



OPEN

## Deep residual neural-network-based robot joint fault diagnosis method

Jinghui Pan<sup>1</sup>✉, Lili Qu<sup>2</sup> & Kaixiang Peng<sup>1</sup>

A data driven method-based robot joint fault diagnosis method using deep residual neural network (DRNN) is proposed, where Resnet-based fault diagnosis method is introduced. The proposed method mainly deals with kinds of fault types, such as gain error, offset error and malfunction for both sensors and actuators, respectively. First, a deep residual network fault diagnosis model is derived by stacking small convolution cores and increasing the core size. meanwhile, the gaussian white noise is injected into the fault data set to verify the noise immunity for the proposed deep residual network. Furthermore, a simulation is conducted, where different fault diagnosis methods including support vector machine (SVM), artificial neural network (ANN), convolutional neural network (CNN), long-term memory network (LTMN) and deep residual neural network (DRNN) are compared, and the simulation results show the accuracy of fault diagnosis for robot system using DRNN is higher, meanwhile, DRNN needs less model training time. Visualization analysis proved the feasibility and effectiveness of the proposed method for robot joint sensor and actuator fault diagnosis using DRNN method.

Recently, industrial robots have been widely used on many applications, such as automobile production line, aerospace, communication, and consumer electronics<sup>1–3</sup>. As an outstanding representatives of mechatronics technology, robot joint module integrates a large number of components, including hollow motor, servo driver, harmonic reducer, brake, encoder, into a limited space<sup>4</sup>. Considering complex and changeable working environment of robot joints, it is inevitable that kinds of faults will occur. If there is no fault diagnosis mechanism before fault happens, it will affect the production efficiency, product quality, and even expose human life in danger. Therefore, how to detect and locate the faults quickly and accurately is the most urgent thing<sup>5</sup>.

Researchers have been focused on Fault detection and fault tolerant control of robot joints for many years, and they have proposed a lot of practical fault diagnosis methods, including hardware redundancy and theoretical analysis-based fault diagnosis methods.

Among theoretical analysis-based robot joints fault diagnosis methods, observer is widely used<sup>6,7</sup>. Due to the fast convergence characteristic of sliding mode method, the error could attenuate as designed, therefore, it ensures the rapidity of observer, so it is used at all places in robot joint fault diagnosis<sup>8,9</sup>. In addition, backstepping algorithm, Takagi–Sugeno method as well as Luenberger observer are also applied for fault diagnosis<sup>10–12</sup>. However, most industrial robots are affected by disturbance or noise, so it is necessary to consider the effect of disturbance in robot fault diagnosis. For robot system, the first thing that comes to mind is to design disturbance observer. There are many methods for disturbance observer design, such as output feedback method<sup>13</sup>, nonlinear disturbance observer<sup>14</sup> and feedback linearization disturbance observer design method<sup>8</sup>.

However, the trickiest problem of observer-based robot joint fault diagnosis method is that the gain of observer is very hard to design. At present, in the design of observer gain, the cost function should be determined first, and the observer gain is selected to minimize the cost function. The above design process extends the range for the value of the gain, which has a great impact on the performance of the observer. Stability of the observer is also an important consideration, and it is mostly guaranteed by Lyapunov function, which is very hard to find<sup>15</sup>.

The difficulty in gain design in observer-based robot fault diagnosis system promotes the research of robot fault diagnosis based on redundant sensors. Thanks to the development of sensors, sensors integrated with detection element, actuators and power supply have made a great breakthrough, such as magnetic detection chip, velocity measurement and gravity measuring elements<sup>16</sup>, and they have become very important feedback components in robot joint fault diagnosis system.

<sup>1</sup>Institute of School of Automation, University of Science and Technology Beijing, Beijing 100083, China. <sup>2</sup>Institute of School of Mechatronic Engineering and Automation, Foshan University, Foshan 528231, China. ✉email: panjinghuiwork@126.com

Three-dimensional gyroscope is used to measure the movement of the object. For a six degree of freedom robot joint system, low-cost MEMS magnetic, angular velocity and gravity sensors are used to estimate the joint angle of a rotating manipulator<sup>17</sup>. The sensor is installed at any position of the manipulator link in any attitude, and the attitude and position information of the joint system is obtained by above attitude measurement sensors.

However, the robot fault diagnosis system based on redundant sensors not only increases the structural complexity, but also increases the hardware cost of the system. In addition, additional sensors also increase the possibility of sensor failure in the system. With the development of artificial intelligence, the fault diagnosis method based on data-driven is developed. Robot system based on data-driven method only needs to be trained using fault data, which is easy to be collected when there is a fault on robot system, so there is no need to use redundant sensors.

Neural network (NN) method has the advantages of massively parallel, distributed processing, self-organization and self-learning capacity, so it is very popular on fault diagnosis<sup>18–20</sup>. The main stream of NN include basic perceptron, Feed-Forward NN (FFNN), long short-term memory (LSTM), CNN and RNN etc.<sup>21–23</sup>. Perceptron is the most original version of NN, which is just a simple neuron. As actual problems need more neurons to participate in the calculation, networks consist of many neurons are proposed, which are the NN in the ordinary sense. There are many variants of the NN. Zhang used multi-layer FFNN for mechanical fault diagnosis<sup>24</sup>. R. Sánchez researched Spur Gearbox with feed forward back propagation artificial NN to realize fault identification and classification<sup>25</sup>. Recently, Khoulood studied LSTM on reflective fiber fault detection, which reduced operation and maintenance expenses<sup>26</sup>. Albert also tried LSTM neural network for photovoltaic array fault diagnosis, and realized automatic feature extraction. Furthermore, to improve the feature extraction ability of neural network, convolution operation is adopted, where a convolution kernel is used to scan the full figure<sup>27</sup>. Chen researched CNN for bearing fault diagnosis<sup>28</sup>. Bo used adaptive label propagation technique combined with deep convolution variational autoencoder to achieve emerge fault diagnosis<sup>29</sup>. However, regular NN can't extract adequate information needed by fault diagnosis architecture due to the simple structure of this network. FFNN requires abundant data sets to cover all situations, or it may fail. The matrix calculation of CNN is complex, especially in multi-layer NN structure. Moreover, CNN faces a severe problem, and that is the gradient loss in parameter updating by back propagation.

