

Article

Incremental Single-Class Fault Detection and Diagnosis Method for Rolling Bearings Based on OS-ELM

Huijuan Hao ^{1,2,*}, Yuanyuan Zhao ^{1,2} , Yu Chen ^{1,2} , Yu Zhang ^{1,2} and Dan Wang ^{1,2}

¹ Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China

² Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan 250014, China

* Correspondence: haohj@sdas.org

Abstract: Aiming at the problem of current equipment fault diagnosis models based on deep learning being unable to automatically identify new class faults according to the updated fault data, in this paper we propose an incremental single-class fault diagnosis method based on an online sequential extreme learning machine (OS-ELM). In addition to detecting new types of faults, this method can perform class-incremental learning based on new-class fault data, treating the new-class faults as known faults for ongoing fault detection and diagnosis tasks. This approach first constructs a feature extraction network with a dual-encoder structure to extract data features. Subsequently, the extracted features are used to build a fault diagnosis network based on OS-ELM, where the novelty of new batches of data is determined by the update magnitude of OS-ELM. When a new-class fault is detected, a new OS-ELM representing the current new class is constructed using the new batch of data and added to the fault diagnosis network, thereby achieving incremental model updates. The proposed method is validated through experiments on the CWRU dataset and MFPT dataset. The results demonstrate that the accuracy of this method on the CWRU dataset is 99.62%, while on the MFPT dataset it reaches 98.80%. Compared to other incremental single-class models, this method exhibits excellent fault recognition and diagnosis capabilities.

Keywords: novelty detection; fault diagnosis; deep learning; incremental learning



Citation: Hao, H.; Zhao, Y.; Chen, Y.; Zhang, Y.; Wang, D. Incremental Single-Class Fault Detection and Diagnosis Method for Rolling Bearings Based on OS-ELM. *Electronics* **2023**, *12*, 4099. <https://doi.org/10.3390/electronics12194099>

Academic Editor: Mohamed Benbouzid

Received: 30 August 2023

Revised: 26 September 2023

Accepted: 28 September 2023

Published: 29 September 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Rolling bearings are crucial components of rotating machinery. However, due to complex working conditions and harsh environments they can easily become damaged and malfunction. The health condition of these bearings can significantly impact the overall performance of the machinery. In the event of a malfunction, the consequences can range from minor reductions in production efficiency and delays in production schedules to severe and detrimental production accidents. Therefore, effective monitoring and diagnosis of the operational state of bearings during the operation of mechanical equipment holds significant importance in maintaining normal equipment operation [1].

In recent years, with the substantial improvement of computer performance, fault diagnosis methods based on deep learning have flourished. Compared with traditional methods that rely on manual feature extraction, deep learning models have powerful deep-level feature extraction capabilities, and have been widely used in bearing condition monitoring and fault classification tasks of rotating machinery [2]. Currently, a large number of fault diagnosis methods have achieved high accuracy in classification. However, these methods often assume that all fault states are known, overlooking the detection of unknown faults. When new class faults occur in the equipment, these methods lack the capability to detect such faults, and can only erroneously classify them into the initially known fault types. The detection of new-class faults mainly involves anomaly detection

and novelty detection. Anomaly detection falls under the category of binary classification, primarily distinguishing abnormal data from normal data. In [3], the authors proposed a hybrid deep learning framework for fault detection by combining a deep de-noising autoencoder and one class support vector machine (OC-SVM) capable of distinguishing faulty data from normal data. However, this method is too limited, as it only detects faults without considering their occurrence history, which is unfavorable for analyzing the causes of faults and finding solutions in industrial applications. Novelty detection is a multi-class problem, primarily involving the distinguishing multiple new class faults from data of various known fault types. This implies that the model is trained solely on data consisting of known fault types, then determines whether the test data belong to known fault types or unknown faults. In [4], the authors employed a convolutional autoencoder network to reconstruct samples and set a threshold based on the reconstruction error to achieve novelty detection. While this approach can detect new faults and classify known faults simultaneously, it cannot differentiate between different kinds of new faults, and is unable to represent and distinguish faults in detail. These limitations hinder its practical application. Additionally, due to the complex operating environment of rotating machinery, in practical engineering applications training data are usually accumulated as the equipment operates. In order to ensure normal operation of the fault diagnosis system, it is often required that the model can continuously learn new knowledge from new data, has the ability to update the historical model, continuously optimizes the current model, and adapts to external data changes, which is referred to as incremental learning. In the face of increasingly complex and intelligent mechanical equipment, constructing fault diagnosis models with incremental learning capabilities can address the challenges posed by current deep learning methods, which struggle to automatically update and adapt to changing data. This has significant implications for saving storage space, reducing diagnostic time, and improving fault diagnosis efficiency.

Therefore, the research motivation of this method can be summarized by the following three aspects:

- (1) Existing fault diagnosis methods rarely involve the detection of new types of faults, and many fault diagnosis models do not have the ability to classify known faults and detect unknown faults at the same time. Even if when methods for detecting new classes can detect such classes, they cannot further distinguish between different new classes of faults.
- (2) Most current deep learning models do not have the ability to automatically update; as the models cannot be updated according to the arrival of new data, they lack the ability to learn new knowledge and perform new diagnostic tasks.
- (3) Existing fault classification methods often require a large number of labeled samples for training, and there are few studies on the identification and diagnosis of new types of faults under unsupervised conditions.

To meet the requirements of fault diagnosis outlined above, in this paper we propose an incremental single-class fault recognition and diagnosis method based on an online sequential extreme learning machine (OS-ELM). Initially, this method combines the powerful feature extraction capability of a convolutional neural networks (CNN) with the unsupervised learning ability of an autoencoder to construct a feature extraction network with unsupervised learning capability. An encoder is added to this network and the loss function is optimized, allowing the network to better capture the data distribution in the latent vector space, thereby further enhancing the network's feature extraction capability. Then, the extracted features are input into the OS-ELM network for fault identification and diagnosis; at this point, it can be determined whether a fault is a new or known fault according to how much the newly collected data have modified the OS-ELM. When all OS-ELMs indicate a new fault, a new OS-ELM is trained using the newly collected data. In cases where a known fault is detected, a comparison is made with each OS-ELM to identify the one with the least amount of modification. The fault type represented by that specific OS-ELM is then diagnosed accordingly. When a certain type of fault data reaches a certain amount, these data can be used to update the existing OS-ELM to enhance

its fault condition representation capability. The proposed method achieves an average diagnostic accuracy of 99.62% on the Case Western Reserve University (CWRU) bearing dataset, surpassing OC-SVM-based diagnostic methods by 17.55% and Isolation Forest (IF) fault diagnosis methods by 7.29%. Validation experiments were conducted on the Mechanical Fault Prevention Technology Association (MFPT) dataset, where our method achieved a diagnostic accuracy of 98.80%.

In summary, the main contributions of this paper are as follows:

(1) The design of a convolutional autoencoder network with dual encoders for feature extraction from fault data. The network's feature extraction capability is enhanced through unsupervised learning and dual-encoder loss constraints, effectively reducing the reconstruction error of normal data and increasing the reconstruction error of anomalous data, thereby improving the performance of new-class fault detection.

(2) The combination of the feature extraction network and OS-ELM network allows fault types to be judged based on the magnitude of OS-ELM updates using new fault data. This simultaneously achieves the detection of new faults and the classification of known faults. With the incremental learning capability of OS-ELM, the approach enables incremental single-class fault recognition, enabling continuous model optimization to adapt to changes in external data.

(3) Through comparison with single-class fault diagnosis methods based on OC-SVM and diagnosis methods based on IF, we demonstrate that the proposed method outperforms other fault diagnosis methods in terms of new class recognition and fault diagnosis, validating the effectiveness of the proposed method.

The rest of this paper is organized as follows. Section 2 introduces the relevant work in the field of mechanical equipment fault diagnosis. Section 3 provides a detailed explanation of the method and principles proposed in this paper. Section 4 presents the experimental validation. Finally, Section 5 provides the conclusion.

2. Related Work

Mechanical equipment fault diagnosis methods can generally be categorized into three types: qualitative analysis methods, mathematical modeling methods, and data-driven methods [5]. Among them, qualitative analysis methods require rich industrial experience and knowledge, while mathematical modeling methods rely on the physical characteristics and linear rules when faults occur. Therefore, they are not suitable for complex industrial equipment in today's context [6]. Data-driven methods are based on monitoring data from equipment, and can obtain feature variables that are more conducive to model recognition or possess physical significance through feature engineering or extraction. These characteristic variables can reflect the operating state of the equipment more effectively and conveniently. In [7], a state recognition model trained using these characteristic variables was able to effectively evaluate the equipment state quantitatively without relying on precise physical models and extensive prior knowledge, and made good research progress in the field of fault diagnosis.

In deep learning, CNNs [8] have excellent performance in feature extraction, especially for image recognition tasks. Their sparse connection and weight-sharing characteristics greatly reduce the parameters required by the model, thereby realizing improved model performance. In recent years, CNN-based methods have almost swept the field of fault diagnosis, achieving promising results. Zhao et al. [9] constructed a normalized CNN model and employed it for fault diagnosis in different operating conditions and on imbalanced datasets. Their results demonstrated that this method achieves high accuracy compared to traditional machine learning methods. Xie et al. [10] proposed a fault diagnosis model based on the continuous wavelet transform (CWT) and a two-dimensional convolutional neural network. Vibration signals were transformed into two-dimensional time–frequency images using CWT, followed by feature extraction through the two-dimensional CNN. Finally, fault classification was conducted using fully connected layers and a softmax classifier. On a dataset of ball screw faults, this method attained an accuracy of 99.67%.

