

A domain adaptation model for early gear pitting fault diagnosis based on deep transfer learning network

Proc IMechE Part O:
J Risk and Reliability
2020, Vol. 234(1) 168–182
© IMechE 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/1748006X19867776
journals.sagepub.com/home/pio
 SAGE

Jialin Li¹ , Xueyi Li¹ , David He^{1,2} and Yongzhi Qu³

Abstract

In recent years, research on gear pitting fault diagnosis has been conducted. Most of the research has focused on feature extraction and feature selection process, and diagnostic models are only suitable for one working condition. To diagnose early gear pitting faults under multiple working conditions, this article proposes to develop a domain adaptation diagnostic model-based improved deep neural network and transfer learning with raw vibration signals. A particle swarm optimization algorithm and L2 regularization are used to optimize the improved deep neural network to improve the stability and accuracy of the diagnosis. When using the domain adaptation diagnostic model for fault diagnosis, it is necessary to discriminate whether the target domain (test data) is the same as the source domain (training data). If the target domain and the source domain are consistent, the trained improved deep neural network can be used directly for diagnosis. Otherwise, the transfer learning is combined with improved deep neural network to develop a deep transfer learning network to improve the domain adaptability of the diagnostic model. Vibration signals for seven gear types with early pitting faults under 25 working conditions collected from a gear test rig are used to validate the proposed method. It is confirmed by the validation results that the developed domain adaptation diagnostic model has a significant improvement in the adaptability of multiple working conditions.

Keywords

Early gear pitting, multiple working conditions, transfer learning, improved deep neural network

Date received: 16 December 2018; accepted: 20 June 2019

Introduction

Gears are some common transmission devices in machinery and widely used in aircraft, automobile, machine tools, and so on. In addition, due to the harsh working conditions, the gears have a higher fault rate. Gear faults include broken teeth, cracked teeth, and tooth pitting. Gear pitting fault is responsible for 31% of all faults.¹

In recent years, numerous research projects have been done on the diagnosis of gear pitting, which can be summarized into two types: model-based methods and data-driven methods.² In model-based methods, experts usually establish a dynamic modeling to simulate the system operation and then modify it based on the error between the actual outputs and the ideal outputs.³ Applying a model-based method requires not only a thorough understanding of the system, but also multiple parameter adjustments to optimize the model, and the accuracy of the model directly affect the diagnosis result. For example, Park et al.⁴ used finite element models of two gear faults to simulate the gear

operation to obtain transmission error and used it to identify the different characteristics. Shi et al.⁵ established a double motor torque and rotational speed coupling model to have a detailed simulation analysis on the situation that the experiment platform is difficult to realize or test. On the contrary, data-driven methods do not require much experience with the system, and we can use a model established by the data to diagnose the gear faults. The traditional data-driven methods typically involve three necessary processes: (1) feature

¹School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China

²Department of Mechanical and Industrial Engineering, The University of Illinois at Chicago, Chicago, IL, USA

³School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan, China

Corresponding author:

David He, Department of Mechanical and Industrial Engineering, The University of Illinois at Chicago, 842 West Taylor Street, Chicago, IL 60607, USA.

Email: davidhe@uic.edu

extraction, (2) feature selection, and (3) pattern recognition.⁶ Saravanan et al.⁷ used wavelet analysis to extract features from vibration signals and used two pattern recognition methods, artificial neural network (ANN), and proximal support vector machine (PSVM), to diagnose gearbox faults. Wu and Chan⁸ used acoustic emission signals instead of vibration signals for gear faults diagnosis, and a continuous wavelet transform technique combined with a feature selection of energy spectrum is used to generate the inputs of ANN. In the study by Samanta et al.,⁹ statistical features extracted from time domain signals were applied as the inputs of ANN and SVM. In addition, the genetic algorithm (GA) is applied for optimization. Traditional pattern recognition methods such as ANN and SVM can only achieve shallow learning tasks, and the diagnosis performance is directly affected by feature selection process.^{10,11} Moreover, the feature selection process is done manually, largely depending on prior diagnostic knowledge. And the feature selection method of one faulty diagnosis issue may not be applicable to another issue.

In recent years, enthusiasm for deep learning has been triggered by Hinton et al.¹² Deep learning can overcome the shortcomings of the shallow model. When it is applied to faults diagnosis, the feature selection process can be omitted, which can save time and labor. There are many different methods for deep learning, and according to the training method, it can be divided into two types: supervised training and unsupervised training.¹³ Methods for supervised training include deep neural network (DNN)¹⁴ and convolutional neural network (CNN).^{15,16} Methods for unsupervised training include deep belief network (DBN)^{17,18} and autoencoder (AE).^{19,20} Heydarzadeh et al.²¹ applied the discrete wavelet transform (DWT) results of three common monitoring signals (vibration, acoustic, and torque) as the inputs of the DNN to diagnose the five classes of gear faults. Sun et al.²² applied a dual-tree complex wavelet transform (DTCWT) to extract multi-scale features of signals. In addition, the CNN is applied for gear fault diagnosis. Shao et al.²³ also applied DTCWT for feature extraction and used adaptive deep belief network (ADBN) for fault diagnosis. Jia et al.²⁴ used AE technology to pre-train the parameters of the DNN to diagnose rotating machinery faults. Several of the references presented above used different deep learning methods to diagnose mechanical faults, but all include feature extraction process such as DWT. Manual feature extraction process is time-consuming and labor-intensive, and unsuitable extraction methods will also affect the diagnosis results. Jing et al.²⁵ proposed an adaptive gearbox faults diagnosis method based on deep convolutional neural network (DCNN), and there is no feature extraction process in the article, and the raw data collected from the experiment were directly applied as the inputs of the DCNN. Wang et al.²⁶ proposed the adaptive deep convolutional neural network (ADCNN)

method to diagnose bearing faults. Qu et al.²⁷ used the deep sparse autoencoder (SAE) method to diagnose gear pitting: the authors combined dictionary learning with sparse coding and then stacked it into the AE network and diagnosed two types of gear conditions (health, pitting) with raw data as the inputs of deep SAE.

The domain adaptability of the diagnostic model is also a key evaluation criterion. Ren et al.²⁸ proposed a new feature extraction method for diagnosing rolling bearing faults under varying speed conditions. Considering the increase in energy when the ball passes through the fault, the frequency values are divided by instantaneous speed and corresponding amplitude to form a new fault feature array, and the Euclidean distance classifier was used for recognition. Tong et al.²⁹ proposed domain adaptation using transferable features (DATF) to solve the diagnosis of different working conditions. They used maximum mean discrepancy (MMD) to reduce the marginal and conditional distributions simultaneously during domains across. Cheng et al.³⁰ first transformed the vibration signal into a recurrence plot (RP) with two dimensions and then utilized speed up robust feature to extract fault features considering the visual invariance characteristic of the human visual system (HVS). Liu et al.³¹ applied Hilbert–Huang transform (HHT), singular value decomposition (SVD), and Elman neural network to solve the bearing fault diagnosis under variable working conditions. This method is mainly used to apply the SVD method to reduce the dimension of the instantaneous amplitude matrix and obtain the insensitive fault feature. Zhang et al.³² applied the method of transfer learning (TL) to make diagnostic methods quickly adaptable to other working conditions.

Most of the aforementioned gear pitting fault diagnosis methods include feature extraction and feature selection process. Moreover, the conventional diagnostic model is only suitable for fault diagnosis under one working condition. This article proposes a newly developed DNN methodology for diagnosis of early gear pitting faults. Meanwhile, particle swarm optimization (PSO) algorithm and L2 regularization are used to optimize the traditional DNN. In addition, TL is combined to develop a deep transfer learning network (DTLN) to improve the domain adaptability of the diagnostic model. The innovation of the proposed method is that the feature extraction and selection process are omitted, and the domain adaptability of the network is improved. The rest of the article is organized as follows: in “The proposed method” section, the methodology of the proposed method is introduced. In “Experiment setup and data segmentation” section, the data collected from the experimental test rig and preprocess of the collected vibration data are explained. In “Results and discussions” section, the validation of proposed method using the collected vibration data is reported. Finally, “Conclusions” section concludes the article.

The proposed method

The improved deep neural network

Conventional DNN. DNN has a fully connected network structure: neurons in the adjacent layers are connected to each other, and neurons in same layer are not connected to each other. The forward propagation process of DNN is similar to that of ANN. The calculation principles of data passing through layer m in DNN are shown in equations (1)–(3)³³

$$u_k^m = \sum_{i=1}^n w_{ki}^m x_i^m \quad (1)$$

$$z_k^m = u_k^m - b_k^m \quad (2)$$

$$y_k^m = f(z_k^m) \quad (3)$$

where x_i^m is the i th input value of layer m , w_{ki}^m is the weight of layer m , u_k^m is the weighted sum of all inputs, b_k^m is bias vector, $f()$ is the activation function, and y_k^m is the output of layer m .