To overcome the gradient loss in parameter updating process, the RNN is used on fault diagnosis of robot actuator. The output of last layer is feed through to the next one, thus, the gradient loss problem which exists in deep learning NN is solved. A multi-scale cluster-graph convolution network with multi-channel RNN is proposed in<sup>30</sup>. Motor fault diagnosis under nonstationary conditions is studied in<sup>31</sup>. Also, the bearing fault location using RNN is researched in<sup>32</sup>. These scientific research on robot fault diagnosis using RNN extremely enriched robot application filed. Chen detailed the application of RNN on multiple domains<sup>33</sup>, and the bearing mechanic fault diagnosis under unknown situations is also a hotspot.

However, the existing NN fault diagnosis mostly focuses on the mechanical fault of robot actuators, without considering the fault classification of sensors and actuators at the same time. Both of the sensors and actuators of robot can have constant deviation and constant gain faults, and the sensors are seriously disturbed by noise. In addition, the difference of the fault data between the sensors and the actuator is very minor, and the use of conventional NN methods leads to low identification accuracy and long training and diagnosis time.

As an extension of our previous research in<sup>34</sup>, In this paper, a data-driven deep residual neural network fault diagnosis method is proposed for robot joints system. It covers many kinds of faults, including sensor and actuator faults, furthermore, the faults for both sensor and actuator are grouped in detail. Firstly, a deep residual neural network is constructed using small convolution kernel for fault diagnosis. Then, gaussian white noise obtained by MATLAB is injected into the fault data set to verify the noise immunity for the proposed deep residual network. Finally, the feasibility and effectiveness of the proposed deep residual network fault diagnosis method for robot joint sensor and actuator fault diagnosis are proved by simulation. The main contributions of this paper are as follows.

- (1) The robot sensor and actuator fault feedback data are very similar, which could be seen from our previous research [This is the article title of our previous research and it should be placed at reference], so this could result in ambiguous data feature boundary and this cause low diagnosis accuracy. The DRNN is adopted and the diagnosis architecture is well designed to eliminant gradient loss.
- (2) This paper gives a data fusion method for robot sensor and actuator feedback signals. Where the Gaussian white noise is considered to simulate the real working condition. The robot fault diagnosis using fused data simplifies dimension of input.
- (3) Comprehensive and comparison experiments are conducted, where the current popular NN methods are adopted to show their effectiveness and to compare with DRNN. The results show that DRNN owns faster converging speed and higher accuracy.

This paper is organized as follows: The model of robot joint is established in part II, and the fault data set is obtained according to the fault model in the first part. Then basic principle of DRNN is introduced in part III. In part IV, simulation and analytic visualization of fault classification results are conducted. The author summarizes the research at the end of this paper.

## Model and fault dataset

**Mathematical model of robot joint.** The dynamic model of robot joint is established by Lagrange balance method from the point of view of energy<sup>35</sup>.

$\rho$	$f_a$	Fault type
1	Not zero	Constant deviation fault
$0 < \rho < 1$	Zero	Constant gain fault
0	Not zero	Actuator stuck
0	Zero	Actuator broken

**Table 1.** Actuator fault type table.

$$S_0: D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \tag{1}$$

where  $\tau$  is the vector for torque with dimension of  $n$ ,  $q, \dot{q}, \ddot{q} \in R^n$  are state variables of angular position, angular velocity and angular acceleration,  $D(q)$  and  $C(q, \dot{q})$  are square matrix with dimension of  $n$ , denote inertia matrix and Coriolis force matrix respectively,  $G(q) \in R^n$  is the gravity moment vector.

Let position and speed of joints be the state variables, and use variable substitution  $x = [x_1 \ x_2]^T = [q \ \dot{q}]^T$ , then (Eq. 1) could be rewritten as

$$S_1: \begin{cases} \dot{x} = Ax + Bf(q, \dot{q}, \tau) \\ y = Ex \end{cases}, \tag{2}$$

where  $x$  is the state variables for  $S_1$ ,  $y$  is the system output,  $f(q, \dot{q}, \tau)$  is the system coupling term, and the specific values of matrix  $A, B$  and  $E$  are as follows.

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$f(q, \dot{q}, \tau) = D^{-1}(q)[\tau - C(q, \dot{q})\dot{q} - G(q)].$$

**Fault model of robot joint actuator.** Robot joint actuator faults can be roughly divided into two categories according to the mechanism on the occurrence of joint faults, and they are multiplicative fault and additive fault.

For  $S_1$ , when the actuator has multiplicative failure, the system (Eq. 2) could be written as follows

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \rho\tau, \tag{3}$$

where  $\rho \in [0, 1][0, 1]$  is the effective factor of the actuator. A zero of  $\rho$  means the actuator is completely broken while one means it works very well.

In addition to the effective factor  $\rho$ , there may be additional torque offset fault, and it could be represented in the following equation.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + f_a, \tag{4}$$

where  $f_a$  is the actuator failure function, and its value is positively correlated with degree of damage.

Now both  $\rho$  and  $f_a$  would decide which kind of faults the actuator is currently suffering. The following four fault models can be established by further dividing the faults.

ErrA1: Constant deviation fault of actuator.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + f_a, (f_a \neq 0).$$

ErrA2: Constant gain fault of actuator.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \rho\tau, (\rho \in (0, 1)).$$

ErrA3: The actuator is completely stuck.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = f_a, (f_a \neq 0).$$

ErrA4: The actuator is completely broken.

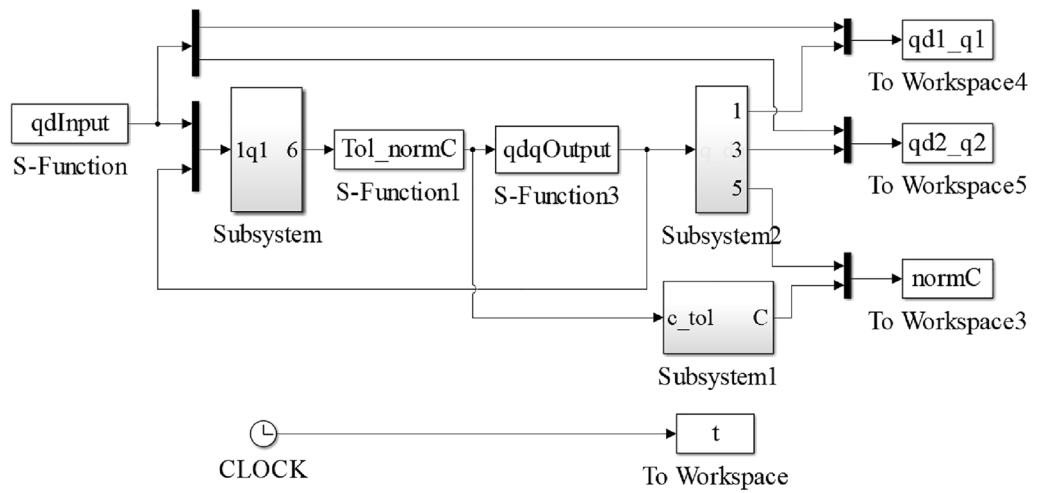
$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = 0.$$

Table 1 concludes above mentioned actuator fault types.

