Fault Diagnosis for Bearing Based on 1DCNN and LSTM

1. Introduction

With the development and progress of science and technology, machinery has become more automated and intelligent. Because rolling bearings have the advantages such as high efficiency, convenience, and easy installation, they have been widely used in engineering practice, such as mining scraper conveyors, planetary gearboxes, aircraft engines, and marine machinery. But at the same time, its working environment is getting worse and worse. Generally, it requires continuous high-load work in a harsh environment with humid and corrosive gas. Therefore, it is prone to various failures.

Rolling bearings are often used in rotating machinery, and their failure seriously affects the normal operation of rotating machinery. Periodic inspection and after-the-fact diagnosis are often used for traditional fault diagnosis of rotating machinery, but they are hard to carry out and time-consuming for most rotating machinery operating environments. As the important part of the rotating machinery, once the failures of rolling bearings are not dealt with in time, they will directly lead to the halt of machinery and then affect the normal operation of the system, which will not only cause significant economic losses but also threaten the personal safety of workers.

Domestic and foreign scholars have carried out researches based on the rolling bearing operation characteristics and fault characteristics information in various forms and different methods. The researches have achieved positive results in the field of rolling bearing fault diagnosis. Three steps such as feature extraction, feature selection, and fault classification, are generally included in fault diagnosis. Because of the nonstationary nature of vibration signals, the short-time Fourier transform, wavelet analysis, empirical mode decomposition, etc. are usually used for feature extraction [1–3], and then the extracted features are inputted into the network models for classification, such as convolutional neural network, BP neural network, long short-term memory network, Bayesian classification, etc. [4–8]. In the references such as [9, 10], the original signal was first...
converted into a two-dimensional image containing fault information through wavelet transform, and CNN was then used to learn robust features automatically, for fault diagnosis. A novel feature extraction and fault diagnosis method for planetary gears based on variational mode decomposition (VMD), singular value decomposition (SVD), and CNN is proposed by Liu et al. [11]. Yuan et al. [12] proposed an empirical modal decomposition (EMD) and CNN based fault feature extraction method for rotating machinery and investigated its performance on the fault diagnosis of rolling bearing. Although these two methods adopted in [11, 12] had a small amount of calculation, the similarity among the converted matrices is so high that it is difficult to distinguish between them, which will lead to the problem of long model training time and low accuracy. In [13, 14], short-time Fourier transform and two-dimensional CNN are combined for feature extraction and fault classification. However, the process of converting the input into an image takes more time, which increases the workload. Furthermore, decomposing the original signal will inevitably cause the loss of information, which will hurt the diagnostic accuracy.

Although there has been a lot of research on fault diagnosis of bearings, and intelligent diagnosis methods have been widely applied to bearing fault diagnosis, the learning ability of diagnosis models is still weak, which affects the effectiveness and accuracy of diagnosis results negatively. In recent years, deep learning models have been increasingly used in bearing fault diagnosis due to its powerful capabilities of data processing and feature learning. For example, BP neural network (BPNN) was applied to the fault identification of the gearboxes by Zhu et al. [5]. Although the BPNN has a high degree of self-learning and self-adaptation capabilities, its convergence speed is very slow, and it is extremely prone to local minimization and overfitting, which often leads to training failure. Yin et al. [6] proposed a fault diagnosis method for the wind turbine gearbox based on an optimized cosine loss LSTM neural network (Cos-LSTM). Yang et al. [7] proposed an improved LSTM model for the diagnosis of the electromechanical actuators (EMA), which took advantage of the correlation among sensors. RNNs are mostly used to process sequence data, but they are prone to explosion and disappearance of gradients. Although these problems are improved by the LSTM, the training process still takes a lot of time. Bi LSTM [15], another variant of RNN, has been applied to fault diagnosis, but the experimental results show that its diagnosis effect is not very excellent. Therefore, The LSTM was improved with weight amplification [16] and macroscopic-microscopic attention [17] by Qin et al. The improved model was used for the life monitoring of bearings and its validity was proved through experiments.

For the vibration signal of bearings is a natural time series signal, the 1D CNN can be applied to fault diagnosis of bearings. Jing et al. [4] performed a fault detection of the gearbox bearing using the proposed CNN based end-to-end method. Eren et al. [18] proposed an improved adaptive CNN classifier to extract fault characteristics of bearings and diagnosed them in real-time by combining feature extraction and classification of traditional pattern recognition methods. A novel wide first layer kernel deep convolutional neural network (WDCNN) method was proposed by Zhang et al. [19]. During the above methods, the relevance of data points in the original bearing data is easily destroyed during data segmentation, and the information related to fault may be lost. Hence, it is difficult to capture the fault information of the bearing signal over time by relying on the CNN network alone. So, the combination of RNN and CNN was more and more used in fault diagnosis. A novel Convolutional Long Short-Term Memory Recurrent Neural Network (CRNN), with higher accuracy and poorer generalization ability, was proposed by Amin Khorrama et al. [20].

