processes

Article

A Time–Frequency Residual Convolution Neural Network for the Fault Diagnosis of Rolling Bearings

Chenxi Wu [1,2,3], Rong Jiang [1,*], Xin Wu [1], Chao Zhong [1] and Caixia Huang [1]

1. School of Mechanical Engineering, Hunan Institute of Engineering, Xiangtan 411104, China; wuchenxi@hnie.edu.cn (C.W.); wuxin@hnu.edu.cn (X.W.); zhongchao@hnie.edu.cn (C.Z.); huanglq003@163.com (C.H.)
2. Hunan Provincial Engineering Laboratory of Wind Power Operation, Maintenance and Testing, Hunan Institute of Engineering, Xiangtan 411104, China
3. Hunan Provincial Key Laboratory of Vehicle Power and Transmission System, Hunan Institute of Engineering, Xiangtan 411104, China

* Correspondence: tefangchen@163.com

Abstract: A time–frequency residual convolution neural network (TFRCNN) was proposed to identify various rolling bearing fault types more efficiently. Three novel points about TFRCNN are presented as follows: First, by constructing a double-branch convolution network in the time domain and the frequency domain, the respective features in the time domain and the frequency domain were extracted to ensure the rich and complete feature representation of raw data sources. Second, specific residual structures were designed to prevent learning degradation of the deep network, and global average pooling was adopted to improve the network's sparsity. Third, TFRCNN was better than the other models in terms of prediction accuracy, robustness, generalization ability, and convergence. The experimental results demonstrate that the prediction accuracy rate of TFRCNN, trained using mixing load data, reached 98.88 to 99.92% after optimizing the initial learning rate and choosing the optimizer and loss function. It was verified that TFRCNN can adaptively learn to extract deep fault features, accurately identify bearing fault conditions, and overcome the limitations of classical shallow feature extraction and classification methods, as well as common convolution neural networks. Hence, this investigation revealed TFRCNN's potential for bearing fault diagnosis in practical engineering applications.

Keywords: deep learning; double branch; fault diagnosis; generalization ability; prediction accuracy; robustness; rolling bearings

check for updates

Citation: Wu, C.; Jiang, R.; Wu, X.; Zhong, C.; Huang, C. A Time–Frequency Residual Convolution Neural Network for the Fault Diagnosis of Rolling Bearings. *Processes* **2024**, *12*, 54. https://doi.org/10.3390/pr12010054

Academic Editor: Jie Zhang

Received: 24 November 2023
Revised: 18 December 2023
Accepted: 21 December 2023
Published: 25 December 2023

1. Introduction

A rolling bearing is a core rotary component of rotating machinery. Bearing failures caused by variable load impact, fatigue spall, mechanical wear, etc., can seriously affect the normal operation of rotating machinery. Therefore, studies on accurate bearing fault diagnosis methods are of great significance for high efficiency and adaption to the development of complex, intelligent, large-scale rotating machinery. Currently, fault diagnosis encompasses mainly classical and emerging deep learning methods.

Classical methods include feature extraction, feature dimensionality reduction, recognition and classification, etc. They start with statistical analysis [1,2], an autoregressive model (AR) [3], empirical mode decomposition (EMD) [4], Fourier transform (FT) [5], wavelet analysis [6,7], singular value decomposition (SVD) [8], and other methods to extract signal features which largely determine the effect of fault diagnosis. Then, feature dimensionality reduction is performed using methods such as principal component analysis (PCA) [9] and manifold learning [10] to remove noise and redundant components in order to obtain low-dimensional sensitive features. Finally, trained machine learning methods, such as artificial neural networks (ANNs) [11–13] and support vector machine

(SVM) [14–17], are used to identify and classify unknown fault features. Classical diagnostic methods combining feature extraction and pattern recognition have undoubtedly achieved remarkable results. However, with the increasing scale, complexity, and intelligence of rotating machinery, classical methods have exposed the following shortcomings in processing large multimodal uncertain signals:

1. There is a reliance on expert experience and special knowledge to manually extract fault features, and feature information is subjectively selected and incomplete.
2. A model trained with specific fault data is prone to overfitting, has poor generalization ability and versatility, and has difficulty meeting the diagnosis requirements of the complex background of big data.
3. Due to the separation of feature extraction from fault recognition, neither can be uniformly adjusted and optimized, resulting in a loss of fault information, and affecting the diagnostic results.
4. Back-propagation neural networks (BPNNs), SVMs, and other shallow networks have weak feature extraction ability and poor nonlinear fitting ability, making it difficult to mine comprehensive feature information and establish an accurate complicated mapping relationship between signals and fault types.

