Fault Diagnosis of Motor Bearings Based on a Convolutional Long Short-Term Memory Network of Bayesian Optimization

ZHEN LI1, YANG WANG1,2, AND JIANENG MA1
1School of Electrical Engineering, Shanghai Dianji University, Shanghai 201306, China
2Department of Control Science and Engineering, Zhejiang University, Hangzhou 310000, China
Corresponding author: Yang Wang (wangyang@sdju.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62003206.

ABSTRACT As the main driving equipment of modern industrial production activities, if a motor fails, it causes serious consequences. Bearings are the component with the highest motor failure frequency. It is of practical engineering significance to establish a high-precision algorithm diagnostic model for motor bearings. At present, in data-driven motor bearing fault diagnosis methods, the method of manually adjusting hyperparameters is usually adopted in complex network structure models with many hyperparameters. To realize the automatic optimization selection of hyperparameters, in this paper, a motor bearing fault diagnosis algorithm based on a convolutional long short-term memory network of Bayesian optimization (BO-CLSTM) is proposed. The algorithm combines the Bayesian optimization algorithm (BO), a long short-term memory network (LSTM) and the convolutional layer of a convolutional neural network (CNN). It saves the considerable workload of manually adjusting the hyperparameters, has good noise resistance, and realizes the true end-to-end motor bearing fault diagnosis. The proposed method is trained based on the original vibration signal of the bearing, and the accuracy of the final model reaches 100%. In addition, compared with other advanced fault diagnosis methods based on deep learning, the performance of the proposed method is significantly improved.

INDEX TERMS Deep learning, LSTM, Bayesian optimization, fault diagnosis of motor bearings.

I. INTRODUCTION

With the rapid development of science and technology, modern industrial technology has made great progress. This progress moves industrial equipment in the direction of large-scale, complex and intelligent equipment [1]. Bearing plays the role of bearing and transferring load in motor equipment. It often works in bad conditions. Under the influence of high temperature, high speed, erosion, wear and other factors, bearings have become one of the most vulnerable parts of the motor [2], [3]. If the bearing fails and is not maintained or repaired in time, it may cause the equipment to malfunction. Therefore, it has special practical significance to establish an intelligent, efficient and fast fault diagnosis system for motor bearings. Adopting fault diagnosis systems, we can timely find the abnormal position and damage severity of the motor bearing while ensuring the normal operation of the motor and avoiding huge losses caused by accidents. Implementing fault diagnosis systems also enables condition-based maintenance strategies so as to maximize the useful life of the motor.

In recent years, deep learning has developed rapidly. It has made great achievements in the fields of image processing, machine vision and speech recognition [4]–[6]. Deep learning is a breakthrough in the field of modern artificial intelligence [7]. The recognition rate of many traditional recognition tasks based on deep learning has increased significantly. Deep learning has incomparable advantages in feature extraction and selection, nonlinear fitting, and pattern recognition. It is widely used in various fields, and it also provides new ideas for motor bearing fault diagnosis and prediction [8], [9]. Because of its special network structure and ability to handle complex recognition tasks, a large
number of scholars have been attracted to study its theories and applications. For example, Li Yanfeng et al. proposed a novel approach to rolling bearing fault diagnosis using singular value decomposition (SVD) and multiple deep belief network (DBN) classifiers. This method reconstructs the vibration signals of rolling bearings under different conditions in phase space and obtains the feature matrix. Then, the feature matrices are decomposed by SVD to obtain the singular values. Finally, a multiple DBN classifier model was developed to identify rolling bearing faults [10]. Wang et al. proposed a bearing fault diagnosis method based on the Hilbert envelope spectrum and DBN. Compared with other methods, this method can greatly simplify the fault diagnosis process and obtain better diagnosis performance [11]. Traditional intelligent fault diagnosis systems usually encapsulate feature extraction, feature selection and classification into different modules. To realize the intelligent diagnosis of bearings, many scholars have performed much research on learning the optimal features automatically and effectively through appropriate training by directly taking the original time series data of sensors as input. In [12], a layered algorithm based on stacked LSTM was proposed to overcome the shortcomings of shallow structures, and the framework of rolling bearing fault diagnosis systems was constructed by using this algorithm. In [13], aiming at fault visualization and automatic feature extraction, this paper presented a new and intelligent bearing fault diagnostic method by combining symmetrized dot pattern (SDP) representation with a squeeze-and-excitation-enabled convolutional neural network (SE-CNN) model. Experimental results show that this method has good generalization ability. In [14], Gong et al. proposed a novel improved CNN-SVM method and applied it to the rapid intelligent fault diagnosis of motor rolling bearings. The results showed that the improved CNN-SVM algorithm improves the accuracy of fault recognition. In [15], a compact adaptive one-dimensional convolutional neural network (CNN) classifier was used to directly take the original time series data of sensors as input for bearing fault diagnosis. In [16], convolutional neural networks with two convolution kernels of different sizes and long short-term memory networks were used to directly input the original signals for bearing fault diagnosis. In [17], a comprehensive benchmark study of DL-based classic models was carried out using different data segmentation strategies, input formats, normalization methods and enhancement methods on seven datasets. More research on deep learning technology in bearings can be found in [18]–[20]. These methods have played a certain role in rolling bearing fault diagnosis, but there are still some areas that can be improved. (1) In previous studies, the original time-domain signal input model was directly classified. Although good results have been achieved, the final diagnosis accuracy can be improved. (2) In the motor bearing fault diagnosis algorithm based on deep learning, there are many hyperparameters that determine the accuracy of the algorithm model. At present, the work of hyperparameter selection requires considerable time and computation. It is very meaningful to use an efficient optimization algorithm to realize the fast and automatic selection of hyperparameters. (3) In the actual fault diagnosis of motor bearings, the collected signals are often combined with considerable random noise. This noise interferes with the diagnostic accuracy of the model. Reducing noise interference on the fault diagnosis model and improving the robustness of the motor bearing fault diagnosis model are problems to be solved in practical applications.