Jiao et al. [11] conducted a comprehensive review of the application of CNNs in machine fault diagnosis, offering detailed insights into CNN-based fault diagnosis algorithms. In addition, these deep learning methods have been applied in other fields, such as prediction of mechanical equipment life [12,13]. Zhang et al. [14] proposed an integrated multi-head dual sparse self-attention network based on an improved transformer and a comprehensive logarithmic-based sparse strategy in MLSN to predict the service life of machinery, achieving high accuracy on the aircraft turbofan engine dataset. As the complexity of modern industrial equipment increases, the amount of data that can be collected is increasing exponentially. Because most recorded data in actual operations are unlabeled, labeling which fault category the data belong to is a very large project [15]. Consequently, new requirements for intelligent diagnosis of industrial equipment have emerged, namely, diagnosing equipment under unsupervised conditions. As a kind of unsupervised representation learning, autoencoders [16] can monitor the learning effect without labels by measuring the reconstruction error between input and output. Their structure consists of an encoder-decoder network in which the dimensionality of the feature representation layer is smaller than that of the input layer, usually as a compressed nonlinear representation of the input data, which can achieve the purpose of dimensionality reduction [17]. Principi et al. [18] proposed an unsupervised motor fault detection method based on a deep autoencoder. Utilizing the unsupervised learning nature of autoencoders, their model was trained using unlabeled data and a threshold was set based on the reconstruction error. This threshold was then used to differentiate between normal and faulty data. Their results indicated that the detection accuracy of this method surpasses other machine learning approaches. In practical applications of deep learning, researchers have widely employed the encoder-decoder network structure for autoencoders, often in combination with other methods, to enhance fault diagnosis capabilities. Chen et al. [19] proposed a convolutional autoencoder network model combining a CNN and autoencoder for the task of diagnosing gearbox faults. Their model performed well compared to other common deep learning methods. Yu et al. [20] used a convolutional autoencoder to learn effective features for multivariate fault diagnosis and optimized the loss function to improve the feature extraction ability of the network. With the continuous development of deep neural networks, in order to solve the unsupervised or semi-supervised problem of fault detection, Akcay et al. [21] proposed an anomaly detection method based on a generative adversarial network (GAN). An adversarial autoencoder was used as a generator and the overall objective function was optimized, resulting in better performance compared to traditional autoencoder-based methods.

In the initial stage of mechanical equipment operation, data can only be collected under normal operating conditions. During this period, fault detection is called anomaly detection, which mainly distinguishes fault data from normal data [22]. This implies using only normal data to train the model and then judging whether the test data is normal or not. If it is detected that the test data do not belong to the normal category, a new fault is detected and the fault classification method can be used to further classify the fault in the next step. Zhang et al. [23] performed anomaly detection on a wind turbine bearing dataset based on a one-dimensional convolutional neural network and a long short-term memory network. Neupane et al. [24] used batch normalization and CWT technology to conduct fault diagnosis experiments on bearing data using a convolutional neural network, obtaining a high accuracy rate. Typically, training samples are obtained gradually during the operation of mechanical equipment, which requires the network to continuously acquire knowledge from new data samples in order to allow the network to continuously detect new unknown faults while diagnosing known faults. Polikar et al. [25] proposed an incremental learning algorithm that can meet the above requirements. This algorithm can continue to learn new knowledge on the basis of the existing model, allowing it to adapt to the changing external environment. Carino et al. [26] and Delgado-Prieto et al. [27] proposed incremental learning methods based on OC-SVM for fault diagnosis in different scenarios. Liu et al. [28] built an adaptive incremental model based on an extreme learning machine (ELM) which was able to incrementally update the model based

on fault data. However, this method needs to repeatedly use all the data to update the model in order for the updated model to identify new faults. This method relies on a large number of data samples; moreover, network training takes a long time and consumes a large amount of computing resources. Therefore, online dynamic updating of the fault identification and classification model cannot be realized [29]. To address this problem, Liang et al. [30] proposed the OS-ELM algorithm. This algorithm uses sequential learning to replace the batch learning strategy of ELM, which has the advantages of fast calculation speed and strong generalization ability of ELM while avoiding the need to retrain on historical data; moreover, it can be updated online as new data arrive. This algorithm has a great advantage in the scenario where data are added one-by-one or block-by-block. Wang et al. [31] provided a review and summary of extreme learning machines and their extensions. However, existing methods based on OS-ELM have all been experimentally verified using labeled datasets, making it difficult to identify and diagnose new faults without labels. Therefore, existing research cannot simultaneously realize the identification of new types of faults, the diagnosis of known faults, and the automatic update of models without labels.

3. Incremental Single-Class Fault Detection and Diagnosis Method Based on OS-ELM

This section introduces the working principle of the proposed method in detail, including the model structure, training strategy, diagnostic principle and process, etc. Figure 1 shows the basic structure of the incremental single-class fault identification and diagnosis model, which includes two parts: a feature extraction network and a fault diagnosis network. The feature extraction network consists of Encoder-A, Decoder, and Encoder-B, while the fault diagnosis network is composed of multiple OS-ELMs, each of which represents a certain type of fault. The feature vector output of the feature extraction network is used as the input of the fault diagnosis network.

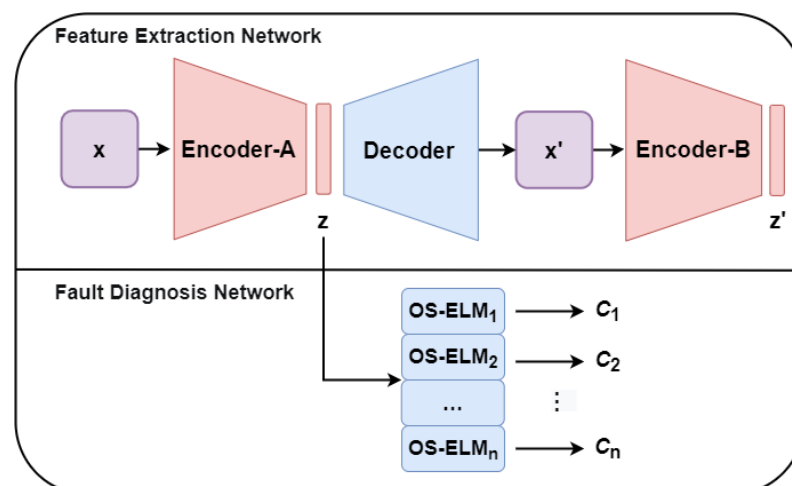


Figure 1. Structure of incremental single-class fault detection and diagnosis model.

3.1. Feature Extraction Network

3.1.1. Network Structure

It can be clearly seen in Figure 1 that the feature extraction network consists of two encoders and one decoder. Encoder-A and Decoder have a symmetrical structure, and together form the autoencoder. An autoencoder is an unsupervised neural network model that learns and reconstructs the original input data through encoding–decoding. A convolutional neural network is a typical feed-forward neural network which extracts features from input data by building multiple filters. By combining the powerful feature extraction ability of the convolution operation and the unsupervised learning ability of the autoencoder, a more efficient unsupervised feature extractor can be obtained. Therefore, we add convolutional layers in the encoder and correspondingly add deconvolutional layers in the

decoder to enhance the feature extraction capability of the network. The detailed structural diagram of the feature extraction network is shown in Figure 2.

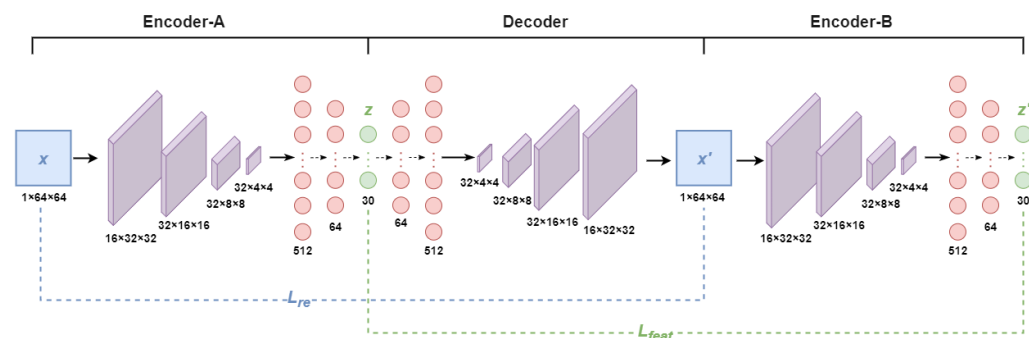


Figure 2. Structural diagram of the feature extraction network.

Encoder-A and Decoder together form an unsupervised convolutional autoencoder network. Encoder-A includes four convolutional layers and two fully connected layers, while Decoder includes two fully connected layers and four deconvolutional layers. The selection of these parameters is initially designed based on the general design principles of the CNN model while comprehensively considering previous research results and continuously conducting experiments to find the optimal parameters. The sample x input to the Encoder-A first passes through four layers of convolution operations; each convolution layer uses a 3×3 convolution kernel to downsample the image, and we set the stride to 2 and the edge padding to 1. Therefore, each layer of convolution operation reduces the image by half. The output size after each layer operation is marked in the figure; it can be seen that the input image with an input size of 64×64 finally becomes a 4×4 feature representation. In order to facilitate extraction of the one-dimensional features of the input image, we flatten the 4×4 feature map into one-dimensional data, input it into the fully connected layer for further feature extraction and compression, and obtain the hidden layer feature z .

Because the decoder needs to form an autoencoder structure with Encoder-A, its structure is designed to be the exact opposite of the Encoder-A. The hidden layer feature z is input into the decoder; it first passes through two full-connection operations, then is de-flattened into two-dimensional data for further deconvolution operations. A deconvolution operation, sometimes called a transposed convolution operation, is often used to expand the image size. To ensure that the enlarged image x' has the same size as the sample x , all deconvolution layers use a 3×3 convolution kernel, a stride of 2, and the same padding. Finally, a reconstructed image x' with the same size as sample x is obtained.

In particular, we improve on this autoencoder by adding an additional Encoder-B. We use this encoder to further compress the reconstructed image x' to obtain the low-dimensional features z' of the reconstructed image. The structure of this encoder is exactly the same as that of the first encoder; thus, the feature z' obtained by the second encoder has the same size as the feature z obtained by the first encoder for comparison.

Rectified Linear Unit (ReLU) is selected as the activation function for all hidden layers of both encoders and decoders.

3.1.2. Training Strategy

The purpose of constructing a feature extraction network is to extract potential features from data samples, then identify and diagnose incremental single-class faults based on the extracted features. The quality of feature extraction directly affects the recognition accuracy of the fault diagnosis model. Therefore, choosing an appropriate training strategy is crucial for subsequent diagnosis. The algorithm proposed in this paper belongs to the incremental learning algorithm. Therefore, only the data in the normal state can be used initially, and

when training the feature extraction network only the data in the healthy state are used for training.

Because Encoder-A and Decoder together form an unsupervised convolutional autoencoder network, the network can be trained by minimizing the loss between the input image and the reconstructed image generated by the decoder; in this way, Encoder-A can obtain the feature distribution implicit in the data in the healthy state. Therefore, we use the mean square error loss function to calculate the reconstruction loss (L_{re}) between the input image and the reconstructed image, expressed as follows:

$$L_{re} = \frac{1}{N} \sum_{i=1}^N ||x_i - x'_i||^2. \quad (1)$$

In Equation (1), N represents the number of samples in a healthy state, x_i is the input sample, and x'_i is the reconstructed sample of x_i , which is the image x' generated by the decoder in Figure 1.