In recent years, deep learning has been successfully applied in the fields of speech image recognition and natural language processing due to its powerful automatic feature extraction capability, providing a new method of fault diagnosis. Proposed by Hinton G. in 2006 [18], deep learning simulates the learning process of the human brain by building a deep neural network, trains network neurons with big data samples, and adaptively learns the feature information hidden in the samples layer by layer, from low to high, to form more abstract high-level features so as to discover the distributed features of samples for recognition. Typical deep learning methods include convolutional neural networks (CNNs), deep belief networks (DBNs), autoencoder networks (AEs), recurrent neural networks (RNNs), etc. Because deep adaptive learning does not require manual feature extraction, it enhances the feature extraction capability and intelligence of the recognition process, eliminates environmental and human influence, avoids the uncertainty and complexity of classical diagnosis methods, and meets the requirements of adaptive fault feature extraction and classification in the context of big data. Therefore, it has been promptly studied in fault diagnosis [19–24]. Among these methods, CNN is prominent [25]; for example, a cyclic spectral-coherence-based CNN can achieve high diagnosis accuracy and better generalization ability by applying domain-related techniques to reduce the difficulty of feature learning and obtain high-level feature representations [26]. A common CNN with Gramian noise reduction can improve the performance of denoising and classification for bearing fault diagnosis [27]. By changing the feature weights of various convolution scales and picking out the key features, an adaptive multi-scale fully convolutional network showed its ability to demonstrate feature extraction, noise immunity, and robustness for bearing fault diagnosis [28]. As shown in Figure 1, a CNN consists of an input layer, convolution layers, pooling layers, fully connected layers, and a classifier; its convolution layer and pooling layer are alternately stacked to form a deep network. The functions of each layer are as follows:

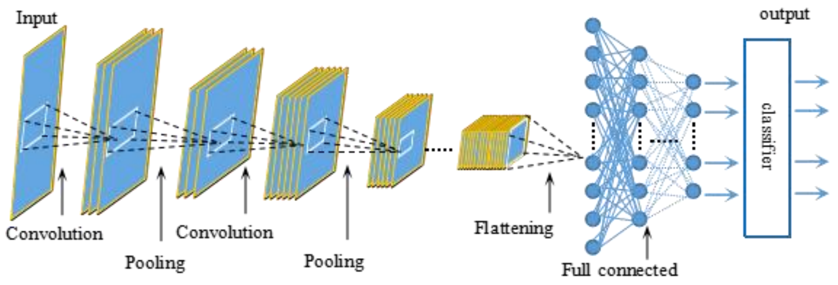


Figure 1. CNN basic structure.

Convolution layer: The convolution kernel performs local scanning and convolution computation on the input feature map to filter and obtain deeper features using the same weight and threshold, according to the stride. Then, the convolution output is nonlinearized via the activation function to improve model capability.

Pooling layer: This layer divides the activated feature map into multiple windows according to the sampling size, and it reduces the feature dimension by selecting the characteristic parameter each window as itself. Pooling can improve feature sparsity and network generalization ability.

Fully connected layer and classifier: by convoluting, activating, pooling, and flattening the output, it is recognized and classified using a fully connected layer and classifier.

Achieving adaptive distributed feature extraction and recognition classification through a deep network composed of the above function layers, CNN has the following characteristics:

1. The network is composed of multiple layers of filters;
2. Weight sharing, local receptive field, and pooling are used to reduce the complexity and overfitting;
3. Big data processing capability.

The current research on CNN for fault diagnosis involves model construction and algorithm selection, network structure and parameter optimization, raw data preprocessing and data representation, data input mode, etc. Results show that CNN has good fault accuracy, convergence, robustness, generalization, universality, etc. In general, research on a deep learning model such as CNN for fault diagnosis is on the rise, but it also faces the following major challenges:

1. The performance of the models cannot be strictly analyzed and proven. For example, it is difficult to discuss and interpret the robustness to improve the reliability of deep learning-based fault diagnosis models.
2. There are no uniform rules for setting the layers and network parameters depending only on expert experience and data characteristics.
3. Model convergence rate is restricted owing to the deep layers, nonlinear structure, and big data.
4. Further research for hybrid fault diagnosis methods is needed to be applied to the complex plant-wide industrial system.
5. Fault diagnosis models should be sufficiently generalized to meet the variable working condition requirements.

