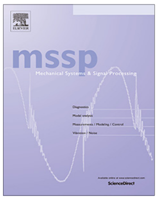
Mechanical Systems and Signal Processing 150 (2021) 107233

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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



A new deep auto-encoder method with fusing discriminant information for bearing fault diagnosis

Wentao Mao [a,b,*], Wushi Feng [a], Yamin Liu [a], Di Zhang [a], Xihui Liang [c]

[a] School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China
[b] Engineering Lab of Intelligence Business & Internet of Things, Henan Province, Xinxiang 453007, China
[c] Department of Mechanical Engineering, University of Manitoba, Winnipeg R3T 5V6, Canada

ARTICLE INFO

Article history:
Received 12 May 2019
Received in revised form 10 August 2020
Accepted 18 August 2020
Available online 4 September 2020

Keywords:
Bearing fault diagnosis
Auto-encoder
Deep learning
Structural information
Discriminant information

ABSTRACT

In recent years, deep learning techniques have been proved a promising tool for bearing fault diagnosis. However, to extract deep features with better representative ability, how to introduce discriminant information about different fault types into the deep learning model is still challenging. Moreover, as deep learning techniques heavily rely on mass of measuring data, relatively small amounts of data may cause over-fitting and reduce model stability as well. To solve such problems, a new deep auto-encoder method with fusing discriminant information about multiple fault types is proposed for bearing fault diagnosis. First, a new loss function is designed by introducing structural discriminant information. Specifically, to improve the feature's representative ability, a new discriminant regularizer is designed in the loss function by using maximum correlation entropy. And to represent the structural information among multiple fault types, a relation matrix for fault types is introduced, then a new regularizer with a symmetric constraint on this matrix is constructed. Second, a gradient descent method is provided to optimise this loss function, and the optimal deep features, as well as fault relatedness, are learned simultaneously. Experimental results on CWRU and IMS bearing data sets show that, compared to several state-of-the-art diagnosis methods, the proposed method can effectively improve the diagnostic accuracy with acceptable time efficiency. And the results on the Kruskal–Wallis Test indicate the proposed method has better numerical stability.

1. Introduction

Rolling bearings are key support element of rotating machinery. Under complex working conditions like a heavy load, strong impact and high rotational speed, rolling bearings are inclined to fail. The failures can be classified into multiple failure states [1]. The failure of a bearing may deteriorate other parts even the whole machinery. Therefore, it is of great theoretical and engineering significance to improve the accuracy and stability of bearings fault diagnosis. In recent years, deep learning-based condition monitoring and health management techniques have received more and more attention [2–5]. A deep learning model is a kind of neural networks which have deep architecture. Because of many advantages such as adjustable structure and adaptive exploration of domain information, deep learning techniques can extract features with better representative ability than the traditional hand-crafted features which are generally extracted from time/frequency/

* Corresponding author at: School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China.
E-mail address: maowt@htu.edu.cn (W. Mao).

https://doi.org/10.1016/j.ymssp.2020.107233
0888-3270/© 2020 Elsevier Ltd. All rights reserved.

time–frequency domains. Some typical deep learning models, e.g., convolutional neural network (CNN) [3], deep belief network (DBN) [4], and deep auto-encoder (DAE) [5] have been successfully applied to bearing fault diagnosis problem. Please refer to Section 2 for more details.

The training a deep neural network heavily relies on a mass of data to achieve good diagnosis performance. If training data are insufficient, the extracted deep features are easy to cause the over-fitting phenomenon. As a result, the diagnosis accuracy and stability will be reduced [6]. However, in many real engineering applications, due to the limitation of measuring device and complex working condition, it may be hard to accumulate enough condition monitoring data for training an effective deep learning model. Therefore, when the amount of training data is relatively insufficient, how to improve the extracted deep features' representative ability as well as deep model's stability becomes a challenging task in data-driven intelligent fault diagnosis, especially the diagnosis of multiple fault types.

Many studies have demonstrated that, in the case of insufficient data, adding more prior information into a machine learning model is helpful to improve the model's generalization performance and stability as well as reduce the over-fitting [6]. To achieve effective bearing fault diagnosis, besides the domain knowledge about fault information contained in raw vibration signals, the output information about the diagnosis model also needs to be considered. The output information contains not only discriminant information about fault classification but also structural relatedness among multiple fault types. Discriminant information is explicit and can be calculated directly. But structural relatedness information is implicit. If we can exploit such relatedness information and integrate them into a deep learning model, the representative ability of extracted features and the model's stability can both be improved. In this paper, we choose DAE as a study object as DAE is widely used to extract fault features from vibration signal due to its simple structure and good representation performance for raw signals. However, as DAE is a kind of unsupervised learning algorithm, it usually needs extra supervised fine-tuning beyond unsupervised pre-training to introduce discriminant information about model output. There is no channel of fusing such information directly in the training process. In addition, although the unsupervised learning mode of DAE can grab the feature representation from raw signals, it is still poor to exploit the inner structural relationship among data which is very important for feature extraction [7,8]. Based on the above analysis, we think the key factors of improving DAE features' representative performance and model stability with insufficient training data are: 1) Effective fusion of discriminant information in a DAE model, and 2) Self-adaptive exploitation of structural information among multiple fault conditions and integrating them into the DAE training process.

Quite different from current researches that adopt unsupervised learning mode, we plan to introduce structural information among various outputs into a DAE model. To utilize such information effectively, we first need to propose a formalized description of the structural relationship among fault types. In our previous work [9], we introduced an output kernel learning (OKL) algorithm [10] to conduct fault diagnosis. That work proved that the intrinsic relatedness among multiple fault conditions could be described by an output kernel with a symmetric structure and the relatedness information is helpful to improve diagnosis accuracy and numerical stability. Then it inspires us a new idea to further solve the above problem. In Ref. [9], the OKL model is merely linked with the output layer of DAE network, which could not effectively improve the discriminant information in feature extraction. In this paper, we try to integrate the output discriminant information and the structural relatedness information directly into the training process of DAE. We hope this development can optimize the process of DAE feature extraction in theory, to obtain representative features under the guidance of discriminant information.

Following the analysis mentioned above, this paper presents an improved DAE method for bearing fault diagnosis. Specifically, two main types of regularizers are added into the DAE loss function. The maximum correlation entropy-based regularizer is for the constraint of discriminant information, and the symmetric relatedness regularizer aims to constrain the structural relatedness information among multiple fault types. To optimize this DAE loss function, a gradient descent method is also proposed. Consequently, the optimal deep features, as well as fault relatedness structure, can be learned simultaneously. It is worth noting that the proposed method adopts the frequency spectrum of raw signal as input. Despite deep learning techniques are capable of using raw vibration signals directly [8], using pre-processed features as the input could reduce model complexity and simplify further feature extraction. Experimental results on two widely-used bearing datasets, CWRU and IMS datasets, show that the proposed method can effectively improve the diagnosis accuracy and numerical stability for multiple fault types. To sum up, the main contributions of this work are as follows:

- This paper proposes a new DAE network model with fusing discriminant information. Different from traditional DAE models, this model integrates discriminant information into the training process. Even if fine-tuning operation is not adopted, this model can also effectively improve the discriminant ability of the extracted features. Moreover, this model can enhance largely the feature extraction ability of a single AE neural network, which will simplify DAE's network architecture.
- This paper presents a gradient descent method to optimize the new DAE network. For better reference, the detailed derivation procedure is also provided.
- This paper presents a new strategy to improve the feature's representative ability by introducing extra information into deep neural networks. With this strategy, this paper verifies the existence of structural information among multiple fault conditions, and proves such information can effectively improve a deep learning model's stability on insufficient data.

The paper is organized as follows. In Section 2, preliminary works about intelligent fault diagnosis are provided. Section 3 is devoted to showing the flowcharts of the proposed method. Experimental results on CWRU and IMS bearing datasets are provided in Section 4, followed by a conclusion of this paper. The last section is an Appendix for the detailed derivation of the model optimization procedure.

2. Preliminary works

As vibration signal can reflect the health states of rotational machinery intuitively, most of the current researches about rolling bearing fault diagnosis work with vibration signals. Generally, current fault diagnosis approaches can be divided into two main types: domain knowledge-based diagnosis approaches and data-driven intelligent diagnosis approaches.

Domain knowledge-based diagnosis approaches mainly rely on experiences of domain experts. This type of approaches is highly targeted to specific application scenarios, and has less dependence on the amount of data. For instance, to improve diagnosis performance of inner-race and outer-race faults, Guo et al. [11] proposed a multi-level noise reduction method based on ensemble empirical mode decomposition (EMD) and wavelet threshold to extract pulse features from strong background noise and interference components. To extract accurate fault characteristics from raw signals, Yan et al. [12] proposed a multiple domains-based fault identification method based on statistical analysis on the results of fast Fourier transform (FFT) and variational mode decomposition (VMD). To extract the periodic impact component embedded in strong noise, Huang et al. [13] proposed a sparse representation method named adapted dictionary free orthogonal matching pursuit (ADOMP). As this method can exploit the representative characteristics of bearing fault, it is considered to be suitable for early fault recognition. Aiming at the fault diagnosis problem under variable speed conditions, Wang et al. [14] proposed a hybrid method based on computed order tracking (COT) algorithm and VMD feature representation. To conduct a robust diagnosis in a strong noise environment, Lei et al. [15] proposed a new nonlinear analysis method named symplectic entropy. These methods, however, heavily rely on the domain knowledge about fault characteristics such as fault characteristic frequency. And the extracted hand-crafted fault features are not universally applicable for different fault types.

Data-driven intelligent diagnosis methods mainly introduce machine learning techniques based on the extracted fault features. The traditional machine learning algorithms used in intelligent diagnosis include support vector machine (SVM) [12], logistic regression [16], etc. Running on some hand-crafted features, these algorithms mainly focus on the model's generalization ability. In recent years, the quick development of deep learning techniques provides a new solution for fault diagnosis. Rather as a single algorithm, deep learning technique is a kind of neural networks with deep architecture. Compared to traditional hand-crafted feature extraction methods, deep learning techniques can extract adaptively representative features from raw data, less relying on considering domain knowledge of fault types [1]. Due to these advantages, deep learning techniques have become a promising tool in the field of bearing fault diagnosis. For instance, Liu et al. [17] proposed a recurrent neural network (RNN) method for bearing fault diagnosis based on the deep auto-encoder network. Shen et al. [18] applied the contractive auto-encoder network to extract robust fault feature in the strong noise environment. Aiming at fault diagnosis problem in a complex and variable environment, Shao et al. [19] proposed a new DAE model to improve the diagnosis accuracy and robustness. To improve the recognition performance for different health states in noisy and variable working condition, Lu et al. [20] introduced stacked denoising auto-encoder (SDAE) model to extract high-order fault features with better robustness. To improve diagnosis model's generalization ability, Zhu et al. [21] started from the structural location information among features and proposed a novel capsule network based on CNN and regression strategy. Shao et al. [22] proposed a new fault diagnosis method based on DBN network. This method adopts wavelet packet transform to generate spectrum data and feeds them into DBN network to improve the convergence speed and diagnostic accuracy. It is worth noting that, different types of deep neural networks have the different performance of feature representation. Among these models, DAE is considered to be more applicable to bearing fault diagnosis [17–20] due to its simple structure and good representation performance for raw signals.

Although these deep learning techniques have comparative advantages to the traditional hand-crafted statistical features, they also have some shortcomings: 1) Mass of data is needed to train an effective model, and 2) Training on deep network architecture is usually computationally expensive. When the training data are insufficient, deep learning techniques tend to fail to extract representative enough features of bearing fault, and as a result, the decision model would be over-fitting. But for bearings condition monitoring, the limitation of the experimental device and sensor technology may cause short data acquisition. Consequently, direct application of the above deep learning techniques on fault diagnosis problems would cause the reduction of diagnosis accuracy and model stability.

From the perspective of machine learning theory, one feasible solution to the above problem is introducing extra prior information into a deep neural network [6]. According to our literature survey, there are some pioneer researches about this idea in various fields. To improve the diagnosis stability and reliability for bearings early fault, Ma et al. [23] proposed a deep residual network-based method which fuses time-domain statistical information of vibration signal. To develop the robustness of trust inference, Ref. [24] introduced some prior information about the latent relatedness between text context-aware stereotypes and credibility into the deep learning model. To improve the prediction accuracy of crude oil price, Ref. [25] proposed a long-short term memory method based on prior knowledge. This method utilizes a data transfer strategy to incorporate the different weights of old and new price data dynamically, then the fluctuation trend of oil price data can be better reflected. To improve the fidelity of image reconstruction, Ref. [26] proposed a new deep learning model called T-Net which incorporates the intrinsic characteristic of MRI image.

To sum up, the works mentioned above testify the positive effect of prior information for training a deep learning model. We also find the prior information used in these works mainly are structural information or temporal information in input data, rather than the discriminant information in output data. There are few researches incorporating prior information, especially discriminant information, into deep learning model for bearing fault diagnosis.

3. A deep auto-encoder model incorporating discriminant information

In order to improve diagnosis accuracy and numerical stability on insufficient data, a new deep auto-encoder model, named discriminant information-based auto-encoder (DIAE), is presented in this section. This model adopts a new loss function which includes a maximum correlation entropy-based regularizer for incorporating discriminant information and a symmetric relation matrix-based regularizer for improving the model's stability. Moreover, a gradient descent method is used to train this new model.

3.1. Auto-encoder and stacked auto-encoder

Auto-encoder (AE) [27] is a kind of unsupervised learning neural network algorithm with three-layers architecture. AE is composed of an encoder and a decoder. The encoder is designed to learn a compressed representation codes of input data, while the decoder is set to reconstruct the input data from the codes. If the reconstruction error is small enough, the codes can be viewed as a proper feature representation of the input data. The sketch map of AE is illustrated in Fig. 1.

Given sample set $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$ with $\mathbf{x}_i \in \mathbb{R}^l$ and $\mathbf{y}_i \in \mathbb{R}^r$, the encoding process transforms $\mathbf{x}_i$ to a hidden representation $\mathbf{h}_i = [h_1, h_2, \ldots, h_m]^T$ by:

$$\mathbf{h}_i = g(\mathbf{W}\mathbf{x}_i + \mathbf{b}) \tag{1}$$

where g is an activation function of the encoder, $\mathbf{W} \in \mathbb{R}^{m\times l}$ is the weight matrix connecting the input layer with a hidden layer, and $\mathbf{b} = [b_1, b_2, \ldots, b_m]$ is a bias vector. Through a decoder, the vector $\mathbf{h}_i$ is transformed backwards to reconstruct the input sample $\mathbf{x}_i$:

$$\hat{\mathbf{x}}_i = \mathrm{g}\left(\mathbf{h}_i\mathbf{W}^T + \mathbf{c}\right) \tag{2}$$

where $\hat{\mathbf{x}}_i$ is decoder vector, $\mathbf{c} = [c_1, c_2, \ldots, c_l]$ is the bias vector of the output layer. In Eq. (2), the transpose of $\mathbf{W}$ is used to simplify the training process [28]. The activation function of encoder and decoder is usually set sigmoid function: $g(x) = \frac{1}{1+\exp(-x)}$.

