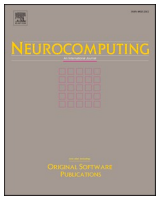
Neurocomputing 611 (2025) 128634

Contents lists available at ScienceDirect

Neurocomputing

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



Unsupervised anomaly detection of multivariate time series based on multi-standard fusion

Huixin Tian [a,c,*], Hao Kong [b,c], Shikang Lu [a,c], Kun Li [d]

[a] School of Control Science and Engineering, Tiangong University, Tianjin 300387, China
[b] School of Artificial Intelligence, Tiangong University, Tianjin 300387, China
[c] Tianjin Key Laboratory of Intelligent Control of Electrical Equipment, Tiangong University, Tianjin 300387, China
[d] School of Economics and Management, Tiangong University, Tianjin 300387, China

ARTICLE INFO

Communicated by Zidong Wang

Keywords:
Anomaly detection
Unsupervised learning
Multivariate time series
Industrial data
Real-time detection

ABSTRACT

With the wide application of the Internet of Things, industrial systems generate a large amount of multivariate time series data every day. Industrial data often contains a wide variety of anomalies. The existing anomaly detection methods have some shortcomings in dealing with large-scale, high-dimensional, and real-time industrial data. Particularly in the absence of prior knowledge, unsupervised learning methods are easily influenced by noise. Therefore, we propose a new unsupervised anomaly detection algorithm of multivariate time series based on multi-standard fusion (MSAD). Specifically, MSAD first classifies the data by analyzing the data density and sample spacing, thereby converting unsupervised anomaly detection into weakly supervised anomaly detection. Then, based on the characteristics of the data and deep learning technology, the degree of abnormality in the sample is comprehensively measured from four perspectives: holistic, local, time, and deep feature learning. At this stage, MSAD 's ability to learn edge features of data has been improved. Finally, based on MSAD, we design an anomaly detection method for industrial data streams in combination with data filtering techniques. The experimental results show that MSAD can detect anomalies more accurately and show stronger robustness to the influence of noise.

1. Introduction

The advent of big data technology has ushered in new challenges and revolutions across various sectors in today's society, and the application of data in the industrial field has become more and more significant [1]. This trend stems from the increasingly large and diverse data streams generated by industrial systems. These data not only contain important information about the production process, but also reflect various aspects of equipment performance, quality control, supply chain management, and holistic operations. However, due to the complexity of the equipment operation state, the uncertainty of the external environment, the multiple interferences in the production process, and other factors, industrial data often produces abnormal data in various forms in the production process. These abnormal

data can have several adverse effects on industrial production, such as reducing production efficiency, impacting product quality, and diminishing the accuracy of data analysis and prediction models [2]. Detecting and processing abnormal data in the industrial production process is crucial for improving efficiency, enhancing product quality, and reducing troubleshooting time. In this context, data anomaly detection has become a pivotal task in the industrial field. However, industrial data faces some special challenges, such as large-scale, high-dimensional, unlabeled, data noise, and real-time requirements [3]. Therefore, how to perform anomaly detection on industrial data under these special challenges is the current difficulty.

In recent years, with the rapid development of deep learning, it has demonstrated remarkable prowess in learning complex data representations, including high-dimensional data, time series data, and graph data [4,5]. Deep learning provides more powerful tools and methods for data analysis and application in various fields and has become an important research direction in the field of anomaly detection. In the field of deep learning, anomaly detection based on reconstruction has always been a high-profile research topic, which has led to extensive discussion and application. For example, Zhang et al [6]. combined autoencoder (AE) with GANs and proposed a mutual adversarial network (MAN) of switched sub-networks during training. Chen et al

* Corresponding author at: School of Control Science and Engineering, Tiangong University, Tianjin 300387, China.
E-mail address: icedewl@163.com (H. Tian).

https://doi.org/10.1016/j.neucom.2024.128634
Received 8 April 2024; Received in revised form 23 August 2024; Accepted 15 September 2024
Available online 19 September 2024

[7]. combined ensemble learning with GANs and proposed a supervised anomaly detector (EAL-GAN) that combines a generator with multiple discriminators. Multiple discriminators use integrated strategies to combat training, and the generator can fully learn the complex correlation of data. There are also some methods based on the encoder-decoder model [8,9]. Although these studies have made great contributions to anomaly detection, there are still some problems in practical applications. For example, the particularity of industrial data is that it is usually unlabelled because monitoring and tagging data is an expensive and time-consuming task in actual industrial production and is often difficult to implement, so requiring a large amount of labeled data to train the model may not be possible. Although the reconstruction-based method is also applicable to unsupervised anomaly detection [10,11], there are still unresolved problems in actual deployment. In certain scenarios, the boundary distribution between normal and abnormal data may overlap, causing numerous normal instances to be incorrectly classified as exceptions, which leads to more false positives in the detection process. Moreover, in the realm of unsupervised anomaly detection, reconstruction-based methods frequently fall short of capturing the boundary distribution between normal and abnormal data [12]. This often results in higher false alarm rates compared to semi-supervised detection methods, introducing additional complexity to anomaly detection.

In the industrial production process, real-time anomaly detection plays a vital role. For example, when anomalies may pose safety risks, such as chemical leakage or the risk of fire, timely detection of anomalies can prevent hazardous consequences. In continuous production processes, such as refining, chemical, or electricity production, real-time anomaly detection is required to maintain the continuity of production and avoid unnecessary downtime. Tang et al [13]. introduced a composite semantic enhancement encoder (CSAE) based on self-supervised comparative learning, which realized real-time detection of high-dimensional abnormal patterns in industrial environments through two data enhancement layers and a constraint layer. However, the complexity of this method is very high. Marco et al [14]. screened actual industrial time series data, selected portions that might have anomalies, preprocessed them to obtain a sliding window for real-time detection, and then used a deep learning autoencoder model to detect point anomalies. Real-time anomaly detection is performed by reconstructing each window and evaluating the reconstruction error of the last point. However, this method is only applicable to detecting point anomalies, requiring the reconstruction of each sliding window and evaluating the reconstruction error of its last point, which may increase computational complexity. Zhang et al [15]. proposed an anomaly detection method based on kernel density estimation (KDE). This method uses a series of sliding windows to enhance the detection effect of context anomalies. First, the data is divided into several equal and non-repetitive small windows, and KDE is applied to each small window for density estimation. Then, the anomaly is judged by comparing the difference between the last window (the window where the detection point is located) and other small windows. However, the KDE-based method requires manual tuning of parameters for different data sources, which takes a lot of time and effort. The KDE model needs to be reconstructed every time a new data point arrives, which also consumes a lot of computing resources. The current anomaly detection methods usually focus on improving the accuracy, while ignoring the real-time anomaly detection requirements of data streams [16]. Some models perform well in processing high-time complexity data, but they often require a lot of computing resources.

To address the above challenges, we propose a new unsupervised anomaly detection algorithm of multivariate time series based on multi-standard fusion (MSAD). MSAD extracts data information from multiple angles and focuses on the edge features of the data. It not only fully learns the characteristics and distribution of data but also utilizes the powerful fitting ability of deep learning to measure the degree of abnormality of data from various angles, and it is also suitable for real-time streaming data. The primary task of MSAD is to classify data through data density and sample spacing analysis, add labels to the samples according to their distribution, and transform the unsupervised method into a weakly supervised one. On this basis, the degree of abnormality of the sample is measured from three perspectives: holistic, local, and time. Then, we will select samples with low anomaly to train the variational generative model. To increase the generalization of the algorithm, we use the generative adversarial method to train the variational generative model and introduce an auxiliary classifier in the process of generative adversarial to strengthen the learning ability of the variational

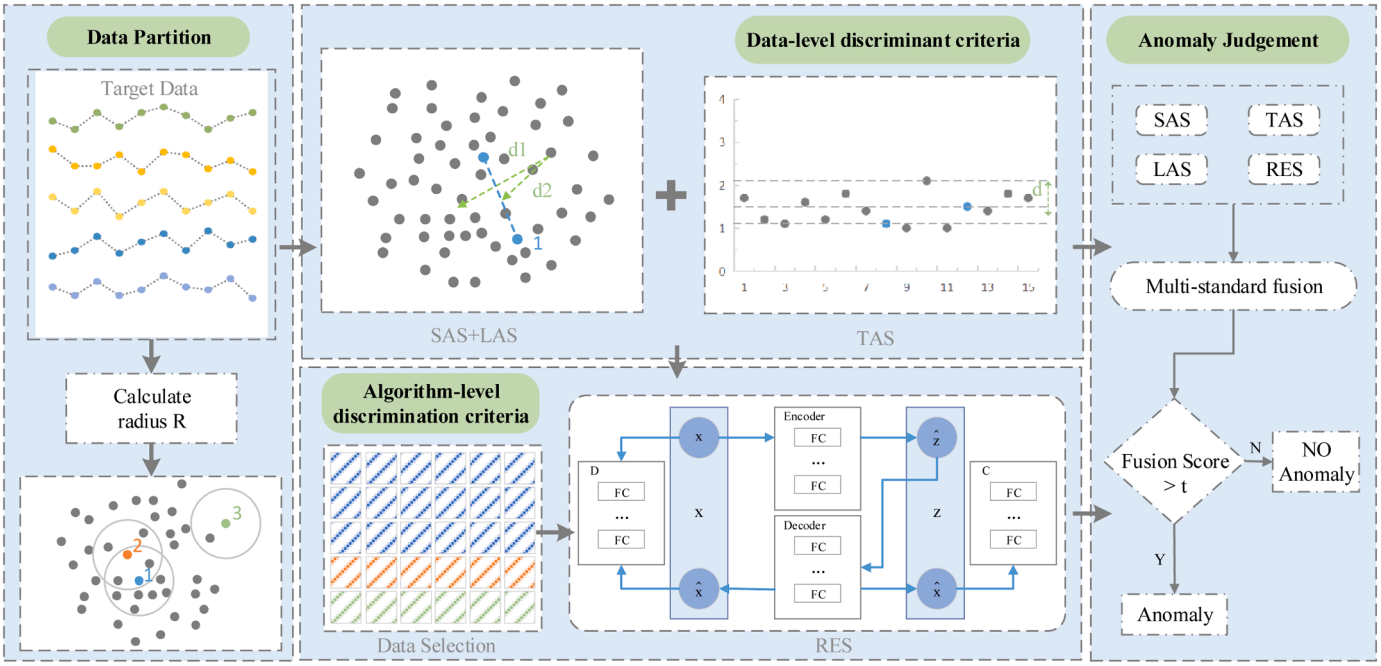


Fig. 1. The process of MSAD.

generative model for edge data. Finally, we use data filtering and sliding window methods to achieve real-time detection of data. In summary, the main contributions of this paper are as follows:

(1) To better learn the complex distribution of industrial data, we designed a data partitioning method that divides the data into three regions based on density. This approach transforms unsupervised anomaly detection into weakly supervised anomaly detection, allowing the detection algorithm to more effectively learn the characteristics of normal samples.

(2) In light of the complex distribution and diversity of abnormal samples in industrial data, we have defined a new unsupervised anomaly detection method based on multi-angle information fusion. This method detects anomaly samples from multiple perspectives, including holistic, local, time, and depth feature extraction, with particular consideration given to the boundary samples of the dataset.

(3) To achieve real-time detection of industrial streaming data, we proposed an unsupervised real-time anomaly detection algorithm based on sample replacement, building on MSAD. This method accurately detects anomalies, reduces false positive rates, and conserves computational resources.

The organization of the remaining sections in this paper is as follows: Section 2 provides a detailed introduction to the framework and specific implementation method of MSAD. In Section 3, we present the experimental results and in-depth analysis, and the concluding Section 4 summarizes the entire paper and provides future perspectives.