Aiming at the traits of rotating machinery, such as complex structure composition, big data monitoring, and operating condition, a time–frequency residual convolution neural network (TFRCNN) was proposed. This objective was to fully extract the diverse characteristics of the big state signals of rotating machinery bearings, preventing the background noise from weakening the bearing fault signals and mitigating learning degradation with the growth of network layers. Thus, measures were taken to accomplish TFRCNN effectiveness as follows: CNN was used to take advantage of its integration of deep feature extraction from big data and fault classification, the anti-interference ability of TFRCNN was strengthened by extracting the time and frequency domain features of the bearing state signals, and a residual structure was adopted as part of TFRCNN to solve the degradation of the deep network. Regarding TFRCNN, the biggest difference compared to other CNNs is the dual-branch structure within the time and frequency domains, which allows for avoiding complex feature extraction techniques, leading to high diagnosis accuracy and better generalization ability. Hence, TFRCNN has potential in practical engineering applications.

The contributions of TFRCNN drawn from the experiments are summarized below.

1. TFRCNN can extract rich and comprehensive time–frequency features and enhance anti-interference ability through its dual-branch structures within both time and frequency domains. Its diagnosis performance outperforms that of FSRCNN and TSRCNN just with a single branch of time or frequency.

2. TFRCNN achieves end-to-end fault diagnosis employing a convolution neural network to integrate feature extraction and fault classification, avoiding the disadvantages of conventional methods due to the separation of feature extraction and fault classification.
3. Original data directly fed into TFRCNN can avoid individual prejudice and the need for professional knowledge for feature extraction, which may lead to a loss of feature information.
4. TFRCNN solves the degradation problem of deep learning via residual networks. It also improves sparsity, generalization ability, and reduces prediction times via global average pooling.

Thus, TFRCNN has the excellent advantages of high prediction accuracy and fast speed, strong robustness, broad generalization ability, and good sparsity for bearing fault diagnosis.

The rest of the paper is organized as follows: TFRCNN is presented in Section 2. In Section 3, TFRCNN is validated by identifying the bearing fault locations and severities. Finally, the conclusions are drawn in Section 4.

2. Proposed Method

The complex mechanical structure, operating conditions, and big data monitoring of rotating machinery make the characteristic information of bearing faults complicated and diverse, making them suitable for deep distributed feature extraction from big data using a CNN. Because original bearing time signals contain sensitive characteristic information and the clustering of spectral signals can weaken noise interference, both were selected as the sample data. A residual network structure was adopted to prevent the learning degradation of the deep network while enhancing diagnostic effectiveness. Hence, a time–frequency residual convolution neural network, TFRCNN, was proposed.

2.1. TFRCNN

TFRCNN structure is shown in Figure 2.

TFRCNN consists of two convolution neural networks. One accepts one-dimensional time domain data, named the time signal residual convolution neural network (TSRCNN). The other accepts two-dimensional frequency domain data, named the frequency signal residual convolution neural network (FSRCNN).

TFRCNN carries out network learning and feature extraction from the two types of data, respectively. Each branch is successively stacked with a simple convolution layer, a maximum pooling layer, eight residual convolution blocks (see blocks I and II in Figure 2), a global pooling layer, and three fully connected layers, among which twenty convolution and fully connected layers are trained. The rectified linear unit (ReLU) activation function is used to update the network parameters in error back-propagation. Its linearity makes the gradient proportional to the ReLU output to avoid gradient disappearance.

Here, the principle of TSRCNN is illustrated assuming a bearing time domain sample containing 512 sampling values, written as a 512 $\times$ 1 array.

First, 64 time domain samples were fed into the first convolution layer as a batch. Each sample convolved with 64 kernels of 3 $\times$ 1, and 64 of 512 $\times$ 1 features were extracted. The setting number of features were extracted via convolution to denoise the input signal. The computation and the risk of network overfitting were reduced by sharing the convolution kernel weights.

Second, a pooling kernel of 3 $\times$ 1 slid on the convolution output features at two strides to select the maximum value in the window, renewing 64 of 256 $\times$ 1 features. These features were then transmitted into the residual convolution block. The pooling layer used down-sampling without parameters, which reduced the network scale and prevented overfitting.