There are many activation functions available. The following introduces two commonly used in DNN sigmoid function and ReLU function as shown in equations (4) and (5). Equation (6) is the derivative function of ReLU³⁴

$$f_{\text{sig}} = \frac{1}{e^{-z_k} + 1} \quad (4)$$

$$f_{\text{ReLU}} = \max(0, z_k) \quad (5)$$

$$\frac{d}{dz} f_{\text{ReLU}} = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases} \quad (6)$$

Both activation functions have their own advantages and disadvantages. The output of sigmoid is from 0 to 1, so it can control the amplitude change in the deep learning. But it contains exponential calculation, so the amount of calculation is large. And when sigmoid is used as the activation function, with the increase in number of layers and neurons, the gradient and sparsity problems cannot be solved well. The advantage of the ReLU compared to the sigmoid is that it has better sparseness and can recognize the fault feature from the multi-scale signal features in deep learning. The derivative of ReLU is 1 or 0 so that the network can solve the problems of gradient descent and gradient explosion in a better way. However, the forced sparsity of ReLU also leads to neuron “necrosis,” resulting in the model that cannot extract valid features. Moreover, it cannot limit the amplitude like sigmoid activation function.

The last layer of the network has a softmax classifier as shown in equation (7). We will get the vector y after inputting the vector x to the DNN network. There must be an error between the actual output y and the desired output o , and we can use the error to modify the network weights and biases. There are two commonly used loss function: equation (8) is the mean square error loss function and equation (9) is the cross-entropy loss function³⁵

$$f_{\text{softmax}} = \frac{e^i}{\sum_j e^j} \quad (7)$$

$$E_{\text{MSE}} = \frac{1}{2} (o - y)^2 \quad (8)$$

$$E_{\text{cross-entropy}} = -([o \ln y] + (1 - o) \ln(1 - y)) \quad (9)$$

where o is the ideal output vector, y is the actual output vector, E_0 is the error of output vector.

Equations (10) and (11) modify the network weights and biases: the loss function partial derivative for the weights and biases is multiplied with the learning rate. Equations (12)–(14) demonstrate that the cross-entropy loss function can train the network faster than the traditional mean square error loss function

$$\Delta w = -\eta \frac{\partial E_{\text{MSE}}}{\partial w} \quad (10)$$

$$\Delta b = -\eta \frac{\partial E_{\text{MSE}}}{\partial b} \quad (11)$$

$$\frac{\partial E_{\text{MSE}}}{\partial w} = (o - y) f'_{\text{sig}}(wx + b)x \quad (12)$$

$$\frac{\partial E_{\text{cross-entropy}}}{\partial w} = (y - o)x \quad (13)$$

$$\frac{\Delta \text{MSE}}{\Delta \text{cross-entropy}} = \frac{|o - y| \cdot |f'_{\text{sig}}(wx + b)| \cdot |x|}{|y - o| \cdot |x|} \leq 0.25 \quad (14)$$

where Δw is the correction of weights, Δb is the correction of biases, and η is the learning rate.

When the cross-entropy function is chosen as the loss function and sigmoid as the activation function, the last layer of weights is corrected as shown in equation (15). Equation (16) can be obtained by applying equation (13) to equation (15), and equation (18) obtained after equation (17) is applied to equation (16). Similarly, the correction amount of biases Δb can be obtained as shown in equation (19)

$$\Delta w = -\eta \frac{\partial E_{\text{cross-entropy}}}{\partial w} \quad (15)$$

$$\begin{aligned} \frac{\partial E_{\text{cross-entropy}}}{\partial w} &= -\left(\frac{o}{y} \cdot \frac{\partial y}{\partial w} - \frac{1 - o}{1 - y} \cdot \frac{\partial y}{\partial w}\right) \\ &= -\left[\frac{o - y}{y(1 - y)} \cdot \frac{\partial y}{\partial w}\right] \end{aligned} \quad (16)$$

$$\frac{\partial y}{\partial w} = y(1 - y)x \quad (17)$$

$$\Delta w = -\eta(y - o)x \quad (18)$$

$$\Delta b = -\eta(y - o) \quad (19)$$

Improved DNN. We used the ELU function³⁶ instead of the activation function ReLU. The ELU function is shown in equation (20), and equation (21) is its derivative function. Figure 1 shows the graph of two activation functions. In Figure 1, X and Y are the input and output of the activation function, respectively. Comparing equations (6) and (20), we can see that the

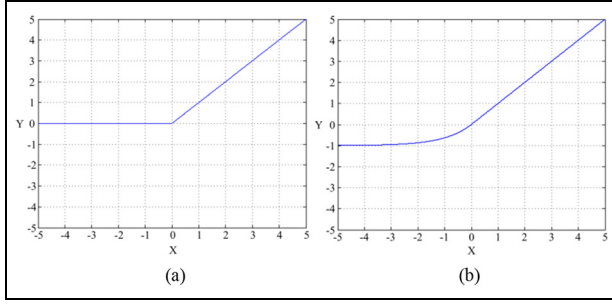


Figure 1. Comparison of two activation functions: (a) ReLU and (b) ELU.

ELU function has no change when z is more than 0, and changed when z is less than 0. Therefore, it retains the advantage of ReLU that prevents the gradient disappearing and saves the partly information of less than 0. Thus, the average of the neurons is closer to 0, and it can reduce the bias shift of the active unit. Since the soft saturation characteristic of the function is activated when the input value is small, the robustness to noise is improved

$$f_{\text{ELU}} = \begin{cases} z_k, & z > 0 \\ \alpha(e^{z_k} - 1), & z \leq 0 \end{cases} \quad (20)$$

$$\frac{d}{dz}f_{\text{ELU}} = \begin{cases} 1, & z > 0 \\ \alpha e^x, & z \leq 0 \end{cases} = \begin{cases} 1, & z > 0 \\ f_{\text{ELU}}(z_k) + \alpha, & z \leq 0 \end{cases} \quad (21)$$

In order to avoid overfitting of the DNN, L2 regularization³⁷ is used to correct the loss function, as shown in equation (22). Equations (23) and (24) reveal the nature of L2 regularization optimization. The L2 regularization term is added in the error function, and it will directly affect the network parameter correction. As shown in equations (23) and (24), the correction of the bias does not change, but the weight correction changes. Equation (26) is the final weight update function. It can be found that after the L2 regularization, the effect of weight decay is achieved because the weight is multiplied by a coefficient less than 1

$$E_{\text{new}} = -([o \ln y] + (1 - o)\ln(1 - y)) + \frac{\lambda}{2n} \sum_{w_{ki}} w_{ki}^2 \quad (22)$$

$$\frac{\partial E}{\partial w} = \frac{\partial E_0}{\partial w} + \frac{\lambda}{n} w \quad (23)$$

$$\frac{\partial E}{\partial b} = \frac{\partial E_0}{\partial b} \quad (24)$$

$$w_{\text{new}} = w - \eta \left(\frac{\partial E_0}{\partial w} + \frac{\lambda}{n} w \right) \quad (25)$$

$$w_{\text{new}} = \left(1 - \frac{\eta\lambda}{n} \right) w - \eta \frac{\partial E_0}{\partial w} \left(1 - \frac{\eta\lambda}{n} < 1 \right) \quad (26)$$

where E is the output error corrected by L2 regularization, λ is the coefficient of L2 regularization, and n is the sample size.

TL and fine-tuning strategy

The improved deep neural network (IDNN) can perform fault diagnosis well, but the trained network can only diagnose faults under one working condition. When the data under other working conditions are applied as inputs to diagnose the fault, a new diagnostic model needs to be trained to adapt this working condition. In practical applications, the working conditions of the equipment change over time. Training a network for each working condition requires not only a large amount of time, but also a large number of training samples. So a diagnostic model adapted to multiple working conditions is highly desirable.

TL^{38–40} is a popular approach in machine learning. It can be applied between two related domains to reduce training time and save training samples. Combining IDNN with TL to develop a DTLN can make the diagnostic model more adaptable to different working conditions. Figure 2 shows a comparison between traditional machine learning and TL. In traditional machine learning, each task or domain requires separate training of its corresponding diagnostic model. This not only requires a large number of training samples for each working condition, but also takes a lot of time.

The training process of TL shown in Figure 2 includes the source domain D_s and target domain D_t . The source domain in TL is the same as domain A in traditional machine learning. When the target domain is used for training, the pre-trained diagnostic model in the source domain is transferred to the target domain, and then the pre-trained model is fine-tuned with a small number of target domain samples. In this way, a domain diagnostic model can be used under multiple working conditions with only a small number of samples and less training time.

