Citation: Aijun Yan, Zijun Cheng. Anomaly detection method based on Siamese attention representation network. *Journal of Harbin Institute of Technology(New Series)*. DOI:10.11916/j.issn.1005-9113.25021

Anomaly Detection Method Based on Siamese Attention Representation Network

Aijun Yan [1, 2, 3 *] *and Zijun Cheng* [1, 2]

(*1. School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China;*
2. Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, China;
3. Beijing Laboratory for Urban Mass Transit, Beijing 100124, China)

Abstract: To solve the limitations of the unsupervised anomaly detection method, which cannot utilize the prior knowledge of anomalies, relies heavily on the anomaly distribution hypothesis, and is insensitive to edge anomalies near normal data, a contrastive learning method based on Siamese attention representation network is proposed to realize anomaly detection. In the method, two encoder networks constructed using multi-head attention learn different representations of the same unlabeled data, and a self-supervised representation learning model is implemented from one encoder network to another encoder network using contrast learning structure. Then, a classifier network is added to realize the detection of abnormal data through weakly supervised classification learning based on a small amount of labeled data. The experimental results show that this method can effectively improve the recall rate of anomaly detection and can be applied in the field of data analysis.

Keywords: anomaly detection, Siamese network, multi-head attention, representation learning, contrastive self-supervised learning, weakly supervised learning

CLC number: TP183 **Document code**: A **Article ID**: 1005-9113(2025)00-0000-16

0 Introduction

Anomaly (outlier) detection is the process of finding data instances that have significant deviations from most data instances. As one of the indispensable key technologies in the field of artificial intelligence (AI), such as data mining, machine learning, and computer vision, anomaly detection is often used in the pre-processing stage of data analysis, and plays an increasingly important role in practical applications, such as industrial monitoring[1], telecommunication fraud[2], health care[3], and network security[4]. Due to the scarcity of abnormal data, the complexity of abnormal situations, the imbalance in the number of abnormal data, and the absence of abnormal data labels, it is challenging for supervised or semi-supervised methods to play a role. Most of the research on anomaly detection focuses on unsupervised anomaly detection as indicated in Refs.[5-7]. The classical anomaly detection methods are mainly divided into statistical anomaly detection[8], density-based anomaly detection[9], distance-based anomaly detection[10], and clustering-based anomaly detection[11]. These methods are usually limited by the data itself, or cannot be applied to high-dimensional data, or are not suitable for data sets with unknown distribution, or have high computational costs, or exhibit low processing efficiency of large data sets[12]. In particular, it is insensitive to some outliers that are very close to normal data, which often leads to misjudgment of data and a low recall rate of anomaly detection.

In recent years, deep learning (DL) has shown strong learning ability in the representation learning of complex data, such as high-dimensional data, time series data, spatial data, and graph data, which has attracted much attention in the research of deep anomaly detection[13]. It uses a deep neural network to learn the feature representation of different data

Received 2025-04-02.

Sponsored by National Natural Science Foundation of China (Grant Nos.62373017, 62073006), Beijing Natural Science Foundation of China (Grant No.4212032)

* Corresponding author: Aijun Yan, Ph.D, Professor. Email: yanaijun@bjut.edu.cn.

structures, and judges the abnormal situation of data through the abnormal scoring function, which can effectively improve the performance of anomaly detection, such as Autoencoders (AE) or Variational AE (VAE), Generative Adversarial Networks (GAN), self-supervised classification, etc.

Anomaly detection based on AE network[14-15] mainly relies on the encoding and decoding processes to reconstruct the original data, and the data reconstruction error is used as the anomaly score to judge the anomaly. However, the data reconstruction process of encoding and decoding may be affected by abnormal data, resulting in deviation. GAN-based anomaly detection[6, 16] learns the potential feature distribution of normal data by a generative network, and some form of residual between the generated instance and the real instance is defined as an anomaly score to achieve the judgment of abnormal data. However, the training of GAN has the problem of convergence failure or mode collapse, and the abnormal score will be affected by the generator network. self-supervised classification[17] involves building a self-supervised classification model to learn the normal representation of data, which identifies data inconsistent with the classification model as abnormal. However, this method mainly realizes the self-supervised learning process through feature transformation, and its operation is limited to image data.

Since the proposal of contrastive self-supervised learning, with the introduction of algorithms such as MoCo[18], SimCLR[19], BYOL[20], and SimSiam[21], contrastive self-supervised anomaly detection methods based on data such as images and sequences have been gradually proposed[22-24]. These methods realize the representation of data through contrastive learning. The calculation of the abnormal score depends on the designed comparative loss function or is set according to the method of downstream tasks. Because it is mostly aimed at images, sequences, and other types of data, there is a lack of research on the most common table data anomaly detection.

To sum up, in the learning process of deep anomaly detection, the effect of semi-supervised anomaly detection is good, but it often requires a sufficient number of labeled samples; unsupervised anomaly detection is the most commonly used, but the learning process of feature representation is susceptible to abnormal data and cannot make use of the prior knowledge of existing labels. In fact, no matter whether there is abnormal data with labels, it is not difficult to collect and label a small amount of abnormal data, which also makes the weak supervision method more suitable for anomaly detection. However, how to realize the learning process of anomaly detection through a few labeled samples is still a difficult problem for weakly supervised learning. With the proposal and development of self-supervised learning, generative or contrastive self-supervised learning[25] performs well in the field of computer vision because of its deep representation learning without sample labeling. This enables it to be gradually applied in anomaly detection. The self-supervised learning method can learn the feature representation of the data without labeling the sample data, which can effectively avoid the sample label problem. However, the learning process of the generative self-supervised method will also be affected by the abnormal data. The contrast self-supervised learning can shorten the distance between the positive samples and lengthen the distance between the negative samples to better distinguish the edge outliers. Therefore, the self-supervised learning can better adapt to the representation learning of unlabeled data.

From the perspective of spatial distribution, the diversity of abnormal data makes it difficult to cluster them in adjacent areas, while normal data often have similar representations. Based on this point, if the self-supervised learning of two attention contrast structures is used to narrow the region of normal data, so as to expand the distance between abnormal and normal, the distinction between abnormal and normal may be more obvious. It is helpful for weakly supervised anomaly detection learning based on a small amount of labeled sample data. Therefore, this study mainly proposes a data anomaly detection method based on Siamese Attention Representation Network (SiamARN) by combining comparative self-supervised learning and weak supervised learning for table data. The contrastive learning structure based on Siamese network[26] is used to expand the distance between different data to avoid the problem that edge anomalies are difficult to detect. The representation network of Multi-Head Attention (MHA)[27] is added to improve the learning performance of data deep representation to realize the deep representation model

of unlabeled data. A classifier network is added after the representation model to realize anomaly detection through weakly supervised learning of a small number of labeled samples. Finally, experiments show that the method can better realize the detection of abnormal data and improve the recall rate of anomaly detection. The main contributions of this study are as follows:

1) A contrastive self-supervised learning method is proposed to improve the discrimination of data in normal and abnormal ranges, which is helpful to realize the weakly supervised learning of data anomaly classification with a small amount of labelled samples in the later stage.

2) Data augmentation based on data processing is used to obtain different views of the data, and an encoder network with multi-head attention is proposed to help the data get a suitable deep representation, which helps to improve the performance of the model in anomaly detection.

3) An anomaly detection method combining self-supervision and weak supervision is proposed, and a series of experiments are designed to prove the effectiveness and superiority of the method.

The rest of this study is organized as follows: Section 1 is a brief description of the relevant work research. Section 2 introduces the principle and framework of the method. Section 3 shows the experimental details and results analysis. Section 4 summarizes the full text.

1 Contrastive Learning

The Siamese network is a connected network model for data comparison, which realizes the comparison and judgment between data through two or more input shared weight neural networks. As a supervised learning algorithm, it is often used in face verification[24], target tracking[28], one-time learning[29], and so on. In recent years, Siamese network has begun to be applied to the research of self-supervised learning. By comparing the loss function, the network parameters are learned to realize the deep representation of complex data such as images and sequences.