An end-to-end fault diagnosis model using deep learning technology, which employs 1D CNN for feature self-extracting and LSTM for learning the temporal dependencies among features, has been proposed to make up for the above shortcomings in this paper. This model integrates the traditional processes of signal noise reduction, feature extraction, feature selection, and feature classification, which simplifies the diagnosis process to the greatest extent. The combination of LSTM and CNN makes up for the deficiency of using CNN to process time series data alone and improves the robustness of the model. The improved algorithm not only has fewer training parameters and diagnosis time, but also has superior generalization ability, so it is more suitable for dealing with the accurate identification and real-time diagnosis of rolling bearing faults. The proposed algorithm can be extended to the application of fault diagnosis of a variety of rotating machinery.

The paper is comprised of 5 sections. In Section 2, the techniques applied in the proposed model are presented. Section 3 describes the structure of the model in detail. Section 4 conducts simulation tests on the model and does comparison and validation with other classical models. Finally, Section 5 presents the conclusions of the presented research.

2. Methodology

2.1. Convolutional Neural Network. CNN is a typical feed-forward deep neural network with powerful feature extraction capability inspired by primate perception mechanism [21, 22]. It is specifically designed to process data that has a known grid-like topology. It extracts representative features hidden in the input data layer by layer using multiple filters [21], while combining sparse connectivity and parameter weight sharing mechanisms to downsample and refine the data dimensionality in time and space. This reduces the number of training parameters and effectively avoids overfitting of the algorithm. At the same time, it is invariant to other forms of transformation, so it is widely used [21]. The basic architecture diagram of CNN is shown in Figure 1.

A typical convolutional neural network usually includes a convolutional layer, a pooling layer, an activation layer, and a fully connected layer. Convolutional layers are the core component of CNN [22]. The input of one-dimensional convolution is one-dimensional data. Therefore, the dimensionality of its kernel and the dimensionality of its
output are also one-dimensional. The operation of the one-dimensional convolution is shown in Figure 2.

By analyzing Figure 2, the output of the convolution operation is expressed as

$$y_l = \omega_l \ast x_l + b_l,$$

where $l$ is the number of layers of the convolutional layer, $\omega_l$ represents the weight matrix of convolution kernels, $x_l$ represents the input of the layer $l$, $b_l$ is the bias of layer $l$, and $y_l$ is the output of layer $l$.

The pooling layer is another important operation of the convolutional neural network, which improves the robustness of extracted features. Its role is to reduce the dimensionality of the previous layer of data and remove redundant information [23]. This reduces the number of parameters and computation in the network, which is conducive to restraining overfitting and improving the generalization ability of the model. The proposed method in this paper combines the max pooling with the average pooling to process feature maps. The two pooling operations are shown in Figure 3.

Fully connected layer (FCL), which is usually located at the tail of the CNN, plays the role of “classifier” in the convolutional neural network. The softmax is usually applied to the output layer of a multiclassification problem to ensure that the output value ranges from 0 to 1, and the sum of all output values is 1. The output value means the probability of this output, and the one with the highest probability is taken as the final prediction result.

2.2. Long Short-Term Memory Network. Recurrent neural network (RNN) is a kind of deep neural network [24]. In RNN, the information on the time dimension can be shared by adding the connections (i.e., weights) among neurons in the same layer, which makes it suitable to deal with time series problems. LSTM is a special kind of recurrent neural network. It solves the problem of information redundancy by adding a “gate” structure at the appropriate location, which allows information to be selectively retained or forgotten as it flows through each neuron. Due to the ability to strengthen the weight of primary information and weaken the weight of irrelevant information, such problems as gradient disappearance, gradient explosion, and inability to handle long-range dependencies in traditional recurrent neural networks are solved [25] in the LSTM. The internal data operation of the LSTM cell is shown in Figure 4.

In Figure 4, $W_f$ and $b_f$ are the weights and bias of the input gate, respectively, $W_o$ and $b_o$ are the weights and bias of the output gate, respectively, $W_c$ and $b_c$ are the weights and bias of the cell status, respectively, $W_f$ and $b_f$ are the weights and bias of the forget gate, respectively, $h_{t-1}$ represents the output of the previous moment and $x_t$ represents the input of this moment, and they form the new vector $[h_{t-1}, x_t]$. The operation of matrix multiplication of the vector $[h_{t-1}, x_t]$ with $W_c$ and $b_c$ by the tanh activation function to obtain the input value $C_t$ at $t$ moment is calculated as follows:

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c).$$

The operation of matrix multiplication of the vector $[h_{t-1}, x_t]$ with the weights and parametric terms of each gate by the sigmoid activation function yields the forgotten gate $f_t$, the input gate $i_t$, and the output gate $o_t$ at $t$ moment, respectively.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f),$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o).$$

By multiplying the $f_t$ with the last cell state $C_{t-1}$ element by element, the information that needs to be forgotten and remembered can be determined, which realize the control on $C_{t-1}$; by multiplying the $i_t$ with the current input cell state $C_t$ element by element, the information in $C_t$ needed to be saved and utilized will be determined, which can achieve control of $C_t$. Then, the state value of the hidden node at moment $t$ is obtained by summarizing these two products as follows:

![Figure 1: Structure of convolutional neural network.](image-url)
C_t = f_t * C_{t-1} + i_t * C_t'. \quad (4)

Finally, the output at t moment \( h_t \) is determined by the unit state \( C_t \), the output gate \( o_t \), and the tanh function as follows:

\[ h_t = o_t \cdot \tanh(C_t). \quad (5) \]

### 3. Implemented Classifier Based on 1DCNN and LSTM

An end-to-end intelligent fault diagnosis algorithm based on 1DCNN-LSTM, which can quickly and accurately diagnose the faults of rolling bearings under various load conditions, is proposed in this paper.