Training an AE model aims to minimize the loss function which is usually constructed by MSE or cross entropy. By utilizing a gradient descent optimization algorithm, the optimal weight matrix and bias vector can be found.

By stacking multiple AEs, a stacked deep auto-encoder (SDAE) network can be constructed. In SDAE, the input data of each hidden layer is the output data of the previously hidden layer (the input layer is viewed as the 0th hidden layer). A softmax

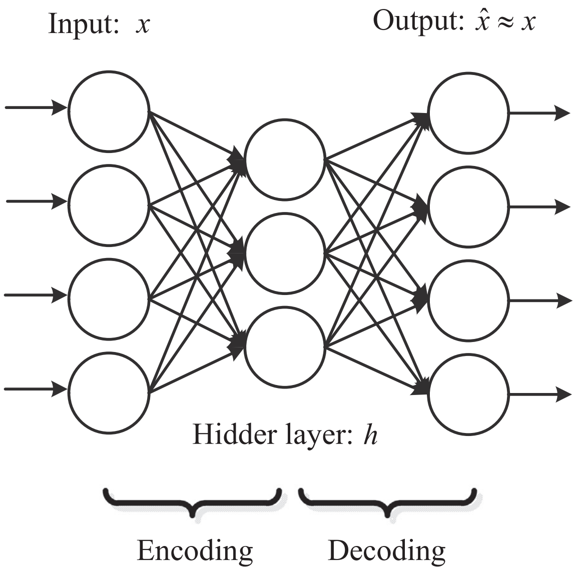


Fig. 1. Sketch map of the auto-encoder network.

layer can link to the last layer for conducting classification. The training process of SDAE contains two steps: unsupervised pre-training and supervised fine-tuning. Pre-training aims to initialize the weight matrix of each AE by using a gradient descent algorithm, and fine-tuning is used to adjust the network by means of discriminant information from softmax classifier. In fine-tuning, a back-propagation algorithm is used to update the weight of all AEs via minimizing the whole loss function of SDAE.

3.2. Loss function of DIAE

Before entering the following section, we denote the sample set $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$ by a matrix form $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_n] \in \mathbb{R}^{n\times l}$ and $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_n] \in \mathbb{R}^{n\times r}$. Here $\mathbf{y}_i$ is an r-dimensional one-hot row vector in which the value of the ith dimension is 1 while the values of the other dimensions are all 0, and r is the number of fault types.

The encoding process of DIAE is similar to the AE model described in Section 3.1, i.e., mapping the input sample $\mathbf{x}_i$ into a hidden layer by using Eq. (1). For better distinction, we replace $\mathbf{W}$ in Eq. (1) with a new letter $\mathbf{W}_X$, i.e., $\mathbf{h}_i = g(\mathbf{x}_i\mathbf{W}_X)$. Here $\mathbf{h}_i$ and $g(\cdot)$ both have the same meaning of Eq. (1). For the convenience of theoretical derivation, we fuse the bias symbol $\mathbf{b}$ into $\mathbf{W}_X$, i.e., removing the symbol $\mathbf{b}$ from Eq. (1) while adding an extra dimension into the sample $\mathbf{x}_i$ as: $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in-1}, 1]$. This operation will bring the same effect of using the bias term in Eq. (1).

Different from the classical AE model, the decoding process of DIAE includes two parts. One is the reconstruction of input data using the hidden representation $\mathbf{h}_i$:

$$\hat{\mathbf{x}}_i = g\left(\mathbf{h}_i\mathbf{W}_X^T\right) \tag{3}$$

And the other part is the prediction of input data using $\mathbf{h}_i$:

$$\hat{\mathbf{y}}_i = g(\mathbf{h}_i\mathbf{W}_T) \tag{4}$$

where $\mathbf{W}_T \in \mathbb{R}^{m\times r}$ is the discriminant weight matrix from the hidden layer to the output $\mathbf{y}_i$. Eq. (4) is able to introduce discriminant information in the DIAE model. Moreover, to represent the structural information among different fault types, we introduce a symmetric matrix $\mathbf{L} \in \mathbb{R}^{r\times r}$ into DIAE model. Each element of $\mathbf{L}$ indicates the degree of relatedness between every two fault types. Then the prediction of input data can be re-written as:

$$\tilde{\mathbf{y}}_i = \hat{\mathbf{y}}_i\mathbf{L} = g(\mathbf{h}_i\mathbf{W}_T)\mathbf{L} \tag{5}$$

From Eq. (5), we introduce not only the discriminant information but also the structural information among different fault types. Based on Eq. (5), we construct the loss function of DIAE, as follows:

$$J = \frac{\alpha}{2n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(-\kappa_\sigma(\mathbf{y}_i, \hat{\mathbf{y}}_i\mathbf{L})) + \frac{\beta}{2}\left\|\mathbf{L} - \mathbf{L}^T\right\|_F^2 + \frac{\lambda}{2}\left(\|\mathbf{W}_X\|_F^2 + \|\mathbf{W}_T\|_F^2\right) \tag{6}$$

where α, β, λ are the regularization parameters. As the matrix $\mathbf{W}_X, \mathbf{W}_T$ and $\mathbf{L}$ are all randomly initialized, we can find the optimal $\mathbf{W}_X^*, \mathbf{W}_T^*$ and $\mathbf{L}^*$ by minimizing J. For a better understanding, we describe each term in Eq. (6) as follows:

- The first term $\frac{1}{2n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2$ is reconstruction error which is the same as the classical AE. Through minimizing this term, the representation ability of features for input data can be improved.
- The second term $\frac{1}{n}\sum_{i=1}^{n}(-\kappa_\sigma(\mathbf{y}_i, \hat{\mathbf{y}}_i\mathbf{L}))$ is designed to represent the discriminant information. Minimizing this term would improve the discriminant ability of the extracted features. Here $\kappa_\sigma(\mathbf{y}_i, \hat{\mathbf{y}}_i\mathbf{L})$ is the Gaussian kernel:

$$\kappa_\sigma(\mathbf{y}_i, \hat{\mathbf{y}}_i\mathbf{L}) = 1\Big/1\left(\sqrt{2\pi}\sigma\right)\left(\sqrt{2\pi}\sigma\right)\exp\left(-(\mathbf{y}_i - \hat{\mathbf{y}}_i\mathbf{L})^2/2\sigma^2\right) \tag{7}$$

  As pointed by [19], $\frac{1}{n}\sum_{i=1}^{n}\kappa_\sigma(a, b)$ can be viewed as an approximate calculation of the correntropy $V_\sigma(A, B)$:

$$V_\sigma(A, B) = E(\kappa_\sigma(A, B)) = \int \kappa_\sigma(A, B)dF_{AB}(a, b) \tag{8}$$

  where $A = [a_1, a_2, \ldots, a_N]^T$ and $B = [b_1, b_2, \ldots, b_N]^T$ denote two stochastic variables, $E(\kappa_\sigma(A, B))$ is the expectation of $\kappa_\sigma(A, B)$, $\kappa_\sigma(\cdot, \cdot)$ can be any Mercer kernel, and $F_{AB}(a, b)$ is joint probability density function (PDF). As in many real applications, the PDF of finite samples is unknown, and the above approximate calculation of correntropy is usually adopted. Because correntropy has a good mathematical property for nonlinear and local similarity metric while maximum correntropy is insensitive to complex and non-stationary background noise [29,30], minimizing the second term in Eq. (6) is considered to be capable of improving the discriminant ability of the extracted features.
- The third term $\left\|\mathbf{L} - \mathbf{L}^T\right\|_F^2$ is designed to constrain the degree of relatedness among different fault types. Minimizing this term aims to enforce $\mathbf{L}$ to be symmetric.
- The fourth term $\left(\|\mathbf{W}_X\|_F^2 + \|\mathbf{W}_T\|_F^2\right)$ is the Frobenius norm of $\mathbf{W}_X$ and $\mathbf{W}_T$. Minimizing this term is to prevent over-fitting.

The output of the hidden layer in DIAE is the extracted feature representation of input data. Minimizing Eq. (6) will result in a special process of feature extraction that works under the guidance of discriminant information. That means, beyond the input information like in classical AEs, more output information is introduced to further improve the representation ability of the extracted features, and structural information among fault types represented by the matrix **L** is also introduced to improve model's numerical stability. If two fault conditions are originally more related to each other, the discriminant information will push the weight value in **L** to be smaller. And from the second term in Eq. (6), a lower weight value of L will reduce the effect of the classification between these two fault conditions. Minimizing Eq. (6) can be solved by using a gradient descent algorithm, as listed in the next section.

3.3. Optimization of DIAE

For convenience, we simplify the minimization of Eq. (6) by using matrix form, as:

$$\min_{\mathbf{W}_X,\mathbf{W}_T,\mathbf{L}} J_1 + J_2 + J_3 + J_4 \tag{9}$$

where:

$$J_1 = -\mathrm{tr}\left(K_\sigma\left(\hat{\mathbf{Y}}\mathbf{L},\mathbf{Y}\right)\right)$$

$$J_2 = \frac{\alpha}{2}\left\|\hat{\mathbf{X}}-\mathbf{X}\right\|_F^2$$

$$J_3 = \frac{\beta}{2}\left\|\mathbf{L}-\mathbf{L}^T\right\|_F^2$$

$$J_4 = \frac{\lambda}{2}\left(\|\mathbf{W}_X\|_F^2 + \|\mathbf{W}_T\|_F^2\right)$$

We adopt an alternative optimization method to solve Eq. (9). First, fix $\mathbf{W}_T$ and **L**, then use the gradient descent algorithm to optimize $\mathbf{W}_X$; Second, fix $\mathbf{W}_X$ and **L**, then use the gradient descent algorithm to optimize $\mathbf{W}_T$; Finally, fix $\mathbf{W}_X$ and $\mathbf{W}_T$, then use the gradient descent algorithm to optimize **L**. Starting from initializing $\mathbf{W}_X, \mathbf{W}_T, \mathbf{L}$, these three steps run in turn until convergence. Specifically, the values of $\mathbf{W}_X, \mathbf{W}_T, \mathbf{L}$ can be calculated by using the following equations:

$$\mathbf{W}_X = \mathbf{W}_X + \eta\frac{\partial J}{\partial \mathbf{W}_X} \tag{10}$$

$$\mathbf{W}_T = \mathbf{W}_T + \eta\frac{\partial J}{\partial \mathbf{W}_T} \tag{11}$$

$$\mathbf{L} = \mathbf{L} + \eta\frac{\partial J}{\partial \mathbf{L}} \tag{12}$$

where η is the learning rate. The partial derivatives of J with respect to $\mathbf{W}_X, \mathbf{W}_T, \mathbf{L}$ respectively, i.e., $\frac{\partial J}{\partial \mathbf{W}_X}, \frac{\partial J}{\partial \mathbf{W}_T}, \frac{\partial J}{\partial \mathbf{L}}$

$$\frac{\partial J}{\partial \mathbf{W}_X} = \frac{\partial J_1}{\partial \mathbf{W}_X} + \frac{\partial J_2}{\partial \mathbf{W}_X} + \frac{\partial J_4}{\partial \mathbf{W}_X} = \frac{C}{\sigma^2}\mathbf{X}^T\left(\left(\mathbf{G}\mathbf{W}_T^T\right)\odot dg(\mathbf{X}\mathbf{W}_X)\right) + \alpha_c\left(\mathbf{X}^T((\mathbf{E}\mathbf{W}_X)\odot dg(\mathbf{X}\mathbf{W}_X)) + \mathbf{E}^T\mathbf{H}\right) + \lambda\mathbf{W}_X \tag{13}$$

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{W}_T} &= \frac{\partial J_1}{\partial \mathbf{W}_T} + \frac{\partial J_4}{\partial \mathbf{W}_T}\\ &= \frac{C}{\sigma^2}\mathbf{H}^T\left(\left(\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T\right)\odot \mathrm{dg}(\mathbf{H}\mathbf{W}_T)\right) + \lambda\mathbf{W}_T\end{aligned} \tag{14}$$

$$\frac{\partial J}{\partial \mathbf{L}} = \frac{\partial J_1}{\partial \mathbf{L}} + \frac{\partial J_4}{\partial \mathbf{L}} = \frac{C}{\sigma^2}\hat{\mathbf{Y}}^T\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right) + \beta\left(\mathbf{L}-\mathbf{L}^T\right) \tag{15}$$

where $\odot$ is Hadamard product, and $\mathbf{H} = [\mathbf{h}_1; \mathbf{h}_2; \ldots; \mathbf{h}_n] \in \mathbb{R}^{n\times m}$ is the output of the hidden layer. For the convenience of presenting the derivative results, we simplify the frequently occurring formulas in Eqs. (13)–(15) as follows:

$$\mathbf{B} = -\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)^T/2\sigma^2 \tag{16}$$

$$C = -1/\sqrt{2\pi}\sigma \tag{17}$$

$$\mathbf{D} = \mathbf{I}\odot\exp(\mathbf{B}) \tag{18}$$

$$\mathbf{G} = \left(\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L} - \mathbf{Y}\right)\mathbf{L}^T\right) \odot dg(\mathbf{H}\mathbf{W}_T) \tag{19}$$

$$\mathbf{E} = \left(\hat{\mathbf{X}} - \mathbf{X}\right) \odot dg\left(\mathbf{H}\mathbf{W}_X^T\right) \tag{20}$$

Please refer to the Appendix for more details of the derivation. Similar to stacked AE described in Section 3.1, stacked DIAE (SDIAE) with K layers also utilizes a layer-by-layer greedy training strategy, as shown in Fig. 2. The matrix $\mathbf{W}_X^k$ between the kth hidden layer $\mathbf{h}^k$ and the $(k+1)$th hidden layer $\mathbf{h}^{k+1}$ can be optimized by setting $\mathbf{h}^k$ as the input layer and $\mathbf{h}^{k+1}$ as the hidden layer. as the hidden layer. When $k = 0, h^0$ is the input layer. When $k = K$, i.e., the last hidden layer $\mathbf{h}^K$ is the extracted features of SDIAE. To predict the expected label $\hat{\mathbf{y}}_i$ of the input sample $\mathbf{x}_i$, we link a softmax classifier with $\mathbf{h}^K$. But different from stacked AE, SDIAE is a kind of feed-forward neural network algorithm, without any fine-tuning operations. The discriminant information will be introduced in DIAE in the process of minimizing Eq. (6), and the extracted features will have better discriminative ability without extra supervised fine-tuning procedure. The network structure shown in Fig. 2 is then simplified and the time of model training is shortened as well, since most computational cost of stacked AE techniques exists in the fine-tuning procedure.

Compared to other deep learning techniques, SDIAE fuses structural information among multiple outputs, while this kind of information will enhance the effect of classification, which can be demonstrated by Eqs. (5) and (6). The matrix $\mathbf{L}$ can assign different weights on the classification function. If two fault conditions are originally more related to each other, the discriminant information will push the weight value in $\mathbf{L}$ to be smaller. And from Eq. (6), a lower weight value of L will reduce the effect of the classification between these two fault conditions. Consequently, under the guidance of such structural information, the discriminant ability of the extracted features can also be enhanced. This advantage is also verified in Section Experiments.

To sum up, the detailed training procedure of SDIAE is presented by Algorithm 1.

