

Research Article Convolutional Recurrent Neural Network for Fault Diagnosis of High-Speed Train Bogie

Kaiwei Liang, Na Qin, Deqing Huang , and Yuanzhe Fu

Institute of Systems Science and Technology, School of Electrical Engineering, Southwest Jiaotong University, Chengdu 611756, China

Correspondence should be addressed to Deqing Huang; elehd@home.swjtu.edu.cn

Received 10 July 2018; Revised 9 October 2018; Accepted 17 October 2018; Published 23 October 2018

Guest Editor: Liang Hu

Copyright © 2018 Kaiwei Liang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Timely detection and efficient recognition of fault are challenging for the bogie of high-speed train (HST), owing to the fact that different types of fault signals have similar characteristics in the same frequency range. Notice that convolutional neural networks (CNNs) are powerful in extracting high-level local features and that recurrent neural networks (RNNs) are capable of learning long-term context dependencies in vibration signals. In this paper, by combining CNN and RNN, a so-called convolutional recurrent neural network (CRNN) is proposed to diagnose various faults of the HST bogie, where the capabilities of CNN and RNN are inherited simultaneously. Within the novel architecture, the proposed CRNN first filters out the features from the original data through convolutional layers. Then, four recurrent layers with simple recurrent cell are used to model the context information in the extracted features. By comparing the performance of the presented CRNN with CNN, RNN, and ensemble learning, experimental results show that CRNN achieves not only the best performance with accuracy of 97.8% but also the least time spent in training model.

1. Introduction

As a prevalent and economical means of transportation, the development of high-speed train (HST) has been an interest of many countries, especially in China. Meanwhile, with the increasing of train speed and the application of lightweight design, it is crucial to ensure the safe operation and ride stability of HST. Since it becomes an accepted practice that the HST must fail safe, the fault diagnosis of HST has attracted a surging amount of attention. When the fault occurs, train safety monitoring device might issue an alarm signal, which ensures that the fault would not be developed to a serious failure. Nevertheless, certain key components of train are still not effectively monitored, such as bogie. According to [1], the main factors influencing the stable running of HST are closely related to the performance state of bogie. The effective diagnosis and identification of the fault conditions of bogie have become a focus of early warning and health maintenance of HST

Bogie is one of the most important components in the structure of railway vehicles. Since the structure of bogie is designed to facilitate the installation of springs and dampers, bogie exerts a good performance in vibration damping. Train can be reliably supported on railway track by bogie that includes the wheelsets and suspension elements, as shown in Figure 1. To be more specific, the suspension elements of bogie consist of air springs, lateral dampers, antiyaw dampers, and so on. Due to track irregularities, the train has to experience irregular vibration. The bogie can mitigate the interaction between vehicles and rails to effectively reduce vibration. However, the abnormal vibration caused by bogie failure may result in poor ride comfort and even side rollover.

The mechanism of bogie failure is so complicated and the signal features are so obscure to master the fault laws of bogie. Compared with traditional signal processing methods, deep learning methods can adaptively extract fault features and achieve intelligent diagnosis. This paper will employ a deep learning method called convolution recurrent neural network (CRNN) to identify various faults of HST bogie.

1.1. Related Work. Many researches have been conducted on the fault diagnosis of HST bogie, where the main ideas include signal-processing-based methods, machine learning, and deep learning.



FIGURE 1: Structure of the bogie.

The signal-processing-based methods are generally used to analyze the collected signals and extract the timefrequency-domain features that are most relevant to the fault information. In [2], the features of the bogie acceleration signal are analyzed in the time and frequency domains so as to diagnose faults. In [3], the combination of power spectrum and principal component analysis is proposed to extract signal characteristics for fault diagnosis. The traditional spectral analysis method is based on Fourier transform, and suitable for the feature extraction of stationary signals. However, the vibration signal of the actual system is the nonstationary signal, so modern time-frequency analysis methods represented by wavelet transform [4-6], empirical mode decomposition (EMD) [7-9], and Hibert-Huang transform [10-12] are widely used in the fault diagnosis. The ensemble empirical mode decomposition (EEMD) method presented by Wu and Huang [13] is the improvement of the EMD method and can decompose the signal into several intrinsic mode functions (IMF), which reflects the timefrequency characteristics of the signal. With EEMD applied to fault diagnosis of bogie, [14] studied the relationship between fault types and energy moment features in each IMF of bogie signals.

Traditional pattern recognition only focuses on the classification stage. Feature extraction is considered as an independent problem, which is mainly based on manual methods such as modern signal processing and expert knowledge. In contrast, feature extraction and classification are simultaneously trained in deep learning [15]. In addition, deep learning is more suitable for big data analysis than modern signal processing methods. Therefore, deep learning has become the technology of choice for fault diagnosis in recent years.

