Applied Mathematics, Modeling and Computer Simulation C. Chen (Ed.) © 2022 The authors and IOS Press. This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0). doi:10.3233/ATDE220071

Research on 1D-CNN Detection Methods of High-Speed Railway Catenary Dropper Faults Based on Acceleration Sensors

Yansheng GONG^a and Wenfeng JING^{b,1}

^aChina Railway First Survey and Design Institute Group Co., Ltd, Xi'an, China ^bXi'an Jiaotong University, Xi'an, China

Abstract. Droppers are key components of high-speed railway overhead catenary systems, which are exposed to the external environment and are prone to breakage faults due to the impact of wind force and the pantograph on moving trains day after day. How to identify dropper breakage or relaxation faults through acceleration signals installed in the carrier cable and contact wire is a challenging problem. In this study, the experimental section of the Lanzhou-Xinjiang high-speed railway was simulated on the basis of the bow-network dynamic simulation model, in which the overhead catenary system was subjected to the force of pulsating wind alone or pulsating wind and the pantograph at the same time. In the experiment, we collected 10 channel signals from five acceleration sensors when two droppers were normal or broken. We established a 1D-CNN model of four categories and then determined the hyperparameters of the deep network structure and the important parameters of the network optimization scheme through the Bayesian optimization algorithm. Furthermore, we selected the lowest sensor number to identify dropper fracture faults by a large number of experiments according to mechanics principles. The experimental results show that the proposed methods have a higher identification accuracy rate, recall rate, and robustness than the traditional artificial feature extraction approaches. Therefore, the detection methods proposed provide an effective way to identify high-speed railway catenary dropper faults on the basis of acceleration sensors.

Keywords. Overhead catenary system, dropper, accelerometer, convolutional neural network, fault detection.

1. Introduction

The overhead catenary system (OCS) is an important instrument for ensuring stable power supply for electric locomotives. Such systems work in the open air and are prone to breakage faults caused by factors such as the external environment, climate, and the impacts of the train pantograph and the wind from day and day [1]. In particular, the core component of the OCS, the dropper, plays a role in fixing the contact wire by connecting the carrier cable. Under the action of stress, it is prone to breakage failure, which affects the quality of the bow-net flow and threatens the safety of train operation. Figure 1 shows a scene of the over-head catenary system, and the dropper in the red box is slack and bears no force. For example, there were as many as 60 dropper breakages on a certain

¹ Corresponding Author, Wenfeng JING, Xi'an Jiaotong University, Xi'an, China; E-mail: wfjing@xjtu.edu.cn.

railway line in 2011. Therefore, the online monitoring of the working status of droppers to find their failures in a timely manner is a powerful measure to ensure OCS safety.



Figure 1. The dropper in the red box is slack and bears no force in a scene of the overhead catenary system.

Currently, in order to ensure train safety, OCS maintenance is mainly carried out by on-site inspection, manual video browsing of the 2C system, and picture verification of the 4C system. These methods are inefficient, easily miss defects, and costly. Therefore, online detection of dropper failures is a technical problem that needs to be solved urgently by the railway department.

In recent years, researchers have proposed a variety of methods for the detection of catenary defects. In general, they can be divided into contact and non-contact detection methods. The non-contact method uses image recognition technology to find the dropper fault from visual inspection. The authors of [1] applied cascaded convolutional neural networks to detect defects on catenary support devices. Chen et al. [2] proposed a visionbased method by applying deep convolutional neural networks (DCNNs) to the detection of fastener defects. Huang et al. [3] implemented defect detection of seven key areas (including droppers) of the catenary based on convolutional neural networks using highdefinition images taken by the 4C inspection train. However, these methods have limited detection accuracy and are prone to false positives and false negatives. In particular, it should be pointed out that fault recognition methods based on images cannot recognize a situation in which the dropper seems normal visually, but the actual stress has changed. In order to overcome the shortcomings of the non-contact detection method, it is also necessary to install the acceleration sensor on the contact net to obtain the data of the running state signals and study the intelligent fault diagnosis methods. Zhao et al. [15] proposed a feature extraction method for high-speed rail fault diagnosis based on integrated signal empirical mode decomposition (EEMD) and fuzzy entropy by extracting the mean value of the fuzzy entropy of the intrinsic modal function as features and then using a neural network to identify each working condition. The authors of [1] analyzed the dynamic response characteristics of faults in different catenary structures from the perspective of catenary-pantograph coupling dynamics, and they used support vector machines to identify fault features. Zhao et al. [4] using the hybrid filter to denoise the noisy signal and the wavelet transform of the signal, calculated the Lipschitz

exponent according to the principle of modulus maxima, and realized the detection of wire breakage. How-ever, the failure recognition rate of these methods needs to be further improved.