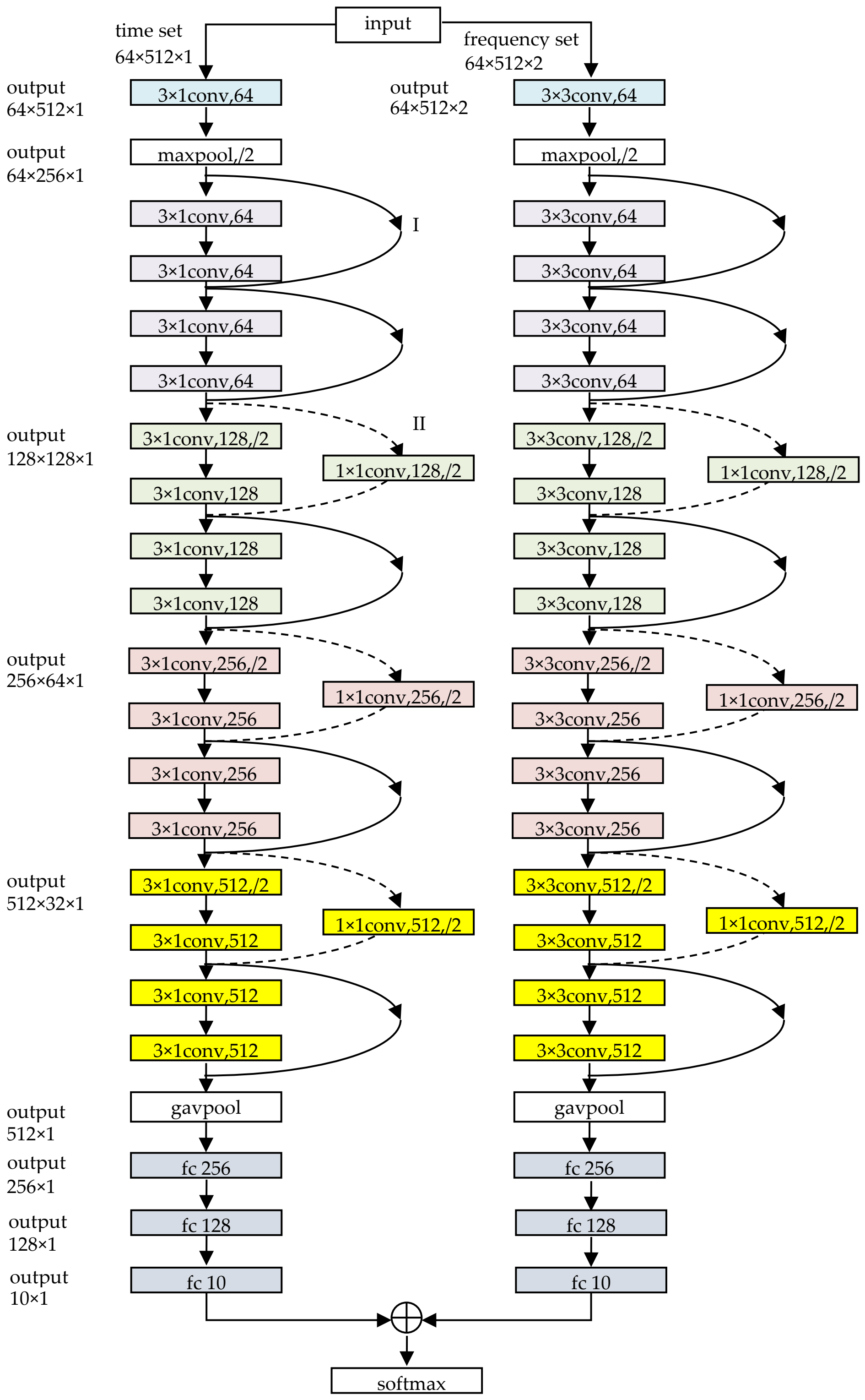


Figure 2. TFRCNN architecture.

A total of eight residual convolution blocks are The majority of TSRCNN is composed of eight residual convolution blocks. After a series of mapping through the residual convolution blocks, the number of features increased from 64 to 512, each with a size of 32 × 1. The use of a residual convolution block is to solve the degradation of ordinary convolution in deep network learning. The structure of a residual convolution block is described in Figure 3.