Algorithm 1. Training algorithm of SDIAE

```
Input: Training sample set $\{\mathbf{X},\mathbf{Y}\} = \{\mathbf{x}_i,\mathbf{y}_i\}_{i=1}^n$, the number of hidden layers $K$, the neuron number in each hidden
  layer $\{m^j\}_{j=1}^K$, the maximum iteration number $max\_iter$
Output: The weight matrix of each hidden layer $\mathbf{W}_X = \left\{\mathbf{W}_X^i\right\}_{i=1}^K$, and the output of each hidden layer $\left\{\mathbf{H}^i\right\}_{i=1}^K$
Calculate $\mathbf{H}^0 = \mathbf{X}$
for $i \leftarrow 1, K$ do
  Construct DIAE network with $m^i$ hidden neurons;
  Initialize randomly $\mathbf{W}_X^i$ and $\mathbf{W}_T^i$; Initialize $\mathbf{L}$ as a unit diagonal matrix;
  Set $iter = 1$;
  while $iter < max\_iter$ do
    Use an alternative optimization method to update $\mathbf{W}_X^i$, $\mathbf{W}_T^i$, $\mathbf{L}^i$ using
    Eqs. (10)–(12) respectively;
    $iter \leftarrow iter + 1$;
    If the difference of the loss function J in two consecutive iterations is
    smaller than the threshold, break the loop;
  end while
  Construct the $i-$th hidden layer of SDIAE using $\mathbf{W}_X^i$;
  Calculate $\mathbf{H}^i$ using $\mathbf{H}^i = g\left(\mathbf{H}^{i-1}\mathbf{W}_X^i\right)$;
  $\mathbf{X}^{i+1} \leftarrow \mathbf{H}^i$;
end for
return $\mathbf{W}_X^i$
```

4. Experiments

To verify the effectiveness of the proposed SDIAE on bearing fault diagnosis, we run comparative experiments on two widely-used bearing fault datasets, i.e., CWRU dataset and IMS dataset. For a comprehensive comparison, we compare SDIAE with fourteen typical fault diagnosis methods. These fourteen methods include one state-of-the-art signal processing method, i.e., modified multi-scale symbolic dynamic entropy (MMSDE) [31], and thirteen typical machine learning-based methods (four shallow models and nine deep models), as listed in Table 1. These four shallow models include output kernel learning (OKL) [10], SVM [12], random forest (RF) [32] and sparse Bayesian extreme learning machines (SBELM) [33]. The nine deep learning-based diagnosis methods include five traditional deep neural networks, i.e., CNN [3], DBN [4], SAE [5], contractive auto-encoder (CAE) [18] and SDAE [27], and four state-of-the-art methods, i.e., DTCWPT [34], SIFT-CNN [35], DFCNN [36] and MC-CNN [37]. It is worth noting that the last four methods all obtained good diagnostic results on CWRU bearing dataset. For each deep learning model, it is necessary to conduct model selection by choosing and testing different hidden neurons. Finally, we choose the neurons with the highest diagnostic accuracy as the used one. Here we would like to give a particular explanation of eight typical methods, as follows:

- MMSDE [31] was proposed to evaluate and examine the dynamic characteristic of the vibration signal. Based on the features extracted by using symbolic entropy, this method introduces the mRMR approach to choose representative ones. And the chosen features are fed into SVM to recognize fault types. In Ref. [31], MMSDE has been compared by the modified multi-scale entropy (MMPE) and multi-scale permutation entropy (MMSE), we think MMSDE is a state-of-the-art fault diagnosis method from the perspective of signal analysis.

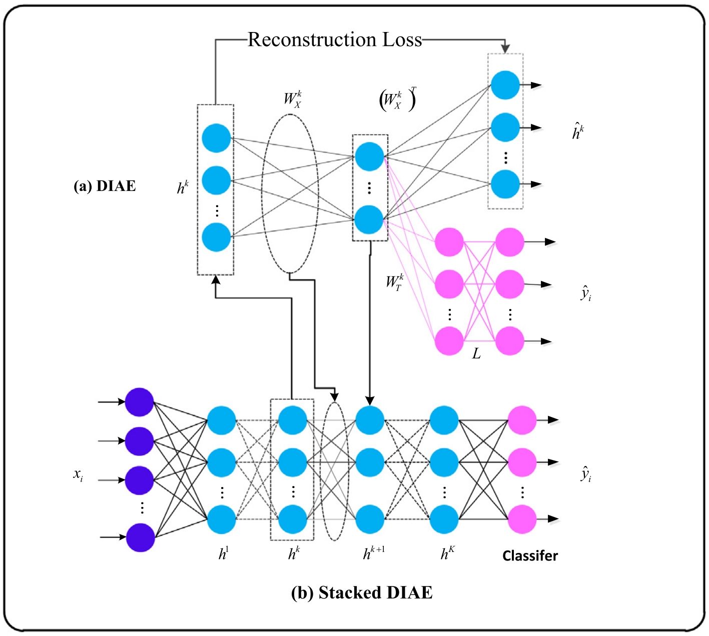


Fig. 2. Structure of the stacked DIAE.

Table 1
Input description of all the 15 methods for comparison.

Type	Methods	Input Data
Shallow Model	SVM [12]	71-dimensional Heterogeneous Feature [34]
	OKL [10]	
	RF [32]	
	SBELM [33]	
	MMSDE-SVM [30]	30-dimensional MMSDE Feature [32]
Deep Model	CNN [3]	FFT spectrum
	SAE [5]	
	CAE [18]	
	DBN [4]	
	SDAE [27]	
	SDIAE	
	DTCWPT [34]	Raw signal
	SIFT-CNN [35]	
	DFCNN [36]	
	MC–CNN [37]	

- SBELM [33] was proposed to identify multiple fault types for automotive engines. Through feature extraction by using ensemble empirical mode decomposition (EEMD) and correlation coefficient, this method proposed a new probabilistic feed-forward neural network to tackle multi-modal fault data. As this method is mainly for the multi-classification problem, we think it is very suitable to make a comparison with our method for diagnosis of multiple fault types.
- RF [32] was also proposed to diagnose multiple fault types under different working conditions for the gearbox. This method can not only provide diagnosis results with high accuracy but also evaluate the inner structure of heterogeneous features. Therefore, we think it is proper for comparison.
- OKL [10] was proposed to tackle the multi-classification problem. The best advantage of OKL is the ability to exploit the inner relationship among different outputs with comparable generalization performance. We think OKL can provide a comprehensive comparison from the perspective of structural prediction.
- DTCWPT [34] utilized dual-tree complex wavelet packet transform to pre-process raw vibration signals and built an original feature set from each frequency-band. Finally this method constructed an adaptive DBN network to improve the convergence rate and diagnostic accuracy.
- SIFT-CNN [35] utilized short-time Fourier transform to obtain an image of raw signal, and then used the scaled exponential linear unit (SELU) function and hierarchical regularization on CNN network to improve diagnostic accuracy.
- DFCNN [36] pre-processed the raw signal of bearings into the form of two-dimensional image, and then utilized CNN to conduct fault diagnosis. DFCNN has two convolutional layers, two pooling layers, two dropout layers and two fully-connected layers. The convolutional kernel is set 10 × 10.
- MC-CNN [37] proposed a multi-scale cascade CNN by adding a multi-scale information fusion layer on raw signal. Then a set of input data are constructed to contain more distinguishable information and the diagnostic performance can be enhanced as well.

For the four shallow models, we extract 71-dimensional heterogeneous features [38] as model inputs. The definition of the 71-dimensional features is shown in Table 2. Because deep learning techniques are capable of feature extraction, we directly feed the frequency-domain signal by FFT into the five traditional deep models (CNN, DBN, SAE, CAE and SDAE). For DTCWPT, SIFT-CNN, DFCNN and MC-CNN, the inputs all are raw signals, as stated above. Moreover, considering the random initialization of SBELM and all deep models (including SDIAE), we run the experiments 50 times repeatedly and calculate the mean value of the obtained accuracy results as the final result. Furthermore, we utilize the Kruskal–Wallis test on the results of 50 trials to verify the models numerical stability. Besides, we also compare these methods in terms of training time.

Table 2
Definition of 71-dimension fault features.

Method	Formula	Dimension
Bispectrum Analysis	$B_x(w_1, w_2) = \sum_{\tau_1=-\infty}^{\infty}\sum_{\tau_2=-\infty}^{\infty} c_{3x}(\tau_1, \tau_2)e^{-j(w_1\tau_1+w_2\tau_2)}$	10
GARCH Model	$r_t = c_1 + \sum_{i=1}^{R}\phi_i r_{t-i} + \sum_{j=1}^{M}\phi_j\varepsilon_{t-j} + \varepsilon_t \ \varepsilon_t = \mu_t\sqrt{h_t} \ h_t = k + \sum_{i=1}^{q} G_i h_{t-i} + \sum_{i=1}^{p} A_i\varepsilon_{t-i}^2$	4
EMD	$x(t) = r_n + \sum_{i=1}^{n} IMF_i$	10
WPD	$E_j(n) = \sum_{s=0}^{S/2^j-1}\left[c_{j,n}^s\right]^2 \ x_n = \frac{E_j(n)}{\sum_{m=0}^{2^j}E_j(m)}$	16
Complex Envelope Analysis	$\tilde{h}(t) : H\{h(t)\} := h(t) * \frac{1}{\pi t} = \frac{1}{\pi}\int_{-\infty}^{\infty} h(t)\frac{d\tau}{t-\tau}$	18
Impulse Factor	$X_{if} = \frac{\max(\lvert x_i\rvert)}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$	1
Margin Factor	$X_{mf} = \frac{\max(\lvert x_i\rvert)}{\left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\lvert x_i\rvert}\right)^2}$	1
Shape Factor	$X_{sf} = \frac{\max(\lvert x_i\rvert)}{\left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right)^{1/2}}$	1
Kurtosis Factor	$X_{kf} = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\bar{x}}{\sigma}\right)^4}{\left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right)^2}$	1
Crest Factor	$X_{cf} = \frac{\max(\lvert x_i\rvert)}{\left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right)^{1/2}}$	1
Peak-to-Peak Value	$X_{ppv} = \max(x_i) - \min(x_i)$	1
Skewness Value	$X_{sv} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\bar{x}}{\sigma}\right)^3$	1
Kurtosis Value	$X_{kv} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\bar{x}}{\sigma}\right)^4$	1
Square Root of the Amplitude	$X_{sra} = \left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\lvert x_i\rvert}\right)^2$	1
Root Mean Square	$X_{rms} = \left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right)^{1/2}$	1
Frequency Center	$X_{fc} = \frac{1}{N}\sum_{i=1}^{N}f_i$	1
RMS Frequency	$X_{rmsf} = \left(\frac{1}{N}\sum_{i=1}^{N}f_i^2\right)^{1/2}$	1
Root Variance Frequency	$X_{rvf} = \left(\frac{1}{N}\sum_{i=1}^{N}(f_i - X_{fc})^2\right)^{1/2}$	1

All experiments are conducted on a computer with a Core i3 7100 3.9 GHz processor and 8 GB RAM running Matlab 2014a and Python.

4.1. Dataset description

The CWRU bearing dataset was provided by Case Western Reserve University (CWRU) Electrotechnics Lab [39]. In this dataset, different fault types were artificially machined by using electro-discharge at different location of the inner race, outer race and ball with damage size of 0.007 in, 0.014 in, 0.021 in and 0.028 in. Therefore, four fault types were provided: normal condition, inner race fault, outer race fault and ball fault. The motor loads are of 0, 1, 2 and 3 hp. The fault data were recorded with sampling frequency of 12 kHz at fan end (FE) and drive end (DE), and with a sampling rate of 48 kHz at the drive end.

The IMS bearing dataset was provided by the NSF I/UCR Center for Intelligent Maintenance Systems (IMS) with support from Rexnord Corp. in Milwaukee, WI [40]. This dataset contains two different test-to-failure experimental data. The recording durations with inner race fault on bearing 3 and ball fault on bearing 4 are both from Oct 22, 2003, 12:06:24 to Nov 25, 2003, 23:39:56. The recording duration with ball fault on bearing 1 is from Feb 12, 2004, 10:32:39 to Feb 19, 2004, 06:22:39. The sampling frequency is 20 kHz. In this experiment, we choose the normal condition data collected at the first hour, the outer race fault data collected at 164 h, the inner race fault and ball fault data collected at 827 h.

4.2. Experimental settings

In this section, we design two kinds of experiments with multiple fault conditions on CWRU dataset by combining the values of fault type (inner race/outer race/ball fault), motor load (0/1/2 hp) and damage size (0.007/0.014/0.021 inch) respectively. In each experimental trial, we randomly take 70% of the collected samples for training and the remaining for test. The detailed settings are as follows:

Experiment 1: We fix damage size as 0.014 inches, and select three fault types. Each fault type has three motor loads. Then we get nine classes of bearing fault. The sampling rate is 48 kHz. For each condition, we collect 100 samples to build the training set. Each sample contains 1024 time points of the raw signal. Finally, we have 900 samples in total. The information of the 9 fault conditions is listed in Table 3.

Experiment 2: We fix fault type as ball fault, and adjust damage size and motor loads. Three loads (0/1/2 hp) and three damage sizes (0.007/0.014/0.021 inches) gives 9 classes of bearing fault. The sampling rate is 48 kHz. For each condition, we collect 100 samples, each of which contains 1024 time points of the raw signal. The reason we choose ball fault is it has been found in our previous work [41], ball fault is harder to be diagnosed than the other fault types. The information of the 9 fault conditions is listed in Table 4.

For IMS dataset, we only have whole-life vibration signal, without varying working conditions just like in CWRU dataset. Therefore, we choose the signal of bearing 1 from 100h10min to 116h14min as outer race fault data, and choose the signal of bearing 3 and 4 from 735h50min32sec to 754h35min as inner race fault and ball fault respectively. For normal condition, we choose the signal at the starting part of bearing 1 from 27h40min to 44h20min. Then we have four fault classes, each of which has 200 samples. And each sample contains 512 time points.

To have an overall observation, we plot the raw time signal and frequency signal by FFT of nine fault classes listed in Table 3, as shown in Fig. 3. For better illustration, we use 2048 sampling points to display the characteristics of different fault classes.

From Fig. 3, the signals of different fault conditions have obvious distinctions no matter in time domain or frequency domain, but some fault conditions like H1/H4/H7 have more similar shape than the others, especially around the peak. We observe that these three fault classes have some similar physical characteristics, for instance, the same damage size. Following this analysis, we further exploit such relatedness among different fault conditions from the view of feature distribution. Taking Experiment 1 as an example, we run T-SNE algorithm to visualize the 71-dimensional features listed in Table 2, as shown in Fig. 4.

Table 3
Class information of Experiment 1 on CWRU dataset with different fault type and load.

Class	Name	Fault types	Load (hp)
H1	Inner1-14	Inner	1
H2	Ball1-14	Ball	1
H3	Outer1-14	Outer	1
H4	Inner2-14	Inner	2
H5	Ball2-14	Ball	2
H6	Outer2-14	Outer	2
H7	Inner3-14	Inner	3
H8	Ball3-14	Ball	3
H9	Outer3-14	Outer	3

Table 4
Class information of Experiment 2 on CWRU dataset with varying damage size and load.