Particle swarm optimization

PSO is a method inspired by the behavior of birds searching for food, which was proposed by Kennedy and Eberhart. In previous papers,^{41,42} PSO algorithm was analyzed in detail. Similar algorithms include ant colony optimization (ACO)⁴³ and GA,⁴⁴ all of which are inspired by the behavior or laws of biology.

PSO is widely used, since it has a great adaptability, easy implementation, and few parameters to be set. Its basic principle can be described as n particles in a P -dimensional space, and their speed and location changed over time. The particle i of speed and position can be expressed by $v_i = (v_{i1}, v_{i2}, \dots, v_{ip})$ and $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$, P_f is the fitness of particle, and the size of the fitness corresponds to the distance between each bird and food. The extremum of individual P_b and extremum of population g_b can be updated according to particle fitness, and then we can use the individual extremum and the population extremum to

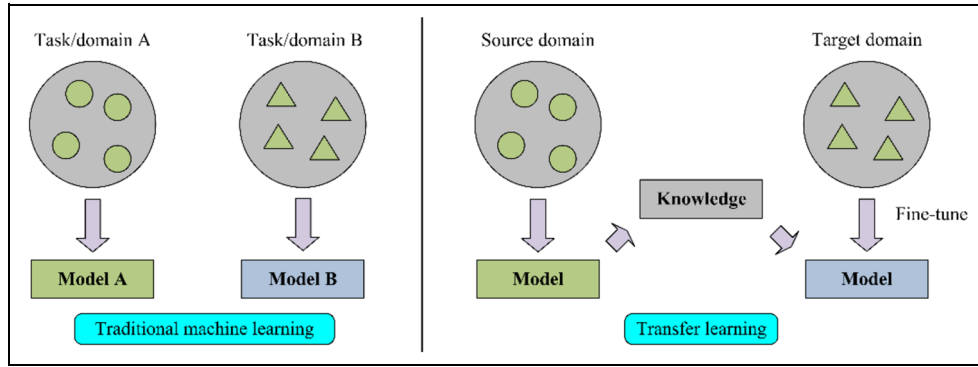


Figure 2. Different learning processes between traditional machine learning and transfer learning.

calculate the particle velocity and position, as shown in equations (27) and (28)

$$v_{ij}(t+1) = rv_{ij}(t) + c_1\varepsilon_1(P_{bj}(t) - x_{ij}(t)) + c_2\varepsilon_2(g_{bj} - x_{ij}(t)) \quad (27)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (28)$$

where i is the i th particle; j is the j th dimensional of the P -dimensional space; c_1 and c_2 are the learning factor; c_1 is the particle's own part, expresses its own understanding and influence on the optimization; c_2 is the social part, indicates that the particles are affected by the population; t is the number of iterations; ε_1 and ε_2 are random numbers that are evenly distributed between 0 and 1; and r is the inertia weight of particle, indicates that it is affected by the last speed.

PSO is used to optimize the parameters in the DNN. If the DNN contains a total of k parameters, the dimensional space j in equations (27) and (28) is equal to k . The number of particles is set empirically, and each particle contains j parameters. Select the best performing particle after t iterations and attach its j parameters to the DNN as the initial parameters. The weight determines the influence of the previous speed of the particle on the current speed, which plays a role in balancing the global search and the local search. As shown in equation (29), the weights are linear decay with iterations. This makes the particle swarm algorithm have strong search ability at the beginning of the iteration, and good local search ability in the later stage⁴⁵

$$r = r_{\max} - \frac{t}{t_{\max}}(r_{\max} - r_{\min}) \quad (29)$$

where r_{\max} is the set maximum weight, r_{\min} is the set minimum weight, and t_{\max} is the maximum number of iterations.

The position and velocity of the particles all have a range. When the velocity or position value is out of range, the processes as shown in equation (30) will be performed

$$v_{ij} = \begin{cases} v_{\max} & v_{ij} > v_{\max} \\ v_{\min} & v_{ij} < v_{\min} \end{cases}; \quad x_{ij} = \begin{cases} x_{\max} & x_{ij} > x_{\max} \\ x_{\min} & x_{ij} < x_{\min} \end{cases} \quad (30)$$

The range of particle velocity cannot be too large, otherwise the system will be unstable and it is easy to "skip" the optimal solution during particle iteration. Particle activity range setting is also similar to the speed setting, and limiting the particles' position helps find the optimal solution.

The initial position of the particles is randomly assigned within a certain range, and the optimal solution found by the several iteration may not be the global optimal solution. Therefore, the position of the particles should be mutated at a certain probability, which can increase the diversity of particles and find optimal solution in a new area. After repeating the above-mentioned operation several times, the global optimal solution can be found.

The framework and diagnostic process of DTLN

Figure 3 shows the framework of DTLN. It can be seen that the overall framework of DTLN is divided into two parts: (1) when the training and test data are in the same working condition, perform ①-② (purple circles marked in Figure 3) and (2) when the test data (target domain) are different from the training data (source domain), perform ①-③-④-⑤.

The detailed diagnostic process of the DTLN is defined as follow:

Step 1: Select one working condition data from all the collected data. Then, cut the raw data into n segmentations with the same amount of points. Finally, divide all segmentations into two groups, 80% of which is used for training and the remaining 20% for testing.

Step 2: Set the structure of the IDNN, set the minimum training error and the maximum epoch of training, and use the PSO algorithm to generate the initial weight and bias of IDNN.

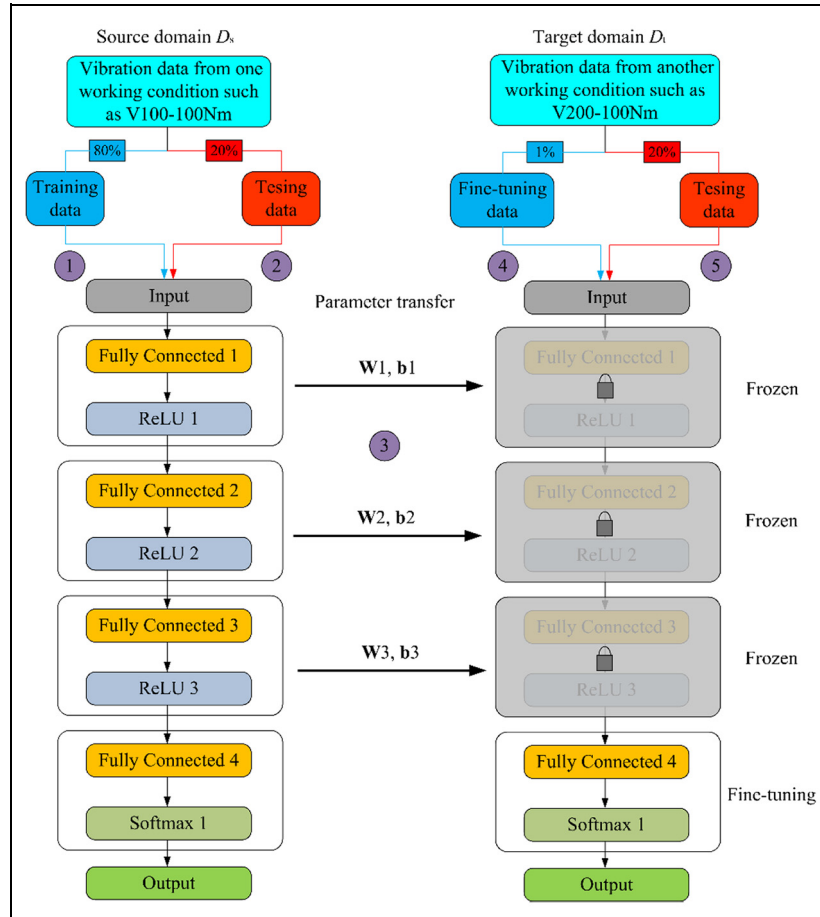


Figure 3. The framework of the deep transfer learning network.

Step 3: Randomly select a batch of segmentations as the inputs of the IDNN.

Step 4: Get the actual output through IDNN network, and use the cost function corrected by L2 regularization to calculate the error between the actual output and the ideal output.

Step 5: Compute the gradients of weights and biases in each layer with the back propagation algorithm, and update the weights and biases with the learning rate.

Step 6: Change another batch of segmentations to repeat Steps 3–5 until all the training data are used up.

Step 7: Repeat Steps 3–6 until training epochs reach the maximum epoch or the output error reaches the minimum set value.

Step 8: Test the trained network with the testing data. When the test working condition is the same as the selected working condition in *Step 1*, the trained network will be directly used for fault diagnosis. Otherwise, Steps 9–10 will be performed.

