Engineering Applications of Artificial Intelligence 126 (2023) 107071

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A graph neural network-based data cleaning method to prevent intelligent fault diagnosis from data contamination

Shuhui Wang [a], Yaguo Lei [a], Bin Yang [a,*], Xiang Li [a], Yue Shu [b], Na Lu [c]

[a] Key Laboratory of Education Ministry for Modern Design and Rotor-Bearing System, Xi'an Jiaotong University, Xi'an, 710049, China
[b] State Key Laboratory of Compressor Technology (Compressor Technology Laboratory of Anhui Province), Hefei, Anhui, 230031, China
[c] School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an, 710049, China

ARTICLE INFO

Keywords:
Mechanical fault diagnosis
Data cleaning
Graph neural network
Graph clustering

ABSTRACT

The success of deep learning (DL) based-mechanical fault diagnosis hinges on the high quality of training data. However, it is difficult to acquire high-quality mechanical monitoring data due to data contamination: 1) Monitoring device irregularities, such as sensor malfunction and signal transmission disruption, bring anomalies into the training data; 2) human labour-based data annotation inevitably produces incorrectly labeled data. These two types of data contamination degrade the performance of DL models. To address the aforementioned issue, this paper proposes a graph neural network-based data-cleaning method. In the first stage, a group anomaly detector is designed to identify the presence of anomalous data. This detector incorporates affinity graphs for depicting data groups and subsequently calculates the group anomaly score to determine the abnormal group. In the second stage, a graph clustering model is developed to relabel the mislabeled data. This model takes advantage of the graph neural network's proficiency in handling affinity graphs to prepare clean labels for subsequent network training. Experimental results, conducted on a pump and an industrial robot joint reducer, show the proposed method's ability to effectively detect anomalous data and rectify incorrect labeling, surpassing the performance of baseline methods in mechanical fault diagnosis.

1. Introduction

Intelligent fault diagnosis plays an important role in maintaining the safety, dependability, and effectiveness of mechanical equipment. The exponential advancement in sensor technologies, coupled with the deployment of DL, has initiated a data-driven revolution in the sphere of intelligent fault diagnosis (Lei et al., 2022), (Lei et al., 2020). Based on the superior learning capacity from a large amount of data, various DL models are exploited for intelligent fault diagnosis, such as auto-encoders (AEs) (Lei et al., 2016), (Meire et al., 2023), graph neural networks (GNNs) (Ghorvei et al., 2023), (Kavianpour et al., 2022), convolutional neural networks (CNNs) (Wang et al., 2023a), (Yang et al., 2023). Given that these data-driven approaches extract diagnostic knowledge from data, the quality of data directly impacts the efficacy of DL models. While it is impractical to collect high-quality data from only one device, the recently emerging federated learning enables strong diagnosis models through their powerful capacity to learn from distributed data in different edge devices (Guo et al., 2022), (Yu et al., 2023). This learning paradigm reduces the influence of bad data by extending training data from more local clients rather than directly removing the bad data, which in turn increases the computational costs for training models.

DL models, as data-driven tools, necessitate high-quality data to ensure precise and reliable mechanical fault diagnosis (Xu et al., 2019). However, in real-world scenarios, the data acquired from machines often suffer from contamination. The degradation in data quality is attributable to two aspects: anomalous data and incorrect labeling. (1) Anomalies in data, an unavoidable byproduct of irregularities in monitoring devices, deteriorate the overall data quality. In industrial environments, monitoring device failures are common due to harsh conditions, introducing anomalies in the monitored data. For example, sensor malfunctions can lead to signal drift, which will render the monitoring signal incapable of reflecting the health information of the equipment. Besides, failures such as those in the transmission line will result in signal transmission disruptions, consequently impairing the ability of the monitoring signals to reflect the health information of the equipment. (2) Incorrect labeling is another issue. In practice, labeling the monitoring data is typically executed by diagnostic experts.

* Corresponding author.
E-mail address: binyang@xjtu.edu.cn (B. Yang).

https://doi.org/10.1016/j.engappai.2023.107071
Received 1 June 2023; Received in revised form 20 August 2023; Accepted 28 August 2023
Available online 4 September 2023

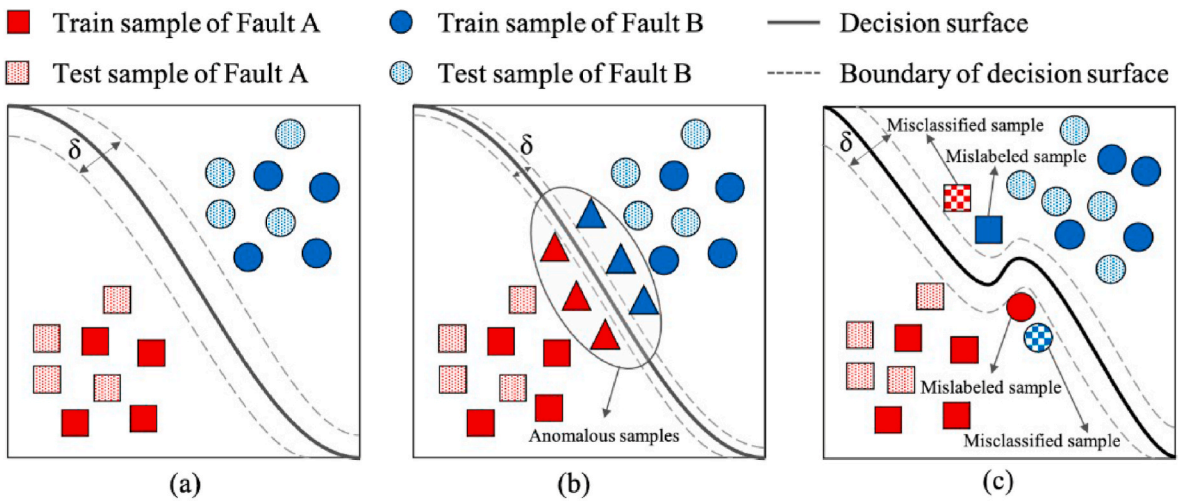

Fig. 1. Simulation of negative effects of data contamination on DL-based fault diagnosis: (a) Normal training on high-quality data, (b) Overfitting on anomalous samples, and (c) Overfitting on mislabeled samples.

However, due to environmental noise, the collected data may appear unapparent fault characteristics, leading to inadvertent mis-identifications. Furthermore, labeling a large amount of data is a tedious job. The high intensity of the workload inevitably introduces some random errors in the data labeling process.

When the contaminated data are served to train DL models, the performance of DL models is inevitably degraded (Lei et al., 2020). The negative effects of data contamination for training DL models can be observed in Fig. 1. The marker indicates fault types, while the colour indicates labels. Fig. 1(a) shows the DL model trained under high-quality data, where there are no anomalous samples and mislabeled samples. δ denotes the margin between the decision boundaries. Fig. 1(b) shows the DL model trained under anomalous samples. The DL model tends to overfit anomalous samples by searching for more precise decision boundaries, increasing the training duration. Fig. 1(c) illustrates the DL model trained under mislabeled samples, where some of the "Fault A″ is incorrectly labeled as "Fault B″ and some of the "Fault B″ is incorrectly labeled as "Fault A". The trained model finally overfits the mislabeled

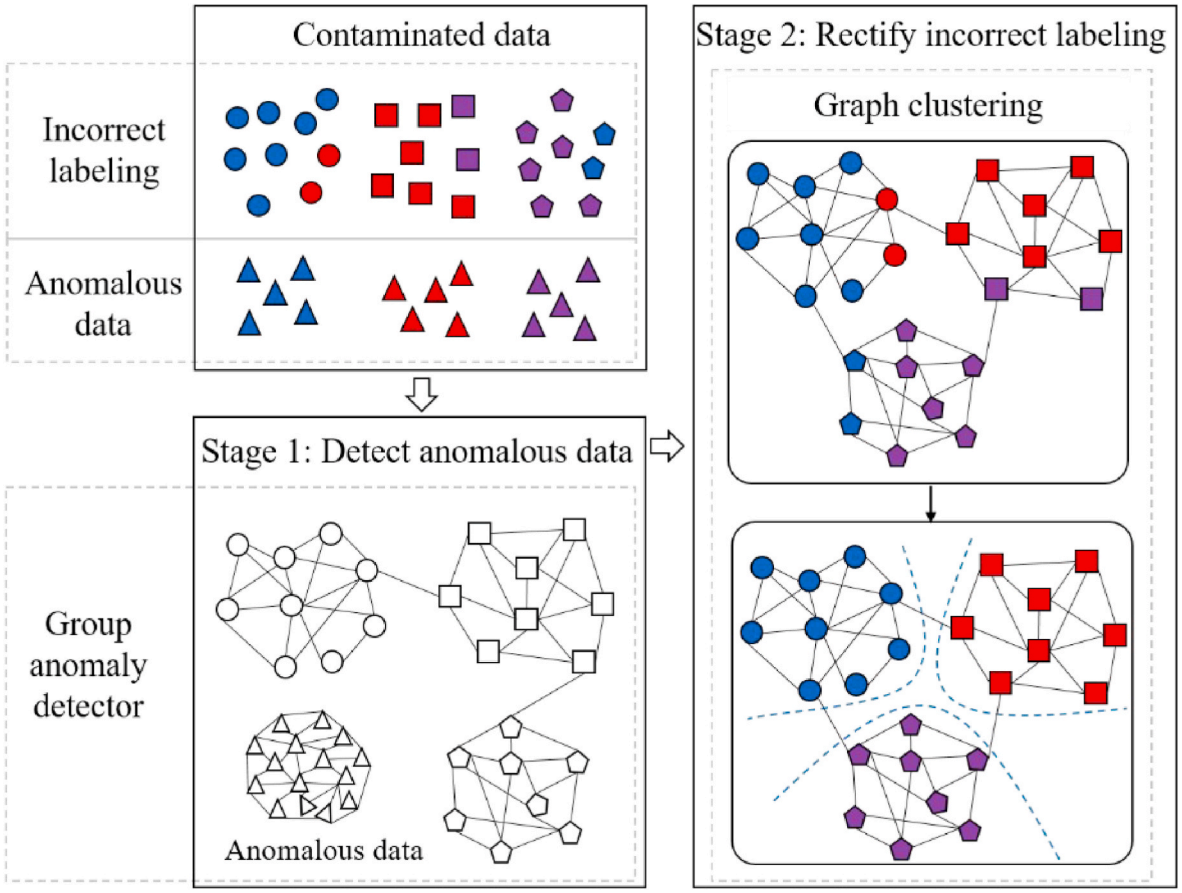

Fig. 2. Overview of the proposed method.