Contrastive learning is a discriminative representation learning method based on the contrastive thought. The main idea is to make positive samples close to each other and negative samples far away from each other. This method has been widely used in self-supervised learning, called contrastive self-supervised learning. Among them, the Siamese network structure is usually used as a carrier for contrastive learning, such as MoCo and SimCLR, which realize image representation learning through positive and negative sample contrast. With the proposed algorithms such as BYOL and SimSiam, the comparative learning method gets rid of the model's requirements for negative samples and realizes its own contrastive learning under different data augmentation. In particular, SimSiam[21] proposed that the stop-gradient operation could prevent model collapse. These algorithms are mainly used in computer vision, natural language processing, and so on. However, these data augmentation methods are often applicable to images, sequences, and other data to obtain different positive and negative (or positive) samples to achieve deep representation learning.

Here, we mainly focus on anomaly detection of tabular data, and propose different data processing methods as augmentation methods to obtain different views of data, and distinguish normal data from abnormal data through the attention contrast learning process. The collapse of the model is prevented by batch normalization (BN) and stop-gradient (stopgrad) operation.

2 Method

Aiming at the problems of insensitivity to edge anomalies and waste of prior knowledge of anomalies in anomaly detection algorithms, an anomaly detection method called SiamARN is proposed. This section mainly introduces the overall framework, the implementation process and the pseudo code of the SiamARN.

2.1 SiamARN Framework

The structure of SiamARN is shown in Fig. 1, which mainly includes two parts: data representation learning process and classification learning process. Among them, the representation learning process includes data augmentation processing, encoder network and predictor network. Classification learning is realized by pre-training encoder model and classifier network.

Firstly, data augmentation is used to obtain different views of the input x, namely, view 1: x_1 and view 2: x_2, as two different inputs for the

subsequent contrastive structure.

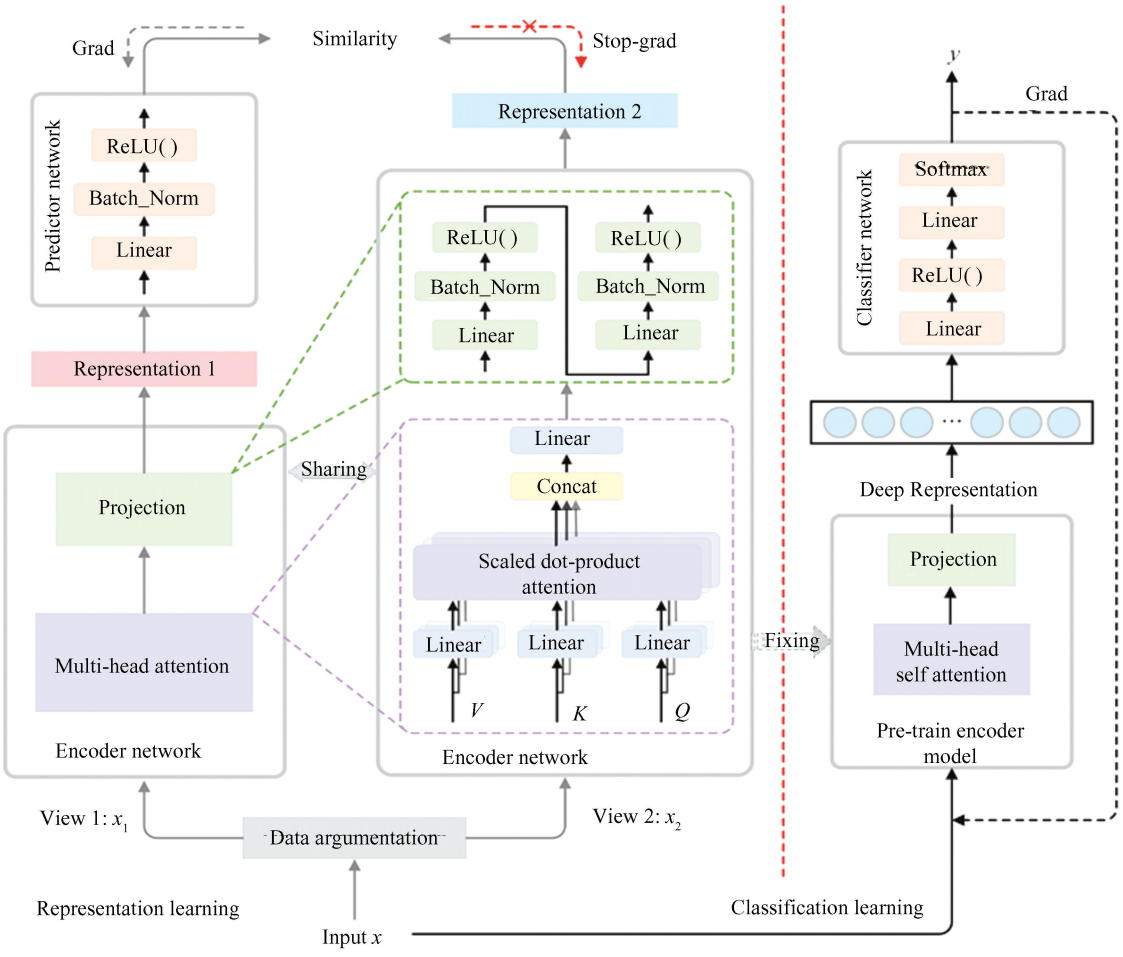


Fig.1 The structure of SiamARN

Secondly, the encoder network is composed of Multi-Head Attention (MHA) module and projection module. In the MHA module, K, Q, and V represent key, query, and value, respectively. The projection module contains two layers of the same fully connected layer, Batch Normalization (BN) layer and ReLU layer. The encoder network is mainly used to learn the deep representation of tabular data. The weights are shared between the encoder to obtain two different vector representations, representation 1 and representation 2, and fix the network to send it into the representation learning. Then, the predictor network is composed of a set of fully connected layers, BN layers, and ReLU layers, which is mainly used to realize the learning from representation 1 to representation 2. The similarity is used to judge the error between the two representations and the stop gradient is added to prevent the model from collapsing, where grad denotes gradient, stop-grad denotes stop gradient.

Finally, the classifier network is a two-layer fully connected network, which is connected to the pre-trained encoder model obtained by representation learning. Its function is to classify and judge the deep representation of the output of the encoder network. By fine-tuning the network parameters of classifier and encoder with a few samples, weakly supervised learning of anomaly detection under deep representation is realized.

2.2 Algorithm Implementation

2.2.1 Simple Siamese network

The representation learning process of data is a simple Siamese network learning process, and its simplified structure is shown in Fig. 2. The encoder network f shares weights, and the predictor network h matches the output of one view with the output of another view.

The function representation corresponding to the

encoder network f and the predictor network h is shown in Eqs. (1)-(2).

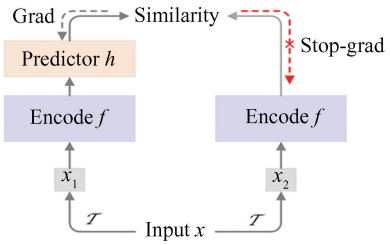


Fig.2 Simple Siamese network structure

$$\boldsymbol{e} = f(\boldsymbol{x}) \tag{1}$$

$$\boldsymbol{p} = h(f(\boldsymbol{x})) \tag{2}$$

The data $\boldsymbol{x} \in \mathbb{R}^n$ is represented by two different views x_1, x_2 through the data augmentation process T. Through the encoder network and the predictor network in turn, the vector $\boldsymbol{p}_1$ and vector $\boldsymbol{e}_2$ are obtained, respectively. The negative cosine metric function is used to calculate the similarity and minimize it, as shown in Eq. (3):

$$\mathrm{Sim}(\boldsymbol{p}_1, \boldsymbol{e}_2) = -\frac{\boldsymbol{p}_1}{\|\boldsymbol{p}_1\|_2} \cdot \frac{\boldsymbol{e}_2}{\|\boldsymbol{e}_2\|_2} \tag{3}$$

where $\|\cdot\|_2$ is L_2-norm. Further define a symmetrized loss function with a lower bound of -1:

$$\mathcal{L} = \frac{1}{2}\mathrm{Sim}(\boldsymbol{p}_1, \boldsymbol{e}_2) + \frac{1}{2}\mathrm{Sim}(\boldsymbol{p}_2, \boldsymbol{e}_1) \tag{4}$$