At present, there are few studies in the literature on fault diagnosis for catenary accelerations, but similar mechanical fault diagnosis research can be used for reference. Huang [3] and others proposed a fault diagnosis method that combined wavelet packet decomposition (WPD) and multi-scale permutation entropy (MPE). After wavelet packet decomposition of the original vibration signal, all sub-sequences of the sub-band signals were extracted to obtain the average permutation entropy, and finally, the Hidden Markov Model (HMM) was used to classify and identify rolling bearings. Janssens et al. [5] used a Fourier transform to transform the signal data into frequency domain pictures and then utilized a 2D convolutional neural network for fault recognition. Wang et al. [6] proposed an adaptive deep convolutional neural network (CNN) to diagnose rolling bearing faults, applied particle swarm optimization to initialize the weights of the CNN, and used t-stochastic Neighbor Embedding (t-SNE) to visualize the learning situation of each layer of the network. Although the two-dimensional convolutional neural network has excellent feature learning capabilities, it still needs to preprocess the original signal data and convert it into a two-dimensional image. The one-dimensional convolutional neural network is an important branch of the convolutional neural network [7,8], which can effectively solve tasks such as speech recognition, electroencephalogram (ECG) classification, and signal-based fault detection. Zhang [9] and others proposed a onedimensional convolutional neural network (1D-CNN) model for bearing fault diagnosis, which directly acts on the original time-domain signal and successfully avoids other complex operations. Abdeljaber et al. [10] also used one-dimensional vibration signals as input to effectively solve the problem of fault identification and location. The research efforts above show that it is feasible to use a one-dimensional convolutional neural network for fault diagnosis.

In this study, the experimental section of the Lanzhou-Xinjiang high-speed railway was simulated on the basis of the bow-network dynamic simulation model, in which the overhead catenary system was subjected to the forces of pulsating wind alone or pulsating wind and the pantograph simultaneously. We collected 10 channel signals from five acceleration sensors when two droppers were normal or broken. We established a 1D-CNN model of four categories and then determined the super parameters of the deep net-work structure and the important parameters of the network optimization scheme through the Bayesian optimization algorithm. Furthermore, we selected the lowest sensor number to identify dropper fracture faults by a large number of experiments according to mechanics principles.

The structure of this article is as follows: Section 2 describes the problems of catenary dropper breakage detection, and Section 3 introduces the methodology. Section 4 pro-vides the results and discussion.

2. Problem Statement

The overhead catenary system is always in operation. It is impossible to install sensors on it for experiments. In this study, the acceleration sensors were installed on the contact cable and carrier cable by simulation to obtain the acceleration signal data of the droppers. From the data, we established a deep network model of the dropper breakage detection and furthermore selected the lowest sensor number to identify dropper fracture faults by experiments according to mechanics principles.

2.1. Test Section of the High-speed Railway

In this study, the dynamic simulation model of the bow-net was established for two semianchor sections with a mileage of K3066+568.795~K3065+588.795 on a high-speed railway line. The relevant design parameters of the catenary are shown in table 1.

Parameter name	Numerical value	Parameter name	Numerical value
Nominal span (m)	50	Load line tension (kN)	21.0
Dropper form	Simple chain type	Contact cable tension (kN)	28.5
Cable type	JTMH120	Quality per unit length of cable (kg/m)	1.065
Contact cable type	CTMH150	Quality per unit length of contact line (kg/m)	1.350
Sling interval (m)	5	Rigid suspension chord section (mm ²)	28
Dropper number	9	Chord fault analysis interval (m)	750~800
Pull out value (m)	0.25	Wind speed (m/s)	20

Table 1. Overhead catenary system design parameters

The pantograph uses the DSA250 type, which is equivalent to the three-reduced mass-damping-stiffness model in figure 2. Their design parameters are shown in table 2. Assuming that the static contact force is 70 kg, the aerodynamic force is calculated according to 0.00097v2 in the standard IEC 62486, where v is the pantograph running speed with single bow operation, and v is 250 km/h.