**Fault model of robot joint sensor.** Similar to robot joint actuator fault, sensor fault can be divided into four groups, and they are constant deviation fault, constant gain fault and stuck fault as well as sensor completely broken. Sensors are used to detect system state, so sensor failure directly affects the state feedback of the system, which means the output of the system is directly affected.

Here, four types of sensor faults are listed and corresponding mathematical formula are also deduced.

ErrS1: Constant deviation fault of sensor.



**Figure 1.** Model of robot joint control system.

$$\begin{cases} \dot{x} = Ax + Bf(q, \dot{q}, \tau) \\ y = Ex + f_b, (f_b \neq 0) \end{cases}$$

where  $f_b$  is the sensor failure function with the same dimension of  $x$ .  $f_b$  is positively correlated with degree of damage.

ErrS2: Constant gain fault of sensor.

$$\begin{cases} \dot{x} = Ax + Bf(q, \dot{q}, \tau) \\ y = \lambda Ex, (\lambda \in (0, 1)) \end{cases}$$

where  $\lambda$  is the effective factor of the sensor.

ErrS3: The sensor is at constant output point.

$$\begin{cases} \dot{x} = Ax + Bf(q, \dot{q}, \tau) \\ y = f_b, (f_b \neq 0) \end{cases}$$

When this happens, it means there is no reaction for the sensor when robot joint moves.

ErrS4: The sensor is at zero output point.

$$\begin{cases} \dot{x} = Ax + Bf(q, \dot{q}, \tau) \\ y = 0, (f_b \neq 0) \end{cases}$$

The effective factor of the sensor is zero, and it means the sensor does not work anymore. The actuator and sensor faults are combined together, and considering (Eq. 1) and (Eq. 2), the expression after integration is given in (Eq. 5).

$$S_1: \begin{cases} \dot{x} = Ax + BD^{-1}(q)[\rho\tau + f_a - C(q, \dot{q})\dot{q} - G(q)] \\ y = \lambda Ex + f_b \end{cases} \quad (5)$$

**Fault data acquisition.** The model of the robot joint control system is established by MATLAB/Simulink as shown in Fig. 1. The total simulation time is set to 40 s and the sampling frequency is 100 Hz.

From the joint sensor fault model and actuator fault model in (Eq. 5), they effect the system in a different way. However, through the model derivation and transformation, the sensor fault can be transformed into actuator fault through the first-order filter, which simplifies the analysis procedure. In order to reflect as much as information using collected fault data set, the sensor and actuator fault data are pre-processed according to the following formula.

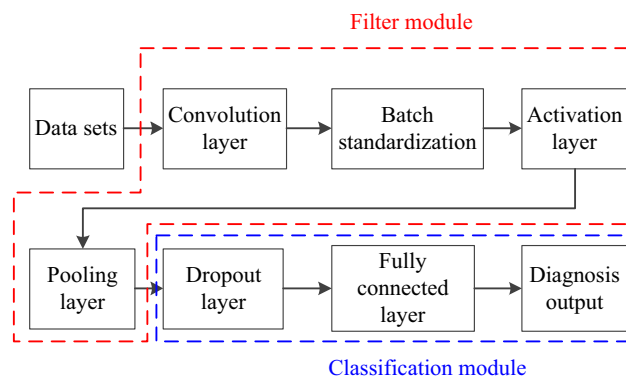
$$Fault = a\Delta Sensor + bActuator + Noise \quad (6)$$

where *Fault* denotes fault data set needed,  $\Delta Sensor$  represents the difference between sensor output and settings, *Actuator* is the output of actuator,  $a$  and  $b$  are sensor and actuator faults coefficient respectively. Item *Noise* represents the gaussian white noise, and it adds with sensor and actuator feedback signals, and three of them form the fault signals to verify the noise immunity ability of the RNN.

In our study, fault types for both sensor and actuator are considered, including constant gain fault, constant deviation fault, stuck fault, and inactive fault. Above eight kinds of faults are researched separately for sensor and

Fault types	Parameters
ErrA1-F1	$\rho = 1, f_a = 5$
ErrA2-F2	$\rho = 0.8, f_a = 0$
ErrA3-F3	$\rho = 0, f_a = 30$
ErrA4-F4	$\rho = 0, f_a = 0$
Actuator & sensor deviation fault-F5	$\begin{cases} \rho = 1, f_a = 5 \\ \lambda = 1, f_b = 0.1 \end{cases}$
ErrS1-F6	$\lambda = 1, f_b = 0.1$
ErrS2-F7	$\lambda = 0.8, f_b = 0$
ErrS3-F8	$\lambda = 0, f_b = 0.3$
ErrS4-F9	$\lambda = 0, f_b = 0$
Norm-F10	$\begin{cases} \rho = 1, f_a = 0 \\ \lambda = 1, f_b = 0 \end{cases}$

**Table 2.** Actuator fault type table.



**Figure 2.** Diagram for CNN fault diagnosis system.

actuator. furthermore, constant deviation fault occurred simultaneous on sensor and actuator is also considered. Table 2 lists all ten kinds of system types including normal condition.

### Fault diagnosis based on resnet

**Architecture of RESNET.** RESNET is a kind of deep CNN, and it provides forward information through residual feedforward. Not only the depth of the network is expanded, but also the gradient disappearance phenomenon in the process of parameter updating is avoided by RESNET, which improves the strong adaptive capacity of the network.

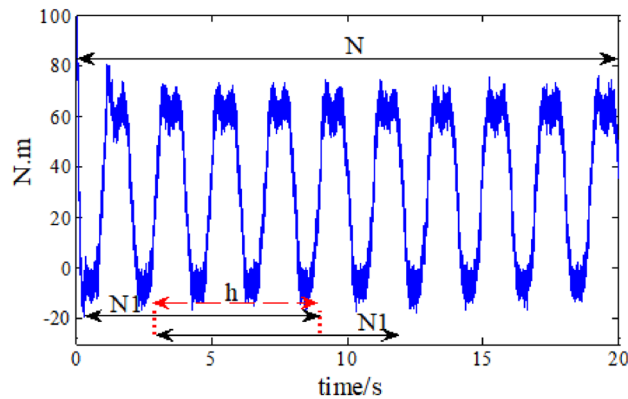
Generally, CNN consists of five parts: convolutional layer, batch standardization, active layer, pooling layer, and dropout layer. The whole CNN architecture is constructed by fully connected layer (FCL), as shown in Fig. 2.