samples, which decreases the model's accuracy in testing samples. This is also known as the noisy label problem (Song et al., 2022) in existing studies.

To address model degradation due to data contamination, existing studies have demonstrated much progress by developing various combating strategies towards noisy labels. The combating strategies for training reliable DL models can be summarized into three categories: In the first category, researchers developed robust DL models regardless of the label noise information. By doing so, they designed new loss functions that are robust to label noise (Wang et al., 2019). Wang et al. (Wang and Li, 2022) introduced GCE loss to deep neural networks for robust fault diagnosis of mechanical equipment under noisy labels. Liang et al. (2022) employed an improved GCE loss for rolling bearing fault diagnosis under both label noise and environmental noise. In the second category, researchers developed noise-tolerant DL models which can self-correct noisy labels by making a premise of label noise information. Nie et al. (Nie and Xie, 2021a) introduced a label noise transition matrix to the training process of deep diagnostic models when they tries to reduce the label noise impact. Later they (Nie and Xie, 2021b) improved the label noise transition matrix and finally reached comparable results under heavy noisy labels. In addition to the above two categories, researchers also suggested new training strategies such as meta-learning to mitigate the bad impact induced by noisy labels. Existing studies show that these combating strategies do improve the performance of DL models to some extent. However, noisy labels remain in the learning process of DL models. Thus the efficiency of these strategies is contingent on precise usage or elimination of noisy labels. For instance, noise-tolerant DL models require an accurate estimation of label noise information, which is difficult to achieve in practical applications. Besides noisy labels, anomalous data also degrade DL model performance (Yao et al., 2023). Long et al. (2022) reported an image-based data-cleaning algorithm to locate anomalous data for wind turbine monitoring. Yao et al. (2023) provided a data-cleaning method for wind turbine state prediction by calculating the Thompson tau-local outlier factor of the collected SCADA data and then used LSTM to verify its effectiveness. Results show that the data cleaning method can reduce the DL model complexity by preprocessing the anomalous data in advance. Although significant progress has been accomplished in either the noisy labels or anomalous data, few works make discussions when the two scenarios are encountered together.

To address the aforementioned limitations, a two-stage mechanical data-cleaning method is proposed. The method is expected with the capacity to clean anomalous data, as well as deal with noisy labels. To handle the data contamination mentioned above, data groups for different fault types are modeled by affinity graphs in the method. Thanks to the excellent performance in dealing with graphs (Ghorvei et al., 2023), (Kavianpour et al., 2022), graph neural networks are employed to deal with affinity graphs for enhanced intelligent fault diagnosis. The main contributions are as follows: First, a group anomaly detector is designed to identify the presence of anomalous data. The detector incorporates data graph structure as well as data local density to obtain anomalous data group candidates and then calculates the group anomaly score of these candidates to decide the abnormal group. Second, a graph clustering model is developed to rectify the mislabeled data. Graph neural networks are leveraged to deal with data graph structure thus avoiding the bad impact of noisy labels on the successive network training.

The rest of the paper is organized as follows. Section 2 details the graph neural network-based data cleaning method. Section 3 presents experimental verifications and discussion. Finally, conclusions are drawn in Section 4.

2. Graph neural network-based data cleaning method

2.1. Overview of the proposed method

The overview of the proposed method is demonstrated in Fig. 2. The square, circle and pentagon represent distinct sorts of faults whereas the triangle denotes anomalous data. The use of colour signifies manual annotation. Correct labeling for the square, circle and pentagon should be blue, red and purple. The contaminated data that may deteriorate DL model training are shown in Fig. 2. The graph neural network-based data cleaning method is also illustrated in Fig. 2, which contains two stages. In the first stage, a group anomaly detector is developed by constructing affinity graphs embedded in the frequency domain of the data points. The k-nearest neighbour (k-NN) graph is utilized in the detector to connect data points that are closer together. Simultaneously, data density is redefined based on the k-NN graph to gather data points into distinct groups. The detector then calculates group anomaly scores for each cluster to determine the presence of anomalous data. In the second stage, a graph clustering model is employed for label correction of the mislabeled data. Relabeling of mislabeled data can be determined based on the majority of label information from neighbouring data points. A differentiable k-means layer is introduced to graph neural networks to facilitate graph clustering. This layer calculates a soft assignment for nodes to each of the cluster centers after graph embedding. Unlike an independent clustering procedure outside of the graph learning process, the graph clustering model performs differentiable optimization throughout the entire graph embedding and clustering process, resulting in more accurate outcomes.

2.2. Group anomaly detector

The group anomaly detector is designed based on the k-NN graph and data density. Given a set N points $X = \{x_1, x_2, \cdots, x_i, \cdots, x_N\}$ in a D-dimensional space, constructing k-NN graph is to find k-nearest neighbours of any data point x_i in X given some distance function, like Euclidean distance. The k-NN graph can be expressed by an adjacency matrix:

$$a_{i,j} = \begin{cases} w_{i,j}, & x_i \in \mathcal{N}_j^k \text{ or } x_j \in \mathcal{N}_i^k \\ 0, & otherwise \end{cases} \# \tag{1}$$

where $a_{i,j}$ is the element of the adjacency matrix A, $\mathcal{N}_j^k$ denotes the k-nearest neighbours of the data point x_j, and there is no self-loop in the graph. The edge weight $w_{i,j}$ of the k-NN graph is calculated by:

$$w_{i,j} = \exp\left(-s_{i,j}^2 \Big/ \gamma\right) \# \tag{2}$$

where $s_{i,j}$ is some sort of distance function, γ is the bandwidth. Data density is an important feature describing a group of data points (Li et al., 2022). In density peak clustering, for example, data density is defined by the number of data points within a pre-defined cutoff distance. In other scenarios, data density can be represented by some form of distance (e.g. reciprocal of the distance) that contains a pre-defined number of data points. Since for containing a given number of data points, a smaller distance means a larger density. To well measure the data density in high-dimensional space, it is redefined by some researchers as:

$$\rho_i = \left|\mathcal{N}_i^k\right|^2 \Big/ \sum_{x_j \in \mathcal{N}_i^k} s_{i,j} \# \tag{3}$$

where $\left|\mathcal{N}_i^k\right|$ denotes the number of k-nearest neighbours of x_i. The ρ_i takes consideration of the number of data points that are connected to x_i as well as the average distance of its k-nearest neighbours. This measure

typically works well for high-dimensional data, but it may not perform effectively when there are varying densities in the data space. For mechanical monitoring data cleaning, we consider the following data density of data x_i:

$$\zeta_i = \left|\mathcal{N}_i^k\right|\rho_i \Big/ \sum_{x_j \in \mathcal{N}_i^k} \rho_j \# \tag{4}$$

where ρ_i, ρ_j denote the data density defined by Eq. (3). The ζ_i is defined based on a k-NN graph. It is a normalized density of data point x_i in the k-NN graph by taking the mean values of its k-nearest neighbours. Hence, we call ζ_i as k-NN density.

After constructing the k-NN graph, data with high affinity are connected and therefore form different groups. However, unnecessary cross-group edges are inevitable during graph construction. The cross-group edges can be represented by:

$$e_{i,j} = \left\{ \left(x_i, x_j\right) \middle| x_i \notin \curlywedge, x_j \notin \curlywedge, \exists x_u \in \mathcal{N}_i^k \text{and } x_u \in \curlywedge, s_{i,j} > \min\left(s_{i,u}\right) \right\} \# \tag{5}$$

where $e_{i,j}$ is the edge connecting x_i and x_j, $\curlywedge$ denotes the collection of the border points. To eliminate the latent cross-group edges, we need to find the main part of the formed groups. Similar to density-based spatial clustering of applications with noise (DBSCAN), we first decide on core points and border points using the k-NN density. The connected components of the core points can be regarded as the main part of the formed groups. The main parts of the formed groups are thus obtained by deleting the edges between core points and border points as well as the cross-group edges. Since there are c classes of health conditions, we initialize the first c connected components as c initial groups.