Class	Name	Damage size (in)	Load (hp)
H1	ball2-7	0.007	2
H2	ball2-14	0.014	2
H3	ball2-21	0.021	2
H4	ball1-7	0.007	1
H5	ball1-14	0.014	1
H6	ball1-21	0.021	1
H7	ball3-7	0.007	3
H8	ball3-14	0.014	3
H9	ball3-21	0.021	3

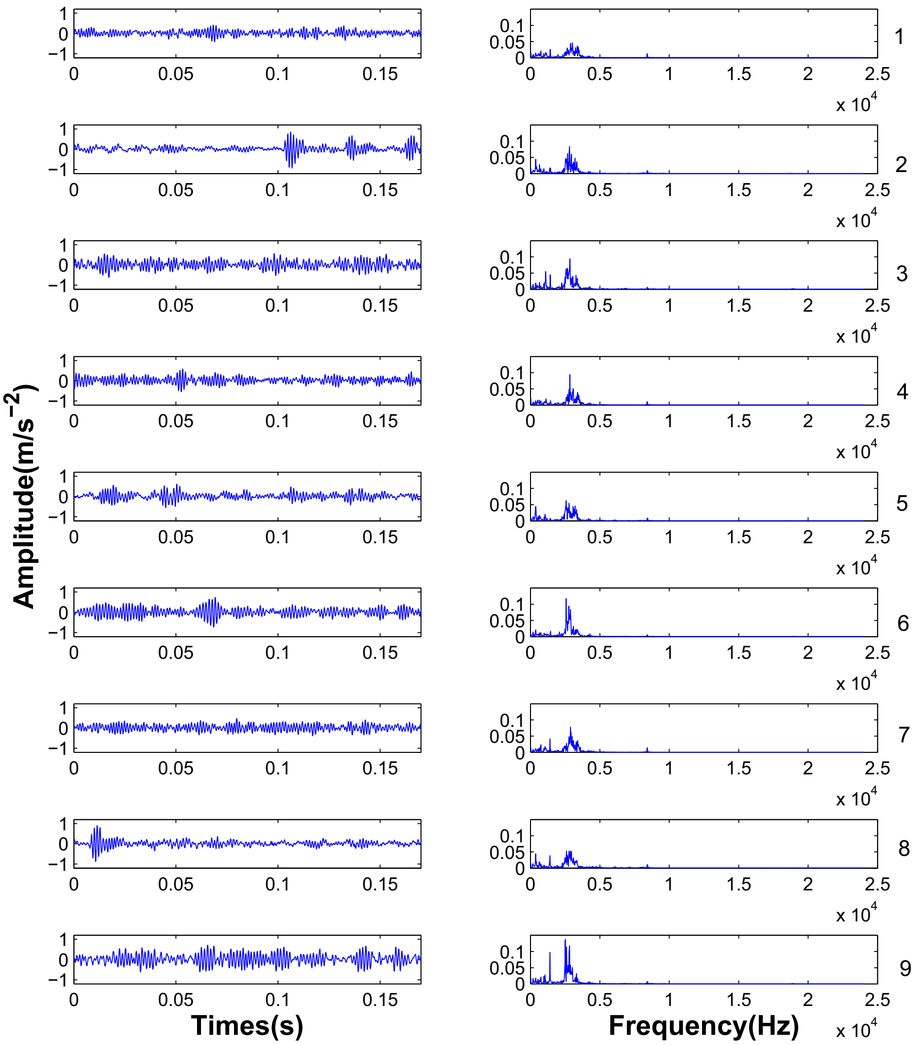


Fig. 3. Raw time signals and frequency signals by FFT of all nine fault classes of CWRU dataset. The title of each sub-figure corresponds to the index of the fault class in Table 3.

From Fig. 4, we find that some fault conditions like the fault classes H1/H4/H7 and H3/H6/H9 have obvious aggregation effect in feature distribution, while the feature distributions of some fault conditions like the fault classes H1/H2/H3 are more scattered than the others. From Table 3, the phenomenon in Fig. 4 indicates that with same damage size, the motor load tends to facilitate the similarity of feature distribution while the fault types dominate the discrimination of fault features.

From the view of machine learning theory, this phenomenon indicates the existence of inner relatedness among multiple fault conditions. This relatedness, or called prior information, can provide extra domain knowledge about fault classes to the diagnosis model. How to effectively utilize such prior information to improve the diagnosis performance is our initial research motivation.

4.3. Experimental results

4.3.1. Results of Experiment 1

In this section, we provide the comparative results of SDIAE and the other ten methods. For SVM, OKL and SBELM, 5-fold cross-validation is used to seek the optimal hyper-parameters including kernel parameter and regularization parameters. The Gaussian kernel function $\kappa(x, x') = \exp\left(-\|x - x'\|^2/\sigma\right)$ is used in SVM and OKL where σ is kernel parameter. For MMSDE, we generate 30-dimensional features as the input of SVM, and then utilize 5-fold cross-validation to conduct model selection. For SBELM, the number of hidden neurons is set 120. For RF, we set 1000 trees to build the diagnosis model. For SAE, the network architecture is set [512, 200, 100, 9] where the first number is for the input layer, and the last number is for the softmax layer, and the iteration numbers of unsupervised pre-training and supervised fine-tuning are set 50 and 300 respectively. For CNN, we set two convolutional layers, two pooling layers and two fully-connected layers. The convolutional kernel is set 5 × 5, and the sliding step is set [1,1]. The pooling kernel is set 2 × 2 while max pooling and zero padding are used for pooling. The neuron numbers of two fully-connected layers are set 16 and nine respectively. The iteration number is set 900. For CAE, the network architecture is set [512, 256, 128, 9], the iteration numbers of pre-training and fine-tuning are set 150 and 2000 respectively. For DBN, the network architecture is set [512, 512, 128, 64, 9], the data batch is set 50, and the number of back-propagation is set 400. For SDAE, the network architecture is set [512, 200, 100, 9], the noise degree is set 0.02, and the iteration numbers of pre-training and fine-tuning are set 50 and 300 respectively. For DTCWPT, the architecture of DBN is [512, 400, 250, 100, 9], the data batch is 50, and the number of back-propagation is set 400. The settings of SIFT-CNN, DFCNN and MC-CNN are same to the references [35–37] respectively. For SDIAE, the network architecture is set [512, 530, 9], the iteration number of DIAE is set 2000, and the iteration number of the softmax layer is set 800. The regularization parameters of Eq. (6) is set: $\alpha = 0.0001, \beta = 1, \lambda = 0.1$. Finally, the diagnosis accuracy and training time of these 15 methods are shown in Fig. 5.

From Fig. 5(a), ten deep models outperform four shallow models and MMSDE by around 20–30% in terms of diagnosis accuracy, which means deep learning techniques have better performance for diagnosis of multiple fault types. The reason is deep models can extract some features with better representative ability. We also observe that, in the four shallow models, OKL has higher accuracy than the other three models (SVM, SBELM and RF), which demonstrates the effectiveness of structural prediction on multiple outputs. More importantly, among all 15 methods, the diagnosis accuracy of the proposed method SDIAE reaches 96.13%, which is about 0.6% higher than the second best method MC-CNN (95.56%).

On the other hand, from Fig. 5(b), four shallow models spend much less training time than the other deep models. This phenomenon is the advantage of traditional methods with a shallow model, but these methods heavily rely on the rules of feature extraction. In all ten deep models, CAE spends the least training time (15.12 s), followed by SDAE (32.22 s). The proposed method SDIAE ranks the fourth (49.02 s), a little higher than MC-CNN (45.82 s). The reason is that SDIAE spends more time on optimizing the new loss function in the training process. But the accuracy of SDIAE is obviously higher than CAE and

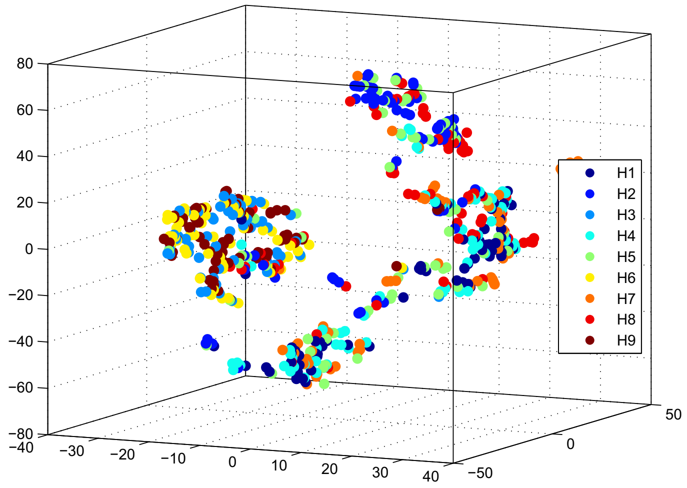
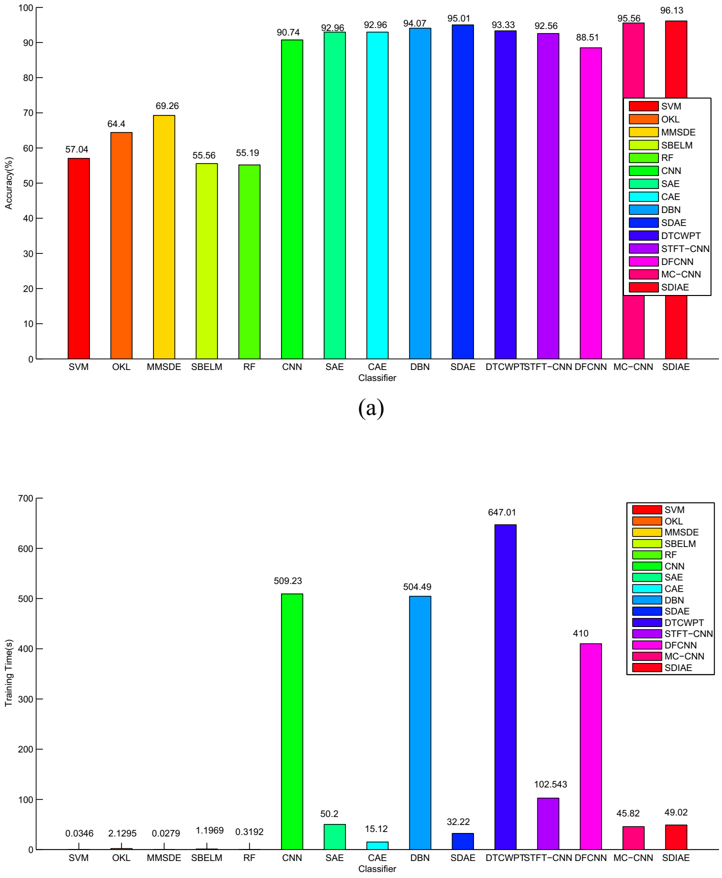


Fig. 4. Scatter plot of fault features generated by T-SNE on CWRU dataset (The legend H1-H9 are the class index of nine fault conditions in Table 3).



(a)

(b)

Fig. 5. Comparative results of all 15 methods on Experiment 1 in terms of (a) diagnosis accuracy and (b) training time.

SDAE and a bit higher than MC-CNN. Considering diagnosis accuracy and training time together, our method can get the highest accuracy in an acceptable time.

Moreover, we have tested the numerical stability of the proposed method. Here we choose SDAE for comparison, as SDAE and SDIAE both got satisfactory diagnosis performance in the above analysis. We run each method on Experiment 1 50 times repeatedly, and use the Kruskal–Wallis test on the obtained results. The mean value and standard deviation plus the p-value of the Kruskal–Wallis test of 50 trials are reported in Table 5. Please note that if a p-value is less than 0.05, it indicates the standard deviations in Table 5 have significant difference and can be used for comparison.

From Table 5, the standard deviation of SDIAE is lower than the value of SDAE. Moreover, the p-value of the Kruskal–Wallis test is much less than 0.05, which means the difference between these two standard deviations is meaningful. Therefore, SDIAE is considered to be more stable than SDAE, which also indicates the effectiveness of structural information among multiple outputs for improving the models stability.

As SDIAE adopts an alternative optimization method to train the model, the convergence speed is important. Fig. 6 provides a box diagram of the iteration number and diagnosis accuracy of SDIAE. Here the iteration number of softmax layer is set from 50 to 1950 with an interval of 50. With each iteration number, the experiments are repeated 40 times. The network architecture of SDIAE is set [512, 530, 9], the iteration number of pre-training is set 2000, and the learning rate is set 0.002.

Table 5
Mean value and standard deviation of 50 accuracy results by SDIAE and SDAE on Experiment 1. The p-value of the Kruskal–Wallis test on the standard deviation is also listed.

Method	Accuracy (%)	Standard Deviation	p-value
SDAE	95.01	0.0075	
SDIAE	96.13	0.0031	1. 5233e−9

From Fig. 6, it is clear that SDIAE achieves the highest accuracy when the iteration number reaches 200, but the prediction results have more outliers. As the iteration number increases, the outliers gradually decrease, and SDIAE tends to be stable.

Furthermore, to check the effectiveness of the symmetric matrix **L**, we remove the third term of **L** in Eq. (6) and re-plot the above box diagram, as shown in Fig. 7. By comparing these two figures, when the iteration number exceeds 1450, SDIAE with the constraint of **L** has much fewer outliers than SDIAE without such constraint. This phenomenon indicates the structural information among outputs can effectively improve the numerical stability of deep learning model.

Moreover, for better understanding of the structural relationship among different fault conditions, we visualize the matrix **L** learned by SDIAE, as shown in Fig. 8. As stated in Section 3.2, the matrix **L** represents the inner structure of output relatedness. In Eq. (6), **L** is constrained to be symmetric, as the relationship between two fault conditions is considered to be symmetric. It is clear that Fig. 8 is almost symmetric, which proves the proposed method SDIAE can learn the symmetric relationship among different fault types. It also proves the optimization method stated in Section 3.3 is effective. Specifically, we find in the lower part of Fig. 8, there are two obvious areas with heavy blue colour blocks. According to the column, three groups of blue colour blocks have a closer degree of colour: H1/H4/H7, H2/H5/H8, H3/H6/H9, which means these groups of fault conditions have closer relatedness. The relatedness values are also listed in Fig. 8(b). In the obtained matrix L, some weights are very close to zero, and only a few weights are obviously higher than zero. These non-zero weights just mean the relatedness level between two fault conditions. This phenomenon is consistent with Fig. 4 in which some fault conditions like H1/H4/H7 and H3/H6/H9 have more aggregate feature distribution than others. So we think our method SDIAE can learn and utilize the structural relationship among different fault conditions using the matrix **L**.

Also, to give an intuitional effect, we provide the feature distribution from SDIAE, as shown in Fig. 9. For comparison, we also provide the feature distribution from two deep models CAE and SDAE. For the sake of illustration, we use T-SNE algorithm to reduce the feature dimension to three dimensions. In Fig. 9(a), we can observe that three groups of fault conditions H1/H4/H7, H2/H5/H8 and H3/H6/H9 have a relatively closer distribution of traditional 71-dimensional domain features respectively. And in Fig. 9(b–c), CAE and SDAE promote this relatedness effect inside each group, with a certain degree of crossover. Better than these three kinds of features, our method SDIAE is capable of exploiting a good enough distinction of feature distribution while maintaining the original distribution structure of each feature. Again, the effectiveness of the structural information among different fault conditions can be verified from the feature level.