Step 9: Transfer the parameters of the trained IDNN to the new diagnostic model.

Step 10: Fine-tune the new diagnostic network with a small amount of data (fine-tuning with 1% of all data has a significant improvement) from the target domain. Finally, the fine-tuned model is used to diagnose the fault.

Experiment setup and data segmentation

Experiment setup

The experimental test rig and gear pitting type are shown in Figure 4. The gearbox is driven by two 45 kW Siemens servo motors: motor 1 is the drive motor and motor 2 is the load motor. The gearbox contains a pair of spur gears. The driving gear connected to the motor 1 has 40 teeth, the driven gear connected to the motor 2 has 72 teeth, and the gear module is 3 mm. The gearbox was also equipped with a lubrication and cooling system, and the vibration sensor is mounted on the bearing housing of the driven gear.

Table 1 describes the gear pitting condition in Figure 4. Six different early pitting were designed manually by a drill on the driven gear, and the degree of gear pitting is gradually increased, as shown in Table 1. The setting of gear pitting fault simulates the process of gear pitting from small to large and can also analyze the relationship between pitting type and fault diagnostic accuracy.

This article proposes to establish a gear pitting diagnosis model suitable for various working conditions, so the vibration data of various working conditions are collected to construct and test the model. In the experiment, vibration signal under five speed conditions and five torque conditions are collected, a total of 25

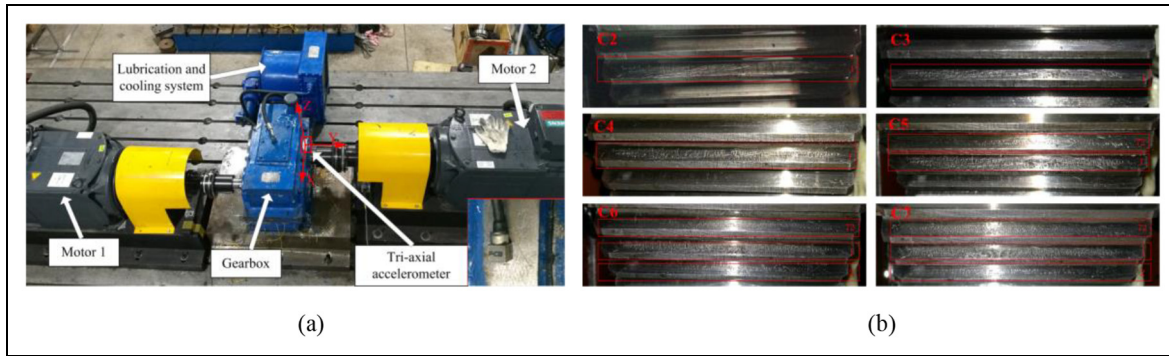


Figure 4. (a) Experimental test rig and (b) gear pitting type.

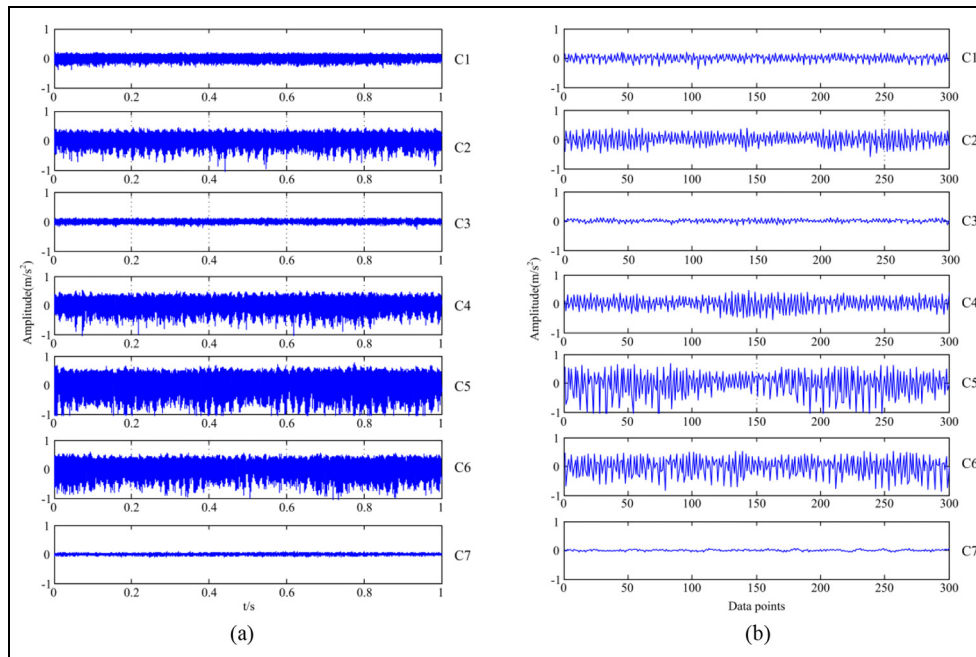


Figure 5. The vibration signal of 100 r/min-100 Nm: (a) one second signals and (b) one segmentation signals.

working conditions, as shown in Table 2. Note that the circles in Table 2 represent the six conditions used in the mixed working condition diagnosis in “Diagnosis results of IDNN under multiple working conditions” and “Diagnostic results with DTLN” sections.

Data segmentation

The tri-axial accelerometer was mounted on the bearing housing of the driven gear and collected vibration signals in all the three directions, with a sampling rate of 10,240 Hz. In this article, the vibration signals of seven kinds of gears under 25 working conditions are collected. Comparing the vibration signals of all the three directions, the amplitude of Z-axis is the largest. Therefore, we use the Z-axis vibration signal in the diagnosis of gear pitting faults. The vibration signal in the Z-axis of 100 r/min-100 Nm working condition is shown in Figure 5(a).

We collected vibration signal five times in each gear fault type (C1–C7). So there are 35 files in each working condition and 60,000 data points per file. The number of data points in each file is too large to be directly used as input to the DNN, so we cut the raw signal into suitable segmentation. The advantage of data segmentation is that the number of neurons in input layer is reduced, which in turn reduces the complexity of the DNN structure and makes the network fit more quickly. On the contrary, the training sample size and sample diversity is increased, and the diagnostic accuracy of the network is improved.

The sampling rate is 10,240 Hz and the max rotation speed is 500 r/min, so approximately 1200 data points per gear rotation can be computed. We put 300 data points (quarter of per gear rotation collected data) in each segmentation.⁴⁶ So each file is divided into 200 segmentations, a total of 7000 segmentations. About 80% of all data are used for training and the rest is

Table 1. Driven gear pitting type.

Label	Gear pitting type		
	72nd tooth	First tooth	Second tooth
C1	Healthy	Healthy	Healthy
C2	Healthy	10% in middle	Healthy
C3	Healthy	30% in middle	Healthy
C4	Healthy	50% in middle	Healthy
C5	10% in middle	50% in middle	Healthy
C6	10% in middle	50% in middle	10% in middle
C7	30% in middle	50% in middle	10% in middle

Table 2. Experimental working conditions.

Speed (r/min) \ Torque (Nm)	100	200	300	400
100	○	▲	○	▲
200	▲	▲	▲	▲
300	○	▲	▲	▲
400	▲	▲	▲	▲
500	○	▲	▲	▲

used for testing. The diagnostic model training matrix dimension for each working condition is 300×5600 , and the testing matrix dimension for each working condition is 300×1400 .

Results and discussions

Diagnosis results of IDNN under working condition 100 r/min-100 Nm

First, we should decide the structure of the IDNN: the number of neurons in input layer is equal to the number of data points in segmentation (300 neurons), seven neurons in the output layer (corresponding to seven gear types), and contained three hidden layers (300, 200, and 100 neurons). The minimum training error is set to 0.01 and the maximum training epochs is set to 150. All samples are randomly branched, and then each branch is trained in turn, and one training epoch is completed when all branches are trained.

Figure 6(a) shows the effect of PSO on training. By comparison, it is found that after PSO optimization, the initial error is reduced from 25 to 2, and the number of training epoch is also greatly reduced, which means that PSO optimization can shorten the training process and make the training process more stable.

Table 3 shows the effect of PSO algorithm on training time and training accuracy. The term *NAN* in the table indicates that the network does not converge. The PSO algorithm allows the network to start with good initial parameters. In this case, it is possible to choose a larger learning rate and speed up the network convergence.

Figure 6(b) shows the influence of the magnitude of the L2 coefficient λ on the diagnostic accuracy. It can be seen from the figure that when λ is equal to 0, that is, there is no L2 optimization, the accuracy is about 0.9. As the value of λ increases, the accuracy shows an upward trend. The accuracy reaches the maximum value of 0.96386 when λ is equal to 0.35. As L2 coefficient λ continues to increase, the fluctuation of the accuracy becomes larger, that is, the stability of the diagnostic model decreases.

