

Article

# Imbalanced Data Fault Diagnosis Based on an Evolutionary Online Sequential Extreme Learning Machine

Wei Hao <sup>1,2,\*</sup> and Feng Liu <sup>1</sup>

<sup>1</sup> School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China; fliu@bjtu.edu.cn

<sup>2</sup> Department of Information Technology, CRRC Qingdao Sifang Limited Company, Qingdao 266111, China

\* Correspondence: 15112100@bjtu.edu.cn; Tel.: +86-158-0642-8776

Received: 9 June 2020; Accepted: 20 July 2020; Published: 22 July 2020



**Abstract:** To quickly and effectively identify an axle box bearing fault of high-speed electric multiple units (EMUs), an evolutionary online sequential extreme learning machine (OS-ELM) fault diagnosis method for imbalanced data was proposed. In this scheme, the resampling scale is first determined according to the resampling empirical formulation, the K-means synthetic minority oversampling technique (SMOTE) method is then used for oversampling the minority class samples, a method based on Euclidean distance is applied for undersampling the majority class samples, and the complex data features are extracted from the reconstructed dataset. Second, the reconstructed dataset is input into the diagnosis model. Finally, the artificial bee colony (ABC) algorithm is used to globally optimize the combination of input weights, hidden layer bias, and the number of hidden layer nodes for an OS-ELM, and the diagnosis model is allowed to evolve. The proposed method was tested on the axle box bearing monitoring data of high-speed EMUs, on which the position of the axle box bearings was symmetrical. Numerical testing proved that the method has the characteristics of faster detection and higher classification performance regarding the minority class data compared to other standard and classical algorithms.

**Keywords:** axle box bearing; fault diagnosis; imbalanced samples; online sequential extreme learning machine; artificial bee colony; high-speed EMU

## 1. Introduction

In recent years, remarkable achievements have been made in the construction of high-speed electric multiple units (EMUs), with over 3500 sets of standard high-speed EMUs put into service and a railway operating distance reaching 29,000 km in China alone. Thus, the safety issue of high-speed EMUs is of great importance. The axle box bearing, which is one of the essential parts of a high-speed EMU, runs at a very fast speed under complicated and threatening environments and suffers a heavy load [1]. It poses a risk to railway safety when a train fully loaded with passengers is running at a high speed [2]. Therefore, it is of great importance to research techniques for bearing fault diagnosis [3–6] and the time is ripe for such a study, thanks to the development of big data, artificial intelligence, and the Internet.

As imbalanced as the axle box bearing states typically are, being able to identify the minority class faults with accuracy is much more crucial than identifying the normal ones in the majority class [7]. To shed light on this, when a faulted axle box bearing is diagnosed as normal, it might lead to derailment and death, causing great casualties. However, regarding a normal state axle box as a faulted one would probably bring a temporary halt to the train. Yet, the first situation is much more severe

and undesirable than the second one. Apart from that, the time between the start of a fault and the axle cut cannot be regularly measured, as it sometimes only lasts a few minutes; therefore, rapid and precise fault diagnosis is required [8–10]. Put simply, to realize the identification of imbalanced axle box bearing fault states in high-speed EMUs, our major task was to enhance the accuracy and efficiency of the diagnosis of the fault class.

Compared to the standard classification algorithms, an online sequential extreme learning machine (OS-ELM) [11] requires fewer training parameters and fewer manual interventions while providing quick feedback, has better generalization performance without repeatedly adjusting parameters [12], and is widely used in many fields [13–15]. Due to the input weights and hidden biases being assigned randomly, the performance of an OS-ELM may be unstable [16]. According to this problem, this study proposed an ABC-OSELM method, in which the artificial bee colony (ABC) algorithm was developed to look for the optimal input weights and hidden layer biases of an OS-ELM under different numbers of hidden layer nodes.

Most of the standard classification algorithms assume that it costs the same amount to make mistakes for each class. However, samples are not distributed evenly, resulting in low sorting accuracy in the minority class. Over the past few years, a great number of researchers have been dedicated to imbalanced data sorting algorithms. Ömer Faruk Arar combined cost-sensitive learning with the neural network algorithm and creating software for the estimation of defects among imbalanced samples [17]. Piotr Porwik et al. analyzed imbalanced and incomplete medical data using a K-means nearest neighbor (K-NN) classifier that was based on a proper orthogonal decomposition, effectively figuring out the phase of liver fibrosis [18]. By enhancing the sample balance among the training datasets, Ugo Fiore et al. put forward a scheme using a generative adversarial network (GAN) to oversample the minority class to identify frauds with credit cards with more success [19]. Qian et al. [20] designed a global resampling algorithm to manage classification problems that identify the quantity of undersampling and oversampling, and according to experience, it immensely boosts the environmental monitoring function. Jidong Wang et al. proposed a data enhancement method, which reduced the impact of data imbalance, enhanced the generalization performance of the network, and was proved to be effective for the problem of data imbalance in the actual power system [21]. An intelligent bearing fault diagnosis method was proposed by Gu et al. [22] to solve the imbalance problem through vibration experiments under varying conditions using an angular domain resampling, thus indicating that the method was the best method under the same test conditions.

There are two approaches used to solve the sorting problems of imbalanced data in the major class. The first strategy is fundamentally data-based, which involves balancing samples with different classes of undersampling methods and oversampling methods [19–23], such as the random resampling method and the synthetic minority oversampling technique (SMOTE). The other is rooted in algorithms, which involve optimizing classical sorting algorithms, such as integrated learning, cost-sensitive learning [17], single-class learning [24], and active learning; this strategy takes advantage of imbalanced supplemented data to keep the characteristics of the original data distribution such that the classification precision and identification ability is increased [25–27].

In this study, the fault diagnosis model of imbalanced data, which was based on an ABC-OSELM, incorporated imbalanced data resampling, the ABC algorithm, and an OS-ELM, and was used to manage the imbalance characteristics of the axle box bearing state data of a high-speed EMU and to enhance the accuracy and real-time processing of the fault diagnosis.

Two contributions were made in this study. First, the method reconstructed the imbalanced data and extracted the characteristics of data using an undersampling–oversampling combined method; it moderated overfitting problems arising from the majority class samples and had better fault classification performance compared to other standard and classical algorithms. Second, the ABC algorithm was applied to the OS-ELM to search for the global optimum sets of the input weights, hidden layer bias, and the number of hidden layer nodes for the OS-ELM. The simulation results show that the method improved the classification accuracy and speed.

The rest of this paper is organized as follows: Section 2 describes the basic principles of the OS-ELM, ABC algorithm, and the ABC-OSELM fault diagnosis method; Section 3 introduces the main calculation steps of the imbalanced data mixed resampling and the proposed imbalanced axle box bearing fault diagnosis based on the evolutionary OS-ELM; Section 4 shows the application of this method to an axle box bearing fault diagnosis and the results of several experiments; and in Section 5, a summary is stated.