At present, the most studied and applied models of deep learning systems are convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The biggest advantage of CNN is feature extraction. Since CNN can utilize the accumulated experience in the training process to select features, it avoids relying on artificial processes to extract features. Furthermore, as the neuron weights on the same convolution kernel are identical, the network can improve training speed through parallel learning. Therefore, convolutional networks have the advantages of faster training speed and higher feature extraction efficiency than traditional deep neural networks. 1D-CNN [16–18], a CNN that uses one-dimensional filters to process time series, has achieved success in natural language processing tasks and voice recognition.

Similar to CNN, RNN is another well-performing neural network, which is also used in many natural language processing tasks. The Long Short-Term Memory (LSTM) [19, 20] is a variation of the RNN. In contrast to standard RNN, LSTM is amenable to overcome the problem of long-term dependencies. Bruin [21] put forward to apply LSTM neural network to fault identification of track circuits. However, the dependencies of time series make it difficult to use LSTM for parallel computation. The calculation speed is far less than that of the CNN. Reference [22] proposed the Simple Recurrent Unit (SRU) model based on the study of the RNN models. Under the premise of guaranteeing the speed, the accuracy of the recognition task using the SRU model was hardly affected.

To benefit from both CNN and RNN, the two approaches can be integrated into a combined network, which has several convolutional layers followed by multiple recurrent layers. Ullah [23] combined CNN and a Deep Bidirectional LSTM network into a kind of CRNN, which is adopted to process video data for action recognition. Also, Lopez [24] put forward a CRNN method, which incorporated twodimension convolutional neural network into LSTM and was used to classify the network traffic. Similar methods had been applied to the classification mask in audio signals [25, 26] and electrocardiography signals [27].

This paper presents a joint neural network CRNN that integrates 1D-CNN and SRU. For recognizing the bogie vibration signals, the proposed CRNN has the advantages of 1D-CNN and SRU respectively:

(1) The convolutional layers of the presented CRNN have the same properties as 1D-CNN, which can detect hidden features directly from the time series data effectively and does not depend on manual selection.

(2) The recurrent layer of the recommended CRNN has the same advantages as RNN, and it has the ability to mine timing related information.

(3) Compared with the LSTM, the proposed CRNN containing SRU cell can better improve the training speed with the effective recognition accuracy.

The 1D-CNN part of the presented CRNN extracts the depth characteristics of the bogie signals. The stacked SRU section learns the sequence information of the signal frame in each layer of forward delivery. Therefore, the proposed method can quickly identify bogie sequence information to ensure the real-time and accuracy requirements of diagnosis. These advantages make the presented method more suitable for the fault diagnosis of bogie.

The rest of this paper is organized as follows. Section 2 analyzes bogie signals. The recommended CRNN structure is explained in Section 3. Section 4 discusses experimental results, including evaluation indicators and comparisons with the other state-of-the-art methods. Section 5 concludes the paper and summarizes the potential future works.

2. Analysis of Signals for Bogie

2.1. Acquisition of Original Signal. For specific diagnostic objects, it is necessary to select the appropriate diagnostic method by analyzing the characteristics of the signals. For this reason, the bogie vibration signals need to be collected and analyzed before realizing fault diagnosis of the bogie. It will be at highly risk to install the bogie with faults on real operating HST. In order to simulate the characteristics of faulty bogie in actual operation, the multibody dynamic model of bogie is built by SIMPACK software [28, 29].

In this paper, the data for analysis are collected from SIMPACK. The Simulation Data Set (SDS) consists of seven

TABLE 1: Seven kinds of health conditions of the bogie.

Condition Label	Health conditions
0	normal condition
1	failure of the air springs
2	failure of the anti-yaw damper
3	failure of the lateral damper
4	failure of the air spring + the antiyaw damper
5	failure of the air spring + the lateral damper
6	failure of the lateral damper + the antiyaw damper



FIGURE 2: Suspension system and multibody dynamic model of the HST. (a) Suspension system and (b) multibody dynamic model.

subsets, which are relevant to seven different conditions showed in Table 1. The SDS is essentially a mechanical vibration signal, which consists of the acceleration and relative displacement of the main components of the bogie. The connection of the main components of the simulation model is shown in Figure 2(a). As can be seen from this figure, the bogie is connected to the vehicle body by the secondary suspension, while the wheelset is also connected to the bogie by the primary suspension. Figure 2(b) shows the simulation model of HST, which uses LMA treads and CN60 tracks. The Wuhan-Guangzhou track spectrum [30] is adopted as the excitation of simulation. Based on this model, the nonlinear relationship of wheel-rail contact and suspension can be fully considered.

TABLE 2: The description of parameters of simulation.

Parameter	Value
Sampling frequency	243Hz
Track spectrum	Wuhan-Guangzhou track spectrum
Vehicle speed	200 km/h
Sensor channels	58

There are 58 sensor channels installed on the HST to monitor the bogie operation status. Table 2 lists the simulation parameters in the experiments. Each subset of SDS indicates the vibration signal with a specific health condition, containing 51030 sampling points and 58 channels. In each sample, the extracted signals contain 486 sample points of the simulation data. Since the sampling frequency is 243 Hz, each sample records a signal with a length of 2 seconds. Hence, there are totally 105 samples for each health condition.