In order to further constrain the model to ensure that the model learns a more accurate and stable latent distribution representation, we additionally add Encoder-B to extract the hidden features of the reconstructed image x' and add the hidden layer features output by the two encoders to the backpropagation optimization process. Because the encoder can extract the lower-dimensional potential distribution of the sample, as the training process progresses the feature representations extracted by two encoders grow closer to the real distribution of normal data and the distance between the two features becomes smaller. We use the Euclidean distance to define the feature loss (L_{feat}) between two hidden layer features, expressed as follows:

$$L_{feat} = ||z - z'|| \quad (2)$$

where z and z' represent the hidden layer feature representations output by Encoder-A and Encoder-B, respectively.

In general, the final target loss function of the feature extraction network can be expressed as

$$L = w_{re}L_{re} + w_{feat}L_{feat} \quad (3)$$

where w_{re} and w_{feat} are hyperparameters, which are weighted parameters used to adjust the influence of each part of the loss function on the overall objective function.

In addition, during the training and construction of the feature extraction network its parameters are continuously updated and the data distribution of other network layers changes, which leads to poor training effects of the network model. Therefore, we add a batch normalization layer between the convolutional layer and the activation layer. Before inputting the output data to the next layer, we first normalize the data to a standard normal distribution, in order to improve the model convergence speed, reduce the possible gradient dispersion and gradient explosion problems, and improve the stability of the model [32].

3.2. Fault Diagnosis Network

3.2.1. Method Overview

The proposed method for incremental single-class fault detection and diagnosis in this paper not only requires a strong capability for novelty detection and fault classification, it demands the ability to learn incrementally. OS-ELM is based on ELM, inheriting all the advantages of ELM. It primarily addresses the learning problem when data arrives in batches. The fault diagnosis network described in this section consists of multiple OS-ELM units, each representing a specific fault state. We assume that initially only the healthy state of the mechanical equipment is known. Under the current normal operating state, a batch of data is collected as a training set to train an OS-ELM₁ representing the current state. The weight values β_1 of this OS-ELM are recorded and a new class detection threshold δ is set.

When a new sample is input to the fault diagnosis network, it is first used to compute a new weight value β_{new_1} using OS-ELM₁. Then, the distance between the two weight values is compared with the threshold. If this is less than the threshold, the new sample

belongs to the type represented by that OS-ELM, i.e., the normal class. If it is greater than the threshold, it is classified as a new class. Simultaneously, these samples are collected as a training set to train a new OS-ELM, which is added to the fault diagnosis network.

When new samples are input again, they are first evaluated by OS-ELM₁. If they are determined to not belong to the normal type, they are then input to other OS-ELM units for evaluation until the class that satisfies the condition is found. When a certain amount of data for a specific fault class is collected, these data can be used to update the existing OS-ELM, enhancing its ability to represent fault conditions. Figure 3 provides a detailed overview of the composition and diagnostic method of the fault diagnosis network.

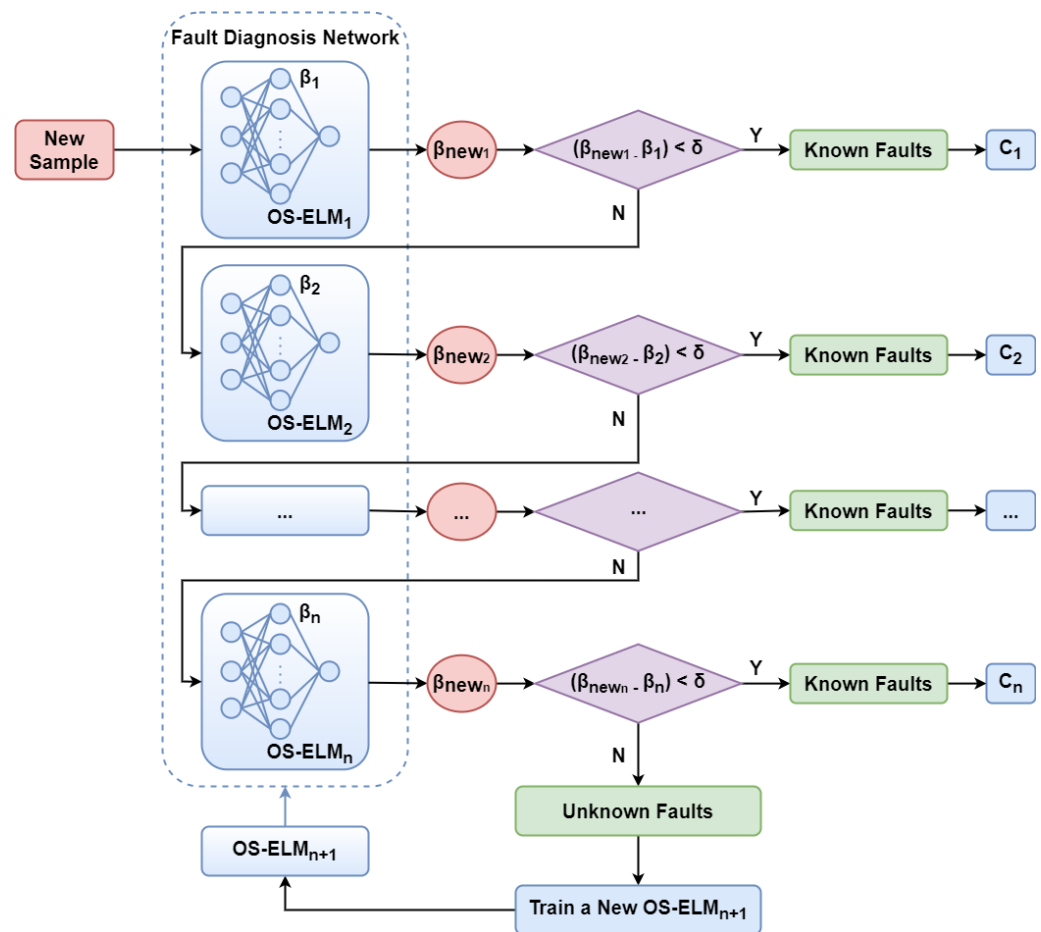


Figure 3. Structure and diagnosis method of the fault diagnosis network.

3.2.2. OS-ELM Training

OS-ELM updates the output weights of hidden layer nodes according to data changes, thereby adapting to scenarios where the training data are constantly being updated. It is a feed-forward neural network with a single hidden layer, and the training process mainly includes two stages, as shown in Figure 4. The first stage is the initial training stage, in which a small number of known data samples are used to obtain the initial output weight β_1 ; the second stage is the online learning stage, in which newly collected samples are used to update the initial weight β_1 obtained in the first stage to obtain a new output weight β_{new} .

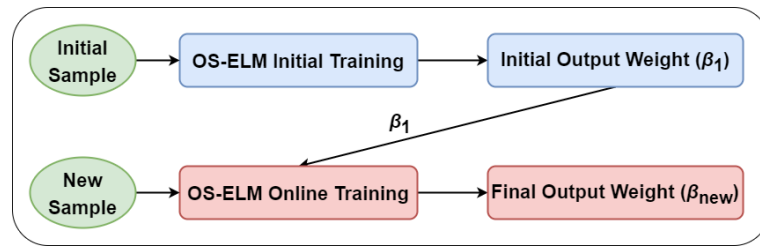


Figure 4. Weight update mechanism of OS-ELM.

When there are N_0 training samples X_i in the initial stage $i = 1, 2, \dots, N_0$, the input weights w_i and bias b_i of the hidden layer are first randomly generated, then the hidden layer output matrix H_1 is calculated:

$$H_1 = \begin{bmatrix} g(w_1, b_1, X_1) & \dots & g(w_m, b_m, X_1) \\ \vdots & \ddots & \vdots \\ g(w_1, b_1, X_{N_0}) & \dots & g(w_m, b_m, X_{N_0}) \end{bmatrix} \quad (4)$$

where g is the sigmoid activation function.

Assuming that the expected output value is $T_1 = [Y_1, Y_2, \dots, Y_{N_0}]^T$, the goal is to find the minimum output weights β that satisfy $\|H\beta - T\|$. Then we can use the following formula to find the minimum output weight β_1 that satisfies $\|H\beta - Y\|$:

$$\beta_1 = M_1 H_1^T T_1 \quad (5)$$

where $M_1 = (H_1^T H_1)^{-1}$ is the generalized inverse matrix in the initial phase.

After obtaining the initial output weights β_1 , the initial phase concludes. When new samples are introduced into the model, the output weights β_{new} for the new samples are determined using the Extreme Learning Machine theory, as follows:

$$\beta_{new} = \beta_1 + M_{new} H_{new}^T (T_{new} - H_{new} \beta_1). \quad (6)$$

Similarly, when the $k + 1$ batch of data arrives, we use the recursive least squares method to obtain the output weight:

$$M_{k+1} = M_k - M_k H_{k+1}^T (I + H_{k+1} M_k H_{k+1}^T)^{-1} H_{k+1} M_k, \quad (7)$$

$$\beta_{k+1} = \beta_k + M_{k+1} H_{k+1}^T (T_{k+1} - H_{k+1} \beta_k). \quad (8)$$

In Equation (8), M_k is used to store intermediate computation results, H_{k+1} represents the output from the hidden layer for the new data, and T_{k+1} corresponds to the expected output values for the new data.

At this time, we update the OS-ELM model according to the new data and obtain a new output weight β_{new} . Later, the online learning mechanism of OS-ELM can be used to set the new class detection threshold and incrementally update the model.

3.2.3. Novelty Detection

In the previous section, we introduced the working principle of OS-ELM. This section describes the new class detection and fault diagnosis principle of the fault diagnosis network in detail.

We implement the new class detection function according to how much OS-ELM is modified by the new samples. Assuming that an OS-ELM representing a normal class is trained using samples in a normal state, when the new sample belongs to the normal class the update to the model in the online learning phase will be small, that is, the generated β_{new} will not differ greatly from the β_1 in the initial stage. However, when the new samples do not belong to the normal class, the update to the model in the online learning stage will

be much greater than that of the normal samples, and the β_{new} generated based on the new samples will differ greatly from the β_1 in the initial stage. Therefore, new samples of the same type can be used to learn OS-ELM online, record the modification amount of the output weight, and set the threshold according to the modification amount to realize the recognition of new classes.

Assuming that initially there is a batch of training samples $(X_i, Y_i), i = 1, 2, \dots, N_0$, it is worth noting that the samples input to the fault diagnosis network in this article are feature samples extracted by the feature extraction network, rather than raw data collected directly from mechanical sensors. In addition, because the method proposed in this paper is an unsupervised learning algorithm, the training samples do not have labels. Therefore, when training OS-ELM we need to assign pseudo-labels to the training samples. The pseudo-labels are obtained by the following formula:

$$Y_i = \sum_{n=1}^{d(X_i)} f(X_i(n)) \quad (9)$$

$$f(X_i(n)) = \begin{cases} \cos(X_i(n)), & n \in \{x|x = 2k, k \in \mathbb{N}\} \\ \sin(X_i(n)), & n \in \{x|x = 2k + 1, k \in \mathbb{N}\} \end{cases} \quad (10)$$

where $d(X_i)$ represents the number of elements in the sample vector X_i .