After obtaining the c initial groups, the remaining points, which include points in the left connected components and the border points, are reassigned to one of the initial groups through a density chain (Li and Cai, 2023). To be specific, for each remaining point, there exists the largest density among its neighbours. Therefore, a density chain can be founded through a series of successive data points with increasing k-NN density values. In the density chain, the largest k-NN density may correspond to several data points rather than a unique one. The first arrival point to the largest density is used to determine the group assignment. When the first arrival point belongs to one of the initial groups, the remaining point is assigned to this group. Otherwise, the remaining point is assigned to a group which has the shortest distance to the first arrival point. The group anomaly scores are calculated after obtaining the c final groups, which can be written by:

$$g = \text{abs}\left(\text{skewness}\left(\frac{1}{N}\sum_j s_{i,j}\right)\right) \# \tag{6}$$

2.3 Graph clustering model

The graph clustering model aims at clustering data into different clusters and hence, for each cluster, its label can be decided by the majority of labels inside it. This article follows a rough assumption regarding inaccurate labels, i.e., the number of mislabeled data in the whole dataset is less than 50%. Therefore, data inside a cluster can be relabeled by the cluster label. The mislabeled data are thus fixed. Rather than directly performing k-means clustering after graph convolution, the graph clustering here integrates k-means clustering as a layer into the graph convolution network. The whole graph clustering procedure is a complete forward and backward pass.

In the forward pass, the graph and node features first flow through a two-layer graph convolution module (Kipf and Welling, 2017). Node features are propagated through the graph and then produce the graph embeddings, which can be written as:

$$Z = \sigma\left(\widehat{A}\ \text{ReLU}\left(\widehat{A}XW^{(0)}\right)\ W^{(1)}\right) \# \tag{7}$$

where X is the node features, $\widehat{A}$ is the normalized adjacency matrix, which is calculated by:

$$\widehat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}} \# \tag{8}$$

where $\widetilde{A} = A + I_N$, $\widetilde{D} = diag(\widetilde{d}_i)$, $\widetilde{d}_i = \sum_j \widetilde{a}_{i,j}$. The graph embedding then flows to the differential k-means clustering layer (Wilder et al., 2019). In the differential k-means clustering layer, every data point can be understood as belonging to different clusters simultaneously. The layer needs to calculate responsibilities for the data points to the clusters, which describes the degree of a data point belonging to a cluster. Similar to ordinary k-means clustering, the differential k-means clustering layer contains an assignment step and an update step. Specifically, for any data point x_i, its graph embedding is denoted as z_i. Let μ_p denotes the center of cluster p. In the assignment step, each embedding z_i is given a soft assignment to every cluster center. The soft assignment of z_i to one of the cluster center μ_p are calculated based on distance, which can be expressed by:

$$r_{i,p} = \frac{\exp\left(-\beta\left\|z_i - \mu_p\right\|\right)}{\sum_l \exp\left(-\beta\left\|z_i - \mu_l\right\|\right)}, \forall p = 1, \cdots, P, i = 1, \cdots, N \# \tag{9}$$

where $\|\bullet\|$ denotes any norm, β is a hyperparameter called stiffness. The r_{ip} is also known as the responsibility of cluster p for the embedding z_i. For any embedding z_i, it is obvious that the sum of p responsibilities equals 1, i.e., $\sum_i r_{ip} z_i = 1$. In the update step, the cluster centers are updated by the means of the embeddings that they are responsible for:

$$\mu_p = \frac{\sum_i r_{i,p} z_i}{\sum_i r_{i,p}}, \forall p = 1, \cdots, P \# \tag{10}$$

after several iterations of the assignment step and update step, the differential k-means clustering layer produces the final pair (μ, r).

In the backward pass, the update process is done by backpropagating gradients from the loss function to the cluster that produces the embeddings z. The derivative $\frac{\partial \mu}{\partial z}$ and $\frac{\partial r}{\partial z}$ can be calculated using the implicit function theorem. Hence, by defining a function f,

$$f_q^l(\mu, z) = \mu_q^l - \frac{\sum_i r_{i,p} z_i^l}{\sum_i r_{i,p}}, \forall p = 1, \cdots, P \# \tag{11}$$

then, $\frac{\partial \mu}{\partial z} = -\left[\frac{\partial f(\mu,z)}{\partial \mu}\right]^{-1} \frac{\partial f(\mu,z)}{\partial z}$, $\frac{\partial r}{\partial z}$ can be obtained using chain rule according to Eq. (10). The $\frac{\partial f(\mu,z)}{\partial z}$ and $\frac{\partial f(\mu,z)}{\partial \mu}$ can be calculated as:

$$\begin{cases} \dfrac{\partial f_q}{\partial z_i} = -\dfrac{R_q z_i \left[\dfrac{\partial r_{i,q}}{\partial z_i}\right]^T - C_q \left[\dfrac{\partial r_{i,q}}{\partial z_i}\right]^T}{{R_k}^2} - \dfrac{r_{iq}}{R_q} I \\ \dfrac{\partial f_q}{\partial \mu_p} = \delta_{q,p} I - \dfrac{R_q \sum_{i=1}^N z_i \left[\dfrac{\partial r_{i,q}}{\partial \mu_p}\right]^T - C_q \left[\sum_{i=1}^N \dfrac{\partial r_{i,q}}{\partial \mu_p}\right]^T}{{R_q}^2} \end{cases} \# \tag{12}$$

where $R_q = \sum_{i=1}^N r_{i,q}$, $C_q = \sum_{i=1}^N r_{i,q} z_i$.

The objective function for this optimization problem is based on the concept of modularity, which is defined as:

$$Q(r) = \frac{1}{2m} \sum_{u,v \in V} \sum_{p=1}^{P} \left[A_{u,v} - \frac{d_u d_v}{2m}\right] r_{u,p} r_{v,p} \# \tag{13}$$

where $r_{u,p}$ equals to 1 if node u is assigned to cluster p, otherwise, it takes

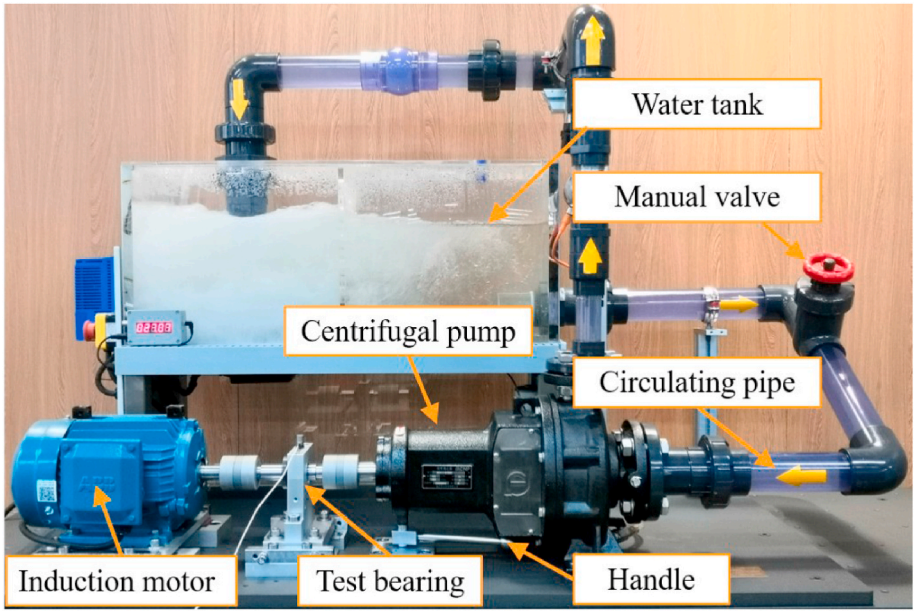

Fig. 3. Pump test rig.

Table 1
Data contamination of the pump dataset.

Health condition	Label	Anomaly setting	Incorrect labeling
Normal condition (NC)	H1	Anomaly→NC	H6→H1
Rotor imbalance (RI)	H2	Anomaly→RI	H5→H2
Rotor misalignment (RM)	H3	Anomaly→RM	H4→H3
Bearing inner race fault (IR)	H4	Anomaly→IR	H3→H4
Bearing outer race fault (OR)	H5	Anomaly→OR	H2→H5
Cavitation fault (CF)	H6	Anomaly→CF	H1→H6

Table 2
Anomalous data settings of the pump dataset.

Sample	Label	Anomaly setting ($\eta = 10\%$)	Anomalous data
0–399	H1	Anomaly→NC	0–19
400–799	H2	Anomaly→RI	400–419
800–1199	H3	Anomaly→RM	800–819
1200–1599	H4	Anomaly→IR	1200–1219
1600–1999	H5	Anomaly→OR	1800–1819
2000–2399	H6	Anomaly→CF	2000–2019

0.

3. Case study

3.1. Case 1: Experiments on a pump

3.1.1. Experimental setup

In this section, the proposed method is evaluated using a pump dataset. Fig. 3 displays the pump test rig, which consists of an induction motor, an external bearing, a centrifugal pump, a water tank and a circulating pipe. Six health conditions are considered for the test centrifugal pump: normal condition, rotor imbalance, rotor misalignment, bearing inner race fault, bearing outer race fault, and cavitation fault. Electrical discharge machining is used to generate sharp trenches to imitate bearing defects. Pulling the handle beneath the centrifugal pump simulates rotor misalignment. The assembly of an asymmetric mass block on the shaft simulates rotor instability. Controlling the manual valve simulates a cavitation fault (Wang et al., 2023b). Each health condition contains 400 samples. The sampling frequency is set at 12,800 Hz.