2. MSAD

In this paper, an unsupervised anomaly detection algorithm MSAD for multivariate time series based on multi-standard fusion is proposed. The framework is shown in Fig. 1. MSAD comprises four parts: the data division section, the data-level anomaly discrimination standard calculation section, the deep learning-level anomaly discrimination standard calculation section, and the anomaly determination section. Specifically, the raw data is first processed and then categorized into three classes based on its distribution. On this basis, anomaly detection criteria at the data level, namely SAS, LAS, and TAS, are calculated. Then, data is selected based on SAS, LAS, and TAS to train the reconstruction network. Finally, anomalies are determined based on the multiple obtained anomaly criteria. We provides a detailed introduction to the method of data partitioning in Section 2.1, describes the anomaly discrimination criteria at the data level in Section 2.2, and elaborates on the anomaly discrimination criteria at the algorithm level in Section 2.3. Section 2.4 discusses the detailed process of anomaly detection using MSAD, while Section 2.5 focuses on the realization of real-time anomaly detection with MSAD.

2.1. Data partition

In comparison to unsupervised anomaly detection, semi-supervised anomaly detection is generally more accurate. This is attributed to its utilization of solely normal data during training, which helps the model fully learn the characteristics of normal data and avoids the influence of outliers on the model. However, semi-supervised methods usually require a large amount of labeled data, and in areas such as industrial production, the acquisition of data labels may not be easy. Therefore, for the integration of deep learning technology into the anomaly detection of intricate industrial data, this paper introduces a data category division method based on data density and sample spacing. Specifically, we divide the samples into three different categories according to the density of the data and the distance between them, and artificially add labels to the data, thereby converting unsupervised anomaly detection into weakly supervised anomaly detection.

Given a time series data set $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_t, \boldsymbol{X}_N\} \in R^{N*l}$, where $\boldsymbol{X_t} = \{x_1, x_2, \cdots, x_l\}$, that is, the training set has N samples, and each sample has l eigenvalues. We divide the samples into three categories according to the density and distance of the data, which are dense samples, sub-dense samples, and edge samples. The specific definitions are as follows:

- Dense sample ϑ_{ds}: For instance $\boldsymbol{X}_i$, if there are no less than ns instance neighbors in a circle with a center of $\boldsymbol{X}_i$ and a radius of R, such instance $\boldsymbol{X}_i$ is called a dense sample.
- Sub-dense sample ϑ_{sds}: For instance $\boldsymbol{X}_i$, if there are less than ns instance neighbors in a circle with a center of $\boldsymbol{X}_i$ and a radius of R, but there are dense instances $\boldsymbol{X}_j$, such instance $\boldsymbol{X}_i$ is called sub-dense sample.
- Edge sample ϑ_{es}: For instance $\boldsymbol{X}_i$, if there are fewer than ns instance neighbors in a circle with a center of $\boldsymbol{X}_i$ and a radius of R, and there is no dense instance $\boldsymbol{X}_j$, such instance $\boldsymbol{X}_i$ is called an edge sample.

2.1.1. Radius Calculation

In order to divide the data accurately, it is very important to determine the radius (R) and the number of instances (ns) in the circle. We determine the radius according to the distance between each sample and its ns nearest neighbor sample. To be specific, for the data set $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_t, \boldsymbol{X}_N\} \in R^{N*l}$, first select an appropriate nc according to the distribution $\boldsymbol{X}$. This paper uses the K-means clustering method to cluster the data $\boldsymbol{X}$. Next, we operate on each cluster. For each instance $\boldsymbol{X}_i$ in the cluster, calculate the distance d between $\boldsymbol{X}_i$ and its surrounding instances, and select the nsth value as the distance d_i of the current instance $\boldsymbol{X}_i$ in the order of distance from near to far. To save time in the calculation of this step, this paper uses the ball tree to search for the

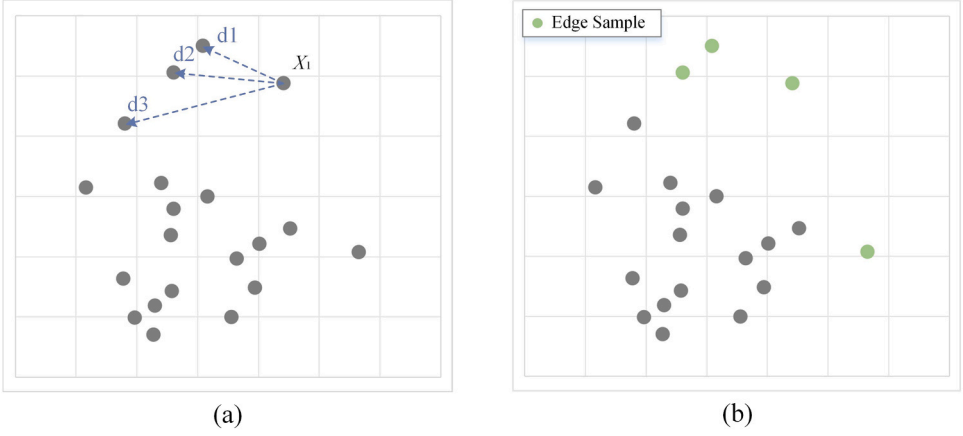


Fig. 2. Radius calculation example.

adjacent points of the instance $\boldsymbol{X}_i$. After calculating the distance d_i of all instances in each cluster, we sort the distance values and remove 20 % of the farther values. Finally, the selected average distance is calculated as the radius R of the data category. The specific radius calculation process is described in Algorithm 1.

Algorithm 1. : Radius calculation

Input: Data $\boldsymbol{X}$, The number of clustering nc, The number of samples ns

Output: Radius R

Initialize $CX1 \leftarrow \varnothing$

Clustering $\boldsymbol{X}$ using the K-means method yields the cluster ensemble $\boldsymbol{C}$

for each c_i in $\boldsymbol{C}$ **do**

  Initialize $CX2 \leftarrow \varnothing$

  Create a ball tree b_i for c_i

  for each $\boldsymbol{X}_j$ in c_i **do**

    Use the ball tree b_i to search the nsth nearest neighbor $\boldsymbol{X}_k$ of $\boldsymbol{X}_j$

    Calculate the distance d between $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$, then $CX2 \leftarrow d$

  end for

  Sort $CX2$ in ascending order, then extract the top 80% values from the sorted $CX2$ and place them into $CX1$

end for

$R \leftarrow$ mean($CX1$)

return R

To better understand Algorithm 1, we take Fig. 2 as an example to illustrate.

Firstly, we assume that the number of clusters is 1 and the number of instances in the circle is 3. Then we calculate the current distance d_i for each instance $\boldsymbol{X}_1$. Taking $\boldsymbol{X}_1$ in Fig. 2(a) as an example, the distance between it and the surrounding adjacent points is calculated. After calculating the distance between $\boldsymbol{X}_1$ and the adjacent points as d_1、d_2、and d_3, we select d_3 as the current distance of $\boldsymbol{X}_1$.

After calculating the current distance of each sample, the distance values are sorted, and the edge samples with larger distance values are excluded, as shown in Fig. 2(b). The mean value of the current distance of the remaining samples is calculated as the radius R. By excluding some outliers and noise, a more accurate and reliable radius R is obtained.

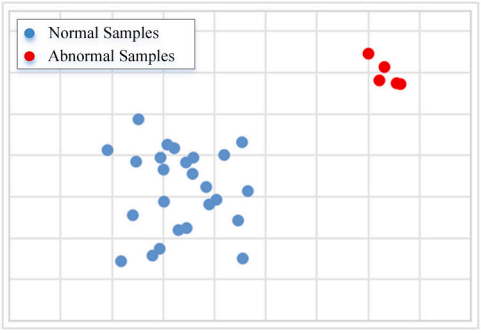


Fig. 3. Dense anomaly distribution.

2.1.2. Data classification

After determining the radius R, we divide the data into three different categories. Firstly, a ball tree is constructed for data $\boldsymbol{X}$, and the search radius is set to R. Then for each instance $\boldsymbol{X}_i$ in the data, calculate the number of samples in a circle centered on $\boldsymbol{X}_i$ and with radius R. If the number of samples is not less than ns, $\boldsymbol{X}_i$ is divided into the first category, that is, dense sample ϑ_{ds}. If the number of samples is less than ns, but there are dense samples in the circle, it is divided into the second category, that is, sub-dense sample ϑ_{sds}. If the number of samples is less than ns and there is no dense sample in the circle, it is divided into the third category, that is, edge sample ϑ_{es}.

Generally speaking, the outliers in the data are irregular and rare, but there are also cases as shown in Fig. 3. The outliers show a relatively dense distribution, which has a great impact on the accuracy of data classification. Therefore, to divide dense samples more accurately, we increase the secondary screening of dense samples. It can be seen from Fig. 3 that it is not enough to consider only the number of samples in the circle, but also the number of samples around the circle. We will make a secondary screening of $\boldsymbol{X}_i$ according to the number of internal samples centered on the dense instance $\boldsymbol{X}_i$ and ranging from R to 1.5 * R, and change the category of $\boldsymbol{X}_i$ with a small number of samples to edge sample. The detailed process of data classification is shown in Algorithm 2.

Algorithm 2. : Data Classification

Input: Data $\boldsymbol{X}$, Radius R, The number of samples ns

Output: $Label$ - a list of the categories of each $\boldsymbol{x} \in \boldsymbol{X}$

Initialize $Label \leftarrow \varnothing$
Initialize $Label_2 \leftarrow \varnothing$
Create a ball tree b for $\boldsymbol{X}$
for each $\boldsymbol{X}_i$ in $\boldsymbol{X}$ **do**
    $M_1 \leftarrow$ Using b search to find the sample points within a circle centered at $\boldsymbol{X}_i$ with 1.5 * R
    $M_2 \leftarrow$ Search for sample points in M_1 whose distance from $\boldsymbol{X}_i$ is less than R
    $Label_2[i] \leftarrow$Search for the number of sample points in M_1 whose distance from $\boldsymbol{X}_i$ is great than R
    $num_1 \leftarrow$ Calculate the number of samples in M_2
    if $num_1 \geq ns$ **then**
        $Label[i] \leftarrow 1$
    else
        for each m_j in M_2 **do**
            $k \leftarrow$ The index of m_j in $\boldsymbol{X}$
            $d \leftarrow$ Calculate the distance between $\boldsymbol{X}_i$ and m_j
            if $Label[k] == 1$ and $d \leq R$ **then**
                $Label[i] \leftarrow 2$
            else
                $Label[i] \leftarrow 3$
            end if
        end for
    end if
end for
Sort $Label_2$, take the top 10% indices, and set the values at these indices in $Label$ to 3
return $Label$

In the previous section, we assume the number of clusters and the number of samples in the circle for the example of Fig. 2 and calculate the radius R. Now it is divided into categories, and the results are shown in Fig. 4.

2.2. Data-level discriminant criteria

In unsupervised anomaly detection, data-based methods play a crucial role. Due to the characteristics of unsupervised learning, we do not have prior label information to guide anomaly detection, so we need to rely on the inherent laws and structure of the data itself to identify outliers. To explore the characteristics of data comprehensively and deeply, this paper will examine the anomaly of data from three different perspectives: holistic, local, and time.

2.2.1. Spatial anomaly score

Clustering is often used in unsupervised anomaly detection. The degree of anomaly of the sample is measured by the distance from the sample to the center of the cluster. Normal data points tend to gather near the center of the cluster, while outliers are more likely to stay away from the center of the cluster. However, in unlabeled data, the location of the cluster center may be affected by outliers, resulting in the cluster center shifting to outliers, and the distance of some normal points will increase accordingly, resulting in some normal points being misjudged as outliers. To increase the accuracy of anomaly detection and obtain a higher recall rate, this paper only uses dense samples when calculating the location of the cluster center. Dense samples are relatively more likely to be normal samples, so it is more reliable to determine the distribution center of the data. We illustrate this with Fig. 5.

In general, the degree of anomaly is measured by calculating the distance from $\boldsymbol{X}_i$ to the cluster center. However, for the case shown in Fig. 6, it is not appropriate to directly use distance to measure the degree of sample anomaly. Although the distance d_1 from $\boldsymbol{X}_1$ to the cluster center is greater than the distance d_2 from $\boldsymbol{X}_2$ to the cluster center, it is clear that $\boldsymbol{X}_2$ is more likely to be abnormal than $\boldsymbol{X}_1$. Therefore, to better use the distance between the sample point and the cluster center point to measure the degree of abnormality of the sample, we add the operation of standard deviation to the calculation of distance, so that we can better measure the degree of abnormality of the sample.