Based on the previous research foundation, this paper proposes a motor bearing fault diagnosis algorithm with automatic optimization of hyperparameters. The algorithm combines the convolutional layer of the CNN and LSTM and uses the Bayesian optimization method to optimize the hyperparameters of the model. It can save the workload and time of hyperparameter adjustment and truly realize end-to-end intelligent fault diagnosis technology. As shown in Figure 1. The main contributions of this research are as follows:

1) A new motor bearing fault diagnosis algorithm is proposed. Based on the original time-domain signal, the motor bearing fault diagnosis model is established directly to obtain the weak features between the data to improve the accuracy of motor bearing fault diagnosis.
2) Combined with the convolutional layer of a convolutional neural network and a long short-term memory network, a motor bearing fault diagnosis model is established. The model reduces the influence of noise in the original signal, and the experimental results show that the model has good robustness.

II. THE CLSTM ALGORITHM

In the CNN, the essence of the convolutional kernel is a set of filters, and the result of the convolution can be regarded as the response of the input to the convolutional kernel function. When the input signal is filtered by the CNN, the frequency change can be reduced [21]. 2D-CNNs and multidimensional CNNs are usually used in image and video processing. One reason for using 1D-CNN is that in the field of bearing fault diagnosis, the results measured by sensors are mostly
one-dimensional time-domain signals collected directly from the data acquisition system. Different from image recognition, each sampling point of the vibration signal represents the amplitude, and the order of points represents the time series. Time series may contain fragments of periodic or short pulse signals, which are usually important state signal features and may not exist in two-dimensional images. Using a two-dimensional or multidimensional CNN, it is necessary to transform the original vibration signal into a two-dimensional time-frequency image and then input it into a two-dimensional convolutional neural network for classification and prediction. This may lead to the weakening or even loss of some important features in the original data. In contrast, two-dimensional input usually has higher dimensions than one-dimensional input, which makes the CNN more complex and consumes more computing resources. Therefore, a one-dimensional convolutional neural network (1D-CNN) is relatively simple to solve these problems. It is more suitable for bearing fault diagnosis [22].

To obtain a model with better feature extraction ability and robustness, we simultaneously use the advantages of CNN and LSTM. The CNN and LSTM are combined into one structure, and an improved fault diagnosis method is proposed in this paper. The combination of CNN and LSTM is used to enhance the feature extraction capability of the model. As shown in Figure 2, the CLSTM algorithm is mainly composed of an input layer, 1D convolution layer, LSTM layer and output layer.

