensions and utilizes a conditional normalizing flow to estimate the density. Lower densities indicate more likely anomalies.

4.4. Performance Evaluation

In our experiments, we implement CAE-AD based on PyTorch. The hidden size of the LSTM encoder is 64, and the dimension of latent variable z is 18 empirically. The deviation of Gaussian noise σ_1, σ_2 and the temperature parameter τ are set to 0.5, 0.5 and 0.25 empirically in our experiment, respectively. The hidden state of the decoder and the reconstruction data are concatenated and transformed by the fully-connected neural network to conduct the input of the LSTM decoder. The window size w and sliding window interval l are 36 and 10, respectively.

[1] *https://github.com/NetManAIOps/OmniAnomaly*
[2] *https://github.com/khundman/telemanom*

Table 1
The properties of the datasets.

Dataset	Subset	Dimension	Train	Test	Anomalies(%)
SMD	28	38	708405	708420	4.16
SMAP	55	25	135183	427617	13.13
MSL	27	55	58317	73729	10.72

Results and Analysis. Table 2 reports the precision, recall, and the F1-best scores of CAE-AD and other baseline models. We use bold font for the best F1 score and underline font for the second-best F1 score. As demonstrated in Table 2, the performance of CAE-AD outperforms all the baseline models. Specifically, the F1-best score of CAE-AD is 0.9376 on SMD datasets, which is slightly higher than that of OmniAnomaly and far better than the other methods. For SMAP and MSL datasets, CAE-AD achieves the F1-best score of 0.9302 and 0.9119 respectively. The case studies of CAE-AD are shown in Fig. 4a and Fig. 4b, demonstrating that CAE-AD can correctly detect the abnormal points. The abnormal timesteps of SMD machine-1–1 datasets are relatively concentrated, while the abnormal timesteps of SMD machine-2–3 datasets are relatively scattered. CAE-AD achieves stable performance in the two subsets, which further validates the ability of CAE-AD to learn normal patterns of MTS data.

IForest [24] randomly selects features and splits the values to separate abnormal points, but the temporal dependency of MTS data is not considered. Compared with SMAP and MSL datasets, the SMD datasets contain more extreme abnormal values with sudden peaks or valleys, which are easy for iForest to isolate. So iForest presents a lower performance on SMAP and MSL datasets with the F1-best score of 0.5738 and 0.4762, respectively.

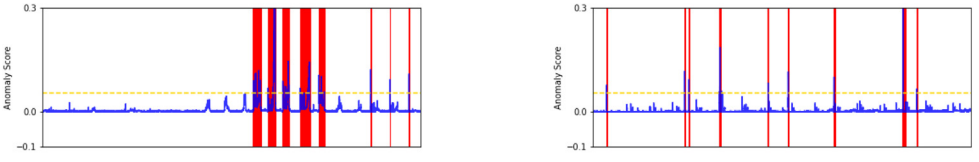
EncDec-AD [26] is a seq2seq based model with LSTM encoder-decoder structure. This method models normal time series behaviour, and utilizes reconstruction errors to detect anomalies. LSTM encoder and decoder are employed to capture temporal features of time series, so EncDec-AD achieves a better performance than iForest. But this method can not characterize the dynamics of time series.

LSTM-NDT [18] is a prediction-based model with one LSTM layer, which takes temporal information into consideration. For SMAP and MSL datasets, LSTM-NDT predicts the telemetry data for the next time step using the historical telemetry data for the first channel and the one-hot encoded information for the other channels. The results show that LSTM-NDT achieves high F1-best scores of 0.9147 on SMAP datasets. However, due to the complex network states and high dimensional data on SMD dataset, LSTM-NDT fails to capture the normal pattern and performs poorly.

OmniAnomaly [34] conducts a stochastic recurrent neural network to learn the robust representation of MTS data and achieves the second F1-best scores on SMD datasets. However, OmniAnomaly ignores the local invariant information, and the noise data can be given high anomalous scores that may be mistaken for abnormal points. CAE-AD introduces instance contrasting, which is helpful for latent variables to learn the local invariant property of MTS, and data augmentation methods

Table 2
Experiment Results: the precision, recall and $F1_{best}$ scores of baselines and CAE-AD. The highest $F1_{best}$ score is bolded, and the second best result is underlined.