The confusion matrixes of standard DNN (SDNN) method and IDNN are shown in Figure 7. The activation function ReLU is used in the standard DNN. It can be seen that the improved method has a better diagnostic accuracy. The misdiagnosis of the two methods is consistent (case1: C2 misjudge as C4, case2: C2 misjudge as C6, case3: C5 misjudge as C6, case4: C6 misjudge as C4). The initial judgment of misdiagnosis is due to the occasional single-tooth engagement of the gearbox resulting in a change in the type of fault.

The diagnostic accuracy of four methods for diagnosing gear pitting faults under the 100 r/min-100 Nm is shown in Table 4. When the SVM and ANN methods were used, 12 statistical characteristics (mean, root mean square (RMS), variance, etc.) were extracted from the time domain and frequency domain. On the contrary, the standard DNN method and proposed method used the raw vibration signal as the input.

The fault type of the gear is the type corresponding to the neuron with maximum value. The diagnostic

Table 3. The effect of PSO on the training time and diagnostic accuracy.

	Learning rate	PSO time	Network computing time	Total	Accuracy
With PSO	0.1	20.817 s	39.61 s	60.42 s	0.9364
Without PSO	0.1	—	NAN	NAN	0.1429
Without PSO	0.05	—	136.644 s/stop in max epochs (250)	136.64 s	0.8364

PSO: particle swarm optimization.

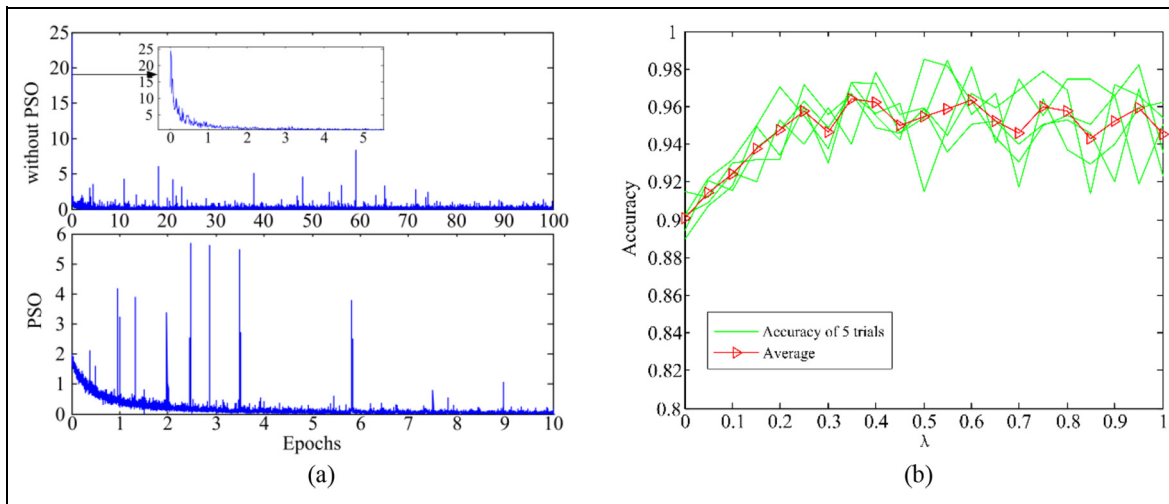


Figure 6. Training error curve of hybrid model: (a) influence of PSO and (b) influence of L2 coefficient λ .

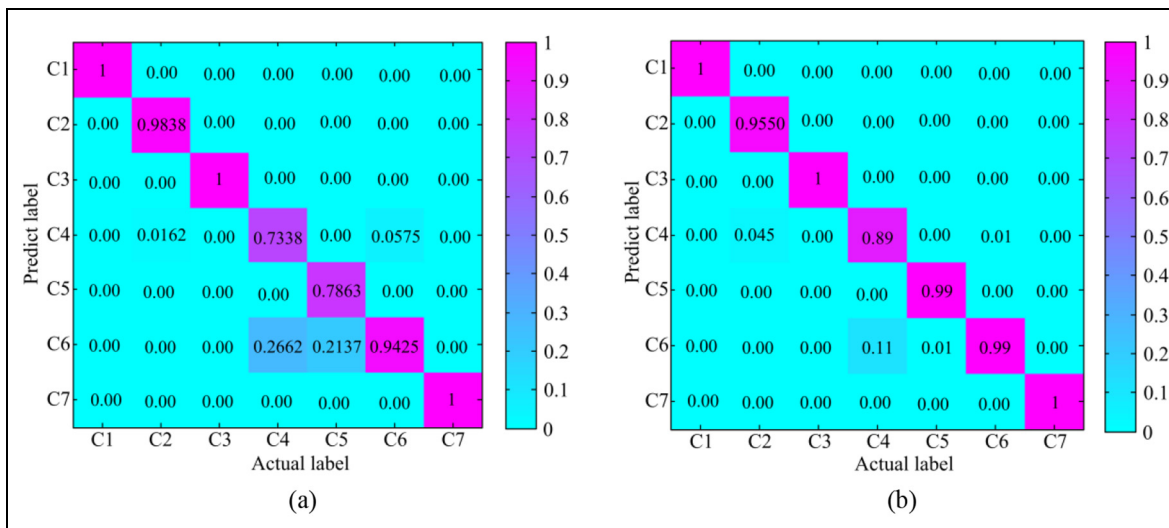


Figure 7. Confusion matrixes: (a) SDNN and (b) IDNN.

accuracy has not differed when the maximum output of neurons is 0.5 and 0.99. Therefore, the diagnostic accuracy cannot fully represent the diagnostic ability of the network. We perform principal component analysis (PCA) on the output matrix of the network to further analyze the diagnostic ability of three methods, and then used the first two principal components (PCs) of the PCA results to form a scatter figure, as shown in Figure 8. The diagnostic accuracy of Figure 8(a) and (b) is similar, but from the PCA results, we can know the diagnostic ability of SDNN method is significantly better than ANN method. Compared with the SDNN method, diagnostic ability of the IDNN has also improved significantly.

The parameter setting during training also affects the diagnostic accuracy. Figure 9 shows the effect of parameters learning rate and batch size (samples in each batch). Figure 9(a) and (b) shows that as the learning rate and batch size increase, the accuracy decreases.

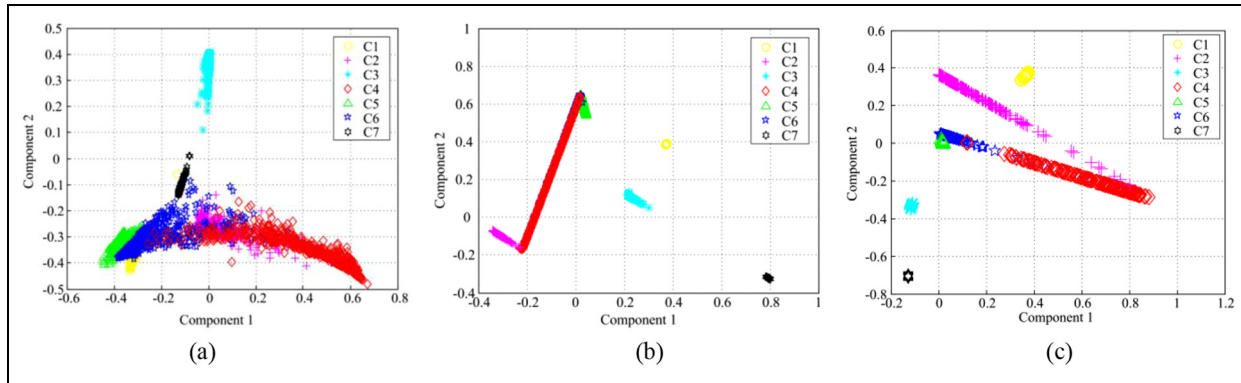
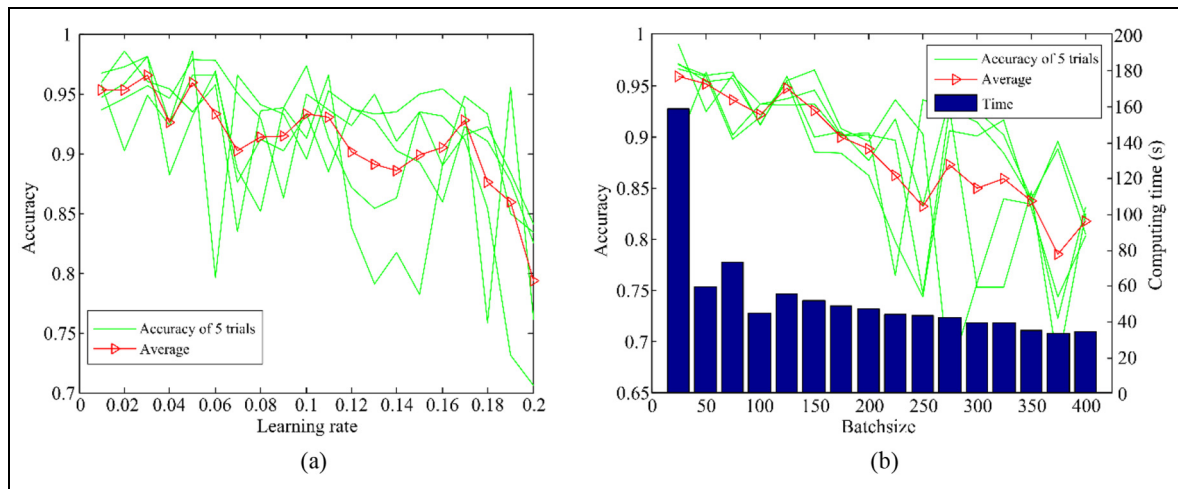
Diagnosis results of IDNN under multiple working conditions