2.2. Frequency-Domain Analysis of Signal

2.2.1. Fast Fourier Transform. The traditional fast Fourier transform (FFT) is a method of analyzing signals, which can convert time-domain signals into frequency-domain signals [31–33], thereby providing more frequency-domain information. As displayed in Figure 3, the time-domain vibration signals of bogie are transformed into frequency-domain signals by FFT. By comparing Figure 3(a) with Figure 3(b), it can be seen that they are extremely similar in frequency domain. Comparing Figures 3(c) with 3(e), they both have relatively prominent frequency components during the frequency band of 0-5 Hz. In Figures 3(d) and 3(f), there are frequency multiplication characteristics in the frequency band of 0-20 Hz. Further, in Figure 3(g), the peak points are 25 Hz and 36 Hz.

2.2.2. A Method Combining EEMD and Autoregressive Spectrum Analysis. FFT method could cause spectral aliasing, spectral leakage, and barrier effects. To overcome the deficiencies of FFT method, a method combining EEMD and Autoregressive (AR) spectrum analysis is proposed to analyze bogie signals. The AR model is the widely used mathematical model in the time series analysis [34–37], and it has the characteristics of accurate frequency location, which can reflect the peak information in the power spectrum. EEMD decomposes the complex unsteady vibration signals into several single-component signals with a mean value of zero and a local symmetry with respect to the time axis. The EEMD method is equivalent to the smoothing of the original signal. Therefore, more effective analysis results could be achieved by applying this combined method.

First, EEMD is applied to deal with the vibration signals of bogie, and the 12 IMF components (from IMF0 to IMF11) and a residual component can be sequentially obtained. The first 8 components after decomposition contain the most significant information of the original signals. Accordingly, the AR spectrum analysis based on Burg algorithm is performed on the first 8 IMF components. The peak positions of EEMD-AR power spectrum can be clearly observed in Figure 4.



FIGURE 3: Comparison of frequency-domain analysis of bogie signals under 7 health condition: (a) normal condition; (b) failure of the air springs; (c) failure of the antiyaw damper; (d) failure of the lateral damper; (e) failure of the air spring + the lateral damper; (g) failure of the lateral damper + the antiyaw damper.



FIGURE 4: Comparison of EMD-AR spectra of the first 8 IMF components of bogie signals in 7 health conditions: (0) normal condition; (1) failure of the air springs; (2) failure of the antiyaw damper; (3) failure of the lateral damper; (4) failure of the air spring + the lateral damper; (5) failure of the air spring + the lateral damper; (6) failure of the lateral damper + the antiyaw damper.

6

Under different operating conditions of HST bogie, the center frequency of each IMF component gradually decreases. In addition, the components from IMF3 to IMF5 reflect the signal characteristics under different operating conditions. From the power spectrum of these components as shown in Figure 4, the bogie generates resonance with the track spectrum mainly containing low-frequency orbit excitation of 5-25 Hz, and the maximum amplitude appears in the corresponding frequency band. However, EEMD-AR spectral method is a fault diagnosis method based on signal process-ing technology, and it depends on manual processes in feature selection and extraction. Hence, it is difficult to automatically acquire and analyze the deep features of HST bogie based on the EEMD-AR spectrum method.

3. Fault Diagnosis of the Bogie Based on CRNN

Since the fault mechanism of bogie is complicated and the features of signal are not evident, the signal processing method cannot extract the signal features effectively and timely. Hence, CRNN was used as the model for the fault diagnosis of bogie. As depicted in Figure 5, the framework contains five parts:

(1) SDS, considered as the input, is fed to a one-dimension convolutional block, which is composed of $l_c \in \mathbb{N}$ alternating one-dimension convolutional layers and one-dimension pooling layers;

(2) the feature maps output by the last convolutional layer are unstacked over the time axis;

(3) the feature maps of the unstack layer are passed to $l_r \in \mathbb{N}$ recurrent layers;

(4) $l_f \in \mathbb{N}$ fully connected layers with tanh activation functions receive the outputs of the last recurrent layer and encode them to retain the useful information;

(5) output layer with softmax function estimates the prediction probabilities of the sample for each class.

3.1. Convolutional Layers. CNN has the characteristics of sparse weights, which can detect small and meaningful features by using convolutional filters that are much smaller in size than the input. This means that CNN reduces the number of parameters that need to be stored and significantly improves the efficiency of feature extraction. The convolutional layer of CNN generally consists of two parts: (1) the first part performs the convolution operation to extract features; (2) the second part performs the pooling operation to adjust the output of convolutional layer.

In CRNN, the convolutional layer function is regarded as a feature extractor. The bogie vibration signals under each health condition are passed as inputs to the CNN layer with one-dimension convolution filters. The feature maps x_j^l are obtained through convolution operation for *l*-th convolutional layer ($l \in l_c$), which is elaborated below:

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{1}$$



FIGURE 5: Framework of the proposed CRNN.

where k_{ij}^l and b_j^l represent the weight and bias of *j*-th convolutional filter, respectively, and M_j is the number of input feature maps.