To prevent the model from collapsing, a stop-gradient is added to $\mathcal{L}$ so that $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are regarded as constants to obtain a new loss function:

$$\mathcal{L} = \frac{1}{2}\mathrm{Sim}(\boldsymbol{p}_1, \mathrm{stopgrad}(\boldsymbol{e}_2)) + \frac{1}{2}\mathrm{Sim}(\boldsymbol{p}_2, \mathrm{stopgrad}(\boldsymbol{e}_1)) \tag{5}$$

To explain the effect of stop-gradient and predictor network in the model, it is described from the perspective of expectation maximization (EM). The process involves two sets of variables, and the stop-gradient operation introduces an additional variable φ. Rewrite the loss function as:

$$\mathcal{L}(\theta, \gamma) = \mathbb{E}_{x,\mathcal{T}}[\|\Phi_\theta(\mathcal{T}(\boldsymbol{x})) - \varphi_x\|_2^2] \tag{6}$$

where Φ represents the network parameterized by θ, $\mathbb{E}[\cdot]$ represents the distribution of $\boldsymbol{x}$ and φ. It is worth noting that φ is not necessarily the output of the network. The optimization goal is to find θ and φ to minimize the expectation of the loss function, that is:

$$\min_{\theta,\gamma} \mathcal{L}(\theta, \varphi) \tag{7}$$

Based on the above expression, Eq. (6) can be solved by an alternately optimized, that is, fixing one set of parameters and optimizing another set of parameters, as shown below:

$$\theta^t \leftarrow \underset{\theta}{\mathrm{argmin}}\ \mathcal{L}(\theta, \varphi^{t-1}) \tag{8}$$

$$\varphi^t \leftarrow \underset{\varphi}{\mathrm{argmin}}\ \mathcal{L}(\theta^t, \varphi) \tag{9}$$

where t represents the index of the alternating process, $\leftarrow$ represents the allocation.

For the solution of θ: the stochastic gradient descent (SGD) optimizer is used to solve Formula (8). Stop-gradient operation is a natural result because φ^{t-1} is a constant when the gradient does not propagate back.

For the solution of φ: Formula (9) can be solved independently for each φ_x. As shown in Eq. (6), the expectation is the distribution of augmentation $\mathcal{T}$, φ_x can be derived by Formula (10). It shows that φ_x is assigned to the average representation of the augmentation distribution of $\boldsymbol{x}$.

$$\varphi_x^t \leftarrow \mathbb{E}_{\mathcal{T}}[\Phi_\theta(\mathcal{T}(\boldsymbol{x}))] \tag{10}$$

The Siamese network model is alternately approximated by Formulas (8) and (10). In this process, an augmentation method is first selected, and the augmentation $\mathcal{T}$ is only sampled once, so that $\mathbb{E}[\cdot]$ is ignored, and the Formula (11) is obtained.

$$\varphi_x^t \leftarrow \Phi_{\theta^t}(\mathcal{T}'(\boldsymbol{x})) \tag{11}$$

Then, substituting Formula (11) into Formula (8), we get Formula (12):

$$\theta^{t+1} \leftarrow \underset{\theta}{\mathrm{argmin}}\ \mathbb{E}_{x,\mathcal{T}}[\|\Phi_\theta(\mathcal{T}(\boldsymbol{x})) - \Phi_{\theta^t}(\mathcal{T}'(\boldsymbol{x}))\|_2^2] \tag{12}$$

where θ^t is a constant. "$\mathcal{T}'$ implies another view due to its random nature. This formulation exhibits the Siamese architecture. If we implement Eq. (12) by reducing the loss with one SGD step, then we can approach the SimSiam algorithm: a Siamese network naturally with stop-gradient applied."

The above process does not involve the role of the predictor network h in the model. As shown in Fig. 2, the predictor network h is used to minimize the output of the encoder network f, which is expressed as follows:

$$\mathbb{E}_p[\|h(\boldsymbol{e}_1) - h(\boldsymbol{e}_2)\|_2^2] \tag{13}$$

The corresponding optimal solution satisfies:

$$h(\boldsymbol{e}_1) = \mathbb{E}_e(\boldsymbol{e}_2) = \mathbb{E}_{\mathcal{T}}[\Phi_\theta(\mathcal{T}(\boldsymbol{x}))] \tag{14}$$

Obviously, Eq. (14) is similar to Formula (10). In Formula (11), since the expectation $\mathbb{E}[\cdot]$ is ignored in one sampling, it can be considered that

the function of predictor h is to fill this gap. Because it is almost impossible to directly calculate the expectation $\mathbb{E}[\cdot]$ of the augmentation $\mathcal{T}$, but it is also a feasible scheme to learn the appropriate parameters by using the prediction network to achieve the prediction expectation. After multiple epochs of training, the sampling of $\mathcal{T}$ is implicitly distributed in the parameters of the predictor.

2.2.2 SiamARN algorithm

1) DA algorithm. It is a technique that generates new samples by making certain changes to the data. DA has been widely used in computer vision and natural language processing, such as geometric transformation, fuzzy processing, random occlusion, and other operations on image data, random insertion, random exchange, random deletion, and other operations on text data. However, for general tabular data, there are few data augmentation methods to change it to obtain different augmented data. In fact, in contrastive learning, one of the purposes of data augmentation is to contrast the representation of data in different views, and to obtain the deep representation of data by the contrastive learning process, which means that in anomaly detection, we need to obtain different representations of data in some way for contrastive learning without changing the original distribution of data. Therefore, this study proposes selecting different data processing methods to obtain representations of tabular data from different perspectives for the contrastive learning process of data. There are three main methods used:

a) Min-Max normalization: $x' = \dfrac{x - x_{\min}}{x_{\max} - x_{\min}}$.

b) Z-score normalization: $x' = \dfrac{x - \bar{x}}{s}$, where $\bar{x}$ is the mean, s is the standard deviation.

c) L_1-norm normalization: $x' = \dfrac{x}{\|x\|_1}$, $\|x\|_1$ is L_1-norm.

2) Encoder network. As shown in Fig. 1, the encoder network is composed of MHA and projection modules. The weights are shared between the two encoder networks. The augmented data learns the data feature relationship through the MHA module, and then connects two groups of fully connected networks. Each group of networks includes a fully connected layer, a BN layer, and a ReLU layer. The encoder network is used to realize the deep representation of data, and the detailed process is as follows.

Suppose that the input dimension of the single-branch encoder network is $(b,1,n)$, where b represents the batch size, n is the feature dimension of data x. MHA is used to learn the representation of $\mathcal{T}(\boldsymbol{x})$, the number of heads is H, and each attention head is calculated by scaling dot product attention:

$$\mathrm{Att}_{H_i} = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}_{H_i}\boldsymbol{K}_{H_i}^{\mathrm{T}}}{\sqrt{d_k/H}}\right)\boldsymbol{V}_{H_i} \tag{15}$$

where H_i represents i-th head, $\boldsymbol{Q}_{H_i} = W_{Q_I}\mathcal{T}(x)$ represents query matrix, $\boldsymbol{K}_{H_i} = W_{K_i}\mathcal{T}(x)$ represents key matrix, $\boldsymbol{V}_{H_i} = W_{V_i}\mathcal{T}(x)$ represents value matrix, d_k is the key dimension. After connecting the output of each head, the MHA is:

$$\mathrm{Att}_H = \mathrm{concat}(\mathrm{Att}_{H_1},\cdots,\mathrm{Att}_{H_H})\boldsymbol{W}_H^O \tag{16}$$

where W_H^O represents a learnable parameter matrix. Attention Att_H is sent to the fully connected layer f_{L}, and the internal covariate shift problem[30] in the network learning process is solved by the BN layer f_{BN}. The output of the encoder network is obtained by the ReLU activation function.