“Diagnosis results of IDNN under working condition 100 r/min-100 Nm” section shows the results of applying IDNN to diagnose gear faults under 100 r/min-100 Nm working conditions. This section applies a variety of working conditions to verify the adaptability of IDNN for diagnosing multiple working conditions. Figure 10(a), (c), and (e) shows the diagnostic accuracy of the three methods (SVM, ANN, and IDNN) in 25 working conditions (as shown in Table 2). It can be found from Figure 10(e) that IDNN method has a high accuracy under each working condition, but it is necessary to retrain the network when the working conditions change. Figure 10(b), (d), and (f) shows the cross-diagnosis accuracy of six working conditions (labeled as circles in Table 2) without retraining the network. It can be seen from Figure 10(f) that the diagnostic

Table 4. Diagnostic accuracy of four methods.

Method	C1	C2	C3	C4	C5	C6	C7	Average
SVM	0.995	0.865	0.975	0.755	0.665	0.765	0.995	0.8594
ANN	0.995	0.895	0.995	0.765	0.845	0.99	I	0.9264
SDNN	I	0.985	I	0.735	0.785	0.945	I	0.9214
IDNN	I	0.955	I	0.89	0.99	0.99	I	0.975

SVM: support vector machine; ANN: artificial neural network; SDNN: standard deep neural network; IDNN: improved deep neural network.

**Figure 8.** The PCA result of three kinds of network outputs: (a) ANN, (b) SDNN, and (c) improved DNN.**Figure 9.** Influence of the network parameter on diagnostic accuracy: (a) learning rate and (b) batch size.

accuracy is better only when the training and testing data are from the same working condition. In other words, a trained IDNN developed under one working condition is only applicable to the same working condition and cannot be used in other working conditions.

Diagnostic results with DTLN

As can be seen from Figure 10, IDNN has a good diagnostic accuracy under each working condition.

However, a well-trained IDNN under one working condition can only diagnose the data under this condition. In order to improve the working condition adaptability of the diagnostic model, this article proposes a DTLN based on TL. This section applies six working conditions (labeled as circles in Table 2) to test the adaptability of the DTLN. The six working conditions are as follows: *A*: 100 r/min-100 Nm, *B*: 100 r/min-300 Nm, *C*: 100 r/min-500 Nm, *D*: 300 r/min-100 Nm, *E*: 500 r/min-100 Nm, and *F*: 500 r/min-500 Nm.

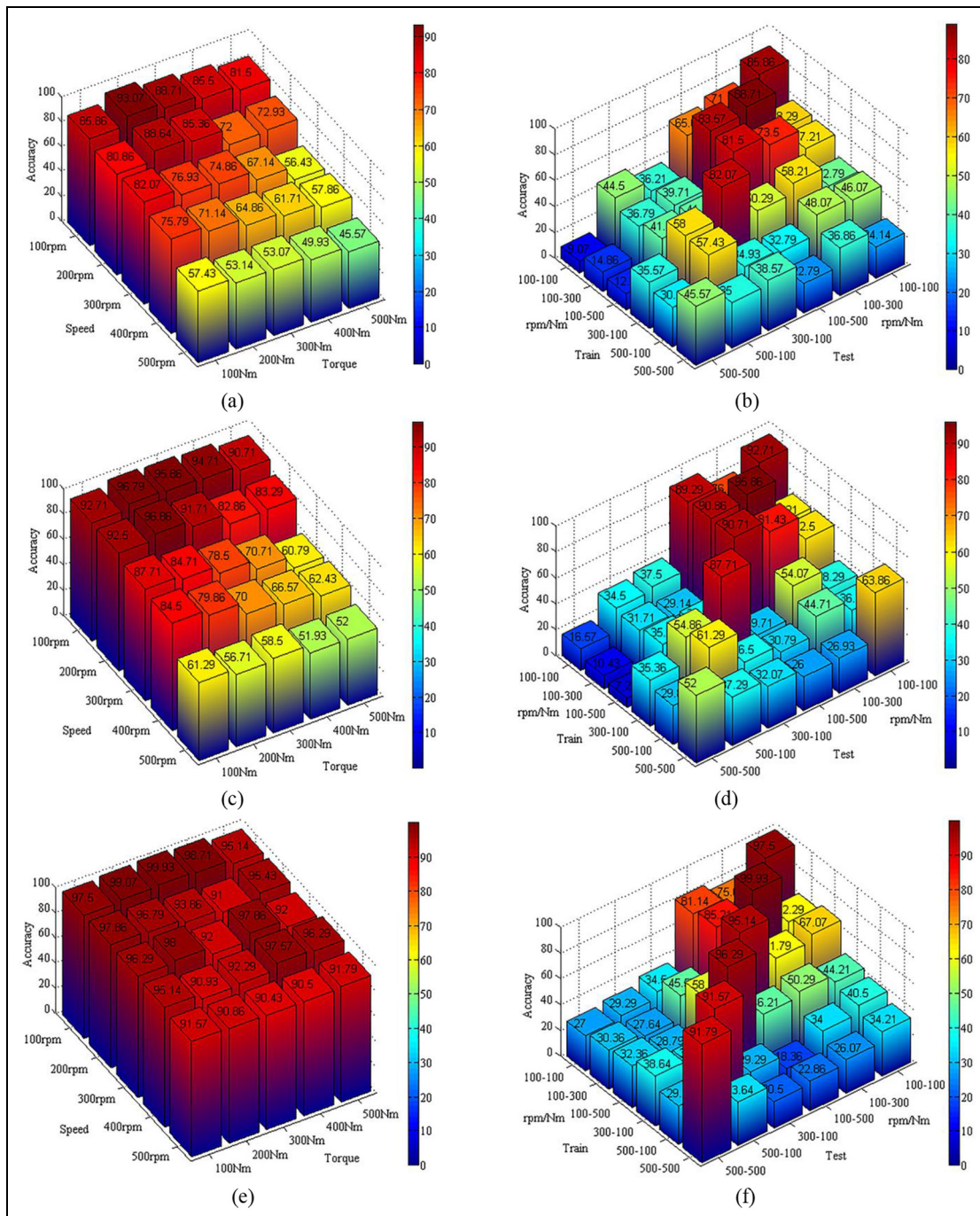


Figure 10. Diagnostic accuracy of different methods: (a), (b) SVM method; (c), (d) ANN method; (e), (f) IDNN method. The training data and test data used in (a), (c), and (e) are from the same working condition; the training data and test data used in (b), (d), and (f) are from different working conditions.