## 2. ABC-OSELM

### 2.1. OS-ELM

The axle box bearing fault diagnosis model is not static but is continuously self-optimized through updated training data. The classical algorithms are very time- and resource-consuming because the new data and the past data have to be retrained together while new data arrives. An OS-ELM introduces the concept of time into the model training, where monitoring data is batched together, which can provide better generalization performance than other popular sequential learning algorithms [11] and keeps the advantages of fast learning speed and simple parameter selection.

#### 2.1.1. Initialization

An extreme learning machine (ELM) [28] is the initialization phase of the OS-ELM, which is a smart algorithm that is applied to a single-hidden-layer feedforward neural network (SLFN). It transforms the task in the tradition of training single-layer parameters to linear equation computations through randomly assigning values to weights between the input layer and the hidden layer, as well as to the bias vector parameters in the hidden layer. The output weights are derived using the minimum norm least-squares solution.

An ELM is a single-hidden-layer feedforward neural network, as Figure 1 shows, which consists of  $N$  input nodes,  $L$  hidden nodes, and  $M$  output nodes.  $x_i$  represents the input vector of the  $i$ th input node, such as the axle box bearing information and working conditions, and  $y_i$  represents the output vector of the  $i$ th output node.

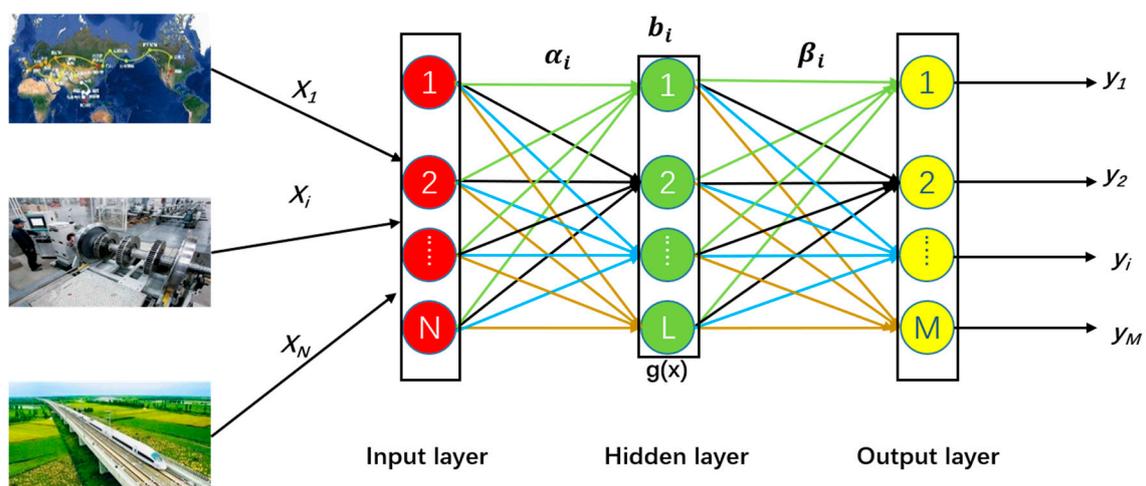


Figure 1. Structure of an extreme learning machine (ELM).

A training dataset  $D$  is given as  $D = \{(x_i, y_i) | (x_i, y_i) \in R^N \times R^M\}$ , where  $x_i$  and  $y_i$  denote an  $N \times 1$  vector and an  $M \times 1$  vector, respectively. For an ELM model with  $L$  hidden nodes, the output can be given as Equation (1):

$$f_L(x) = \sum_{i=1}^L \beta_i G(a_i, b_i, x), \quad x \in R^N, a_i \in R^N, \quad (1)$$

in which the output weight matrix is:

$$\beta_{L \times M} = \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1M} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2M} \\ \vdots & \vdots & \dots & \vdots \\ \beta_{L1} & \beta_{L2} & \dots & \beta_{LM} \end{bmatrix}. \quad (2)$$

where  $\beta_{ij}$  denotes the weight from the  $i$ th hidden node to the  $j$ th output and  $\beta_i$  denotes the output weight vector of the  $i$ th hidden node.

$a_i$  is an input weight vector between the input nodes and the  $i$ th hidden node,  $b_i$  is the bias weight of the  $i$ th hidden node, and  $G(a_i, b_i, x)$  is the output of the  $i$ th hidden node as a function of the input  $x$ . The activation function  $g(a, b, x)$  can be any bounded nonconstant piecewise continuous function. The sigmoid function is usually chosen as an activation function and is shown as follows:

$$g(a, b, x) = \frac{1}{1 + \exp(-(ax + b))}, \quad (3)$$

$$G(a_i, b_i, x) = g(a_i \cdot x + b_i), \quad b_i \in R. \quad (4)$$

A feedforward neural network with an activation function  $g(x)$  can produce a zero-error estimation of the outputs; therefore, there exists  $a_i$ ,  $b_i$ , and  $\beta_i$  that can be substituted into Equation (5):

$$f_L(x_j) = \sum_{i=1}^L \beta_i G(a_i, b_i, x) = y_j, \quad j = 1, \dots, N. \quad (5)$$

Then, Equation (5) can be rewritten as Equation (6):

$$H\beta = Y. \quad (6)$$

In Equation (6),  $H$  denotes the hidden layer output matrix of the ELM:

$$H(a_1, \dots, a_L, b_1, \dots, b_L, x_1, \dots, x_L) = \begin{bmatrix} g(a_1 x_1 + b_1) & \dots & g(a_L x_1 + b_L) \\ \vdots & \dots & \vdots \\ g(a_1 x_N + b_1) & \dots & g(a_L x_N + b_L) \end{bmatrix}_{N \times L}. \quad (7)$$

$Y$  is the matrix form of the target output:

$$Y = [y_1, y_2, \dots, y_M]^T. \quad (8)$$

Due to the minute contribution the random value assignments to the hidden layer parameters  $a_i$  and  $b_i$  makes to the model precision, the output weight matrix  $\beta$  can be described as:

$$\beta = H^\dagger Y, \quad (9)$$

where  $H^\dagger$  is the Moore–Penrose generalized inverse of the hidden layer output matrix  $H$ , which can be solved using an orthogonal projection, iteration, singular value decomposition (SVD), etc. The orthogonal projection method can be used if  $H^T H$  is nonsingular such that  $H^\dagger$  can then be written as  $H^\dagger = (H^T H)^{-1} H^T$ . To elevate the generalization performance and robustness of the solution, a regularization parameter  $C$  is introduced; thus, Equation (9) is further rewritten as:

$$\beta = \left( \frac{I}{C} + H^T H \right)^{-1} H^T Y. \quad (10)$$

SVD has prevailed in recent ELM studies when it comes to deriving  $H^\dagger$ .