Note that only a set of fully connected network calculations is implemented here. Because the two sets of calculation processes are the same, the second is ignored.

3) Predictor network. The predictor network is explained in Section 2.2.1. It is only applied to the pre-training learning process of data representation and will not be used as a reserved part of the pre-training model.

4) Classifier network. The classifier network performs anomaly detection and judgment on the deep representation data, which consists of a fully connected layer, a ReLU layer, an output layer, and a Softmax layer. It forms an anomaly detection model together with the pre-trained encoder network. In the learning process of the model, the network parameters are fine-tuned and learned through a small amount of labeled data, and the prediction results of the data are output through the Softmax layer. The network selects cross entropy as the loss function.

Based on the above analysis, the pseudo-code of SiamARN algorithm is shown as follows.

Algorithm 1: SiamARN

Input: Unlabeled dataset X;
Output: Encoder network f, anomaly detection result y

Initialize hyper-parameters;
While epoch $\leqslant$ $\text{epoch}_{\max}$ do
  Sample a batch ($X - \{x\} \in \boldsymbol{R}^{b\times 1\times n}$);
  Get the augmentation data $x_1 = \mathcal{T}(x)$, $x_2 = \mathcal{T}(x)$;
  Get the deep representation $e_1 = f(x_1)$, $e_2 = f(x_2)$;
  Get the prediction $p_1 = h(f(x_1))$, $p_2 = h(f(x_2))$;
  Add stop gradient e_1, $e_2 \leftarrow e_1.\text{detach}()$, $e_2.\text{detach}()$
  Loss function: $\mathcal{L} = \frac{1}{2}\text{sim}(p_1, \text{stopgrad}(e_2)) + \frac{1}{2}\text{sim}(p_2, \text{stopgrad}(e_1))$;
  Update model parameters θ of encoder network and predictor network;
end
Save encoder network;
Add classifier network;
Initialize hyper-parameters of classifier network;
Select a small number of label data;
While epoch $\leqslant$ $\text{epoch}_{\max}$ do
  Sample a batch (X);
  Cross entropy loss function: $\mathcal{L} = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$;
Update model parameters θ of classifier network;
end;
Save SiamARN model;

2.3 Anomaly Detection Based on SiamARN

The anomaly detection process based on SiamARN is divided into three steps. The process is shown in Fig. 3, including the representation learning step, classification learning step, and anomaly detection step.

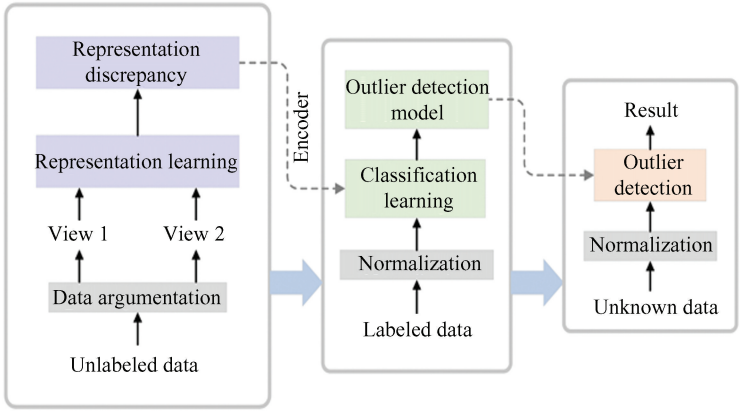


Fig.3 Anomaly detection procedure

Firstly, the unlabeled data can get different views of the data through DA. The two views can obtain the representation description of the data through representation learning, corresponding to the encoder network in Fig. 1, and send it into the classification learning. Then, a small number of labelled data are trained for classification learning after standardization. The purpose is to obtain a trained anomaly detection model for the final data anomaly judgment. Finally, after the unknown data is

standardized, the anomaly detection is carried out in the saved model, and the judgment result shows whether the data is abnormal.

3 Experimental Evaluation

This section mainly introduces the experimental content and result analysis of the SiamARN method. Several public data sets are selected for comparative experiments to verify the effectiveness of the proposed method in anomaly detection. The algorithm is based on Python 3.6 version and PyTorch deep learning framework.

3.1 Experimental Data

In the experiment, 14 public data sets in different fields were selected to verify the performance of SiamARN method in anomaly detection. The experimental data include medical field data, such as thyroid, breast cancer, diabetes, etc., as well as industrial field data, such as flight abnormalities, glass production, etc., and some common data sets, such as digital representation, letter pronunciation, etc. The feature dimensions of these data sets are different, covering a variety of data set sizes. Moreover, the proportion of abnormal data samples is also different, and it contains a variety of abnormal detection data situations, which have a certain universality. All experimental data are from https://odds.cs.stonybrook.edu/. Detailed data are shown in Table 1.

Table 1 Description of experimental data

Dataset	Instances	Features	Outliers	Proportion of outliers(%)	Remark
annthyroid	7200	6	534	7.42	Thyroid disease detection data
breastw	683	9	239	35.00	Breast cancer data
ionosphere	351	33	126	36.00	Ion layer anomaly data set in a satellite communication system
optdigits	5216	64	150	3.00	About 0-9 digital data sets
satellite	6435	36	2036	32.00	Satellite abnormal state data
vertebral	240	6	30	12.50	CT image spine segmentation data
wbc	278	30	21	5.60	White blood cell data
arrhythmia	452	274	66	15.00	Abnormal heart rhythm data
glass	214	9	9	4.20	Physical and chemical properties of glass
musk	3062	166	97	3.20	Molecular data of musk
pima	768	8	268	35.00	Diabetes data for Pima Indians
shuttle	49097	9	3511	7.00	Operation data of the space shuttle
vowels	1456	12	50	3.40	Vowel data on continuous acoustic characteristic variables

3.2 Experimental Settings

The accuracy rate (Acc), recall rate (R), F1 score (F1) and Area under the Precision - Recall Curve (AUPR) were selected as the evaluation indexes. The calculation formulas of the first three evaluation indexes are as follows:

$$\mathrm{Acc} = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TN} + \mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{20}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{21}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{TP}}{2 \times \mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{22}$$

where TP represents true positive, TN represents true negative, FP represents false positive, FN represents false negative.

The number of fully connected layer neurons in the encoder network and the predictor network was 128, and the number of fully connected layer neurons in the classifier network was 64 and 2, respectively. The initial learning rate is 0.002, which decreases with the increase in epoch. The batch sizes of the two learning were 64 and 32, respectively. The optimizer was SGD. In the SiamARN algorithm, due to the weakly supervised learning of a few samples, only 10% - 20 % of the data was selected as a small number of labeled data for anomaly detection learning. The parameters of other anomaly detection algorithms directly used the default parameters in the PyOD4 library, and the corresponding data set was divided into a training set (0.8) and a test set (0.2).

3.3 Experimental Results and Analysis

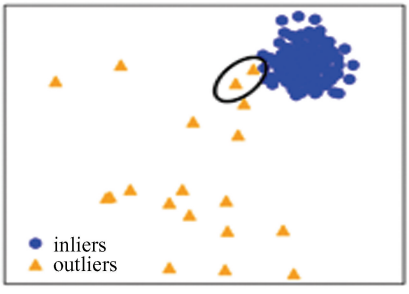
3.3.1 Experimental results of SiamARN

This section mainly discusses the role of key structures and parameters in SiamARN. The

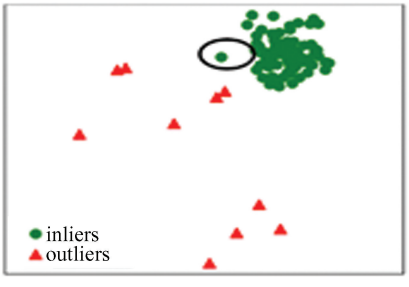
experiment mainly includes the comparison of data changes before and after representation learning, the comparison of DA method selection, the ablation experiment of network parameters, and the comparison experiment of MAH and deep neural network.