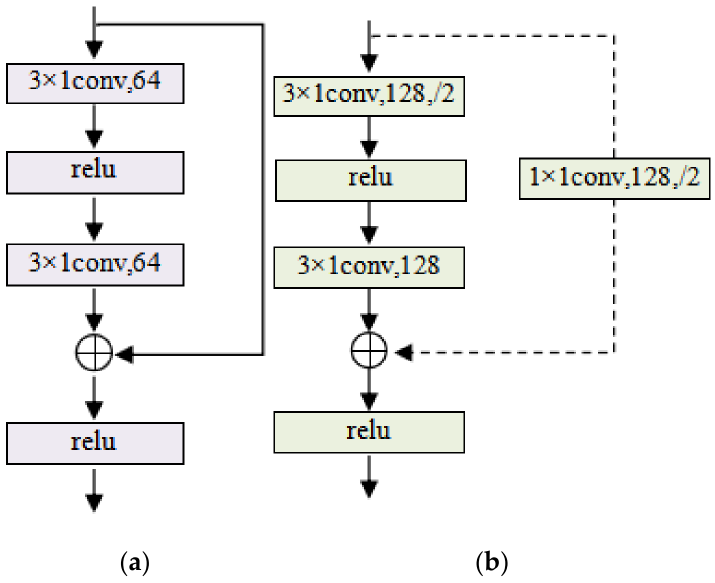


Figure 3. Residual convolution block: (**a**) identity residual block; (**b**) projection residual block.

Global average pooling averaged all the elements of each output feature of the last residual block, yielding 512 smoothing features as the input neurons of the first fully connected layer. Global average pooling can significantly decrease model parameters, accelerate convergence, save computing consumption, and prevent overfitting.

There were three fully connected layers behind the global average pooling layer. The input neurons were weighted and summed by the layers, and the output neuron vector was obtained after the activation of ReLU. The output vector of the first layer had 256 neurons and the second had 128 neurons. The output vector of the third layer contained 10 neurons, each representing the proportion of a sample in one of the 10 types of bearing faults.

The structure and principle of FSRCNN was similar to TSRCNN, but its frequency domain sample was a two-dimensional array, 512 × 2, composed of real and imaginary parts, so the corresponding convolution and pooling were two-dimensional calculations with a convolution kernel of 3 × 3. The algorithm of TFRCNN is described by Algorithm 1.

Algorithm 1. Algorithm of TFRCNN for training and testing

```
Input: $X_{input}$, $Y_{label}$, $N$, $learn\_rate$, $split\_rate$
Output: $W$, $accuracy$
1: begin
2: $X_{train}$, $Y_{train}$, $X_{test}$, $Y_{test}$ = split ($X_{input}$, $Y_{label}$, $split\_rate$)
3: $XFFT_{train}$, $XFFT_{test}$ = fft ($X_{train}$, $X_{test}$)
4: for $n = 0 \rightarrow N$ do                                          #TFRCNN training
5:     $out_{train}$ = $TFRCNN_{model}$.predict($X_{train}$, $XFFT_{train}$)
6:     loss = losses.crossentropy ($Y_{train}$-$out_{train}$)
7:     $gradient$ = gradients ($loss$, $variables$)
8:     $W$ = optimizers.nadam($learn\_rate$, $gradient$, $variables$)        #TFRCNN weight value
9: end for
10: $out_{test}$ = $TFRCNN_{model}$.predict($X_{test}$, $XFFT_{test}$)              #TFRCNN test
11: Probability = softmax($out_{test}$)
12: Prediction = argmax (Probability)
13: $l$ = len ($Y_{test}$)
14: $total\_real$ = sum (Prediction[$i$] = $Y_{test}[i]$)
15: $total\_num = l$
16: $accuracy = total\_real/total\_num$
17: print ('accuracy = ', $accuracy$)
18: end
```

2.2. Main Blocks

2.2.1. Residual Block

As the number of network layers increased, the gradient turned weak or even disappeared, and the capability of learning and prediction subsequently became worse. Therefore, the residual convolution network shown in Figure 3 was applied to solve the degradation of the deep network.

The residual convolution networks shown in Figure 3a,b represent two types of residual blocks (I and II) as illustrated in Figure 2. These networks are composed of a two-layer convolution-stacked residual mapping (left) and a shortcut connection (right). Hence, the risk of network degradation could be eliminated by weakening the relationship between layers using the interlayer connection of the shortcut branch.

In Figure 3a, both the input shape of the residual block and the output shape of the residual mapping are 64 × 256 × 1, so an identity shortcut was used for a shortcut connection, directly adding the input of the residual block and the output of the residual mapping as the output of the residual block.

In Figure 3b, the output shape of the residual mapping is 128 × 128 × 1, inconsistent with the input shape 64 × 256 × 1 of the residual block, so the projection shortcut with a convolution kernel shape of 128 × 1 × 1 was used to convolve with the input of the residual block at two strides, which made its output shape identical to the output of residual mapping, and both were then added together as the output of the residual block.

2.2.2. Normalization Block

After the input sample passed through the first convolution layer of TFRCNN, the convolution output features were normalized using the batch normalization (BN) layer to meet the distribution with a mean of 0 and variance of 1, as shown in Figure 4. The application of a BN block partially ruled out the risk of gradient disappearance or explosion and accelerated network convergence.