The settings of data contamination of the pump dataset are demonstrated in Table 1. The health condition and its corresponding label are listed in the first two columns. Anomalous data are collected when the sensor drops from the test pump. As these anomalous data are mixed into the collected data, they are labeled with the given health conditions. As shown in the third column, "Anomaly→NC" refers that the anomalous data mixed into the normal condition with a given probability η (i.e., the label noise ratio). For incorrect labeling, the mislabeled class pairs are listed in the fourth column. The class pair "H6→H1" means that when the ground truth of a sample is H1, we set its label as H6 with a given η.

Incorrect labeling is also known as the noisy label problem. In many existing studies, researchers validated their methods by various noisy label settings, which can be concluded into two categories, i.e., symmetric label noise and asymmetric label noise. Symmetric label noise refers to random noise that may be introduced during expert labeling, simulating random annotation errors by uniformly flipping a certain percentage of the original labels to the remaining labels. Asymmetric label noise, on the other hand, pertains to incorrect annotations occurring between class pairs that are similar to one another. This type of label noise is more prevalent in real-world applications. It can be found that the incorrect label settings in the experimental section all adopt asymmetric label noise. The transition matrix T can be written as:

$$T = \begin{bmatrix} 1-\eta & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} \\ \frac{\eta}{6-1} & 1-\eta & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} \\ \frac{\eta}{6-1} & \frac{\eta}{6-1} & 1-\eta & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} \\ \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & 1-\eta & \frac{\eta}{6-1} & \frac{\eta}{6-1} \\ \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & 1-\eta & \frac{\eta}{6-1} \\ \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & \frac{\eta}{6-1} & 1-\eta \end{bmatrix}$$

3.1.2. Diagnosis results

3.1.2.1. Results of anomalous data detection. This section presents the

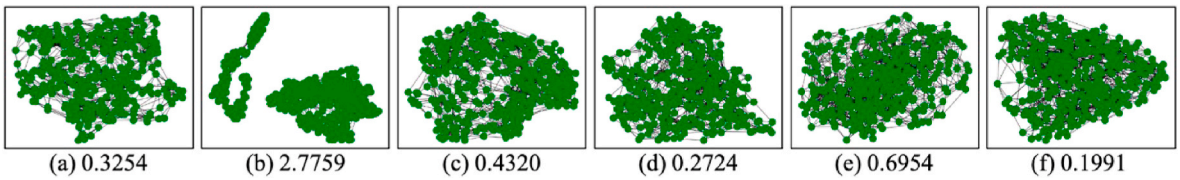

Fig. 4. Group anomaly score of the pump dataset.

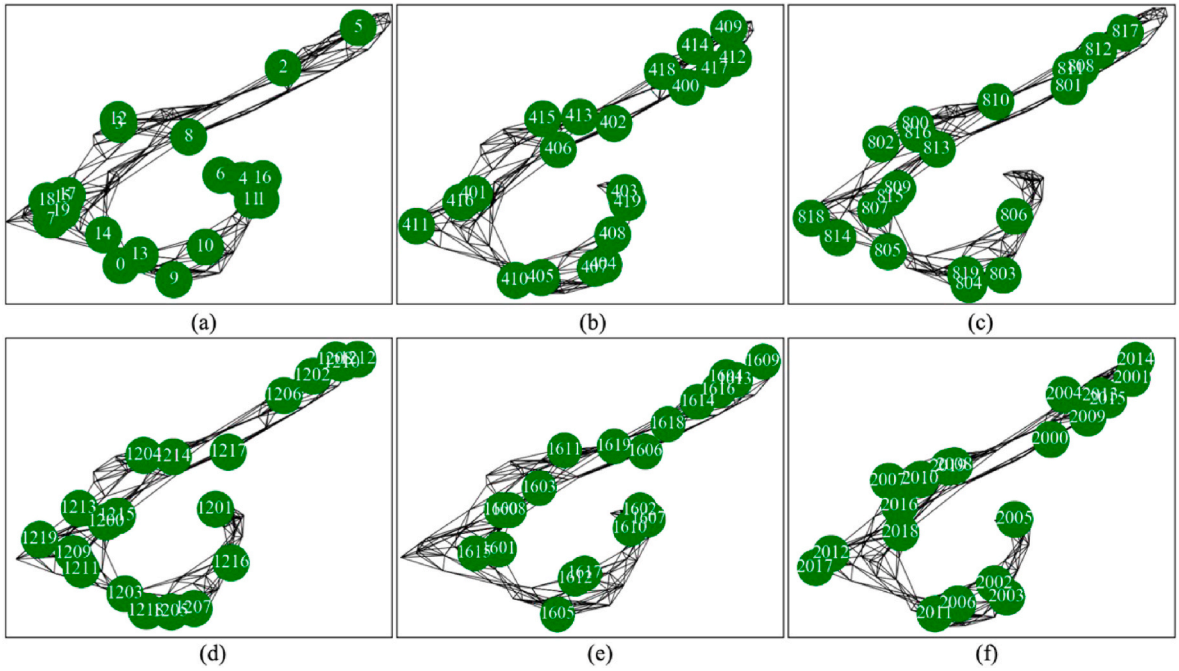

Fig. 5. The detected anomalous data of the pump dataset in (a) H1 (b) H2 (c) H3 (d) H4 (e) H5 (f) H6.

Table 3
Incorrect labeling settings of the pump dataset.

Sample	Label	Incorrect labeling ($\eta = 10\%$)	Mislabeled data
0–399	H1	H6→H1	380–399
400–799	H2	H5→H2	780–799
800–1199	H3	H4→H3	1180–1199
1200–1599	H4	H3→H4	1580–1599
1600–1999	H5	H2→H5	1980–1999
2000–2399	H6	H1→H6	2380–2399

detection results of anomalous data. Table 2 displays the anomalous data settings of the pump dataset. The settings of anomalous data are listed in the third column. For example, when the label of samples 0–399 is H1, the anomalous data inside the samples are 0–19. We can check the detection results by referring to the settings shown in Table 2.

In the first stage, the pump dataset is divided into different groups based on *k*-NN graphs. Group anomaly scores are then calculated for the resulting groups. Anomalous data exists in the group whose score exceeds a threshold. As a general guideline, a skewness value beyond the range of [−2, 2] is considered indicative of substantial non-normality (Hair et al., 2022). Therefore, the threshold for the group anomaly score is set at 2. Fig. 4 illustrates the formed groups and their corresponding group anomaly scores. The second group, with a score of 2.7759, is identified as the abnormal group. In Fig. 4, there are two connected components in the abnormal group, where anomalous data existed in one of them. We can analyze the anomaly group by visualizing the data within it. Fig. 5 reveals the left-side component of the abnormal group. According to the settings in Table 2, the anomalous data in H1 are 0–19, in H2 are 400–419, in H3 are 800–819, in H4 are 1200–1219, in H5 are 1800–1819 and in H6 are 2000–2019. As demonstrated in Fig. 5 (a), the detected anomalous data in H1 are 0–19, which aligns with the settings in Table 2. The detected anomalous data in H2–H6 can be found in Fig. 5(b)–(f).

3.1.2.2. Results of mislabeled data detection. This section presents the detection results of mislabeled data. Table 3 provides the incorrect labeling settings for the pump dataset. The settings of incorrect labeling are listed in the third column. For example, when the label of samples 0–399 is H1, the mislabeled data inside the samples are 380–399. According to the third column, the ground truth of samples 380–399 is H6.

Upon detecting the existence of anomalous data, this stage groups the pump dataset into different clusters using the graph clustering model. After clustering, samples of each health condition are relabeled. According to the settings in Table 1, we know that the label for NC is H1, the label for RI is H2, the label for RM is H3, the label for IR is H4, the label for OR is H5 and the label for CF is H6. Therefore, after the label correction (stage 2), all NC samples are supposed to be relabeled with H1, all RI samples are supposed to be relabeled with H2, all RM samples are supposed to be relabeled with H3, all IR samples are supposed to be relabeled with H4, all OR samples are supposed to be relabeled with H5, and all CF samples are supposed to be relabeled with H6. Fig. 6 displays the detected mislabeled data for different health conditions. In Fig. 6, the detected mislabeled data are marked in red, while others are marked in blue. According to Table 3, samples of NC are (0–379, 2380–2399), samples of RI are (400–779, 1980–1999), samples of RM are (800–1179, 1580–1599), samples of IR are (1200–1579, 1180–1199), samples of OR are (1600–1979,780-799) and samples of CF are (2000–2379, 380–399).