First, the input layer inputs the original vibration signal $X = \{x_1, x_2, x_3, \cdots, x_n\}$ of the motor bearing into the network. Then, through the first convolutional layer, the main function of the convolutional layer is to convolve the input local region with the convolutional kernel to obtain the feature mapping. The convolution layer realizes local area connection and weight sharing. The convolution kernel slides along the direction of width and height in the input region. The weight of the convolutional kernel is multiplied by the value of the corresponding position in the region, and then all the products are added to obtain an output. When the convolutional kernel traverses the whole region, the output feature map $u_i^{(1)}$ is obtained [23], [24]. The parameters of the convolutional layer are shown in Table 1.

![FIGURE 2. CLSTM algorithm structure diagram.](image)

![FIGURE 3. The structure of the LSTM unit.](image)

TABLE 1. Parameters of the convolutional layer in the CLSTM model.

| No. | parameter    | value   |
|-----|--------------|---------|
| 1   | Input size   | 1024×1  |
| 2   | Kernel       | 5×1     |
| 3   | Stride       | 1       |
| 4   | Output size  | 1020×1  |

The forward calculation process of the LSTM layer can be expressed by formulas (2) to (5). The forget gate is a key component of the LSTM unit, which can control which information should be retained and which should be forgotten, and avoid the gradient disappearance and explosion problems.
caused by the backpropagation of errors over time [27].

\[
f_{t}^{(i)} = \sigma(W_{f} \cdot [h_{t-1}^{(i)}, u_{t}^{(i)}] + b_{f}).
\]  

(2)

where \(f_{t}^{(i)}\) is the output value of the forgetting gate, \(\sigma\) is the sigmoid activation function, \(W_{f}\) is the weight matrix of the forget gate, and \(b_{f}\) is the bias matrix of the forget gate.

The input gate is used to control how much of the network’s current input data \(u_{t}^{(i)}\) flows into the memory unit. That is, the effect of the information in the memory unit \(C_{t-1}\) at the previous moment on the current memory unit \(C_{t}\). The calculation formula is as follows:

\[
i_{t}^{(i)} = \sigma(W_{in} \cdot [h_{t-1}^{(i)}, u_{t}^{(i)}] + b_{in}).
\]  

(3)

\[
\hat{C}_{t}^{(i)} = \tanh(W_{C} \cdot [h_{t-1}^{(i)}, u_{t}^{(i)}] + b_{C}).
\]  

(4)

\[
C_{t}^{(i)} = f_{t}^{(i)} \cdot \hat{C}_{t-1}^{(i)} + i_{t}^{(i)} \cdot \hat{C}_{t}^{(i)}.
\]  

(5)

where \(u_{t}^{(i)}\) is the input feature, \(h_{t-1}^{(i)}\) is the hidden feature of the output at the previous moment, \(W_{in}\) and \(W_{C}\) are the weight matrixes of the input gate, \(b_{in}\) and \(b_{C}\) are the bias matrix of the input gate, \(\tanh(\cdot)\) is the activation function, \(i_{t}^{(i)}\) is the output value of the input gate, \(\hat{C}_{t}^{(i)}\) is the new information candidate value, and \(C_{t}^{(i)}\) is the new cell memory unit; the subscript is the time state, and the superscript is the number of network layers.

The output gate controls the influence of the memory unit \(C_{t}\) on the current output value \(h_{t}\). That is, which part of the memory unit will be output at time step \(t\). The value of the output gate is shown in equation (6), and the output \(h_{t}\) of the LSTM unit at time \(t\) can be obtained by equation (7).

\[
\text{out}_{t}^{(i)} = \sigma(W_{out} \cdot [h_{t-1}^{(i)}, u_{t}^{(i)}] + b_{out}).
\]  

(6)

\[
h_{t} = \text{out}_{t} \cdot \tanh(C_{t}).
\]  

(7)

where \(\text{out}_{t}^{(i)}\) is the output value of the output gate, \(W_{out}\) is the weight matrix of the output gate, \(b_{out}\) is the bias matrix of the output gate, and \(h_{t}^{(i)}\) is the output value of the current LSTM unit.

Stack these LSTM units according to the time step and the number of network layers to obtain a complete LSTM network model. After passing through the LSTM network layer by layer, the output feature \(h_{t} = \{h_{1}, h_{2}, h_{3}, \ldots, h_{T}\}\) of the LSTM layer is finally obtained.

