Multiscale Kernel Based Residual Convolutional Neural Network for Motor Fault Diagnosis Under Nonstationary Conditions

Ruonan Liu, *Member, IEEE*, Fei Wang, *Student Member, IEEE*, Boyuan Yang, and S. Joe Qin, *Fellow, IEEE*

***Abstract*—Motor fault diagnosis is imperative to enhance the reliability and security of industrial systems. However, since motors are often operated under nonstationary conditions, the high complexity of vibration signals raises notable difficulties for fault diagnosis. Therefore, considering the special physical characteristics of motor signals under nonstationary conditions, in this article, we propose a multiscale kernel based residual convolutional neural network (CNN) for motor fault diagnosis. Our contributions mainly fall into two aspects. First, we notice that each motor fault category has various patterns in vibration signals due to the changing operational conditions of the motor. To capture these patterns, a multiscale kernel algorithm is applied in the CNN architecture. Second, since the motor vibration signals are made up of many different components from different transfer paths, they are very complex and variable. To enable the architecture to extract fault features from deep and hierarchical representation spaces, sufficient depth of the network is needed, which will lead to the degradation problem. In the proposed method, residual learning is embedded into the multiscale kernel CNN to avoid performance degradation and build a deeper network. To validate the effectiveness of the proposed networks, a normal motor and five motors with different failures are tested. The results and comparisons with state-of-the-art methods highlight the superiority of the proposed method.**

***Index Terms*—Deep learning, motor fault diagnosis, multiscale kernel convolutional neural network (MK-CNN), residual learning.**

Manuscript received May 25, 2019; revised July 28, 2019; accepted August 18, 2019. Date of publication September 17, 2019; date of current version February 28, 2020. This work was supported by the National Science and Technology Major Project (2017-I-0001-0001). Paper no. TII-19-2029. *(Corresponding author: Boyuan Yang.)*

R. Liu is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and also with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: liuruonan04@163.com).

F. Wang is with the Institute of Cyberspace Research, Zhejiang University, Hangzhou 310007, China, and also with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: wfei.cs@gmail.com).

B. Yang is with the School of Electrical and Electronic Engineering, University of Manchester, M13 9PL Manchester, U.K. (e-mail: yangboyuanxjtu@163.com).

S. J. Qin is with the Department of Chemical Engineering and Materials Science and the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: sqin@usc.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2941868

I. Introduction

MOTORS have been widely used in modern industrial systems, such as wind turbines and vehicles. However, as motors have become more and more complex and expensive, the tolerance for performance degradation, productivity decrease, and safety hazards are also becoming less and less [1]. On the other hand, no matter how good quality the products are, they will deteriorate over time [2]. Therefore, as an effective means to estimate the reliability of motors and reduce the risk of unplanned shutdowns, fault diagnosis is of vital importance in modern industrial systems [3].

Because shallow learning models are unable to extract complex features, traditional fault diagnosis methods usually combine feature extractors with shallow learning models, such as artificial neural networks or support vector machines (SVMs) [4]. Feature extractors transform the raw signals into low-dimensional vectors so that they can be easily matched, and are relatively invariant with respect to transformations and distortions [5]. Most commonly used feature extractors, Fourier transform, wavelet transform (WT) [6], empirical mode decomposition [7], spectral kurtosis [8], and sparse representations [9], [10] are all widely used in industrial practice. The performance of the shallow learning models depends heavily on the quality of the extracted features from the collected signals [11].

The construction of feature extractor needs relevant prior knowledge and is rather specific to the task. However, motors have become increasingly complicated and diversified, which make it time consuming to construct a feature extractor for each type of motor. On the other hand, with the advancement of sensor techniques, the collection of industrial data becomes more convenient. Therefore, traditional fault diagnosis methods have been re-examined from the point of big data [12], [13]. Recently, deep learning technologies have led to a series of breakthroughs in the field due to its attractive characteristic that directly learns the high-level and hierarchical representations from massive raw data [14], [15]. Convolutional neural networks (CNNs) [16], deep belief networks (DBNs) [17], [18], residual CNN (ResCNN) [19], and autoencoders (SAEs) [20] are popular deep

learning methods used for various fault diagnosis applications. In [21], a deep autoencoder is used for fault feature mining and intelligent diagnosis of rotating machinery with massive data. Gan *et al.* proposed a hierarchical diagnosis network by collecting DBNs for the hierarchical identification of mechanical fault pattern recognition of rolling element bearings [17]. An enhanced deep feature fusion method for rotating machinery fault diagnosis is proposed in [11]. Hu *et al.* developed an intelligent fault diagnosis method for high-speed train based on deep neural networks [22]. Oh *et al.* proposed a scalable and unsupervised feature engineering method using vibration imaging and deep learning for rotor system diagnosis [15]. Shao *et al.* designed a convolutional DBN model with Gaussian visible units for bearing fault diagnosis [23]. Inspired by CNNs and WT, Pan *et al.* proposed a deep neural network, called LiftingNet, to learn features adapted from raw mechanical data without prior knowledge [24]. Many other fault diagnosis tasks have also been greatly benefited from deep models [25].

However, mechanical vibration signals are usually long one-dimensional (1-D) complex signals. Due to the varying operational conditions and noisy background, deeper and more complicated features should be extracted for mechanical fault diagnosis. In addition, the mechanical signals under nonstationary conditions can be much more complex. Thus, if deep networks are used to diagnose mechanical malfunctions, deep 1-D architectures are needed to extract features from such complicated signals. However, experiments find that when deep networks start converging, a problem has been exposed: with the network depth increasing, accuracy tends to saturate and then degrade, which is not caused by overfitting [26], [27]. Therefore, the increase in deep network layers may lead to even more serious degradation problems. On the other hand, due to the fixed single-scale convolutional kernel and pooling size, the input signals are analyzed with a fixed scale in traditional deep networks, whereas industrial systems are always working under variable conditions, which lead to time-varying signals. Therefore, although deep learning is a powerful tool for data analysis, it is less effective to extract features from time-varying signals.