Method	SMD			SMAP			MSL		
	Pre	Rec	$F1_{best}$	Pre	Rec	$F1_{best}$	Pre	Rec	$F1_{best}$
iForest [24]	0.5424	0.8957	0.6757	0.6481	0.5147	0.5738	0.5556	0.4167	0.4762
EncDec-AD [26]	0.9014	0.6764	0.7729	0.9814	0.7794	0.8688	0.931	0.75	0.8307
LSTM-NDT [18]	0.4726	0.7213	0.5711	0.8676	0.9672	<u>0.9147</u>	0.8056	0.9375	0.8656
OmniAnomaly [34]	0.9260	0.9149	<u>0.9204</u>	0.9502	0.5482	0.6953	0.9245	0.8502	0.8858
Transformer-AD [38]	0.8390	0.7242	0.7745	0.9076	0.8676	0.8872	0.9667	0.8056	0.8787
USAD [3]	0.7951	0.9418	0.8622	0.9032	0.8235	0.8615	0.8684	0.9167	<u>0.8918</u>
DROCC [14]	0.7469	0.7905	0.7681	0.4788	0.9881	0.6451	0.5944	0.9641	0.7354
RANSynCoders [1]	0.7536	0.8451	0.7967	0.9413	0.2116	0.3455	0.2862	0.9385	0.4386
GANF [8]	0.6408	0.8532	0.7319	0.6141	0.9869	0.7571	0.4720	0.9854	0.6383
CAE-AD	0.9265	0.9491	**0.9376**	0.8824	0.9836	**0.9302**	0.8611	0.9688	**0.9119**



(a) Anomaly Score of Machine-1-1 (b) Anomaly Score of Machine-2-3

Fig. 4. Case study of anomaly score on SMD machine-1–1 and SMD machine-2–3 datasets. The red region highlights the truth anomaly intervals.

in the time domain and frequency domain enhance the anti-noise capability of CAE-AD. Consequently, the OmniAnomaly method has a lower recall rate than our CAE-AD.

Transformer-AD [38] utilizes multi-head attention to learn the dependent information among different timesteps. As demonstrated in Table 2, Transformer-AD achieves F1-best scores of 0.7745 and 0.8787 on SMD and MSL datasets respectively, which performs better than iForest and LSTM-NDT. The reason is that the attention mechanism calculates attention scores of the input sequence and can capture historical information of input data at each time stamp, while LSTM-based methods selectively retain historical information, which may miss certain useful information. Therefore, Transformer-AD has better overall performance than LTSM-based methods, such as LSTM-NDT. However, Transformer-AD does not consider the inherent nature of MTS, such as local time invariance, and it performs inferior to CAE-AD.

USAD [3] conducts two autoencoders with one encoder and two decoders. In the training phase, USAD adversely trains two autoencoders, and aims to amplify the reconstruction errors of anomalies. With proper regularization, USAD is capable to utilize the advantages of autoencoder framework and achieves the second F1-best scores on MSL datasets. However, USAD framework may also amplify the noises in the inference phase. Consequently, USAD obtains inferior overall performances with respect to CAE-AD.

DROCC [14] utilizes the gradient-ascent method to generate anomalous instances, which alleviates the representation collapse problem. However, the real-world data distribution is complex and the generated anomalies may not match the ground-truth anomalies. CAE-AD only focuses on modelling normal data patterns, so it is demonstrated that DROCC performs inferior to CAE-AD on all three datasets.

RANSynCoders [1] learns the synchronous representation in MTS data, and applies bootstrapped autoencoders to learn the reconstructed lower bound and upper bound. Anomalies are identified by comparing the inputs and the decoded bounds. For SMAP and MSL datasets, the first dimension is telemetry data, and the other dimensions are 0/1 switch data. These switch data contain a lot of zero elements and it is difficult for the bootstrap aggregation based autoencoders to reconstruct them. Therefore, it shows that RANSynCoders performs poorly in SMAP and MSL datasets.

GANF [8] uses a graph-based dependency encoder to learn correlate information in series dimension and applies conditional normalizing flows to model the densities of MST. However, GANF ignores the intrinsic invariant characteristics and does not constrain the adjacency matrix of nodes. As demonstrated in Table 2, the performance of GANF is lower than CAE-AD.