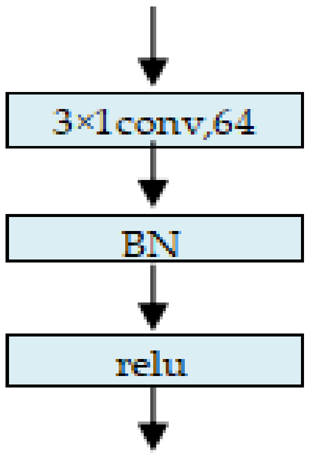


Figure 4. Normalization block.

2.2.3. Softmax Classifier

The softmax classifier is shown in Figure 5. It is a normalized exponent function and was used to exponentially transform each neuron in the output vector of the fully connected layer, increase the discrimination from each other, and normalize the neuron value between 0 and 1. The converted vector represents the probability distribution that the test sample belongs to various faults, and the index number of the maximum probability element was picked out as the fault category of the sample. As such, the sample feature extracted using TFRCNN was converted into the probability distribution of various faults and then classified.

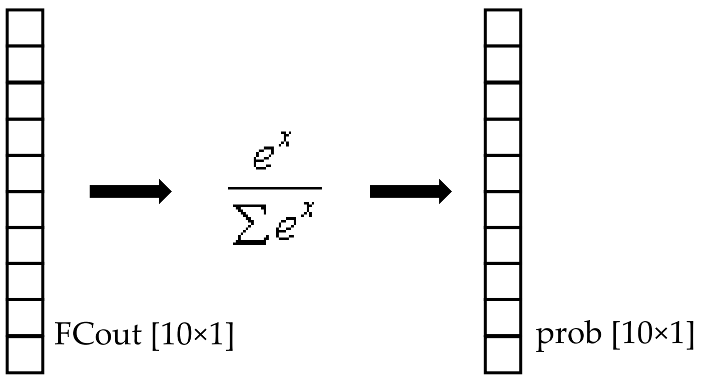


Figure 5. Softmax classifier.

3. Experiments and Results

In this section, TFRCNN is compared with TSRCNN, FSRCNN, VGG13, and other models.

3.1. Environment Configuration

The configurations for the experiment are listed in Table 1.

Table 1. Environment configuration.

Item	Version	Function
Windows10	Win10 Home	Operating system
CPU	Intel (R) Core (TM) i5-9400 2.90 GHz	
RAM	SODIMM, 24.0 GB	
GPU	NVIDIA GeForce GTX 1660 Ti 6.0 GB	
Cuda	10.1	GPU operation platform
Anaconda	2.1.1	Package and environment manager
Python	3.7	Programming language
Pycharm	2021.3	Python IDE
Third-party library	Tensorflow2.1	Deep learning function library
	Numpy	Data sort, merging, packaging
	Scipy	Open ". mat" format data
	Sklearn	Dividing train and test sets

3.2. Experimental Data

The experimental data were taken from the bearing data center at the Case Western Reserve University (CWRU). The CWRU bearing database was used because it is a public and authoritative database with a comprehensive range of bearing fault types. The test rig is shown in Figure 6. It consists of a 2-horsepower motor, a torque transducer, a dynamometer, and control electronics. The experiment samples were acquired at the 12 o'clock position at the drive end of the motor housing, with a sampling frequency of 12 KHz, and they were divided into four types of signals, including ball defect, inner race defect, outer race defect, or normal according to the fault location. There were three types of signals, with 0.1778 mm, 0.3556 mm, and 0.5334 mm defect points based on the fault size. The samples could consequently be categorized into 10 types of bearing state signals, labeled 0–9. Table 2 lists its categories and labels.