Furthermore, CNN fault diagnosis architecture could be divided into two parts, and they are filtering and classification module. The filtering module can extract features from the input data set, and the classification module can process and classify the features extracted from the filtering module. Both the two modules use parameter sharing mechanism and sparse connection to reduce the amount of model training procedure and to improve the training efficiency of the network<sup>36</sup>.

**Enhancement of dataset.** It usually needs large training sample when using RESNET for fault diagnosis. Traditional data set enhancement methods mainly include: flip transformation, scaling, rotation transformation, mirror transformation, and translation transformation<sup>37</sup>. However, these methods are not suitable for one-dimensional sample data. In order to expand the acquired robot joint fault data, a sliding sampling data set enhancement method is proposed, as indicated in Fig. 3.

The system data over a period is obtained, and data sampling points of  $N_1$  is used each time the network needs to be trained. Assuming a total number of  $m * N_1$  data is obtained, then the network could be trained by  $m$  times according to above method. In order to expand the coefficient of utilization for data, the start point of the second data set is  $h$  backwards then the first one, and the rest is roughly the same.

When the sliding step size is small, that means  $h$  is quite large, we can get more data samples, which can well meet the needs of data sets for deep convolution neural network in training. Comprehensive consideration, the sliding step selected here is 29.



**Figure 3.** Schematic diagram of data set enhancement method.

**RESNET fault diagnosis model based on data driven.** RESNET is a kind of deep convolutional neural network. In order to reduce the use of the model parameters and the calculation of the network, this paper uses several small convolution kernels and stacks them all, thus the convolution kernel size is increased, which enhances the ability of feature extraction for the network. The whole structure using RESNET in fault diagnosis from data input to fault diagnosis output is shown in Fig. 4.

Conv3\*1,4,6 in Fig. 4 represents a convolution kernels dimension, and its depth is 4, sliding step is 6. BN, ReLU, GAP and Dense are batch standardization, active layer, pooling layer and fully connected layer respectively in Fig. 2. The structure of the whole network is regularized to prevent over fitting, when input and output dimensions of each convolution module are inconsistent, the residual jump is carried out through virtual channel, which is indicated by dotted line in Fig. 4.

### Simulation verification and analysis

The simulation platform used is PyCharm, combining with Python3.6 interpreter. RESNET architecture is based on Keras with Tensorflow as its background. Tensorflow is an end-to-end open-source machine learning platform, and it has a comprehensive and flexible ecosystem, which contains a variety of tools, libraries, and community resources. Keras is an open-source artificial neural network library written by python. It can be used as a high-level application interface of tensorflow to design, debug, evaluate, apply, and visualize the deep learning model<sup>38</sup>.

**Fault dataset.** A variety of fault types listed in Table 1 are studied. The fault data sets are obtained for each kind of sensor and actuator faults, and we get 4000 data for every kind of fault type, around 216 data points are contained in the aforementioned 4000 data points. The sliding sampling enhancement method is used, as shown in Fig. 3, and 1000 samples are taken for each kind of fault, so a total of 10,000 samples are studied, including one data set under normal operation environment. The fault diagnosis results need to be evaluated, so 1000 samples are divided a by a ratio of 7:2:1, which means 70% of the data is used to train the neural network, 20% is used for validation and the rest for testing.

**Hyper-parameters of RESNET.** According to previous research, hyper parameters in deep learning algorithm not only affect the performance of the algorithm itself, but also affect the expression ability of the trained model. However, about how to set hyper parameters, there is no mature theory. At present, most of the hyper parameter settings are based on the trial and error. This paper takes the advice from Y.bengio<sup>39</sup>. The corresponding hyper parameters are set according to whether these hyper parameters can increase or decrease the capacity of the model.

In the process of parameter updating, the exponential decay learning rate is used. At first, a large learning rate is used to get the optimal solution quickly, and then the learning rate is gradually reduced to keep the model stable in the later stage of training. The initial learning rate  $\eta_0$  is set to 0.2, and decay learning rate  $\xi$  is set to 0.99, so the decay rate is updated per round. The expression for decay rate is as follows.

$$\begin{cases} \eta = \eta_0 \times \xi^{(H/L)} \\ H = Epoch/Batch\_k \end{cases}$$

where  $\eta$  denotes exponential decay learning rate,  $H$  stands for the current round, and  $L$  is the period,  $Batch\_k$  is the number of iterations. When a complete data set passes through the neural network once and then returns, this process is called *Epoch* and here we set *Epoch* to 40.

In order to alleviate the over fitting of neural network,  $l_2$  regularization method is used in this paper. Regularization is to introduce the model complexity index into the loss function and suppress the noise in the training data set by weighting the parameters  $w$  in the neural network. The loss function is as follows.