In summary, CAE-AD outperforms the baseline methods on the three public datasets. The autoencoder structure of CAE-AD helps to obtain the reconstructions of time series, which is the key design that can be easily adapted to different datasets for the anomaly detection task. Moreover, the multi-grained contrasting method helps the CAE-AD model to learn multi-granularity temporal-dependent information, which enables the model to accurately construct a normal profile and makes abnormal data more discriminative.

4.5. Multi-grained Contrasting Analysis

In order to demonstrate the effectiveness of the contextual contrasting and instance contrasting, we evaluate the performance of CAE-AD and the three variants, namely CAE-Inst-AD, CAE-Cont-AD and AE-AD. Among them, CAE-Inst-AD retains the instance contrastive loss, while removing the contextual contrastive loss. CAE-Cont-AD leaves the contextual contrastive loss while removing the instance contrastive loss. AE-AD is constructed using the LSTM autoencoder without any contrastive loss. With the comparative experiments, we learn the response of CAE-AD to the proposed contextual contrastive loss and instance contrastive loss.

As shown in Fig. 5, CAE-AD achieves the best performance on the three datasets and AE-AD achieves the lowest overall performance. AE-AD models normal time series behaviour, and utilizes reconstruction errors to detect anomalies. However, without multi-grained contrasting methods, AE-AD can not comprehensively describe the multi-scale temporal information

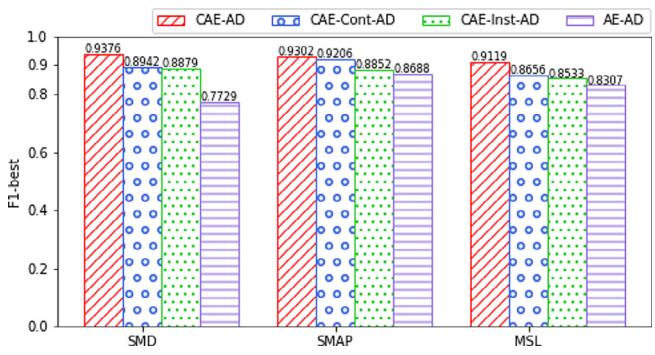


Fig. 5. F1-best of CAE-AD and the variants of CAE-Inst-AD, CAE-Cont-AD and AE-AD on SMD, SMAP and MSL datasets.

of data, so it fails to learn steady normal data patterns, which further indicates that proper regularization for autoencoder based model is necessary to improve the reconstruction abilities.

CAE-Cont-AD obtains F1-best scores of 0.8942, 0.9206, and 0.8656 on SMD, SMAP, and MSL datasets respectively. CAE-Inst-AD has lower performance compared to CAE-Cont-AD. The reason is probably that contextual contrastive loss aims to maximize the similarity of the observations at neighbouring timesteps and minimize the similarity at distant timesteps. By considering timestep-level observations, contextual contrasting can learn fine-grained contextual information of MTS data. However, instance contrastive loss maximizes the similarity of different views of the observations in the same window while minimizing the similarity of the observations in different windows within the minibatch. Considering window-level observations, instance contrasting can learn coarse-grained contextual information, which is also called local invariant characteristics of MTS data. Without contextual contrastive loss, CAE-Inst-AD may not perform well for reconstruction. Therefore, CAE-Cont-AD achieves higher F1-best scores than CAE-Inst-AD for SMD, SMAP, and MSL datasets.

Meanwhile, without instance contrastive loss, CAE-Cont-AD fails to capture local invariant characteristics of MTS data, so it demonstrates inferior performance to CAE-AD. Compared with the three variants, CAE-AD conducts multi-grained contrasting, so the latent variables can capture both temporal-dependent and local invariant characteristics, which validates that contextual contrasting and instance contrasting are complementary. Therefore, CAE-AD can extract the normal data pattern and demonstrate superior performance.

To sum up, the autoencoder model trained with multiple well-designed contrastive losses outperforms the typical autoencoder, which proves that our multi-grained contrasting methods are key designs in CAE-AD.

4.6. Parameter Sensitivity

In this subsection, we study the effects of different parameters on the performance of CAE-AD. The main parameters of our proposed method contain the window size w, sliding window interval l, dimension of latent variables z, and the dimension of hidden states h in the encoder. All the experiments in this section are conducted on machine-1–6 of the SMD dataset.