We use $SAS(\boldsymbol{X}_i)$ to represent the spatial anomaly score of instance $\boldsymbol{X}_i$. For a given time series data $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_t, \boldsymbol{X}_N\} \in R^{N*l}$, the spatial anomaly score SAS is defined as:

$$SAS(\boldsymbol{X}_i) = \frac{dist(\boldsymbol{X}_i, \boldsymbol{X}_{ct})}{\sum_{j=1}^{n} dist(\boldsymbol{X}_j, \boldsymbol{X}_{ct})} \tag{1}$$

where

$$dist(\boldsymbol{X}_i, \boldsymbol{X}_{ct}) = \frac{\sqrt{\sum_{k=1}^{1} (x_{ik} - x_{ctk})^2}}{\sigma_X} \tag{2}$$

$dist(\boldsymbol{X}_i, \boldsymbol{X}_{ct})$ is the Euclidean distance between $\boldsymbol{X}_i$ and $\boldsymbol{X}_{ct}$, $\boldsymbol{X}_{ct}$ is the cluster center of dense samples, l is the characteristic number of a given data set $\boldsymbol{X}$, and σ_X is the standard deviation of $\boldsymbol{X}$.

The spatial anomaly score of the sample is calculated by Eq. (1), and we can measure the degree of anomaly from the holistic perspective of the data.

2.2.2. Local abnormal scores

In the previous section, this paper introduces the method of calculating the abnormal score of the data sample space in detail and considers the abnormal degree of the sample from the holistic perspective of data distribution. However, although this measure from a general perspective can reflect the degree of abnormality of the data to a certain extent, it has some limitations. Since the distribution of industrial data is often complex, there may be abnormal values within the internal scope, so only considering the degree of abnormality of the sample from the holistic perspective may miss some important outliers. Therefore, to detect anomalies more accurately, in this section, we will introduce a method LAS to measure the degree of sample anomalies from a local perspective.

Considering that the distribution of abnormal samples relative to normal samples is often more disordered and significantly deviates from normal data. Therefore, for each instance $\boldsymbol{X}_i$, we first search the adjacent points whose category is ϑ_{ds}, and then sort them according to the order of distance, select the first ns samples as $\widehat{\boldsymbol{X}}$, and calculate the center $\widehat{X}_{ct}$ of these samples. Finally, the distance between $\boldsymbol{X}_i$ and $\widehat{X}_{ct}$ is calculated as the local anomaly score of $\boldsymbol{X}_i$. We use $LAS(\boldsymbol{X}_i)$ to represent the local anomaly score of instance $\boldsymbol{X}_i$. For a given time series data $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_t, \boldsymbol{X}_N\} \in R^{N*l}$, the local anomaly score LAS is defined as:

$$LAS(\boldsymbol{X}_i) = \frac{dist(\boldsymbol{X}_i, \widehat{X}_{ct})}{\sum_{j=1}^{n} dist(\boldsymbol{X}_j, \widehat{X}_{ct})} \tag{3}$$

In this section, we use the degree of difference between each sample and the surrounding dense samples as an anomaly discrimination criterion by calculating the degree of difference between each sample and the surrounding dense samples. This method measures the degree of abnormality of each sample from a local perspective. Through this method, we can identify which data points are abnormal in the local range, to better understand and process the data.

2.2.3. Time anomaly score

For time series data, there are still some limitations in considering the degree of sample anomaly from the holistic and local perspectives. Time series data usually have continuity and time dependence[17], it is very important to consider the degree of anomaly in the time dimension for application scenarios such as anomaly detection.

Considering that the abnormal samples are more disordered and deviate from the normal data over some time, for each instance $\boldsymbol{X}_i$, we first search the adjacent points whose sampling time is similar and the category is ϑ_{ds}. Then, according to the order of the distance between $\boldsymbol{X}_i$ and these adjacent points, the first ns samples are selected as $\widehat{\boldsymbol{X}}$, and the center $\widehat{X}_{ct}$ of these samples is calculated. Finally, the distance between $\boldsymbol{X}_i$ and $\widehat{\boldsymbol{X}}_{ct}$ is calculated as the time anomaly score of $\boldsymbol{X}_i$. We use $TAC(\boldsymbol{X}_i)$ to represent the time anomaly score of instance $\boldsymbol{X}_i$. For a given time series data $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_t, \boldsymbol{X}_N\} \in R^{N*l}$, the time anomaly score TAS is defined as:

$$TAS(\boldsymbol{X}_i) = \frac{dist\left(\boldsymbol{X}_i, \widehat{\boldsymbol{X}}_{ct}\right)}{\sum_{j=1}^{n} dist\left(\boldsymbol{X}_j, \widehat{X}_{ct}\right)} \tag{4}$$

In this section, we calculate the difference between each sample and its adjacent dense samples in time as an anomaly criterion. This method focuses on accurately measuring the degree of abnormality of each sample from the perspective of time.

In Section 2.2, we conduct an in-depth analysis and discussion of anomaly discrimination criteria at the data level. We focus on evaluating the degree of data anomalies in a more comprehensive way from three perspectives. Firstly, we examine the holistic distribution of the data, conducting a comprehensive analysis of the dataset's structure and characteristics to identify irregular patterns or outliers. Secondly, we focus on the local information, that is, the degree of anomaly of the sample in a certain area of its distribution, to more accurately locate possible anomalies. Finally, we assess the sample differences before and after over time, tracking data changes through time series analysis to identify potential anomalies at different time points. By comprehensively analyzing these three dimensions, we gain a deeper understanding of potential abnormal features in the data, offering a more accurate foundation for anomaly detection.

2.3. Algorithm-level discrimination criteria

To effectively address large-scale, high-dimensional, and complex anomaly patterns in industrial data, this paper introduces an anomaly discrimination standard based on deep learning, building upon prior discussions. Specifically, we employ reconstruction to assess the abnormality of the data. The reconstruction-based anomaly discrimination method determines whether it is an anomaly according to the difference between the reconstructed samples, and encoding-decoding is an important reconstruction method. We aim to utilize the adversarial training idea to develop an encoding-decoding model with heightened

sensitivity to anomalies.

2.3.1. Adversarial training

The adversarial idea aims to promote the generator and discriminator to achieve Nash equilibrium by optimizing the generator and discriminator [18]. The generator G receives a random vector z and generates a generated sample as similar as possible to the real sample. The discriminator D identifies the samples generated by the generator from the input samples as much as possible, and the two form a mutually antagonistic relationship. Inspired by this adversarial training, this paper applies this idea to the training coding-decoding structure, uses the variational generative model as the generator, and uses the neural network with multiple fully connected layers as the discriminator. The strong learning ability of the variational generative model and the strong generalization ability of GAN are used for anomaly detection [19,20]. Specifically, the encoder compresses the input data into a low-dimensional representation z, and then the decoder decompresses $z$it into new data z'. At the same time, the discriminator receives z' and determines whether it is a real sample, guiding the generator to continuously optimize, and finally, the generator generates samples that are highly similar to the real sample. Due to the significant difference between the characteristics of abnormal samples and normal samples, it is difficult for the generated model to fully learn and master its characteristics. Therefore, in the process of data reconstruction, abnormal samples will be effectively detected.

In general, when using the variational generative model for reconstruction anomaly detection, the outliers in the input samples may hurt the training stability and model generalization ability of the variational generative model. Therefore, when using reconstruction for unsupervised anomaly detection, it is necessary to make the model learn the characteristics of abnormal samples as little as possible. We will calculate the comprehensive anomaly score $CAS(\boldsymbol{X}_i)$ (Comprehensive anomaly score) of the instance $\boldsymbol{X}_i$ according to the various anomaly standards obtained in Section 2.2, and eliminate the samples with larger anomaly scores before training the variational generative model.

$$CAS(\boldsymbol{X}_i) = SAS(\boldsymbol{X}_i) + LAS(\boldsymbol{X}_i) + TAS(\boldsymbol{X}_i) \tag{5}$$

In actual training, GAN has problems such as training difficulties, the loss of generators and discriminators that cannot indicate the training process, and the lack of diversity of generated samples [21,22]. Especially in the face of high-dimensional and complex industrial data, these problems are more obvious. The existence of these problems is related to the mechanism of GAN. The Nash equilibrium reached by confrontation training is only an ideal state. However, in actual training, due to various reasons, confrontation training often only reaches a pseudo-equilibrium state. To solve these problems, this paper proposes an improved method, which is based on Wasserstein distance and combined with Smooth-L1Loss loss function to improve the loss function of the generator, so that the generator can better learn the characteristics of normal samples.

When using Wasserstein distance as the loss function, generative adversarial networks, such as WGAN, are trained by minimizing the distance between $\boldsymbol{p}_z(\boldsymbol{x})$ and $\boldsymbol{p}_r(\boldsymbol{x})$. Similar to the training process of the original GAN, we first train the discriminant network D. The goal of the discriminator is to maximize the distribution difference between the real sample and the generated sample as much as possible, as shown in Eq. (6).

$$L_D = E_{x\sim P_r}[D(G(\boldsymbol{X}_i))] - E_{x\sim P_r}[D(\boldsymbol{X}_i)] \tag{6}$$

where $\boldsymbol{X}_i$ is the training data, and $G(\boldsymbol{X}_i)$ is the data generated by the generator G.

During the training of the generator, its objective is to minimize the distribution difference between real samples and generated samples as much as possible, while also being sensitive to anomalous samples. However, the presence of anomalous samples in the training data may lead to unstable effects on the model parameter updates. Therefore, when the error is large, G selects linear loss, which effectively prevents excessive influence from anomalous samples, as shown in Eq. (7). When the error is small, G chooses quadratic loss, providing smooth gradients to help the generator learn the features of normal samples more accurately, as shown in Eq. (8).

$$|\boldsymbol{X} - \boldsymbol{X}'| - 0.5 \tag{7}$$

$$0.5 * (\boldsymbol{X} - \boldsymbol{X}')^2 \tag{8}$$

Therefore, the loss function of the generator is shown in Eq. (9).

$$L_G = \begin{cases} 0.5 * (E_{x\sim P_r}[D(\boldsymbol{X}_i)] - E_{x\sim P_r}[D(G(\boldsymbol{X}_i))])^2 if |E_{x\sim P_r}[D(\boldsymbol{X}_i)] - E_{x\sim P_r}[D(G(\boldsymbol{X}_i))]| < 1, \\ |E_{x\sim P_r}[D(\boldsymbol{X}_i)] - E_{x\sim P_r}[D(G(\boldsymbol{X}_i))]| - 0.5 otherwise. \end{cases} \tag{9}$$

In this way, the model's response to outliers will be smoother, which can reduce the impact of outliers on the reconstruction model training, thereby improving the learning ability of normal sample characteristics.

2.3.2. Edge distribution of data

Sometimes the distribution of abnormal data overlaps with the edge distribution of normal data, which increases the difficulty of anomaly detection. Taking Fig. 7 as an example, it can be observed that the distribution of some abnormal samples is very similar to the edge distribution of the data. However, only using the generative adversarial training method may not be enough to fully capture the edge distribution of normal data. This may cause some outliers to be mistakenly identified as normal, that is, false positives. Therefore, learning the edge distribution of normal data helps to reconstruct the model to better identify anomalies.

To make the variational generative model better learn the edge characteristics of data, we introduce an auxiliary classifier C in generative adversarial training. We completed the division of data categories in Section 2.1, dividing the data into ϑ_{ds}、ϑ_{sds} and ϑ_{es}. We enhance the ability of the variational generative model to learn the potential features of edge distribution by classifying ϑ_{ds}、ϑ_{sds} and ϑ_{es} during adversarial training. We use the classification model with a multi-layer fully connected layer as the auxiliary classifier C of the anomaly detection algorithm. The loss function L_C uses the cross entropy loss function, L_C as shown in the equation:

$$L_C = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{M} y_{ic}\log(p_{ic}) \tag{10}$$

Where N is the number of training samples, M is the number of sample categories, y_{ic} is the ith sample category label c, and p_{ic} is the probability that the ith sample is category c.