1) Representation learning

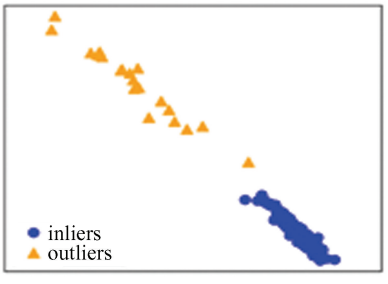
Fig. 4 shows two cases in anomaly detection. Fig. 4(a) shows anomaly data located at the edge of the normal data range, which are often misjudged as normal because they are close to the normal data. Fig. 4(b) shows normal data located at the edge of the normal data range. Because they are free from the range, it is often misjudged as abnormal. Fig. 4(c) and (d) are the deep representations obtained by the encoder network. For the above two cases, it is obvious that the distance between normal data is narrowed after the representation learning. The distinction between normal data and abnormal data becomes obvious, making it easier to realize the anomaly detection.



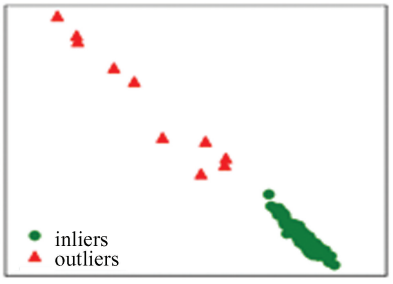
(a) Anomaly data located at the edge of the normal data range



(b) Normal data located at the edge of the normal data range



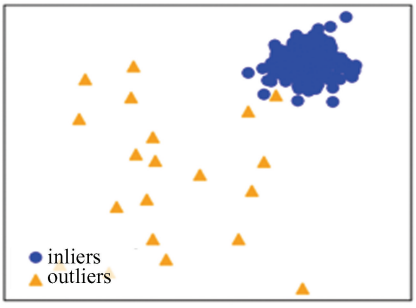
(c) Deep representation obtained by the encoder network (anomaly data)



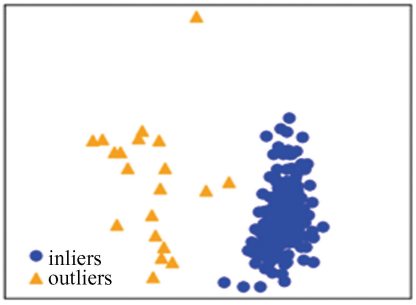
(d) Deep representation obtained by the encoder network (normal data)

Fig. 4 Data comparison before and after representation learning

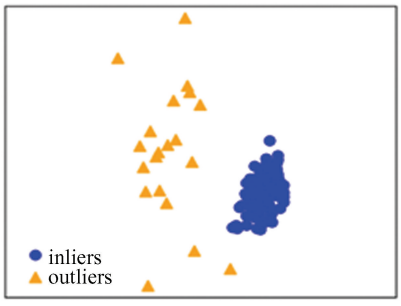
Fig. 5 describes the distribution of the deep representation of the original data in different dimensions through representation learning from a multi-dimensional perspective. Among them, Fig. 5(a) is the original data map, Figs. 5(b), (c) and (d) are the representations of the encoder network output on different feature dimensions. It can be seen that the distinction between abnormal and normal in multi-dimensional is still more obvious, but there are still individual anomalies slightly close to the edge of normal. Due to the variability of anomalies, it is difficult for contrastive representation learning to cluster all anomalies in similar regions, so abnormal data are often scattered in multiple locations, which can be distinguished from normal data gathered together.



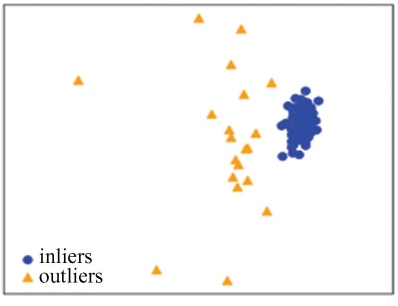
(a) Original data map



(b) Representation of the output of the encoder network on dimensions 1 and 2



(c) Representation of the output of the encoder network on dimensions 1 and 3



(d) Representation of the output of the encoder network on dimensions 1 and 4

Fig. 5 Representation learning of multi-dimensional data

2) DA analysis

Table 2 shows the anomaly detection results obtained after selecting different DA methods to process the data. Obviously, in SiamARN, representation learning can be achieved under the premise of the two different views of data. According to the experimental results, in most cases, there are small differences between the left and right order results of the same DA combination, but this difference itself contains the randomness of network learning, which has little effect on the experimental results and can often be ignored.

Among them, the experimental results based on the combination of the Z-score processed view and the original view are the best, and the experimental results based on the combination of $(L_1, -)$ data based on Max-Min are the worst. Different DA methods correspond to different experimental results. In general, the results are similar. This shows that, in most cases, the choice of DA has a certain impact on the performance of the model, but the impact is relatively small. The only special case is shown in the last column in Table 2, which is quite different from the results of other combinations. Therefore, in the selection of DA method, it is still necessary to select a more appropriate method after a certain comparative experiment.

Table 2 Experimental results under different DA methods

DA	(-,Z-score)	(Z-score,-)	(-,Max-Min)	(Max-Min,-)	Z-score		Max-Min	
					$(-, L_1)$	$(L_1, -)$	$(-, L_1)$	$(L_1, -)$
Acc	**0.9810**	0.9780	0.9766	0.9766	0.9766	0.9795	0.9751	0.9488
R	0.9745	**0.9960**	0.9686	0.9580	0.9959	0.9831	0.9841	0.9536
F1	**0.9724**	0.9701	0.9686	0.9661	0.9681	0.9708	0.9669	0.8281
AUPR	**0.9543**	0.9431	0.9430	0.9481	0.9395	0.9485	0.9410	0.8781

Note: "-" indicates that no DA processing was performed on the input data.

3) Ablation experiment.

Table 3 and Fig.6 show the results of the ablation experiment and the loss function curve of the representation learning process, respectively, where SG represents the stop-gradient, BN represents the batch standardization, Proj represents the Projection module, and Pred represents the predictor module. The experiments discuss the role of stopping gradient and batch normalization in the network. When both stop-gradient and batch normalization are retained, the experimental results of the model are the best. It can be seen from Table 3 that the experimental results of the model gradually deteriorate as the stop-gradient and batch standardization disappear in turn. Among them, the results of (c), (e), (g) and (h) are the worst, and their common feature is that there is no batch standardization of projection module, which indicates that the batch standardization under this module has a significant influence on the experimental results of the model. Obviously, batch standardization has a great optimization effect on the learning effect of the model. According to the experimental results, it seems that the experimental results of the model do not decrease much when there is no stop-gradient. However, combined with Fig. 6, the loss function will decrease rapidly without stop-gradient, which also shows that the stop-gradient plays a role in preventing model collapse in the representation learning.

Table 3 Results of ablation experiment

Case	Stop-grad	BN		Acc	*R*	F1 score	AUPR
		Proj.MLP	Pred.MLP				
(a) Normal	✓	✓	✓	**0.9301**	0.8686	0.8831	**0.8267**
(b) No SG	–	✓	✓	0.9284	**0.8691**	**0.8853**	0.8258
(c) No BN.Proj	✓	–	✓	0.6831	0	0	0.3168
(d) No BN.Pred	✓	✓	–	0.9015	0.8231	0.8401	0.7618
(e) No SG+BN.Proj	–	–	✓	0.6868	0	0	0.3131
(f) No SG+BN.Pred	–	✓	–	0.8982	0.8116	0.8351	0.8579
(g) No BN	✓	–	–	0.6996	0.0464	0.0886	0.3467
(h) No All	–	–	–	0.6853	0	0	0. 3147

Comparing the loss function curve in Fig. 6, it shows that the loss function decreases rapidly and is very close to the minimum loss of −1 when there is no stop gradient or batch normalization. Especially when the two do not exist at the same time, the loss function curve directly becomes −1. Obviously, the model finds a 'convenient' path in the learning process, such as an extreme possibility: all weights are updated to 0. In this case, the model collapse phenomenon, also known as collapsing solutions, occurs. According to Fig. 6, the two better cases are (a) and (d), respectively. Compared with other cases, batch standardization and stopping gradient based on the projection module play a key role in preventing model collapse.