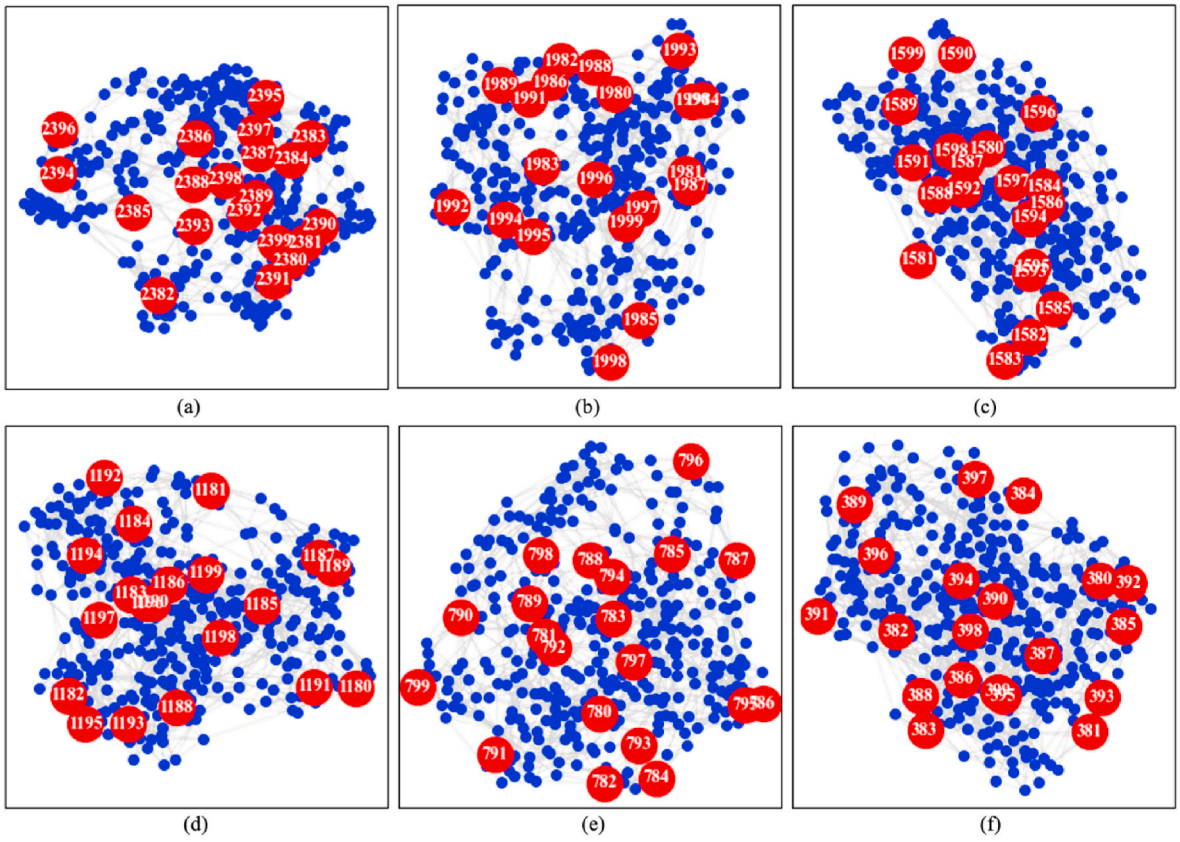

Fig. 6. The detected mislabeled data of the pump dataset in (a) H1 (b) H2 (c) H3 (d) H4 (e) H5 (f) H6.

Table 4
Parameter set of the CNN.

Module	Layers	Parameter size	Output shape	Activation function
Feature encoder	1-D Conv	16@5*1	16×1020	/
	1-D BatchNorm	16	16×1020	ReLU
	1-D MaxPool	4	16×255	/
	1-D Conv	32@5*1	32×248	/
	1-D BatchNorm	32	32×248	ReLU
	1-D MaxPool	4	32×62	/
	1-D Conv	16@5*1	16×58	/
	1-D BatchNorm	16	16×58	ReLU
	1-D MaxPool	2	16×29	/
Classifier	Flatten	/	464	/
	FC	464×256	256	ReLU
	FC	256×C	C	Softmax

As illustrated in Fig. 6(a), the detected mislabeled data in H1 is 2380–2399, which is consistent with the settings mentioned above. The detected mislabeled data in H2–H6 can be found in Fig. 6(b)–(f).

3.1.3. Analysis of data contamination impact

In this section, we analyze the impact of data contamination on DL models' performance. We use the widely adopted CNN as the basic diagnostic model. The model's detailed parameters can be found in Table 4. It comprises three convolutional blocks and two fully connected layers. Each convolutional block consists of a convolutional layer with batch normalization and a pooling layer. The convolutional kernel size is 5×1, and the number of channels for these convolutions is 16, 32, and 16, respectively. The two fully connected layers are employed for classification. The diagnostic model is trained with noise ratios ranging from 10% to 40%.

Fig. 7 (a) presents the training curves of the pump dataset under noisy labels. To provide a better understanding, the accuracies in the curves are calculated based on the ground truth, not the noisy labels. In Fig. 7 (a), the training curves under different noise ratios exhibit a similar pattern: initially increasing to high accuracy, then dropping from that point, and finally stabilizing at a certain value. The increase is because clean data are easier to fit than those with label noise (Arpit et al., 2017). The subsequent drop signifies that the model begins to fit label noise after learning the pattern from clean data. The stabilizing value of the accuracy curve is related to the noise ratio η. For instance, when η equals 10%, the value stabilizes at 90%. When η increases to 20%, the value drops from 90% to approximately 80%. Likewise, as the noise ratio increases to 30% and 40%, the accuracy decreases to 70% and 60%. It can be observed that the value remains around the percentage of clean labels, indicating that the diagnostic model will gradually overfit the mislabeled data during the training process. This overfitting can confuse the diagnostic model, making it challenging to distinguish real fault types from mislabeled ones.

Fig. 7 (b) displays the training curves of the pump dataset under anomalous data. As the training process progresses, the diagnostic model eventually converges. The convergence speed varies with different noise ratios; with larger label noise ratios, the convergence speed becomes slower. Moreover, when comparing the training curves between Fig. 7 (a) and (b), it is evident that anomaly leads to greater training time and resource consumption. For example, when the noise ratio is 40%, as shown in Fig. 7 (b), the training curve converges at 3000 epochs, whereas the same level ratio in Fig. 7 (a) takes only 250 epochs to converge.

Fig. 8 presents the training curves of the pump dataset under both noisy labels and anomalous data, where each contributes equally to the noise ratio. The training curves exhibit two typical characteristics: 1) the curves stabilize at a certain value related to the noise ratio of mislabeled

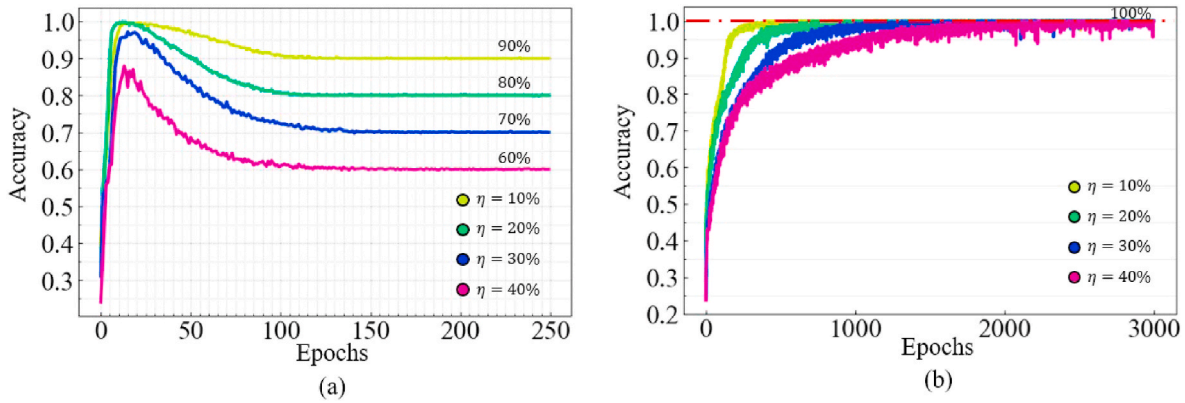

Fig. 7. Training curve of the pump dataset under (a) noisy labels (b) anomalous data.

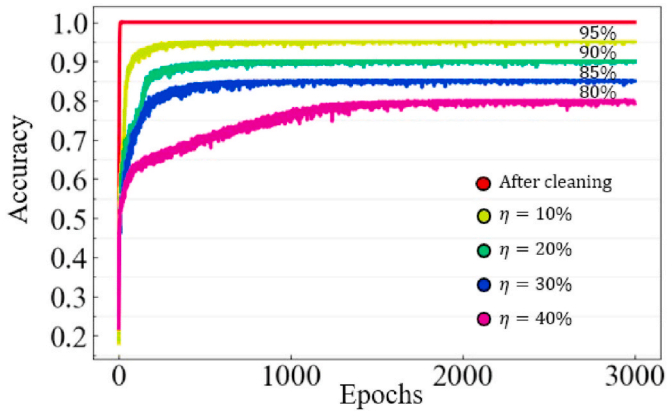

Fig. 8. Training curve of the pump dataset under the noisy labels and anomalous data.

Table 5
Average accuracy of different methods of the pump dataset.

Methods	$\eta = 10\%$	$\eta = 20\%$	$\eta = 30\%$	$\eta = 40\%$
GCE	0.9767	0.9621	0.9121	0.8342
SCE	0.9804	0.9550	0.9108	0.8771
Forwards	0.9729	0.9571	0.9012	0.8554
Backwards	0.9679	0.9346	0.9033	0.8413
CE before data cleaning	0. 9446	0.9008	0.8492	0.7921
CE after data cleaning	1	1	1	1

data; 2) the curves converge to that stable value with a large number of epochs. The two characteristics coincide with phenomena manifested in Fig. 7 (a) and Fig. 7 (b). The red line in Fig. 8 is the training curve after data cleaning. It can be seen that the performance of the model is significantly improved compared to other curves trained under noise ratios. Meanwhile, the time efficiency of training DL models is also significantly improved.