We also testify the comparative performance of SDIAE from the classifier level. We choose two fault conditions H1 and H4 which are considered most difficult to be classified and use linear SVM to build a classification model on different types of features, as shown in Fig. 10. Same to Fig. 9, we use T-SNE algorithm to reduce the original dimension to two dimensions. It is obvious that the classification performance on SDIAE features is much better than other features.

Besides diagnosis accuracy and standard deviation, we also introduce two widely-used evaluation indexes, detection rate and false alarm rate, to make a comprehensive comparison. We adopt the following definition of these two indexes [42]:

- Detection rate = (True detections)/(True examples) = TP/(TP + FN);
- False alarm rate = (False detections)/(All detections) = FP/(TN + FP);

where TP is true positive, FN is false negative, FP is false positive, TN is true negative. Due to space limitation, we provide the results of SDIAE and four deep learning-based diagnosis methods (MC-CNN, DTCWPT, SIFT-CNN and DFCNN) in terms of detection rate and false alarm rate, as shown in Fig. 11. These four methods for comparison are considered as the stat-of-the-art diagnosis methods. From Fig. 11, SDIAE gets the highest detection rate and the lowest false alarm rate (both purple lines) than the other four methods on most of nine fault classes. In the four methods for comparison, only DTCWPT (blue line) gets close performance to SDIAE. The other methods listed in Fig. 5 have similar comparative results.

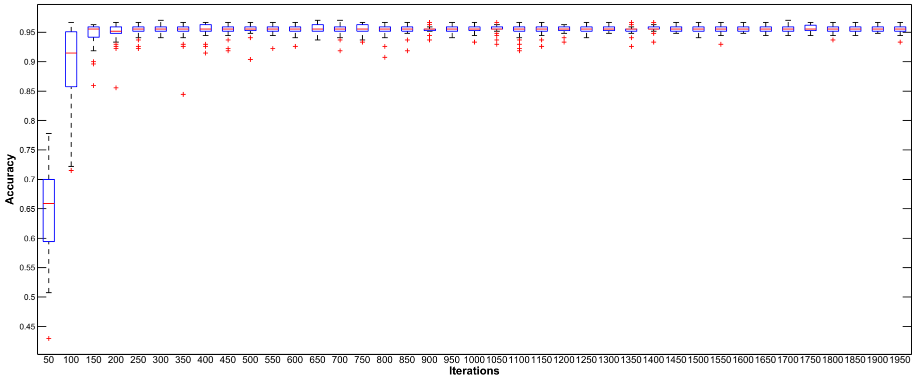


Fig. 6. Box diagram of the iteration number and diagnosis accuracy of SDIAE on Experiment 1 (the sign of red plus indicates outlier).

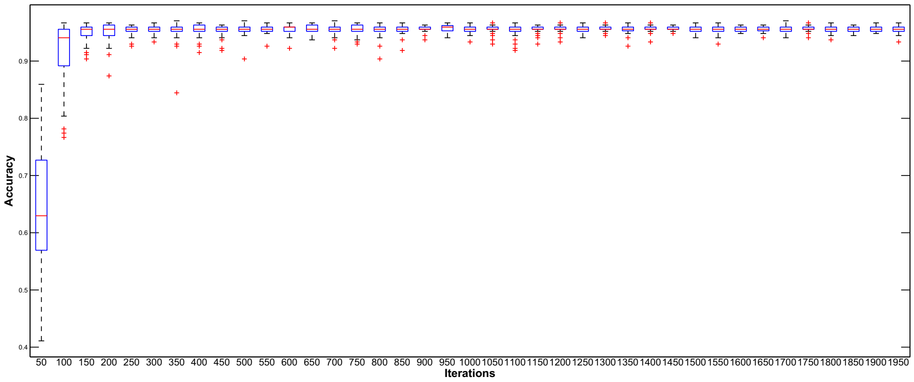
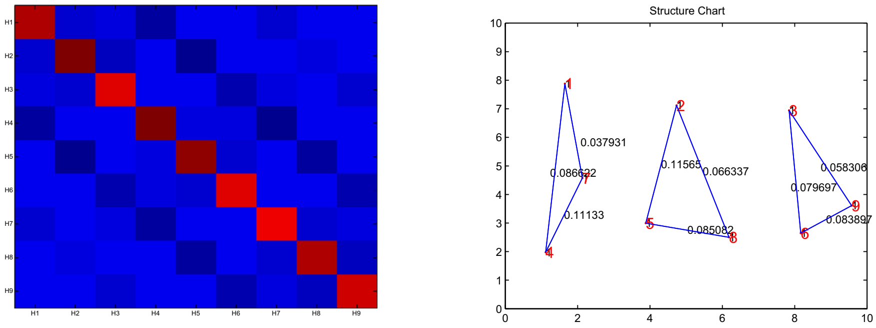


Fig. 7. Box diagram of the iteration number and diagnosis accuracy of SDIAE without the constraint of the matrix **L**.



(a) (b)

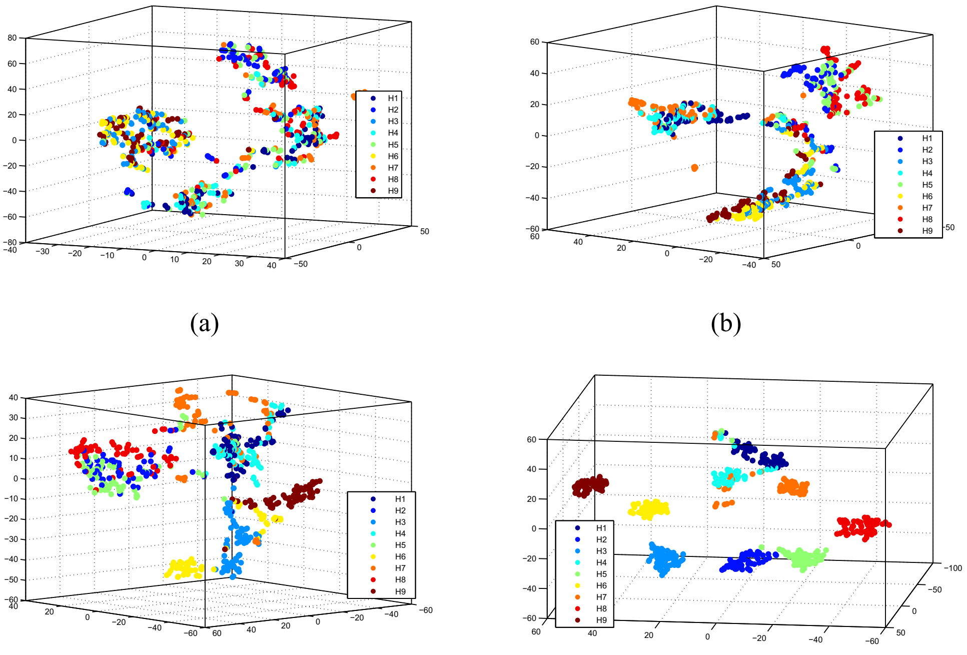
Fig. 8. Structural relationship among nine fault classes in Table 3 with (a) the visualized matrix **L** learned by SDIAE and (b) the higher weight values in the **L** (In (a), the darker the blue, the smaller the value at the corresponding position of **L**. Conversely, the deeper the red, the larger the value at the corresponding position of **L**. In (b) the number 1–9 indicate the fault classes listed in Table 3). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Furthermore, as there are total nine fault classes to be diagnosed, confusion matrix [42] is introduced to evaluate the comparative advantage of SDIAE in more details. Here we choose MC-CNN for comparison, and provide the confusion matrix of SDIAE and MC-CNN in Fig. 12. It is obvious that SDIAE and MC-CNN both get relatively high false alarm rate on H1/H4/H7. Especially for MC-CNN, the rate of misclassifying H1 to H4/H7 reaches 21%. Moreover, MC-CNN gets higher false alarm rate on H3/H6 and H5/H8. Conversely, SDIAE can accurately recognize H2/H3/H6/H9, while get lower false alarm rate than MC-CNN on the other fault classes. The other methods have similar The whole comparative results are consistent with Fig. 11, and again, demonstrate the effectiveness of structural information of multiple fault conditions in the deep learning model.

4.3.2. Results of Experiment 2

As stated in Section 2, this experiment plans to utilize the structural information among different damage sizes and motor loads to improve the diagnosis performance while keeping the fault type unchanged. The nine fault classes of Experiment 2 have been provided in Table 4. Most of the experimental settings are identical with Experiment 1. Here we adjust the data batch and the back-propagation number of DBN as 100 and 500 respectively. The iteration number of pre-training of SAE is set 100. The iteration number of CNN is set 1000. The iteration numbers of pre-training and fine-tuning of CAE are set 400 and 2500 respectively. The network architectures of SDAE and SDIAE are set [512, 256, 128, 9] and [512, 1500, 9] respectively. The iteration numbers and softmax layer of SDIAE are set 1000 and 500 respectively, and the regularization parameters are set $\alpha = 0.001, \beta = 1$, and $\lambda = 0.145$. The comparative results of all 15 methods are provided in Fig. 13.

From Fig. 13, we get a similar comparison effect to Fig. 5. To be specific, all deep models outperform four shallow models (SVM, OKL, SBELM, RF) and MMSDE, which shows the effectiveness of deep feature extraction. As a signal analysis method, MMSDE also obtains a relatively high accuracy, which indicates the multi-scale analysis for the time-domain signal can extract better representative features than traditional statistical features. But these features are all inferior to the deep features. Even as the worst performing method in the deep models, CNN still gets at least 20% higher accuracy than MMSDE. Again, SDIAE gets the highest diagnosis accuracy, which demonstrates the effectiveness of fusing structural information



(a)

(b)

(c)

(d)

Fig. 9. Scatter plot of fault features generated by T-SNE on Experiment 1 with (a) the traditional 71-dimensional features, (b) the CAE features, (c) the SDAE features and (d) the SDIAE features.

among outputs. Moreover, although SDIAE takes about 28.84 s of training time more than SDAE whose spent time is least, the accuracy of SDIAE is 1.79% higher than SDAE. It is clear that our method SDIAE can reach a higher precision in an acceptable time.

Similar to Experiment 1, we choose SDAE and SDIAE to verify the numerical stability, as shown in Table 6. The p-value of the Kruskal–Wallis test is much less than 0.05, which means the difference in standard deviation between SDAE and SDIAE from 50 repeated trials is statistically significant.

We visualize the matrix **L** learned by SDIAE, as shown in Fig. 14. For comparison, we also illustrate the distribution of 71-dimensional statistical features which are described in Table 1. Please note the feature distribution is plotted on the first three components extracted by T-SNE algorithm. Quite similar to Fig. 8, some inner structures of relatedness among different fault conditions have been exploited in terms of the colour degree of blocks. Specifically, three groups of fault conditions, i.e., H1/H4/H7, H2/H5/H8, H3/H6/H9, have closer relatedness. Referencing from Table 4, we find that with same fault type, damage size is considered to dominate more the relatedness. But unlike Fig. 8, we observe that H1 has high relatedness with H4 and H7, but H4 is more related with H1 than with H7, as shown in Fig. 14(c). This phenomenon is also found in the conditions H5-H9. This relatedness phenomenon can not be reflected very well in Fig. 14(a), which indicates that the structural information among outputs is not definitely identical with the one among input data.

4.3.3. Results on IMS dataset

As IMS dataset provides whole-life degradation data, the health conditions are relatively simpler than CWRU dataset. With no idea of the specific damage size and motor load, we collect four health conditions: normal state, outer race fault, inner race fault, ball fault. To provide an overall understanding, we illustrate the raw signal and frequency signal in Fig. 15. Here we choose 2048 sampling points and use FFT to get the frequency signal.

From Fig. 15, the signals of the four fault classes have different shapes in time domain and frequency domain, which indicates the separability of these four fault classes. Meanwhile, some conditions, e.g., normal state and outer race fault, have more similar data shape than others, especially around the peak in frequency domain. The reason may be that the data of these two health conditions are collected from the same bearing (bearing 1 in the second experiment) while the other fault classes are from another bearing under different working condition. To further exploit such relatedness, we provide the data distribution of different health conditions, as shown in Fig. 16. Same to [9], we choose two features, i.e., the 4th dimension of GARCH model feature and Kurtosis value feature, from the 71-dimensional features which are described in Table 1. It is clear that the features of the normal state and outer race fault locate nearer than the other two conditions while inner race fault and ball fault are rather closer. We think this separability between the former two and the latter two conditions indicates the existing of structural information in these data.

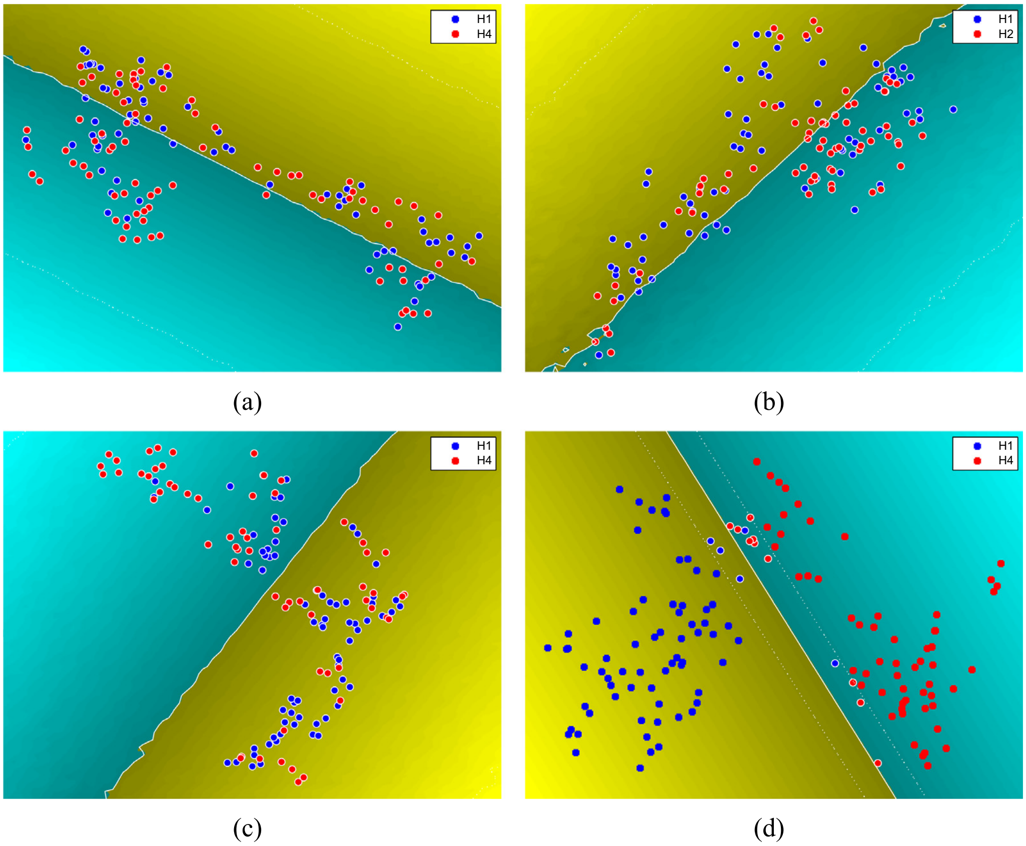


Fig. 10. Classification performance of linear SVM for the fault conditions H1 and H4 on (a) the traditional 71-dimensional features, (b) the CAE features, (c) the SDAE features and (d) the SDIAE features.