4) Attention

Table 4 compares the performance of the encoder network using MHA and deep neural network (DNN) on the anomaly detection method. Among them, the deep network is composed of three groups of fully connected networks. The structure and parameters of the network are consistent with the predictor network *h*. Table 5 shows the average time for the model running 1 epoch under the two networks. Fig. 7 compares the convergence of model learning under the same data set.

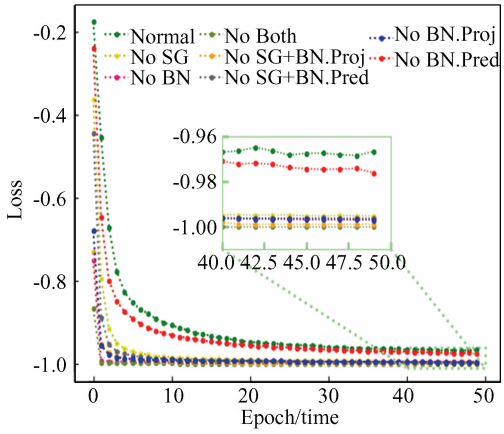


Fig.6 Loss function in ablation experiment

Table 4 Experimental results of MHA and DNN

Dataset	MHA				DNN			
	Acc	*R*	F1	AUPR	Acc	*R*	F1	AUPR
annthyroid	**0.9675**	**0.7413**	**0.7767**	**0.6244**	0.9513	0.3883	0.5388	0.3865
arrhythmia	**0.9004**	**0.4118**	**0.5545**	**0.4379**	0.8962	0.3333	0.4719	0.3622
breastw	**0.9648**	**0.9460**	**0.9500**	**0.9015**	0.9502	0.9098	0.9289	0.8954
glass	**0.9953**	**1.0**	**0.9524**	**0.9091**	0.9720	0.6667	0.7273	0.5520
ionosphere	0.8889	0.6953	0.8203	0.8064	**0.9373**	**0.8413**	**0.9060**	**0.8827**
musk	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9990	0.9709	0.9852	0.9719
optdigits	**0.9960**	**0.8954**	**0.9288**	**0.8670**	0.9906	0.6993	0.8137	0.6891
pima	**0.8242**	0.6750	**0.7368**	**0.6660**	0.8099	**0.6809**	0.7245	0.6443
satellite	**0.8974**	**0.7607**	**0.8225**	**0.7557**	0.8797	0.7092	0.7889	0.7225
shuttle	0.9957	0.9432	0.9689	0.9435	**0.9961**	**0.9642**	**0.9726**	**0.9487**
vertebral	**0.9625**	**0.7586**	**0.8302**	**0.7246**	0.9333	0.4583	0.5789	0.4143
vowels	**0.9966**	**0.8810**	**0.9367**	**0.8844**	0.9800	0.5000	0.6234	0.4303
wbc	**0.9894**	**0.8182**	**0.9000**	**0.8288**	0.9841	0.7273	0.8500	0.7430
wine	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9845	**1.0000**	0.8571	0.7500

Table 5 Running time

Network	Time(s)
MHA.pre_train	3.0513
MHA. train	8.9237
DNN.pre_train	3.0800
DNN. train	9.2560

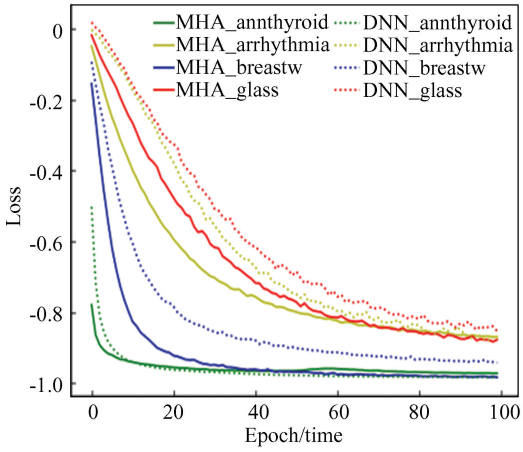


Fig.7 Loss function of MHA and DNN

Compared with Tables 4-5 and Fig.7, in terms of model performance, training time, and convergence speed, the representation learning structure using MHA is significantly better than DNN. MHA processes multiple independent attention heads in parallel, and each head can focus on capturing a specific correlation, to capture the deep correlation between data features. While improving the learning efficiency of the model, it can better realize the deep representation learning of data. Therefore, MHA is more suitable for the method proposed in this paper.

3.3.2 Contrast experiment

In the experiment, seven commonly used anomaly detection methods were selected for comparison, including KNN[31], LOF[32], CBLOF[11], OCSVM[33], IForest[34], KPCA[35], ABOD[10], and AE[36]. Experimental results for these methods are derived from an open source Python toolkit for anomaly detection, PyOD4[37]. Tables 6-9 show the experimental results of SiamARN and its comparison methods on 14 different data sets.

Table 6 Accuracy comparison of SiamARN versus seven anomaly detection methods

Dataset	KNN	LOF	CBLOF	OCSVM	IForest	KPCA	ABOD	AE	SiamARN
annthyroid	0.8803	0.8869	0.8717	0.8511	0.8876	0.8733	0.9258	0.9008	**0.9675**
arrhythmia	0.8650	0.8539	0.8739	0.8717	0.8827	0.5111	0.8451	0.8801	**0.9004**
breastw	0.7379	0.5988	0.7701	0.7672	0.7818	0.6149	0.6501	0.7321	**0.9648**
glass	0.8738	0.9112	0.8692	0.8785	0.8831	0.4953	0.8692	0.9206	**0.9953**
ionosphere	0.7664	0.7493	0.7664	0.7606	0.7692	0.5613	0.7664	0.7692	**0.8889**
musk	0.9095	0.8828	0.9363	0.8883	0.9337	0.4605	0.8439	0.9997	**1.0000**
optdigits	0.8877	0.8921	0.8963	0.8753	0.8844	0.4593	0.8631	0.6484	**0.9960**
pima	0.6679	0.6510	0.6680	0.6523	0.6732	0.5456	0.6771	0.6445	**0.8242**
satellite	0.7226	0.6870	0.7812	0.7638	0.7803	0.4914	0.6730	0.7855	**0.8974**
shuttle	0.8747	0.8551	0.8801	0.9667	0.9709	0.9196	0.8617	0.9868	**0.9957**
vertebral	0.8125	0.7500	0.7833	0.7750	0.7958	0.5708	0.7667	0.8083	**0.9625**
vowels	0.9505	0.9279	0.9203	0.8750	0.9004	0.5337	0.9293	0.9457	**0.9966**
wbc	0.9338	0.9444	0.9339	0.9312	0.9312	0.4841	0.8995	0.9365	**0.9894**
wine	0.8605	0.8992	0.9224	0.8450	0.9070	0.5581	0.8527	0.9147	**1.0000**

In the experiment, the more commonly used classical anomaly detection methods and deep anomaly detection methods are selected for comparison. The results show that the SiamARN anomaly detection method proposed in this paper shows the best performance. In most cases, the performance of classical anomaly detection methods is worse than that of deep anomaly detection methods, and the deep anomaly detection algorithm is worse than SiamARN. On some datasets, the performance of deep anomaly detection, such as AE algorithm, is basically similar to SiamARN, but the applicability and generalization of AE are obviously inferior to SiamARN.

The advantage of SiamARN mainly lies in the contrastive learning structure. Different views of its own data are used as contrastive inputs, and the similarity of deep representations is used as a loss function for training. This allows the model to learn

without relying on the labelled samples, without building negative samples for contrast, and without considering the interference of abnormal data. It maps the original data to a new feature space, so that the abnormal and normal data with an adjacent relationship are far away from each other. Therefore, the distinction between abnormal and normal data is improved. The resulting deep representation can be easily distinguished between abnormal and normal data by a simple classifier network.