We divide the training samples into two subsets X_i^a and X_i^b on average, $i = 1, 2, \dots, N_1$, $N_1 = \frac{1}{2}N_0$, one for training in the initial stage of OS-ELM and one for setting the new class detection threshold in the online learning stage. According to the OS-ELM training algorithm introduced in the previous section, we use X_i^a to train an OS-ELM representing the normal class and obtain the output weight β_1 of OS-ELM. In order to avoid the contingency caused by a single calculation, we divide X_i^b into k batches; each batch contains N_1/k samples, these k batches are sequentially input into OS-ELM for online learning phase training, and k updated output weights β_{new} are obtained. First, we calculate the Euclidean distance between β_1 and each β_{new} , defined as follows:

$$D_k = \sqrt{(\beta_1 - \beta_{new})^2} \quad (11)$$

In Equation (11), D_k represents the Euclidean distance between β_1 and the β_{new} obtained from the k -th batch.

After obtaining these distances for the k -th batches, they form a distance set $D = \{D_i | i = 1, 2, \dots, k\}$. The threshold is set using the box plot method [33]. As shown in Figure 5, the upper quartile of this set is chosen as the threshold.

$$\delta = Q3 \quad (12)$$

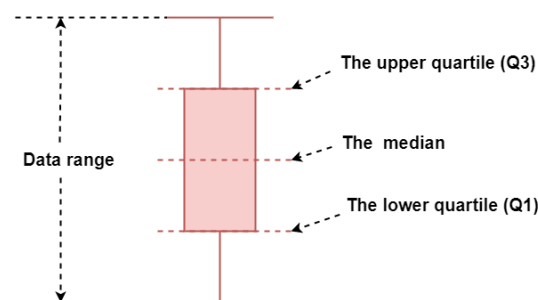


Figure 5. Principle of threshold selection.

After the threshold is determined, the test samples are input into the fault diagnosis network for novelty identification. To avoid randomness, after obtaining the distance set D

a sliding window is introduced to further segment D . The size of the sliding window is set to 10 and the sliding step is set to 1. The segmentation method is illustrated in Figure 6.

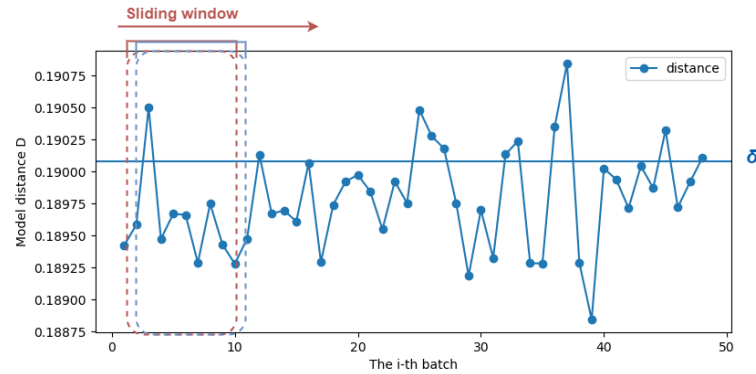


Figure 6. Novelty identification method based on sliding window.

When there are five or more distances greater than the threshold in each window, it is determined that the batch of samples does not belong to the category, that is, the unknown category, which can be expressed as follows:

$$Output = \begin{cases} Unknown\ class, & Num(D_i > \delta) \geq 5 \\ Known\ class, & Num(D_i > \delta) < 5 \end{cases} \quad (13)$$

If the batch of samples is judged as a new class by the current OS-ELM, it is input to other OS-ELMs to continue the judgment, and the novelty identification method is the same as above. When the OS-ELM representing all known types considers the sample to be a new class, it is identified as an unknown fault. The pseudocode of the algorithm used for new class detection is shown in Algorithm 1.

Algorithm 1 New fault detection algorithm based on OS-ELM

Input: training samples $(X_i, Y_i), i = 1, 2, \dots, N$, the number of hidden layer nodes of OS-ELM m , activation function *sigmoid*.

Output: Fault diagnosis results.

- 1: Initialization phase S_1 : train an OS-ELM network using X_i^a sub-dataset;
 - 2: **for** $j = 1$ to m **do**
 - 3: Randomly generate the weights w_i and b_i from the input layer to the hidden layer;
 - 4: **end for**
 - 5: Calculate the hidden layer output matrix H_1 .
 - 6: Calculate the output weight β_1 and record.
 - 7: $k = 0$.
 - 8: Online learning phase S_2 : Retrain the OS-ELM network using the $k + 1$ batch of data.
 - 9: Calculate β_{new} based on Equations (7) and (8) and record.
 - 10: $k = k + 1$.
 - 11: Calculate the Euclidean distance between β_1 and each β_{new} based on Equation (11) and form a distance set D .
 - 12: Determine whether the sample is a known sample or an unknown sample based on Equation (13).
 - 13: **return** *Output*.
-

3.2.4. Incremental Learning Mechanism

The fault identification and diagnosis model proposed in this paper has incremental learning ability. When a piece of equipment is put into production, initially only data about its normal operating state are available. Therefore, the model initially learns from normal data, detects a new fault when it first occurs, collects data under that fault condition, and builds a new OS-ELM to add to the fault diagnosis network. When the new sample is input

again and all OS-ELMs in the fault diagnosis network judge the sample as a new fault, then N data are collected under this fault condition and an OS-ELM is trained representing a new fault in order to add to the fault diagnosis network. When the fault diagnosis network contains n OS-ELMs representing n categories, denoted as $\{C_1, C_2, \dots, C_n\}$, then the new fault is labeled as C_{n+1} , and the number of OS-ELMs in the fault diagnosis network increases by one.

3.3. Diagnosis Process

The proposed incremental single-class fault detection and diagnosis method based on OS-ELM aims to construct a comprehensive model that combines new class fault detection, known fault classification, and incremental updates. The goal is to address the issue of existing models being unable to identify unknown faults. The flowchart of the fault diagnosis method is illustrated in Figure 7. The method first preprocesses the collected raw signals and then trains an unsupervised feature extraction network to extract low-dimensional features of data samples. Finally, a baseline OS-ELM is trained using sample features and a threshold is set to detect unknown faults. When new test data are collected, after data preprocessing, the sample features are extracted by the feature extraction network, then the extracted features are input into the fault diagnosis network for fault identification and diagnosis. When there are multiple OS-ELMs in the fault diagnosis network, that is, there are multiple known fault types, they are compared with each OS-ELM one by one until a class that meets the criteria is identified, resulting in an output of the fault type. If the new class determination conditions are not satisfied for all OS-ELMs, an output of an unknown fault is generated. Meanwhile, data related to this condition are collected to train a new OS-ELM, which is then added to the fault diagnosis network. This fault condition is subsequently treated as a known fault for ongoing fault identification and diagnosis.

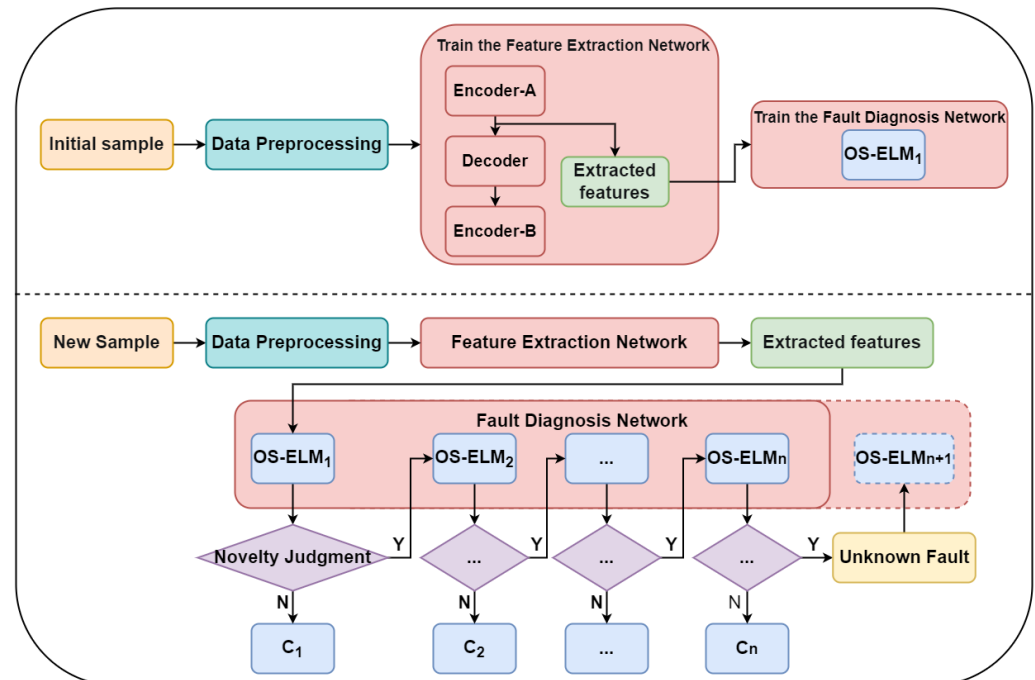


Figure 7. Flow chart of fault identification and diagnosis.

4. Experiment

To assess the performance of the proposed incremental single-class fault detection and diagnosis method, experiments were conducted using two publicly available bearing datasets. These experiments involved comparisons with similar research methods from recent years; diagnostic accuracy is reported for each dataset.

4.1. Dataset Introduction

4.1.1. CWRU Dataset

The CWRU dataset [34] is a publicly available dataset provided by the Case Western Reserve University Bearing Data Center. In the experiments, acceleration signals were collected using an accelerometer at the motor drive end for bearing fault diagnosis. The sampling frequency was 12 kHz. As shown in Table 1, a set of samples was selected including one category of data representing normal operating conditions and three categories representing faulty states. These three fault categories are inner race fault, rolling element fault, and outer race fault, each with a fault diameter of 0.007 inches.

Table 1. CWRU dataset.

Fault Condition	Class	Number of Original Data Points	Number of Samples
Normal	C1	243,938	3796
Inner race fault	C2	121,265	1879
Balls fault	C3	122,571	1900
Outer race fault	C4	121,991	1891

In the data preprocessing stage, overlapping sampling is used for signal segmentation. When performing signal segmentation on the collected raw data, there is an overlapping area between each segment of the signal and the next segment; the specific sampling method is shown in Figure 8.