To address the abovementioned problems, a multiscale kernel-based ResCNN (MK-ResCNN) architecture is proposed in this article for motor fault diagnosis. The main contributions of this article are summarized as follows.

1) We propose a well-designed deep network, termed MK-ResCNN, for motor fault diagnosis. In MK-ResCNN, multiscaled convolutional kernels are used to capture the characteristics of raw fault signals from multiple scales, which promote the robustness and represent ability of the captured characteristics even under nonstationary conditions. In addition, we apply the identity mapping and residual mapping to make very deep networks applicable for learning efficient fault characteristics, meanwhile to overcome the performance degradation problem that appears in traditional deep networks.
2) Since the proposed method is based on a CNN, it inherits the advantages of the CNN that can extract features and recognize faults from raw time-series signals without the help of signal processing techniques.

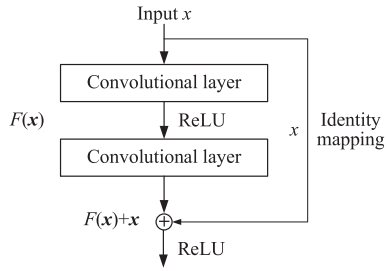

Fig. 1. Block of residual learning.

3) The proposed approach is evaluated through a motor fault simulation experiment with a comprehensive performance evaluation. The results are compared with state-of-the-art results in the field of fault diagnosis under nonstationary conditions to demonstrate the superiority of our method.

The rest of this article is organized as follows. Section II describes the details of the proposed approach. Experimental setup and data description are illustrated in Section III. In Section IV, the proposed method and five classical and state-of-the-art methods are applied to analyze the same experimental signals to show the effectiveness of the proposed method. Finally, Section V concludes this article.

II. Proposed Approach

A. Residual Learning

The analysis object for industrial system fault diagnosis is usually a long 1-D complex vibration signal. In addition, the changeable operational conditions of motors can even increase the difficulty. If we want to use deep networks to diagnose mechanical malfunctions, deeper 1-D architectures are needed, because generally the deeper a network is, the more complex features it can extract, and the better the performance is. However, previous experiments show that there exists a degradation problem in deep networks: when the network depth increases, accuracy saturates and then degrades, and the addition of more layers can lead to an even higher training error, which is not caused by overfitting [26], [27].

The degradation problem illustrates that a deep network is not easy to train. Theoretically, if the additional layer does not learn anything, but just copies the features of last layer (which is called identity mapping), the training error should not increase. Inspired by this intuition, residual learning [27] is embedded in the proposed framework. For a deep network architecture, and the input x, the learned feature is denoted as $H(x)$. Now, we expect the network to learn the residual $F(x) = H(x) - x$ because residual learning is easier than the traditional feature learning. The residual learning adopts every few stacked layers, as shown in Fig. 1. The output $\boldsymbol{y}$ is obtained by a shortcut connection operation is given as follows:

$$\boldsymbol{y} = F(\boldsymbol{x}, W_i) + \boldsymbol{x} \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the input and output vectors of the layers considered. Every building block has a multilayer architecture.

$F(\boldsymbol{x}, W_i)$ is the residual function, which represents the residual mapping to be learned. Take the building block in Fig. 1 as an example: there are two layers, and $F(\boldsymbol{x}, W_i)$ can be represented as

$$F = W_2\sigma(W_1\boldsymbol{x}) \tag{2}$$

where σ represents the activation function. In this article, σ is set to be the rectified linear unit (ReLU) function. The biases are omitted here to simplify the expression.

The dimensions of $\boldsymbol{x}$ and y must be equal. If the input and output dimension are unequal, a linear projection W_s can be performed by the shortcut connections to match the dimensions

$$\boldsymbol{y} = F(\boldsymbol{x}, \{W_i\}) + W_s\boldsymbol{x}. \tag{3}$$

If the residual value is not equal to 0, the network performance can still improve by adding the number of layers in the network. On the other hand if the residual value is 0, then the current layer is just an identity mapping, which will neither improve nor degrade. In this way, the degradation problem can be avoided, and therefore a deeper network can be built.

B. Convolutional Neural Networks

A CNN is a variant of neural networks, which consists of convolutional layers, an activation function layer, and pooling layers.

Each convolutional layer consists of several convolutional units. The loss function is optimized by a backpropagation algorithm, such as the gradient descent algorithm [28], conjugate gradient method [29], and AdaBoost algorithm [30]. The aim of convolution operation is to extract different levels of hierarchical features from raw data. The first convolutional layer may only can extract some low-level features. The more convolutional layers are, the more complex features can be extracted. Compared with other networks, the CNN exploits sparse connectivity by making the kernel smaller than the input and enforcing a local connectivity pattern among neurons of adjacent layers. Thus, the complicated interactions between units can be described more efficiently, and the overfitting risk can also be reduced. Each kernel (or weight matrix) in the CNN is used across the entire visual field, but learnt only once instead of learning a separate set of weights for every location. Therefore, a CNN is an extremely efficient way that applies the same linear transformation of a local region across the entire input to describe transformations.

In the activation function layer, ReLU is widely used because it can improve the training speed significantly [31]. It is defined as follows:

$$f(x) = \max(0, x). \tag{4}$$

The pooling layer is another important part in a CNN. It is subsampling in essence. Intuitively, this mechanism can be effective because the precise location of a feature is far less important than its relative position. Pooling will continuously reduce the size of the data space, so the number of parameters and calculation cost will also decrease, which also controls overfitting [32]. The pooling operation also makes the feature maps extracted by a CNN invariant to small translations of the input. In general, the pooling layer is periodically inserted between the convolutional layers of the CNN. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer, which is the most common pooling technique.

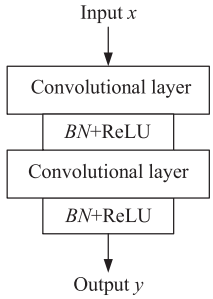

Fig. 2. MK-ResCNN subblock.