Table 7 Recall rate comparison of SiamARN versus seven anomaly detection methods

Dataset	KNN	LOF	CBLOF	OCSVM	IForest	KPCA	ABOD	AE	SiamARN
annthyroid	0.3090	0.3633	0.2921	0.1667	0.4532	0.4045	0	0.3146	**0.7413**
arrhythmia	0.4091	0.3636	0.3940	0.4242	0.4090	**0.6515**	0.3788	0.3689	0.4118
breastw	0.4169	0.0711	0.3556	0.3347	0.3849	0.4268	0	0.3054	**0.9460**
glass	0.3333	0.5556	0.3333	0.1111	0.3333	0.7778	0.3333	0.1111	**1.0000**
ionosphere	0.3492	0.3175	0.3492	0.3651	0.3571	0.5556	0.3492	0.6587	**0.6953**
musk	0.3711	0.1031	**1.0000**	0.3711	1.0000	0.6701	0.1134	0.9897	**1.0000**
poptdigits	0.0733	0.2133	0.4667	0.0667	0.2733	0.5600	0.2267	0.1343	**0.8954**
pima	0.1455	0.1418	0.1940	0.1530	0.1940	0.5112	0.1903	0.1231	**0.6750**
satellite	0.2102	0.1582	0.3094	0.2903	0.3193	0.5845	0.1684	0.3222	**0.7607**
shuttle	0.2202	0.1752	0.3469	0.9613	0.9846	0.6283	0.2421	0.8992	**0.9432**
vertebral	0	0.1333	0.0667	0.1000	0.0333	0.6333	0.1000	0.1333	**0.7586**
vowels	**0.9400**	0.6800	0.7600	0.2800	0.3800	0.3800	0.9400	0.1400	0.8810
wbc	0.7143	0.6667	0.7143	0.7143	0.7143	0.6190	0.7143	0.4286	**0.8182**
wine	0	0.7000	0.8000	0.6000	0.5000	0.7000	0.1000	0	**1.0000**

Table 8 F1 scores comparison of SiamARN versus seven anomaly detection methods

Dataset	KNN	LOF	CBLOF	OCSVM	IForest	KPCA	ABOD	AE	SiamARN
annthyroid	0.2768	0.3228	0.2524	0.1426	0.3743	0.3214	0	0.3200	**0.7767**
arrhythmia	0.4696	0.4211	0.4771	0.4912	0.5046	0.2801	0.4065	0.3149	**0.5545**
breastw	0.4079	0.1104	0.5199	0.5016	0.5526	0.4368	0	0.4438	**0.9500**
glass	0.1818	0.3448	0.1765	0.0714	0.1935	0.1148	0.1765	0.1053	**0.9524**
ionosphere	0.5176	0.4762	0.5176	0.5227	0.5263	0.4762	0.5176	0.6721	**0.8203**
musk	0.2063	0.0528	0.4987	0.1739	0.4887	0.0730	0.0440	0.9948	**1.0000**
optdigits	0.0362	0.1021	0.2056	0.0299	0.1197	0.0562	0.0870	0.2105	**0.9288**
pima	0.2342	0.2209	0.2897	0.2350	0.2930	0.4398	0.2914	0.1947	**0.7368**
satellite	0.3241	0.2423	0.4723	0.4375	0.4590	0.4210	0.2459	0.4874	**0.8225**
shuttle	0.2008	0.1474	0.2926	0.8051	0.8286	0.5279	0.2003	0.9068	**0.9689**
vertebral	0	0.1176	0.0714	0.1000	0.0392	0.2695	0.0968	0.1481	**0.8302**
vowels	0.5663	0.3931	0.3958	0.1333	0.2077	0.0530	0.4772	0.1505	**0.9367**
wbc	0.5455	0.5714	0.5455	0.5357	0.5357	0.1176	0.4412	0.4286	**0.9000**
wine	0	0.5185	0.6154	0.37850	0.4545	0.1972	0.0952	0	**1.0000**

Then, why not directly use the above KNN and other simple algorithms for anomaly detection on deep representation data? This approach is similar to the deep learning method for feature extraction mentioned in Introduction. On the one hand, the method of deep learning based on feature extraction mainly extracts low-dimensional features from high-dimensional and nonlinear complex data, such as images and sound waves, to reduce data complexity.

However, for tabular data, the representation learning result of SiamARN is not a simple dimensionality reduction. On the contrary, the number

of features of deep representation output by the encoder may be more than the original data. Its main purpose is to narrow the distance between similar data, which makes the normal data close to each other. Although abnormal data may exist in various cases, their distribution is different from that of normal data, so it can achieve the purpose of keeping away from normal data. On the other hand, Ref.[13] mentioned that the process before and after the method is completely disconnected, which easily leads to unsatisfactory abnormal scores. Compared with SiamARN, the encoder network used for representation learning participates in the later anomaly detection. Therefore, the weakly supervised learning process with a small number of label samples can be used to distinguish abnormal data from normal data. This is also illustrated by the experimental results in Tables 6-9.

Table 9 AUPR comparison of SiamARN versus seven anomaly detection methods

Dataset	KNN	LOF	CBLOF	OCSVM	IForest	KPCA	ABOD	AE	SiamARN
annthyroid	0.1288	0.1527	0.1174	0.0825	0.1850	0.1520	0.0742	0.1532	**0.6244**
arrhythmia	0.3117	0.2747	0.3267	0.3315	0.3557	0.1671	0.2568	0.1481	**0.4379**
breastw	0.5082	0.3426	0.5690	0.5675	0.5920	0.3915	0.3500	0.4908	**0.9015**
glass	0.0697	0.1575	0.0680	0.0432	0.0735	0.0575	0.0651	0.0485	**0.9091**
ionosphere	0.5828	0.5474	0.5828	0.5638	0.5879	0.3910	0.5828	0.5744	**0.8064**
musk	0.0729	0.0321	0.3322	0.0621	0.3233	0.0363	0.0312	0.9900	**1.0000**
optdigits	0.0284	0.0369	0.0769	0.0281	0.0418	0.0292	0.0344	0.3674	**0.8670**
pima	0.3855	0.3704	0.3921	0.3730	0.3972	0.3679	0.4009	0.3632	**0.6660**
satellite	0.3986	0.3482	0.5269	0.4821	0.5215	0.3238	0.3397	0.5367	**0.7557**
shuttle	0.0964	0.0813	0.1345	0.6685	0.7054	0.3126	0.0955	0.8295	**0.9435**
vertebral	0.1250	0.1224	0.1218	0.1225	0.1224	0.1542	0.1219	0.1306	**0.7246**
vowels	0.3829	0.1990	0.2116	0.0492	0.0756	0.0321	0.3026	0.0523	**0.8844**
wbc	0.3310	0.3519	0.3310	0.3220	0.3220	0.0614	0.2438	0.2154	**0.8288**
wine	0.0775	0.3115	0.4155	0.1946	0.2471	0.1036	0.0789	0.0775	**1.0000**

3.3.3 Interpretability analysis

The interpretability of the SiamARN method mainly includes the self-learning process of the Siamese contrastive structure and the feature representation learning based on MAH.

The self-learning process of Siamese contrastive structure refers to maximizing the similarity between different enhanced views of the same input to learn feature representation. It introduces dynamic targets and asymmetry through predictors and stop gradient, implicitly achieving the goal of contrastive learning. The stop gradient interrupts the gradient backpropagation process of one branch, and the predictor only acts on one branch. This asymmetry forces the model to learn a non-trivial representation and prevents the model from collapsing. Thus, the model gradually converges to a meaningful feature representation in the alternating optimization process.

The feature representation learning of MHA refers to the feature representation of the input obtained by the encoder network in the representation learning. MHA parallel computes the attention weight through multiple independent 'heads'. Each head has an independent query, key, and value weight matrix to generate different attention distributions.

The outputs of each head are concatenated and linearly transformed to obtain the final result. It allows the model to focus on multiple dependency patterns simultaneously, including local and global relationships. The weight calculation can visually display the correlation information between the features of each position and other features, enhancing the interpretability of the model.

4 Conclusions

In this paper, a weakly supervised anomaly detection method called SiamARN is proposed. The contrastive self-supervised learning is used to realize the representation learning of unlabelled data, and the weakly supervised classification learning for data anomaly detection is realized by using a few existing

labelled data. Experimental results show that the method exhibits better performance in anomaly detection.