If the initial training data are given as  $D_0 = \{(x_i, y_i) | (x_i, y_i) \in R^n \times R^m, i \in [1, N_0]\}$ , then the initial hidden layer output matrix  $H_0$  and output weight vector  $\beta^{(0)}$  is given using an ELM.  $\beta^{(0)}$  can be written as:

$$\beta^{(0)} = P_0 H_0^T Y_0, \quad (11)$$

$$P_0 = (H_0^T H_0)^{-1}, \quad (12)$$

$$Y_0 = [y_1, y_2, \dots, y_{N_0}]^T. \quad (13)$$

### 2.1.2. Online Sequential Learning

Given that the information for the  $(k+1)$ th batch is known, then the information is utilized to calculate the hidden layer output matrix  $H_{k+1}$ ; the output weight vector  $\beta^{(k+1)}$  is given as:

$$\beta^{(k+1)} = \beta^{(k)} + P_{k+1} H_{k+1}^T (Y_{k+1} - H_{k+1} \beta^{(k)}), \quad (14)$$

$$P_{k+1} = P_k - P_k H_{k+1}^T (I + H_{k+1} P_k H_{k+1}^T)^{-1} H_{k+1} P_k. \quad (15)$$

The online sequential learning steps above are repeated and the iterative hidden layer output matrix  $H$  and output weight vector  $\beta$  are updated. In this way, the neural network's classification and generalization performance are reinforced, as well as the fault diagnosis accuracy.

## 2.2. Artificial Bee Colony

The ABC algorithm, which was put forward by Karaboga, is a new type of smart optimization algorithm that imitates the collection behaviors within a swarm of bees [29]. It is easily manipulated and robust with condensed control parameters. The evolution of the solution entails three types of bees: employed bees, onlooker bees, and scout bees. The ABC algorithm realizes multi-population collaborative optimization with high quality and reduced searching time. Meanwhile, the unique random probe mechanism implemented by scout bees effectively prevents becoming stuck at a local optimum. Two main phases are used to build ABC architectures: initialization and optimization.

### 2.2.1. Initialization

First, the scale of the colony  $N$  is established. One employed bee is assigned to each nectar source and equal amounts of onlooker bees and employed bees are maintained. Then, the initialization parameters are set: the maximum cycle number ( $MCN$ ), the quit limit ( $limit$ ), and the upper and lower bounds in the search space.

### 2.2.2. Optimization

#### Employed Bees

- The employed bees should exploit nectar sources and find out the amounts of nectar in food sources. Every one of the nectar sources represents a feasible solution to the problem, and the nectar amounts represent the fitness value of the solution.
- The employed bees then locate a new candidate source away from the previous one and compare the nectar amounts of the two, then choose the richer one. The new candidate source location is generated from its predecessor using Equation (16):

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}). \quad (16)$$

where  $v_{ij}$  denotes the new location, while  $x_{ij}$  denotes the previous location and  $x_{kj}$  is an arbitrary source location in the neighborhood of the previous location, with the indices  $k, i = 1, 2, \dots, N$ , where  $N$  marks the colony scale, and  $j = 1, 2, \dots, D$ , where  $D$  marks the dimensions of the problem.  $\varphi_{ij}$  is a random number following a uniform distribution within  $[-1, 1]$ .

### Onlooker Bees

After exploiting the nectar sources, the employed bees share the information of these sources with onlooker bees. The onlooker bees then select the probability-related source using Equation (17):

$$p_i = \frac{fit_i}{\sum_{n=1}^N fit_n}, \quad (17)$$

where  $fit_i$  is the nectar amount of the  $i$ th source;  $i = 1, 2, \dots, N$ ; and  $N$  marks the colony scale.

The onlooker bees also produce a new candidate source location and calculate the nectar amounts of it, and then select the better source out of the previous source and the candidate, as was mentioned above.

### Scout Bees

Provided that there is no further progress in a particular source after the *limit* times of employed bees' and onlooker bees' exploitation, the evolution of the solution gets stuck at a local optimum, which means that this source or solution should be thrown out of the employed bees' task set. In this situation, the employed bees will then work as scout bees in search of a new arbitrary source location in place of the inferior source to propel the evolution. When this abandoned source is the present optimum, its information will be recorded. The new arbitrary source location is found using Equation (18):

$$x_{ij} = x_{min, j} + rand(0, 1) (x_{max, j} - x_{min, j}), \quad (18)$$

where  $x_{max, j}$  and  $x_{min, j}$  mark the upper and lower bounds of source  $x_{ij}$  in the  $j$ th dimension, respectively.

The optimization stage in step (2) will not cease until it meets the *MCN*, and thus, the best nectar source is obtained.

### 2.3. ABC-OSELM

It has been observed that the performance of OS-ELM depends highly on the chosen set of input weights, hidden layer bias, and the number of hidden layer nodes. OS-ELM may display worse performance in the case of non-optimal parameters. In this study, the ABC algorithm was used to find the optimal set of input weights and hidden layer bias with different numbers of hidden layer nodes for the OS-ELM. The structure of the evolutionary OS-ELM method is shown in Figure 2 and works as follows:

1. Set the chunk size  $P$  and the maximum hidden layer nodes number  $H_{max}$ .
2. Generate the initial population randomly. Each initial solution vector  $u_i$  contains input weights, a hidden layer bias, and the initialization parameters, i.e., *MCN*, *limit*, and upper/lower bounds of the search space need to be established depending on the number of hidden layer nodes:

$$u_i = [w_{11}, w_{12}, \dots, w_{1n}, w_{21}, w_{22}, \dots, w_{2n}, \dots, w_{m1}, w_{m2}, \dots, w_{mn}, b_1, b_2, \dots, b_n], \quad (19)$$

where  $n$  and  $m$  are the numbers of nodes in the input layer and the hidden layer, respectively.

3. For each individual, calculate the output weight matrix  $\beta$
4. Calculate the fitness value of each solution vector.
5. Repeat the employed bees', onlooker bees', and scout bees' search processes until *MCN* cycles are completed.
6. Record the best parameters and update the diagnosis model.