Figure 2. Three-reduction quality model of the pantograph.

Symbol	Numerical value	Symbol	Numerical value
<i>m</i> ₃ (kg)	7.51	c_3 (Ns/m)	0
<i>m</i> ₂ (kg)	5.855	c_2 (Ns/m)	0
<i>m</i> ¹ (kg)	4.645	c_1 (Ns/m)	70
<i>k</i> ₃ (N/m)	8380	$f_{3}(\mathbf{N})$	0.5
<i>k</i> ₂ (N/m)	6200	$f_2(\mathbf{N})$	3.5
k_1 (N/m)	80	f_1 (N)	3.5

Table 2. Parameters of the dsa250 pantograph simulation model

In the OCS, the length of a span is 50 meters, and there are nine droppers evenly distributed in a span. The first dropper is installed near the left support frame at 5 meters, and the midspan dropper is located in the middle of a span. The installation positions of the accelerometers are shown in figure 3. Acceleration sensor #1 is added close to the contact wire clamp and near the wrist arm at 0.7 m, and sensor #2 and sensor #3 are installed at the middle position of the carrier cable and near the right support frame, respectively. Sensor #4 and sensor #5 are located in the middle and right positions of the contact wire.



Figure 3. Installation space distribution of sensors.

The entire test section is 1.2 km long, and the selected breakage point of the droppers was at span #21.

2.2. Simulation Data

We simulated two working conditions: (1) working condition A involves only the impact of pulsating wind, there is no dropper breakage (normal), or there is breakage in the first dropper, the middle dropper, or the first dropper and the middle dropper simultaneously; (2) working condition B involves the joint impact of pulsating wind and the ballistic force of the train pantograph running at 250 km/s, there is no dropper breakage (normal), or there is breakage in the first dropper, the middle dropper, or the first dropper and midspan dropper simultaneously. The above conditions amount to a total of eight cases, and each case was simulated and calculated 200 times using finite element analysis software. In the experiment, the horizontal and vertical acceleration signals of sensors $\#1\sim\#5$ were obtained with a sampling frequency of 277 Hz. The sample size of the simulation data is as follows:

2.3. Research Target

The goal of this research was to apply the simulation data in table 3 to establish a prediction model under working conditions A and B for no dropper breakage (Normal), dropper #1 breakage, dropper #2 breakage, and simultaneous breakage of droppers #1 and #2 to determine the lowest acceleration sensor number and their installation positions such that the dropper fracture faults can be identified accurately for working conditions A and B.

Working condition	There is no pantograph action	Running state	Number of samples
		No hanger breakage (normal)	200
Working	Only pulsating wind impact	First dropper breakage	200
condition of A		Middle dropper breakage	200
		Simultaneous breakage of the first dropper and the middle dropper	200
		No hanger breakage (normal)	200
Working condition of B	Simultaneous impact	First dropper breakage	200
	with pulsating wind and train pantograph	Middle dropper breakage	200
	aun punograph	Simultaneous breakage of the first dropper and the middle dropper	200

Table 3. The sample size of simulation data

3. Methodology Development

In this study, the acceleration signals obtained in the simulation experiments were 10channel one-dimensional time series data. As described in this section, a fourclassification 1D-CNN deep network model was established by the data set.

3.1. Four-classification 1D-CNN Model

A Convolutional Neural Network (CNN) is a typical network structure in feed-forward neural networks, which combines multiple filters to extract the features of input data. A CNN is generally composed of three parts to map the original data to the hidden layer feature space: an input layer, some feature extraction layers, and a few fully connected layers, where the feature extraction layers are composed of multiple convolutional layers, activation layers, and pooling layers. The fully connected layers carry out feature fusion to classify the data.

According to the traits for the dropper fault detection, we established the 1D-CNN deep network model shown in figure 4. It is a four-classification model that includes an input layer I; three convolutional layers C1, C2, and C3; three activation layers A1, A2, and A3; three pooling layers P1, P2, and P3; two fully connected layers FC1 and FC2; and one output layer O with four dimensions.



Figure 4. Four-classification 1D-CNN network structure.