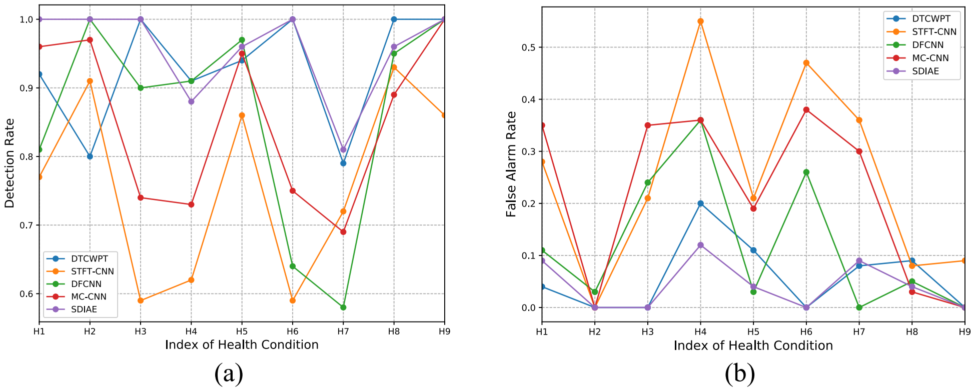
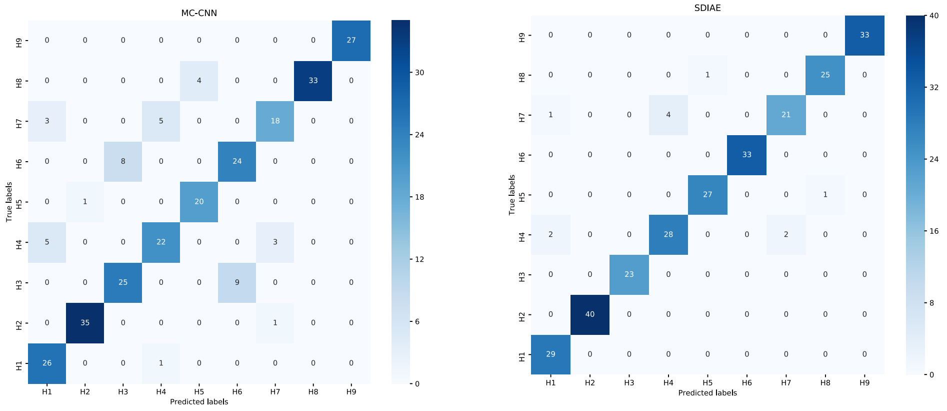


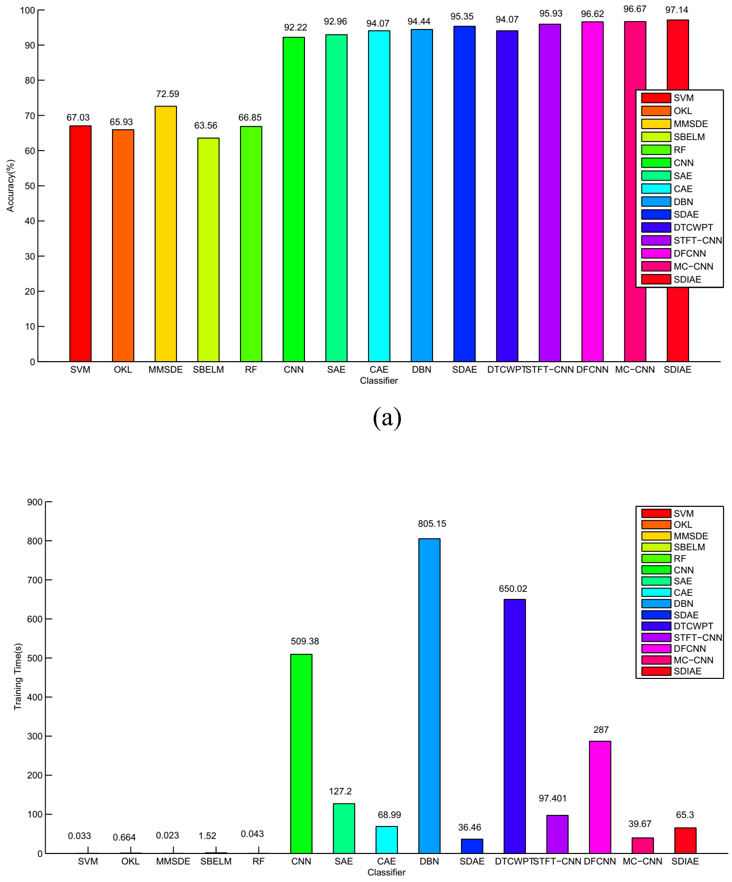
Fig. 11. Comparative results of five deep learning-based diagnosis methods on Experiment 1 in terms of (a) detection rate and (b) false alarm rate..

Fig. 17 provides the comparative results of all 15 methods in terms of diagnosis accuracy and training time. Here the basic experimental setting is the same as Experiment 1. Besides, for DBN, the network architecture is set [512, 512, 128, 64, 9], while the data batch and learning rate are set 50 and 0.075 respectively. The back-propagation number is set 200. For CAE, the network architecture is set [256, 500, 128, 4], the iteration numbers of pre-training and fine-tuning are set 200 and 700 respectively. For SDAE, the network architecture is set [256, 200, 100, 4], the iteration numbers of pre-training and fine-tuning are set 50 and 300 respectively. For DTCWPT, the network architecture is set [256, 500, 250, 100, 4] and the data batch is set 100. For DFCNN, the neuron number of two fully-connected layers are set 100 and 4 respectively, and the input scale is set 16 × 16 × 1.For SDIAE, the network architecture is set [256, 330, 4], the number of pre-training



(a) (b)

Fig. 12. Confusion matrix on Experiment 1 by (a) MC-CNN and (b) SDIAE (The x-axis means nine fault conditions listed in Table 3)..



(a)

(b)

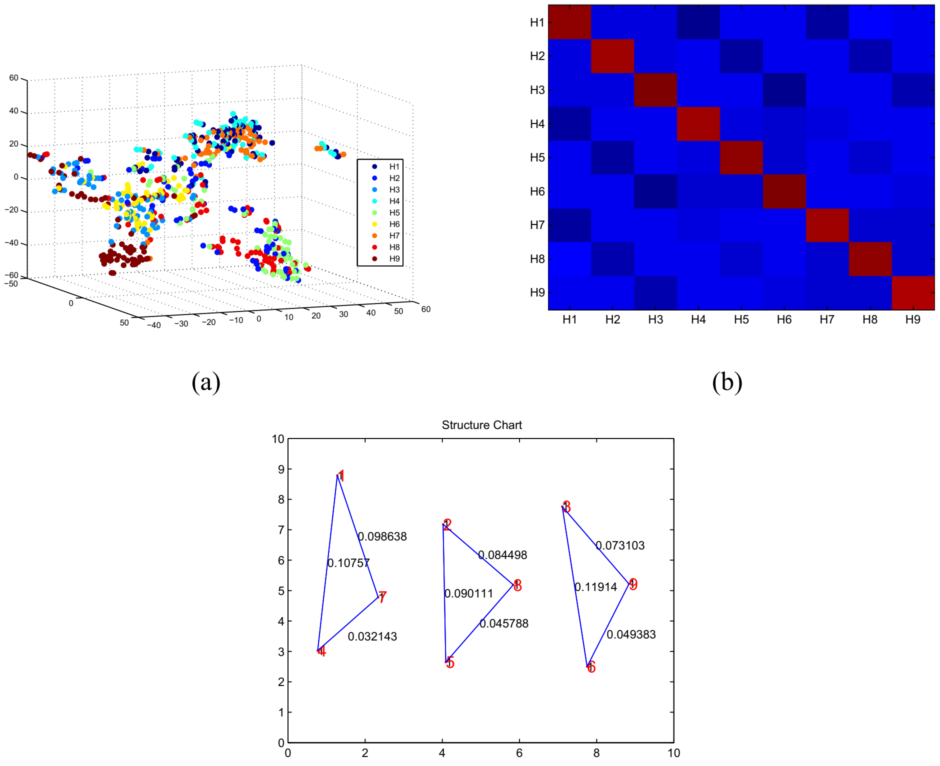
Fig. 13. Comparative results of Experiment 2 in terms of (a) diagnosis accuracy and (b) training time.

is set 900, the iteration number of softmax layer is set 2000, and the regularization parameters are set: $\alpha = 0.08, \beta = 1, \lambda = 0.145$.

Compared with Figs. 5 and 13, 17 has not very large distinction of diagnosis accuracy between all the methods. That is because the four health conditions are relatively easy to be classified, which can be testified by Figs. 15 and 16. Consequently, all methods can get a high diagnosis accuracy. Meanwhile, there is no obvious relatedness structure among four health con-

Table 6
Mean value and standard deviation of 50 accuracy results by SDIAE and SDAE on Experiment 2. The p-value of the Kruskal–Wallis test on the standard deviation is also listed.

Method	Accuracy (%)	Standard deviation	p-value
SDAE	95.35	0.0104	
SOAE	97.14	0.0057	1.34E−15



(a)

(b)

(c)

Fig. 14. Effect of structural information with (a) the distribution of input data on 71-dimensional features, (b) the visualized matrix **L** learned by SDIAE among nine fault conditions and (c) the higher weight values in the **L**.

ditions. Even in this case, our method SDIAE still gets the highest accuracy (equal to SDAE), but the training time is less. This comparative result indicates that SDIAE can convergent at fast speed on the easily distinguishable data.

Table 7 shows the mean value and standard deviation of 50 repeated trials by SDAE and SDIAE. It also shows the p-value of the Kruskal–Wallis test is much less than 0.05. That means, although the standard deviation of SDIAE (0.0066) is almost equal to the one of SDAE (0.0067), the difference is statistically significant. Then our method SDIAE is considered more stable than SDAE, which demonstrates the effectiveness of structural information among health conditions. Also, in the case of only four conditions available, the structural information of these conditions is not clear enough, so we think it is the reason why SDIAE only gets a very small improvement than SDAE. Please note that from Fig. 17, the training time spent by SDIAE is much less than SDAE.

Fig. 18 visualizes the relationship matrix **L** learned by SDIAE on IMS dataset. As only four conditions are selected, the relatedness structure is not very clear. We can find the health condition 1 and 4 are more related than others, while the condition 2 and 3 have higher relatedness. This phenomenon is consistent with the observation in Fig. 16.

Same to Experiment 1, we also exam the convergence speed of SDIAE on IMS dataset. Figs. 19 and 20 provide box diagrams of the iteration number and diagnosis accuracy of SDIAE with and without the constraint term by the matrix **L** in Eq. (6) respectively. In this experiment, the network architecture of SDIAE is set [256, 330, 4], the iteration number of pre-training is set 900, and the learning rate is set 0.002. From Fig. 19, SDIAE achieves the highest accuracy when the iteration number reaches 700, but at this time, many outliers appear. With the iteration number increasing, the outlier number gradually decreases, and the model tends to be stable. Moreover, compared with Fig. 20, SDIAE with the constraint term by

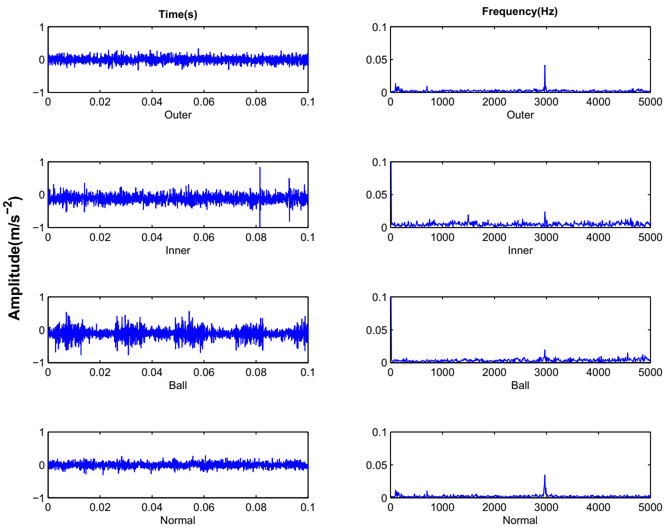


Fig. 15. Raw time signals and their FFT frequency signals of different health conditions on IMS dataset.

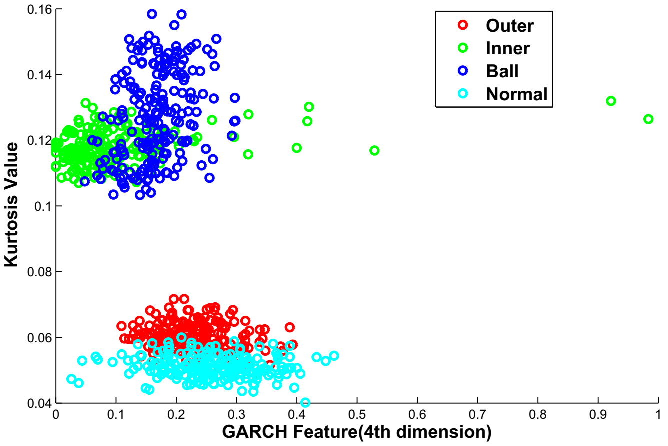


Fig. 16. Scatter plot of feature distribution on IMS dataset [9] (The x-axis is the 4th dimension of GARCH model feature and the y-axis is the Kurtosis value feature).

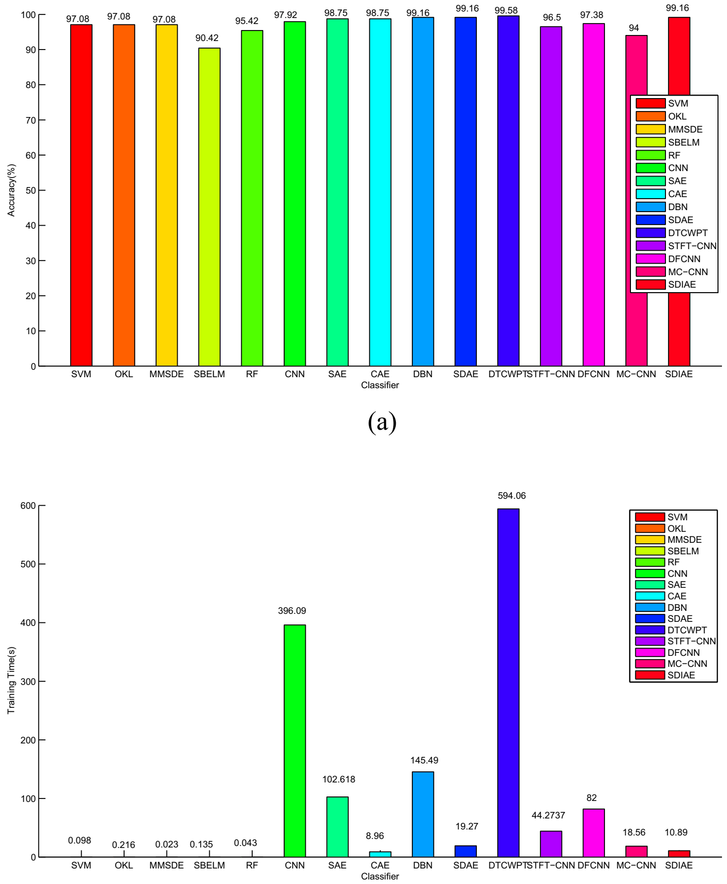
the matrix **L** have much fewer outliers than SDIAE without such constraint. This comparison indicates the model in Fig. 19 is more stable.