#### 3.1. Proposed 1DCNN-LSTM Fault Detection Model

The proposed end-to-end fault diagnosis model based on 1DCNN and LSTM includes multiple stacked convolutional layers, pooling layers, LSTM layer, and fully connected layer. The first two pooling layers adopt max pooling to extract the most noticeable and prominent features, and the last layer adopts average pooling to retain more effective information. The LSTM layer is added after the fully connected layer to capture the time dependence of the features extracted by convolution operation and fully characterize the time series data. Finally, the 10-cell dense layer with a softmax activation function as the output function is chosen as the last layer. The softmax function outputs a value set, any one of which ranges from 0 to 1 and is used to determine the fault type. The structure is shown in Figure 5.

#### 3.2. Fault Diagnosis Process

The fault diagnosis based on 1DCNN-LSTM includes two steps. Firstly, the model based on 1DCNN-LSTM is trained and built using training dataset; then, the model can be used to make fault diagnoses on real data. The fault diagnosis flowchart of the 1DCNN-LSTM is shown in Figure 6.

The model building process is described as follows:

Step 1: divide the original data into datasets with different working conditions.

Step 2: perform the operation of overlap sampling on the dataset using sliding windows and add labels to the segmented samples using one-hot encoding [26].
The output of the previous layer =

MaxPooling Kernel =

AveragePooling Kernel =

Figure 3: Schematic diagram of max pooling and average pooling.

\begin{align*}
\text{Input} & \quad b_i \\
& \quad W_i \\
& \quad h_{t-1} \\
& \quad x_t \\
\text{New memory} & \quad b_c \\
& \quad W_c \\
& \quad \text{tanh} \\
& \quad \times \\
& \quad C_t \\
& \quad \text{tanh} \\
& \quad \times \\
& \quad h_t \\
\text{Forget} & \quad b_f \\
& \quad W_f \\
& \quad \sigma \\
& \quad \times \\
& \quad f_t \\
\text{Output} & \quad b_o \\
& \quad W_o \\
& \quad \sigma \\
& \quad o_t \\
\text{Forget} & \quad b_f \\
& \quad W_f \\
& \quad \sigma \\
& \quad \times \\
& \quad f_t \\
\end{align*}

Figure 4: The internal data operation of the LSTM cell.
Step 3: divide the dataset into the training set, validation set, and test set.

Step 4: set the initial parameters of the model.

Step 5: train the model using the training dataset. Tune the parameters by executing the forward propagation and the backward propagation iteratively until the accuracy of the diagnosis on the validation dataset can meet the actual requirements. If the accuracy is satisfying, go to step 6 and save this model; if not, jump to step 4.

Step 6: verify the trained model using the test dataset, and assess the diagnosis ability of this model.

4. Experiment Results and Analysis

4.1. Experimental Conditions. The TensorFlow framework from Google and the python language are used in the experiment of this paper. The experiment program is run on the computer with 1.80 GHz Intel (R) core (TM) i5-8250u CPU and 8 GB memory. The bearing fault dataset collected at the test-stand (shown in Figure 7) of Case Western Reserve University (CWRU) [27] is used as the experimental data. The drive end bearing shown in Figure 7, which is SKF6205, is used as the subject in this paper. The sampling frequency of the platform is 12 kHz. The data samples were collected at 12,000 samples per second, which represent the vibration acceleration velocity values. Three bearing components, such as the inner raceway, the outer raceway, and the rolling element, and four load conditions, such as 0 hp, 1 hp, 2 hp, and 3 hp, were considered and studied in the dataset of CWRU. Three different levels of damage with diameters of 0.007 inches, 0.014 inches, and 0.021 inches, respectively, were implanted in each component of the bearing by the electrodischarge machining (EDM) technique [28], and 9 fault types were got.

4.2. Dataset Description. In this paper, the dataset under the 0 hp load condition is selected for the model validation experiment, and the datasets under 1 hp, 2 hp, and 3 hp load conditions are selected for the performance experiment of the model migration generalization. There are 10 statuses including 9 fault statuses and one normal status in the dataset under every kind of load condition, where the number of samples in each status is 1000 and the total number is 10000. Each sample contains 2048 data points. The time-domain waveform diagrams of 10 bearing operating states are shown in Figure 8, respectively, where the x-axis represents the sampling indices, and the y-axis represents the amplitude. One-hot encoding technology is used to label the sample data (the results are shown in Table 1), and the samples with the same label are randomly divided into three datasets such as the training set, the validation set, and the test set according to the predefined proportions. For each bearing state, 70% of the samples are randomly selected for model training, 20% are used for model validation, and 10% are left for model testing. The dataset partition under different conditions is shown in Table 2.