Finally, the obtained \(h_{t}\) is fully connected to calculate the information \(p_{t}\) needed by the next output layer. In the output layer, softmax is used as the classifier of the model to classify different types of signals. The formula of the softmax function is as follows:

\[
q(z_{j}) = \frac{e^{z_{j}}}{\sum_{i} e^{z_{i}}},
\]  

(8)

where \(z_{j}\) is the logarithm of the \(j\)-th neuron.

The output of this CLSTM network is:

\[
y_{\text{pred}} = \text{soft max}(W^{d+1} p_{t}).
\]  

(9)

With the network model output, we can obtain the loss value of the model by comparing the predicted output value \(y_{\text{pred}}\) of the model with the actual value \(y\). Then, the parameters of the model are updated through the back propagation process. The loss value is calculated by the cross-entropy loss function, as shown in equation (10). The model loss value represents the error between the predicted value of the neural network and the expected output value, and the accuracy rate represents the consistency between the predicted output value of the neural network and the real result.

\[
\text{loss} = -\frac{1}{n} \sum x [y \ln y_{\text{pred}} + (1 - y) \ln(1 - y_{\text{pred}})].
\]  

(10)

In the training process, each time \(m\) group of data samples is extracted as a batch, and each iteration of a batch of data is used to calculate the training loss. Then, the Adam optimizer [28] is used to adjust the training parameters, and the training loss value and training accuracy are recorded every iteration. During the test, the data of the test set are input into the trained model according to the same batch to calculate the accuracy.

### III. THE BO-CLSTM METHOD

The accuracy of the CLSTM algorithm is very dependent on selecting a suitable set of hyperparameter combinations. However, the hyperparameters influence each other, and it is theoretically difficult to select a suitable hyperparameter combination only by prior experience. This not only requires experienced professionals but also consumes considerable time. The earliest automated parameter adjustment method is the grid search strategy [29], but when there are many hyperparameters in the deep neural network, its efficiency is low. For example, assume that a learning model has \(K\) hyperparameters that need to be adjusted. Among them, the \(j\)th \((j = 1, 2, 3, \ldots, K)\) hyperparameter has \(L^{j}\) choices. The grid search determines that the resulting candidate solutions have

\[
n = \prod_{j=1}^{K} L^{j}.
\]

Therefore, as the number of hyperparameters increases, the number of candidate solutions increases exponentially, which leads to low grid search efficiency. Random search [30] is more efficient than grid search in practical applications, but it is blind, easily misses some important information, and has poor stability.

The Bayesian optimization algorithm (BO) is a hyperparameter optimization algorithm with better performance. It is a hyperparameter search method based on a probability model to achieve evaluation and estimation. Different from grid search and random search, the Bayesian optimization method uses prior information and observed results to define the posterior distribution of the function space. When the form of the objective function is not clear, some features of the objective function are described through a priori information. Thus, the Bayesian optimization method has fewer iterations, is faster, and can optimize hyperparameter values more efficiently [31]. Moreover, the Bayesian optimization algorithm has been proven to be superior to other advanced
For a new sample $x_{t+1}$, the covariance matrix of the Gaussian process is updated due to the addition of the new sample. The covariance matrix is updated as follows:

$$
K = \begin{bmatrix}
    k(x_{t+1}, \cdots, x_1, x_{t+1}) \\
    \vdots & \ddots & \vdots \\
    k(x_{t+1}, x_1) & \cdots & k(x_{t+1}, x_t)
\end{bmatrix}.
$$

The posterior probability distribution of the $t+1$ time can be expressed as:

$$
P(f_{t+1} | D_{1:t}, x_{t+1}) = \mathcal{N}(\mu, \sigma^2).
$$

$$
\mu = k^TK^{-1}y_{1:t},
$$

$$
\sigma^2 = k(x_{t+1}, x_{t+1}) - k^TK^{-1}k.
$$

where $P(f_{t+1} | D_{1:t}, x_{t+1})$ is the posterior probability distribution of the $t+1$ time, $y_{1:t}$ is the actual value of the first $t$ samples, and $y_t = f(x_t) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ has a Gaussian noise.