Finally, to further check the discriminant ability of the features extracted by SDIAE, we remove the softmax layer and directly use Eq. (5) as the output of SDIAE. The reason for this operation is to reduce the effect of the softmax layer and highlight the discriminant results caused by the features. Similar to Figs. 19 and 20, we repeat the experiment 50 times with and without the constraint of **L** respectively, and list the results on all three experiments in Table 8. To make a fair comparison, we keep all parameters same. For Experiment 1, the network architecture is set [512, 800, 9]. For Experiment 2, the network architecture is set [512, 1500, 9]. For IMS dataset, the network architecture is set [512, 900, 4]. The iteration number and learning rate of three experiments are all set 2000 and 0.002 respectively.

From Table 8, SDIAE with the constraint of **L** gets higher accuracy and much lower standard deviation than the model without such constraint on all three experiments. Even on IMS dataset, these two models achieve equal accuracy, the model with the constraint of **L** gets lower standard deviation. This comparison demonstrates the effectiveness of structural information among different fault conditions again.

5. Conclusion

In this paper, a new deep learning algorithm is proposed for bearing fault diagnosis. The key contribution is fusing discriminant information and structural information among different fault conditions in a deep auto-encoder model. And the



(a)

(b)

Fig. 17. Comparative results of all 15 methods in terms of (a) diagnosis accuracy and (b) training time.

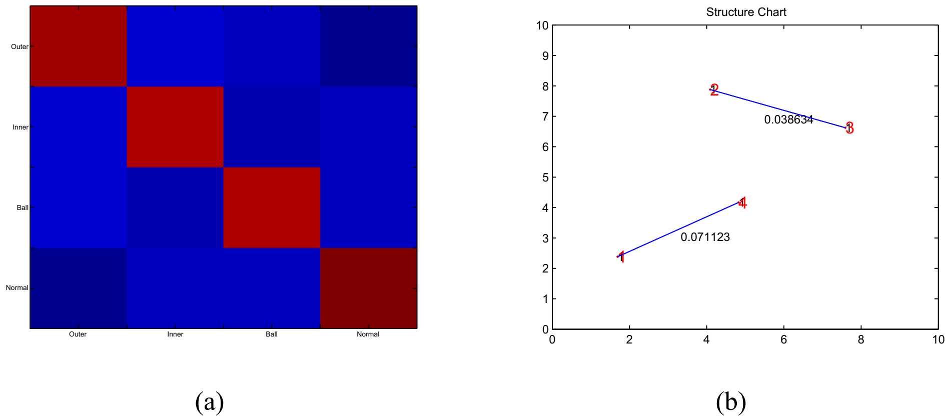
Table 7
Comparison of numerical stability between SDAE and SDIAE on IMS dataset.

Method	Accuracy(%)	Standard deviation	p-value
SDAE	99.16	0.0067	
SDIAE	99.16	0.0066	7.50E−08

main target of this work is to improve the representative ability of features under the guidance of discriminant information. From the experimental results, the following conclusions can be drawn:

- Compared with the methods with typical deep learning models and shallow models, the fusion of discriminant information into deep auto-encoder can exactly obtain higher diagnosis accuracy, especially when there isan insufficient number of training data.
- The relatedness structure among multiple fault conditions can be learned by our method, and the structural relationship information is really helpful to improve the stability of the deep neural network.
- In the case of a relatively large number of fault conditions, our method works well as the structural information among outputs can be fully exploited and expressed. Even with distinguishable data with fewer fault types, our method can still achieve satisfactory performance with less training time.

In our next work, we plan to extend the proposed architecture to the domain adaption problem. Moreover, different types of structural information (like manifold learning) can be introduced to optimize the deep model for various applications.



(a)

(b)

Fig. 18. Structural relationship among four health conditions on IMS dataset with (a) the visualized matrix **L** learned by SDIAE and (b) the higher weight values in the **L** (The number 1–4 indicates the four fault classes in the x-axis of subfigure (a)).

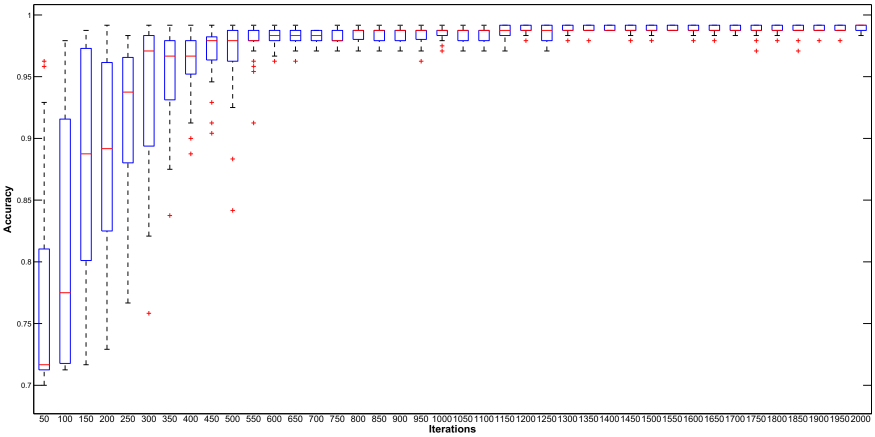


Fig. 19. Box diagram of the iteration number and diagnosis accuracy of SDIAE on IMS dataset.

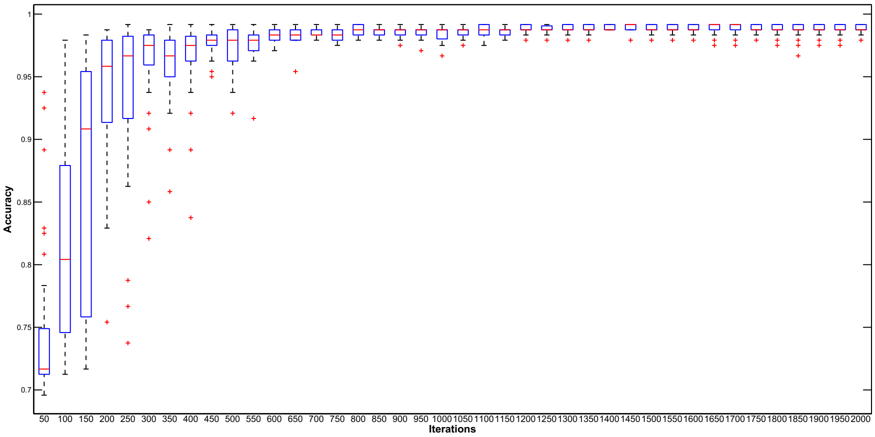


Fig. 20. Box diagram of the iteration number and diagnosis accuracy of SDIAE without the constraint of the matrix **L** on IMS dataset.

Table 8
Mean value of 50 accuracy results and deviation (in bracket) of SDIAE with and without the constraint of **L** on three experiments.

	Experiment 1	Experiment 2	IMS dataset
SDIAE without the constraint of L (%)	90.45 (0.142)	93.33 (0.0093)	98.3 (0.0019)
SDIAE (%)	94.31 (0.0057)	95.03 (0.0076)	98.3 (0.0011)

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.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. U1704158) and NSERC Discovery Grant of Canada (No. RGPIN-2019-05361).

Appendix A. Appendix

In this appendix, we provide the derivation process of minimizing Eq. (6). As shown in Eq. (9), the loss function J can be simplified by $J = J_1 + J_2 + J_3 + J_4$. The meaning of J_1 to J_4 can be found in Section 3.3. For convenience, we restate some commonly used formulas: $\hat{\mathbf{Y}} = \mathrm{g}(\mathbf{H}\mathbf{W}_T), \hat{\mathbf{X}} = \mathrm{g}\left(\mathbf{H}\mathbf{W}_X^T\right), \mathbf{H} = \mathrm{g}(\mathbf{X}\mathbf{W}_X), K_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{(x-y)^2}{2\sigma^2}\right)}$. The derivative formulas involved are as follows:

$$d\hat{\mathbf{Y}} = d\mathrm{g}(\mathbf{H}\mathbf{W}_T) \odot d(\mathbf{H}\mathbf{W}_T)$$

$$d\hat{\mathbf{X}} = d\mathrm{g}\left(\mathbf{H}\mathbf{W}_X^T\right) \odot d\left(\mathbf{H}\mathbf{W}_X^T\right)$$

$$d\mathbf{H} = d\mathrm{g}(\mathbf{X}\mathbf{W}_X) \odot d(\mathbf{X}\mathbf{W}_X)$$

In the first place, we need to calculate the partial derivatives of J with respect to $\mathbf{W}_X, \mathbf{W}_T, \mathbf{L}$ respectively, as follows:

1) Calculate $\frac{\partial J}{\partial \mathbf{W}_X}$The partial derivative of J with respect to $\mathbf{W}_X$ can be written as:

$$\frac{\partial J}{\partial \mathbf{W}_X} = \frac{\partial J_1}{\partial \mathbf{W}_X} + \frac{\partial J_2}{\partial \mathbf{W}_X} + \frac{\partial J_4}{\partial \mathbf{W}_X} \tag{21}$$

We first calculate the first term of Eq. (21), i.e.,$\frac{\partial J_1}{\partial \mathbf{W}_X}$. Denote $J_1 = -\mathrm{tr}(C \exp(\mathbf{B}))$, where $\mathbf{B} = -\frac{-(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y})(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y})^T}{2\sigma^2}, C = -\frac{1}{\sqrt{2\pi}\sigma}$, then we have:

$$\begin{aligned} dJ_1 &= -\mathrm{tr}(d(C \exp(\mathbf{B}))) \\ &= -C\,\mathrm{tr}(\exp(\mathbf{B}) \odot d\mathbf{B}) \\ &= -C\,\mathrm{tr}\left((\mathbf{I} \odot \exp(\mathbf{B}))^T d\mathbf{B}\right) \end{aligned} \tag{22}$$

Let $\mathbf{D} = \mathbf{I} \odot \exp(\mathbf{B})$, then:

$$\begin{aligned} dJ_1 &= -C\mathrm{tr}\left(\mathbf{D}^T d\mathbf{B}\right) \\ &= -C\mathrm{tr}\left(\mathbf{D}^T d\mathbf{B}\right) \end{aligned} \tag{23}$$

and:

$$d\mathbf{B} = -\left(d\hat{\mathbf{Y}}\mathbf{L}\left(\hat{\mathbf{Y}}\mathbf{L} - \mathbf{Y}\right)^T + \left(\hat{\mathbf{Y}}\mathbf{L} - \mathbf{Y}\right)\mathbf{L}^T d\hat{\mathbf{Y}}^T\right)/2\sigma^2 \tag{24}$$

then we have:

$$\begin{aligned} dJ_1 &= -C\mathrm{tr}\left(\mathbf{D}^T \tfrac{-1}{2\sigma^2} d\hat{\mathbf{Y}}\mathbf{L}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right) + \left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T d\hat{\mathbf{Y}}^T\right) \\ &= \tfrac{C}{2\sigma^2}\mathrm{tr}\left(\mathbf{D}^T d\hat{\mathbf{Y}}\mathbf{L}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)^T + \left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T d\hat{\mathbf{Y}}^T\right) \\ &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\mathbf{D}^T d\hat{\mathbf{Y}}\mathbf{L}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)^T\right) \\ &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\mathbf{L}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)^T \mathbf{D}^T d\hat{\mathbf{Y}}\right) \\ &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\mathbf{L}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)^T \mathbf{D}^T (dg(\mathbf{HW}_T)\odot d(\mathbf{HW}_T))\right) \\ &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\left(\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T\right)^T (dg(\mathbf{HW}_T)\odot d(\mathbf{HW}_T))\right) \end{aligned} \tag{25}$$

Let $\mathbf{G} = \left(\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T\right)\odot dg(\mathbf{HW}_T)$, then we have:

$$\begin{aligned} dJ_1 &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\mathbf{G}^T d\mathbf{HW}_T\right) \\ &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\mathbf{W}_T\mathbf{G}^T d\mathbf{H}\right) \\ &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\mathbf{W}_T\mathbf{G}^T (dg(\mathbf{XW}_X)\odot d(\mathbf{XW}_X))\right) \\ &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\left(\left(\mathbf{GW}_T^T\right)\odot dg(\mathbf{XW}_X)\right)^T d(\mathbf{XW}_X)\right) \\ &= \tfrac{C}{\sigma^2}\mathrm{tr}\left(\left(\left(\mathbf{GW}_T^T\right)\odot dg(\mathbf{XW}_X)\right)^T \mathbf{X}d\mathbf{W}_X\right) \end{aligned} \tag{26}$$

So we have:

$$\frac{\partial J_1}{\partial \mathbf{W}_X} = \frac{C}{\sigma^2}\mathbf{X}^T\left(\left(\mathbf{GW}_T^T\right)\odot dg(\mathbf{XW}_X)\right) \tag{27}$$

Now we calculate the second term of Eq. (21), i.e., $\frac{\partial J_2}{\partial \mathbf{W}_X}$. Because:

$$\begin{aligned} J_2 &= \tfrac{\alpha_c}{2}\left\|\hat{\mathbf{X}}-\mathbf{X}\right\|_F^2 \\ &= \tfrac{\alpha_c}{2}\mathrm{tr}\left(\left(\hat{\mathbf{X}}-\mathbf{X}\right)\left(\hat{\mathbf{X}}-\mathbf{X}\right)^T\right) \end{aligned} \tag{28}$$

we have:

$$\begin{aligned} dJ_2 &= \tfrac{\alpha_c}{2}\mathrm{tr}\left(d\hat{\mathbf{X}}\left(\hat{\mathbf{X}}-\mathbf{X}\right)^T + \left(\hat{\mathbf{X}}-\mathbf{X}\right)d\hat{\mathbf{X}}^T\right) \\ &= \alpha_c\mathrm{tr}\left(\left(\hat{\mathbf{X}}-\mathbf{X}\right)^T d\hat{\mathbf{X}}\right) \\ &= \alpha_c\mathrm{tr}\left(\left(\hat{\mathbf{X}}-\mathbf{X}\right)^T\left(dg\left(\mathbf{HW}_X^T\right)\odot d\left(\mathbf{HW}_X^T\right)\right)\right) \\ &= \alpha_c\mathrm{tr}\left(\left(\left(\hat{\mathbf{X}}-\mathbf{X}\right)\odot dg\left(\mathbf{HW}_X^T\right)\right)^T d\left(\mathbf{HW}_X^T\right)\right) \end{aligned} \tag{29}$$