The pooling process follows the convolution process, which plays a role of secondary extraction. In this paper, max



FIGURE 6: The structure diagram of the LSTM cell: σ_f , σ_i , and σ_o are the logistic sigmoid activation function of forget gates, input gates, and output gates. x_t , h_{t-1} , c_{t-1} are the inputs of the LSTM cell; h_t and c_t represent the outputs of the LSTM cell.

pooling method is adopted to reduce the dimension of data and to preserve useful information as

$$\widehat{x}_{j}^{l} = f\left(\beta_{j}^{l}down\left(x_{j}^{l}\right) + b_{j}^{l}\right)$$

$$\tag{2}$$

where β_j^l and b_j^l represent the weight and bias of max pooling; down() means the max pooling function. After performing operation in l_c layers, the output of CNN block is a tensor with deep features, which contains the most effective information in a small dimension.

3.2. Recurrent Layers. RNN is the neural network for modelling sequential data. At the current time t the network learns the lossy refinement h_t from relevant information of the past sequences $(x_t, x_{t-1}, ..., x_1)$. In this way, RNN can adaptively model information captured from the past sequences. Here, the RNN block of the CRNN uses the SRU cell, which has faster computational capability than the LSTM. The structural characteristics of the SRU and LSTM are compared below.

3.2.1. LSTM Cell. Standard RNN architecture performs poor when using long-term information to process tasks. As the length between the relevant information increases, the ability of the RNN to concatenate related information becomes weaker. LSTM is an advanced RNN architecture aiming to handle long-term dependencies. Compared with standard RNN, LSTM adds a new state referred to cell state, which is used to preserve long-term information. The LSTM controls the cell state through the structure called "gate", which can store the information in the cell state. The architecture of LSTM cell is displayed in Figure 6. The LSTM cell outputs a hidden vector h_t and a cell state vector c_t at each time step. More specifically, the computation of h_t and c_t at the time step t can be explained as follows:

$$f_t = \sigma_f \left(W_f \cdot [h_{t-1}, x_t] \right) + b_f \tag{3}$$

$$i_t = \sigma_i \left(W_i \cdot \left[h_{t-1}, x_t \right] \right) + b_i \tag{4}$$



FIGURE 7: The structure diagram of SRU cell: σ_f and σ_r are the logistic sigmoid activation function of forget gates and reset gates. x_t and c_{t-1} mean the inputs of the SRU cell; h_t and c_t represent the outputs of the SRU cell.

$$o_t = \sigma_o \left(W_o \cdot \left[h_{t-1}, x_t \right] \right) + b_o \tag{5}$$

$$\widetilde{c} = \tanh\left(W_c \cdot [h_{t-1}, x_t]\right) + b_c \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c} \tag{7}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{8}$$

where σ_f , σ_i , σ_o , W_f , W_i , W_o , W_c , and b_f , b_i , b_o , b_c , respectively, represent the logistic sigmoid activation functions, weight matrices, and bias vectors of forget gates, input gates, output gates, and cell states. \odot means element-wise multiplication.

3.2.2. SRU. Time series tasks such as machine translation and speech recognition all rely on the RNN model. However, the sequence dependency of RNN makes it difficult to parallelize computations, so its computational speed is not as fast as CNN. In addition, as the scale of the deployment model enlarges, the real-time nature of the model will also be seriously affected. Reference [22] proposed the SRU model based on the research of LSTM and other models. Under the premise of guaranteeing speed, there is not much loss in accuracy. The architecture of SRU cell is displayed in Figure 7. More concretely, SRU cell simplifies the calculation of the LSTM cell, parallelizing the calculation process. In [22], the computation of SRU cell at the time step t is defined as follows:

$$\widetilde{x}_t = W \cdot x_t \tag{9}$$

$$f_t = \sigma_f \left(W_f \cdot x_t \right) + b_f \tag{10}$$

$$r_t = \sigma_r \left(W_r \cdot x_t \right) + b_r \tag{11}$$

$$c_t = f_t \odot c_{t-1} + (1 - f_t) \odot \widetilde{x}_t \tag{12}$$

$$h_t = r_t \odot \tanh(c_t) + (1 - r_t) \odot x_t \tag{13}$$

where σ_f , σ_r are the logistic sigmoid activation functions of forget gate f_t , reset gate r_t . W, W_f , W_r , respectively, represent

TABLE 3: Result of the CRNN models with different layer in the structure (%).

No	The Structure of CRNN		A (0/)	Itti
	CNN layers (cells)	RNN layers (cells)	Accuracy (%)	iterations
1	1 (116)	1 (450)	88.2%	1620
2	1 (116)	2 (450*2)	93.6%	1200
3	2 (116/232)	1 (450)	95.6%	920
4	2 (116/232)	2 (450*2)	96.3%	940
5	2 (116/232)	4 (450*4)	97.4%	800
6	2 (116/232)	5 (450*5)	96.5%	1020

weight matrices of linear transformation \tilde{x}_t , forget gate, and reset gate. b_f and b_r , respectively, represent bias vectors of forget gate and reset gate.