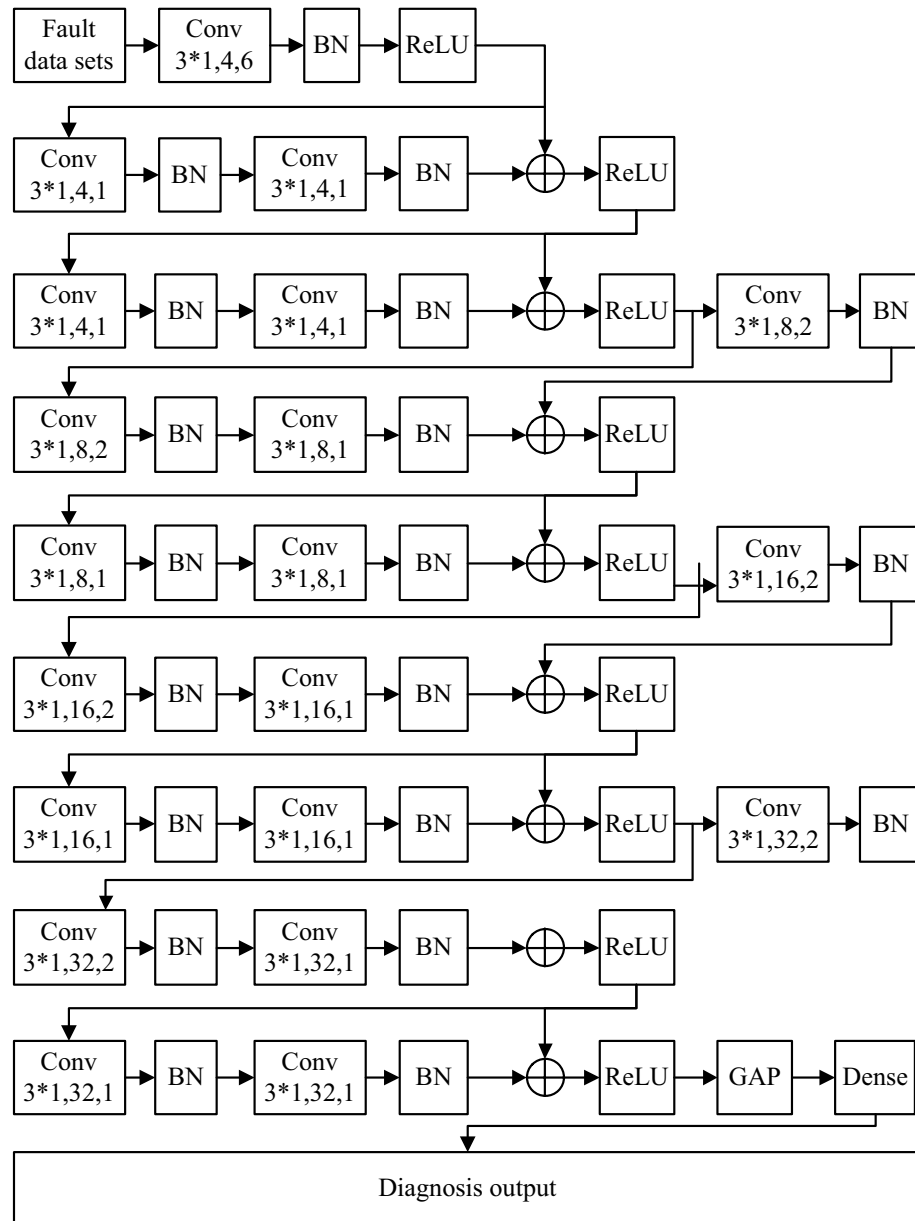


Figure 4. RESNET model diagram.

$$\begin{cases} Loss = Loss_{all} + REGULARIZER \times Loss(w) \\ Loss(w) = \sum_i |w_i| \end{cases} \quad (7)$$

where  $Loss_{all}$  represents loss function of all parameters in the model,  $REGULARIZER$  is the regularization weight,  $w$  generally refers to the weight in the forward propagation of neural network, and  $Loss(w)$  is the result of  $l_2$  regularization of parameter  $w$ .

The Adma optimal algorithm is used and the procedure of weight updating is as follows.

Step 1: Give the iteration step  $\varepsilon = 0.001$ .

Step 2: Set the decay rate for matrix calculation,  $\rho_1 = 0.99$ ,  $\rho_2 = 0.999$ .

Step 3: Determine the convergence threshold  $\delta = 10^{-8}$ .

Step 4: Initialize network weight  $\theta$ , and initializes first and second moment variables  $s, r$ , and set  $s = 0, r = 0$ .

Step 5: Set the simulation time step to 0.0001.

Step 6: Small data set with  $m$  samples are collected from the training set, use  $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$  denotes it and set corresponding goals  $y^{(i)}$ .



Network type	Average accuracy on training set	Average accuracy on testing set
ANN	68.32 ± 0.63	68.16 ± 1.24
SVM	63.16 ± 0.23	63.02 ± 1.36
CNN	97.68 ± 1.65	95.73 ± 1.82
LTMN	99.87 ± 0.12	98.67 ± 0.24
RESNET	99.91 ± 0.09	99.87 ± 0.13

**Table 3.** Actuator fault type table.

*Step 7:* Calculate gradient variable  $g \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(x^{(i)}; \theta), y^{(i)})$ , and update biased first moment estimation  $s \leftarrow \rho_1 s + (1 - \rho_1)g$  as well as biased second moment estimation  $r \leftarrow \rho_2 r + (1 - \rho_2)g$ .

*Step 8:* Correct the deviation of the first moment  $\hat{s} \leftarrow \frac{s}{1 - \rho_1}$  and deviation of the second moment  $\hat{r} \leftarrow \frac{r}{1 - \rho_2}$ .

*Step 9:* Calculate incremental weight error  $\Delta\theta = -\varepsilon \frac{\hat{s}}{\sqrt{\hat{r} + \delta}}$ , and update it to the network weight  $\theta \leftarrow \theta + \Delta\theta$ .

*Step 10:* If the convergence threshold in Step 3 is not met, then, back to Step 6, else end the iterative process.

Machine learning is a highly empirical process, accompanied by a large number of iterations. In order to find the most suitable model, a lot of model training is needed. The optimizer is a tool to guide the neural network to update parameters. Using the optimization algorithm can help to quickly train the model and find the optimal solution. In this paper, we choose the Adma optimization algorithm, which does not occupy memory, and it is an efficient random optimization algorithm.

**Fault diagnosis results.** In order to verify the feasibility and effectiveness of the RESNET fault diagnosis method used in this paper for robot joint sensor and actuator fault diagnosis, the artificial neural network, support vector machine, convolution neural network, long-term memory network and deep residual network methods are used for comparative analysis and verification. The average accuracy and standard deviation of different network training results are compared in Table 3.

It can be seen from Table 3 that compared with the traditional data-driven fault diagnosis method (ANN and SVM), the fault recognition accuracy of the deep learning fault diagnosis method is significantly improved. The average accuracy of network training results is less than 70% using traditional fault diagnosis method, but the average accuracy of CNN, LTMN and RESNET is more than 95%. Meanwhile, the average accuracy of RESNET in both training and test set is more than 99%, which means RESNET is suitable and effective for fault diagnosis of robot joint sensors and actuators faults.

For the purpose of further analyzing the diagnosis effect of three kinds of deep neural networks (CNN, LTMN and RESNET) on robot joint sensor and actuator faults, the accuracy and loss function of each kind of network in the training set and testing set are drawn with the help of tensorflow, and the results are shown in Fig. 5 and Table 4.