After the surrogate model is established through the Gaussian process, an acquisition function is selected to determine the next set of hyperparameter combinations from the posterior distribution of the model. To balance the global search ability and local search ability of the algorithm and prevent the algorithm from falling into the local optimum, the Bayesian optimization algorithm needs to find a balance between development and exploration [35]. In this paper, expected improvement (EI) is selected as the collection function, as in the formulas:

$$
\alpha(x) = [\mu(x) - q^+]\Phi(q) + \sigma(x)\phi(q).
$$

$$
x_{t+1} = \arg \max_x \alpha(x).
$$

$$
z = (\mu(x) - p^+)/\sigma(x).
$$

where $\alpha(x)$ represents the utility function, $\mu(x)$ represents the mean value, $\sigma(x)$ is the standard deviation, $q^+$ is the current maximum utility function value, $\Phi(q)$ represents the cumulative probability function of the Gaussian distribution, $\phi(q)$ represents the probability density function of the Gaussian distribution, and $x_{t+1}$ represents the parameter combination evaluated $x+1$ times.

The collection function is used to select the hyperparameter combination from the surrogate function model to be evaluated in the next step, and $(x_{t+1}, y_{t+1})$ is continuously input into the Gaussian process to modify the surrogate function model so that the surrogate model can approach the real distribution of the objective function more quickly and accurately.

The above content introduces the approximate calculation process of the hyperparameter combination selection of the Bayesian optimization CLSTM algorithm. Next, the specific algorithm flow of the Bayesian optimization is summarized in Algorithm 1, where $x$ is the hyperparameter combination and $\alpha(x)$ is the collection function. Bayesian optimization establishes a surrogate model (probabilistic model) based on...
the past evaluation results of the objective function and establishes the relationship between the hyperparameter space and the model performance through the surrogate model. In each iteration, the acquisition function selects the most potential hyperparameters for experimentation based on the feedback of the current surrogate function. A group of real results of the model hyperparameter evaluation is obtained in the experiment, and the results are reinput into the surrogate function for correction. Thus, it is expected that a better combination of hyperparameters with higher efficiency will be found in the iteration process.

Algorithm 1 Algorithm Bayesian Optimization
1: Initialize the model;
2: Randomly select several groups of hyperparameters \( x \), train the model, and obtain the corresponding model evaluation index \( y \). Obtain the initial sampling point set \( D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)\}; \)
3: for \( t = 1, 2, 3, 4, \ldots \) do
4: Use the data in \( D \) to build a proxy function based on the Gaussian distribution;
5: Select an optimal \( x_{t+1} \) by collecting function. \( x_{t+1} = \arg\max \alpha(x) \)
6: Sample the objective function: \( y_{t+1} = f(x_{t+1}) + \epsilon_{t+1} \)
7: Obtain \( D_{t+1} = (x_{t+1}, y_{t+1}) \) and add \( D_{t+1} \) to the data set \( D = D \cup D_{t+1} \)
8: end for

According to the motor bearing fault diagnosis model established by BO-CLSTM proposed in this research, the related processes and methods are summarized as follows:

Step 1: Collect the normal and fault motor signals. Under certain working conditions, a vibration sensor is used to measure the vibration signals of the outer ring, inner ring and balls. The measured vibration signal is divided into a training set and a test set.

Step 2: Optimize the network model. Through a Bayesian optimization algorithm, the Gaussian surrogate model of the hyperparameter combination and evaluation value is established. The learning rate, the number of network layers, the number of hidden layer nodes and the training batch are selected automatically to obtain the optimal network structure.

Step 3: Save the model. The optimized hyperparameter training model is saved for later motor bearing vibration signal fault diagnosis.

Step 4: Fault classification. According to the diagnostic model trained in the previous step, the motor bearing data are input to the diagnostic model, and the CLSTM network is used to obtain the fault features to classify and identify the fault.