3.1.1. Input Layer

As shown in figure 3, the length of the catenary span length is 50 meters. Sensors $\#1 \sim \#5$ collect 201 discrete points of signal data from a total of 10 channels of acceleration signals in the horizontal and vertical directions at a sampling frequency of 279 Hz Convolutional Layers. As shown in figure 4, 201 × 10-scale signal data are directly input into the 1D-CNN deep network model.

3.1.2. Convolutional Layer

After the input layer, convolutional kernels (Convolutional Kernels) are used to convolve the input data (or the feature map of the previous layer) to capture the local features of the acceleration signal. Each neuron of the convolutional layer is only connected to some neurons of the previous layer and then summarizes the local features at a higher layer, thereby obtaining the overall features. The convolutional layer has a weight-sharing feature: that is, when each convolution kernel traverses an input, the parameters are fixed, which greatly reduces the number of parameters of the convolutional layer and prevents the introduction of too many parameters. The expression of the convolution operation is

$$y^{l(i,j)} = k_i^l * x^{l(r^j)} = \sum_{j'=0}^{w-1} k_i^{l(j')} x^{l(j+j')}$$
(1)

where $k_i^{l(j')}$ is the *j*'-th weight of the *i*-th convolution kernel at the *l*-th layer; $x^{l(r^j)}$ is the *j*-th local area convolved in the *l*-th layer; *w* is the width of the convolution kernel.

As shown in figure 3, the convolutional layer C1 is set as a large convolution kernel, which extracts the features of the larger neighborhood of the input signal data and effectively suppresses noises. The kernel size can be selected in the set $\{3,5,7,9,11,16\}$, and the number of convolution kernel filters can be set to 16 or 32. The C2 and C3 convolutional layers can generally use small convolution kernels, for example, 3×1 , and the number of convolution kernels is 32 or 64. The one-dimensional convolution

operation is shown in figure 5. For the input signal data of 201×10 , zero-padding is taken as 2, a step of 1 is used to slide in the convolution area until the input data are traversed, and the output feature vector is 101×1 .



Figure 5. One-dimensional convolution diagram.

3.1.3. Activation Layer

In order to enhance the expression ability of the network, the activation function is used to perform a non-linear transformation on the feature vector output through the convolution operation. Activation layers A1, A2, and A3 follow the convolution layers C1, C2, and C3, respectively. The dimensions of the activation layer are the same as those of the convolution layers. Three types of activation functions can be used: Sigmoid function, hyperbolic tangent function (Tanh), and modified linear unit function ReLU. The specific activation function used by each activation layer is determined by the hyperparameter optimization method in Section 3.2.

The output range of the sigmoid activation function is between (0,1), has symmetry, and can be deduced every-where in the definition domain. However, because of its soft saturation, gradient explosion and gradient disappearance can easily occur when the deep neural network backpropagates.

$$a^{l(i,j)} = sigmoid(y^{l(i,j)}) = \frac{1}{1 + e^{-y^{l(i,j)}}}$$
(2)

The Tanh function also has soft saturation, and the problems of gradient disappearance and gradient explosion may also occur. However, because its output mean is closer to 0 compared with that output by the Sigmoid function, it converges faster.

$$a^{l(i,j)} = \tanh(y^{l(i,j)}) = \frac{e^{y^{l(i,j)}} - e^{-y^{l(i,j)}}}{e^{y^{l(i,j)}} + e^{-y^{l(i,j)}}}$$
(3)

Because of its linear and unsaturated nature, ReLU can effectively solve the problem of gradient disappearance during backpropagation, and it adds sparse expression ability to the network. However, as the training number increases, there may be neuron death. This is because its gradient has been 0 during the process of backpropagation, and the weight cannot be updated.

$$a^{l(i,j)} = relu(y^{l(i,j)}) = \max\left\{0, y^{l(i,j)}\right\}$$
(4)

3.1.4. Pooling Layer

In order to reduce network parameters and improve the robustness of the network, pooling layers P1, P2, and P3 are added after the activation layers A1, A2, and A3, respectively. Pooling is a down-sampling process, which aggregates the information of the local area to obtain a feature value for the feature signal after convolution, which can greatly reduce the parameters of the neural network and avoid over-fitting. In particular, if the maximum pooling function is used, when the local information undergoes some small changes, the neurons in the next layer do not change, making the network very robust [11].