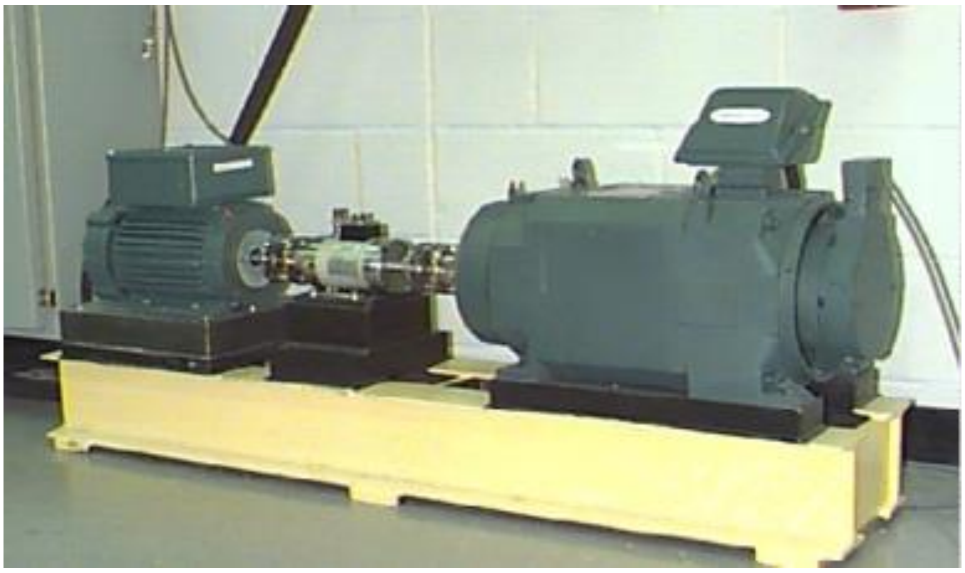


Figure 6. Rolling bearing test rig.

Table 2. Sample categories and labels.

Sample	Fault Location	Fault Size (mm)	Label
B007	Ball	0.1778	0
B014	Ball	0.3556	1
B021	Ball	0.5334	2
IR007	Inner race	0.1778	3
IR014	Inner race	0.3556	4
IR021	Inner race	0.5334	5
OR007	Outer race	0.1778	6
OR014	Outer race	0.3556	7
OR021	Outer race	0.5334	8
Normal			9

In Table 3, DS0, DS1, DS2, and DS3 respond to the load powers of 0, 735 w, 1470 w, and 2205 w, respectively, and DS4 is a collection of all the loads. The training and test sets for each dataset were composed of 10 types of samples, listed in Table 2.

Table 3. Experiment datasets.

Dataset	Load Power (W)	Speed (rpm)	Total Samples	Training Set	Test Set
DS0	0	1792	3048	2048	1000
DS1	735	1772	3048	2048	1000
DS2	1470	1750	3048	2048	1000
DS3	2205	1730	3048	2048	1000
DS4	collection	collection	12,192	2048	10,144

3.3. Model Training and Testing

The training and diagnostic flow of TFRCNN is illustrated in Figure 7.

- Step 1. Network initialization. Randomly generate weights w and bias b, set initial learning rate lr, and the number of training iteration N.
- Step 2. Network output. The training set is passed forward through the network, and, finally, the network output function of w and b is obtained.
- Step 3. Gradient derivation. The gradient is the partial derivative of the network loss function with respect to w and b. The loss function is the difference between the network output function and the real value.

- Step 4. Network update. Optimize the learning rate, calculate the gradient, and update network parameters w and b.
- Step 5. Repeat Steps 2 to 4, and train TFRCNN from the 1st till the Nth.
- Step 6. Model testing. Test TFRCNN and evaluate its fault diagnosis performance.

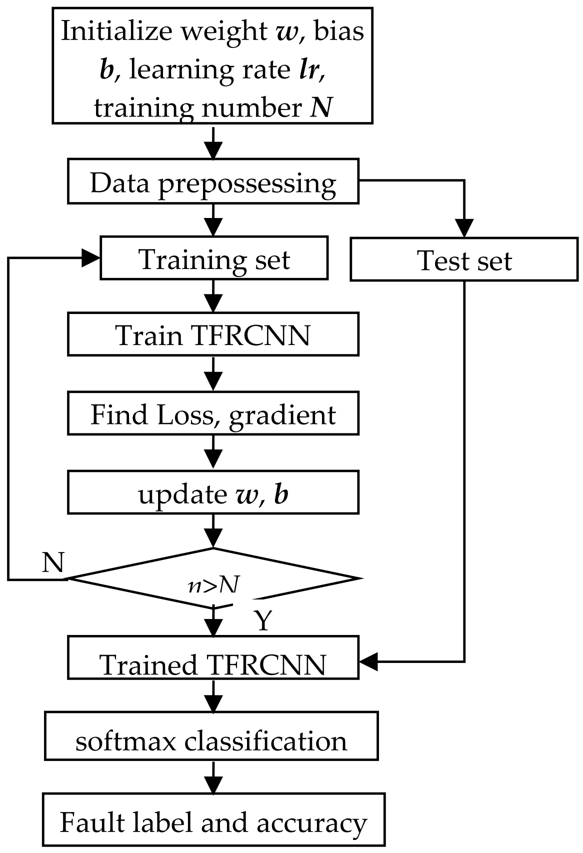
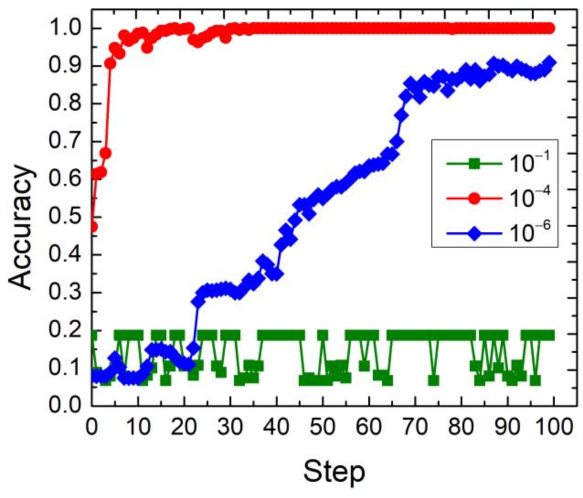