C. Multiscale Kernel CNNs

Industrial system fault diagnosis is a time-series signal recognition or regression problem. However, there are still some challenges to solve the problem: first, the single-scale convolutional kernel size makes the network extract features from only one scale. However, signals will not stay in the same scale due to the change of components, systems, or sampling frequency [33], which means that a fixed convolutional kernel size is not suitable for every signal. Second, industrial systems are usually not working under ideal conditions. Many factors may cause the change of signals, such as variable wind speed for wind turbines, changing loads, and mission profiles for engines. Consequently, the ability to analyze signals over changing operation conditions must be considered if we want a fault diagnosis method to be widely applied in industrial systems.

To address these problems, a multiscale kernel CNN (MK-ResCNN) is proposed in this article. First, we construct a basic CNN block as follows:

$$\begin{aligned} y &= \boldsymbol{W} \otimes \boldsymbol{x} + \boldsymbol{b} \\ s &= \mathrm{BN}(y) \\ h &= \mathrm{ReLU}(s) \end{aligned} \tag{5}$$

where $\otimes$ is the convolution operator. BN is the batch normalization operation [34], which can improve generalization and allow us to use much higher learning rates.

Then, the subblock of the MK-CNN is constructed by stacking two basic CNN blocks, as shown in Fig. 2. As described before, residual learning extends the network to a very deep structure without degradation problem, so the residual learning structure is explored between each CNN subblock to construct a deep network for complex feature extraction. Let Basic denotes the basic CNN block, which is corresponding to (5), and the subblock is formalized as follows:

$$\begin{aligned} h_1 &= \mathrm{Basic}(x) \\ h_2 &= \mathrm{Basic}(h_1) \\ y &= h_2 + x \\ \hat{h} &= \mathrm{ReLU}(y). \end{aligned} \tag{6}$$

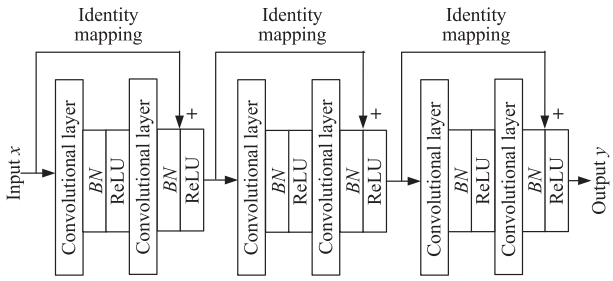

Fig. 3. Illustration of an MK-CNN block, which consists of three subblocks. Identity mapping is applied after BN operation in each subblock to avoid the degradation problem.

Then, a CNN block can be constructed by stacking several subblocks, as shown in Fig. 3. Let subblock denotes the subblock, which is corresponding to (6), then the block can be formalized as follows:

$$\hat{h}_1 = \text{subblock}(x)$$
$$\hat{h}_2 = \text{subblock}(\hat{h}_1)$$
$$\hat{h}_3 = \text{subblock}(\hat{h}_2). \tag{7}$$

Thus, the MK-CNN architecture is constructed by multiple CNN blocks with different convolutional kernel sizes.

We noticed that multiscale CNN has been successfully applied for fault diagnosis of wind turbine gearboxes recently [35]. However, the multiscale in this article means the flexible convolutional kernels, and multiscale in [35] represents flexible averaging strides. That is, in [35], the original input is converted by three ways: $s = 1$, $s = 2$, and $s = 3$, where s is the length of a nonoverlapping window. For example, $s = 2$ is computing the average of every two items in the original signals with a stride of 2. $s = 3$ is computing the average of every three items in the original signals with a stride of 3. In this way, the proposed method in [35] can generate multiscale lengths of original signals by computing average as samples. The proposed method in this article applies different scales of convolutional kernels instead of downsampling or taking averages on original signals. This is because downsampling and taking averages cannot change the outline shape of original signals, which may limit the performance of a CNN. Using different sizes of kernels can learn features from original signals with different views with multiple scales, making CNN learn features in multiscale views.

D. End-to-End Multiscale Residual Learning Architecture

The structure of the MK-ResCNN is graphically illustrated in Fig. 4. In the experiment, time-series signal segments are used as inputs of the MK-ResCNN directly. Since a CNN shows an advantage in feature extraction, we construct an MK-ResCNN architecture with three CNN blocks. In order to extract features from different receptive fields, each block has different convolutional kernels. The kernel sizes of different MK-ResCNN blocks are set to be 1×3, 1×5, and 1×7. This is because the feature map obtained by a layer with big convolutional kernel can also be obtained by multiple layers with small convolutional kernels. Big convolutional kernel may lead to an increase in complexity and computation [36]. Therefore, small convolutional kernels are used more often in practice. Then, unlike traditional CNN models, the pooling operation is excluded in a CNN block here in order to extract more detailed features. After the convolution blocks, the features are fed into a global average pooling layer to keep the robustness against translation of the framework and reduce the number of weights and prevent overfitting [32]. We decided to use an average pooling layer to keep the global feature maps obtained from the convolutional layers (average pooling uses the average value from each of a cluster of neurons at the prior layer). The feature vectors after pooling layers in different blocks are concatenated into a vector as the input of a fully connected network, which is followed by a softmax layer for fault recognition.

There are two reasons why just one fully connected network is added here. First, the parameters in fully connected networks are more than those in a CNN, which makes it hard to train and can lead to the overfitting problem; and second, only one fully connected layer means that features are mostly extracted by a CNN, so the network architecture can be estimated directly from results. That is, if the network is strong enough, only one fully connected layer is needed for classification; if the network is not strong enough, the addition of fully connected layer will not improve the performance a lot. And the adding parameters of the fully connected network will also cause overfitting.

Since mechanical signals are variable and noisy, deeper networks are always needed to extract the deep-hierarchical features, which makes the degradation problem an inevitable problem. To the best of our knowledge, we are the first to provide a multiscale kernel CNN embedded with residual learning for fault diagnosis, which can guarantee that the performance cannot be influenced by the depth of the network [32].

The flowchart of the MK-ResCNN based fault diagnosis method is shown in Fig. 5. There are four steps in this framework.

Step 1: Data acquisition. The motor vibration signals are collected by a data acquisition system and sensors that installed in the test motor.