4.3. Parameter Design. The choice of model parameters has an important impact on the accuracy and generalization ability of the model. The important parameters (i.e., hyperparameters) include the size of the convolution kernel, the number of the convolution kernels, the size of the pooling kernel, the number of the pooling kernels, the activation function, and the optimizer. The selection of hyperparameters plays an important role in the quality and robustness of the model. The suitable hyperparameters can make the model parameters reach the optimal values faster and improve the accuracy of the model. But currently, there is no clear rule for determining the parameters, which heavily depends on experimental experience. In this paper, the different parameters are compared by experiments. These parameters include the parameters of the convolution layer, the number of fully connected neurons, and the number of LSTM neurons. Some results of the experiments are shown in Table 3, where “Conv1d_1” represents the first one-dimensional convolution layer and Conv1d_2 represents the second one-dimensional convolution layer.

It can be seen from Table 3 that increasing the number of convolution layers and the number and the size of the convolution kernels can generally improve the accuracy of...
the model. However, it can be concluded from the No. 8, No. 9, and No. 10 tests that the excessive increase of the number of convolution layers and the number and the size of convolution kernels not only reduces the accuracy but also increases the calculation time remarkably. Then, the hyperparameters of the convolution layers are obtained. In the No. 11 test, the accuracy improves obviously when the number of LSTM neurons is set as 32 and the number of fully

Figure 6: Flowchart of the fault diagnosis based on 1DCNN-LSTM.
Figure 7: Motor driving mechanical system used by CWRU.

Figure 8: Typical vibration signals of 10 faults.

Table 1: The result of one-hot encoding.

| Label | The result of one-hot encoding |
|-------|--------------------------------|
| 0     | [1, 0, 0, 0, 0, 0, 0, 0, 0, 0] |
| 1     | [0, 1, 0, 0, 0, 0, 0, 0, 0, 0] |
| 2     | [0, 0, 1, 0, 0, 0, 0, 0, 0, 0] |
| 3     | [0, 0, 0, 1, 0, 0, 0, 0, 0, 0] |
| 4     | [0, 0, 0, 0, 1, 0, 0, 0, 0, 0] |
| 5     | [0, 0, 0, 0, 0, 1, 0, 0, 0, 0] |
| 6     | [0, 0, 0, 0, 0, 0, 1, 0, 0, 0] |
| 7     | [0, 0, 0, 0, 0, 0, 0, 1, 0, 0] |
| 8     | [0, 0, 0, 0, 0, 0, 0, 0, 1, 0] |
| 9     | [0, 0, 0, 0, 0, 0, 0, 0, 0, 1] |
connected neurons is set as 64. From the No. 12 and No. 13 tests, it can be seen that the increase of the number of LSTM neurons and fully connected neurons does not improve the performance.

Based on the above results, different pooling strategies, activation functions [29, 30], and optimizers [31, 32] are combined and applied on the proposed model with the hyperparameters of No. 11 test in Table 3. The performance comparison is shown in Table 4, where “Pooling_1” represents the first pooling layer and “Pooling_2” represents the second pooling layer and “Pooling_3” represents the third pooling layer.

In the No. 20 test of Table 4, the accuracies of training and test are 1.0 and 0.999, respectively, which represents the best effect of the proposed model with different pooling strategies, activation functions, and optimizers. In addition, Adam is identified as the best optimizer of the model. fX_hus, the 1DCNN-LSTM model has superior fault diagnosis significance for accurate diagnosis in practical applications. 4.4. Performance of Migration and Generalization. In the actual application of rotating machinery, the load on the bearings may change at any time. So, the migration and generalization performance of the proposed model is tested in this paper. The experimental results are compared with multilayer perceptron network (MLP), BPNN [33], LSTM [34], LeNet-5 [35], WDCNN [19], and CRNN [20]. The number of parameters, epochs, training time, and test running time of these models are listed in Table 6. The comparison results of the experiments are shown in Figure 9, where “A ⟷ B” represents that the training set in dataset A is selected as the training data and the test data in dataset B is selected as the test data.

As shown in Figure 9, MLP, whose average accuracy is less than 70%, performs the worst in migration and generalization. LeNet-5 and CRNN perform better, both reaching approximately 80% accuracy. The average accuracy of WDCNN reaches about 90%. In contrast, the proposed model is much more accurate than the other models, achieving more than 95%, which proves that the proposed model can adapt to the changes of the load, which is of great significance for accurate diagnosis in practical applications. Thus, the 1DCNN-LSTM model has superior fault diagnosis capabilities and is more suitable for the accurate diagnosis of bearing faults.