IV. EXPERIMENTAL ANALYSIS
A. DATASET INTRODUCTION
The experimental data are from the motor-driven mechanical system standard acceleration dataset from the Case Western Reserve University (CWRU) bearing data center (Fig. 5). The experimental platform consists of a 1.5 kW motor, a torque sensor, a power meter and an electronic controller composition. This experiment uses the data of the drive end bearing, the bearing model is 6205-2RS JEM SKF, the motor speed is 1,772 r/min, the load is 1 hp, and the sampling frequency is 12 kHz. The vibration signal was collected by a 16-channel data recorder [36]. The bearing failure dataset is shown in Table 3, including normal conditions and 3 types of failures: inner ring failure, outer ring failure and ball failure. Each type of failure includes damage degrees of 0.007 inches, 0.014 inches and 0.021 inches, and there are 10 states in total. The partial time-domain waveforms of the vibration signals of the 10 bearing states are shown in Figure 6. It can be seen in the figure that the time-domain diagram of the normal signal is in a state of uniform fluctuation. However, in other fault states, some irregular wave states and shock features appear in the time-domain signals that are quite different from normal signals. Therefore, it is very feasible to extract the time-domain features of the vibration signals by a deep learning algorithm. Next, the original vibration signal is divided into 10,000 data samples, a total of 10 states, 1,000 for each type, and the length of each data sample is 1,024. The dataset is divided into a training set and a test set at a ratio of 7:3; the training set contains 7,000 samples, and the test set contains 3,000 samples.

B. EXPERIMENTAL VERIFICATION AND ANALYSIS
The BO-CLSTM model proposed in this research is implemented by programming in the open source TensorFlow.
All the tested algorithms and models are implemented in the Python 3.6.12 development environment of PyCharm.

In the above model, we use the Bayesian optimization method to determine the hyperparameter values of the CLSTM algorithm. In the experiment, the learning rate is 0.05, the number of network layers is 1, the number of time steps is 20, the number of hidden layer nodes is 102, the training batch is 123 as the initial hyperparameter combination, and the number of iterations is set to 60. To facilitate the analysis, through the visualization method, the process of Bayesian optimization is more intuitively displayed. The results are shown in Figure 7. In the figure, the abscissa is the number of iterations, and the ordinate is the highest accuracy that the network model can achieve after n iterations. It can be seen in the figure that under the initial hyperparameter combination, the accuracy of CLSTM is less than 92%. Then, a new set of hyperparameter combinations is selected through the Bayesian optimizer. Under this set of hyperparameter combinations, the accuracy of the CLSTM model reaches 99.5%. Finally, the ideal hyperparameter combination is obtained when the hyperparameters are updated to the 50th iteration so that the model is optimized, and the accuracy rate is 100%. More details of Bayesian optimization are shown in Figure 8. In Figure 8, we show a visual two-dimensional graph for each optimized hyperparameter. The yellow area shows better performance, the blue area shows poor performance, the black point is the hyperparameter sampling position of the optimization algorithm, and the red asterisk is the best parameter position found. Figure 8 shows that the learning rate is 0.0008690938486803831, the number of LSTM layers is 4, the number of hidden layer nodes is 371, the number of time steps is 20, the input dimension is 51, and the training batch is 10, which makes the diagnosis accuracy optimal. Since the Bayesian optimizer establishes an alternative hyperparameter search space model and searches in this dimension instead of searching in the original actual hyperparameter space, the Bayesian optimization is faster. Then, the most accurate model is saved for subsequent bearing fault diagnosis performance testing. Table 4 shows the information of each layer of the optimized model, including the output dimensions of each data layer and the number of parameters.

In the CLSTM training process, after each traversal of the training set during training, a test is performed on the test set, and the loss and accuracy of the model in the training set and the test set are recorded. The training loss, training accuracy, test loss, and test accuracy of the optimized CLSTM model are shown in Figure 9. Figure 9(a) shows the learning curve between the training accuracy and test accuracy and the number of iterations. The abscissa is the number of iterations, and the ordinate is the diagnostic accuracy of the model. Figure 9(b) is the learning curve between the model loss value and the number of iterations, the abscissa is the number of iterations, and the ordinate is the model loss on the data set. It can be seen in the figure that the model converges quickly during training. After 27 iterations, the loss value eventually approaches 0, and the recognition accuracy reaches 100%. On the test set, the model’s recognition accuracy is initially only 60%. As the training continues, the recognition accuracy rate increases rapidly and then gradually stabilizes and finally reaches 100% accuracy on the test set.