Combining equations (3)-(8), it can be figured out that the output h_t of LSTM cell at the time step t depends on the h_{t-1} from the previous step t - 1. However, the main design principle of SRU is that the gate calculation only depends on the current input x_t . For input x_t of SRU cell, (9)-(11) can be calculated in parallel, so linear transformation \tilde{x}_t , forget gate f_t , and reset gate r_t are amenable to parallelize calculation in the computer. Moreover, as shown in (7)-(8), the element-wise multiplication is adopted to update the cell state c_t , which depends on the calculation of previous step. The matrix multiplications in SRU take up less computing resources and time. Due to the independent architecture of SRU, SRU can be trained as fast as CNN.

In CRNN, recurrent layers with SRU cell are used to learn the extracted features. After staking the feature maps output by convolutional layers, the output of staking layer is transmitted to SRU as frames. Hence, the RNN block of the presented CRNN is capable of mining the context information h_t , which is used as the inputs of the fully connected layers.

3.3. Fully Connected Layer. In this paper, the RNN block of the presented CRNN is followed by a fully connected layer with a hyperbolic tangent activation function, which acts as the output layer of CRNN mapping the hidden features $h_t^{l_r+l_c}$ learned from the stack layers of CNN and RNN to the tag space $h_t^{l_r+l_c+l_f}$ of the sample as

$$h_t^{l_r+l_c+l_f} = \tanh\left(W \cdot h_t^{l_r+l_c} + b\right). \tag{14}$$

In softmax layer, the softmax function is adopted to turn $h_t^{l_r+l_c+l_f}$ into probabilities a_i for each class. The process of training the neural network is the process to optimize the cost function. To minimize the value of cross-entropy cost function, the weights and biases of convolutional, recurrent, and fully connected layers are iteratively updated by back-propagation method. The computation of cross-entropy cost function in softmax layer is elaborated below.

$$C = -\frac{1}{n} \sum_{i=1}^{n} y_i \ln a_i$$
 (15)

where a_i means the probability distribution of prediction for each category (i = 1, 2, ..., n) after performing

softmax operation and y_i is the true class distribution of sample.

CNN and RNN can be considered as two special cases of the CRNN in this paper: (i) CNN is equivalent to CRNN with multiple convolutional layers and zero recurrent layers; (ii) RNN is composed of CRNN with zero convolutional layers and several recurrent layers. In order to evaluate the effectiveness of using CRNN for fault diagnosis of the bogie, in Section 4, we conduct comparison experiments on different structures such as LSTM, 1D-CNN, ensemble learning (i.e., random forest (RF), Gradient Boost Decision Tree (GBDT), and XGBoost and CRNN.

4. Experimental Results

For the sake of evaluating the proposed method, different methods are compared based on the same HST datasets and experimental environment. The experiments are all performed on Tensorflow and a desktop machine with Inter Core i5-7400 Processor. The advantages of the CRNN structure are further explained by analyzing the recognition accuracy rate and time-consuming situation of different methods.

4.1. Setting. The acquisition of HST simulation data is described in Section 2. In order to avoid the occasionality of the experiment, the experimental data is randomly sorted. The data is divided into 735 samples. Each sample contains 2 seconds vibration signals. From the samples created, 80% are for training and 20% are adopted for testing. The recommended method is tested on test datasets.

4.1.1. CRNN Structure Testing Result. The results of the proposed CRNN model with different layers are presented in Table 3. As can be seen from the results, the fifth CRNN model that contains 2 ID-CNN layers and 4 SRU layers achieves the best result. The first five experiments indicated that the fitting ability of the model will be gradually increased as the number of neural network layers expands. However, the experimental data contains noise. If the model overfits the data, the recognition accuracy of the model will be degraded, which confirms that the last experiment model is overfitting. Therefore, the CRNN structure that consists of 2 ID-CNN layers and 4 SRU layers is adopted.

4.1.2. Different Architecture Testing Result. We have compared the proposed CRNN with the state-of-the-art methods. The

TABLE 4: Hyperparameters used for comparison experiments.

(a)	The architectures	based on	CNN	RNN.	and	CRNN
	(u)	The arenitectures	Dasca on	OLALA	, 11111,	ana	OIGUIA

		HST SDS200	
	CRNN	1D-CNN	LSTM
Conv_1d cells	116/232	116/232	-
pool_1d size	3	3	-
RNN cells	450/450/450/450	-	450/450/450/450
l_c 1D-CNN layers	2	2	-
<i>l_r</i> RNN layers	4	-	4
l_f Fully-connected layers	1	1	1
learning rate	0.005	0.0003	0.001
	(b) The architectures based on e	nsemble learning	

	RF	XGboost	GBDT
estimators size	1000	1000	1000
learning rate	-	0.15	0.01
max depth	500	3	8
max features	sqrt	-	sqrt

hyperparameters used in the evaluation for each different methods are presented in Table 4. The number of iterations of training is set to 2000, and a dropout rate of 50% is used to prevent overfitting. Adam optimizer [38] is an efficient optimizer that occupies less computer resources and allows the model to converge faster. In addition, it can iteratively update neural network weights based on training data. So Adam optimizer is used to train the proposed model and make comparisons.