Step 2: Data segmentation. Since we aim to build an end-to-end diagnosed system to make the system more intelligent, the collected vibration signals are cut into samples and used as training samples directly.

Step 3: Training the MK-ResCNN model. After the vibration signals are cut into samples, the samples are used for MK-ResCNN model training. The Adam algorithm is applied here to optimize the loss function.

Step 4: Fault diagnosis. The vibration signals of the test motor are also cut into samples in the same way as training samples, and used as the input of the trained MK-ResCNN model for fault recognition. The output of softmax regression can reflect the condition or the failure type of the test motor.

III. Experiment and Data Description

In practice, the working conditions of motors are always nonstationary. Because of the variable scales of features, fault diagnosis under nonstationary conditions is much more difficult

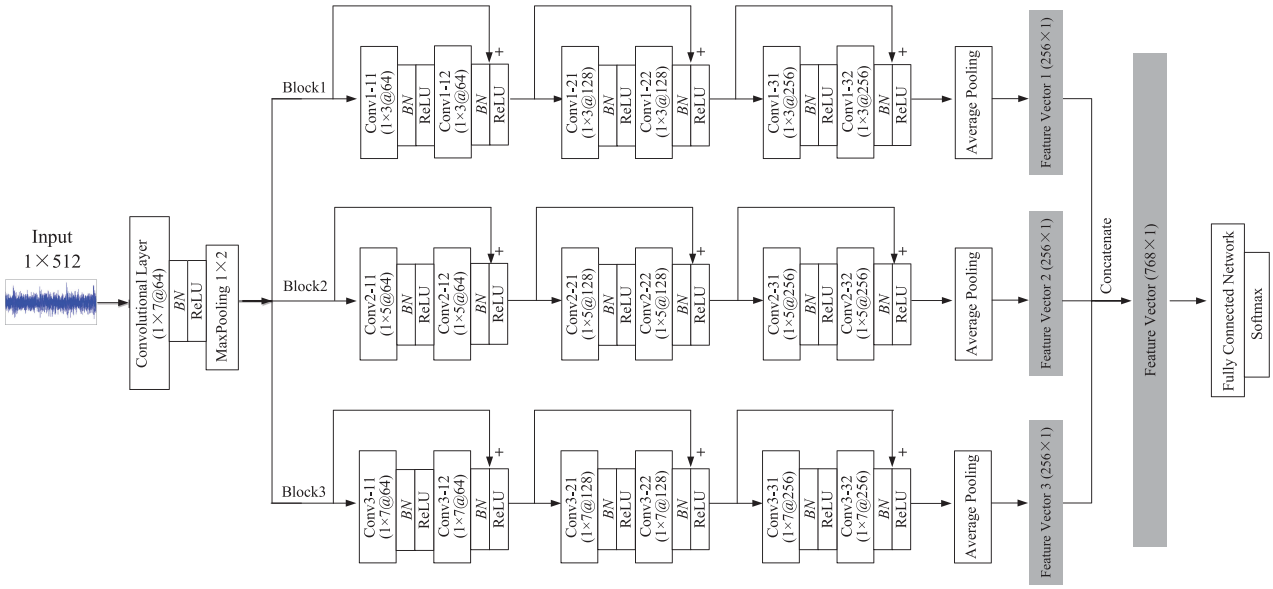

Fig. 4. Structure of the MK-ResCNN. The architecture consists of three CNN blocks with kernel sizes of 1×3, 1×5, and 1×7 respectively. Residual learning is applied in each CNN subblock to avoid the degradation problem.

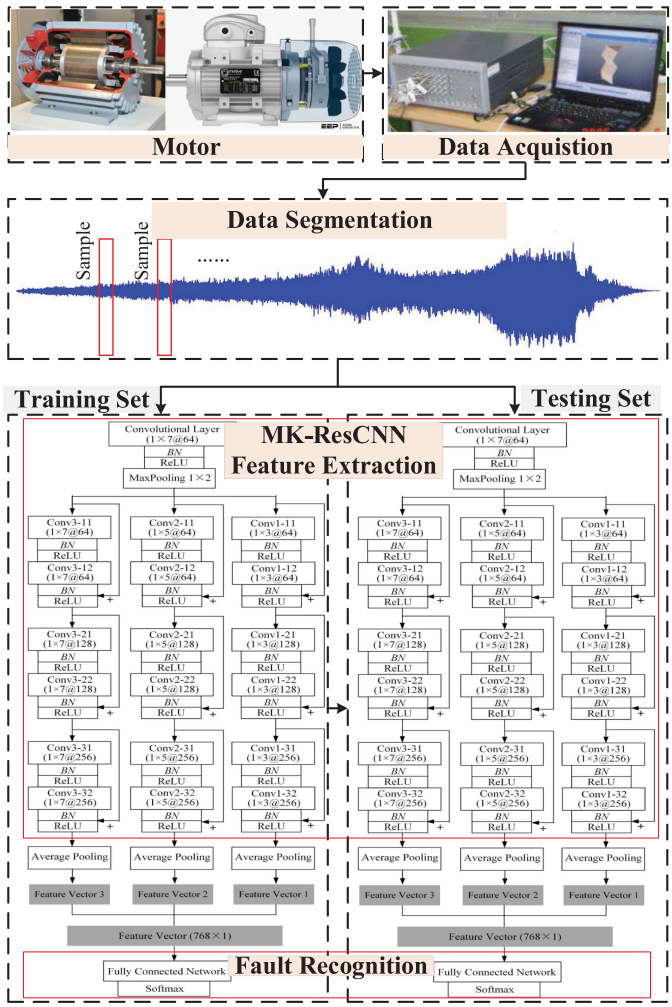

Fig. 5. Flowchart of the MK-ResCNN based fault diagnosis method.