Figure 5 shows the fault diagnosis accuracy and loss on both training and validation datasets. The reasons that validation loss is better than training loss may be deduced as follows. The datasets under the same working condition are grouped proportionally as 7:2:1, so the validation datasets only occupy 20% of all. This may cause the accuracy difference.

The dropout layer in training process. In order to accelerate the model training time, the dropout layer is introduced. While the dropout doesn't operate at validation stage. Combining Fig. 5, Table 4, it can be seen that the highest accuracy of CNN in training data set is 99.2%, and that number is 97.3% in test data set. The highest accuracy of LTMN and RESNET in training data set and test data set is 100%, and their accuracy curve is relatively smoother. Obviously, both LTMN and RESNET can better reflect the fault information from the original data set and can make more accurate judgment of sensor and actuator faults in robot joints. But from the point of view of the training time, RESNET needs 2.5 T, while LTMN needs 12 T, which is more than five times of RESNET.

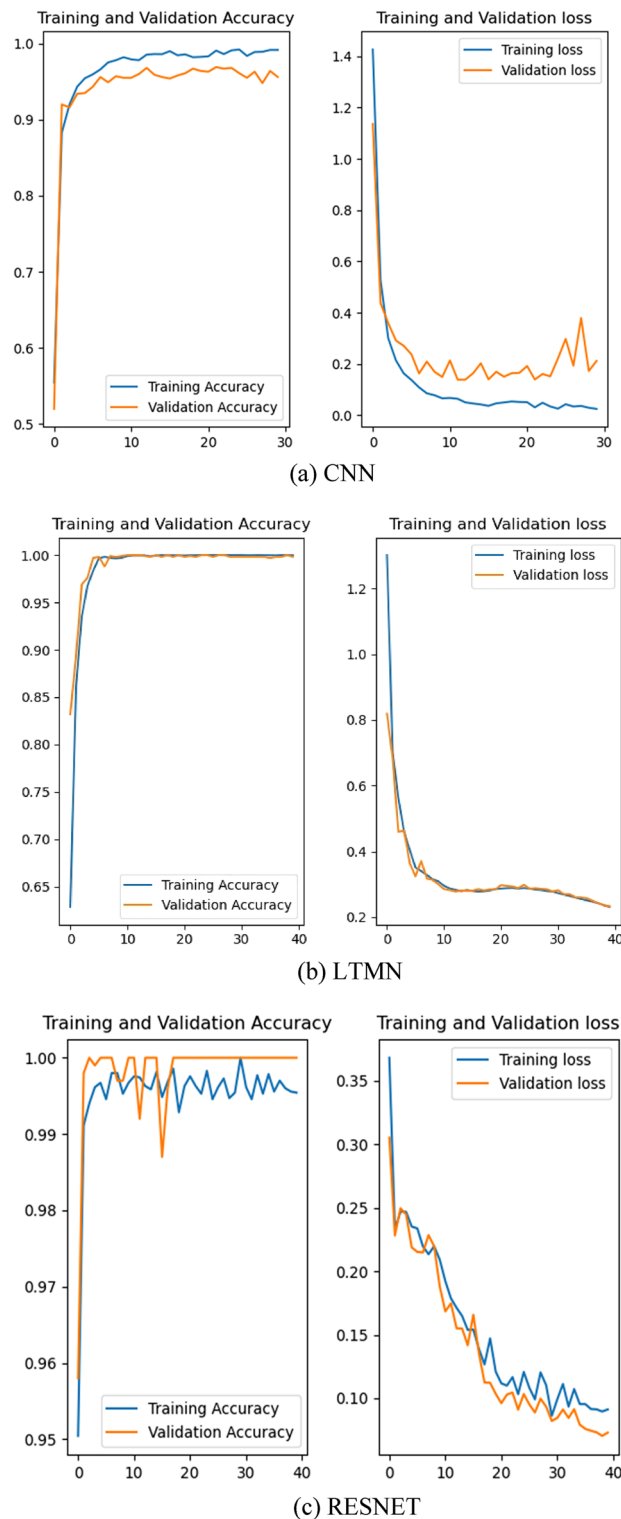
Thus, compared with other data-driven fault diagnosis methods, the deep RESNET network used in this paper has higher accuracy in the fault diagnosis of robot joint sensors and actuators, and the training time cost is relatively less, so it is more practical.

Considering that the initial value of the neural network is random, in order to avoid the inaccuracy of fault diagnosis results and verify the reliability of each training result, the cross method is used to train the fault data set for ten times. The results are shown in Fig. 6.

As can be seen from Fig. 6, the training accuracy of RESNET in each experiment is more than 99%, and the lowest accuracy is 99.6%. Therefore, it can be concluded that the best fault diagnosis method of robot joint sensor and actuator is RESNET-based neural network in our study.

**Visualization analysis of RESNET classification results.** Visualization analysis of RESNET is carried out to study the intermediate process of neural network. t-distributed stochastic neighbour embedding (t-SNE) is used to reduce the data dimension of the output of each residual block in RESNET architecture, and the results are shown in Fig. 7.

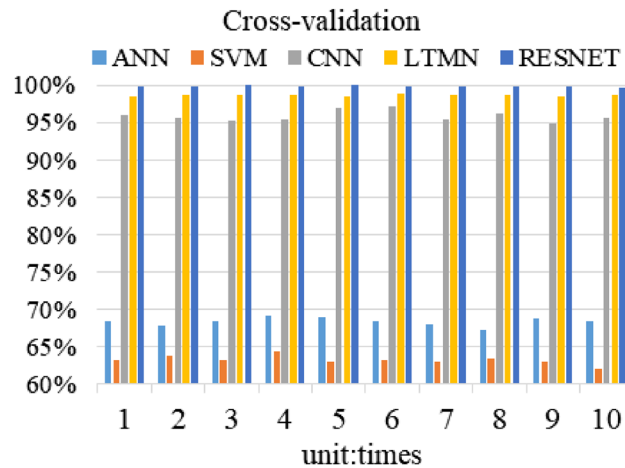




**Figure 5.** Fault diagnosis results of three kind of neural networks.

Network type	Max accuracy for training (%)	Max accuracy for testing (%)	Stable time	Training time
CNN	99.2	97.3	4 Epoch	1 T
LTMN	100	100	7 Epoch	12 T
RESNET	100	100	5 Epoch	2.5 T