The first case we study is how window size w affects the performance of CAE-AD. The larger the window size, the more information the model can obtain. We use different window sizes $w = [10, 20, 30, 40, 100]$. As shown in Fig. 6(a), the best result is achieved for window size $w = 30$. It is observed that a smaller window size tends to yield a lower F1-best score,

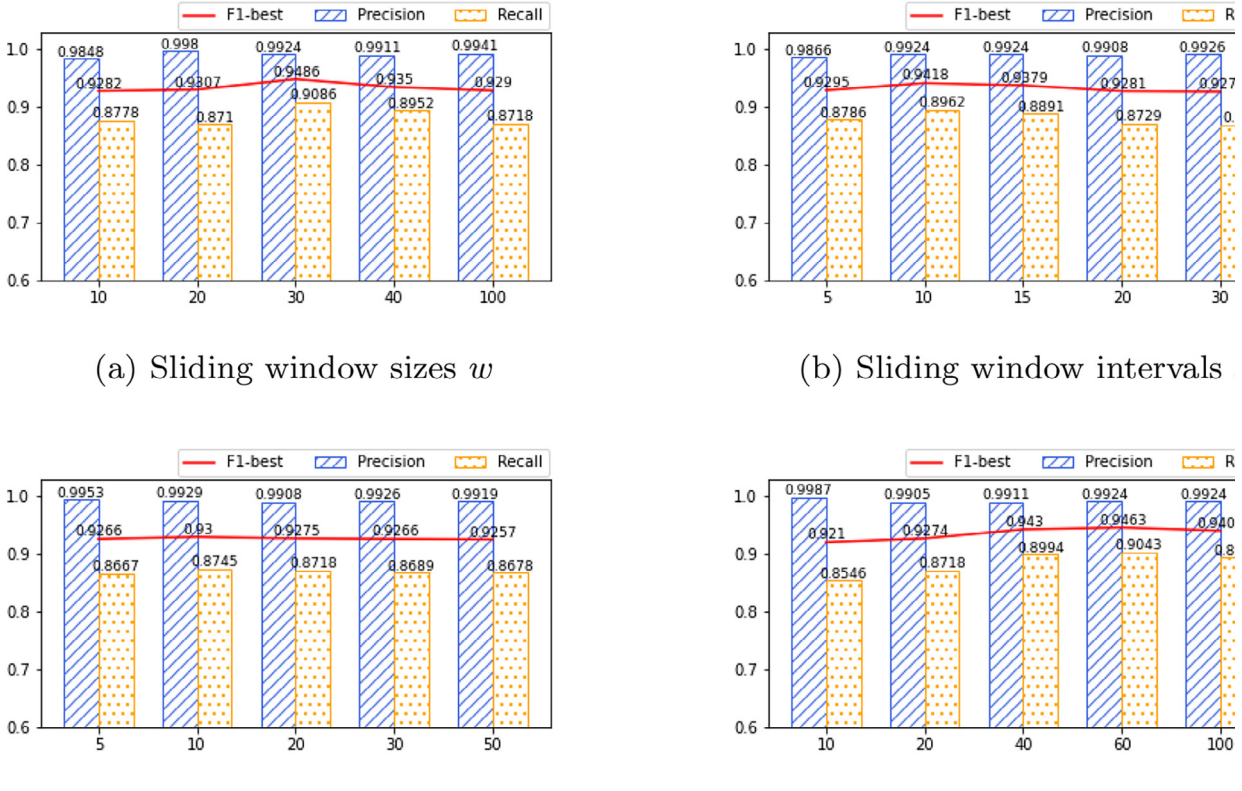


Fig. 6. The sensitivity of different parameters on machine-1–6 dataset of SMD. F1-best, Precision and recall are demonstrated to evaluate the responses of CAE-AD to (a) different window sizes w, (b) different sliding window intervals l, (c) different dimensions of latent variables z, (d) different dimensions of hidden states h in the encoder.

since CAE-AD is not capable of describing the correlation of different timesteps when the input sequence contains less information. Historical reconstruction errors are used to calculate the anomaly score of the data at the last timestep in the window. If a window is too large and may contain longer segments of anomalies, the normal data at the last timestep in the window will obtain a higher anomaly score and thus can be misjudged as an abnormal point. So, a larger window size demonstrates a lower F1-best score. For server machine datasets, a proper range from 20 to 40 can achieve better performance.