The commonly used pooling functions are average pooling and maximum pooling. The average pooling and the maximum pooling use the average value and the maximum value in the pooling window as the output value, respectively. The mathematical formulas of the two functions are as follows:

$$p^{l(i,j)} = \frac{1}{w} \sum_{t=(j-1)w+1}^{jw} a^{l(i,t)}$$
(5)

$$p^{l(i,j)} = \max_{(j-1)w+1 \le t \le jw} (a^{l(i,t)})$$
(6)

where $a^{l(i,t)}$ is the activation value of the *t*-th neural unit of the *i*-th feature diagram at the *l*-th layer; *p* is the output value of the *t*-th neural unit of the *i*-th feature map in the *l*-th layer after pooling; *w* is the width of the pooling window.

3.1.5. Fully Connected Layer

In order to classify the operating state of the droppers, two fully connected layers, FC1 and FC2, are designed after the last pooling layer. FC1 is composed of a one-dimensional feature vector formed by flattening the output of the last pooling layer P3, and FC2 is a vector whose dimensions can be 50, 100, 256, or 512. The expression of the fully connect-ed layer is

$$z^{(l+1)(j)} = \sum_{i=1}^{n} w_{ij}^{l} a^{l(i)} + b_{j}^{l}$$
(7)

where $z^{(l+1)(j)}$ is the output value of the *j*-th neuron in the (l+1)-th layer; is the weight between the *i*-th neuron in the *l*-th layer and the *j*-th neuron in the (l+1)-th layer; b_j^l is the bias of all neurons in the *l*-th layer against the *j*-th neuron in the (l+1)-th layer.

3.1.6. Output Layer

According to table 3, working conditions A and B contain four types of faults: no dropper breakage (normal), first dropper breakage, middle dropper breakage, and simultaneous breakage of the first dropper and middle dropper. Aiming to detect these four kinds of faults, a deep network classification model was established. The output layer uses the "Softmax" function to convert the output of the neuron to a value between 0 and 1, which corresponds to the four running states above.

In particular, to avoid overfitting the network, Dropout is used after the C1, C2, C3, and FC2 layers, and the ratio d is randomly selected in the interval [0, 0.8].

3.2. Hyperparameter Optimization

The 1D-CNN deep network model described in Section 3.1 contains many hyperparameters that have a great impact on the performance of deep networks. As shown in table 4, the hyperparameter space has 14 dimensions, and there are up to 55296 possible combinations of hyperparameter. Because the deep network training and model evaluation need to take a long time for a set of determined hyper-parameters, hyperparameter optimization is a very complex process.

Super parameter		Value range
C1_kernel_size		3,5,7,9,11,16
C1_filters		16, 32
C1_strides		1, 2,
C1, C2, C3, FC2 activation fu	nction	relu, tanh, sigmoid
C1, C2 and C3 and FC2_Drop	oout	[0,0.8]
An optimization method		rmsprop, adam, adagrad, nadam
Batch_size		16, 32, 64
Number of FC2 hidden layer u	units	50, 100, 256, 512

Table 4. 1d-cnn hyperparameter space

The objective function of hyperparameter optimization is multi-peak, non-convex, high-dimensional, and even without expression, and it is difficult to solve. The hyperparameter optimization methods commonly used include grid search, random search, genetic algorithm, particle swarm optimization, and Bayesian optimization [12]. Grid search needs to traverse all the parameter combinations, which can easily cause the combination to explode, so its calculation speed is very slow, and it easily obtains local optimization for non-convex problems. Genetic algorithms and particle swarms require a large number of function values of the initial population and are generally not used because their computational efficiency is too low. Bayesian optimization [13] uses existing prior information to solve optimization problems and does not have a clear

objective function. Compared with random or grid search, its evaluation cost of the objective function is lower. On this basis, the Bayesian optimization algorithm was used in this study for deep network hyperparameter optimization.

The Bayesian optimization algorithm [14] has four core elements: domain space, objective function, probabilistic proxy model, and acquisition function. For the hyperparameter optimization in this paper, the specific core elements are as follows:

- Domain space: it indicates the range of each hyperparameter value and is determined according to the probability distribution of each hyperparameter, as shown in table 4.
- Objective function: it takes the negative value of the prediction accuracy rate of the four-category model on the test set as the objective function value of hyperparameter optimization.
- Probabilistic proxy model: It is the true optimization objective of Bayesian optimization methods, and it can avoid the time-consuming hyperparameter evaluation process, thereby reducing computational overhead and improving efficiency. The commonly used proxy models include the Gaussian process, random forest, and Tree-structured Parzen Estimator (TPE) [14], among others. The TPE probabilistic proxy model was used in this study.
- Acquisition function: the acquisition function is derived from the TPE proxy model [14].