Then, the confusion matrix is introduced to show the migration and generalization ability of the proposed model clearly and the results are shown in Figure 10, where the horizontal axis represents the type of the bearing fault predicted by the 1DCNN-LSTM model, and the vertical axis represents the true type of the bearing fault. It can be concluded that the 1DCNN-LSTM model can achieve stable and efficient classification under different load conditions. In summary, the proposed model, which has superior migration and generalization performance and achieves high diagnostic accuracy under different load conditions, can efficiently extract the fault characteristics of the bearing and realize the effective classification of the fault.

### Table 2: Description of the CWRU dataset.

| Fault location | Normal | Inner race | Outer race | Ball |
|----------------|--------|------------|------------|------|
| Damage diameter (inches) | 0 | 0.007 | 0.014 | 0.021 |
| Label | 0 | 1 | 2 | 3 |
| Train | 700 | 700 | 700 | 700 |
| Validation | 200 | 200 | 200 | 200 |
| Test | 100 | 100 | 100 | 100 |
| Dataset A | 700 | 700 | 700 | 700 |
| Validation | 200 | 200 | 200 | 200 |
| Test | 100 | 100 | 100 | 100 |
| Dataset B | 700 | 700 | 700 | 700 |
| Validation | 200 | 200 | 200 | 200 |
| Test | 100 | 100 | 100 | 100 |
| Dataset C | 700 | 700 | 700 | 700 |
| Validation | 200 | 200 | 200 | 200 |
| Test | 100 | 100 | 100 | 100 |
| Dataset D | 700 | 700 | 700 | 700 |
| Validation | 200 | 200 | 200 | 200 |
| Test | 100 | 100 | 100 | 100 |

Load (hp)

| Train | 700 | 700 | 700 | 700 |
| Validation | 200 | 200 | 200 | 200 |
| Test | 100 | 100 | 100 | 100 |

Table 3: Hyperparameters of No. 11 test.
**Table 3: Performance comparison of the model with different hyperparameters.**

| No. | Number of conv1d_1 kernels | Conv1d_1 kernel size | Number of conv1d_2 kernels | Conv1d_2 kernel size | Number of conv1d_3 kernels | Conv1d_3 kernel size | Number of conv1d_4 kernels | Conv1d_4 kernel size | Fully connected neurons | LSTM neurons | Train accuracy | Test accuracy | Calculation time (seconds) |
|-----|-----------------------------|----------------------|-----------------------------|----------------------|-----------------------------|----------------------|-----------------------------|----------------------|------------------------|--------------|----------------|--------------|------------------------|
| 1   | 16                          | 64                   | —                           | —                    | —                           | —                    | —                           | —                    | 32                     | 16           | 0.963         | 0.944        | 36                      |
| 2   | 32                          | 64                   | —                           | —                    | —                           | —                    | —                           | —                    | 32                     | 16           | 0.972         | 0.947        | 37                      |
| 3   | 64                          | 64                   | —                           | —                    | —                           | —                    | —                           | —                    | 32                     | 16           | 0.974         | 0.952        | 39                      |
| 4   | 32                          | 128                  | —                           | —                    | —                           | —                    | —                           | —                    | 32                     | 16           | 0.989         | 0.922        | 40                      |
| 5   | 32                          | 64                   | 32                          | 5                    | —                           | —                    | —                           | —                    | 32                     | 16           | 0.963         | 0.918        | 42                      |
| 6   | 32                          | 64                   | 64                          | 5                    | 64                          | 3                    | —                           | —                    | 32                     | 16           | 0.995         | 0.974        | 43                      |
| 7   | 32                          | 64                   | 32                          | 5                    | 128                         | 3                    | —                           | —                    | 32                     | 16           | 0.991         | 0.978        | 63                      |
| 8   | 32                          | 64                   | 32                          | 5                    | 64                          | 3                    | 64                          | 3                    | 32                     | 16           | 0.985         | 0.973        | 67                      |
| 9   | 32                          | 64                   | 32                          | 5                    | 64                          | 3                    | 128                         | 3                    | 32                     | 16           | 0.978         | 0.972        | 72                      |
| 10  | 32                          | 64                   | 32                          | 5                    | 64                          | 3                    | 128                         | 3                    | 32                     | 16           | 0.990         | 0.967        | 43                      |
| 11  | 32                          | 64                   | 32                          | 5                    | 64                          | 3                    | 64                          | 3                    | 32                     | 32           | 1.0           | 0.990        | 43                      |
| 12  | 32                          | 64                   | 32                          | 5                    | 64                          | 3                    | —                           | —                    | 64                     | 64           | 0.982         | 0.967        | 43                      |
| 13  | 32                          | 64                   | 32                          | 5                    | 64                          | 3                    | —                           | —                    | 128                    | 32           | 0.985         | 0.963        | 43                      |

The bold values represent a set of parameters with the best experimental results. They are also the parameters of convolutional layer, LSTM layer, and fully connected layer that the model proposed in this paper finally chooses.
5. Stability Performance under Different Datasets

The learning ability and stability of the proposed model under different load conditions should be verified. First, the four datasets named A, B, C, and D are used to train and test the model. The experimental results are shown in Figure 11, which reveals the training process and test process curves of the model on the dataset under different load conditions.