The second factor we learn is how CAE-AD responds to different sliding window intervals l. During data preprocessing, a sliding window is used to obtain a sequence of observations, and the sliding window interval affects the size of repeated observations. Fig. 6(b) summaries the obtained results for different intervals $l = [5, 10, 15, 20, 30]$. We observe that CAE-AD achieves the highest F1-best score when the interval reaches 10, and larger intervals lead to inferior performance. The reason may be that larger intervals contain less repeated observations, so CAE-AD may fail to capture the temporal dependency of continuous observations.

Next, we explore the impacts of the dimension of latent variables z. Fig. 6(c) demonstrates the F1-best scores of different dimensions of latent variables $z = [5, 10, 20, 30, 50]$. We can observe that the lower dimensions of latent variables obtain better performance. The reason may be that the encoder is capable of reducing the dimension of latent variables. A higher dimension of latent variables may contain more noises, so it is difficult to learn the normal pattern in high dimension features, which indicates that we can choose the dimension of latent variables lower than the dimension of raw observations.

Finally, we analyze the impact of the dimension of hidden states in the encoder. We explore different dimensions of hidden states $h = [10, 20, 40, 60, 100]$ in our experiments. As Fig. 6(d) shows, CAE-AD achieves a stable performance when hidden dimension reaches 40. When the hidden dimensions in the shared-weights encoder are higher than 40, it is observed that hidden dimensions are sightly sensitive to the performance of CAE-AD. The reason is that the multi-grained contrasting plays an important role in capturing normal data patterns, and the parameters or the network of the encoder can have a wider range of choices.

4.7. Ablation Study

In this section, we study the effect of each part of our model. Specifically, we repeat our experiments with/without key modules. As demonstrated in Table 3, our proposed model CAE-AD achieves the best performance on the SMD dataset. Directly removing the attention module and not using the contextual contrasting loss (CAE-AD_WO_ATTENTION_CONT), the F1 score drops by 22.77% compared to the Baseline. Further, we use a fully-connected neural network to replace the attention mechanism and retain the contextual contrastive loss (CAE-AD_WO_ATTENTION), and the performance is much improved compared to removing it directly, which proves that both the attention mechanism and contextual contrasting method work together to learn the temporal representation of data.

Besides, we also remove the time-domain data augmentation marked as CAE-AD_WO_FREQAUG and the frequency-domain data augmentation marked as CAE-AD_WO_TIMEAUG in turn and only use the single data augmentation method to generate different views of MTS. The results show that the model does not perform well in this case, which validates that various data augmentation methods help comprehensively describe dynamic characteristics in MTS. Finally, we replace the well-designed LSTM decoder with a simple fully-connected network (CAE-AD_WO_DECODER). The results show that the performance drops sharply compared to the Baseline. This is because the latent vector z does not directly contain the information at all time steps, so the sequential decoding method is necessary for the decoder to explore the dependency characteristics of MTS. These ablation studies validate that each module in our model is useful and necessary.

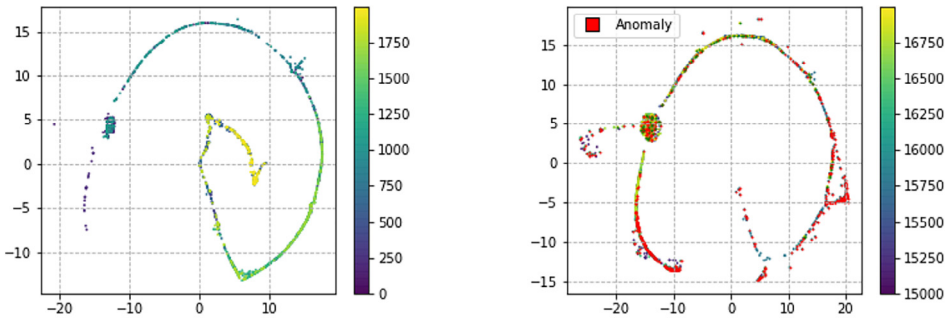
4.8. Visualization of Latent Variables

To further demonstrate the effectiveness of CAE-AD to learn the normal data pattern, we conduct an additional experiment for the visualization of latent variables.