**Table 4.** Actuator fault type table.



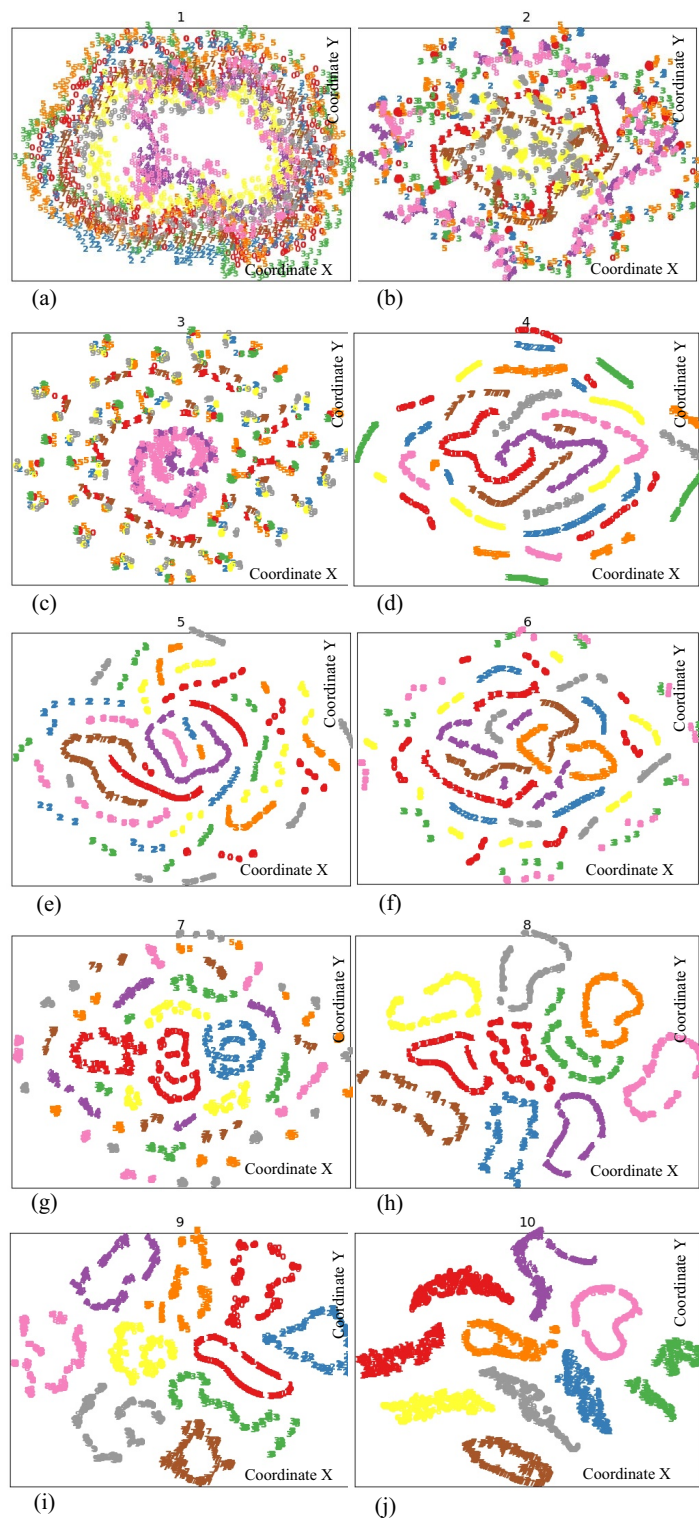
**Figure 6.** Cross validation result.

Figure 7a is the result of t-SNE visualization of fault data set with signal to noise ratio setting to 10. Figure 7b–i correspond to the t-SNE visualization results of the outputs of the eight residual blocks in the RESNET architecture in Fig. 4. Figure 7j is the final output of RESNET neural network. Different curves with different colours in Fig. 7 represent different fault types listed in Table 1. A total of 10 colours correspond to 9 faults and 1 health state for robot joint system.

It can be seen from the Fig. 7 that with the increase of residual blocks, the data expression ability of RESNET network is gradually enhanced, and finally the accurate classification of nine kinds of fault types is realized. From the visualization analysis results of data in two-dimensional space, the feasibility and effectiveness of RESNET fault diagnosis algorithm proposed in this paper are further verified.

## Conclusion

A novel RESNET based neural network fault diagnosis method for robot joint system is proposed in this paper. Aiming at the problem that the traditional robot joint fault diagnosis algorithm cannot accurately locate the fault, a data-driven RESNET fault diagnosis algorithm is proposed. The fault models of robot joint sensor and actuator are built to simulate various fault states in order for fault data acquisition. After getting the fault data set needed, the RESNET architecture constructed by small convolution kernel is studied, and by increasing the residual blocks, the convolution kernel is gradually increased to improve the fault extraction ability of the model. Compared with other data-driven based fault diagnosis algorithms, simulation results show that the accuracy of fault diagnosis based on RESNET is more than 99%, which is the highest among all studied methods and meanwhile the model training time is less.



**Figure 7.** Output of residual block.

#### Data availability

The data presented in this study are available on request from the corresponding author.

#### Code availability

The data presented in this study are available on request from the corresponding author.