Obviously, the training process and testing process are stable, which proves that the proposed model has good learning ability on datasets with different loads. Sometimes it is far from scientific and comprehensive to evaluate an algorithm model only by the accuracy, so the precision, recall, and F1 [36], whose meaning are explained in equations (6)–(8), respectively, are used to evaluate the model. The evaluation results are shown in Table 7.

\[
\text{precision} = \frac{TP}{TP + FP} \tag{6}
\]

where the TP represents the number of positive samples predicted as positive by the model and the FP represents the number of negative samples predicted as positive by the model.

\[
\text{recall} = \frac{TP}{TP + FN} \tag{7}
\]

where the FN represents the number of positive samples predicted as negative by the model.

\[
F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{8}
\]

where the precision represents the proportion of true positive samples in the total samples predicted as positive by the model,

Table 4: Performance comparison of the models with different pooling strategies, activation functions, and optimizers.

| No. | Pooling_1 strategies | Pooling_2 strategies | Pooling_3 strategies | Activation functions | Optimizer   | Train accuracy | Test accuracy |
|-----|-----------------------|----------------------|----------------------|----------------------|-------------|---------------|---------------|
| 14  | Max                   | Max                  | Max                  | Elu                  | Adam        | 0.971         | 0.944         |
| 15  | Max                   | Max                  | Max                  | Relu                 | RMSprop     | 0.984         | 0.955         |
| 16  | Max                   | Max                  | Max                  | Sigmoid              | Adam        | 0.964         | 0.932         |
| 17  | Average               | Average              | Average              | Elu                  | Adam        | 0.990         | 0.957         |
| 18  | Average               | Average              | Average              | Relu                 | RMSprop     | 0.963         | 0.935         |
| 19  | Average               | Average              | Average              | Sigmoid              | Adam        | 0.981         | 0.968         |
| 20  | Max                   | Max                  | Average              | Elu                  | Adam        | **1.0**       | **0.999**     |
| 21  | Max                   | Average              | Max                  | Relu                 | Adam        | 0.989         | 0.981         |
| 22  | Average               | Max                  | Sigmoid              | Adam                 |            | 0.990         | 0.979         |

The bold values represent a set of parameters with the best experimental results. They are also the parameters of pooling layer, activation function, and optimizer that the model proposed in this paper finally chooses.

Table 5: Hyperparameter of the 1DCNN-LSTM.

| Network layer       | Parameters | No. of filters | Size of filter | Stride | Padding | Activation |
|---------------------|------------|----------------|----------------|--------|---------|------------|
| Convolution layer 1 |            | 32             | 64 * 1         | 32     | Same    | Elu        |
| Max pooling layer 1 |            | 1              | 4 * 1          | 2      | Same    |            |
| Convolution layer 2 |            | 32             | 5 * 1          | 1      | Same    | Elu        |
| Max pooling layer 2 |            | 1              | 4 * 1          | 2      | Same    |            |
| Convolution layer 3 |            | 64             | 3 * 1          | 1      | Same    | Elu        |
| Average pooling layer 1 | | 1              | 4 * 1          | 2      | Same    |            |
| Fully connected layer |        | 64             | —              | —      | —       | Elu        |
| LSTM layer          |            | 32             | —              | —      | —       | —          |
| Output layer        |            | 10             | —              | —      | —       | —          |

Table 6: Performance comparison of different models.

| Number of parameters | Epochs | Train time (s) | Test time (s) |
|----------------------|--------|----------------|---------------|
| LeNet-5              | 5121746| 20             | 625           | 0.794        |
| BPNN                 | 659722 | 20             | 213           | 0.917        |
| MLP                  | 566346 | 20             | 156           | 0.862        |
| LSTM                 | 659722 | 20             | 432           | 2.651        |
| WDCNN                | 54790  | 20             | 107           | 0.351        |
| CRNN                 | 21210  | 20             | 73            | 0.149        |
| 1DCNN-LSTM           | **28298**| 20             | **43**        | **0.153**    |

The bold values represent the parameters used in the proposed model in this paper.
and the recall represents the proportion of the samples predicted as positive by the model in the total positive samples.

As can be seen from Table 7, the precision and recall of the proposed model for each fault type under different datasets are very high. It shows that the model can learn the robustness characteristics of faults, which is of great significance to fault diagnosis in practical applications.

Then, the dataset of XJTU-SY [37] is used to further verify the stability and learning ability of the proposed model. As shown in Figure 12, the bearing test bench consists of an alternating current (AC) induction motor, a motor speed regulator, a support shaft, two support bearings (heavy roller bearings), and a hydraulic loading system.
A total of three different operating conditions are set in the accelerated degradation experiments. The horizontal vibration signals of different working parts under the above three working conditions are integrated and used as experimental data in this paper. The description of the dataset is shown in Table 8, where 37.5 Hz represents the vibration frequency, 12 kN represents the load of the bearing, and the meanings of the other two conditions are similar. The experimental results are shown in Figure 13.