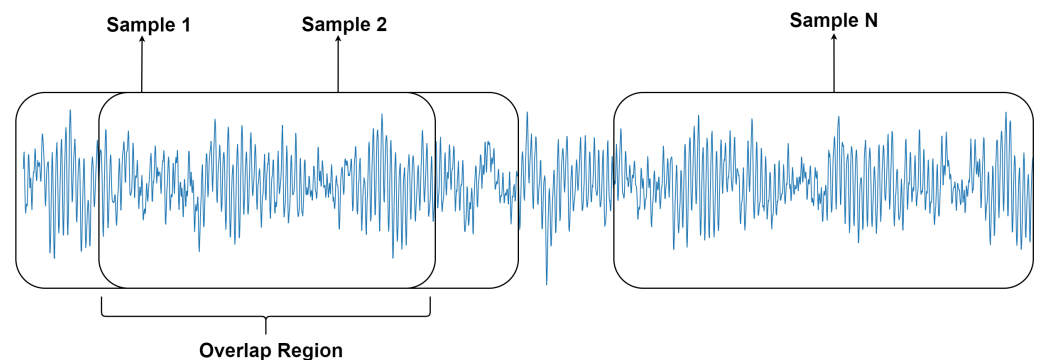


Figure 8. Overlap sampling method.

Each sample contained 1024 data points, and the size of the overlapping area was set to 960. After signal segmentation, the number of samples of each type is shown in the last column of Table 1. The segmented samples were subjected to continuous wavelet transform, and the one-dimensional signal samples were converted into two-dimensional time–frequency images. The size of the time–frequency map was 64×64 , which is a reasonable size that our model can handle. Figure 9 shows the signal samples and corresponding time–frequency diagrams under four kinds of different fault conditions. The upper portion of the image depicts a one-dimensional vibration signal, which is a signal that varies with time. Through continuous wavelet transform, the one-dimensional signal can be converted from the time domain to the time–frequency domain. This transformation allows for the simultaneous acquisition of both time and frequency information from the signal. The lower part of the image shows the corresponding two-dimensional time–frequency map, with time represented on the horizontal axis and frequency on the vertical axis. The brightness and color of each point on the map indicate the strength or energy of the signal at different time and frequency points. This enables us to observe the time and frequency characteristics of the signal in greater detail. Such representations find widespread utility in fields such as signal processing, pattern recognition, and numerous other applications.

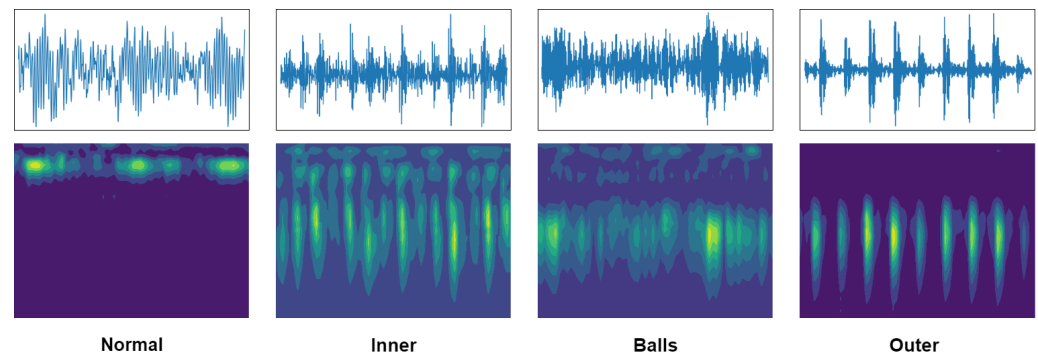


Figure 9. The signal converted to a time–frequency plot.

The method in this paper is a fault diagnosis model with new class recognition and incremental learning capabilities, imitating the scenario where only the data in the normal operating state are available when the mechanical equipment is put into production. In our experiments, we assume that the known types are only in normal operation state and the other types of faults are all unknown faults. Therefore, 70% of the samples in the normal state are randomly selected as training samples for preliminary training of the network and the remaining 30% and all other types of samples are input into the network as test samples for fault identification and diagnosis.

4.1.2. MFPT Dataset

The MFPT dataset [35] is a bearing failure dataset published by the Society for Machinery Failure Prevention Technology. The dataset includes vibration data under normal operating conditions and inner ring fault data and outer ring fault data at different sampling frequencies. In this paper, we randomly select the data under normal operation, the type of inner ring fault, and the type of outer ring fault as the experimental data to verify the effectiveness of the method proposed. The experimental data are shown in Table 2.

Table 2. MFPT dataset.

Type of Fault	Class	Sampling Rate (sps)	Load (lbs)	Number of Data Points	Sample Size
normal	C1	97,656	270	585,936	9140
Outer	C2	48,828	25	146,484	2273
Inner	C3	48,828	0	146,484	2273

Data points were segmented using the overlapping sampling method shown in Figure 7. The length of each sample was 1024, and the overlapping area was again set to 960. The number of samples after division is listed in the last column of Table 2. Use CWT to process the data sample and convert it into a 64×64 time–frequency image suitable for the model in this paper. The data samples and the transformed time–frequency diagram are shown in Figure 10. In order to verify the effectiveness of the method proposed in this paper, we assume that the initially available data are only sample data under normal conditions and that the outer and inner ring faults are unknown types of faults. We divide the samples in the normal state into training samples and test samples in a ratio of 7:3; 70% of the samples are used to train the network, and the remaining 30% together with all the data under the outer ring fault and inner ring fault constitute the test set.

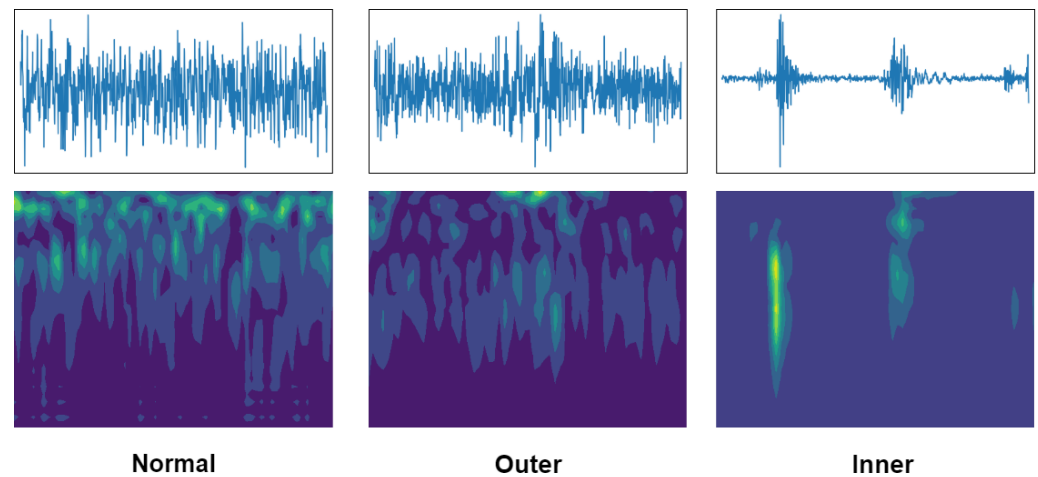


Figure 10. Time–frequency plot of the MFPT dataset.

4.2. Implementation

The experimental environment we used was as follows: Windows 10, Python 3.9.7, and Pytorch 1.13.1; the computer configuration was an Intel Core i7-10700 CPU, 2.90 GHz processor, and 16 GB RAM.

After data preprocessing, adopted the method described in Section 3.1.2 and only used the data obtained under the normal operating state for training the unsupervised feature extraction network. For the hyperparameter selection problem, we used a grid search algorithm to find the solution. In this paper, the limited option set of the hyperparameters is as follows: $learning_rate \in \{0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005\}$, $batch_size \in \{10, 20, 30, 40\}$, $w_r \in \{1, 10, 20, 30, 40, 50\}$, $w_{feat} \in \{1, 10, 20, 30, 40, 50\}$, the number of hidden layer nodes $m \in \{30, 80, 300, 800\}$, we bring the randomly drawn hyperparameters into the training and testing carried out in the network, and the optimal model obtained by searching is saved with the goal of optimizing the test accuracy. The final selection learning rate was 0.00005, the weight decay was set to 0.0001, the Adam optimizer with a batch size of 20 was used to train the feature extraction network, and the weighting parameters were $w_r = 50$ and $w_{feat} = 1$ in the total target loss function. We set Epoch = 100. In order to avoid overfitting, we used the EarlyStopping mechanism to monitor the loss change between adjacent iterations during the training process and terminated the model training in due course.

After the feature extraction network was trained, the test samples were sequentially input into the feature extraction network in a certain order to extract the features and obtain a feature set. Then, the feature set was used to train an OS-ELM and add it to the fault diagnosis network. The number of hidden layer nodes of OS-ELM m was set to 300. In the process of learning and building the fault diagnosis network, the new class recognition and incremental learning capabilities of the method proposed in this paper were first verified, then a comparative experiment was carried out to better illustrate the effect of the proposed model.

4.3. Evaluation Indicators

In the process of fault identification and diagnosis, the test samples are sequentially input into the OS-ELM for judgment. For each OS-ELM, this is a binary classification problem; either the sample belongs to the known class represented by the current OS-ELM, or it belongs to an unknown class. The novelty detection performance of the proposed method can be evaluated according to the novelty detection performance of each OS-ELM. As per Figure 6 in Section 3.2.3, the new class recognition function proposed in this paper is mainly implemented based on sliding windows; if five of every ten windows are greater than the threshold, then the batch of samples is judged to be a new class. Therefore, the new class detection rate of each OS-ELM can be calculated according to the ratio of the

number of windows judged as new classes to the total number of windows. The formula can be expressed as follows:

$$ACC = \frac{Num(Output = Unknown\ class)}{N_{test}/batch_size - 10 + 1} \quad (14)$$

where N_{test} is the total number of samples in the test dataset.

4.4. Experimental Results and Analysis

We analyzed the detection results of the method proposed in this study on two datasets, with the new class detection rate used as the evaluation index. To further demonstrate the diagnostic performance of our model, we compared it with OCSVM-based and IF-based fault diagnosis methods.

4.4.1. Detection Performance on the CWRU Dataset

Table 3 reports the detection results of the proposed method on the CWRU dataset. Each OS-ELM was tested ten times, and the average value of the test was selected as the final fault diagnosis result. The row represents the OS-ELM trained by each type, the column represents the fault type of the test sample, and each cell represents the new class detection rate of a specific OS-ELM when detecting different types of fault samples. It can be clearly seen from the table that when the OS-ELM detects the type of fault, the detection rate of the new type is mostly 0%, while when OS-ELM detects different types of faults the detection rate of the new type is close to 100%; this indicates that all OS-ELMs can sufficiently distinguish the represented faults from other new classes of faults, reaching an average accuracy of 99.62%.

Table 3. Detection results on the CWRU dataset.