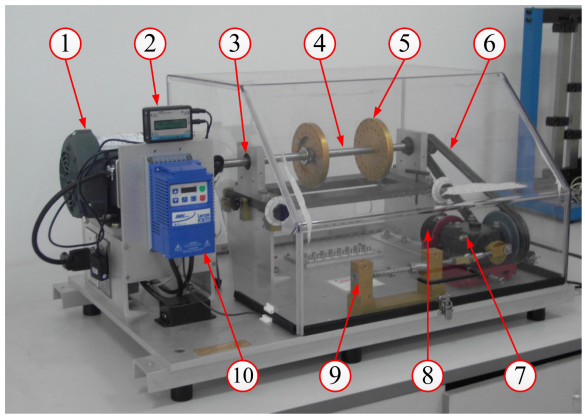

Fig. 6. Experimental setup: (1) induction motor, (2) tachometer, (3) bearing, (4) shaft, (5) load disc, (6) belt, (7) bevel gearbox, (8) magnetic load, (9) reciprocating mechanism, and (10) variable-speed controller.

than that of a constant speed. Studies about the nonstationary operational conditions mostly focus on time–frequency feature extraction of a signal segment, which can be time consuming and poorly generalized. Such features also cannot be used for fault recognition with machine learning methods directly. To verify the effectiveness of the proposed framework in dealing with nonlinear signals, an experiment on an electric machine fault simulator under nonstationary conditions is conducted.

The experiment setup consists of motor, tachometer, bearing, shaft, etc., as shown in Fig. 6. The power supply frequency is 50 Hz. The accelerometer is used to collect the vertical vibration signal. The location of sensor is shown in Fig. 7. The sampling frequency is 12 800 Hz. The rotating speed is controlled manually, which ranges from 0 to 3600 r/min. The rotating speed variations of the test motors are similar with

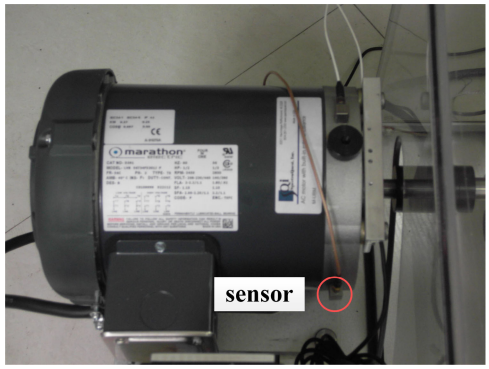

Fig. 7. Sensor location.

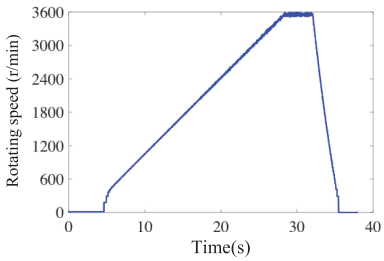

Fig. 8. Rotating speed.

TABLE I
DESCRIPTION OF THE SIX TESTED FAULTY MOTORS

Label	Fault	Description
1	Bowed rotor	Rotor bent in center 0.01
2	Broken rotor bar	An intentionally broken rotor bar
3	Faulty bearing	A motor with an inner race faulty bearing
4	Normal motor	No defect
5	High impedance	Simulates an insulated winding
6	Unbalanced rotor	Intentionally altered rotor caused unbalance

each other, as shown in Fig. 8. The vertical vibration signals are used for analysis in this experiment. There are a normal motor and five motors with different faults are tested in this experiment, including bowed rotor, broken bar, faulty bearing, high impedance, and unbalanced rotor, as described in Table I. Although the high impedance fault is an electrical failure, the simulation of an insulated winding may lead to the change of current signals that go through the stator winding, thus change the electromagnetic vibration of motors, which can be collected by accelerometer. The 2-s vibration signals of the six test motors are shown in Fig. 9. It can be seen that the signals change dramatically during these 2 s. The test motors are three-phase asynchronous motors. The rated power is 735.5 W, and the rated speed is 3600 r/min.

The collected time-series vibration signals are cut into segments as samples. The length of each sample is 512 points. There are 10 788 samples in total. In total, 8630 samples are randomly selected as training samples, whereas the remaining 2158 samples are used as testing samples. The detail

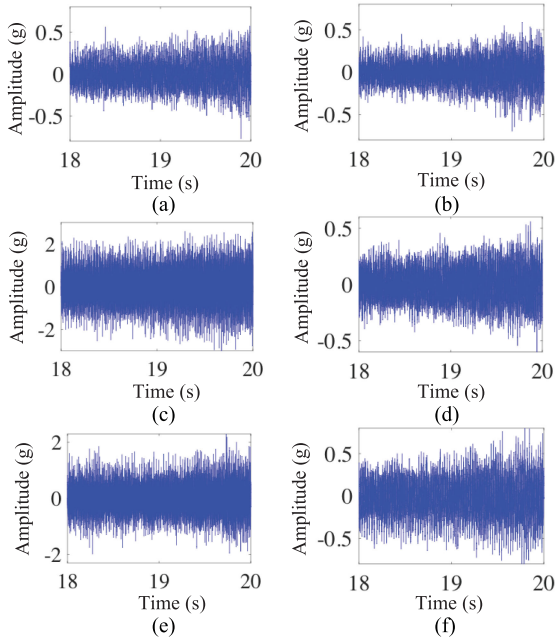

Fig. 9. 2-s vibration samples of different motors. (1) Bowed rotor. (2) Broken rotor bar. (3) Faulty bearing. (4) Normal motor. (5) High impedance. (6) Unbalanced rotor.

TABLE II
NUMBERS OF TRAINING AND TEST SAMPLES

	1	2	3	4	5	6
Total samples	1798	1798	1773	1898	1748	1773
Training samples	1451	1440	1416	1531	1385	1407
Testing samples	347	358	357	367	363	366

description of training samples and testing samples is shown in Table II.

IV. RESULTS AND COMPARISONS

After data collection, the proposed MK-ResCNN method is applied to analyze the vibration signals. Both the training and test procedures are carried out offline. The loss function of the framework is set to be cross entropy loss. To graphically illustrate the learned essential features, t-distributed stochastic neighbor embedding method [37] is employed to provide three-dimensional (3-D) visual representations of the original signals and the feature maps of last layers, as shown in Fig. 10.