As can be seen from Figure 13, the accuracy of the model is 1.0 in the third training and remains stable in subsequent training. During testing, although the accuracy fluctuates at the beginning, it continues to rise after the learning rate decreases, and the accuracy becomes stable in the last 5 epochs.
second training and remains high and stable after the seventh training. The powerful feature learning ability of the proposed model can be proved. It shows that the proposed model can be applied to the bearing diagnosis if the vibration dataset is provided.

5.1. Classification Performance Visualization. Although the 1DCNN-LSTM model has powerful feature extraction and fault classification capabilities, the model process is equivalent to a black box. In order to further prove the effectiveness of this method, the t-distributed stochastic neighbor
embedding (T-SNE) technology [38] is used to visualize the classification effects. After the training is completed, the test data is input into the model. The outputs of the first convolution layer, the last convolution layer, the LSTM layer, and the last hidden layer of the model are reduced in dimension and then visualized. The visualization results are shown in Figure 14. It can be found that as the number of convolutional layer increases, the overlapping part of each category gradually decreases, and the recognition rate is improved. In addition, it can also be seen that the distances among the fault samples are relatively close in the output of the last convolutional layer, but the classification result becomes clearer after the LSTM is adopted. The results show that although CNN has strong capabilities in feature extraction, it has limitations in fault classification, and it is very necessary to employ the LSTM algorithm, which can improve the ability of fault classification. In summary, the proposed model has good classification performance, and it is practical and effective in fault diagnosis of bearings for rotating machinery.

6. Conclusions

Intelligent fault diagnosis method based on deep learning is a hot topic of current research within the field of fault diagnosis. In this paper, a novel technique based on 1DCNN and LSTM is presented to detect bearing faults, and its performance is evaluated. By applying the algorithm to the fault dataset of rolling bearings for diagnosis, the accuracy and superiority of the method are verified, and the following conclusions are obtained:

1. The 1DCNN-LSTM model can be applied to rolling bearing fault diagnosis. 1DCNN that can learn fault features autonomously from one-dimensional vibration signals is used to extract fault features. LSTM is used to learn the correlation among features. By combining the virtues of the 1DCNN and LSTM, the 1DCNN-LSTM can achieve higher diagnostic accuracy.

2. The rolling bearing fault diagnosis method proposed in this paper can accurately identify and locate the fault position of bearings, such as the inner raceway, the outer raceway, and the rolling element with different damage diameters, which can avoid irreversible damage to the bearing.

3. The experiments show that the proposed model can learn features without human expertise and any prior knowledge of fault diagnosis, which has high feasibility and generality. In addition, the CWRU dataset and the XJTU-SY dataset are used to verify the superior performance of the proposed model. The recognition rate of the proposed model on the CWRU dataset and XJTU-SY dataset reached 99.98% and 99.95%, respectively, and the recognition time reaches 0.153 s. The proposed
model, which is suitable for online real-time monitoring and rapid fault diagnosis, can also be expanded to the fault detection of other rotating machinery.

The limitations of the proposed model are as follows: the dataset used in the proposed model is based on a single sensor signal. However, there may be many kinds of signals to be considered when there is a problem with machinery. So the data fusion techniques for the multiple sensors and the new classification model will be the considerations of future research work.

Data Availability

Previously reported CWRU data were used to support this study and are available at https://cseggroups.case.edu/bearingdatacenter/pages/download-data-file. These prior studies (and datasets) are cited at relevant places within the text as [27]. Previously reported XJTU-SY data were used to support this study and are available at https://biaowang.tech/xjtu-sy-bearing-datasets/. These prior studies (and datasets) are cited at relevant places within the text as [37].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