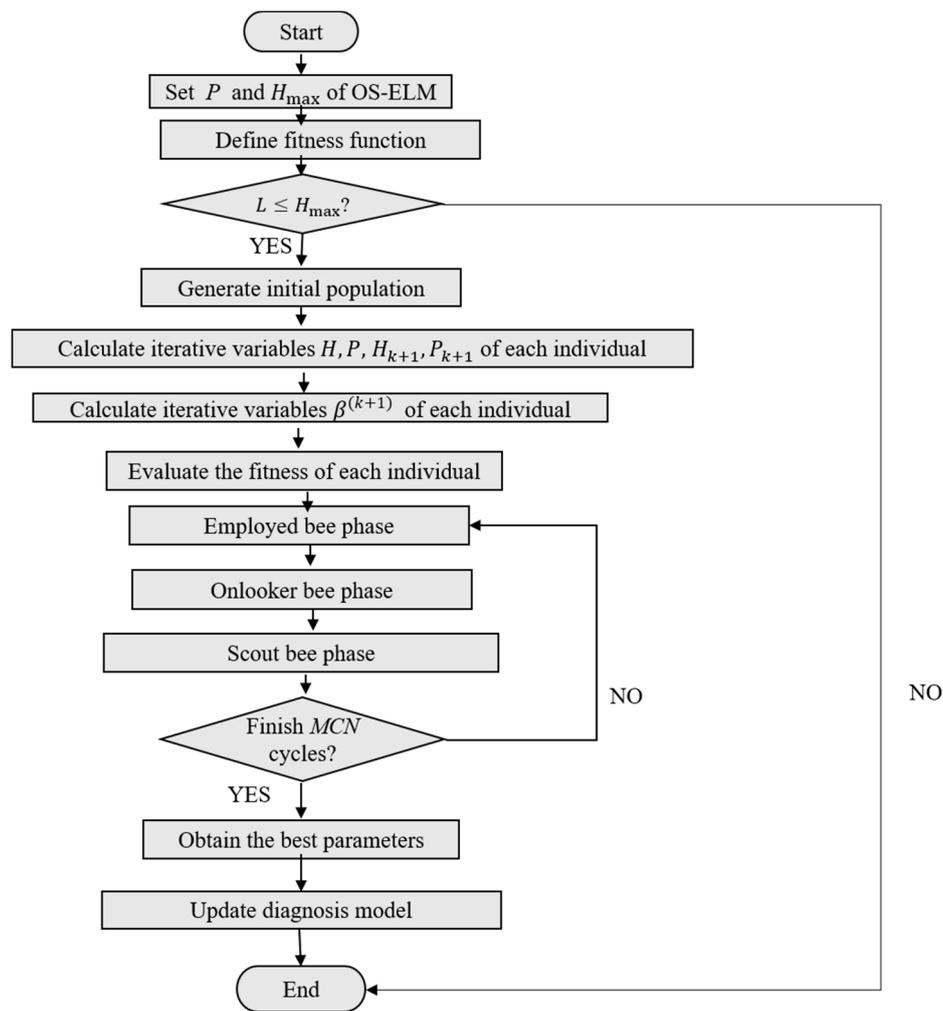


Figure 2. Structure of the evolutionary online sequential (OS)-ELM.

### 3. Imbalanced Axle Box Bearing Fault Diagnosis Based on the Evolutionary OS-ELM

#### 3.1. Imbalanced Data Mixed Resampling

The actual high-speed EMU axle box bearing fault data have an imbalanced feature. While the evolutionary OS-ELM algorithm proposed in Section 2 improves the accuracy of the classification, the sorting accuracy in the minority class for samples with a high proportion of imbalances is not high. In this study, a scheme comprised of an undersampling method, based on the Euclidean distance, and the oversampling K-means SMOTE, based on the clustering distribution, was established to manage the problems of data imbalance. The procedure is presented in Figure 3, which proceeds as follows:

1. The sample set  $D$  made up of historical monitored data of high-speed EMU axle box bearings is classified according to the sample's properties. The minority class sample set is defined as  $S$ . The majority class sample set is defined as  $M$ .
2. Using the resampling empirical Equation (20) proposed in Porwik et al. [18] for the imbalanced data dichotomy, the resampling scales are set as follows:

$$c = -0.097R + 1.428, d = 0.198R + 0.738, \quad (20)$$

where  $c$  is the oversampling scale for the minimum class ( $c \geq 1$ ) and  $d$  is the undersampling scale for the maximum class ( $0 \leq d \leq 1$ ).  $R$  denotes the ratio of the number of the minority class samples to the number of majority class samples.

3. Oversampling of the minority class samples using K-means SMOTE is based on the clustering distribution. In the beginning, filter the targeted clusters and save the ones that are numerically dominant within the minority class samples by generating  $k$  clusters using K-means algorithms. Then, distribute the quantity of the clustering samples while preferring the sparse groups. To prevent undesirable noise, only data relating to key indicators are oversampled, while the rest are kept in their original form to highlight the more informative and significant indicators. Finally, new samples are generated in each selected cluster with SMOTE and added to the minority class sample set  $M$ .
4. Undersample the majority class samples in non-boundary domains with the method based on the Euclidean distance. First, locate the center of the majority class sample set  $M$  by calculating the distances between all the sample points to the center and ranking them in order. From the furthest sample point to the center, delete samples in  $M$  according to the undersampling magnification until  $M$  is quantitatively balanced with the minority class samples. This approach identifies the classification boundary and leaves out the insignificant and distant samples from the classification boundary in the majority class, which ensures that the training dataset is composed of safe samples.
5. Derive a new sample set  $D_{\text{new}}$  from the updated minority class samples and majority class samples.

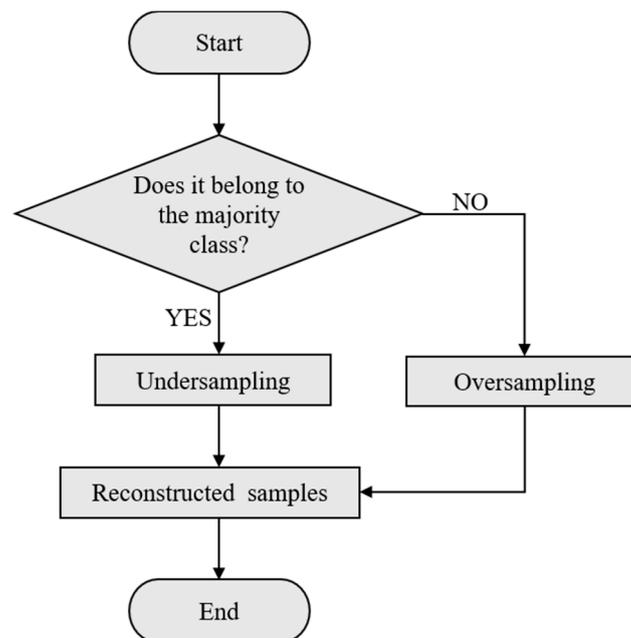


Figure 3. Structure of the imbalanced data resampling.

### 3.2. Diagnosis Process

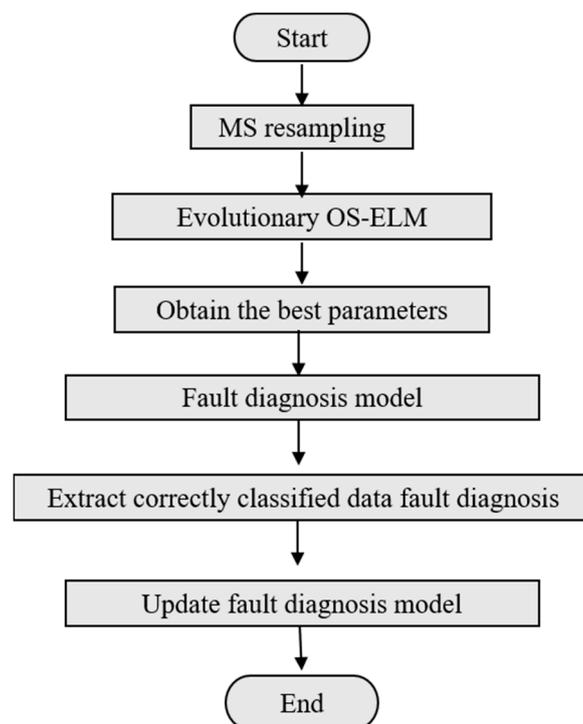
The mixed resampling online sequential extreme learning machine based on the artificial bee colony algorithm (MS-ABC-OSELM) extracts more important information from features with the mixed resampling techniques proposed in Section 3.1 and applies the evolutionary OS-ELM algorithm proposed in Section 2 to search for global optimum sets of the input weights, hidden layer bias, and the number of hidden layer nodes for the OS-ELM. It inherits stability and fast processing speed from the OS-ELM and optimization characteristics from the ABC algorithm, which guarantees high learning accuracy and ensures the least amount of training error.