	Normal	Inner	Balls	Outer	Average
OS-ELM ₁ (Normal)	0%	100%	98.98%	99.83%	
OS-ELM ₂ (Inner)	100%	0.53%	98.37%	100%	99.62%
OS-ELM ₃ (Balls)	100%	98.67%	0%	100%	
OS-ELM ₄ (Outer)	100%	98.53%	100%	0%	

To provide a clearer illustration of the fault diagnosis details and validate the diagnostic accuracy, we provide a detailed description of the experiments conducted on the CWRU dataset.

After training the feature extraction network using the training set, test samples of the normal class were input into the network to extract features. Subsequently, the extracted features were used to train an OS-ELM₁ representing the normal class and a new class detection threshold was set. Following this, samples representing normal operation, inner race faults, rolling element faults, and outer race faults were sequentially input into OS-ELM₁ for detection. The results obtained by OS-ELM₁ are shown in Figure 11, where the x-axis represents sample batches and the y-axis represents the modification of output weights in OS-ELM₁ for that batch of samples; the blue horizontal line represents the new class detection threshold of OS-ELM₁ ($\delta = 0.697$), and each data point on the line chart represents the distance D_k calculated using Equation (14).

From the graph, it can be observed that the updates of the OS-ELM₁ model for normal class samples are mostly distributed below the threshold, while inner race fault, rolling element fault, and outer race fault samples are mostly distributed above the threshold. This indicates that the OS-ELM₁ representing the normal class can accurately determine whether the test samples belong to the normal class or a new class.

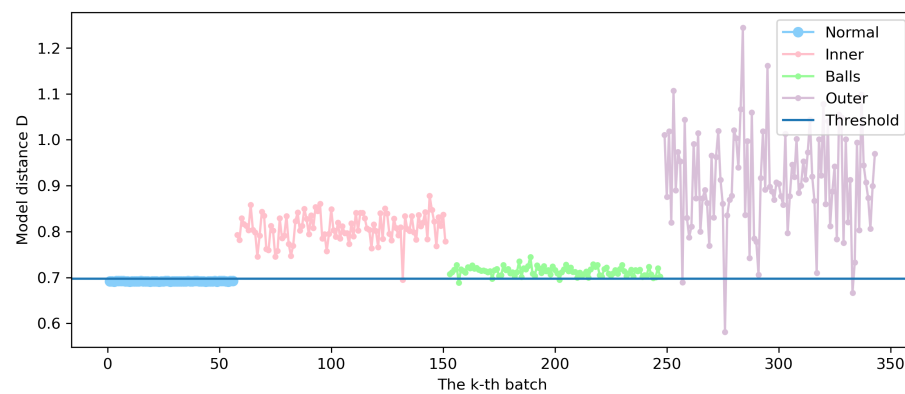


Figure 11. New class detection performance of OS-ELM₁ on the CWRU dataset.

When OS-ELM₁ first detects an inner race fault, the sample batches under this condition are collected. Using the same method, an OS-ELM representing the inner race faults, denoted as OS-ELM₂, is trained, then four types of test samples are input sequentially to test the detection performance of OS-ELM₂. As shown in Figure 12, the sample batches of the inner race fault exhibit updates in the current OS-ELM that are smaller than the threshold. Conversely, the sample batches of the other three fault types lead to larger updates in the model, resulting in distances above the threshold. This indicates that OS-ELM₂ possesses excellent capability for new class recognition and fault diagnosis.

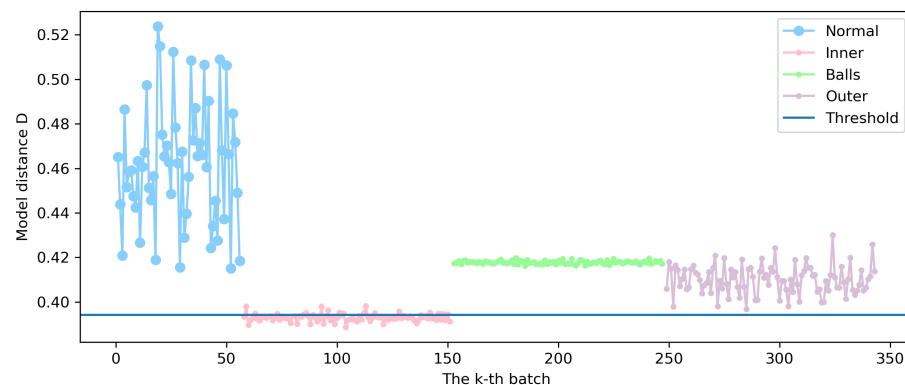


Figure 12. New class detection performance of OS-ELM₂ on the CWRU dataset.

When both OS-ELM₁ and OS-ELM₂ detect a rolling element fault, the sample batches under this condition are collected. An OS-ELM representing the rolling element faults, labeled as OS-ELM₃, is trained using the same method. The new class recognition performance of OS-ELM₃ is evaluated similarly. The results are shown in Figure 13. When the data of fault types other than rolling element faults are input to OS-ELM₃ for detection, most of the updated values of the model are greater than the threshold value, and only a few batches of outer ring faults are less than the threshold value. However, this does not affect the results of new class detection, as we use a sliding window with a step size of 1 and a width of 10 when judging whether it is novel. Therefore, only when more than half of the distances in the window are greater than the threshold is it judged to be a new class.

When OS-ELM₁, OS-ELM₂, and OS-ELM₃ all detect outer ring faults, we use the same method to train an OS-ELM₄ representing outer ring faults and test its new class recognition performance. The results are shown in Figure 14. It is obvious that the threshold of OS-ELM₄ separates the outer ring faults from other faults well, verifying the detection ability of the new class.

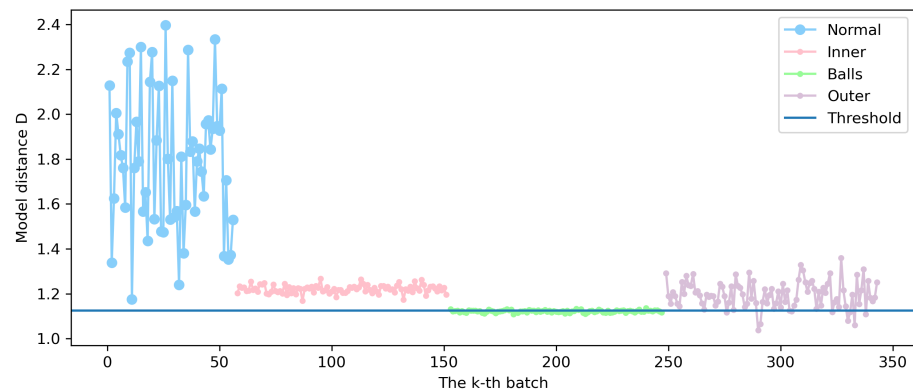


Figure 13. New class detection performance of OS-ELM₃ on the CWRU dataset.

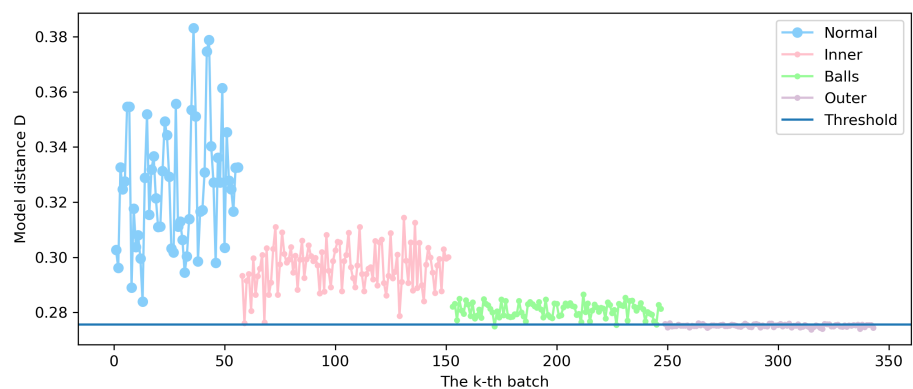


Figure 14. New class detection performance of OS-ELM₄ on the CWRU dataset.

4.4.2. Detection Performance on the MFPT Dataset

Experiments were carried out on the MFPT dataset in the same way. The test results are listed in Table 4. It can be seen that the method in this paper has good new class recognition and fault diagnosis performance on the MFPT dataset, with an average accuracy rate of 98.80%.

Table 4. Detection results on the MFPT dataset.

	Normal	Outer	Inner	Average
OS-ELM ₁ (Normal)	2.13%	96.59%	100%	98.80%
OS-ELM ₂ (Outer)	97.27%	0%	97.85%	
OS-ELM ₃ (Inner)	99.66%	100%	0%	

Similarly, initially only samples of the normal class are available, and the sample features of the normal class are used to train an OS-ELM₁ representing the normal class. Afterwards, the extracted sample features of normal operation, outer ring faults, and inner ring faults are input into OS-ELM₁ for detection. When a new class of faults is detected, samples under the new class of faults are collected, a new OS-ELM is trained, and detection continues. Figures 15–17 show the detection results of each OS-ELM. Overall, all OS-ELMs have satisfactory new class recognition capabilities.

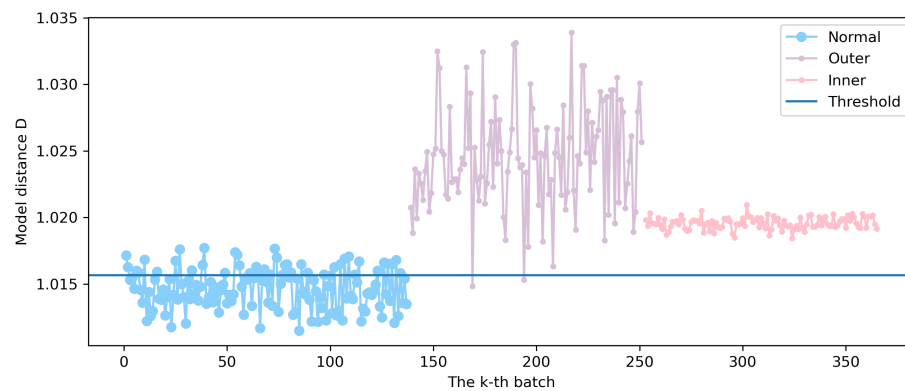


Figure 15. New class detection performance of OS-ELM₁ on the MFPT dataset.