Finally, considering the generative adversarial network and the auxiliary classifier, a joint loss including generation loss, discriminant loss, and classification loss is designed. We apply the classification loss L_C to the training of G and guide variational generative model to learn the distribution of normal data together with L_D. The final generation loss is as the equation:

$$L_G = L_G + \omega L_C \omega \in (0,1) \tag{11}$$

2.4. Abnormal determination

In Section 2.1, we classify the data in detail, which lays the foundation for further anomaly detection. In Section 2.2 and Section 2.3, we calculate the SAS, LAS, TAS, and RES for the dataset $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_t, \boldsymbol{X}_N\} \in R^{N*l}$, respectively. These analysis methods comprehensively consider various aspects of the data, encompassing holistic structure, local features, and temporal changes, to effectively reveal anomalies in the dataset. Ultimately, we compare the anomaly score with the pre-set threshold, thereby achieving the goal of anomaly detection. This comprehensive anomaly detection strategy allows for more accurate identification of potential anomaly patterns, offering an effective approach to enhance data quality and respond promptly to abnormal situations. Algorithm 3 outlines the specifics of MSAD.

Algorithm 3. : MSAD

Input: Data $\boldsymbol{X}$, The number of clustering nc, The number of samples ns

Output: The abnormal sample

1: R ← **Algorithm 1**

2: $Label$ ← **Algorithm 2**

3: Calculation of SAS, LAS, TAS and CAS

4: $\tilde{\boldsymbol{X}}$, $Label_2$ ← Based on CAS, remove the lowest 10% of samples, and obtain the remaining samples and their corresponding labels

5: **for** each epoch **do**

6: Initialize classifier C

7: Classifier C is trained based on $\tilde{\boldsymbol{X}}$ and $Label_2$

8: **end for**

9: **for** each epoch **do:**

10: Initialize generator VGM, discriminator D

11: **for** batch data from $\tilde{\boldsymbol{X}}$ and $Label_2$ **do**

12: Generate $\boldsymbol{X}'$ from VGM

13: dloss ← $L_D\left(\tilde{\boldsymbol{X}}, \boldsymbol{X}'\right)$

14: Update D by dloss

15: closs ← $L_C\left(\tilde{\boldsymbol{X}}, labels\right)$

16: gloss ← $L_G\left(\tilde{\boldsymbol{X}}, \boldsymbol{X}'\right)$

17: Update VGM by gloss and closs

18: **end for**

19: **end for**

20: Calculate reconstruction difference: $RES \leftarrow \left|\boldsymbol{X} - VGM\left(\boldsymbol{X}\right)\right|$

21: Calculate anomaly score: score ← $RES + CAS$

22: **if** score > threshold **then**

23: return abnormal sample

24: **end if**

25: **return** The abnormal sample

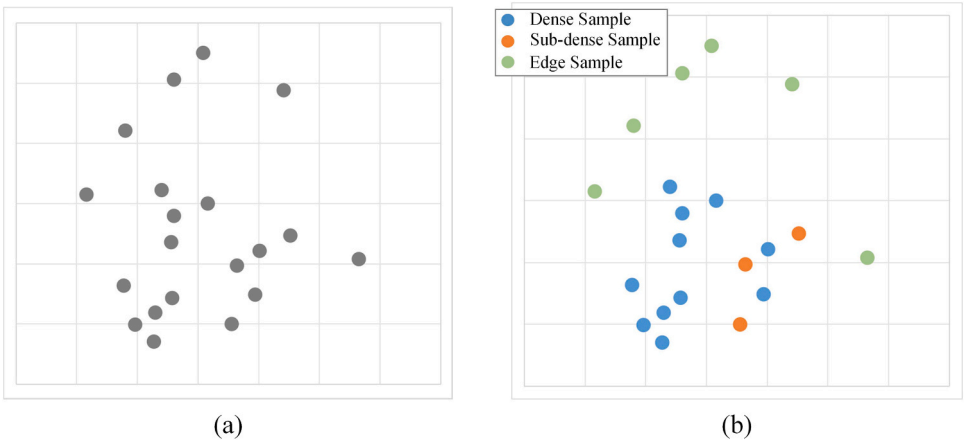


Fig. 4. Category division results.

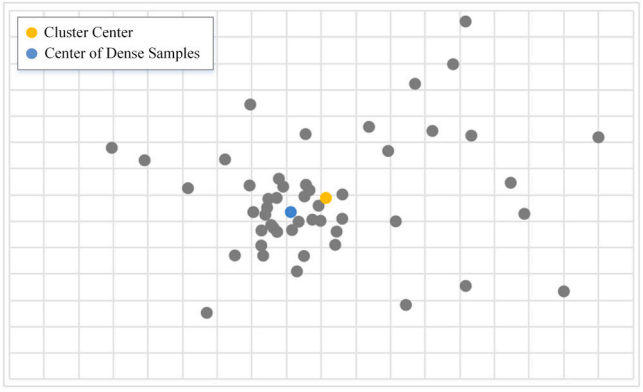


Fig. 5. Comparison of cluster center and dense sample center.

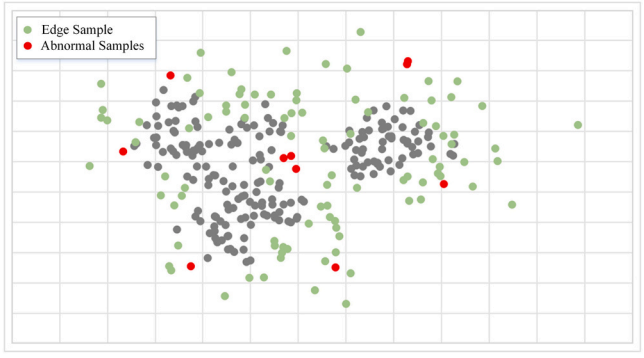


Fig. 7. Abnormal data and edge data distribution.

2.5. Real-time detection based on data filtering

Real-time anomaly detection has consistently held a crucial role in the industrial sector. However, owing to memory constraints, it is not feasible to fully retain historical data. Therefore, an algorithm must be devised to selectively preserve the most pertinent historical information. In this paper, real-time anomaly detection is achieved through the utilization of data filtering and sliding window technology, alongside various proposed standards for discriminating anomalies.

2.5.1. Data filtering

Data filtering technology comes into play after the completion of new sample detection. When $\boldsymbol{X}_{new}$ detection is completed, we will replace the samples in the original data with $\boldsymbol{X}_{new}$. Specifically, we first find the nearest sample to $\mathbf{X_{new}}$ in the original data. We use a layer-by-layer search to find the nearest neighbor sample of $\boldsymbol{X}_{new}$. If the nearest neighbor sample of $\boldsymbol{X}_{new}$ is a dense sample, we will replace it with $\boldsymbol{X}_{new}$ and inherit its category. If the nearest neighbor sample of $\boldsymbol{X}_{new}$ is not a dense sample, then we will use $\boldsymbol{X}_{new}$ to replace the oldest data points that are classified as sub-dense samples or edge samples and $\boldsymbol{X}_{new}$ also inherits its category. We take Fig. 8 as an example to illustrate the way of data replacement. The advantages of this method are as follows. First, it can significantly save memory usage, especially in the face of large-scale industrial data. Second, using similar substitution can keep the change of data distribution within a certain range as much as possible. Third, This method can keep the data as real-time as possible.

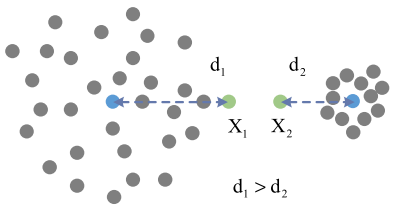


Fig. 6. Distance from point to cluster center.

2.5.2. Real-time detection

Firstly, we need to define the number of clusters and the number of samples within the circle. Next, we apply Algorithm 1 to compute the radius of the current data. Record R to determine if the data category is updated after data filtering. After obtaining the radius, Algorithm 2 is used to classify the original data. The SAS, LAS, and TAS of the original data are calculated, and their clustering centers are recorded to facilitate the subsequent calculation of spatial anomaly scores.

Then, we add the new data to be detected to the original data in a sliding window record it as the current data, and calculate the radius R_{new} of the current data. It will undoubtedly take a lot of time if we reclassify the data and calculate the abnormal score each time the new data point arrives. Therefore, to improve efficiency, we set a threshold T of whether to update the current data category. If the condition of Eq. (12) is met, all saved records are reset. Otherwise, we only calculate the SAS, LAS, and TAS scores of the new data.

Finally, we discriminate the data in the sliding window according to the obtained anomaly score.

$$\frac{R_{new} - R}{R} > T \tag{12}$$

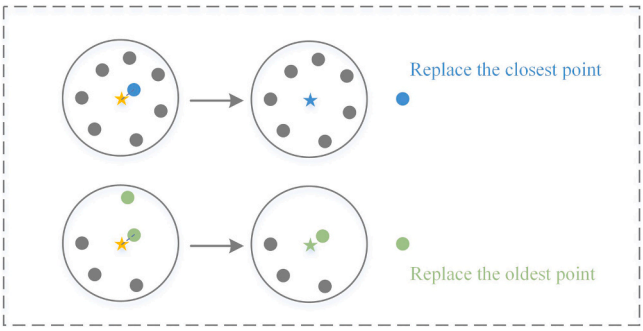


Fig. 8. Data filtering method.

Table 1
Data set information statistics.

Datasets	Instance	Subset	Feature	Anomaly
SMD	1,416,825	28	38	4.2 %
MSL	132,046	27	55	10.53 %
WADI	957,373	1	127	5.84 %
SWAT	944,919	1	51	12.14 %
DWQ	110,815	1	9	1.44 %
NAB	365,558	58	1	0.12 %

3. Experimental results

In this section, the data sets and evaluation metrics employed in the experiment are introduced. A large number of comparative experiments and ablation experiments are carried out with the existing algorithms. Then the experimental results are compared and analyzed to prove the effectiveness of MSAD. The experience is conducted on the i7-6700 Intel Core processor, 32 G RAM, and Microsoft Windows 10 operating system. The model is built on the PyTorch 1.13.1 backend in Python 3.7.4.

3.1. Experimental basis

3.1.1. Datasets

This paper uses six public anomaly detection data sets from five fields to evaluate the model proposed in this paper, as follows.

(1) SMD[1]: Server Machine Dataset is a public data set for multivariate time series anomaly detection. The data set contains 28 servers, each server has 38 features, and the sampling interval is 1 minute. These data have no or little concept drift during data collection [23].

(2) MSL[2]: Mars Science Laboratory is a public dataset for multivariate time series anomaly detection. Gathered by NASA's Curiosity Mars rover as it explores the surface of Mars, it contains a variety of information about the Martian surface and atmospheric environment. The data contains 27 subsets, each entity consists of 55 features [24].

(3) WADI[3]: Water Distribution is a public dataset for multivariate time series anomaly detection. It is collected and published by iTrust, the Cybersecurity Research Center of the University of Technology and Design Singapore, which contains various information from water treatment plants. WADI contains 14 days of normal operation data and 2 days of attack data, with 127 features and a sampling interval of 1 s [25].

(4) SWAT[4]: Secure Water Treatment is a public dataset for multivariate time series anomaly detection. Like the WADI data set, it contains all kinds of information about the water treatment plant. The data set contains 7 days of normal operation data and 4 days of attack data, with 51 features and a sampling interval of 1 s [25].

(5) DWQ[5]: Drinking-water Quality is a public data set for multivariate time series anomaly detection, which is provided by GECCO 2017 Industrial Challenge. It contains a variety of indicators to measure water quality. The data set has 9 features, and the sampling interval is 1 minute.

(6) NAB[6]: The NAB dataset is a new benchmark for evaluating real-time anomaly detection algorithms for streaming data. It is open-sourced by Numenta and contains more than 50 labeled real-world and artificial time series data files [26].

In order to present the key information of each data set more clearly, we summarize the detailed information of these six data sets through Table 1. Because this experiment is an unsupervised anomaly detection experiment, only the data part containing abnormal samples is used.