The fault diagnosis with the MS-ABC-OSELM is modeled, as exhibited in Figure 4, and proceeds as follows:

1. Resample the imbalanced data by extracting the required high-speed EMU axle box bearing data and dividing them into minority class samples and majority class samples. Determine

the resampling scale of the imbalanced data and reconstruct the training dataset with mixed resampling methods to acquire new training samples.

2. Use the evolutionary OS-ELM algorithm to search for the global optimum sets of the input weights and hidden layer bias under different numbers of hidden layer nodes for the OS-ELM, as discussed in Section 2.
3. Report the best combination of variables and note down the optimized input weights and hidden layer bias with the number of hidden layer nodes.
4. Establish the optimized diagnosis model.
5. Update the fault diagnosis model using the accurately diagnosed historical data as the online sequential sample. These online sequential samples are then reconstructed to a balanced distribution. Compute  $H_{k+1}$  and  $\beta^{(k+1)}$ , then update the diagnosis model using the OS-ELM. Repeat step (5) and optimize the high-speed EMU axle box bearing fault diagnosis model through adaptive learning.



**Figure 4.** Flowchart of the mixed resampling online sequential extreme learning machine based on the artificial bee colony algorithm (MS-ABC-OSELM) fault diagnosis modeling.

#### 4. Experiments and Analysis

All computation experiments were conducted in an identical system environment. A PC installed with a 2.60 GHz CPU, 8.00 GB RAM, Windows 10 system, and Python 3.7 was sufficient for the experiments in the study.

##### 4.1. Dataset

Samples were taken from the routine monitoring data of 20 homomorphic trains over 4 months of service, including onboard data of axle box bearings (bearings were manufactured by NTN) and ground-detected data. The selected data were representative in that the sampled trains' routes covered a broad span of districts and complicated environments. To monitor the axle box bearings' state, data of the trains, working conditions, and temperatures of axle box bearings containing 10 dimensions, as presented in Table 1, were analyzed. The temperature of each axle box bearing was recorded using two channels. The preorder temperature marks the tested bearing temperature 10 min prior.

For high-speed EMUs, each bogie has two axles and there is an axle box bearing on each end of each axle, wherefore the installation position of the axial box bearings on the EMU is symmetrical. Considering the symmetrical effect, the temperature of the coaxial bearing, which is on the symmetrical side, is selected as one of the properties. Furthermore, the temperature of the bearing on the same side, which is on the other axle of the bogie, was also selected. Preconditioning included the noise reduction and normalization of the data. Define class 1 as the normal state and class 2 the abnormal state. Samples in class 1 were numerous and centralized, thus belonging to the majority class, while samples in class 2 were sparse and naturally went into the minority class. The number ratio of the minority class samples to the majority class samples was 115:345, showing an obvious imbalanced characteristic in the distribution of the sample states. The majority class was divided into the original training set and testing set at a 1:1 ratio via randomly sampling. Furthermore, the minority class was the same. Therefore, the training and testing sets had the same number of samples and the same imbalance ratio.

**Table 1.** Properties of the input.

| No. | Property                               |
|-----|--|
| 1   | Environment temperature                |
| 2   | Running speed                          |
| 3   | Channel 1 temperature                  |
| 4   | Channel 2 temperature                  |
| 5   | Temperature of the coaxial bearings    |
| 6   | Temperature of the same-sided bearings |
| 7   | Preorder temperature                   |
| 8   | Mile                                   |
| 9   | Acceleration                           |
| 10  | Load                                   |

#### 4.2. Assessment Criteria

Accuracy, which is the ratio of correctly classified samples to the total, cannot perfectly reflect the function of the classification model for imbalanced data. Consequently, this study used a far better method, namely, the G-mean, to assess the classification performance; the G-mean is calculated as the geometric mean of the minority class precision and majority class precision. Furthermore, the F1-measure of the minority class was also used for verification.

Table 2 exhibits the confusion matrix of the binary classification imbalanced data, in which *TP* denotes the true classification of the positive samples, *TN* denotes the true classification of the negative samples, *FP* denotes the false classification of the positive samples, and *FN* denotes the false classification of the negative samples.

**Table 2.** Confusion matrix of the binary classification imbalanced data.

|                        | Predicted Positive Sample | Predicted Negative Sample |
|------------------------|---------------------------|---------------------------|
| Actual positive sample | <i>TP</i>                 | <i>FN</i>                 |
| Actual negative sample | <i>FP</i>                 | <i>TN</i>                 |

Accuracy is defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%. \quad (21)$$

The G-mean is defined as follows:

$$\text{G-mean} = \sqrt{\frac{TN}{TN + FP} \times \frac{TP}{TP + FN}}. \quad (22)$$

The F1-measure is defined as follows:

$$\text{F1-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \times 100\%. \quad (23)$$

where Recall and Precision are defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\%, \quad (24)$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%. \quad (25)$$

### 4.3. Analysis

A support vector machine (SVM), ELM, OS-ELM, ABC-OSELM, and MS-ABC-OSELM were contrastively tested and evaluated in the study concerning the function of imbalanced data classification. An SVM is robust regarding imbalanced data classification and was therefore chosen as a reference for comparison with the others. The kernel function of the SVM took the shape of a radial basis function (RBF) with the kernel parameter  $\sigma$  and penalty parameter  $g$  set as 0.8 and 21, respectively, through the cross-validation searching method. The sigmoid function was collectively chosen as an activation function for the ELM, OS-ELM, ABC-OSELM, and MS-ABC-OSELM models. The initial training dataset contained 100 samples and the batch number was set as 100 to simulate a continuous dataflow for the OS-ELM, ABC-OSELM, and MS-ABC-OSELM.