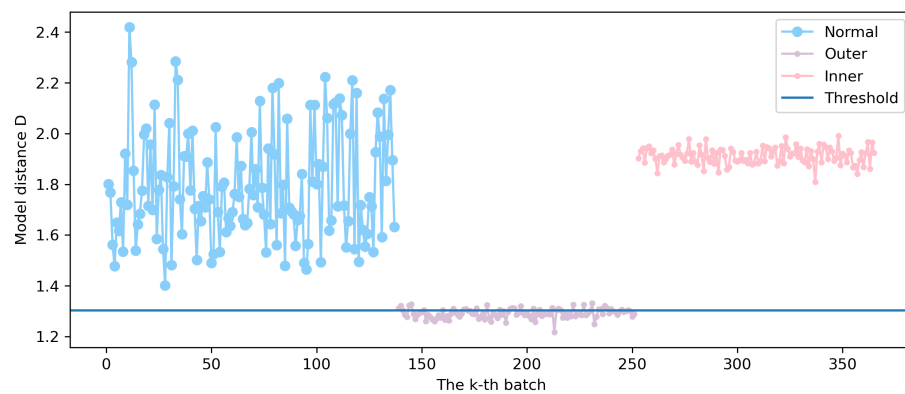


Figure 16. New class detection performance of OS-ELM₂ on the MFPT dataset.

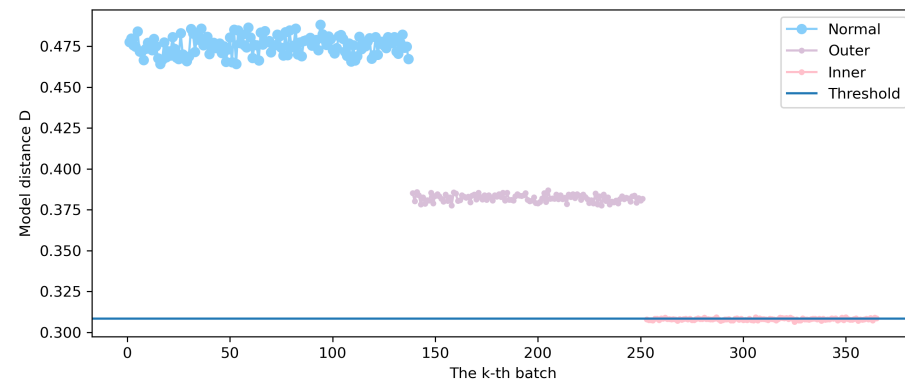


Figure 17. New class detection performance of OS-ELM₃ on the MFPT dataset.

4.4.3. Comparative Experiment

In order to better verify the effectiveness of the proposed method, we took the CWRU dataset as an example and conducted a comparative experiment between our method and two commonly used single-class classification methods, OCSVM and IF. OCSVM is a generalization of a support vector machine for use in anomaly detection scenarios. The method attempts to learn a hyperplane in a high-dimensional feature space that divides all gregarious points to its side and maintains the furthest possible distance from distant points. We selected the Gaussian kernel function and used grid search to determine the γ factor and ν parameter, where the value ranges of γ and ν are $\{2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}\}$, and $\{0.01, 0.05, 0.1\}$. The second method, IF, is an extension of random forest for anomaly detection scenarios. Usually, the algorithm converges faster and is more stable when there are more basic estimator trees. In this experiment, we set the number of basic estimators to 500. Similar to the method proposed in this paper, the sample features are first extracted by a feature extraction network with a double-encoded structure, then input into a single-class

classifier (OCSVM, IF) for fault identification and diagnosis. Tables 5 and 6 show the detection results of the OCSVM and IF methods on the CWRU dataset, respectively.

Table 5. Detection results of the OCSVM method on the CWRU data set.

	Normal	Inner	Balls	Outer	Average
OS-ELM ₁ (Normal)	2.84%	78.72%	75.17%	88.78%	82.07%
OS-ELM ₂ (Inner)	92.56%	3.76%	76.16%	84.83%	
OS-ELM ₃ (Balls)	94.81%	58.81%	3.32%	73.92%	
OS-ELM ₄ (Outer)	88.78%	46.70%	63.75%	0%	

Table 6. Detection results of the IF method on the CWRU data set.

	Normal	Inner	Balls	Outer	Average
OS-ELM ₁ (Normal)	0%	93.64%	83.56%	94.51%	92.33%
OS-ELM ₂ (Inner)	100%	0%	72.87%	94.02%	
OS-ELM ₃ (Balls)	95.64%	76.79%	1.64%	98.63%	
OS-ELM ₄ (Outer)	96.33%	80.54%	92.35%	0%	

In order to more intuitively present the comparison results between this method and methods based on OCSVM or IF, the data in Tables 3, 5, and 6 are visualized below. As shown in Figure 18, each row represents the fault type of the test sample, each column represents the different trained OS-ELMs in the fault diagnosis network, and each cell represents the new class detection rate of each OS-ELM when detecting different types of fault samples. The color intensity is used to visually display the detection rate, with lighter colors indicating higher new class detection rates and darker colors indicating lower rates. In all of our detection results, the colors along the diagonal line are the darkest, indicating that the new class detection rate is very low when the fault types in the training and test sets are the same. This indicates the accurate recognition of known fault types by all three methods. Outside the diagonal, it can be clearly seen that the detection results of the proposed method are very light, indicating that the detection rate of the new class is very high when the fault types of the training set and the test set are different. This proves that our method can identify different fault types. In the diagnostic results based on OCSVM, there are various shades of color blocks in the graph, suggesting that OCSVM's detection ability is weaker when the fault types in the training and test sets are different. In particular, the OCSVM trained on outer race fault data achieves a detection rate of only 46.70% for inner race faults. While the diagnostic results based on the IF method are better than those of OCSVM, the detection rates are lower than those of the proposed method.

In addition to comparing our method these two single-class classification methods, we compared it with similar published fault diagnosis methods on the CWRU dataset. For this comparison, we used the integrated multi-task intelligent bearing fault diagnosis method (TF-MDAE-SAMB-NN) proposed in [36], which is based on representation learning under unbalanced dataset sample conditions, and the fault diagnosis method proposed in [37] which is based on a compact adaptive 1D-CNN classifier, and several others. The classification accuracy results of the compared methods are listed in Table 7.

As can be seen from the table, the incremental single-class fault identification and diagnosis method based on OS-ELM proposed in this article has higher accuracy than other fault diagnosis methods, and is the only one that uses an incremental learning function. When the data distribution changes, the fault diagnosis model proposed in this article does not need to be repeatedly trained in order to update the model, while the other models do not have the ability to be automatically updated. In addition, the other deep learning methods require a large number of labeled samples for training, which requires a great deal of time and effort. The fault diagnosis model proposed in this paper is an unsupervised learning model and does not require labeled sample training, overcoming the difficulties

involved with deep learning methods that require a large amount of sample training data. Therefore, compared with other fault diagnosis methods, the method proposed in this article is better in terms of both accuracy and resource utilization efficiency, verifying its effectiveness.

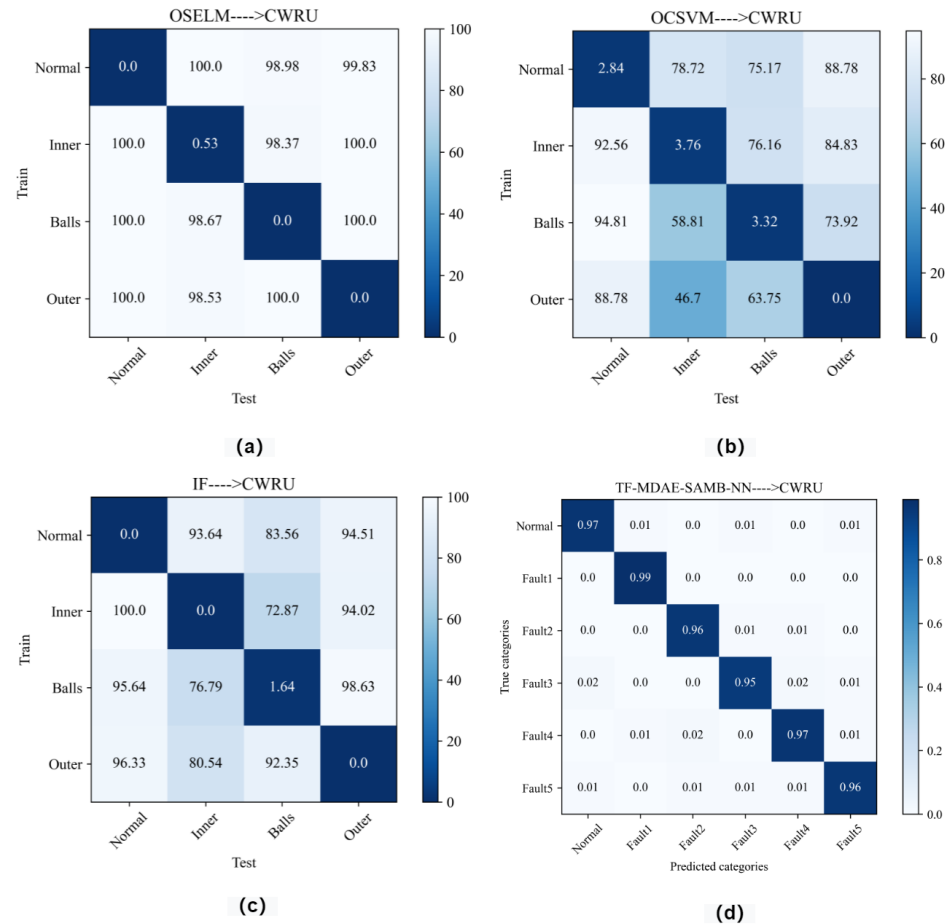


Figure 18. Visualization of new class detection results: (a) OS-ELM, (b) OCSVM, (c) IF, (d) TF-MDAE-SAMB-NN.

Table 7. Fault diagnosis accuracy of different methods on the CWRU dataset.

Method	Average Accuracy of Fault Diagnosis
OCSVM	82.07%
ResNet18	86.88%
2D-CNN	92.28%
IF	92.33%
1D-CNN	93.54%
TF-MDAE-SAMB-NN	96.65%
Proposed OS-ELM	99.62%

It is worth noting that although the accuracy of the proposed method is higher than that of the other methods listed in the table, especially TF-MDAE-SAMB-NN and 1D-CNN, our method does have certain limitations. The method in this paper is most suitable for situations in which various unknown faults appear sequentially, and its diagnostic accuracy is greatly reduced when various types of unknown faults occur out of sequence, as it becomes difficult to collect unknown fault data for each batch.