4. Experiments and Analysis

In the experiments described in this section, network training was performed on the fourclassification deep network model established in working conditions A and B, and the effects of the signals of 10 acceleration sensors on the classification models were analyzed.

The computer configurations used in the experiments were CPU i7 6700, a main frequency of 3.4 GHz, and 32 GB memory. The TensorFlow deep learning framework and Keras library of neural networks were used in the Python development language to build the four-classification 1D-CNN deep network model shown in figure 3.

4.1. Time-frequency Analysis of Acceleration Signals

In order to understand the time–frequency characteristics of acceleration signals when the dropper breaks, the 10 acceleration vibration signals collected by five sensors are shown in figure 6 and figure 7 in the time-domain and frequency-domain diagrams for the four states under working conditions A and B in table 3.

In every subgraph at the top of figure 6 and figure 7, labels a, b, c, and d represent four operating states: the first dropper breakage, the span-center dropper breakage, the simultaneous breakage of the first and span-center dropper, and no dropper breakage, respectively.

From figure 6, it can be found that the vertical acceleration signals of the four operating states are stronger than the horizontal ones. When the droppers break, the amplitudes of vibration acceleration in the vertical direction markedly increase, and the largest frequencies among them move toward high frequencies. Furthermore, it is found that the time-domain and frequency-domain graphs of 10 channels are very different

when the first dropper and the span-center dropper break at the same time. The timefrequency graphs of acceleration signals in four operating states with the simultaneous impact of pulsating wind and the train pantograph are shown in figure 7. Compared with figure 6, it can be seen clearly that the acceleration signals on the 10 channels of the four operating states all increase. Among them, the acceleration signals change markedly when the dropper breaks, and the vibration amplitudes of the three fault states are greater than that in the normal state.



a) Vertical acceleration of sensor #1 on carrier cable



b) Horizontal acceleration of sensor #1 on carrier cable



c) Vertical acceleration of sensor #2 on carrier cable



d) Horizontal acceleration of sensor #2 on carrier cable



e) Vertical acceleration of sensor #3 on carrier cable



f) Horizontal acceleration of sensor #3 on carrier cable



g) Vertical acceleration of sensor #4 on contact cable



Figure 6. Time-frequency graphs of acceleration signals in four operating states with the impact of pulsating wind only (working condition A).

By comparing figure 6 and figure 7, it can be seen that the vibration with the simultaneous impact of pulsating wind and the train pantograph is stronger than that with the impact of pulsating wind only.

From the data described in table 3, the four-classification 1D-CNN deep network model shown in figure 3 was established, and the network models were trained for working conditions A and B separately. The sample size of each working condition was 800, the training set was divided into 512 samples each time, the verification set was 128 samples, and the test set was 160 samples.







time(s)



140



d) Horizontal acceleration of sensor #2 on carrier cable







f) Horizontal acceleration of sensor #3 on carrier cable







h) Horizontal acceleration of sensor #4 on contact cable





j) Horizontal acceleration of sensor #5 on contact cable

Figure 7. Time-frequency graph of acceleration signals in four operating states with the simultaneous impact of pulsating wind and the train pantograph (working condition B).

There are many hyperparameters in the four-classification 1D-CNN deep network model, such as the size of the convolution kernel, the number of convolution kernels, the activation function, the optimizer, and the number of neurons in the fully connected layer, all of which need to select the optimal value. On the basis of the Bayesian optimization method, the TPE probabilistic proxy model was used in this study to automatically optimize the hyperparameters of the 1D-CNN deep network. The number of iterations was set to 30, and the optimal parameters automatically selected by the Bayesian optimization method are shown in table 5.