To analyze the effect of the BO-CLSTM model more intuitively, a confusion matrix is used to compare the
predicted results with the actual failure types. In Figure 10, the abscissa is the predicted motor state, and the ordinate is the various state labels of the real motor. There are 3,000 motor states in the confusion matrix. The predicted values of each motor state correspond to the true values. All faults can be correctly diagnosed without misdiagnosis. The precision, recall and F1-score of each fault are all 1.00. In Figure 11, the t-SNE algorithm is used to reduce the dimensions of the hidden layer features automatically extracted from the BO-CLSTM model to two-dimensional space, and a scatter diagram is drawn to show the low-dimensional mapping results. It can be seen in the figure that the features extracted by the BO-CLSTM model have good clustering properties, and the cluster clusters in various operating states are relatively isolated, which can clearly distinguish these ten different operating states. The above results fully prove that the BO-CLSTM model has high prediction accuracy, and the convergence speed is very fast, which can diagnose motor faults more effectively. By Bayesian optimization of hyperparameters, the model achieves higher accuracy and simultaneously saves the considerable work-load required for parameter tuning.

To evaluate the performance of the proposed model, some deep learning models are compared with the proposed model under the same data. Table 5 lists the parameter information of these methods and their accuracy on the dataset.
TABLE 5. Comparison of bearing fault diagnosis results between different deep learning models.

| Model          | BPNN | SAE   | CNN       | LSTM       | BO-CLSTM   |
|---------------|------|-------|-----------|------------|------------|
| Input layer   | 1024×1 | 1024  | 1024×1×1  | 32×32      | 1024×1×1   |
| Hidden-1      | Dense(512) | Encoder(256) | Conv1d (5 × 1, 1) | LSTMCell (128) | Conv1d (5×1) |
| Hidden-2      | Dense(512) | Encoder(128) | MaxPooling2D(2×1) | LSTMCell (128) | LSTMCell (128) |
| Hidden-3      | Dense(512) | Encoder(64) | Conv1d (5×1, 1) | Dense(10)   | Dense(10)   |
| Hidden-4      | Dense(0) | Encoder(10) | MaxPooling2D(2×1) | 97.20%      | 99.87%      |
| Hidden-5      |       |       |           |            | 100%        |
| Hidden-6      |       |       |           |            |             |
| Hidden-7      |       |       |           |            |             |
| Hidden-8      |       |       |           |            |             |
| Hidden-9      |       |       |           |            |             |
| Hidden-10     |       |       |           |            |             |
| Accuracy      | 97.49% | 98.63% | 99.87%    | 97.20%     | 100%        |

FIGURE 10. Confusion matrix.

FIGURE 11. Low-dimensional mapping results of the BO-CLSTM model.

Classical deep learning models include long short-term memory networks (LSTMs), CNNs, tacked autoencoders (SAEs), and backpropagation neural networks (BPNNs). It can be seen in the table that the accuracy of LSTM on the test set is relatively high, and the final accuracy reaches 99.87%. The accuracy of SAE reached 98.63%, and the accuracies of BPNN and CNN on the test set are 97.49% and 97.20%, respectively. Compared with other methods, the BO-CLSTM algorithm proposed in this paper significantly improves accuracy. It realizes fault diagnosis without misjudgment. This fully proves the superiority of BO-CLSTM in motor bearing fault diagnosis.

Since the actual working environment is often accompanied by noise interference, it is very meaningful to improve the robustness of the diagnostic model and effectively suppress noise. In the experiment, Gaussian white noise signals with different signal-to-noise ratios are artificially added to the vibration signal to simulate the real working noise environment. The white Gaussian noise is mixed with the original time-domain waveform of the vibration signal to form a signal with a signal-to-noise ratio between 2 and 10 dB. After training with training samples without noise signals, 3,000 test samples containing 2-10 dB Gaussian white noise signals are imported into the trained BO-CLSTM model for testing and compared with other deep learning models. The histogram comparison result is shown in Figure 12.

The results show that after training on the same training set, all deep learning methods can achieve a classification accuracy of more than 94% on 3,000 test samples with a signal-to-noise ratio of more than 6 dB. When the signal-to-noise ratio is reduced, the classification accuracy of each model is significantly reduced. When the signal-to-noise ratio of the test set drops from 10 dB to 2 dB, the accuracy of the CNN model decreases the most, by 39.51%. The accuracy of the SAE model decreases by 14.18%, and the accuracy of the LSTM model also decreases by 10.03%. In contrast, the accuracy rates of BO-CLSTM and BPNN decrease by 7.19% and 5.95%, respectively, and have good robustness. However, from the point of view of the speed of decline, the accuracy of BO-CLSTM only decreases by 0.07% before...
In summary, the BO-CLSTM model exhibits high performance in a noisy environment, especially in the case of a small signal-to-noise ratio and can still achieve high classification accuracy. In addition, the BPNN model and the LSTM model have better antinoise performance, but there is a gap between the BO-CLSTM model and the best performance. From the above experimental results, it can be seen that the BO-CLSTM model proposed in this paper has higher accuracy and stronger robustness.