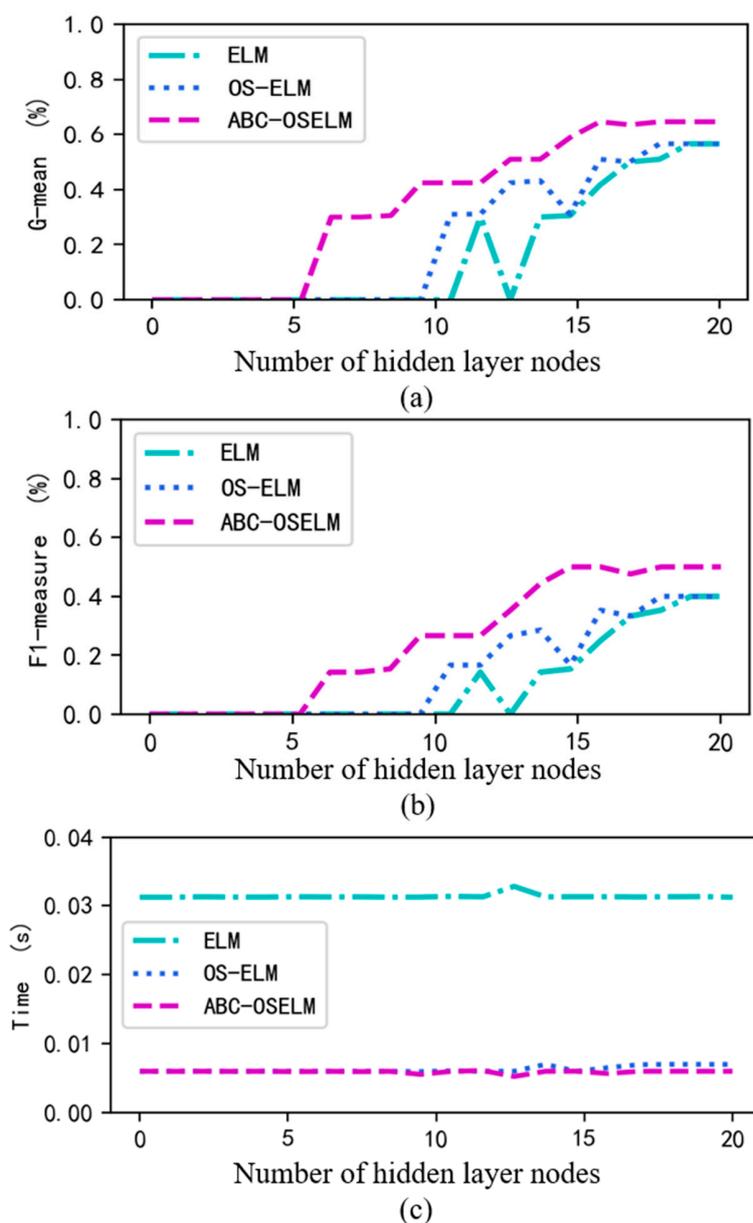
#### 4.3.1. Algorithm Efficiency Analysis on the Original Dataset

To verify the classification effect of the ABC-OSELM, we compared it with other existing methods. The G-mean, F1-measure, and running time were used as the evaluation metrics to analyze the experimental results.

Figure 5 demonstrates the G-mean, F1-measure, and operating times were affected by the number of hidden layer nodes for the ELM, OS-ELM, and ABC-OSELM. Figure 5a,b show that as the number of hidden layer nodes increased, the classification performance increased to a certain level, and then stabilized with the number of hidden layers increasing. The proposed method, ABC-OSELM, found the solution quicker and the solution was better compared to other methods. Figure 5a shows that the G-mean for the three algorithms also increased with the number of hidden layer nodes. It should be noted that the G-mean for ELM and OS-ELM were 0 when the number of hidden layer nodes was less than 10 nodes, while the ABC-OSELM identified minority class samples after six nodes. The G-means of the ELM and OS-ELM were always lower than that of the ABC-OSELM. The trend of the F1-measure was consistent with G-mean, which is shown in Figure 5b. This means that the ELM and OS-ELM needed more hidden layer nodes to identify the minority class samples. In other words, these two algorithms required more onboard computing resources and had higher energy consumption. From Figure 5c, we can see that the testing time of the three algorithms did not fluctuate significantly with the number of hidden layer nodes. This was because all three models were based on the ELM model, which has the characteristics of fast calculation speed. The testing time was mostly used for data processing, reading, and optimization. The calculation time of the hidden layer was so short that the number of hidden layers had no significant effect on the training and testing time. The training time of the ABC-OSELM was much longer than those of the ELM and OS-ELM because of the optimization. Since the training time did not affect the use of the model, this study did not consider the training time as a factor that was used for comparing performance.

The performances of each classification model are given in Table 3. Using the original dataset for classification, the testing time of the ABC-OSELM algorithms was much shorter than the time for the SVM and ELM, which was 1.17 times that of the OS-ELM. It had the highest classification performance compared to the other three machine learning algorithms, both for the training dataset and the test

dataset. The ABC-OSELM method had displayed high classification performance for the minority class data.



**Figure 5.** The evaluation metrics of the ELM, OS-ELM, and ABC-OSELM changed as a function of the number of hidden layer nodes: (a) G-mean, (b) F1-measure, and (c) time.

**Table 3.** Comparison of four classification models using the original dataset.

| Evaluation Metrics  | SVM <sup>1</sup> | ELM   | OS-ELM | ABC-OSELM    |
|---------------------|------------------|-------|--------|--------------|
| Testing time        | 0.582            | 0.031 | 0.007  | <b>0.006</b> |
| Testing G-mean      | 0.633            | 0.566 | 0.566  | <b>0.646</b> |
| Testing F1-measure  | 0.476            | 0.4   | 0.4    | <b>0.5</b>   |
| Training G-mean     | 0.762            | 0.594 | 0.607  | <b>0.781</b> |
| Training F1-measure | 0.679            | 0.464 | 0.48   | <b>0.708</b> |
| Optimum node        | /                | 19    | 18     | <b>15</b>    |

<sup>1</sup> The support vector machine (SVM) model has no hidden layer node.

Figure 6 compares the classification effects of different algorithms more intuitively. The ABC-OSELM algorithm, using the best parameters set, had a higher classification performance and running efficiency than the other three classifiers using the original dataset.

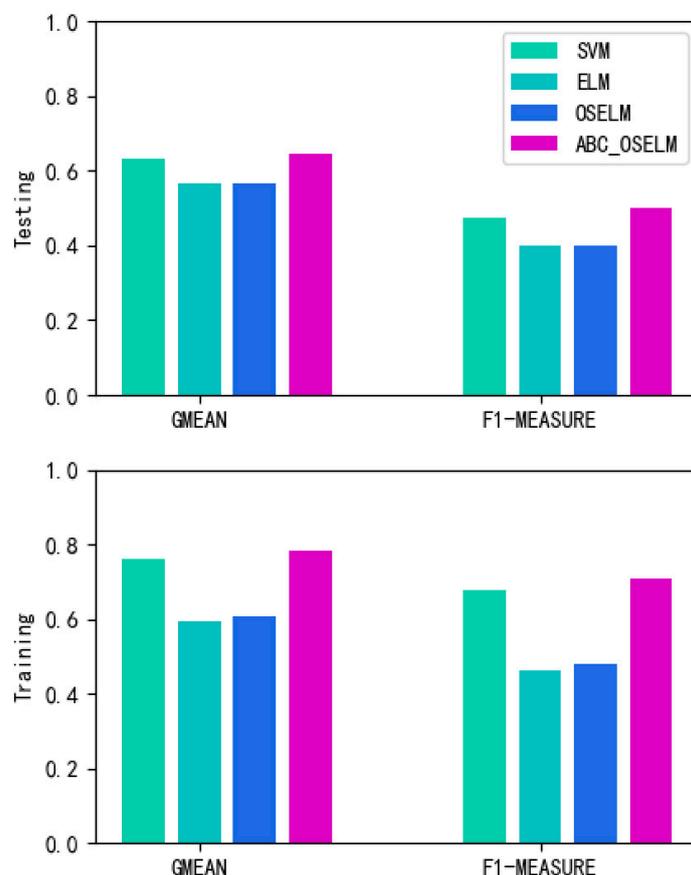


Figure 6. Testing and training evaluation metrics histogram.