CAE-AD is a reconstruction-based model. For a window of MTS data, the siamese encoders compress raw data to a latent variable, and then the decoder reconstructs the data. Contrastive Autoencoder learns the normal data patterns training on the normal data. If an abnormal observation appears as an input to the contrastive autoencoder, our model will encode

Table 3
Ablation study on SMD dataset.

Model	Pre	Rec	$F1_{best}$
CAE-AD	**0.9265**	**0.9491**	**0.9376**
CAE-AD_WO_ATTENTION_CONT	0.6550	0.7749	0.7099
CAE-AD_WO_ATTENTION	0.8172	0.8183	0.8177
CAE-AD_WO_FREQAUG	0.8678	0.6362	0.7342
CAE-AD_WO_TIMEAUG	0.6625	0.6879	0.6750
CAE-AD_WO_DECODER	0.4312	0.8182	0.5648



(a) Normal latent variables (b) Abnormal latent variables

Fig. 7. Visualization of latent variables on machine-1–1 of SMD. Each point means a latent variable and the corresponding color represents the timestep. Specially, red points are latent variables of abnormal points.

the observation to normal latent variable z. In this case, the reconstruction largely deviates from the actual one, and then the abnormal observation is detected correctly.

Specifically, we select the machine-1–1 subset of SMD datasets to obtain the latent variables, and employ t-Distributed t-SNE (Stochastic Neighbor Embedding) [37] to visualize the latent variables. As shown in Fig. 7(a), each point means a latent variable and the corresponding colour represents the timestep of the observation. Latent variables are close to each other in contextual data and far from each other among distant data, which indicates that contextual contrasting and instance contrasting are important for learning multi-scale contextual information. We highlight the latent variables of abnormal observations with the red colour, as shown in Fig. 7(b). The distribution of abnormal latent variables is similar to normal latent variables, indicating that CAE-AD can explicitly learn the representations of the normal data pattern.

5. Discussion

In this paper, we novelly combine the contrastive loss and mean square error (MSE) loss to jointly train a well-designed contrastive antoencoder framework for anomaly detection task. Typical antoencoder only uses MSE loss to train the model for extracting detailed information of reconstructed data, but the MSE loss is sensitive to anomalies, which may lead to inferior performance in anomaly detection task. For example, if the training data contain very few anomaly points, then MSE loss will penalize the model to fit these anomalies, which makes autoencoder model fail to construct a normal profile. Contrastive learning methods are able to learn invariant information in the data, so the contrastive loss is not sensitive to noisy data. However, is it feasible to use only contrastive loss to learn normal data pattern?.

To discuss this question, we simplify the contrastive loss in Eq. (5) as the following formula:

$$\mathcal{L}_{inst} = -\mathop{\mathbb{E}}_{\mathcal{X}}[\log \frac{f_{en}(x_t, z_t)}{\sum_{x_j \in \mathcal{X}} f_{en}(x_j, z_t)}], \tag{10}$$

Where numerator is the positive sample, and the denominator is the sum of one positive sample and all negative samples. In our case, $f_{en}(x_t, z_t) = \exp(\cos(z_t, z_t^+)/\tau)$. The contrastive loss can be considered as categorical cross-entropy loss with probability $p(d = t|\mathcal{X}, z_t) \propto f_{en}(x_t, z_t)/\sum_{x_j \in \mathcal{X}} f_{en}(x_j, z_t)$ to correctly classify the positive sample, where the indicator $d = t$ means x_t is a positive sample.

As proven in previous literature [29], $f_{en}(x_t, z_t) \propto p(x_t|z_t)/p(x_t)$, which means this function descripts the mutual information between x_t and z_t. However, in practice, adjacent or periodic time series segments $x_j (j \neq t)$ may also have strong correlations with z_t and the mutual information between them will be large, resulting in a higher probability $p(d = j|\mathcal{X}, z_t)$. Thus, x_j will be misclassified as a positive sample. This problem can be avoided by using a MSE loss to encourage the model to learn details of reconstruction information. Consequently, only contrastive loss is insufficient to learn the optimal normal pattern.