## References

- Burgner-Kahrs, J., Rucker, D. C. & Choset, H. Continuum robots for medical applications: A survey. *IEEE Trans. Robot.* **31**(6), 1261–1280 (2015).
- Gruver, W. A. Intelligent robotics in manufacturing, service and rehabilitation: An overview. *IEEE Trans. Ind. Electron.* **41**(1), 4–11 (1994).
- Nanjangud, A. *et al.* Robotics and AI-enabled on-orbit operations with future generation of small satellites. *Proc. IEEE* **106**(3), 429–439 (2018).
- Kim, J. H. Multi-axis force-torque sensors for measuring zero-moment point in humanoid robots: A review. *IEEE Sens. J.* **20**(3), 1126–1141 (2019).
- Zhao, B. *et al.* Particle filter for fault diagnosis and robust navigation of underwater robot. *IEEE Trans. Control Syst. Technol.* **22**(6), 2399–2407 (2014).
- KIM, M. J. *et al.* Model-free friction observers for flexible joint robots with torque measurements. *IEEE Trans. Robot.* **35**(6), 1508–1515 (2019).
- Xie, X., Yue, D. & Park, J. H. Observer-based state estimation of discrete-time fuzzy systems based on a joint switching mechanism for adjacent instants. *IEEE Trans. Cybern.* **50**(8), 3545–3555 (2019).
- Guo, J. Robust tracking control of variable stiffness joint based on feedback linearization and disturbance observer with estimation error compensation. *IEEE Access* **8**, 173732–173754 (2020).
- Capisani, L. M. *et al.* Manipulator fault diagnosis via higher order sliding-mode observers[J]. *IEEE Trans. Ind. Electron.* **59**(10), 3979–3986 (2012).
- Yang, Yueneng & Yan, Ye. Backstepping sliding mode control for uncertain strict-feedback nonlinear systems using neural-network-based adaptive gain scheduling. *J. Syst. Eng. Electron.* **29**(3), 580–586 (2018).
- Shaker, M. S. Hybrid approach to design Takagi-Sugeno observer-based FTC for non-linear systems affected by simultaneous time varying actuator and sensor faults. *IET Control Theory Appl.* **13**(5), 632–641 (2019).
- Bernard, P. & Andrieu, V. Luenberger observers for nonautonomous nonlinear systems. *IEEE Trans. Autom. Control* **64**(1), 270–281 (2018).
- Ding, S. *et al.* Disturbance observer design for nonlinear systems represented by input–output models. *IEEE Trans. Ind. Electron.* **67**(2), 1222–1232 (2019).
- Yang, X. *et al.* Disturbance observer based on biologically inspired integral sliding mode control for trajectory tracking of mobile robots. *IEEE Access* **7**, 48382–48391 (2019).
- Chen, W. H. *et al.* A nonlinear disturbance observer for robotic manipulators. *IEEE Trans. Ind. Electron.* **47**(4), 932–938 (2000).
- Ashokaraj, I. A. R. *et al.* Robust sensor-based navigation for mobile robots. *IEEE Trans. Instrum. Meas.* **58**(3), 551–556 (2008).
- Xu, X. *et al.* A novel joint angle estimation method for serial manipulator using mems sensors. *IEEE Trans. Ind. Electron.* **67**(12), 10610–10620 (2020).
- Xia, L. *et al.* Data association-based fault diagnosis of IMUs: Optimized DBN design and wheeled robot evaluation. *IEEE Access* **8**, 59618–59636 (2020).
- Talebi, H. A. & Khorasani, K. A neural network-based multiplicative actuator fault detection and isolation of nonlinear systems. *IEEE Trans. Control Syst. Technol.* **21**(3), 842–851 (2012).
- Vemuri, A. T. & Polycarpou, M. M. Neural-network-based robust fault diagnosis in robotic systems. *IEEE Trans. Neural Netw.* **8**(6), 1410–1420 (1997).
- Li, H. & Zhang, L. A bilevel learning model and algorithm for self-organizing feed-forward neural networks for pattern classification. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(11), 4901–4915 (2020).
- Paiva, E., Paim, A. & Ebecken, N. Convolutional neural networks and long short-term memory networks for textual classification of information access requests. *IEEE Lat. Am. Trans.* **19**(5), 826–833 (2021).
- Gao, Z., Lu, Z., Wang, J., Ying, S. & Shi, J. A convolutional neural network and graph convolutional network based framework for classification of breast histopathological images. *IEEE J. Biomed. Health Inform.* **26**(7), 3163–3173 (2022).
- Bin, Z., Jun, Y. & Wei, Z. Application of multi-layer feed-forward neural network in fault diagnosis based on fbp algorithm. *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* 117–120 (2008).
- Sánchez, R., Arpi A. & Minchala, L. Fault identification and classification of spur gearbox with feed forward back propagation artificial neural network. *2012 VI Andean Region International Conference* 215–215 (2012).
- Abdelli, K., Griesser, H., Ehrle, P., Trotschug, C. & Pachnicke, S. Reflective fiber fault detection and characterization using long short-term memory. *J. Opt. Commun. Netw.* **13**(10), E32–E41 (2021).
- Appiah, A. Y., Zhang, X., Ayawli, B. B. K. & Kyeremeh, F. Long short-term memory networks based automatic feature extraction for photovoltaic array fault diagnosis. *IEEE Access* **7**, 30089–30101 (2019).
- Chen, J. *et al.* Multiscale convolutional neural network with feature alignment for bearing fault diagnosis. *IEEE Trans. Instrum. Meas.* **70**, 1–10 (2021).
- She, B., Tian, F. & Liang, W. Fault diagnosis based on a deep convolution variational autoencoder network. *Yi Qi Yi Biao Xue Bao/Chin. J. Sci. Instrum.* **39**(10), 27–35 (2018).
- Dang, L., Nie, Y., Long, C., Zhang, Q. & Li, G. MSR-GCN: Multi-scale residual graph convolution networks for human motion prediction. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 11447–11456 (2021).
- Liu, R., Wang, F., Yang, B. & Qin, S. J. Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions. *IEEE Trans. Ind. Inform.* **16**(6), 3797–3806 (2019).
- Chen, L. *et al.* Deep residual network for identifying bearing fault location and fault severity concurrently. *IEEE Access* **8**, 168026–168035 (2020).
- Chen, Z. *et al.* Residual deep subdomain adaptation network: A new method for intelligent fault diagnosis of bearings across multiple domains. *Mech. Mach. Theory* **169**, 104635 (2022).
- Peng, K. Sensor and actuator fault diagnosis for robot joint based on deep CNN. *Entropy* **23**, 751 (2021).
- Vázquez, L. A., Jurado, F. & Castañeda, C. E. Real-time decentralized neural control via backstepping for a robotic arm powered by industrial servomotors. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(2), 419–426 (2018).
- LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**(7553), 436–444 (2015).
- Wang, J., Mall, S. & Perez, L. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning* (MIT Press, 2017).
- Si, M., Tarnoczi, T. J., Wiens, B. M. & Du, K. Development of predictive emissions monitoring system using open source machine learning library—Keras: A case study on a cogeneration unit. *IEEE Access* **7**(10), 113463–113475 (2019).
- Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the Trade* (eds Montavon, G. *et al.*) 437–478 (Springer, 2012).

## Acknowledgements

This work was supported by the National Key R&D Program of China under Grant 2019YFB1309900.

## Author contributions

Conceptualization, J.P.; methodology, J.P., L.Q. and K.P.; software, L.Q.; validation, J.P.; formal analysis, K.P.; investigation, J.P. and K.P.; resources, L.Q. and K.P.; data curation, J.P. and L.Q.; writing—original draft preparation, J.P.; writing—review and editing, J.P. and K.P.; visualization, K.P. and L.Q.; supervision, L.Q.; project administration, L.Q.; funding acquisition, L.Q. and K.P. All authors have read and agreed to the published version of the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to J.P.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022