3.1.2. Evaluation Indicators

Precision, recall, and F1-Score are the most widely used evaluation indexes in anomaly detection research. In addition, in order to evaluate the performance of the algorithm more comprehensively, we further introduce accuracy and AUC values as additional evaluation criteria. These indicators cover different aspects of performance and are designed to provide a comprehensive performance analysis.

- Precision (P): It measures the accuracy of the model in predicting anomalies. In anomaly detection, accuracy is crucial to avoid false negatives.
- Recall (R): It measures the accuracy of the model in predicting a positive class. In anomaly detection, the magnitude of the recall measures the number of false positives classes.
- F1-Score (F1): It combines precision and recall and provides a comprehensive performance metric. It is the harmonic mean of precision and recall.
- Accuracy (ACC): It refers to the proportion of the number of correctly classified samples in the detection process of the anomaly detection algorithm to the total number of samples.
- AUC: It measures the performance of the model under different anomaly detection thresholds. It is the area under the ROC curve and is usually used to measure the performance of the anomaly detection model in various situations.

3.1.3. Baseline methods

We comprehensively compare and synthesize the proposed anomaly detection method, MSAD, with ten existing detection methods. These ten methods include traditional anomaly detection methods and anomaly detection methods based on deep learning. In traditional anomaly detection methods, LOF [27] is a method based on density estimation, which detects outliers by calculating the density difference between objects and neighbors. It is relatively less sensitive to noise and outliers but may be affected by high-dimensional data. The IForest [28] method is a tree-based method that uses random forests to estimate the local outlier factors of data points. It has good interpretability and stability but may be less efficient when dealing with large-scale data. The OCSVM [29] method is based on a support vector machine, which transforms the anomaly detection problem into a binary classification problem and has a strong processing ability for high-dimensional data. Among the anomaly detection methods of deep learning, VAE [30], LSTM-VAE [31], WGAN [32], EBGAD[33], OmniAnomaly[34], and USAD [10] are anomaly detection methods based on reconstruction. They judge anomalies by comparing the differences between the generated data and the original data. The DAGMM [11] method is based on density estimation. It uses the Gaussian mixture model to model the

[1] https://github.com/NetManAIOps/OmniAnomaly
[2] https://s3-us-west-2.amazonaws.com/telemanom/data.zip.
[3] https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#wadi.
[4] https://github.com/JulienAu/Anomaly_Detection_Tuto.
[5] https://www.spotseven.de/wp-content/uploads/2017/08/GeccoIC2017TestData.zip
[6] https://github.com/numenta/NAB/tree/master

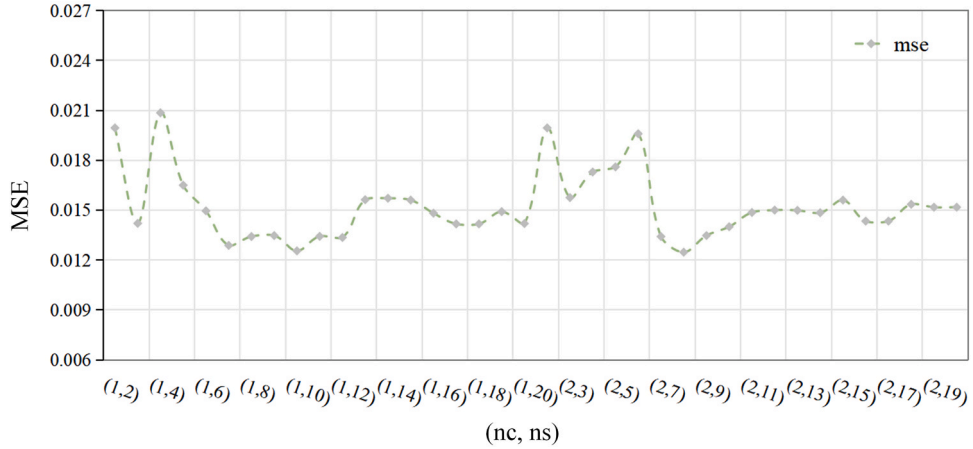


Fig. 9. Linear regression prediction accuracy of dense samples.

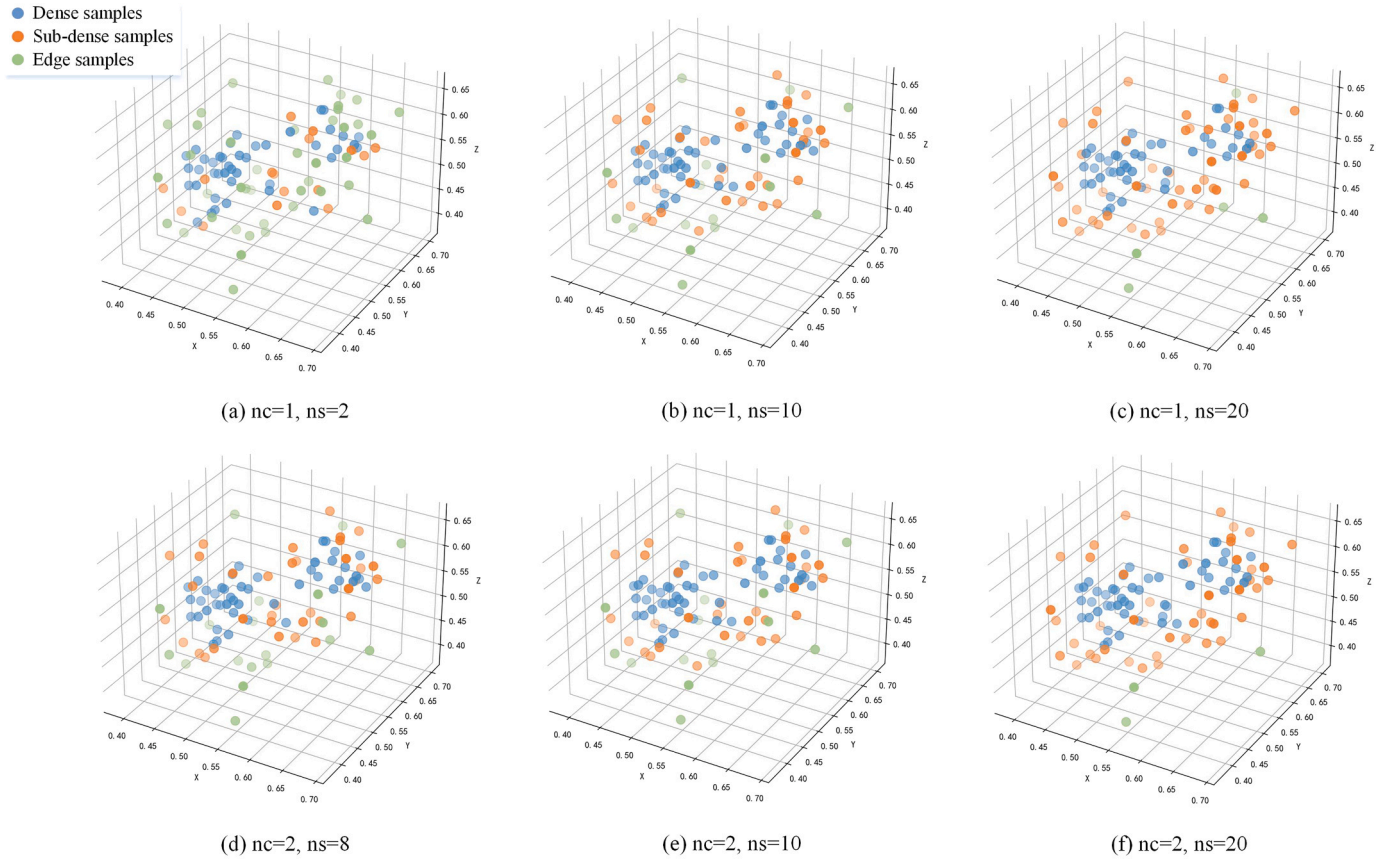


Fig. 10. Classification results under different hyperparameters.

probability density of data and has a strong processing ability for high-dimensional data. Through a comprehensive comparison of these methods, we can more fully understand the performance and superiority of MSAD in the field of anomaly detection.

3.2. Hyperparameter settings

MSAD has two parameters that need to be specified in advance, which are the number of clusters for data clustering and the minimum number of samples required in the circle with R as the radius. In order to find the relatively optimal parameters, we use grid search and cross-validation to traverse the search. The specific method is to use the pre-specified parameters to classify the data, and then use linear regression to predict the samples belonging to ϑ_{ds} and determine the value of the parameters according to the accuracy of the prediction results. Fig. 9 illustrates the prediction error of linear regression model prediction under different parameter settings.

From the Fig. 9, we can see that the error curve fluctuates within a certain range. When the abscissa is (1, 10) or (2, 8), the error value is relatively small. This provides us with a certain reference value. When $(nc,\ ns) = (2,\ 8)$, the prediction accuracy of the linear regression model reaches the highest. Therefore, we can use this parameter as a reference, and select $(nc,\ ns) = (2,\ 8)$ in this data.

In order to show the effect of data category division more intuitively,

we take Fig. 10 as an example to show the effect of data category division under different hyperparameter settings. In Fig. 10 (a), when $(nc, ns) = (1, 2)$, it is obvious that most of the data is incorrectly divided into edge samples. In Fig. 10 (c) (f), due to the large value of ns, the calculation result of R is too large, so most of the data are divided into dense and sub-dense samples. In contrast, in Fig. 10 (b) (d) (e), the classification of data is more reasonable. Therefore, we can consider these values as the basis for initializing the two parameters.

In the reconstruction part of MSAD, we used three fully connected layers to construct the encoder and decoder, with the encoder's output dimension set to 2. The encoder and decoder together form the generator. Similarly, we used three fully connected layers to build the discriminator and classifier. We selected Adam as the optimizer, setting the learning rate to 0.001 for both the classifier and discriminator, and 0.0001 for the generator. The batch size was set to 64. We implemented the VAE, LSTM-VAE, and WGAN detection methods using a three-layer neural network architecture, with all three methods employing the Adam optimizer. For other comparison methods, we referenced and used open source code from GitHub.

3.3. Comparative experiments and analysis

We first use 10 basic comparison methods to compare and evaluate the proposed algorithm on five real-world multivariate time series datasets. We compare the P, R, and F1 of different algorithms on each data set, test the possible abnormal thresholds of each model, and report the results with the highest F1-Score. We use bold to represent the best score in each evaluation indicator and use underscores to represent the second-best score. The results are shown in Tables 2 and 3.

Tables 2 and 3 report the precision, recall, and F1-Score of MSAD and other comparison methods on five datasets. As shown in the table, the performance of MSAD is superior to all comparison methods. Specifically, the F1-Score of MSAD on the SMD dataset is 0.9764, which is higher than other methods. At the same time, the precision of MSAD is as high as 0.9784, and the recall is 0.9748. Compared with other methods, MSAD has a significant performance improvement, especially the traditional anomaly detection methods LOF, IForest, and OCSVM. This simple detection method is not effective on high-dimensional and complexly distributed data. In the test results, although the precision of IForest reached 0.9776, its recall is only 0.8035. This shows that it has high precision in detecting outliers, but there are more false positives. This phenomenon of low recall is also reflected in other comparison methods. The effect of the detection method based on deep learning is relatively improved, and the detection effect of OmniAnomaly is close to that of MSAD. It has a precision of 0.9604 and it has the second-best recall and F1-Score.

The anomaly rate of the MSL data set is 10.53 %, and there are 55 feature numbers. Compared with the SMD data, it is more difficult to detect anomalies, which is well reflected in the detection results. In the detection results of the MSL dataset, the F1-Score of MSAD is 0.9337, which is higher than that of other methods, and the second-best F1-Score is 0.8859, which is larger than that of SMD. The detection results of LOF, IForest, OCSVM, and VAE are relatively poor, especially in the recall.