4.2. Evaluation Metrics. In this work, the classification report function in Sklearn library [39] of Python is used to display the main classification metrics. It can enumerate the precision rate, recall rate and fl-score of each category. For the effect evaluation of the fault diagnosis of the bogies, the classification results can be divided into true positive (TP), false positive (FP), and false negative (FN). These statistical indicators are calculated based on the real category of the sample and the category predicted by the machine. Precision rate P and recall rate R are defined as

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}.$$
(16)

F1-score is a harmonic mean based on P and R, which comprehensively considers the performance metrics of P and R as

$$F1 = \frac{2 \cdot P \cdot R}{P + R}.$$
(17)

As shown in Table 5, CRNN performs best on precision rate, recall rate, and f1-score compared to other methods. The experiment results show that all the neural networks based on deep learning framework have better performance in the

TABLE 5: Classification report of	comparison	experiments.
-----------------------------------	------------	--------------

	classification report		
method	Precision	Recall	F1-score
CNN_1d	0.95	0.95	0.95
LSTM	0.95	0.95	0.95
RF	0.94	0.93	0.93
GBDT	0.94	0.94	0.94
XGBoost	0.95	0.95	0.95
CRNN	0.98	0.98	0.98

above three statistical indicators than those methods based on the ensemble learning framework. It is worth pointing out that the neural networks with different kinds of hidden units have a stronger ability to learn nonlinear models than ensemble learning.

4.3. Time-Consuming Situation. It can be seen from Table 5 that CRNN, 1D-CNN, and LSTM all have higher precision rate than ensemble learning methods such as RF. However, CRNN obtains the best performance of 98% in accuracy, and the accuracy of 1D-CNN and LSTM also exceeds 95%. To further illustrate the superiority of the presented method, it needs comprehensive consideration of the time-consuming conditions and accuracy between CRNN, 1D-CNN, and LSTM. After each iteration of the training network, the accuracy and loss values of the test set are calculated once. As can be seen from the Figure 8(a), the CRNN method's testing loss curve remains at 0.08, and the diagnostic accuracy rate is 97.8%. In Figure 8(b), the testing loss curve of the 1D-CNN method also remains at 0.08, but a diagnostic accuracy of 95.2% is obtained. As displayed in Figure 8(c), the LSTM method testing loss curve remains at 0.21, and the diagnostic accuracy rate is 95.2%. The analysis results show that the CRNN method adopts a deeper neural network



FIGURE 8: The accuracies (y-axis) of CRNN, 1D-CNN, and LSTM for 2000 iterations on HST test sets. x-axis: training iterations used.

after integration of 1D-CNN and SRU. Therefore, CRNN has better detection capabilities than single-structure methods such as 1D-CNN and LSTM. Comparison of accuracy and time-consuming situations is shown in Table 6, from which it can be seen that CRNN maintains an accuracy rate of 97.8% on the test sets after 800 iterations and takes 24m35s. For the same test sets, the CRNN is significantly better than 1D-CNN and LSTM. Compared with 1D-CNN, CRNN does not consume much time while obtaining higher accuracy. In addition, compared with LSTM, CRNN not only reduces the time-consuming amount by 10 times, but also improves the accuracy by 2%. This means that the high accuracy and

Complexity

TABLE 6: Comparison of the time-consuming conditions and accuracy between CRNN, 1D-CNN, and LSTM.

		Method	
	CRNN	1D-CNN	LSTM
highest precision iterations	800	1600	1500
time-consuming situations	24m35s	6m40s	4h53m15s
accuracy	0.974	0.952	0.952



FIGURE 9: The confusion matrix of CRNN. The definition of the condition label of each fault is shown in Table 1.

low time-consuming situation of CRNN can be explained by the integrated structural advantages rather than the model parameter settings.

4.4. Misclassifications. To specifically analyze CRNN's identification of each test sample, the true class and the predicted class assigned the highest probability are compared on the each sample. Based on the comparison results for the each sample, the Sklearn library gives a CRNN confusion matrix as shown in Figure 9. Of the 147 test samples, 144 are accurately identified. It can be seen that the network is very sensitive to the faults of HST bogie and can completely distinguish whether the bogie operation status is normal. Moreover, the network also has a good ability to recognize the various faults of bogie. Compared to the 144 correctly classified samples, the 3 misclassified samples are more interesting. The sample with condition label 3 is misclassified as the faults with the label 6. A sample with fault label 4 is considered as the fault with the label 1, and a sample with fault label 1 is identified as the fault with the label 4. These misclassifications correspond to the results of the analysis in Section 2. The multifault signal features contain the feature components of single fault, which makes it difficult to distinguish between multiple faults and corresponding single faults. Faults with condition labels of 3 and 6 all contain failure factor of lateral dampers, and those with the labels of 1 and 4 both contain failure factor of air spring, which will inevitably cause misclassifications.