The feature map dimensions of the first layer (inputs) and the last layer (fully connect network) are all reduced to three dimensions for feature visualization and easier comparison, as shown in Fig. 10. Different color represents different failure feature. It can be seen that different health states (or classes) are heavily overlapped at the input layer, which demonstrates that the feature information of raw signals are hardly differentiable. And the different failure features extracted by the MK-ResCNN can be easily distinguished or classified this time, thus shows better diagnosed performance, which verifies the effectiveness

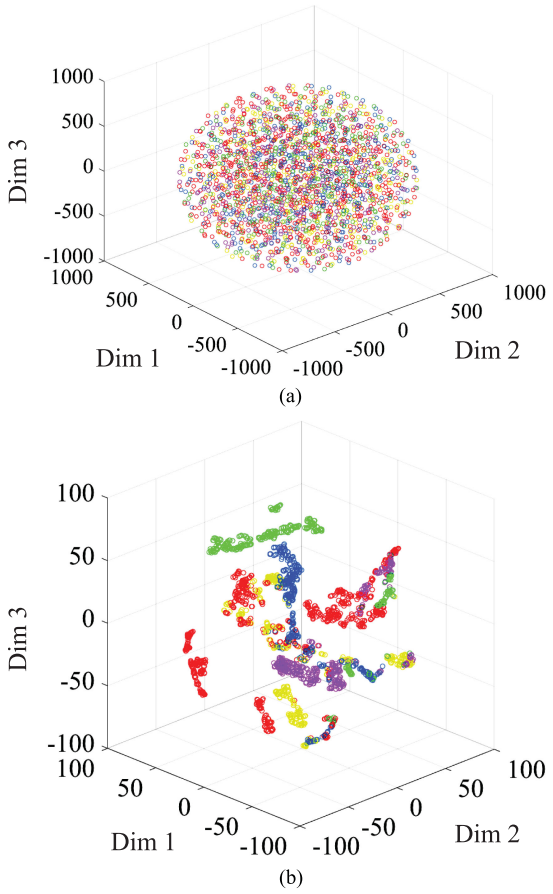

Fig. 10. Feature embedding visualizations of high-dimensional feature maps at different layers in the proposed MK-ResCNN. (a) Original data and (b) output of fully connected layer. Feature maps in the last layer of MK-ResCNN are semantically separable compared to original data, suggesting that the extracted feature maps by the proposed framework are better features for fault diagnosis. Each sample is visualized as a point and samples belonging to the same class have the same color.

TABLE III
CONFUSION MATRIX OF THE PROPOSED MK-RESCNN

label	1	2	3	4	5	6
1	**318**	14	2	7	13	0
2	11	**332**	0	4	4	0
3	4	2	**353**	0	1	0
4	9	9	0	**354**	3	0
5	5	0	2	2	**342**	0
6	0	1	0	0	0	**366**

The bold numbers are the numbers of samples correctly classified.

of the proposed method in learning a discriminative set of features for nonlinear signal fault diagnosis.

The overall accuracy of the MK-ResCNN is 95.69%. To give a more concrete illustration, the confusion matrix of the MK-ResCNN results is shown in Table III. In Table III, the number from 1 to 6 in the first column represents the test data labels in nine different conditions. The number from 1 to 6 in the first row represents the classification result of test data. In addition, the precision ratio p, recall ratio r, and F1 score are used as evaluation indexes for method performance, which are

TABLE IV
CONFUSION MATRIX OF FIVE COMPARISON METHODS AND THE PROPOSED METHOD

label	1	2	3	4	5	6
SVM after Wavelet Processing, accuracy = 47.50%						
p	13.83%	85.47%	74.51%	38.15%	69.15%	3.83%
r	44.44%	24.64%	100%	51.47%	98.05%	100%
F1	21.10%	38.25%	85.39%	43.82%	81.10%	7.37%
ResCNN with kernel size of 1×3, accuracy = 92.03%						
p	86.46%	89.66%	94.12%	95.10%	87.05%	99.45%
r	85.96%	86.99%	97.96%	86.60%	96.05%	99.73%
F1	86.21%	88.31%	96.00%	90.65%	91.33%	99.59%
ResCNN with kernel size of 1×5, accuracy = 92.68%						
p	85.88%	91.90%	94.12%	95.37%	88.43%	99.18%
r	86.63%	88.68%	98.25%	87.72%	95.82%	99.73%
F1	86.25%	90.26%	96.14%	91.38%	91.98%	99.45%
ResCNN with kernel size of 1×7, accuracy = 92.40%						
p	86.74%	90.78%	94.40%	94.28%	87.88%	100%
r	84.08%	87.60%	98.25%	90.10%	95.22%	99.73%
F1	85.39%	89.16%	96.29%	92.14%	91.40%	99.86%
MK-CNN without residual learning, accuracy = 92.22%						
p	84.44%	90.22%	94.12%	95.91%	88.15%	100%
r	85.42%	88.49%	98.82%	86.27%	95.24%	100%
F1	84.93%	89.35%	96.41%	90.48%	91.56%	100%
MK-ResCNN, accuracy = 94.67%						
p	90.20%	91.90%	96.92%	94.82%	93.94%	100%
r	90.20%	92.16%	98.02%	93.05%	94.72%	99.73%
F1	90.20%	92.03%	97.46%	93.93%	94.33%	99.86%

defined as follows:

$$p = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$
$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$
$$F1 = \frac{2p \times r}{p + r} \tag{8}$$

where TP represents the true positive samples, that is, the positive samples that are correctly classified as positive, FP represents the false positive samples, that is, the negative samples but are misclassified as positive samples, and FN represents the false negative samples, that is, the positive samples but are misclassified as negative samples. And the positive sample means the sample that belongs to current failure type, whereas the negative sample means the sample that does not belong to current failure type. p, r, and F1 score of the MK-ResCNN method results are shown in Table IV.