The anomaly rate of the DWQ data set is 1.44 %, its feature number is only 9, and its feature change range is relatively small. Compared with SMD and MSL, anomaly detection is simpler, which is also well reflected in the detection results. The traditional detection models such as LOF, IForest, and OCSVM have a recall of about 0.9 while the precision rate is about 0.99, which achieves better detection results than other data sets. MSAD also achieves the best results on the DWQ dataset, with an F1-Score of 0.9932, which is higher than other anomaly detection methods, but the precision and recall are the second-best results, and the difference in detection performance is smaller than that of SMD and MSL.The difference in detection results between the MSAD and OmniAnomaly methods is relatively small. To verify whether this difference has statistical significance, we conducted multiple experiments with the OmniAnomaly method on the DWQ dataset using the same threshold to assess the stability of its detection performance. We calculated the mean, standard error, and 95 % confidence interval of the F1 scores for the OmniAnomaly method. The results show a sample mean of 0.9918, a standard error of 0.00029, and a 95 % confidence interval of [0.9912, 0.9923]. These results indicate that the F1 score of the OmniAnomaly method is highly stable. Given the narrow confidence interval and small standard error, we can reasonably rule out the possibility that the performance differences are due to random factors.

The SWAT dataset has the highest anomaly rate of 12.14 %, with a feature number of 51. The anomaly rate of the WADI dataset is 5.84 %, and its feature number is as high as 127. These two data come from the same organization and are high-dimensional time series data in the industry. Their anomaly detection is difficult, and traditional anomaly detection methods perform poorly on these two data sets. IForest achieves optimal precision on both datasets but has a low recall of 0.8074 and 0.7426, respectively, which leads to its relatively low F1-Score, a situation that is also reflected in other conventional detection methods.

Table 3
Performance comparison of MSAD anomaly detection algorithm on different data sets.

Dataset	SWAT			WADI		
Model	P	R	F1	P	R	F1
LOF	90.25	73.97	81.30	94.22	78.05	85.38
IForest	95.84	80.74	87.64	**97.11**	74.26	84.16
OCSVM	93.93	91.96	92.93	96.79	78.13	86.47
VAE	95.66	90.38	92.94	96.21	85.83	90.73
LSTM- VAE	95.58	92.49	94.01	96.85	87.43	91.90
WGAN	95.50	93.50	94.49	95.70	95.40	95.55
EGBAD	86.96	85.85	86.40	93.63	83.87	88.48
DAGMM	95.59	92.49	94.02	95.60	92.44	93.99
OmniAnomaly	**96.06**	97.89	96.97	96.20	95.44	95.82
USAD	93.18	98.25	95.65	92.96	90.55	91.74
MSAD	95.32	**99.43**	**97.39**	97.05	**96.75**	**96.90**

Table 2
Performance comparison of MSAD anomaly detection algorithm on different data sets.

Dataset	SMD			MSL			DWQ		
Model	P	R	F1	P	R	F1	P	R	F1
LOF	91.57	74.47	81.76	90.98	70.99	79.75	98.62	88.17	93.10
IForest	97.76	80.35	88.04	90.89	69.24	78.60	**99.54**	88.99	93.97
OCSVM	95.63	77.82	85.75	**91.32**	74.72	82.19	98.97	88.48	93.43
VAE	93.11	82.68	87.54	89.81	78.30	83.66	98.81	97.37	98.08
LSTM- VAE	95.48	84.18	89.45	89.64	86.16	87.87	98.76	98.33	98.55
WGAN	95.03	87.99	91.36	89.59	81.07	85.12	98.40	96.97	97.68
EGBAD	94.26	90.06	92.06	89.19	79.16	83.88	98.36	95.15	96.72
DAGMM	94.30	91.60	92.92	89.82	89.80	87.50	98.62	**99.89**	99.25
OmniAnomaly	96.04	96.15	96.06	89.56	87.63	88.59	99.03	99.56	99.29
USAD	88.85	77.38	82.40	88.54	84.53	86.48	98.02	98.16	98.09
MSAD	**97.84**	**97.48**	**97.64**	90.65	**96.26**	**93.37**	99.14	99.50	**99.32**

Table 4
Performance comparison of MSAD anomaly detection algorithm on different data sets.

Dataset	SMD		MSL		DWQ		SWAT		WADI	
Model	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
VAE	80.12	76.97	72.63	48.62	96.26	69.69	87.95	86.29	83.48	69.50
LSTM- VAE	82.36	83.91	78.71	54.04	97.14	69.57	89.64	84.63	85.49	80.49
DAGMM	87.47	79.72	78.02	52.13	98.52	61.58	67.63	32.59	88.87	65.25
USAD	70.42	43.26	76.48	42.83	96.27	57.12	92.21	79.67	84.77	32.24
OmniAnomaly	92.96	87.68	79.80	50.73	98.61	81.30	94.62	**88.54**	92.17	80.37
MSAD	**95.71**	**95.83**	**87.77**	**64.34**	**98.66**	**93.68**	**95.32**	86.91	**94.18**	**80.87**

The detection effect of VAE, LSTM-VAE and EGBAD is relatively improved, but it is still not ideal. WGAN, DAGMM, OmniAnomaly and USAD all had relatively good results, achieving precision, recall, and F1-Score above 0.9 on both datasets. The F1-Score of MSAD on these two datasets is optimal, which are 0.9739 and 0.9690, respectively. MSAD has a recall of 0.9943 in the SWAT data set, which is the highest value in all data set detection results, indicating that it is very effective in dealing with false positives.

Combined with the above analysis, compared with the existing 10 detection methods, MSAD shows superior comprehensive performance on 5 test data sets. Whether in the case of different abnormal rates or on high-feature datasets, MSAD shows better results. In particular, MSAD can maintain a high recall rate while maintaining a high precision, which indicates that it can not only accurately detect abnormal samples, but also avoid false positives as much as possible.

In addition, to more accurately measure the performance between different methods, we chose the accuracy and AUC value to evaluate the anomaly detection method. Keep the F1-Score unchanged, the evaluation results are shown in Table 4:

According to the experimental results table, we can see that the performance of MSAD is better than other comparison methods in all indicators. Specifically, MSAD maintains the optimal value in terms of accuracy and AUC value, showing excellent anomaly detection ability. In contrast, other methods have certain differences in accuracy and AUC values. Some methods perform well on some indicators, but they are not always optimal. Through the above analysis, we can conclude that the MSAD method has significant advantages in high-dimensional time series anomaly detection tasks, and it performs the most stable and excellent among all comparison methods.

When dealing with deep learning-based methods, the complex internal structures often require substantial computation time. Therefore, in addition to classic performance analysis, it is essential to conduct a time complexity analysis to evaluate these methods' performance [35, 36]. Therefore, in addition to the classic performance analysis, we also conducted a time performance analysis, as shown in Fig. 11. Here, we report the average training time of the comparison methods on the SMD dataset. The training times for WGAN and EGBAD are relatively short, at 9.13 s and 10.33 s, respectively. In contrast, DAGMM and OmniAnomaly have the longest training times due to their complex internal structures, at 145.33 seconds and 337.31 s, respectively. MSAD performs exceptionally well in terms of training time, taking only 37.62 s, making it more efficient than LSTM-VAE (39.67 s) and USAD (78.18 s). This improvement is especially notable compared to OmniAnomaly (337.31 s) and DAGMM (145.33 s), highlighting a significant enhancement in time performance. Overall, MSAD maintains a high anomaly detection accuracy while also demonstrating a significant advantage in terms of time complexity.

3.4. *Visualization of abnormal ratings*

In order to better understand the performance of different algorithms, we visualize their normalized anomaly scores on machine-1–1 in the SMD dataset, and the results are shown in Fig. 12

From the figure, it can be seen that the reconstruction-based LSTM-VAE, WGAN, and MSAD have some similarities in the distribution of anomaly scores. However, compared with MSAD, LSTM-VAE, WGAN, and DAGMM have more false negatives and false positives in anomaly detection, resulting in lower precision or recall. Although they detected many anomalies, there are still some unrecognizable false negatives, which affected their overall performance in terms of F1-Score. On the contrary, our proposed MSAD method has a higher discriminative ability in anomaly detection, which means that MSAD can identify anomalies more accurately and achieve higher precision and recall than other methods.

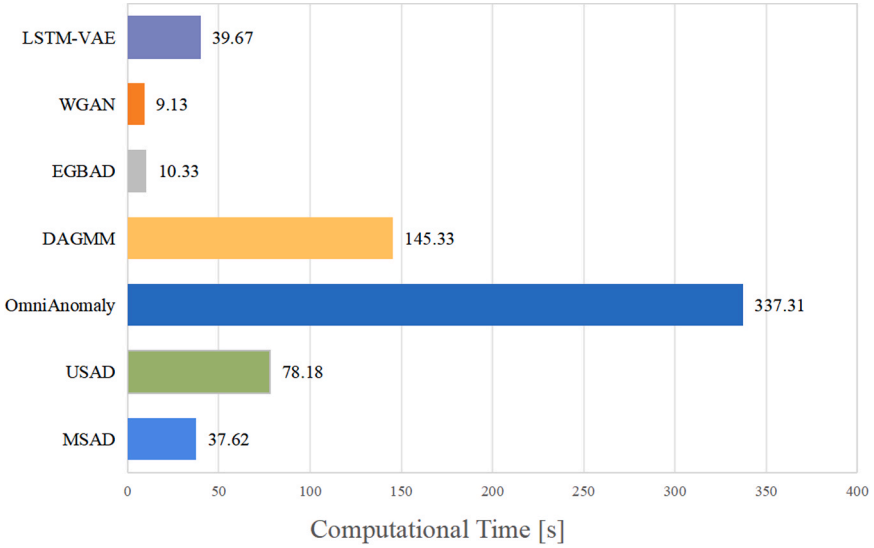


Fig. 11. The training time of each detection method.

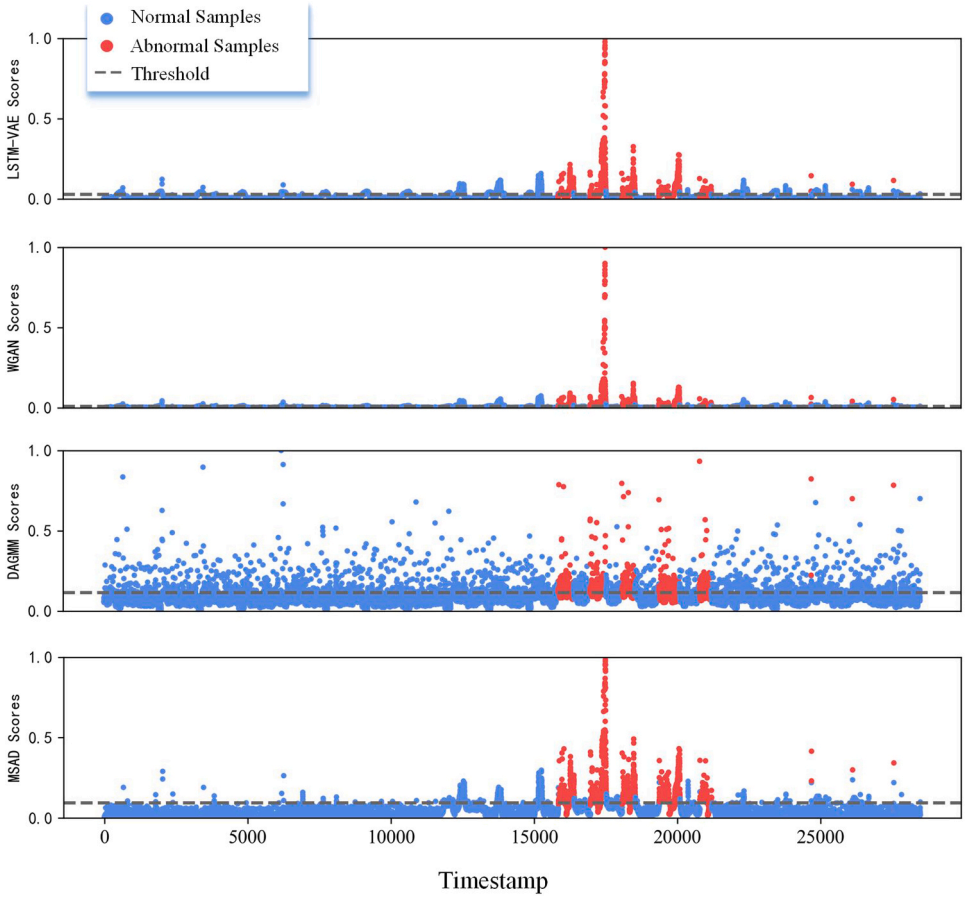


Fig. 12. Visualization of abnormal scores of different algorithms.