In order to visualize the classification result of the test set, the t-SNE [40] method is used to map the features of



FIGURE 10: t-SNE visualization of the output layer of CRNN. The condition label of each fault is defined in Table 1.

the last layer of the proposed CRNN to the low-dimensional space, which is shown in Figure 10. In the low-dimensional space, the boundary between different types is very clear, and the three misclassified test samples are also distributed on the edge of the misclassified class. This verifies that CRNN is very effective in feature extraction and fault recognition of HST bogies. In addition, CRNN also has time-saving advantages over the other welcomed methods. Therefore, CRNN can meet the high-precision and low-time-consuming requirements for fault diagnosis of HST bogie.

5. Conclusion

In this paper, the proposed CRNN is a combination method of 1D-CNN and SRU, which inherits the advantages of two complementary methods. The method first extracts features from bogie signals through a plurality of convolution layers (having a one-dimensional small filter). Then features extracted are passed to the stacked SRU recurrent layers to obtain hidden features with time series correlation. The hidden features are sent to the fully connected layer to calculate the probability of signal classification. The experimental results in Section 4 show that the deep learning method is more effective than the ensemble learning method for the fault diagnosis of HST bogie. More importantly, the recommended CRNN method has significant performance improvements over 1D-CNN and SRU. Specifically, the CRNN not only has a higher accuracy than the conventional model structure (i.e., 1D-CNN and LSTM), but also can significantly reduce the time spent in training. This means CRNN can simultaneously ensure the high efficiency and time saving of HST bogie fault diagnosis.

The experimental results we obtained in this work need to be further studied; i.e., it deserves applying more different methods based on deep learning to the fault diagnosis of HST. As a future direction of work, CRNN and other deep learning methods can be used to solve the pattern recognition of the gradual deterioration of key components that occur during actual operation of HST. For instance, monitoring data can be utilized to estimate the degree of change in train performance and conduct safety assessments. It is even possible to detect early dangers of HST by mastering the deterioration law of fault states.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially aided by the National Natural Science Foundation of China (nos. 61603316, 61773323, 61433011, and 61733015) and the Fundamental Research Funds for the Central Universities (2682018CX15).

References

- C. R. Garcia, A. Lehner, T. Strang, and M. Rockl, "Comparison of Collision Avoidance Systems and Applicability to Rail Transport," in *Proceedings of the 2007 7th International Conference on ITS Telecommunications*, pp. 1–6, Sophia Antipolis, France, June 2007.
- [2] Q. Yan and Y. Liu, "The Analysis of Vehicle Model Establishment and Malfunction Based on Matlab/Simulink," *Journal of East China Jiaotong University*, vol. 29, no. 5, pp. 13–17, 2012.
- [3] W. Duan and W. Jin, "Bogie Fault Signal Analysis of High-speed Trains Based on AR Power Spectrum," *Noise and Vibration Control*, vol. 2, no. 6, pp. 51–56, 2015.
- [4] Z. K. Peng and F. L. Chu, "Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography," *Mechanical Systems and Signal Processing*, vol. 18, no. 2, pp. 199–221, 2004.
- [5] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.
- [6] N. Qin, W. D. Jin, J. Huang, P. Jiang, and Z. M. Li, "High speed train bogie fault signal analysis based on wavelet entropy feature," *Advanced Materials Research*, vol. 753-755, pp. 2286– 2289, 2013.
- [7] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings A*, vol. 454, pp. 903–995, 1998.
- [8] Y. Lei, J. Lin, Z. He, and M. J. Zuo, "A review on empirical mode decomposition in fault diagnosis of rotating machinery," *Mechanical Systems and Signal Processing*, vol. 35, no. 1-2, pp. 108–126, 2013.
- [9] J. B. Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, and F. Fnaiech, "Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals," *Applied Acoustics*, vol. 89, pp. 16–27, 2015.
- [10] N. E. Huang, M. Wu, W. Qu, S. R. Long, and S. S. P. Shen, "Applications of Hilbert-Huang transform to non-stationary"

financial time series analysis," *Applied Stochastic Models in Business and Industry*, vol. 19, no. 3, pp. 245–268, 2003.