To verify the advantage of the proposed multiscale framework in dealing with nonlinear signals, we compared our results to the five different approaches: manual feature selection with SVM, ResCNNs with kernel size 1×3, ResCNN 1×5, ResCNN 1×7, and MK-CNN. SVM is used to analyze the sample vibration signal segments, with five-layer wavelet packet energies as

features. Three different ResCNNs with filter sizes 1×3, 1×5, and 1×7 are tested respectively. Finally, a multiscale kernel CNN without residual learning step (MK-CNN) is also used to analyze the vibration signals for comparison. The architecture of the MK-CNN stays the same with MK-ResCNN in Fig. 4 except that there is no residual learning. All of the models are trained and tested on the same datasets. The overall accuracies of the comparison methods are 93.19%, 93.47%, 92.59%, and 94.07%. The precision ratio p, recall ratio r, and F1 score of analysis results by the four comparison methods, ResCNN with kernel size of 1×3, ResCNN with kernel size of 1×5, ResCNN with kernel size of 1×7, and MK-CNN without residual learning step, are also shown in Table IV for comparison.

It can be seen from Tables III and IV that all p, r, and F1 score of the proposed method are over 90%, which means that the probability of diagnosing two successive samples falsely is 0.01%. If a sample is diagnosed by mistake for once, we can continuously diagnose the next sample, which will push the result to 100%.

On the other hand, as shown in Table IV, it can be concluded that class 1 is the most difficult class that can be easily diagnosed incorrectly, and the results of class 6 perform the best. These conclusions are the same as the MK-ResCNN results. SVM with wavelet packet features performs worst. Because the feature dimensions of the WT method need to be aligned, each feature dimension should has its physical significance. However, chronological feature extractors cannot make sure the information and physical significance of each dimension in WT remain the same. For example, the first dimensional information of the first sample extracted by WT is the sum energy from 1 to 32 points. And the first dimensional information of the second sample is the sum energy from 513 to 544 points. There is no evidence that the energies of these two segments have the same physical meaning, especially under nonstationary conditions. While it is more likely to find the fault information in time-series signals, the segment of time-series signals can be considered as a small factor. When enough factors have been detected, the industrial system is diagnosed as faulty. The convolution operation in a CNN is more likely to find the pattern. Therefore, a CNN can implement the end-to-end time-series signals classification tasks, whereas WT usually neglects the time-series information. As a result, the diagnosed results of the rest comparison methods all perform better than 80%, which are good results for nonlinear signal analysis compared with SVM, but still nearly 10% worse than the proposed diagnosis framework. The difference between the ResCNNs and the proposed MK-ResCNN is the absence of the multiscale architecture, which cannot only extract features from different scales, but also make the framework more suitable for fault diagnosis under nonstationary conditions. The results demonstrate the effectiveness of this architecture. The last part in Table IV is the analysis results of multiscale CNN without residual learning to show the effectiveness of the residual learning algorithm. It can be seen that the MK-CNN performs better than other three methods, but still worse than the proposed diagnose framework. This is because that the network employed in this article is already a deep network (21 layers in total), which may have led to the degradation problem in deep architectures.

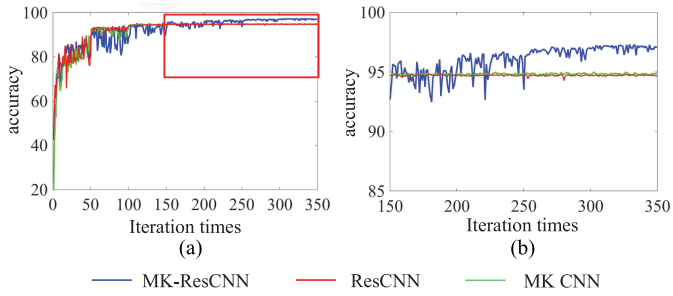

Fig. 11. (a) Training accuracy trends and (b) its local enlargement of the proposed MK-ResCNN method, ResCNN with kernel size of 1×5 and MK-CNN. The blue line represents the training accuracy trend of MK-ResCNN, the red line represents that of ResCNN, and the green line represents that of MK-CNN.

To further illustrate the superiority of the proposed method, the training accuracy trend of the proposed MK-ResCNN method and the alternative methods with the increase in iterations are shown in Fig. 11. During the experiment, we saved the model of every epoch during the training process. The model with the highest test accuracy is applied finally. Since the ResCNN performs best when kernel size is 1×5, the accuracy trends of the rest two ResCNNs are not drawn here. As shown in Fig. 11, at the beginning, the performance of the three methods is nearly the same, and the increased speeds are also similar. After 150 iterations, the proposed MK-ResCNN performs better than the other two methods. The gap between them becomes larger and larger with the increase in iterations. In the end, the training accuracy of the MK-ResCNN method outperforms the other two methods by nearly 2%, which illustrate that both the multiscale algorithm and the residual learning algorithm improve the performance of the diagnosed method.

V. Conclusion

In this article, a novel MK-ResCNN was proposed for motor fault diagnosis under nonstationary conditions. First, residual learning was introduced in the CNN architecture to avoid the degradation problem in deep neural networks. Second, a multiscale architecture was proposed for feature extraction from different scales. Then, a motor failure simulation experiment with time-varying rotating speed was conducted by testing five motors with different faults and one normal motor. The proposed MK-ResCNN was applied to analyze the vibration signals that are collected in the experiment. SVM, different residual learning CNNs without multiscale architecture, and a multiscale CNN without residual learning step were also applied to the same signals for comparison. The results illustrated the effectiveness and outstanding performance. Our work showed that the proposed method not only is an end-to-end fault diagnosis approach that can extract features from raw signals directly, but also is capable for classification tasks of noisy data under nonstationary operation conditions. The multiscale architecture allows the signals to be analyzed from different scales. At the same time, the identity mapping operation covered by residual learning helps the proposed method to extract much deeper features regardless of the degradation problem. The proposed

framework could also be applied to classification of other signals or for fault diagnosis of other components, such as bearing and gearbox, which enlarges its possible applications.

REFERENCES

[1] A. Giantomassi, F. Ferracuti, S. Iarlori, G. Ippoliti, and S. Longhi, "Electric motor fault detection and diagnosis by kernel density estimation and Kullback–Leibler divergence based on stator current measurements," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1770–1780, Mar. 2015.