3.5. Recall comparison experiment

In this section, we investigate the magnitude of recall of anomaly detection methods under different thresholds. In the experiment, we use the percentage method to determine the threshold, starting from the 100 % digits of the abnormal score array, and each 5 % reduction is detected. In this way, we can observe the performance of different algorithms under different thresholds and compare their recall.

The results are shown in Fig. 13. As the number of samples detected as abnormal increases from 5 % to 25 %, MSAD 's recall is in a leading position most of the time, especially before the 85 % threshold, MSAD maintains a leading position on all five datasets. On the two data sets of SMD and MSL, the recall of MSAD has always maintained a greater advantage than other methods. On the DWQ dataset, because of its small anomaly rate, the recall of the four detection methods is almost the same, but MSAD is still in the leading position. On the SWAT dataset, the recall of MSAD has been in the leading position. On the WADI dataset, MSAD maintains a leading position before the 85 % threshold, while after the 85 % threshold, the recall of LSTM-VAE is slightly ahead of MSAD. According to the above experimental results, the MSAD algorithm has a strong ability to identify anomalies. When detecting the same amount of data, it can effectively detect more real abnormal samples and has better anti-noise ability. At the same time, MSAD has shown good performance on multiple data sets, strong applicability, and can cope with the challenges of different data sets.

3.6. Ablation study

In this section, we explore the effects of different modules of MSAD. Specifically, we conducted a comparative experiment of five variants of MSAD. These five variants are MSAD1, MSAD2, MSAD3, MSAD4 and MSAD5, respectively. MSAD1 uses all data for model training to evaluate the effectiveness of the data partitioning proposed in this paper. MSAD2 omits the calculation of spatial anomaly score to study the influence of local anomaly score on the overall anomaly detection effect. MSAD3 omits the calculation of local anomaly scores to explore the contribution of spatial anomaly scores in overall anomaly detection. MSAD4 omits the calculation of time anomaly score to study the influence of time anomaly score on the overall anomaly detection effect. MSAD5 completely adopts the anomaly detection method proposed in this paper. We use bold to represent the best scores in each evaluation index, and the experimental results are shown in Table 5.

For the SMD dataset, MSAD5 is the most prominent among all variants, which is due to its comprehensive use of all the anomaly detection criteria proposed in this paper. These standards play a crucial role in the SMD dataset, helping the algorithm to effectively detect abnormal samples from different angles. MSAD2 achieves the best results in the MSL dataset because the abnormal distribution in the MSL is highly similar to the normal data. We use t-SNE to visualize the machine-3–6 and MSL datasets in the SMD dataset, as shown in Fig. 14. From the overall perspective of data distribution, the boundary between normal samples and abnormal samples is very vague, resulting in poor performance of spatial anomaly scores on MSL. In the WADI dataset, MSAD5 still performs best overall.

In general, MSAD5 performs best in the three data sets. This excellent performance once again proves the effectiveness and necessity of the anomaly discrimination criteria proposed in this paper. The performance of MSAD2, MSAD3, and MSAD4 is different, and MSAD4 is relatively general. This is because the three data sets used in the experiment are time series data sets, and time correlation is important in

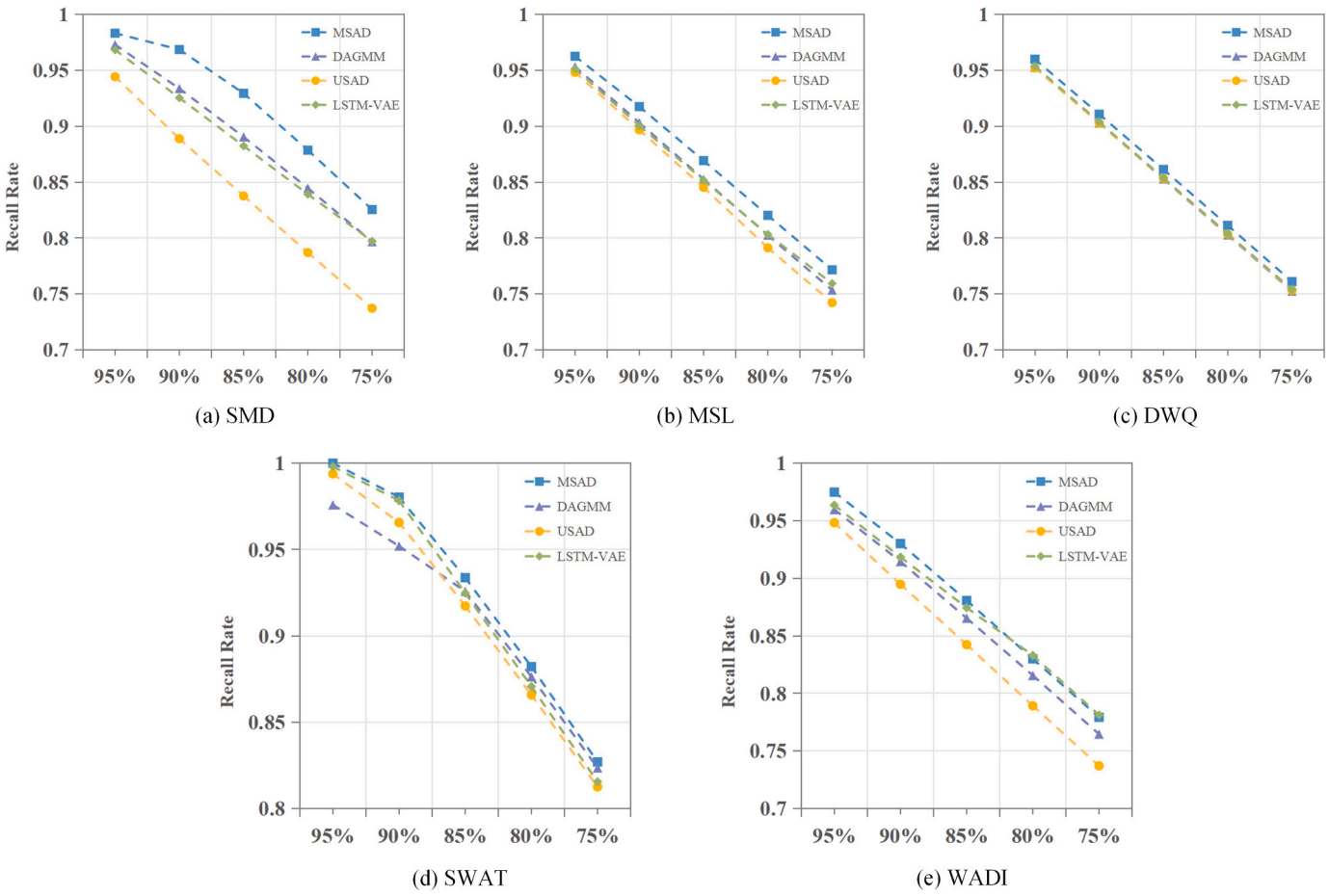


Fig. 13. Comparison of recall of different algorithms.

Table 5
Performance comparison of MSAD ablation experiments.

Dataset	SMD			MSL			WADI		
Model	P	R	F1	P	R	F1	P	R	F1
MSAD1	97.07	94.09	95.51	89.89	95.13	92.28	96.77	96.03	96.40
MSAD2	97.90	96.75	97.30	**89.95**	**95.51**	**92.65**	96.88	96.71	96.79
MSAD3	97.3	95.46	96.30	89.83	95.39	92.53	96.98	95.98	96.48
MSAD4	97.83	94.47	96.03	89.63	95.17	92.32	**97.08**	94.46	95.75
MSAD5	**97.92**	**97.19**	**97.53**	89.87	95.43	92.57	97.05	**96.75**	**96.90**

detecting anomalies. In contrast, the overall performance of MSAD1 was poor. This is because the model is affected by abnormal samples during the training process, resulting in its failure to fully learn the characteristics of normal data. These ablation studies verify that each module in our algorithm is useful and necessary, but for the characteristics of different data sets, it may be necessary to reasonably select different anomaly discrimination criteria.

3.7. Experimental evaluation of clustering methods

We use the K-means method to cluster the data set before adding category labels to the data set. However, due to some known limitations of the K-means clustering algorithm, we selected several alternative clustering methods to investigate their impact on detection performance. The clustering methods include K-means, DBSCAN, hierarchical clustering, and GMM. We conducted experiments while ensuring the same number of clusters for each method, and the results are shown in Table 6. According to Table 6, the results obtained using different clustering methods vary. On the SMD dataset, the GMM clustering method performs the best, with the highest F1 score of 0.9795, which is higher than that of the K-means clustering method. On the MSL and WADI datasets, the K-means clustering method stands out with F1 scores of 0.9337 and 0.9690, respectively, surpassing the performance of other clustering methods. Industrial data often exhibit complex structures and features, so when dealing with different datasets, selecting a clustering method that best suits the characteristics of the dataset itself is advisable.

3.8. Real-time detection

In this section, we carried out real-time detection experiments. We take the first 50 % of the data as the first part of the data, that is, the historical data, and the last 50 % as the second part of the data, that is, the data to be detected, each time with a certain amount of data for detection. Table 7 shows the performance comparison of our proposed method MSAD-RT with other methods. According to the table, MSAD-RT

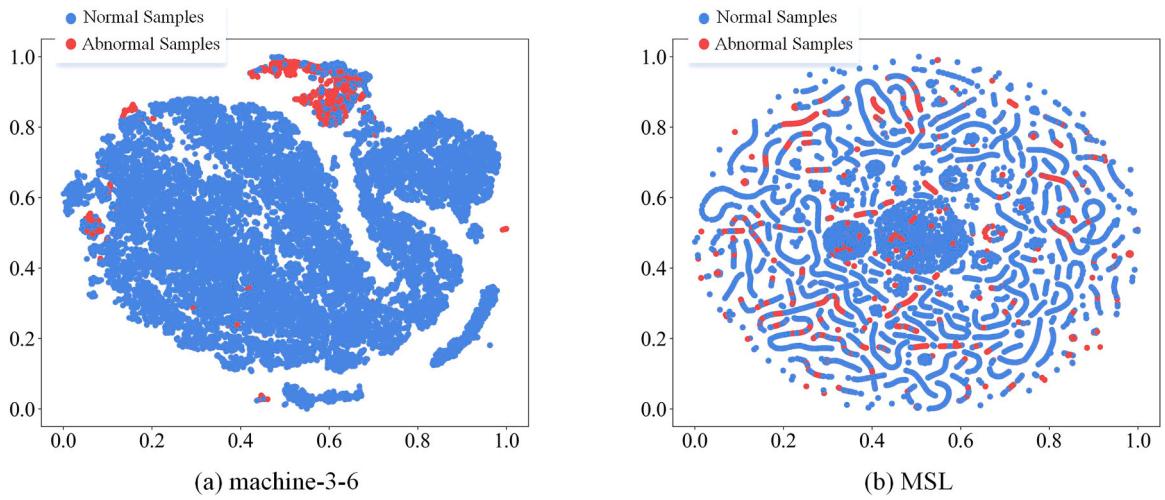


Fig. 14. Machine-3–6 and MSL visualization results.

Table 6
Comparison of anomaly detection results of different clustering schemes.

Dataset	SMD			MSL			WADI		
Model	P	R	F1	P	R	F1	P	R	F1
K-means	97.84	97.48	97.64	**90.65**	**96.26**	**93.37**	**97.05**	**96.75**	**96.90**
DBSCAN	97.41	96.23	96.80	90.45	96.03	93.15	97.01	96.20	96.60
AgglomerativeClustering	**98.17**	97.46	97.79	90.42	96.00	93.13	96.86	96.30	96.58
GMM	97.86	**98.08**	**97.95**	90.21	95.77	92.90	96.91	96.08	96.49

Table 7
Comparison of real-time detection performance.