5. Conclusions

To address the challenging task of detecting new faults in the field of fault diagnosis, in this paper we propose an incremental single-class fault recognition and diagnosis method based on OS-ELM. This method accomplishes both the diagnosis of known fault types and the recognition of new fault types, allowing for differentiation between different new fault classes. In order to reduce the large amount of time required for labeling, we construct a convolutional autoencoder network to extract sample features unsupervised, then add dual encoding loss constraints to optimize the feature extraction model and enhance its feature extraction capabilities. The extracted features are used as inputs to the fault diagnosis network for fault recognition and diagnosis. The proposed method only requires training with data from normal operating conditions, and can differentiate between normal and fault classes when faults occur. It achieves incremental single-class fault recognition and diagnosis by updating the fault diagnosis model. The average diagnostic accuracy of this method on the CWRU data set reached 99.62%, and its diagnostic accuracy on the MFPT data was 98.80%. The effectiveness of the proposed method was verified through comparison with existing single-class fault diagnosis methods and deep learning methods from the literature, showing high diagnostic accuracy. However, it should be noted the proposed method is most suitable for cases in which new fault classes appear sequentially. In scenarios where various new fault classes are mixed together, its fault recognition ability and diagnostic accuracy are likely to decrease due to the difficulty of collecting batch data for new fault classes. Considering the rapid development of deep learning technology, we intend to consider other deep learning frameworks or machine learning methods in our future work to solve this problem and improve the application capabilities of incremental learning models in complex scenarios.

Author Contributions: Conceptualization, H.H. and Y.Z. (Yuanyuan Zhao); Data curation, H.H. and Y.Z. (Yuanyuan Zhao); Formal analysis, H.H. and Y.Z. (Yuanyuan Zhao); Funding acquisition, H.H.; Investigation, Y.Z. (Yuanyuan Zhao), Y.C., Y.Z. (Yu Zhang) and D.W.; Methodology, H.H. and Y.Z. (Yuanyuan Zhao); Project administration, H.H. and Y.Z. (Yuanyuan Zhao); Resources, H.H. and Y.Z. (Yuanyuan Zhao); Software, Y.C.; Supervision, H.H.; Validation, H.H., Y.Z. (Yuanyuan Zhao), Y.C., Y.Z. (Yu Zhang) and D.W.; Visualization, H.H. and Y.Z. (Yuanyuan Zhao); Writing—original draft, H.H. and Y.Z. (Yuanyuan Zhao); Writing—review and editing, H.H. and Y.Z. (Yuanyuan Zhao). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Shandong Provincial Natural Science Foundation, China (grant number ZR2022MF279) and by the Science and Technology SMEs Innovation Capacity Enhancement Project in Shandong Province (grant number 2021TSGC1089).

Data Availability Statement: The datasets used in this article can be obtained from the corresponding author upon request.

Conflicts of Interest: The authors declare that they have no conflicts of interest regarding this work.

References

1. Malla, C.; Panigrahi, I. Review of condition monitoring of rolling element bearing using vibration analysis and other techniques. *J. Vib. Eng. Technol.* **2019**, *7*, 407–414. [\[CrossRef\]](#)
2. Hoang, D.T.; Kang, H.J. A survey on deep learning based bearing fault diagnosis. *Neurocomputing* **2019**, *335*, 327–335. [\[CrossRef\]](#)
3. Zhuang, Z.; Zhang, G.; Dong, W.; Sun, X.; Wang, C. Intelligent fault detection of high-speed railway turnout based on hybrid deep learning. In Proceedings of the AI 2018: Advances in Artificial Intelligence: 31st Australasian Joint Conference, Wellington, New Zealand, 11–14 December 2018; Proceedings 31; Springer: Wellington, New Zealand, 2018; pp. 98–103.
4. Zhao, Y.; Hao, H.; Chen, Y.; Zhang, Y. Novelty Detection and Fault Diagnosis Method for Bearing Faults Based on the Hybrid Deep Autoencoder Network. *Electronics* **2023**, *12*, 2826. [\[CrossRef\]](#)
5. Abid, A.; Khan, M.T.; Iqbal, J. A review on fault detection and diagnosis techniques: Basics and beyond. *Artif. Intell. Rev.* **2021**, *54*, 3639–3664. [\[CrossRef\]](#)
6. Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S.N. A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. *Comput. Chem. Eng.* **2003**, *27*, 313–326. [\[CrossRef\]](#)
7. Qi, R.; Zhang, J.; Spencer, K. A Review on Data-Driven Condition Monitoring of Industrial Equipment. *Algorithms* **2022**, *16*, 9. [\[CrossRef\]](#)

8. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#)
9. Zhao, B.; Zhang, X.; Li, H.; Yang, Z. Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions. *Knowl.-Based Syst.* **2020**, *199*, 105971. [\[CrossRef\]](#)
10. Xie, Z.; Yu, D.; Zhan, C.; Zhao, Q.; Wang, J.; Liu, J.; Liu, J. Ball screw fault diagnosis based on continuous wavelet transform and two-dimensional convolution neural network. *Meas. Control* **2023**, *56*, 518–528. [\[CrossRef\]](#)
11. Jiao, J.; Zhao, M.; Lin, J.; Liang, K. A comprehensive review on convolutional neural network in machine fault diagnosis. *Neurocomputing* **2020**, *417*, 36–63. [\[CrossRef\]](#)
12. Zhang, J.; Huang, C.; Chow, M.Y.; Li, X.; Tian, J.; Luo, H.; Yin, S. A Data-model Interactive Remaining Useful Life Prediction Approach of Lithium-ion Batteries Based on PF-BiGRU-TSAM. *IEEE Trans. Ind. Inform.* **2023**, 1–11. [\[CrossRef\]](#)
13. Zhang, X.; Wang, T. Elastic and reliable bandwidth reservation based on distributed traffic monitoring and control. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 4563–4580. [\[CrossRef\]](#)
14. Zhang, J.; Li, X.; Tian, J.; Luo, H.; Yin, S. An integrated multi-head dual sparse self-attention network for remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2023**, *233*, 109096. [\[CrossRef\]](#)
15. Solorio-Fernández, S.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F. A review of unsupervised feature selection methods. *Artif. Intell. Rev.* **2020**, *53*, 907–948. [\[CrossRef\]](#)
16. Yang, Z.; Xu, B.; Luo, W.; Chen, F. Autoencoder-based representation learning and its application in intelligent fault diagnosis: A review. *Measurement* **2022**, *189*, 110460. [\[CrossRef\]](#)
17. Ma, Y.; Shi, H.; Tan, S.; Tao, Y.; Song, B. Consistency regularization auto-encoder network for semi-supervised process fault diagnosis. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–15. [\[CrossRef\]](#)
18. Principi, E.; Rossetti, D.; Squartini, S.; Piazza, F. Unsupervised electric motor fault detection by using deep autoencoders. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 441–451. [\[CrossRef\]](#)
19. Chen, F.; Liu, L.; Tang, B.; Chen, B.; Xiao, W.; Zhang, F. A novel fusion approach of deep convolution neural network with auto-encoder and its application in planetary gearbox fault diagnosis. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2021**, *235*, 3–16. [\[CrossRef\]](#)
20. Yu, F.; Liu, J.; Liu, D.; Wang, H. Supervised convolutional autoencoder-based fault-relevant feature learning for fault diagnosis in industrial processes. *J. Taiwan Inst. Chem. Eng.* **2022**, *132*, 104200. [\[CrossRef\]](#)
21. Akcay, S.; Atapour-Abarghouei, A.; Breckon, T.P. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Proceedings of the Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Revised Selected Papers, Part III 14; Springer: Berlin/Heidelberg, Germany, 2019; pp. 622–637.*
22. Nabhan, A.; Ghazaly, N.; Samy, A.; Mousa, M. Bearing fault detection techniques-a review. *Turk. J. Eng. Sci. Technol.* **2015**, *3*, 1–18.
23. Zhang, F.; Zhu, Y.; Zhang, C.; Yu, P.; Li, Q. Abnormality Detection Method for Wind Turbine Bearings Based on CNN-LSTM. *Energies* **2023**, *16*, 3291. [\[CrossRef\]](#)
24. Neupane, D.; Kim, Y.; Seok, J. Bearing fault detection using scalogram and switchable normalization-based CNN (SN-CNN). *IEEE Access* **2021**, *9*, 88151–88166. [\[CrossRef\]](#)
25. Polikar, R.; Upda, L.; Upda, S.S.; Honavar, V. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. Syst. Man. Cybern. Part C (Appl. Rev.)* **2001**, *31*, 497–508. [\[CrossRef\]](#)
26. Carino, J.A.; Delgado-Prieto, M.; Iglesias, J.A.; Sanchis, A.; Zurita, D.; Millan, M.; Redondo, J.A.O.; Romero-Troncoso, R. Fault detection and identification methodology under an incremental learning framework applied to industrial machinery. *IEEE Access* **2018**, *6*, 49755–49766. [\[CrossRef\]](#)
27. Delgado-Prieto, M.; Carino, J.A.; Saucedo-Dorantes, J.J.; Osornio-Rios, R.A.; Romeral, L.; Troncoso, R.R. Novelty detection based condition monitoring scheme applied to electromechanical systems. In *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; Volume 1, pp. 1213–1216.*
28. Xing, L.; Wenshuang, W.; Jianyin, Z.; Min, Z. Research on an adaptive online incremental ELM fault diagnosis model. *Syst. Eng. Electron.* **2021**, *43*, 2678–2687.
29. Jalayer, M.; Kaboli, A.; Orsenigo, C.; Vercellis, C. Fault detection and diagnosis with imbalanced and noisy data: A hybrid framework for rotating machinery. *Machines* **2022**, *10*, 237. [\[CrossRef\]](#)
30. 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*, 1411–1423. [\[CrossRef\]](#)
31. Wang, J.; Lu, S.; Wang, S.H.; Zhang, Y.D. A review on extreme learning machine. *Multimed. Tools Appl.* **2022**, *81*, 41611–41660. [\[CrossRef\]](#)
32. Garbin, C.; Zhu, X.; Marques, O. Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimed. Tools Appl.* **2020**, *79*, 12777–12815. [\[CrossRef\]](#)
33. Belhaouari, S.B. Unsupervised outlier detection in multidimensional data. *J. Big Data* **2021**, *8*, 1–27.
34. Lou, X.; Loparo, K.A. Bearing fault diagnosis based on wavelet transform and fuzzy inference. *Mech. Syst. Signal Process.* **2004**, *18*, 1077–1095. [\[CrossRef\]](#)
35. Bechhoefer, E. Condition based maintenance fault database for testing diagnostics and prognostic algorithms. *MFPT Data* **2013**. Available online: <https://www.mfpt.org/fault-data-sets/> (accessed on 25 August 2023).

36. Zhang, J.; Zhang, K.; An, Y.; Luo, H.; Yin, S. An integrated multitasking intelligent bearing fault diagnosis scheme based on representation learning under imbalanced sample condition. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–12. [[CrossRef](#)] [[PubMed](#)]
37. Eren, L.; Ince, T.; Kiranyaz, S. A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *J. Signal Process. Syst.* **2019**, *91*, 179–189. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.