We carefully combine the advantages of the contrastive loss and MSE loss and propose a well-designed contrastive autoencoder framework. In our proposed framework, the autoencoder is capable of capturing detailed information to reconstruct input time series and the contrastive loss can encourage the model to learn invariant information, which improves the model's ability to deal with noisy data. As the experimental results shown, this combination is proved to be effective and necessary for improving anomaly detection performance.

6. Conclusion

This paper proposes a novel contrastive autoencoder for anomaly detection in MTS data, namely CAE-AD. The CAE-AD framework first generates two different views for each segment by applying data augmentation in both the time domain

and frequency domain. Then multi-grained contrasting methods are proposed to learn robust representation in multiple scales of MTS. Contextual contrasting learns temporal features by applying attention mechanism, and instance contrasting learns invariant features within two augmented views of the same sample. The multi-grained contrasting methods are able capture temporal-dependent and local invariant characteristics in MTS data. The experimental results show that CAE-AD outperforms the baseline models and each module in CAE-AD is effective. Furthermore, visualization of learned representations demonstrates the capability of CAE-AD to model the normal data pattern.

For future work, the dynamic threshold selection method will be explored, and more real-world data will be used for experiments to improve the robustness of CAE-AD. We will extend the CAE-AD model to multi-modal datasets and consider improved methods for the long-tail problem in various datasets.

CRediT authorship contribution statement

Hao Zhou: Conceptualization, Methodology, Software, Investigation, Visualization, Formal analysis, Writing - original draft. **Ke Yu:** Conceptualization, Funding acquisition, Resources, Supervision, Writing - review & editing. **Xuan Zhang:** Funding acquisition, Supervision, Writing - review & editing. **Guanlin Wu:** Data curation, Investigation, Validation. **Anis Yazidi:** Investigation, Supervision, Writing - review & editing.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is partially supported by: the National Natural Science Foundation of China under the Grant No. 61601046 and 61171098; the 111 Project of China under the Grant No. B08004; and the project Spacetime Vision: Towards Unsupervised Learning in the 4D World financed by the EEA and Norway Grants 2014–2021 under the Grant No. EEA-RO-NO-2018–04. This work is also supported by BUPT Excellent Ph.D. Students Foundation under the Grant No. CX2022149.

References

[1] Ahmed Abdulaal, Zhuanghua Liu, Tomer Lancewicki, Practical approach to asynchronous multivariate time series anomaly detection and localization, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2485–2494.
[2] Junfeng An, Haoyang Luo, Zheng Zhang, Lei Zhu, Lu. Guangming, Cognitive multi-modal consistent hashing with flexible semantic transformation, Information Processing & Management 59 (1) (2022) 102743.
[3] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, Maria A Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404.
[4] Sabyasachi Basu, Martin Meckesheimer, Automatic outlier detection for time series: An application to sensor data, Knowledge and Information Systems 11 (2) (2007) 137–154.
[5] Wanpracha Art Chaovalitwongse, Ya.-Ju. Fan, Rajesh C Sachdeo, On the time series *k*-nearest neighbor classification of abnormal brain activity, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 37 (6) (2007) 1005–1016.
[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.
[7] Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, Xiuzhen Cheng, Learning graph structures with transformer for multivariate time series anomaly detection in iot, IEEE Internet of Things Journal (2021).
[8] Enyan Dai, Jie Chen, Graph-augmented normalizing flows for anomaly detection of multiple time series, in: International Conference on Learning Representations, 2022, pp. 1–16.
[9] Liang Dai, Tao Lin, Chang Liu, Bo Jiang, Yanwei Liu, Zhen Xu, and Zhi-Li Zhang. Sdfvae: Static and dynamic factorized vae for anomaly detection of multivariate cdn kpis. In Proceedings of the Web Conference 2021, pages 3076–3086, 2021.
[10] Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. arXiv:1702.05538, 2017.
[11] 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. arXiv:2106.14112, 2021.
[12] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, Christopher Leckie, High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning, Pattern Recognition 58 (2016) 121–134.
[13] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. arXiv preprint arXiv:2005.12766, 2020.
[14] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification. In International Conference on Machine Learning, pages 3711–3721. PMLR, 2020.
[15] Lansheng Han, Man Zhou, Wenjing Jia, Zakaria Dalil, Xu. Xingbo, Intrusion detection model of wireless sensor networks based on game theory and an autoregressive model, Information sciences 476 (2019) 491–504.
[16] Kaiming He, Haoqi Fan, Wu. Yuxin, Saining Xie, Ross Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.
[17] David J Hill, Barbara S Minsker, Anomaly detection in streaming environmental sensor data: A data-driven modeling approach, Environmental Modelling & Software 25 (9) (2010) 1014–1022.