3.1.4. Comparison study

This section presents comparison studies with existing baselines, such as the symmetric cross-entropy loss (SCE) (Wang and Li, 2022), generalized cross-entropy loss (GCE) (Liang et al., 2022), forward cross-entropy loss (Forwards) (Nie and Xie, 2021a), and backward cross-entropy loss (Backwards) (Nie and Xie, 2021b). The basic diagnostic model is the previously mentioned CNN. Table 5 displays the experimental results for the baseline methods. CE denotes the basic diagnostic model that learns with normal cross-entropy loss. These methods achieve higher average accuracies on the pump dataset than CE under various noise ratios. But it is hard to determine which method performs better. Moreover, as η increases, the classification accuracy declines, which indicates the baseline methods are still influenced by noisy labels. In contrast to the baseline methods, the proposed method eliminates noisy labels from the outset. Consequently, the diagnosis models are supervised under clean labels, which significantly prompts the model's performance.

3.2. Case 2: Experiments on a robot

3.2.1. Experimental setup

In this section, the proposed method is evaluated using the robot dataset. Fig. 9 displays the industrial robot test rig, which consists of a six-axis industrial robot and a servo controller. Faults are introduced into the RV reducer of the second joint, as indicated by the enlarged view in Fig. 9. Four health conditions are considered for the test planetary gearbox of the RV reducer: normal condition, wear in gear tooth, pitting in gear tooth and crack in gear tooth. The simulated faults are shown in Fig. 10. In the experiment, four motion paths are set, including the reciprocating motion of the second joint alone, the reciprocating motion of the first and second joints, the reciprocating motion of the second and third joints, and the reciprocating motion of all six joints. Each joint moves within the closed range of its limit position. To simulate the end load, a mass block is added at the robot's end-effector, with the load set to 0/10 kg. The motion speed for each joint is set at its maximum. Vibration sensors are installed at the second joint of the industrial robot. The sampling frequency is 6250 Hz.

The experiment is designed to simulate possible faults that may occur in the planetary gearbox of the RV reducer. However, the collected data contains anomalous data due to damage to the test line. The anomalous data are mixed into the dataset of the given health conditions. For incorrect labeling, the mislabeled class pairs are set as H1→H4, H2→H3, H3→H2, and H4→H1. Table 6 presents the settings of data contamination in the robot dataset. The transition matrix of incorrect labeling in this section can be written as:

$$T = \begin{bmatrix} 1-\eta & \frac{\eta}{4-1} & \frac{\eta}{4-1} & \frac{\eta}{4-1} \\ \frac{\eta}{4-1} & 1-\eta & \frac{\eta}{4-1} & \frac{\eta}{4-1} \\ \frac{\eta}{4-1} & \frac{\eta}{4-1} & 1-\eta & \frac{\eta}{4-1} \\ \frac{\eta}{4-1} & \frac{\eta}{4-1} & \frac{\eta}{4-1} & 1-\eta \end{bmatrix}$$

3.2.2. Diagnosis results

3.2.2.1. Results of anomalous data detection. This section presents the

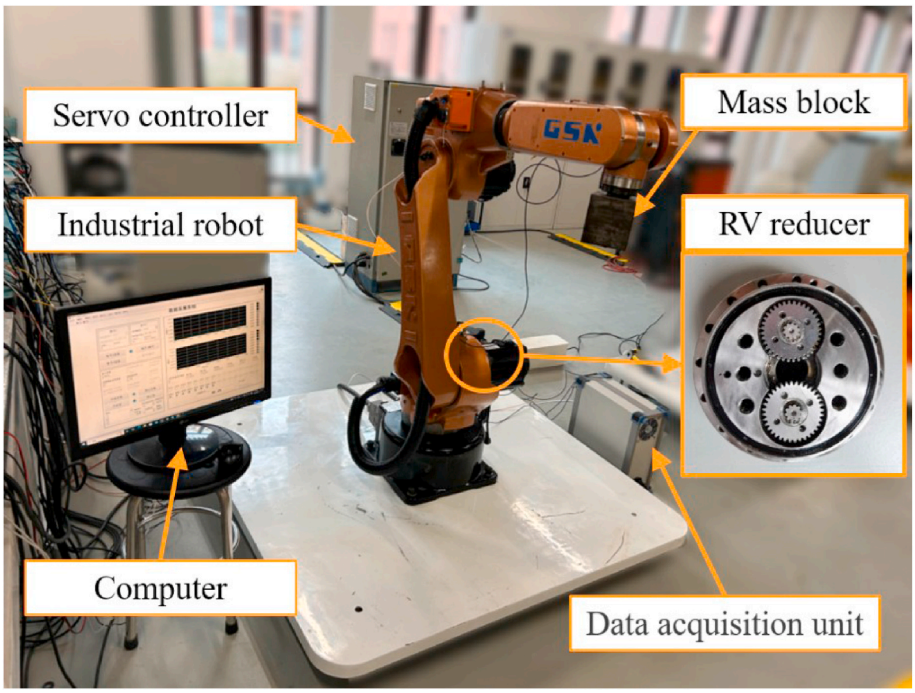

Fig. 9. Robot test rig.

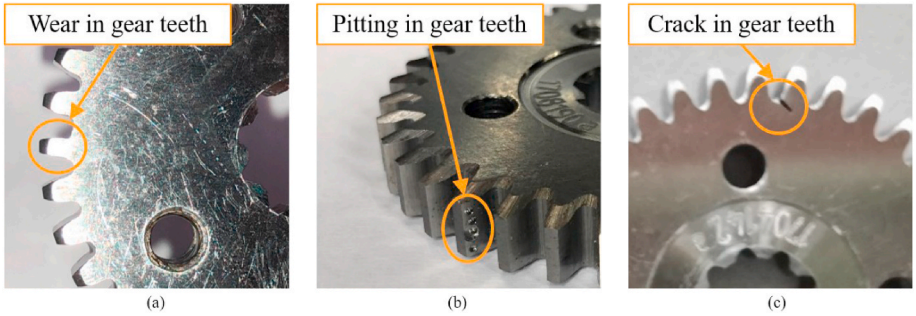

Fig. 10. Simulated faults of the robot dataset.

Table 6
Data contamination of the robot dataset.

Health condition	Label	Anomaly setting	Incorrect labeling
Normal condition (NC)	H1	Anomaly→NC	H1→H4
Wear in gear teeth (WGT)	H2	Anomaly→WGT	H2→H3
Pitting in gear teeth (PGT)	H3	Anomaly→PGT	H3→H2
Crack in gear teeth (CGT)	H4	Anomaly→CGT	H4→H1

Table 7
Anomalous data settings of the robot dataset.

Sample	Label	Anomaly setting ($\eta = 40\%$)	Anomalous data
0–99	H1	Anomaly→NC	0–19
100–199	H2	Anomaly→WGT	100–119
200–299	H3	Anomaly→PGT	200–219
300–399	H4	Anomaly→CGT	300–319

detection results of anomalous data. Table 7 displays the anomalous data settings of the robot dataset. The settings of anomalous data are listed in the third column. For example, when the label of samples 0–99 is H1, the anomalous data inside the samples are 0–19. We can check the detection results by referring to the settings shown in Table 7.

In the first stage, the robot dataset is divided into different groups based on k-NN graphs. Group anomaly scores are then calculated for the resulting groups. As depicted in Fig. 11, the dataset is divided into four groups based on k-NN graphs, with their group anomaly scores provided below them. The scores for the four groups are 3.6109, 1.8920, 0.2185, and 0.0954. It is determined that anomalous data are present in the first group, which has a group anomaly score of 3.6109. We can analyze the first group by visualizing the data within it. Fig. 12 visualizes the detected results. According to the settings in Table 6, the anomalous data in H1 are 0–19, in H2 are 100–119, in H3 are 200–219 and in H4 are 300–319. The detected results in H1 are 0–19 (Fig. 12 (a)), in H2 are 100–119 (Fig. 12 (b)), in H3 are 200–219 (Fig. 12 (c)) and in H4 are 300–319 (Fig. 12 (d)). The results in Fig. 12 demonstrate that the anomalous data in the robot dataset can be effectively detected in this stage.

3.2.2.2. Results of mislabeled data detection. This section presents the detection results of mislabeled data. Table 8 provides the incorrect labeling settings for the robot dataset. The settings of incorrect labeling are listed in the third column. For example, when the label of samples 0–99 is H1, the mislabeled data inside the samples are 80–99. According to the third column, the ground truth of samples 80–99 is H4.

Upon detecting the existence of anomalous data, this stage groups

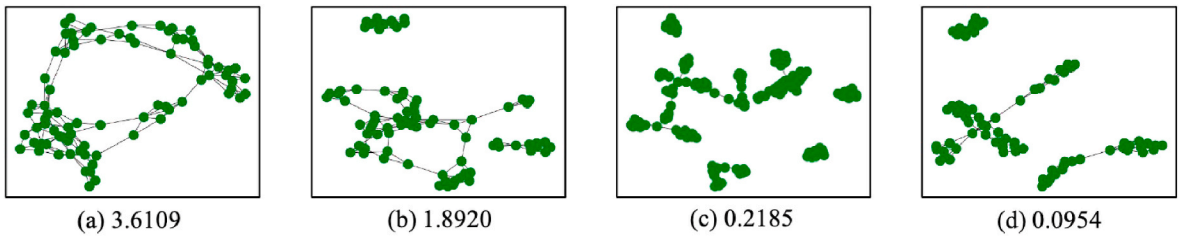

Fig. 11. Group anomaly score of the robot dataset.