Let $\mathbf{E} = \left(\hat{\mathbf{X}}-\mathbf{X}\right)\odot dg\left(\mathbf{HW}_X^T\right)$, then:

$$\begin{aligned} dJ_2 &= \alpha_c\mathrm{tr}\left(\mathbf{E}^T\left(d\mathbf{HW}_X^T + \mathbf{H}d\mathbf{W}_X^T\right)\right) \\ &= \alpha_c\mathrm{tr}\left(\mathbf{E}^T d\mathbf{HW}_X^T\right) + \alpha_c\mathrm{tr}\left(\mathbf{E}^T\mathbf{H}d\mathbf{W}_X^T\right) \\ &= \alpha_c\mathrm{tr}\left(\mathbf{W}_X^T\mathbf{E}^T d\mathbf{H}\right) + \alpha_c\mathrm{tr}\left(\mathbf{E}^T\mathbf{H}d\mathbf{W}_X^T\right) \\ &= \alpha_c\mathrm{tr}\left(\mathbf{W}_X^T\mathbf{E}^T(dg(\mathbf{XW}_X)\odot d(\mathbf{XW}_X))\right) + \alpha_c\mathrm{tr}\left(\mathbf{E}^T\mathbf{H}d\mathbf{W}_X^T\right) \\ &= \alpha_c\mathrm{tr}\left(((\mathbf{EW}_X)\odot dg(\mathbf{XW}_X))^T\mathbf{X}d\mathbf{W}_X\right) + \alpha_c\mathrm{tr}\left(\mathbf{H}^T\mathbf{E}d\mathbf{W}_X\right) \\ &= \alpha_c\mathrm{tr}\left(((\mathbf{EW}_X)\odot dg(\mathbf{XW}_X))^T\mathbf{X} + \mathbf{H}^T\mathbf{E}\right)d\mathbf{W}_X \end{aligned} \tag{30}$$

Therefore, we have:

$$\frac{\partial J_2}{\partial \mathbf{W}_X} = \alpha_c\left(\mathbf{X}^T((\mathbf{E}\mathbf{W}_X)\odot dg(\mathbf{X}\mathbf{W}_X)) + \mathbf{E}^T\mathbf{H}\right) \tag{31}$$

Now we calculate the third term of Eq. (21). As $J_4 = \frac{\lambda}{2}\left(\|\mathbf{W}_X\|_F^2 + \|\mathbf{W}_T\|_F^2\right)$, we have:

$$\frac{\partial J_4}{\partial \mathbf{W}_X} = \lambda \mathbf{W}_X \tag{32}$$

So far, $\frac{\partial J}{\partial \mathbf{W}_X}$ has been calculated, as:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{W}_X} &= \frac{\partial J_1}{\partial \mathbf{W}_X} + \frac{\partial J_2}{\partial \mathbf{W}_X} + \frac{\partial J_4}{\partial \mathbf{W}_X} \\ &= \frac{C}{\sigma^2}\mathbf{X}^T\left(\left(\mathbf{G}\mathbf{W}_T^T\right)\odot dg(\mathbf{X}\mathbf{W}_X)\right) \\ &+\alpha_c\left(\mathbf{X}^T((\mathbf{E}\mathbf{W}_X)\odot dg(\mathbf{X}\mathbf{W}_X)) + \mathbf{E}^T\mathbf{H}\right) + \lambda\mathbf{W}_X \end{aligned} \tag{33}$$

2) Calculate $\frac{\partial J}{\partial \mathbf{W}_T}$

The partial derivative of J with respect to $\mathbf{W}_T$ can be written as:

$$\frac{\partial J}{\partial \mathbf{W}_T} = \frac{\partial J_1}{\partial \mathbf{W}_T} + \frac{\partial J_4}{\partial \mathbf{W}_T} \tag{34}$$

For the first term of Eq. (34), the derivation process of $\frac{\partial J_1}{\partial \mathbf{W}_T}$ is same to Eq. (25), only replacing $\mathbf{W}_X$ by $\mathbf{W}_T$. Then we have:

$$\begin{aligned} dJ_1 &= \frac{C}{\sigma^2}\mathrm{tr}\left(\left(\left(\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T\right)\odot dg(\mathbf{H}\mathbf{W}_T)\right)^T d(\mathbf{H}\mathbf{W}_T)\right) \\ &= \frac{C}{\sigma^2}\mathrm{tr}\left(\left(\left(\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T\right)\odot dg(\mathbf{H}\mathbf{W}_T)\right)^T \mathbf{H}d\mathbf{W}_T\right) \end{aligned} \tag{35}$$

Finally, we have:

$$\frac{\partial J_1}{\partial \mathbf{W}_T} = \frac{C}{\sigma^2}\mathbf{H}^T\left(\left(\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T\right)\odot dg(\mathbf{H}\mathbf{W}_T)\right) \tag{36}$$

For the second term of Eq. (34), as $J_4 = \frac{\lambda}{2}\left(\|\mathbf{W}_X\|_F^2 + \|\mathbf{W}_T\|_F^2\right)$, we have:

$$\frac{\partial J_4}{\partial \mathbf{W}_T} = \lambda\mathbf{W}_T \tag{37}$$

So far, $\frac{\partial J}{\partial \mathbf{W}_T}$ can be calculated as:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{W}_T} &= \frac{\partial J_1}{\partial \mathbf{W}_T} + \frac{\partial J_4}{\partial \mathbf{W}_T} \\ &= \frac{C}{\sigma^2}\mathbf{H}^T\left(\left(\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)\mathbf{L}^T\right)\odot dg(\mathbf{H}\mathbf{W}_T)\right) + \lambda\mathbf{W}_T \end{aligned} \tag{38}$$

3) Calculate $\frac{\partial J}{\partial \mathbf{L}}$

The partial derivative of J with respect to $\mathbf{L}$ can be written as:

$$\frac{\partial J}{\partial \mathbf{L}} = \frac{\partial J_1}{\partial \mathbf{L}} + \frac{\partial J_4}{\partial \mathbf{L}} \tag{39}$$

From Eqs. (23) and (24), we have:

$$\begin{aligned} dJ_1 &= \frac{C}{\sigma^2}\mathrm{tr}\left(\mathbf{D}^T d\left(\hat{\mathbf{Y}}\mathbf{L}\right)\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)^T\right) \\ &= \frac{C}{\sigma^2}\mathrm{tr}\left(\mathbf{D}^T\hat{\mathbf{Y}}d\mathbf{L}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)^T\right) \\ &= \frac{C}{\sigma^2}\mathrm{tr}\left(\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right)^T\mathbf{D}^T\hat{\mathbf{Y}}d\mathbf{L}\right) \end{aligned} \tag{40}$$

So $\frac{\partial J_1}{\partial \mathbf{L}}$ can be calculated by:

$$\frac{\partial J_1}{\partial \mathbf{L}} = \frac{C}{\sigma^2}\hat{\mathbf{Y}}^T\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L}-\mathbf{Y}\right) \tag{41}$$

For the second term of Eq. (39), it is easy to find:

$$\frac{\partial J_3}{\partial \mathbf{L}} = \beta\left(\mathbf{L} - \mathbf{L}^T\right) \tag{42}$$

So far, $\frac{\partial J}{\partial \mathbf{L}}$ can be calculated by:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{L}} &= \frac{\partial J_1}{\partial \mathbf{L}} + \frac{\partial J_4}{\partial \mathbf{L}} \\ &= \frac{C}{\sigma^2}\hat{\mathbf{Y}}^T\mathbf{D}\left(\hat{\mathbf{Y}}\mathbf{L} - \mathbf{Y}\right) + \beta\left(\mathbf{L} - \mathbf{L}^T\right) \end{aligned} \tag{43}$$

References

[1] D. Hoang, H. Kang, A survey on Deep Learning based bearing fault diagnosis, Neurocomputing 335 (2019) 327–335.

[2] F. Jia, Y. Lei, L. Guo, J. Lin, S. Xing, A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines, Neurocomputing 272 (2018) 619–628.

[3] F. Jia, Y. Lei, N. Lu, S. Xing, Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization, Mech. Syst. Signal Process. 110 (2018) 349–367.

[4] H. Shao, H. Jiang, H. Zhang, W. Duan, T. Liang, S. Wu, Rolling bearing fault feature learning using improved convolutional deep belief network with compressed sensing, Mech. Syst. Signal Process. 100 (2018) 743–765.

[5] C. Lu, Z. Wang, W. Qin, J. Ma, Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification, Signal Process. 130 (2017) 377–388.

[6] S. Khan, T. Yairi, A review on the application of deep learning in system health management, Mech. Syst. Signal Process. 107 (2018) 241–265.

[7] S. Ma, F. Chu, Q. Han, Deep residual learning with demodulated time-frequency features for fault diagnosis of planetary gearbox under nonstationary running conditions, Mech. Syst. Signal Process. 127 (2019) 190–201.

[8] O. Abdeljaber, O. Avci, M.S. Kiranyaz, et al, 1-d cnns for structural damage detection: verification on a structural health monitoring benchmark data, Neurocomputing 275 (2018) 1308–1317.

[9] W. Mao, W. Feng, X. Liang, A novel deep output kernel learning method for bearing fault structural diagnosis, Mech. Syst. Signal Process. 117 (2019) 293–318.

[10] F. Dinuzzo, C.S. Ong, G. Pillonetto, P.V. Gehler, Learning output kernels with block coordinate descent, in: Proceedings of the International Conference on Machine Learning (ICML), 2011, pp. 49–56.

[11] J. Guo, D. Zhen, H. Li, Z. Shi, F. Gu, A. Ball, Fault feature extraction for rolling element bearing diagnosis based on a multi-stage noise reduction method, Measurement 139 (2019) 226–235.

[12] X. Yan, M. Jia, A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing, Neurocomputing 313 (2018) 47–64.

[13] W. Huang, H. Sun, J. Luo, W. Wang, Periodic feature oriented adapted dictionary free OMP for rolling element bearing incipient fault diagnosis, Mech. Syst. Signal Process. 126 (2019) 137–160.

[14] Y. Wang, L. Yang, J. Xiang, J. Yang, S. He, A hybrid approach to fault diagnosis of roller bearings under variable speed conditions, Measure. Sci. Technol. 28 (12) (2017) 125104.

[15] M. Lei, G. Meng, G. Dong, Fault detection for vibration signals on rolling bearings based on the symplectic entropy method, Entropy 19 (11) (2017) 607.

[16] J. Shang, M. Chen, H. Ji, D. Zhou, H. Zhang, M. Li, Dominant trend based logistic regression for fault diagnosis in nonstationary processes, Control Eng. Pract. 66 (2017) 156–168.

[17] H. Liu, J. Zhou, Y. Zheng, W. Jiang, Y. Zhang, Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders, ISA Trans. 77 (2018) 167–178.

[18] C. Shen, Y. Qi, J. Wang, G. Cai, Z. Zhu, An automatic and robust features learning method for rotating machinery fault diagnosis based on contractive autoencoder, Eng. Appl. Artif. Intell. 76 (2018) 170–184.

[19] H. Shao, H. Jiang, H. Zhao, F. Wang, A novel deep autoencoder feature learning method for rotating machinery fault diagnosis, Mech. Syst. Signal Process. 95 (95) (2017) 187–204.

[20] C. Lu, Z. Wang, W. Qin, J. Ma, Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification, Signal Process. 130 (2017) 377–388.

[21] Z. Zhu, G. Peng, Y. Chen, H. Gao, A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis, Neurocomputing 323 (2019) 62–75.

[22] H. Shao, H. Jiang, F. Wang, Y. Wang, Rolling bearing fault diagnosis using adaptive deep belief network with dual-tree complex wavelet packet, ISA Trans. 69 (2017) 187–201.

[23] S. Ma, F. Chu, Q. Han, Deep residual learning with demodulated time-frequency features for fault diagnosis of planetary gearbox under nonstationary running conditions, Mech. Syst. Signal Process. 127 (2019) 190–201.

[24] P. Zhou, X. Gu, J. Zhang, M. Fei, A priori trust inference with context-aware stereotypical deep learning, Knowl. Based Syst. 88 (2015) 97–106.

[25] Z. Cen, J. Wang, Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer, Energy 169 (2019) 160–171.

[26] Y. Wu, Y. Ma, D.P. Capaldi, J. Liu, W. Zhao, J. Du, L. Xing, Incorporating prior knowledge via volumetric deep residual network to optimize the reconstruction of sparsely sampled MRI, Magn. Reson. Imag. 66 (2020) 93–103.

[27] C. Lu, Z. Wang, W. Qin, J. Ma, Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification, Signal Process. 130 (2017) 377–388.

[28] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: explicit invariance during feature extraction, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 833–840.

[29] R. He, B. Hu, W. Zheng, X. Kong, Robust principal component analysis based on maximum correntropy criterion, IEEE Trans. Image Process. 20 (6) (2011) 1485–1494.

[30] W. Ma, H. Qu, G. Gui, L. Xu, J. Zhao, B. Chen, Maximum correntropy criterion based sparse adaptive filtering algorithms for robust channel estimation under non-Gaussian environments, J. Franklin Inst. Eng. Appl. Math. 352 (7) (2015) 2708–2727.

[31] Y. Li, Y. Yang, G. Li, M. Xu, W. Huang, A fault diagnosis scheme for planetary gearboxes using modified multi-scale symbolic dynamic entropy and mRMR feature selection, Mech. Syst. Signal Process. 91 (2017) 295–312.

[32] M. Cerrada, G. Zurita, D. Cabrera, R. Sanchez, M. Artes, C. Li, Fault diagnosis in spur gears based on genetic algorithm and random forest, Mech. Syst. Signal Process. 70 (2016) 87–103.

[33] J. Zhong, P.K. Wong, Z. Yang, Fault diagnosis of rotating machinery based on multiple probabilistic classifiers, Mech. Syst. Signal Process. 108 (2018) 99–114.

[34] H. Shao, H. Jiang, F. Wang, et al, Rolling bearing fault diagnosis using adaptive deep belief network with dual-tree complex wavelet packet, ISA Trans. 69 (2017) 187–201.

[35] Y. Zhang, K. Xing, R. Bai, et al, An enhanced convolutional neural network for bearing fault diagnosis based on time-frequency image, Measurement 157 (2020) 107667.

[36] J. Zhang, Y. Sun, L. Guo, et al, A new bearing fault diagnosis method based on modified convolutional neural networks, Chin. J. Aeronaut. 33 (2) (2020) 439–447.

[37] W. Huang, J. Cheng, Y. Yang, et al, An improved deep convolutional neural network with multi-scale information for bearing fault diagnosis, Neurocomputing 359 (2019) 77–92.

[38] T.W. Rauber, F.D.A. Boldt, F.M. Varejao, Heterogeneous feature models and feature selection applied to bearing fault diagnosis, IEEE Trans. Ind. Electron. 62 (1) (2015) 637–646.

[39] Case Western Reserve University Bearing Data Center, Seeded fault test data, Access at: https://csegroups.case.edu/bearingdatacenter/pages/download-data-file..

[40] J. Lee, H. Qiu, G. Yu, et al., IMS bearing data set from University of Cincinnati, Access at: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/..

[41] W. Mao, J. He, Y. Li, Y. Yan, Bearing fault diagnosis with auto-encoder extreme learning machine: a comparative study, Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci. 231 (8) (2017) 1560–1578.

[42] D.S. Wilks, Statistical Methods in the Atmospheric Sciences, second ed., Acadamic Press, Elsevier, USA, 2006.