[18] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, Tom Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 387–395.
[19] Eamonn Keogh, Jessica Lin, Fu. Ada, Hot sax: Efficiently finding the most unusual time series subsequence, in: Fifth IEEE International Conference on Data Mining (ICDM), IEEE, 2005, p. 8.
[20] Istvan Kiss, Béla Genge, Piroska Haller, Gheorghe Sebestyén, Data clustering-based anomaly detection in industrial control systems, in: 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), IEEE, 2014, pp. 275–281.
[21] Hans-Peter Kriegel, Matthias Schubert, Arthur Zimek, Angle-based outlier detection in high-dimensional data, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 444–452.
[22] 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 International Conference on Artificial Neural Networks, pages 703–716. Springer, 2019.
[23] Dapeng Liu, Youjian Zhao, Haowen Xu, Yongqian Sun, Dan Pei, Jiao Luo, Xiaowei Jing, and Mei Feng. Opprentice: Towards practical and automatic anomaly detection through machine learning. In Proceedings of the 2015 Internet Measurement Conference, pages 211–224, 2015.
[24] Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou, Isolation forest, in: Eighth IEEE International Conference on Data Mining (ICDM), IEEE, 2008, pp. 413–422.
[25] Xuanqing Liu, Yu. Hsiang-Fu, Inderjit Dhillon, Cho-Jui Hsieh, Learning to encode position for transformer with continuous dynamical model, in: International Conference on Machine Learning, PMLR, 2020, pp. 6327–6335.
[26] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. arXiv:1607.00148, 2016.
[27] Xuying Meng, Suhang Wang, Zhimin Liang, Di Yao, Jihua Zhou, Yujun Zhang, Semi-supervised anomaly detection in dynamic communication networks, Information Sciences 571 (2021) 527–542.
[28] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, Sheraz Ahmed, Deepant: A deep learning approach for unsupervised anomaly detection in time series, IEEE Access 7 (2018) 1991–2005.
[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv:1807.03748, 2018.
[30] Daehyung Park, Yuuna Hoshi, Charles C Kemp, A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder, IEEE Robotics and Automation Letters 3 (3) (2018) 1544–1551.
[31] Daehyung Park, Hokeun Kim, Yuuna Hoshi, Zackory Erickson, Ariel Kapusta, Charles C Kemp, A multimodal execution monitor with anomaly classification for robot-assisted feeding, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 5406–5413.
[32] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. arXiv:1803.02155, 2018.
[33] Taeshik Shon, Jongsub Moon, A hybrid machine learning approach to network anomaly detection, Information Sciences 177 (18) (2007) 3799–3821.
[34] Su. Ya, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, Dan Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837.
[35] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284, 2022.
[36] Mehmet Turkoz, Sangahn Kim, Youngdoo Son, Myong K Jeong, Elsayed A Elsayed, Generalized support vector data description for anomaly detection, Pattern Recognition 100 (2020) 107119.
[37] Laurens Van Der Maaten, Accelerating t-sne using tree-based algorithms, The Journal of Machine Learning Research 15 (1) (2014) 3221–3245.
[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, pages 5998–6008, 2017.
[39] Xu. Jiehui, Wu. Haixu, Jianmin Wang, Mingsheng Long, Anomaly transformer: Time series anomaly detection with association discrepancy, in: International Conference on Learning Representations, 2022.
[40] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. arXiv:2106.10466, 2021.
[41] Yang Zhang, Nicholas A.S. Hamm, Nirvana Meratnia, Alfred Stein, M. Van De Voort, Paul J.M. Havinga, Statistics-based outlier detection for wireless sensor networks, International Journal of Geographical Information Science 26 (8) (2012) 1373–1392.
[42] Zheng Zhang, Haoyang Luo, Lei Zhu, Guangming Lu, and Heng Tao Shen. Modality-invariant asymmetric networks for cross-modal hashing. IEEE Transactions on Knowledge and Data Engineering, 2022.