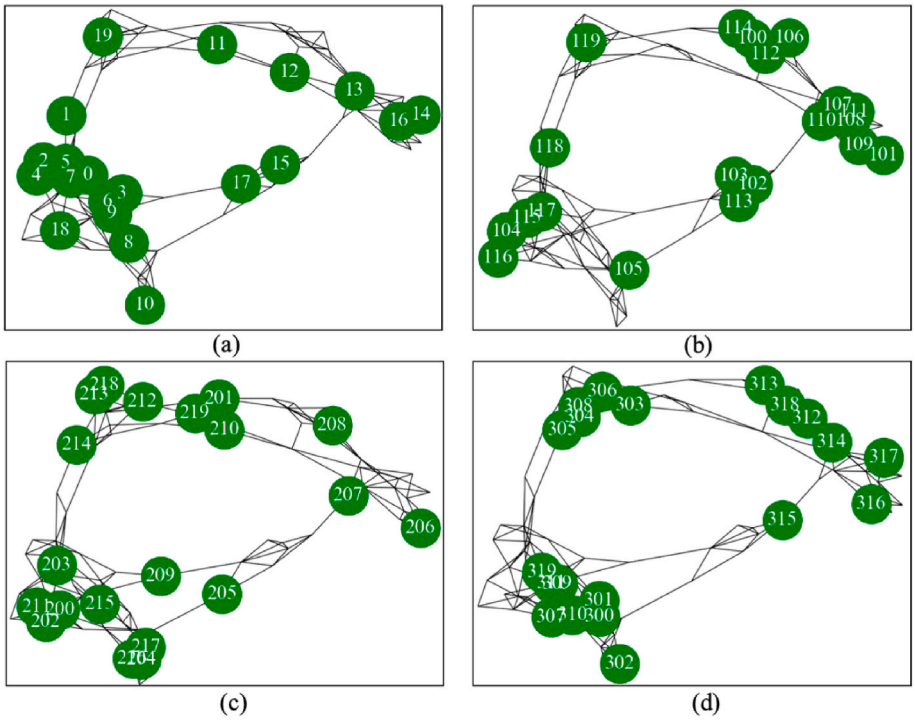

Fig. 12. The detected anomalous data of the robot dataset in (a) H1 (b) H2 (c) H3 (d) H4.

Table 8
Incorrect labeling settings of the robot dataset.

Sample	Label	Incorrect labeling ($\eta = 40\%$)	Mislabeled data
0–99	H1	H4→H1	80–99
100–199	H2	H3→H2	180–199
200–299	H3	H2→H3	280–299
300–399	H4	H1→H4	380–399

the robot dataset into different clusters using the graph clustering model. After clustering, samples of each health condition are relabeled.

We know that the label for NC is H1, the label for WGT is H2, the label for PGT is H3, and the label for CGT is H4 based on the settings in Table 6. As a result, all NC samples are supposed to be relabeled with H1, all WGT samples are supposed to be relabeled with H2, all PGT samples are supposed to be relabeled with H3 and all CGT samples are supposed to be relabeled with H4 after label correction. Fig. 13 displays the detected mislabeled data for different health conditions. In Fig. 13, the detected mislabeled data are marked in red, while others are marked in blue. According to Table 8, samples of NC are (0–79, 380–399), samples of WGT are (100–179, 280–299), samples of PGT are (200–279, 180–199) and samples of CGT are (300–379, 80–99). The detected mislabeled data in H1 are 380–399 (Fig. 13 (a)), in H2 are 280–299 (Fig. 13 (b)), in H3 are 180–199 (Fig. 13 (c)), and in H4 are 80–99 (Fig. 13 (d)). The results in Fig. 13 demonstrate that the mislabeled data in the robot dataset can be effectively detected during this stage.

3.2.3. Analysis of data contamination impact

This section analyzes the influence of data contamination on the robot dataset. Fig. 14 (a) presents the training curves under noisy labels. In Fig. 14 (a), the training curves initially increase, then decrease, and ultimately stabilize at a certain value, which approximates the percentage of noisy labels. For instance, when the accuracy curve is subject to a 10% noise ratio, the value stabilizes at 90%. As the noise ratio increases to 20%, the value drops from 90% to around 80%. Similarly, when the noise ratio rises to 30% and 40%, the accuracy declines to 70% and 60%, respectively.

Fig. 14 (b) illustrates the training curves under anomalous data. As the training process progresses, the diagnostic model eventually achieves 100% accuracy. However, the presence of anomalous data slows down the convergence speed. For example, when the noise ratio is 40%, as depicted in Fig. 14 (b), the training curve converges at 5000 epochs, whereas the same level ratio in Fig. 14 (a) takes only 250 epochs.

Fig. 15 presents the training curves of the robot dataset under both noisy labels and anomalous data, where each contributes equally to the noise ratio. Similar to the previous case study, the training curves exhibit characteristics as: 1) the curves stabilize at a certain value related to the noise ratio of mislabeled data; 2) the curves converge to that stable value with a large number of epochs. The red line in Fig. 15 is the training curve after data cleaning. It can be seen that the performance of the

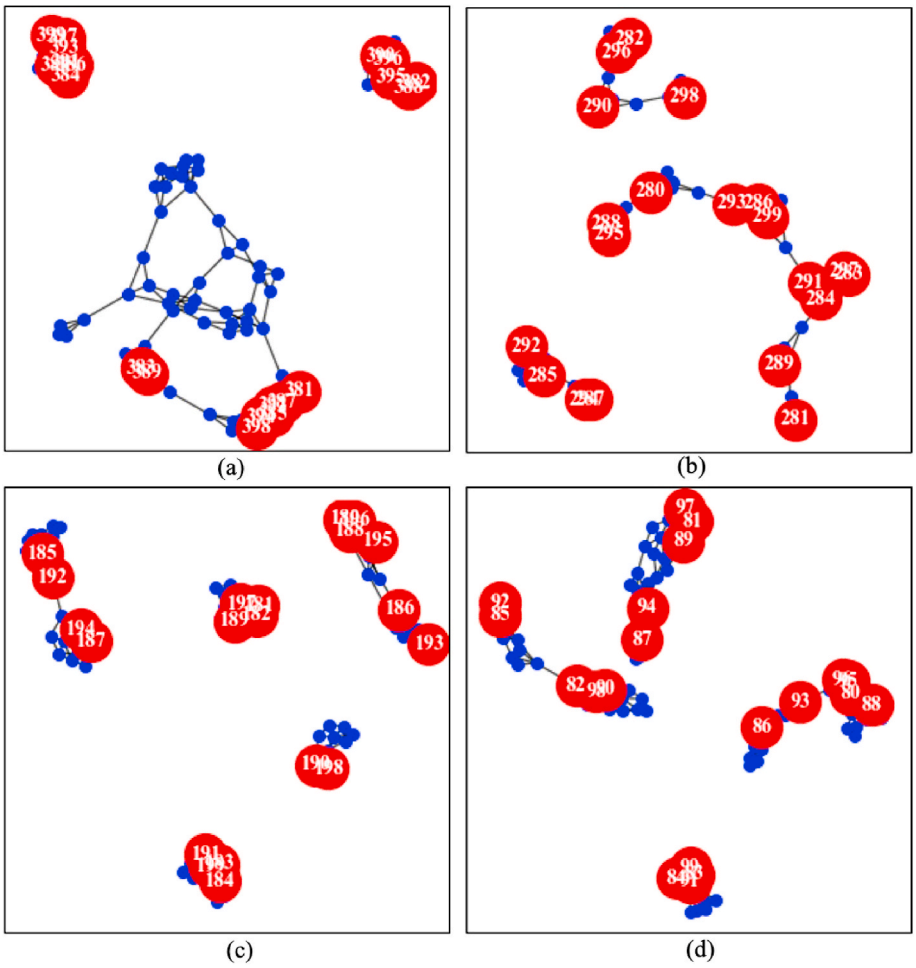

Fig. 13. The detected mislabeled data of the robot dataset in (a) H1 (b) H2 (c) H3 (d) H4.

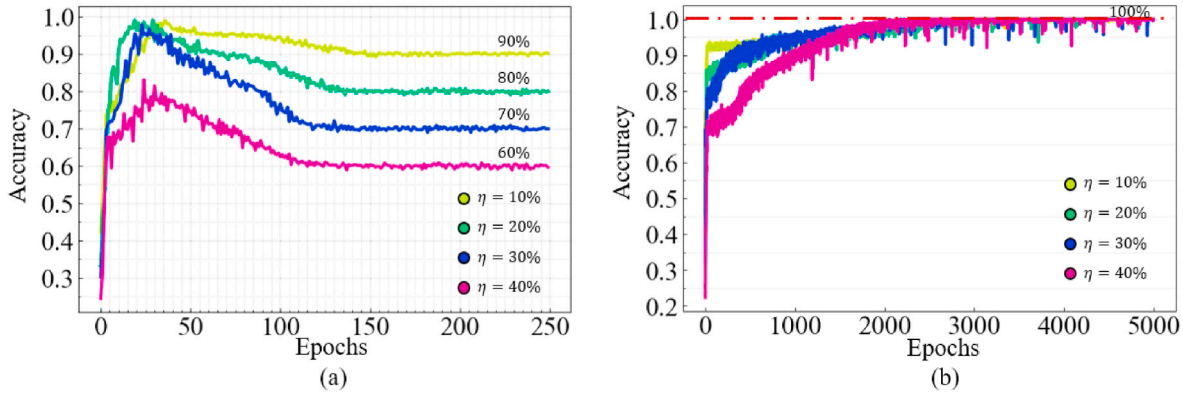

Fig. 14. Training curve of the robot dataset under (a) noisy labels (b) anomalous data.

model is significantly improved as well as the time efficiency after data cleaning.