V. CONCLUSION

In this paper, the convolutional long short-term memory network based on Bayesian optimization is applied to motor bearing fault diagnosis. This method makes up for the shortcomings of traditional fault diagnosis methods, uses a Bayesian optimization algorithm to optimize the hyperparameters, automatically selects the network structure of the diagnosis model, and reduces the process of artificially adjusting the hyperparameters. The method of combining a convolution layer with a long short-term memory network is used to improve the feature extraction ability and robustness of the algorithm. The proposed model can reach 100% accuracy on the public CWRU standard bearing dataset. Experiments in different noise environments verified the antinoise ability of this method. The experimental results show that the method has good performance. Compared with traditional methods, this method has higher accuracy and stronger robustness, realizes real end-to-end fault diagnosis, minimizes human intervention in feature learning, improves the efficiency and results of motor diagnosis, and makes motor bearing fault diagnosis more intelligent. It is of great significance to apply deep learning to the field of motor fault diagnosis, which can easily and quickly predict motor faults and greatly improve the efficiency and safety of industrial production. This paper mainly studies motor bearing faults. In the future, other motor parts still have room for consideration and evaluation, and great progress can be considered to be made in the future.
[23] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[24] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[25] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, “A recurrent neural network based health indicator for remaining useful life prediction of bearings,” *Neurocomputing*, vol. 240, pp. 98–109, May 2017.

[26] J. Collins, J. Sohl-Dickstein, and D. Sussillo, “Capacity and trainability in recurrent neural networks,” 2016, arXiv:1611.09913. [Online]. Available: https://arxiv.org/abs/1611.09913

[27] J. Lei, C. Liu, and D. Jiang, “Fault diagnosis of wind turbine based on long short-term memory networks,” *Renew. Energy*, vol. 133, pp. 422–432, Apr. 2019.

[28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: https://arxiv.org/abs/1412.6980

[29] Y. Bao and Z. Liu, “A fast grid search method in support vector regression forecasting time series,” in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, Berlin, Germany, 2006, pp. 504–511.

[30] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, Feb. 2012.

[31] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the human out of the loop: A review of Bayesian optimization,” *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.

[32] M. A. Gelbart, J. Snoek, and R. P. Adams, “Bayesian optimization with unknown constraints,” 2014, arXiv:1403.5607. [Online]. Available: https://arxiv.org/abs/1403.5607

[33] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams, “Scalable Bayesian optimization using deep neural networks,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 2171–2180.

[34] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, “Multi-task Gaussian process prediction,” *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2008, pp. 153–160.

[35] T. Desautels, A. Krause, and J. Burdick, “Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 827–834, 2012.

[36] H. Shao, H. Jiang, F. Wang, and Y. Wang, “Rolling bearing fault diagnosis using adaptive deep belief network with dual-tree complex wavelet packet,” *ISA Trans.*, vol. 69, pp. 187–201, Jul. 2017.

ZHEN LI received the B.E. degree from the Department of Electrical Engineering, Wanfang College of Science and Technology, Henan University of Technology, Jiaozuo, China, in 2019. He is currently pursuing the master’s degree in electrical engineering with Shanghai Dianji University, Shanghai. His research interests include data-driven fault detection and diagnosis, data mining and analysis, and deep learning.

YANG WANG received the B.E. degree from the Department of Automation, Shenyang University of Technology, Shenyang, China, in 2006, the M.E. degree in measurement and control technology and instrument from Shanghai Maritime University, Shanghai, China, in 2008, and the Ph.D. degree in control science and engineering from Shanghai University, Shanghai, in 2017. She is currently a Lecturer with Shanghai Dianji University, Shanghai. Her research interests include data-driven fault detection and diagnosis, process monitoring, and wireless sensor networks.

JIANENG MA received the B.E. degree from the Department of Mechanical Engineering, Liren College of Yanshan University, Qinhuangdao, in 2019. He is currently pursuing the master’s degree in electrical engineering with Shanghai Dianji University, Shanghai. His research interest includes the prediction of the remaining service life of motor bearings.