Dataset	SMD			WADI			NAB		
Model	P	R	F1	P	R	F1	P	R	F1
K-means-RT	94.43	77.07	84.86	95.52	76.09	84.71	99.94	86.78	92.88
DBSCAN-RT	94.31	76.31	84.32	95.51	79.12	86.55	99.94	92.45	95.99
LOF-RT	94.34	74.63	88.23	95.17	74.79	83.76	99.93	81.67	89.39
IForest-RT	94.46	77.78	85.19	95.68	75.19	84.21	99.94	91.04	95.24
OCSVM-RT	94.27	78.62	85.69	95.48	71.13	87.72	99.94	90.16	94.77
MSAD-RT	**95.05**	**89.79**	**92.31**	**95.87**	**91.64**	**93.71**	99.94	**94.93**	**97.35**

performs best among all methods, especially when dealing with high-dimensional industrial data sets SMD and WADI. With close detection precision, MSAD-RT achieved a recall of 0.8979 and an F1-Score as high as 0.9231. Compared with other traditional detection methods, MSAD-RT has achieved significant improvement, especially in the recall rate, thanks to its multiple anomaly discrimination criteria, which can minimize the generation of false alarms and other situations. In the NAB dataset, various detection methods show good performance because of its unidimensional time series and relatively simple data distribution.

Table 8
Comparison of real-time detection performance for the NAB dataset.

Dataset	NAB		
Model	P	R	F1
ARTime	99.87	88.70	93.93
KNN-CAD	99.78	88.70	93.91
numentaTM	99.89	88.48	93.84
contextOSE	**99.99**	88.74	94.03
EXPoSE	99.90	88.71	93.97
Skyline	99.80	88.73	93.94
windowedGaussian	99.86	88.75	93.98
randomCutForest	99.73	88.77	93.93
MSAD-RT	99.94	**94.93**	**97.35**

The DBSCAN-RT performed particularly well, ensuring a high precision of 0.9994 while achieving a recall of 0.9245 and an F1-Score of 0.9599. MSAD-RT continued to outperform the other assays on the NAB dataset with the highest F1-Score of 0.9735.

In order to further evaluate the real-time detection performance of MSAD-RT, we selected and compared the detection results of various other methods on the NAB dataset. Table 8 shows the performance difference between MSAD-RT and other detection methods on the NAB dataset. Among these algorithms, contextOSE tops the list with its F1-Score of 0.9403. Nonetheless, MSAD-RT shows better strength, with a higher F1-Score of 0.0332 compared to contextOSE and a slightly lower precision rate than contextOSE, but MSAD-RT still has a significant lead in recall, with a 6.19 % increase in recall rate compared to it. These results fully demonstrate the excellent performance of MSAD-RT in the field of real-time anomaly detection, especially in reducing the false alarm rate, the method proposed in this paper shows significant advantages.

4. Conclusions

In this paper, we delve into the problem of anomaly detection in multivariate time series industrial data. We propose a novel unsupervised anomaly detection of multivariate time series based on multi-

standard fusion—MSAD. MSAD introduces a new unsupervised anomaly detection architecture that analyzes data anomalies from multiple perspectives. This architecture integrates various anomaly standards, including spatial anomalies, local anomalies, time anomalies, and data reconstruction errors, to more accurately identify anomalous samples in large-scale industrial data. The multiple anomaly standards used in the MSAD method offer valuable data selection references for subsequent model development. These standards assist researchers and engineers in making more targeted decisions when constructing models, thereby simplifying the model-building process and enhancing the model's applicability and effectiveness. In addition, we have designed an industrial data stream anomaly detection method, MSAD-RT. We used six public datasets to validate the effectiveness of the proposed algorithm. The experimental results show that MSAD outperforms existing state-of-the-art methods in terms of accuracy and efficiency in anomaly detection. Furthermore, the experiments demonstrate the accuracy, stability, and applicability of MSAD across various datasets. It effectively reduces the detection time for anomalous samples while maintaining high prediction accuracy. Therefore, the MSAD method provides strong support for anomaly detection in complex industrial data.

However, the MSAD method still has some limitations. In future work, we will conduct more in-depth research to address these shortcomings. Specifically, different datasets have different distribution characteristics, and the importance of the four anomaly detection criteria may vary depending on the dataset. We aim for MSAD to adaptively assign weights to these four criteria based on the characteristics of the dataset to achieve more accurate detection results. Additionally, in the real-time detection part, the number of anomalies varies across different sliding windows. We hope to adaptively select appropriate thresholds for anomaly detection within samples of different sliding windows.

CRediT authorship contribution statement

Kun Li: Writing – original draft, Supervision, Investigation. **Shikang Lu:** Writing – review & editing, Investigation. **Hao Kong:** Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Huixin Tian:** Writing – review & editing, Supervision, Investigation, Formal analysis, Conceptualization.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant number: 62473283), Tianjin Science and Technology Correspondent Project of China (grant number: 19JCTPJC47600).

References

[1] Q.Q. Sun, Z.Q. Ge, A survey on deep learning for data-driven soft sensors, IEEE Trans. Ind. Inform. 17 (2021) 5853–5866.
[2] N. Kashpruk, C. Piskor-Ignatowicz, J. Baranowski, Time series prediction in industry 4.0: a comprehensive review and prospects for future advancements, Appl. Sci. 13 (2023).
[3] M.A. Belay, S.S. Blakseth, A. Rasheed, P.S. Rossi, Unsupervised anomaly detection for IoT-based multivariate time series: existing solutions, performance analysis and future directions, Sensors 23 (2023).
[4] M. Di Mauro, G. Galatro, F. Postiglione, W. Song, A.J.I.To.N. Liotta, S. Management, Multivariate Time Series characterization and forecasting of VoIP traffic in real mobile networks, IEEE Trans. Netw. Serv. Manag. (2023).
[5] N. Zhou, Z. Zheng, J. Zhou, Prediction of the RUL of PEMFC based on multivariate time series forecasting model, in: 2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS), IEEE, 2023, pp. 87–92.
[6] L.M. Zhang, X.W. Xie, K. Xiao, W.J. Bai, K. Liu, P.P. Dong, MANomaly: mutual adversarial networks for semi-supervised anomaly detection, Inf. Sci. 611 (2022) 65–80.
[7] Z. Chen, J. Duan, L. Kang, G.P. Qiu, Supervised anomaly detection via conditional generative adversarial network and ensemble active learning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2023) 7781–7798.
[8] H. Zhou, K. Yu, X. Zhang, G. Wu, A.J.I.S.A.I.J. Yazidi, Contrastive autoencoder for anomaly detection in multivariate time series, (2022).
[9] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, G. Shroff, LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection, (2016).
[10] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M.A. ZuluagaUsad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, 3395-3404. .
[11] S. Qi, Deep Autoencoding Gaussian mixture model for unsupervised anomaly detection, Int. Conf. Learn. Represent. (2018).
[12] X.C. Zhang, J. Mu, X.T. Zhang, H. Liu, L.L. Zong, Y.G. Li, Deep anomaly detection with self-supervised learning and adversarial training, Pattern Recognit. 121 (2022).
[13] X.G. Peng, H.H. Li, F. Yuan, S.G. Razul, Z.B. Chen, Z.P. Lin, An extreme learning machine for unsupervised online anomaly detection in multivariate time series, Neurocomputing 501 (2022) 596–608.
[14] M. Pota, G. De Pietro, M. Esposito, Real-time anomaly detection on time series of industrial furnaces: a comparison of autoencoder architectures, Eng. Appl. Artif. Intell. 124 (2023).
[15] L.Y. Zhang, J.B. Zhao, W. Li, Online and unsupervised anomaly detection for streaming data using an array of sliding windows and PDDs, IEEE Trans. Cybern. 51 (2021) 2284–2289.
[16] B.J. Zou, K.K. Yang, X.Y. Kui, J. Liu, S.H. Liao, W. Zhao, Anomaly detection for streaming data based on grid-clustering and Gaussian distribution, Inf. Sci. 638 (2023).
[17] M. Gong, J. Sun, X. Xie, Y. Zheng, Multivariate time series prediction based on improved transformer model in computing system, in: 2023 2nd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE), IEEE, 2023, pp. 45–50.
[18] G. Douzas, F. Bacao, Effective data generation for imbalanced learning using conditional generative adversarial networks, Expert Syst. Appl. 91 (2018) 464–471.
[19] S.F. Zhang, Z. Qian, K.Z. Huang, R. Zhang, J.M. Xiao, Y. He, C.Y. Lu, Robust generative adversarial network, Mach. Learn. (2023).
[20] Z.H. Zhai, Auto-encoder generative adversarial networks, J. Intell. Fuzzy Syst. 35 (2018) 3043–3049.
[21] C.Y. Wang, C. Xu, X. Yao, D.C. Tao, Evolutionary generative adversarial networks, IEEE Trans. Evolut. Comput. 23 (2019) 921–934.
[22] J.R. Cheng, Y. Yang, X.Y. Tang, N.X. Xiong, Y. Zhang, F.F. Lei, Generative adversarial networks: a literature review, Ksii Trans. Internet Inf. Syst. 14 (2020) 4625–4647.
[23] Y. Su, Y.J. Zhao, C.H. Niu, R. Liu, W. Sun, D. Pei, M. Assoc CompRobust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network, in: 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), Anchorage, AK, 2019, 2828-2837. .
[24] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. SoderstromAcm, Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding, in: 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), London, ENGLAND, 2018, 387-395. .
[25] A.P. Mathur, N.O. Tippenhauer, Ieee, SWat: A Water Treatment Testbed for Research and Training on ICS Security, in: International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater), AUSTRIA, Vienna, 2016, pp. 31–36.
[26] A. Lavin, S. AhmadIeee, Evaluating Real-time Anomaly Detection Algorithms - the Numenta Anomaly Benchmark, in: IEEE 14th International Conference on Machine Learning and Applications ICMLA, Miami, FL, 2015, 38–44.
[27] M.M. Breunig, H.P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, Sigmod Rec. 29 (2000) 93–104.
[28] F.T. Liu, K.M. Ting, Z.H. ZhouIsolation Forest, in: 8th IEEE International Conference on Data Mining, Pisa, ITALY, 2008, pp. 413-+.
[29] D.M.J. Tax, R.P.W. Duin, Support vector data description, Mach. Learn. 54 (2004) 45–66.
[30] D.P. Kingma, M.J.a.p.a. Welling, Auto-encoding variational bayes, (2013).
[31] D. Park, Y. Hoshi, C.C.J.I.R. Kemp, A. Letters, A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder 3 (2018) 1544–1551.
[32] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, (2017).

[33] H. Zenati, C.S. Foo, B. Lecouat, G. Manek, V.R. Chandrasekhar, Efficient gan-based anomaly detection, arXiv preprint arXiv:1802.06222 (2019).
[34] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, Dan Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (2019) 2828–2837.
[35] M. Di Mauro, G. Galatro, F. Postiglione, W. Song, A.J.C.N. Liotta, Hybrid learning strategies for multivariate time series forecasting of network quality metrics, Comput. Netw. (2024) 110286.
[36] Z. Yang, P. Li, Y. Bao, X. Huang, Speeding up multivariate time series segmentation using feature extraction, in: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), IEEE, 2020, pp. 954–957.

Huixin Tian received her Ph.D. degree in Control Theory and Control Engineering from Northeastern University in 2009. She joined the School of Electrical Engineering and Automation at Tianjin Polytechnic University in 2009 and is currently a professor at the School of Control Science and Engineering. Her research interests include artificial intelligence and big data analysis, industrial intelligence, and smart factories.

Hao Kong received his B.S. degree in Computer Science and Technology from Shandong Agricultural University in 2021. He is currently pursuing his M.S. degree in Artificial Intelligence at Tianjin Polytechnic University. His main research directions are machine learning and anomaly detection.

Shikang Lu received his B.S. degree in Automation from Shandong Agricultural University in 2023. He is currently pursuing his M.S. degree in Control Science and Engineering at Tianjin Polytechnic University. His main research interests are machine learning and anomaly detection.

Kun Li received his Ph.D. degree in Systems Engineering from Northeastern University in 2010. He joined the School of Economics and Management at Tianjin Polytechnic University in 2010, where he is currently a professor.