3.2.4. Comparison study

In this section, comparison studies are conducted with baseline methods. Table 9 presents the results of SCE, GCE, Forward, Backward, and CE on the robot dataset. The baseline methods do not show a significant advantage over CE on the robot dataset. The results from SCE, Forward, and Backward are similar to each other. In comparison, GCE achieves slightly higher average accuracies when η is 20% and 40%. However, the overall performance of the robot dataset using the baseline methods is not satisfactory. By applying the proposed method, DL models are trained with high-quality data, which significantly improves the performance.

4. Conclusion

Data contamination has become a significant obstacle in training reliable DL models, as DL models tend to overfit noisy labels and anomalous data. To address data contamination that occurs in the fault

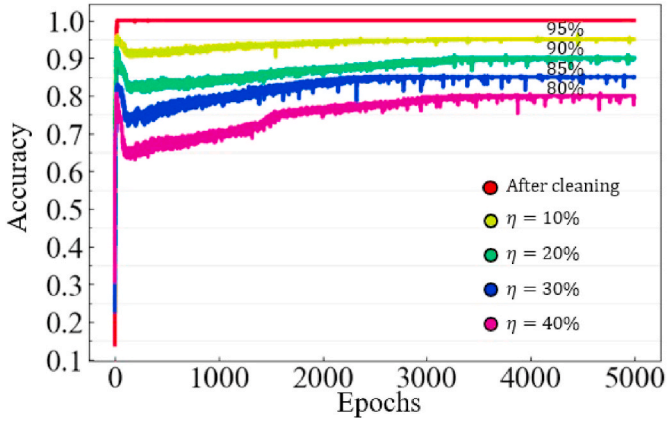

Fig. 15. Training curve of the robot dataset under the noisy label and anomalous data.

Table 9
Average accuracy of different methods of the robot dataset.

Methods	$\eta = 10\%$	$\eta = 20\%$	$\eta = 30\%$	$\eta = 40\%$
GCE	0.9351	0.9135	0.8389	0.8077
SCE	0.9038	0.8197	0.7620	0.6659
Forwards	0.9087	0.8173	0.7332	0.6755
Backwards	0.9087	0.8293	0.7380	0.6635
CE before data cleaning	0.9519	0.9038	0.8534	0.8005
CE after data cleaning	1	1	1	1

diagnosis field, a graph neural network-based data-cleaning method is proposed. The method contains two stages. In the first stage, the existence of anomalous data is detected by comparing group anomaly scores with the threshold. In the second stage, incorrect labeling is corrected through graph clustering. Two case studies are conducted to evaluate the performance of the method. In the pump dataset case, existing methods demonstrate significant accuracy improvement, even though their performance declines as the noise ratio increases. In contrast, existing methods show limited accuracy improvement or even worse classification results in the robot dataset case. Compared to existing methods, the proposed method prepares high-quality data for training deep diagnostic models, eliminating the negative impact of data contamination from the outset.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This research was supported by the National Key R&D Program of China (2022YFB3402100), the National Science Fund for Distinguished Young Scholars of China (52025056), the National Natural Science Foundation of China (52305129), China Postdoctoral Science Foundation (2023M732789), Shaanxi Science and Technology Innovation Team (2023-CX-TD-15), SanQin Scholar Innovation Team, and Open Foundation of State Key Laboratory of Compressor Technology (Compressor Technology Laboratory of Anhui Province, SKL-YSJ202104).

References

Arpit, D., Jastrzkbski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., 2017. A closer look at memorization in deep networks. In: Proc. 34th Int. Conf. Mach. Learn., pp. 233–242.
Ghorvei, M., Kavianpour, M., Beheshti, M., Ramezani, A., 2023. Spatial graph convolutional neural network via structured subdomain adaptation and domain adversarial learning for bearing fault diagnosis. Neurocomputing 517, 44–61.
Guo, L., Yu, Y., Qian, M., Zhang, R., Gao, H., Cheng, Z., 2022. FedRUL: a new federated learning method for edge-cloud collaboration based remaining useful life prediction of machines. IEEE/ASME Trans. Mech. 28, 350–359.
Hair, J., Hult, G., Ringle, C., Saratedt, M., 2022. A Primer on Partial Least Squares Structural Equation Modeling. Sage Publications, Thousand Oaks, CA, p. 66, 2022.
Kavianpour, M., Ramezani, A., Beheshti, M., 2022. A class alignment method based on graph convolution neural network for bearing fault diagnosis in presence of missing data and changing working conditions. Measurement 199, 111536.
Kipf, T., Welling, M., 2017. Semi-supervised Classification with Graph Convolutional Networks arXiv preprint arXiv:1609.02907.
Lei, Y., Jia, F., Lin, J., Xing, S., Ding, D., 2016. An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. IEEE Trans. Ind. Electron. 63, 3137–3147.
Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., Nandi, A.K., 2020. Applications of machine learning to machine fault diagnosis: a review and roadmap. Mech. Syst. Signal Process. 138, 106587.
Lei, Y., Li, N., Li, X., 2022. Big Data-Driven Intelligent Fault Diagnosis and Prognosis for Mechanical Systems. Springer Nature.
Li, R., Cai, Z., 2023. A clustering algorithm based on density decreased chain for data with arbitrary shapes and densities. Appl. Intell. 53, 2098–2109.
Li, K., Gao, X., Jia, X., Xue, B., Fu, S., Liu, Z., Huang, X., Huang, Z., 2022. Detection of local and clustered outliers based on the density–distance decision graph. Eng. Appl. Artif. Intell. 110, 104719.
Liang, P., Wang, W., Yuan, X., Liu, S., Zhang, L., Cheng, Y., 2022. Intelligent fault diagnosis of rolling bearing based on wavelet transform and improved ResNet under noisy labels and environment. Eng. Appl. Artif. Intell. 115, 105269.
Long, H., Xu, S., Gu, W., 2022. An abnormal wind turbine data cleaning algorithm based on color space conversion and image feature detection. Appl. Energy 311, 118594.
Meire, M., Van Baelen, Q., Ooijevaar, T., Karsmakers, P., 2023. Constraint guided autoencoders to enforce a predefined threshold on anomaly scores: an application in machine condition monitoring. J. Dynam. Monit. Diagn. 2, 144–154.
Nie, X., Xie, G., 2021a. A novel normalized recurrent neural network for fault diagnosis with noisy labels. J. Intell. Manuf. 32, 1271–1288.
Nie, X., Xie, G., 2021b. A fault diagnosis framework insensitive to noisy labels based on recurrent neural network. IEEE Sensor. J. 21, 2676–2686.
Song, H., Kim, M., Park, D., Shin, Y., Lee, J., 2022. Learning from noisy labels with deep neural networks: a survey. IEEE Transact. Neural Networks Learn. Syst. https://doi.org/10.1109/TNNLS.2022.3152527.
Wang, H., Li, Y., 2022. Iterative error self-correction for robust fault diagnosis of mechanical equipment with noisy label. IEEE Trans. Instrum. Meas. 71, 1–13.
Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J., 2019. Symmetric cross entropy for robust learning with noisy labels. Proc. Int. Conf. Comput. Vis. 322–330.
Wang, P., Shen, C., Chen, B., Shi, J., Huang, W., Zhu, Z., 2023a. Generic meta-transfer learning model with special neuronal processing parameters for few-shot fault bearing diagnosis. J. Adv. Manuf. Sci. Technol. 3, 2023007.
Wang, S., Lei, Y., Lu, N., Li, X., Yang, B., 2023b. A multi-sensor relation model for recognizing and localizing faults of machines based on network analysis. Front. Mech. Eng. 18, 1–15.
Wilder, B., Ewing, E., Dilkina, B., Tambe, M., 2019. End-to-end learning and optimization on graphs. In: Proc. 33rd Int. Conf. Neural Inf. Process. Syst., pp. 4674–4685.
Xu, X., Lei, Y., Li, Z., 2019. An incorrect data detection method for big data cleaning of machinery condition monitoring. IEEE Trans. Ind. Electron. 67, 2326–2336.
Yang, B., Lei, Y., Li, X., Roberts, C., 2023. Deep targeted transfer learning along designable adaptation trajectory for fault diagnosis across different machines. IEEE Trans. Ind. Electron. 70, 9463–9473.
Yao, Q., Zhu, H., Xiang, L., Su, H., Hu, A., 2023. A novel composed method of cleaning anomy data for improving state prediction of wind turbine. Renew. Energy 204, 131–140.
Yu, Y., Guo, L., Gao, H., He, Y., You, Z., Duan, A., 2023. FedCAE: a new federated learning framework for edge-cloud collaboration based machine fault diagnosis. IEEE Trans. Ind. Electron. https://doi.org/10.1109/TIE.2023.3273272.