Tabl	le 5.	Н	yperp	arame	ter	aut	toma	tic	se	lect	ion	resu	lt	
------	-------	---	-------	-------	-----	-----	------	-----	----	------	-----	------	----	--

Super parameter	The optimal parameter values	Super parameter	The optimal parameter values
'C1_kernel_size'	16	'C1_Dropout'	0.24
'C1_filters'	32	'C2_Dropout'	0.0001
'C1_strides'	2	'C3_Dropout'	0.05
'C1_Activation'	tanh	'FC2_Dropout'	0.75
'C2_Activation'	relu	'optimizer'	adagrad
'C3_Activation'	tanh	'batch_size'	32
'FC2_Activation'	tanh	Dense '(FC2 Layer Number)	256

It can be seen from table 5 that the activation function and the dropout ratio of each layer are different. It can be seen that the automatically selected hyperparameter values

are more flexible than those manually set. Using Keras framework model visualization, the final trained model structure is shown in figure 8.



Figure 8. The 1D-CNN network structure automatically selected by the Bayesian optimization method.

After the hyperparameters of the deep network were determined, the deep network was trained 50 times on the data of working conditions A and B, respectively. In case A, after 100 iterations, the loss of the training set and the verification set can converge, and the recognition rate reaches $96.75\% \pm 0.88\%$. Case B only needs 20 iterations of loss to converge, and the recognition accuracy rate is $99.40\% \pm 0.55\%$. The time to diagnose a signal using the 1D-CNN model is about 0.0064 s, which can effectively meet the requirements of real-time detection. Figure 9 and figure 10 are the training iteration process diagram and the confusion matrix of case A and case B, respectively.





Figure 9. Four-class 1D-CNN model training iterative process.



Using the t-SNE non-linear dimensionality reduction method, figure 11 shows the distribution of all samples by two-dimension features after the convolutional layers C1, C2, and C3. As can be seen from (b) of figure 10, after the first layer C1 of the convolution operation, the four types of acceleration signals can be well classified, and the distances of the four categories increase with the increase in the number of convolution layers, while, in (a) of figure 10, until the third layer C3, the four categories of acceleration signals are basically classified. Figure 11 shows that the 1D-CNN model's non-linear expression ability is enhanced with the increase in the number of convolutional layers.



Figure 11. The classification effect of all samples is visualized by t-SNE dimensionality reduction in each convolution layer.

4.2. Analysis of the Influence of Sensor Installation Quantity and Location on Dropper Fault Detection

The impact of the five sensors on the diagnostic accuracy of the model is analyzed below.

For working conditions A and B, we removed the data of a few sensors and input the acceleration signals to the 1D-CNN model shown in figure 9. The obtained recognition accuracy, model training time, and detection time after 50 training times are shown in table 6.

Working condition	Sensor number	Accuracy	Training time (s/Epoch)	Testing time (s/sample)
	#1-#5	96.75%±0.88%	0.71	0.0064
Working condition	#1, #2, #4	89.98%±1.58%	0.72	0.0061
A	#1, #2, #3, #4	94.39%±1.53%	0.68	0.0059
	#1, #2, #4, #5	94.33%±1.83%	0.67	0.0062
	#1-#5	99.40%±0.55%	1.217	0.0062
Working condition B	#1, #2, #4	99.38%±0.50%	1.153	0.0061
	#1, #4	98.04%±0.82%	1.126	0.0062
	#1,#2	99.97%±0.11%	1.154	0.0062

Table 6. Influence of sensors on model accuracy

It can be seen from table 6 that the number of sensors has little influence on the model training time and detection time, but it has a greater influence on the recognition accuracy of the model.

For working condition A, if only one sensor is removed, the accuracy drop rate is about 2.43%, and the effect of sensor #3 and sensor #5 on classification accuracy is almost the same. When sensors #1-#4 or sensors #1, #2, #4, and #5 are installed, the detection accuracies of the corresponding models do not differ much. Considering the

data comprehensively, it is recommended to install sensors #1-#5 when detection involves the action of pulsating wind only.

For working condition B, when removing sensor #2, the accuracy decreases by 1.36%, and the standard deviation increases; when only sensors #1 and #2 are retained, the accuracy increases instead, reaching 99.97%, and the standard deviation is only 0.11%. Therefore, it is recommended that under working condition B, only the acceleration signal data of sensors #1 and #2 need to be obtained.

Analyzed from the principle of catenary mechanics, working condition B is the simultaneous action of pulsating wind and the train pantograph, and the acceleration signals of the sensors are stronger than those under working condition A (only pulsating wind). For working condition B, the network model more easily obtains higher recognition accuracy.