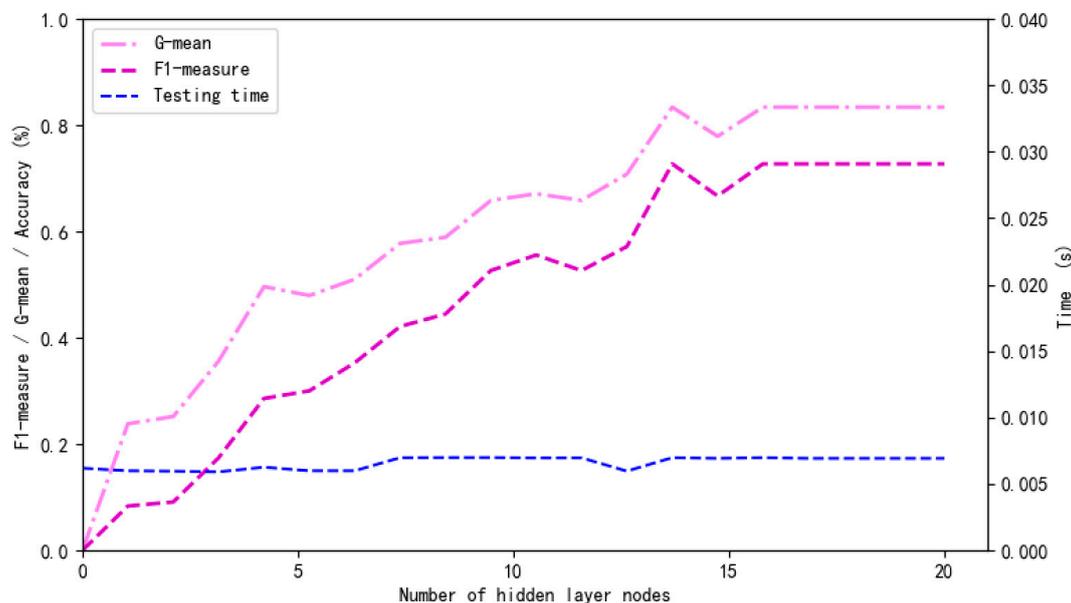
#### 4.3.2. Algorithm Analysis on the Mixed Resampling Dataset

To demonstrate the effectiveness of the approach of imbalanced data mixed resampling, the training dataset was mixed resampled and the test dataset was unchanged. The scale parameter of the mixed resampling training dataset was determined using the resampling empirical Equation (20). The imbalance ratio of the mixed resampling training dataset was changed from 3:1 to 1.59:1. We carried out a series of experiments on the mixed resampling dataset.

Figure 7 shows that when the hidden layer nodes number increased, the G-mean and F1-measure of the MS-ABC-OSELM increased to a certain level and then stabilized, and the testing time fluctuated between 0.005 s and 0.007 s, and did not fluctuate significantly with the number of hidden layer nodes. The MS-ABC-OSELM identified minority class samples after the number of hidden layer nodes was two, which was four less than the ABC-OSELM.

In the experiment, the SVM, ELM, OS-ELM, and ABC-OSELM performed the base learning algorithms separately. Tables 4 and 5 list the comparison results of G-means and F1-measures of the experiments between the resampling dataset and the original dataset. The performances show that the resampling dataset obtained significantly better performances in terms of G-means and F1-measures than the original dataset. It was shown that the proposed method of imbalanced data mixed resampling effectively improved the imbalance index. Comparing the four algorithms on the two datasets, we found that the MS-ABC-OSELM performed the best in terms of both the G-mean and the F1-measure. Its G-mean was 18.7%, 5.4%, 11.2%, and 6.9% better than the ABC-OSELM, MS-SVM,

MS-ELM, and MS-OS-ELM, while the F1-measure of the MS-ABC-OSELM method outperformed the four algorithms by 22.7%, 6%, 12.7%, and 9.1%, respectively. The numerical results show that the mixed sampling process could greatly improve the identification rate of the minority class data.



**Figure 7.** The evaluation metric changes of MS-ABC-OSELM as a function of the number of hidden layer nodes.

**Table 4.** G-mean comparison results for the four classification models.

| Evaluation Metrics | SVM <sup>1</sup> | ELM   | OS-ELM | ABC-OSELM    |
|--------------------|------------------|-------|--------|--------------|
| Resampling dataset | 0.779            | 0.721 | 0.764  | <b>0.833</b> |
| Original dataset   | 0.633            | 0.566 | 0.566  | 0.646        |

<sup>1</sup> The SVM model has no hidden layer node.

**Table 5.** F1-measure comparison results for the four classification models.

| Evaluation Metrics | SVM <sup>1</sup> | ELM | OS-ELM | ABC-OSELM    |
|--------------------|------------------|-----|--------|--------------|
| Resampling dataset | 0.667            | 0.6 | 0.636  | <b>0.727</b> |
| Original dataset   | 0.476            | 0.4 | 0.4    | 0.5          |

<sup>1</sup> SVM model has no hidden layer node.

From the performances above, it can be seen that the MS-ABC-OSELM model built in the study moderated the overfitting problems arising from the majority class samples and had a significant advantage when identifying minority class samples. Furthermore, the MS-ABC-OSELM still maintained a fast calculation speed and lower hidden layer nodes, which meant lower energy consumption for the onboard computing of high-speed EMUs. Put simply, MS-ABC-OSELM made the online axle box bearing fault diagnosis during operation more efficient and accurate.

## 5. Conclusions

Targeted at high-speed EMU axle box bearing state monitoring, in this study, we designed an evolutionary OS-ELM fault diagnosis model that specialized in imbalanced data. Considering that the axle box bearing monitoring data of high-speed EMUs in service is imbalanced, a mixed resampling method was employed to reconstruct the imbalanced data and to extract the characteristics. The ABC algorithm was utilized to globally optimize the input weights and hidden layer bias with

different numbers of hidden layer nodes in the OS-ELM to establish an increment-based diagnosis model. Put together, these schemes (MS-ABC-OSELM) were used to accomplish the goal of axle box bearing state online classification and model adaptive optimization. By testing historically monitored data of operating high-speed EMU axle box bearings, it was demonstrated that the advantages of MS-ABC-OSELM over other classical algorithms were that it was faster at detecting and more accurately classified the minority class data. As a result, the proposed evolutionary MS-ABC-OSELM fault diagnosis model proved to be effective and could diagnose the axle box bearing states of high-speed EMUs online.