References

[1] R. B. W. Heng and M. J. M. Nor, “Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition,” Applied Acoustics, vol. 53, no. 1, pp. 211–266, 1998.
[2] K. Shibata, A. Takahashi, and T. Shirai, “Fault diagnosis of rotating machinery through visualisation of sound signals,” Mechanical Systems and Signal Processing, vol. 14, no. 2, pp. 229–241, 2000.
[3] N. Baydar and A. Ball, “Detection of gear failures via vibration and acoustic signals using wavelet transform,” Mechanical Systems and Signal Processing, vol. 17, no. 4, pp. 787–804, 2003.
[4] L. Jing, M. Zhao, P. Li, and X. Xu, “A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox,” Measurement, vol. 111, pp. 1–10, 2017.
[5] L. Zhu, G. Tang, and X. Hu, “Fault identification of gearbox based on BP neural network improved by artificial fish swarm algorithm,” International Core Journal of Engineering, vol. 5, no. 8, 2019.
[6] A. Yin, Y. Yan, Z. Zhang, C. Li, and R.-V. Sánchez, “Fault diagnosis of wind turbine gearbox based on the optimized LSTM neural network with cosine loss,” Sensors, vol. 20, no. 8, Article ID 2339, 2020.
[7] J. Yang, Y. Guo, and W. Zhao, “Long short-term memory neural network based fault detection and isolation for electro-mechanical actuators,” Neurocomputing, vol. 360, pp. 85–96, 2019.
[8] W. Sun, A. R. C. Paiva, P. Xu, A. Sundaram, and R. D. Braatz, “Fault detection and identification using Bayesian recurrent neural networks,” Computers & Chemical Engineering, vol. 141, Article ID 106991, 2020.
[9] Y. Cheng, M. Lin, J. Wu, H. Zhu, and X. Shao, “Intelligent fault diagnosis of rotating machinery based on continuous wavelet transform-local binary convolutional neural network,” Knowledge-Based Systems, vol. 216, Article ID 106796, 2021.
[10] Y.-M. Hsueh, V. R. Ittangihal, W.-B. Wu, H.-C. Chang, and C.-C. Kuo, “Fault diagnosis system for induction motors by CNN using empirical wavelet transform,” Symmetry, vol. 11, no. 10, Article ID 1212, 2019.
[11] C. Liu, G. Cheng, X. Chen, and Y. Pang, “Planetary gears feature extraction and fault diagnosis method based on VMD and CNN,” Sensors, vol. 18, no. 5, Article ID 1523, 2018.
[12] X. Yuan, T. Zhang, and G. Boscato, “Fault diagnosis for rotating machinery based on convolutional neural network and empirical mode decomposition,” Shock and Vibration, vol. 2017, Article ID 3084197, 12 pages, 2017.
[13] M. P. Tuan, J. M. Kim, and C. H. Kim, “Accurate bearing fault diagnosis under variable shaft speed using convolutional neural networks and vibration spectrogram,” Applied Sciences, vol. 10, no. 18, Article ID 6385, 2020.
[14] Z. Chen, C. Li, and R.-V. Sanchez, “Gearbox fault identification and classification with convolutional neural networks,” Shock and Vibration, vol. 2015, Article ID 390134, 10 pages, 2015.
[15] T. Zhang, Q. Fei, N. Li, and D. Ma, “Fault diagnosis based on modified BiLSTM neural network,” in Proceedings of the 2020 5th International Conference on Intelligent Information Technology (ICIIT 2020), pp. 21–26, Hanoi, Viet Nam, February 2020.
[16] S. Xiang, Y. Qin, C. Zhu, Y. Wang, and H. Chen, “Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction,” Engineering Applications of Artificial Intelligence, vol. 91, Article ID 103587, 2020.
[17] Y. Qin, S. Xiang, Y. Chai, and H. Chen, “Macroscopic–microscopic attention in LSTM networks based on fusion features for gear remaining life prediction,” IEEE Transactions on Industrial Electronics, vol. 67, no. 12, pp. 10865–10875, 2019.
[18] L. Eren, T. Ince, and S. Kiranyaz, “A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier,” Journal of Signal Processing Systems, vol. 91, no. 2, pp. 179–189, 2019.
[19] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, “A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals,” Sensors, vol. 17, no. 2, p. 425, 2017.
[20] K. Amin, M. Khalaoei, and M. Rezghi, “End-to-end CNN+LSTM deep learning approach for bearing fault diagnosis [J],” Applied Intelligence, vol. 51, pp. 736–751, 2020.
[21] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning[M], The MIT Press, Cambridge, MA, 2016.
[22] D. H. Hubel and T. N. Wiesel, “Receptive fields of single neurones in the cat’s striate cortex,” The Journal of Physiology, vol. 148, no. 3, pp. 574–591, 1959.
[23] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[24] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[25] R. Pascanu, T. Mikolov, and Y. Bengio, On the Difficulty of Training Recurrent Neural Networks, JMLR.org, 2012.
[26] L. Jia, Y. Si, T. Xu, and S. Jiang, “Deep convolutional neural network based ECG classification system using information
fusion and one-hot encoding techniques,” *Mathematical Problems in Engineering*, vol. 2018, pp. 1–10, 2018.

[27] The Case Western Reserve University Bearing Data Center, *Bearing Data center Seeded Fault Test Data [EB/OL]*, 2012.

[28] P.-H. Heeren, D. Reynaerts, H. V. Brussel, C. Beuret, O. Larsson, and A. Bertholds, “Microstructuring of silicon by electro-discharge machining (EDM)–part II: applications,” *Sensors and Actuators A: Physical*, vol. 61, no. 1–3, pp. 379–386, 1997.

[29] M. Xia, T. Li, L. Xu, L. Liu, and C. W. de Silva, “Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks,” *IEEE*, vol. 23, no. 1, pp. 101–110, 2018.

[30] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (ELUs),” *Communications and Computer Sciences*, 2015, https://arxiv.org/abs/1511.07289.

[31] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2010.

[32] D. Kingma and J. Ba, “Adam: a method for stochastic optimization,” *Proceedings of ICLR*, 2015, https://arxiv.org/abs/1412.6980v8.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[34] M. Sundermeyer, R. Schlüter, and H. Ney, *LSTM Neural Networks for Language Modeling*, Interspeech, Shanghai, China, 2012.

[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] P. Flach, *Machine Learning [M]*, Cambridge University Press, Cambridge, UK, 2012.

[37] B. Wang, Y. Lei, N. Li, and N. Li, “A hybrid prognostics approach for estimating remaining useful life of rolling element bearings,” *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 401–412, 2020.

[38] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.