5. Results and Discussion

For the detection of catenary dropper breakage or slack, this paper establishes an end-toend detection model based on acceleration signals using 1D-CNN. There are four situations that may occur for the catenary droppers in our test section of a high-speed railway: no dropper breakage (normal), breakage of the first dropper, breakage of the span-center dropper, and simultaneous breakage of the first dropper and the span-center dropper. In particular, this article, combined with the mechanical principle of the overhead catenary system, through a large number of experiments, demonstrates that under condition A (only pulsating wind action), it is necessary to collect the acceleration signals of sensors #1-#5 for fault detection; in condition B (pulsating wind and train pantograph simultaneous action), we can detect the defects of catenary droppers with high accuracy by only using the data of the acceleration signals of sensors #1 and #2.

In summary, the main results of this work include the following:

- Using 1D-CNN, a deep network structure of multichannel signals was established for automatically learning the characteristics of acceleration signals, which is better than the traditional manual feature extraction method.
- The Bayesian optimization algorithm was used to automatically select the structural parameters and network training parameters of 1D-CNN, which greatly improved the efficiency of deep network training.
- The minimum numbers of acceleration sensors under conditions A and B were determined by experiments, which provides a basis for engineering implementation.
- This paper presents the end-to-end detection methods of catenary dropper breakage, and acceleration signal data are the only direct input needed to detect the working state of the droppers.

In short, compared with the traditional artificial feature extraction methods, the methods proposed in this paper have high recognition accuracy and strong robustness, and a new effective method for online identification of catenary dropper faults is explored.

References

- Chen JW, Liu ZG, Wang, HR, et al. High-speed railway catenary detection using the cascaded convolutional neural networks. IEEE International Conference on Imaging Systems and Techniques (IST). Beijing, 2017; 1-6.
- [2] Chen JW, Liu ZG, Wang HR, et al. Automatic defect detection on the catenary support device using deep convolutional neural network. IEEE Transactions on Instrumentation & Measurement. 2018; 67: 257-269.
- [3] Huang NE, Shen Z, Long SR, et al. Empirical mode decomposition and the Hilbert spectrum for nonlinear time series analysis. Proceedings A. 1998; 454: 903-995.
- [4] Zhao LY, Wang LW, Yan RQ. Rolling fault fiagnosis based on wavelet packet decomposition and multiscale permutation. Entropy. 2015; 17: 6447-6461.
- [5] Olivier J, Viktor S, Bram V, et al. Convolutional neural network based fault detection for rolling machinery. Journal of Sound and Vibration. 2016; S0022460X16301638.
- [6] Wang FA, Jiang HK, Shao HD, et al. An adaptive deep convolutional neural network for rolling bearing fault diagnosis. Measurement Science and Technology.2017.
- [7] Liu C, Hou WY, Liu DY. Foreign exchange rates forecasting with convolutional neural network. Neural Processing Letters. 2017; 46: 1095-1119.
- [8] David G, Xie XH, Gary KL. 3D mesh via multi-branch 1D convolutional networks. Graphical Models, 2018; 96: 1-10.
- [9] Zhang W, Peng GL, Li CH. Rolling element fault intelligent diagnosis based on convolutional neural networks using raw sensing signal. Advances in Intelligent Information Hiding and Multimedia Signal Processing. Springer International Publishing. 2017.
- [10] Abdeljaber O, Avci O, Kiranyaz S, et al. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. Journal of Sound and Vibration. 2017; 388: 154-170.
- [11] Lecun Y, Bengio Y. Convolutional networks for images, speech, and time series. The handbook of Brain Theory and neural Networks. MIT Press. 1998.
- [12] Hugo L, Dumitru E, Aaron C, et al. An empirical evaluation of deep architectures on problems with many factors of variation. Proceedings of the 24th International Conference on Machine Learning. 2007; 473-480.
- [13] Pelikan M. The bayesian optimization algorithm. Proc Genetic & Evolutionary Computation Conference. 1999.
- [14] Bergstra J, Bardenet R, Bengio Y, et al. Algorithms for hyper-parameter optimization. Advances in Neural information Processing Systems. 2011; 2546-2554.
- [15] Zhao JJ, Yang Y, Li TR, et al. Application of empirical mode decomposition and fuzzy entropy to HST fault diagnosis. Proceedings of the 8th International Conference on Intelligent Systems and Knowledge Engineering. 2014; 93-10.