**Author Contributions:** Conceptualization, W.H.; data curation, W.H.; formal analysis, W.H.; investigation, W.H.; methodology, W.H.; project administration, F.L.; software, W.H.; supervision, F.L.; validation, W.H.; writing—original draft, W.H.; writing—review and editing, F.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Key R&D Plan of China (grant number 2018YFB1201704).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhao, B.; Yuan, Q.; Zhang, H. A Real-Time Fault Early Warning Method for a High-Speed EMU Axle Box Bearing. *Sensors* **2020**, *20*, 823.
2. Qin, N.; Jin, W.; Huang, J.; Li, Z.M. Ensemble empirical mode decomposition and fuzzy entropy in fault feature analysis for high-speed train bogie. *Control Theory Appl.* **2014**, *31*, 1246–1251.
3. Huang, X.W.; Liu, C.M.; Feng, C.; Wang, X.; Yu, F. Failure diagnosis system for CRH3 electrical multiple unit. *Comput. Integr. Manuf. Syst.* **2010**, *16*, 2311–2318.
4. Zhou, B.; Xu, W. Improved parallel frequent pattern growth algorithm in EMU's fault diagnosis knowledge mining. *Comput. Integr. Manuf. Syst.* **2016**, *22*, 2450–2457.
5. Li, P.; Zheng, J. Fault Diagnosis Based on Multichannel Vibration Signal for Port Machine Trolley Wheel Bearings. *Bearing* **2007**, *9*, 35–38.
6. Liu, J.; Zhao, Z.; Zhang, G.; Wang, G.M.; Meng, S.; Ren, G. Research on fault diagnosis method for bogie bearings of Metro vehicle. *J. China Railw. Soc.* **2015**, *37*, 30–36.
7. Feng, H.; Yao, B.; Gao, Y.; Wang, H.; Feng, J. Imbalanced data processing algorithm based on boundary mixed sampling. *Control Dec.* **2017**, *10*, 1831–1836.
8. Wuchu, T.; Minjie, W.; Guangdong, C.; Yuchao, S.; Li, X. Analysis on temperature distribution of failure axle box bearings of high-speed train. *J. China Railw. Soc.* **2016**, *38*, 50–56.
9. Yin, H.X.; Wang, K.; Zhang, T.Z. Fault Prediction Based on PSO—BP Neural Network About Wheel and Axle Box of Bogie in Urban Rail Train. *Complex Syst. Complex. Sci.* **2015**, *12*, 97–103.
10. Liu, J.; Zhao, Z.; Ren, G. An intelligent fault diagnosis method for bogie bearings of train based on wavelet packet decomposition and EEMD. *J. China Railw. Soc.* **2015**, *37*, 41–46.
11. Liang, N.Y.; Huang, G.B.; Saratchandran, P.; Sundararajan, N. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Trans. Neural Netw.* **2006**, *17*, 1411–1422. [[CrossRef](#)] [[PubMed](#)]
12. Jia, P.; Wu, Q.; Li, L. Objective-driven dynamic dispatching rule for semiconductor wafer fabrication facilities. *Comput. Integr. Manuf. Syst.* **2014**, *20*, 2808–2813.
13. Hussain, T.; Siniscalchi, S.M.; Lee, C.C.; Wang, S.S.; Tsao, Y.; Liao, W.H. Experimental study on extreme learning machine applications for speech enhancement. *IEEE Access* **2017**, *5*, 25542–25554. [[CrossRef](#)]
14. Fan, J.M.; Fan, J.P.; Liu, F. A Novel Machine Learning Method Based Approach for Li-Ion Battery Prognostic and Health Management. *IEEE Access* **2019**, *7*, 160043–160061. [[CrossRef](#)]
15. Lu, S.; Lu, Z.; Yang, J.; Yang, M.; Wang, S. A pathological brain detection system based on kernel based ELM. *Multimed. Tools Appl.* **2018**, *77*, 3715–3728. [[CrossRef](#)]
16. Zhu, Q.Y.; Qin, A.K.; Suganthan, P.N.; Huang, G.B. Evolutionary extreme learning machine. *Pattern Recognit.* **2005**, *38*, 1759–1763. [[CrossRef](#)]
17. Arar, Ö.F.; Ayan, K. Software defect prediction using cost-sensitive neural network. *Appl. Soft Comput.* **2015**, *33*, 263–277. [[CrossRef](#)]

18. Porwik, P.; Orczyk, T.; Lewandowski, M.; Cholewa, M. Feature projection k-NN classifier model for imbalanced and incomplete medical data. *Biocybern. Biomed. Eng.* **2016**, *36*, 644–656. [[CrossRef](#)]
19. Fiore, U.; De Santis, A.; Perla, F.; Zanetti, P.; Palmieri, F. Francesca Perla Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Inf. Sci.* **2019**, *479*, 448–455. [[CrossRef](#)]
20. Qian, Y.; Liang, Y.; Li, M.; Feng, G.; Shi, X. A resampling ensemble algorithm for classification of imbalance problems. *Neurocomputing* **2014**, *143*, 57–67. [[CrossRef](#)]
21. Wang, J.; Xu, Z.; Che, Y. Power Quality Disturbance Classification Based on DWT and Multilayer Perceptron Extreme Learning Machine. *Appl. Sci.* **2019**, *9*, 2315. [[CrossRef](#)]
22. Gu, Y.; Zeng, L.; Qiu, G. Bearing Fault Diagnosis with Varying Conditions using Angular Domain Resampling Technology, SDP and DCNN. *Measurement* **2020**, *156*, 10761. [[CrossRef](#)]
23. Piri, S.; Delen, D.; Liu, T. A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets. *Decis. Support Syst.* **2018**, *106*, 15–29. [[CrossRef](#)]
24. García, V.; Sánchez, J.S.; Mollineda, R.A. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowl. Based Syst.* **2012**, *25*, 13–21. [[CrossRef](#)]
25. Xiong, B.; Wang, G.; Deng, W. Under-Sampling method based on sample weight for imbalanced data. *J. Comput. Res. Dev.* **2016**, *53*, 2613–2622.
26. Gu, X.; Jiang, Y.; Wang, S. Zero-order TSK-type fuzzy system for imbalanced data classification. *Acta Autom. Sin.* **2017**, *43*, 1773–1788.
27. Ju, Z.; Cao, J.; Gu, H. A fuzzy support vector machine algorithm for imbalanced data classification. *J. Dalian Univ. Technol.* **2016**, *56*, 252–531.
28. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst.* **2012**, *42*, 513–529.
29. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).