[2] L. Liao, W. Jin, and R. Pavel, "Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7076–7083, Nov. 2016.

[3] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, 2018.

[4] R. Liu, B. Yang, X. Zhang, S. Wang, and X. Chen, "Time–frequency atoms-driven support vector machine method for bearings incipient fault diagnosis," *Mech. Syst. Signal Process.*, vol. 75, pp. 345–370, 2016.

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[6] R. Yan, R. X. Gao, and X. Chen, "Wavelets for fault diagnosis of rotary machines: A review with applications," *Signal Process.*, vol. 96, pp. 1–15, 2014.

[7] B. Chen, Z. Zhang, C. Sun, B. Li, Y. Zi, and Z. He, "Fault feature extraction of gearbox by using overcomplete rational dilation discrete wavelet transform on signals measured from vibration sensors," *Mech. Syst. Signal Process.*, vol. 33, pp. 275–298, 2012.

[8] J. Antoni, "Fast computation of the Kurtogram for the detection of transient faults," *Mech. Syst. Signal Process.*, vol. 21, no. 1, pp. 108–124, 2007.

[9] B. Yang, R. Liu, and X. Chen, "Fault diagnosis for wind turbine generator bearing via sparse representation and shift-invariant K-SVD," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1321–1331, Jun. 2017.

[10] B. Yang, R. Liu, and X. Chen, "Sparse time-frequency representation for incipient fault diagnosis of wind turbine drive train," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 11, pp. 2616–2627, Nov. 2018.

[11] H. Shao, H. Jiang, F. Wang, and H. Zhao, "An enhancement deep feature fusion method for rotating machinery fault diagnosis," *Knowl.-Based Syst.*, vol. 119, pp. 200–220, 2017.

[12] X.-W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, May 2014.

[13] S. J. Qin, "Process data analytics in the era of big data," *AIChE J.*, vol. 60, no. 9, pp. 3092–3100, 2014.

[14] Z. Liu, Z. Jia, C.-M. Vong, S. Bu, J. Han, and X. Tang, "Capturing high-discriminative fault features for electronics-rich analog system via deep learning," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1213–1226, Jun. 2017.

[15] H. Oh, J. H. Jung, B. C. Jeon, and B. D. Youn, "Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3539–3549, Apr. 2018.

[16] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1310–1320, Jun. 2017.

[17] M. Gan *et al.*, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings," *Mech. Syst. Signal Process.*, vol. 72, pp. 92–104, 2016.

[18] Y. Qin, X. Wang, and J. Zou, "The optimized deep belief networks with improved logistic sigmoid units and their application in fault diagnosis for planetary gearboxes of wind turbines," *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3814–3824, May 2019.

[19] M. Zhao, M. Kang, B. Tang, and M. Pecht, "Multiple wavelet coefficients fusion in deep residual networks for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4696–4706, Jun. 2019.

[20] C. Shi, G. Panoutsos, B. Luo, H. Liu, B. Li, and X. Lin, "Using multiple-feature-spaces-based deep learning for tool condition monitoring in ultraprecision manufacturing," *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3794–3803, May 2019.

[21] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mech. Syst. Signal Process.*, vol. 72, pp. 303–315, 2016.

[22] H. Hu, B. Tang, X. Gong, W. Wei, and H. Wang, "Intelligent fault diagnosis of the high-speed train with big data based on deep neural networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2106–2116, Aug. 2017.

[23] H. Shao, H. Jiang, H. Zhang, and T. Liang, "Electric locomotive bearing fault diagnosis using a novel convolutional deep belief network," *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2727–2736, Mar. 2018.

[24] J. Pan, Y. Zi, J. Chen, Z. Zhou, and B. Wang, "LiftingNet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 4973–4982, Jun. 2018.

[25] M. Zhao, M. Kang, B. Tang, and M. Pecht, "Deep residual networks with dynamically weighted wavelet coefficients for fault diagnosis of planetary gearboxes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4290–4300, May 2018.

[26] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, *arXiv:1505.00387*.

[27] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5353–5360.

[28] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[29] J. Nocedal and S. J. Wright, "Conjugate gradient methods," in *Numerical Optimization*. New York, NY, USA: Springer, 2006, pp. 101–134.

[30] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Mach. Learn.*, vol. 42, no. 3, pp. 287–320, 2001.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[33] M. Misra, H. H. Yue, S. J. Qin, and C. Ling, "Multivariate process monitoring and fault diagnosis by multi-scale PCA," *Comput. Chem. Eng.*, vol. 26, no. 9, pp. 1281–1293, 2002.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[35] G. Jiang, H. He, J. Yan, and P. Xie, "Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3196–3207, Apr. 2019.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2818–2826.

[37] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Ruonan Liu (M'19) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, in 2013, 2015, and 2019, respectively.

She is currently a Postdoctoral Researcher with Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. Her research interests include intelligent manufacturing, computer vision, and machine learning.

Fei Wang (S'18) received the B.S. degree in computer science and technology in 2013 from Xi'an Jiaotong University, Xi'an, China, where he is currently working toward the Ph.D. degree in computer science.

He is also a Visiting Student with the Robostics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include deep learning, time-series mining, and sensory systems.

Boyuan Yang received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, in 2013, 2015, and 2019, respectively.

He is currently a Postdoctoral Researcher with the School of Electrical and Electronic Engineering, The University of Manchester, Manchester, U.K. His research interests include intelligent manufacturing, machine learning, condition monitoring, and wind energy.

S. Joe Qin (F'11) received the B.S. and M.S. degrees in automatic control from Tsinghua University, Beijing, China, in 1984 and 1987, respectively, and the Ph.D. degree in chemical engineering from the University of Maryland, College Park, MD, USA, in 1992.

He is currently the Director of the Center for Machine and Process Intelligence and Fluor Professor with the Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA. His research interests include data analytics, machine learning, process monitoring and fault diagnosis, model predictive control, system identification, semiconductor manufacturing and control, building energy optimization, and predictive maintenance.

Dr. Qin is a Fellow of The Global Home of Chemical Engineers (AIChE) and International Federation of Automatic Control.