- [11] R. Yan and R. X. Gao, "Hilbert-huang transform-based vibration signal analysis for machine health monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 6, pp. 2320–2329, 2006.
- [12] Z. K. Peng, P. W. Tse, and F. L. Chu, "A comparison study of improved Hilbert-Huang transform and wavelet transform: application to fault diagnosis for rolling bearing," *Mechanical Systems and Signal Processing*, vol. 19, no. 5, pp. 974–988, 2005.
- [13] Z. H. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," *Advances in Adaptive Data Analysis (AADA)*, vol. 1, no. 1, pp. 1–41, 2009.
- [14] N. Qin, W. Jin, and J. Huang, "Fault diagnosis of high speed train bogie based on ensemble empirical mode decomposition," *Computer Engineering*, vol. 39, no. 12, pp. 1–10, 2013.
- [15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [16] Y. Kim, "Convolutional Neural Networks for Sentence Classification," https://arxiv.org/abs/1408.5882.
- [17] E. P. Giri, M. I. Fanany, A. M. Arymurthy, and S. K. Wijaya, "Ischemic stroke identification based on EEG and EOG using ID convolutional neural network and batch normalization," in *Proceedings of the 8th International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016*, pp. 484–491, Indonesia, October 2016.
- [18] Y. Aytar, C. Vondrick, and A. Torralba, "SoundNet: Learning sound representations from unlabeled video," in *Proceedings of* the 30th Annual Conference on Neural Information Processing Systems, NIPS 2016, pp. 892–900, Spain, December 2016.
- [19] H. Zhao, S. Sun, and B. Jin, "Sequential Fault Diagnosis Based on LSTM Neural Network," *IEEE Access*, vol. 6, pp. 12929–12939, 2018.
- [20] W. Lu, Y. Li, Y. Cheng, D. Meng, B. Liang, and P. Zhou, "Early Fault Detection Approach With Deep Architectures," *IEEE Transactions on Instrumentation and Measurement*, 2018.
- [21] T. de Bruin, K. Verbert, and R. Babuvska, "Railway track circuit fault diagnosis using recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 523–533, 2017.
- [22] T. Lei, Y. Zhang, and Y. Artzi, "Simple Recurrent Units for Highly Parallelizable Recurrence," https://arxiv.org/abs/1709 .02755.
- [23] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM with CNN Features," *IEEE Access*, vol. 6, pp. 1155–1166, 2017.
- [24] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.
- [25] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection," *IEEE/ACM Transactions* on Audio Speech and Language Processing, vol. 25, no. 6, pp. 1291–1303, 2017.
- [26] E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection," in *Proceedings of the 25th European Signal Processing Conference, EUSIPCO 2017*, pp. 1744–1748, Greece, September 2017.

Complexity

- [27] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional Recurrent Neural Networks for Electrocardiogram Classification," in *Proceedings of the 2017 Computing in Cardiology Conference*.
- [28] J. Xie, T. Li, Y. Yang, and W. Jin, "Learning features from High Speed Train vibration signals with Deep Belief Networks," in Proceedings of the 2014 International Joint Conference on Neural Networks, IJCNN 2014, pp. 2205–2210, China, July 2014.
- [29] 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 Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2106–2116, 2017.
- [30] X. Chen, S. Xiang, and L. Xu, "General Orbital Spectrum of Wuguang Passenger Dedicated Line and Its Inversion Simplified Algorithm," *Journal of Railway Science and Engineering*, vol. 13, no. 2, pp. 226–232, 2016.
- [31] V. K. Rai and A. R. Mohanty, "Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert-Huang transform," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2607–2615, 2007.
- [32] V. Gupta and M. Mittal, "Respiratory signal analysis using PCA, FFT and ARTFA," in *Proceedings of the 2nd International Conference on Electrical Power and Energy Systems, ICEPES* 2016, pp. 221–225, India, December 2016.
- [33] S. Lawrence Marple Jr., "Computing the discrete-time analytic signal via fft," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2600–2603, 1999.
- [34] H. Wang, H. Jiang, and D. Guo, "Bearing Fault Diagnosis Based on EEMD and AR Spectrum Analysis," in *Proceedings of the First Symposium on Aviation Maintenance and Management-Volume I*, vol. 296 of *Lecture Notes in Electrical Engineering*, pp. 389–396, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [35] T. Wang, M. Zhang, Q. Yu, and H. Zhang, "Comparing the applications of EMD and EEMD on time-frequency analysis of seismic signal," *Journal of Applied Geophysics*, vol. 83, pp. 29–34, 2012.
- [36] J. Wang, R. X. Gao, and R. Yan, "Integration of EEMD and ICA for wind turbine gearbox diagnosis," *Wind Energy*, vol. 17, no. 5, pp. 757–773, 2014.
- [37] B. Wang and H. Yang, "A method for interharmonic measurement based on AR model spectral estimation and interpolation FFT algorithm," *Relay*, vol. 34, no. 4, pp. 49–52, 2006.
- [38] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," https://arxiv.org/abs/1412.6980.
- [39] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [40] L. Maaten and G. Hinton, "isualizing data using t-SNE," *Journal* of Machine Learning Research, vol. 9, no. 5, pp. 2579–2605, 2017.



Operations Research

International Journal of Mathematics and Mathematical Sciences







Applied Mathematics

Hindawi

Submit your manuscripts at www.hindawi.com



The Scientific World Journal



Journal of Probability and Statistics







International Journal of Engineering Mathematics

Complex Analysis

International Journal of Stochastic Analysis



Advances in Numerical Analysis



Mathematics



Mathematical Problems in Engineering



Journal of **Function Spaces**



International Journal of **Differential Equations**



Abstract and Applied Analysis



Discrete Dynamics in Nature and Society



Advances in Mathematical Physics