The main advantages of this method are as follows: Firstly, it can solve the problem that unsupervised anomaly detection methods cannot utilize the prior knowledge of existing anomalies. And SiamARN avoids the waste of prior knowledge and improves the recall rate of anomaly detection by a weakly supervised learning process. Secondly, the problem of edge anomaly detection is solved by the contrastive learning structure. The obtained deep representation data makes the distinction between abnormal and normal data more obvious in the feature space. Finally, the introduction of self-supervised representation learning can avoid the problem of dependence on the abnormal distribution hypothesis, and a network structure based on MHA is designed to learn the deep relationship of the data without analyzing the original data.This eliminates the need for analysis and processing of raw data.

However, according to the experimental results, it can be seen that SiamARN still has a large room for improvement in rating indicators, such as the recall rate in some data. Therefore, the research goal of the next stage should focus on data representation learning,that is, how to make the distinction between abnormal data and normal data easier through new comparative learning methods.

References

[1] Tao X, Zhang D, Ma W, et al. Unsupervised anomaly detection for surface defects with dual-Siamese network. IEEE Transactions on Industrial Informatics, 2022, 18(11):7707-7717. DOI:10.1109/TII.2022.3142326.

[2] Li T, Kou G, Peng Y, et al. An integrated cluster detection, optimization, and interpretation approach for financial data. IEEE Transactions on Cybernetics, 2022, 52(12):13848-13861.DOI: 10.1109/tcyb.2021.3109066.

[3] Yang Y,Fan C J, Chen L, et al. IPMOD: An efficient outlier detection model for high-dimensional medical data streams. Expert Systems with Applications, 2022,191(4): 116212.DOI:10.1016/j.eswa.2021.116212.

[4] An P, Wang Z Y, Zhang C J. Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection. Information Processing & Management, 2022,59(2):102844.DOI:10.1016/j.ipm.2021.102844.

[5] Wang X Y,Duan L, Yu Z Y, et al. Robust multi-kernel nearest neighborhood for outlier detection. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(8):4220-4231. DOI:10.1109/TKDE.2024.3364179.

[6] Du X S,Chen J Y, Yu J, et al. Generative adversarial nets for unsupervised outlier detection. Expert Systems with Applications,2024,236(2):121161.DOI:10.1016/j.eswa.2023.121161.

[7] Ke W J, Wei J G, Xiong N X, et al. GSS: A group similarity system based on unsupervised outlier detection for big data computing. Information Sciences, 2023, 620(1): 1-15.DOI:10.1016/j.ins.2022.11.078.

[8] Rousseeuw P J, Van Driessen K. Computing LTS regression for large data sets. Data Mining and Knowledge Discovery,2006, 12(1): 29-45. DOI: 10.1007/s10618-005-0024-4.

[9] Jin W, Tung A K H, Han J W, et al. Ranking outliers using symmetric neighborhood relationship. Pacific-Asia Conference on Knowledge Discovery and Data Mining, Berlin:Springer,2006:577-593. DOI:10.1007/11731139_68.

[10] Kriegel H P, Schubert M, Zimek A. Angle-based outlier detection in high-dimensional data. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM,2008:444-452.DOI:10.1145/1401890.1401946.

[11] He Z Y, Xu X F, Deng S C. Discovering cluster-based local outliers. Pattern Recognition Letters, 2003, 24(9-10):1641-1650. DOI:10.1016/S0167-8655(03)00003-5.

[12] Smiti A. A critical overview of outlier detection methods. Computer Science Review, 2020, 38(11): 100306. DOI: 10.1016/j.cosrev.2020.100306.

[13] Pang G S,Shen C H, Cao L B, et al. Deep learning for anomaly detection: A review. ACM Computing Surveys, 2021,54(2):1-38.DOI:10.1145/3439950.

[14] Chen J H,Sathe S, Aggarwal C, et al. Outlier detection with autoencoder ensembles. Proceedings of the 2017 SIAM International Conference on Data Mining (SDM). Philadelphia:SIAM, 2017: 90-98.

[15] Abhaya A, Kr Patra B. An efficient method for autoencoder based outlier detection. Expert Systems with Applications, 2023,213(Part A): 118904.DOI:10.1016/J.ESWA.2022.118904,

[16] Schlegl T, Seeböck P M, Waldstein S M, et al. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. Medical Image Analysis, 2019,54(5): 30-44.

[17] Wang S Q, Zeng Y J, Liu X W, et al. Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, 536: 5962-5975.

[18] He K M, Fan H Q, Wu Y X, et al. Momentum contrast for unsupervised visual representation learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle,2020:9726-9735.DOI:10.

1109/CVPR42600.2020.00975.

[19] Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations.arXiv, 2020, arXiv: 2002.05709.

[20] Grill J B, Strub F, Altché F, et al. Bootstrap your own latent: A new approach to self-supervised learning. arXiv, 2020, arXiv: 2006.07733.

[21] Chen X L, He K M. Exploring simple Siamese representation learning. arXiv, 2020, arXiv: 2011.10566.

[22] Yang Y Y, Zhang C L, Zhou T, et al. DCdetector: Dual attention contrastive representation learning for time series anomaly detection. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York:ACM, 2023: 3033-3045.

[23] Zou Y, Jeong J, Pemula L, et al. SPot-the-difference self-supervised pre-training for anomaly detection and segmentation. Computer Vision-ECCV 2022,13690:392-408.DOI:10.1007/978-3-031-20056-4_23.

[24] Pei M T, Yan B, Hao H L, et al. Person-specific face spoofing detection based on a Siamese network. Pattern Recognition, 2023, 135 (3): 109148. DOI: 10. 1016/j. patcog.2022.109148.

[25] Liu X, Zhang F J, Hou Z Y, et al. Self-supervised learning: generative or contrastive. IEEE Transactions on Knowledge and Data Engineering, 2023,35(1):857-876. DOI:10.1109/TKDE.2021.3090866.

[26] Bromley J, Guyon I, LeCun Y, et al. Signature verification using a "Siamese" time delay neural network. Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS). Denver Colorado,1993:737-744.

[27] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS). Long Beach, 2017: 6000-6010.

[28] Suljagic H, Bayraktar E, Celebi N. Similarity based person re-identification for multi-object tracking using deep Siamese network. Neural Comput & Applic,2022,34(6): 18171-18182.DOI:10.1007/s00521-022-07456-2.

[29] Yang J F, Zou H, Zhou Y X, et al. Learning gestures from WIFI: a Siamese recurrent convolutional architecture. IEEE Internet of Things Journal, 2019, 6 (6): 10763 - 10772.DOI:10.1109/JIOT.2019.2941527.

[30] Ioffe S, Szegedy C. Batch normalization: Accelerating deep 820 network training by reducing internal covariate shift. International Conference on Machine Learning, 2015:arxiv.org/pdf/1502.03167v3.

[31] Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD). New York: ACM, 2000:427-438. DOI:10.1145/335191.335437.

[32] Breunig M M, Kriegel H, Ng R T, et al. LOF: identifying density-based local outliers. In Proceedings of the 2000 International Conference on Management of Data (SIGMOD), New York:ACM,2000:93-104.

[33] Schölkopf B, Platt J C, Shawe-Taylor J, et al. Estimating the support of a high-dimensional distribution. Neural Computation, 2001, 13 (7): 1443 - 1471. DOI: 10. 1162/089976601750264965.

[34] Liu F T, Ting K M, Zhou Z H. Isolation forest. Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM). Piscataway: IEEE, 2008: 413-422.DOI:10.1109/ICDM.2008.17.

[35] Hoffmann H. Kernel PCA for novelty detection. Pattern Recognit,2007,40(3):863-874. DOI: 10. 1016/j. patcog. 2006.07.009.

[36] Sakurada M, Yairi T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA). New York: ACM, 2014: 4-11. DOI: 10.1145/2689746.2689747

[37] Zhao Y, Nasrullah Z, Li Z. PyOD: A python toolbox for scalable outlier detection. Journal of Machine Learning Research,2019,20(96):1-7.