Figure 11 shows the diagnostic accuracy changes corresponding to an increase in the training sample size of both DTLN and IDNN. The horizontal axis is the target domain sample size used to fine-tune the pre-trained network. The data in working condition A and data in working condition B were used as training data for the results in Figure 11(a) and (b), respectively. The four curves in Figure 11(a) correspond to four cases: (1) case 1 ($A-A$ with IDNN, all samples are used): trains the network with 80% of the data in working condition

A and test with the remaining data in working condition A ; (2) case 2 ($A-B$ with DTLN, with different training sample size): uses DTLN to diagnose faults, where the source domain D_s was used as data in working condition A and the target domain D_t as data in working condition B . As discussed in “Data segmentation” section, the number of samples in each working condition was 7000. Setting the percentage of the data for fine-tuning from 0.1% to 2%, the fine-tuning sample size used was changed from seven ($7000 \times 0.1\%$) to 140

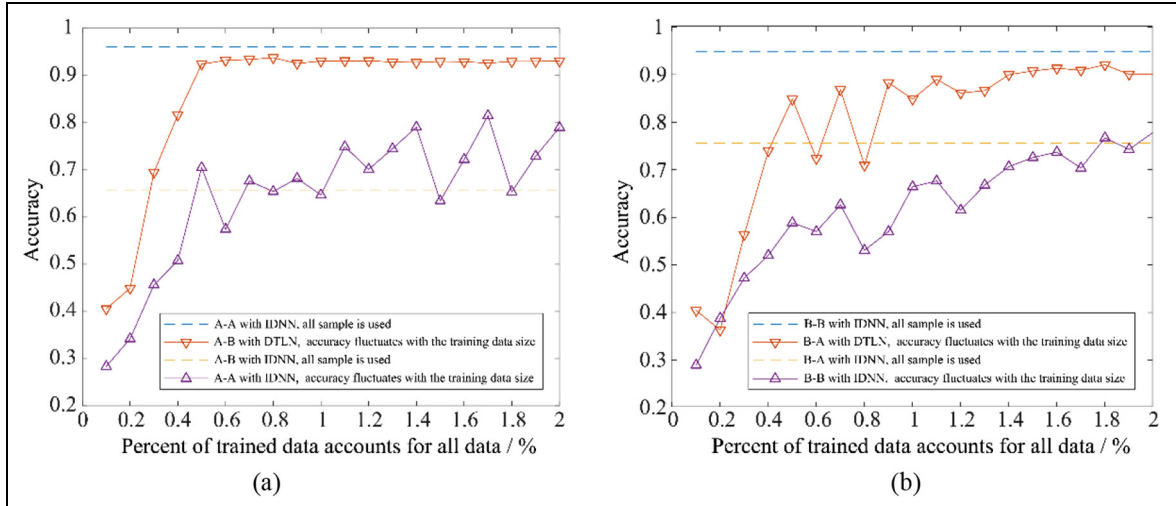


Figure 11. The accuracy changes corresponding to the changes in training sample size for DTLN and IDNN: (a) source domain: working condition A, target domain: working condition B; (b) source domain: working condition B, target domain: working condition A.

(7000 \times 2%). (3) Case 3 (*A-B* with IDLN, all samples are used): trains the network with 80% of the data in working condition *A*, and then test the trained network with 20% data from working condition *B*. (4) Case 4 (*A-A* with IDLN, accuracy fluctuates with training sample size): trains the network with data sample size from seven to 140 in working condition *A*, and then tests the trained network with 20% data in working condition *A*. In Figure 11(b), data in working condition *B* were used as the source domain and data in working condition *A* were used as the target domain. As can be seen from the figure, the DTLN can achieve high diagnostic accuracy with 1% of data for fine-tuning.

Figure 12 shows a comparison of the diagnostic accuracy of DTLN and IDNN with different target domains. Taking Figure 12(a) as an example, the data in working condition *A* as source domain were used to train the model and the data in other five working conditions (*B* to *F*) were used as the target domain to test the model. When using the DTLN method, 5% of the target domain data were used to fine-tune the pre-trained model. Comparing the diagnostic accuracy of the two methods, it can be found that the DTLN is significantly more adaptable to different working conditions than IDNN. The DTLN method not only improves the diagnostic accuracy under multiple working conditions, but also requires fewer training samples and less training time. The IDNN required 67 s to train the model with 80% data in working condition *A*. When the data in working condition *B* were used as the target domain, it took 72 s to develop the model with 80% data in working condition *B*. However, using the DTLN method to fine-tune the model required only 6 s, which reduced the training time by a factor of 10.

In summary, DTLN can not only make model adapt to multiple working conditions, but also save training time and samples.

Conclusions

In this article, a domain adaptation model for early gear pitting fault diagnosis based on deep TL was presented. By combining an IDNN with TL, DTLN was developed to make the diagnostic model have a good diagnostic accuracy under multiple working conditions. The vibration signals for seven types of gears with early pitting faults under 25 working conditions collected from a gear test rig were used to validate the DTLN. Based on the validation results, we can draw the following conclusions:

1. Using PSO optimization to initialize model parameters speeds up the training process. L2 regularization improves the diagnostic ability of the diagnostic model by weight decay during training.
2. The IDNN has a high diagnostic accuracy when the target domain (testing data) and the source domain (training data) are in the same working condition, and the maximum accuracy can reach 99.93%. However, a diagnostic model developed with IDNN is only suitable for fault diagnosis under the same working condition.
3. The DTLN overcomes the shortcomings of IDNN, and greatly improves the adaptability of the diagnostic model to multiple working conditions. Moreover, to fine-tune the pre-trained model, only a small number of target samples and less training time are required.

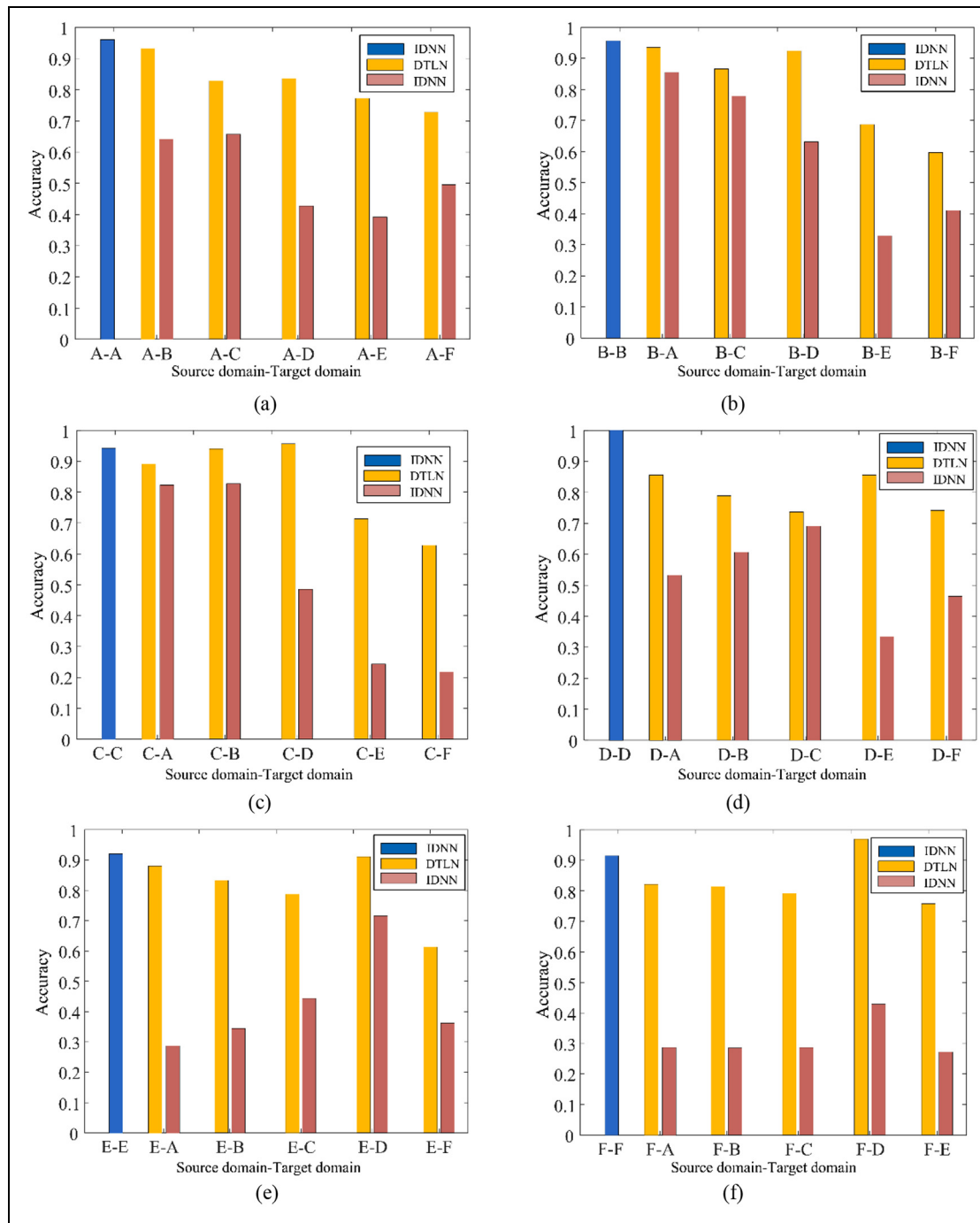


Figure 12. Comparison of diagnostic accuracy between DTLN and IDNN: (a) source domain: working condition A, (b) source domain: working condition B, (c) source domain: working condition C, (d) source domain: working condition D, (e) source domain: working condition E, and (f) source domain: working condition F.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or


publication of this article: This work was funded by the National Natural Science Foundation of China (No. 51675089 and No. 51505353).

ORCID iDs

Jialin Li <https://orcid.org/0000-0002-9940-179X>

Xueyi Li <https://orcid.org/0000-0002-1751-2809>

David He <https://orcid.org/0000-0002-5703-6616>

Yongzhi Qu  <https://orcid.org/0000-0002-5314-023X>

References

1. Ali YH, Rahman RA and Hamzah RIR. Acoustic emission signal analysis and artificial intelligence techniques in machine condition monitoring and fault diagnosis: a review. *J Teknologi* 2014; 69(2): 121–126.
2. Lei Y, Liu Z, Wang D, et al. A probability distribution model of tooth pits for evaluating time-varying mesh stiffness of pitting gears. *Mech Syst Signal Pr* 2018; 106: 355–366.
3. Liu X, Yang Y and Zhang J. Resultant vibration signal model based fault diagnosis of a single stage planetary gear train with an incipient tooth crack on the sun gear. *Renew Energ* 2018; 122: 65–79.
4. Park S, Kim S and Choi JH. Gear fault diagnosis using transmission error and ensemble empirical mode decomposition. *Mech Syst Signal Pr* 2018; 108: 262–275.
5. Shi X, Gao Q, Li W, et al. Simulation study on gear fault diagnosis simulation test-bed of doubly fed wind generator. In: *Proceedings of the 12th international conference on computer science and education*, Houston, TX, 22–25 August 2017. New York: IEEE.
6. Sun W, Shao S, Zhao R, et al. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement* 2016; 89: 171–178.
7. Saravanan N, Siddabattuni VNSK and Ramachandran KI. Fault diagnosis of spur bevel gear box using artificial neural network (ANN), and proximal support vector machine (PSVM). *Appl Soft Comput* 2010; 10(1): 344–360.
8. Wu JD and Chan JJ. Faulted gear identification of a rotating machinery based on wavelet transform and artificial neural network. *Expert Syst Appl* 2009; 36(5): 8862–8875.
9. Samanta B, Al-Balushi KR and Al-Araimi SA. Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. *Eng Appl Artif Intel* 2003; 16: 657–665.
10. Keskes H and Braham A. Recursive undecimated wavelet packet transform and dag SVM for induction motor diagnosis. *IEEE T Ind Inform* 2015; 11(5): 1059–1066.
11. Bi TS, Ni YX, Shen CM, et al. A novel ANN fault diagnosis system for power systems using dual GA loops in ANN training. In: *Proceedings of the 2000 power engineering society summer meeting*, Seattle, WA, 16–20 July 2000. New York: IEEE.
12. Hinton GE, Osindero S and Teh YW. A fast learning algorithm for deep belief nets. *Neural Comput* 2006; 18(7): 1527–1554.
13. Liu W, Wang Z, Liu X, et al. A survey of deep neural network architectures and their applications. *Neurocomputing* 2017; 234: 11–26.
14. Lu W, Wang X, Yang C, et al. A novel feature extraction method using deep neural network for rolling bearing fault diagnosis. In: *Proceedings of the 27th Chinese control and decision conference*, Qingdao, China, 23–25 May 2015. New York: IEEE.
15. Chen ZQ, Li C and Sanchez RV. Gearbox fault identification and classification with convolutional neural networks. *Shock Vib* 2015; 2015(2): 390134.
16. Lu C, Wang Z and Zhou B. Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification. *Adv Eng Inform* 2017; 32: 139–151.
17. Zhang Z and Zhao J. A deep belief network based fault diagnosis model for complex chemical processes. *Comput Chem Eng* 2017; 107: 395–407.
18. Ren H, Chai Y, Qu JF, et al. A novel adaptive fault detection methodology for complex system using deep belief networks and multiple models: a case study on cryogenic propellant loading system. *Neurocomputing* 2018; 275: 2111–2125.
19. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw* 2015; 61: 85–117.
20. Shao HD, Jiang HK, Lin Y, et al. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. *Mech Syst Signal Pr* 2018; 102: 278–297.
21. Heydarzadeh M, Kia SH, Nourani M, et al. Gear fault diagnosis using discrete wavelet transform and deep neural networks. In: *IECON 2016 – 42nd annual conference of the IEEE industrial electronics society*, Florence, 23–26 October 2016. New York: IEEE.
22. Sun W, Yao B, Zeng N, et al. An intelligent gear fault diagnosis methodology using a complex wavelet enhanced convolutional neural network. *Materials* 2017; 10(7): 790.
23. Shao HD, Jiang HK, Wang F, et al. Rolling bearing fault diagnosis using adaptive deep belief network with dual-tree complex wavelet packet. *ISA T* 2017; 69: 187–201.
24. Jia F, Lei Y, Lin J, et al. Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech Syst Signal Pr* 2016; 72–73: 303–315.
25. Jing L, Wang T, Zhao M, et al. An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox. *Sensors* 2017; 17(2): E414.
26. Wang F, Jiang HK, Shao HD, et al. An adaptive deep convolutional neural network for rolling bearing fault diagnosis. *Meas Sci Technol* 2017; 28(9): 095005.
27. Qu YZ, He M, Deutsch J, et al. Detection of pitting in gears using a deep sparse autoencoder. *Appl Sci* 2017; 7(5): 515.
28. Ren Y, Li W, Zhu ZC, et al. A new fault feature for rolling bearing fault diagnosis under varying speed conditions. *Adv Mech Eng* 2017; 9(6). DOI: 10.1177/1687814017703897.
29. Tong Z, Li W, Zhang B, et al. Bearing fault diagnosis based on domain adaptation using transferable features under different working conditions. *Shock Vib* 2018; 2018: 6714520.
30. Cheng Y, Zhou B, Lu C, et al. Fault diagnosis for rolling bearings under variable conditions based on visual cognition. *Materials* 2017; 10(6): E582.
31. Liu HM, Wang X and Lu C. Rolling bearing fault diagnosis under variable conditions using Hilbert-Huang transform and singular value decomposition. *Math Probl Eng* 2014; 2014: 765621.
32. Zhang R, Tao H, Wu L, et al. Transfer learning with neural networks for bearing fault diagnosis in changing working conditions. *IEEE Access* 2017; 5: 14347–14357.

33. Wang Z, Wang J and Wang Y. An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition. *Neurocomputing* 2018; 310: 132–222.
34. Cui JL, Qiu S, Jiang MY, et al. Text classification based on ReLU activation function of SAE algorithm. In: *Proceedings of the international symposium on neural networks*, Hokkaido, Japan, 21–26 June 2017, pp.44–50. Cham: Springer.
35. Ye J. Fault diagnosis of turbine based on fuzzy cross entropy of vague sets. *Expert Syst Appl* 2009; 36(4): 8103–8106.
36. Clevert DA, Unterthiner T and Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). In: *Proceedings of the international conference on learning representations*, San Juan, PR, 2–4 May 2016, <https://arxiv.org/abs/1511.07289>
37. Zhao M, Chow TWS, Zhang H, et al. Rolling fault diagnosis via robust semi-supervised model with capped l2, l1-norm regularization. In: *Proceedings of the IEEE international conference on industrial technology*, Toronto, ON, Canada, 22–25 March 2017. New York: IEEE.
38. Shao SY, McAleer S, Yan RQ, et al. Highly accurate machine fault diagnosis using deep transfer learning. *IEEE T Ind Inform* 2018; 15: 2466–2455.
39. Cao P, Zhang S and Tang J. Pre-processing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning. *IEEE Access* 2018; 6: 26241–26253.
40. Qian WW, Li SM and Wang JR. A new transfer learning method and its application on rotating machine fault diagnosis under variant working conditions. *IEEE Access* 2018; 6: 69907–69917.
41. Mohammadi N and Mirabedini SJ. Comparison of particle swarm optimization and backpropagation algorithms for training feedforward neural network. *J Math Comp Sci* 2014; 12: 113–123.
42. Kulkarni VR and Desai V. ABC and PSO: a comparative analysis. In: *Proceedings of the IEEE international conference on computational intelligence & computing research*, Chennai, India, 14–16 December 2017. New York: IEEE.
43. Chen L, Xiao C, Li X, et al. A seismic fault recognition method based on ant colony optimization. *J Appl Geophys* 2018; 152: 1–8.
44. Rajeswari C, Sathiyabhama B, Devendiran S, et al. A gear fault identification using wavelet transform, rough set based GA, ANN and C4.5 algorithm. *Procedia Engineer* 2014; 97: 1831–1841.
45. Fang H. Monopole-gear design based on neural network and modified particle swarm optimization. *Appl Mech Mater* 2013; 477–478: 368–373.
46. Zhang R, Peng Z, Wu L, et al. Fault diagnosis from raw sensor data using deep neural networks considering temporal coherence. *Sensors* 2017; 17(3): E549.