Figure 7. TFRCNN training and diagnostic flow chart.

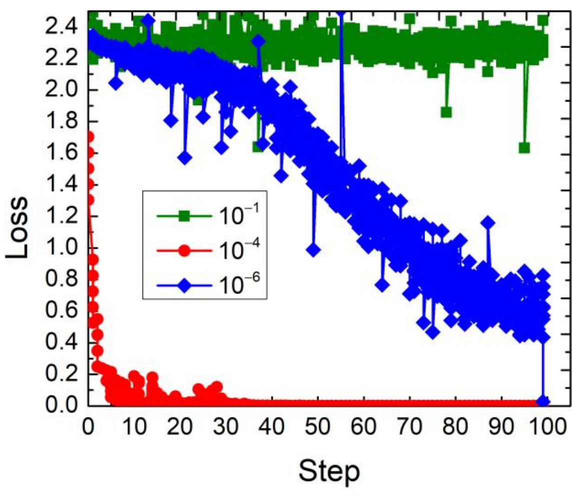
The model performance is affected by the initial learning rate, loss function, and optimizer. Some training and loss curves of TFRCNN, TSRCNN, and FSRCNN are shown in Figures 8–10 with changes in the initial learning rate, loss function, and optimizer.

The optimized training and loss curves for three models were identified via experiments, as shown in Figure 11.

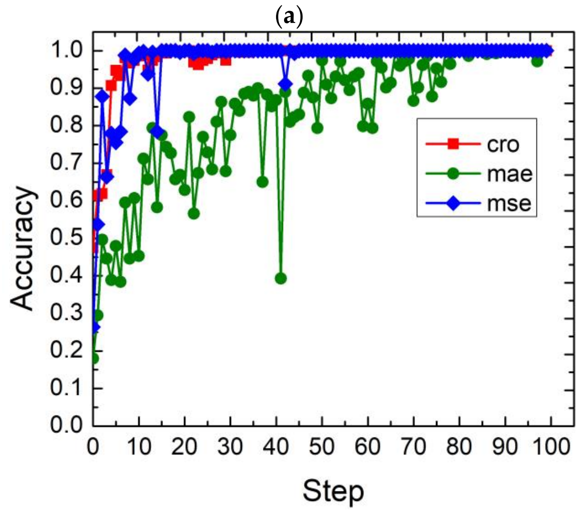
After extensive experiments, the optimized initial learning rate, loss function, and optimizer were determined by comprehensively considering their impact on the models. When the initial learning rate was 10^{-4}, TFRCNN could be optimized by adopting the nadam optimizer and cross-entropy loss function. Similarly, with an initial learning rate of 10^{-4}, both TSRCNN and FSRCNN could be optimized by using the adam optimizer and cross-entropy loss function. Due to the dual branch of the time–frequency feature extraction for TFRCNN, features were enriched and improved from the sources, ensuring the effectiveness of features. Additionally, global average pooling was adopted for the model, which can save computational power and time. During the training process, TFRCNN could quickly converge the loss function to the global optimal solution, thereby reducing the training steps, improving convergence speed, and saving training time. Among the three models, the training accuracy of TFRCNN was higher than that of FSRCNN and TSRCNN, and its convergence speed was close to that of FSRCNN but much faster than that of TSRCNN. Table 4 presents the number of steps and training time for model optimization. According to Figure 10a, when the training accuracy was first over 97% and remained stable at that level or higher in later iterations. The corresponding training step was considered as the number of iterations.



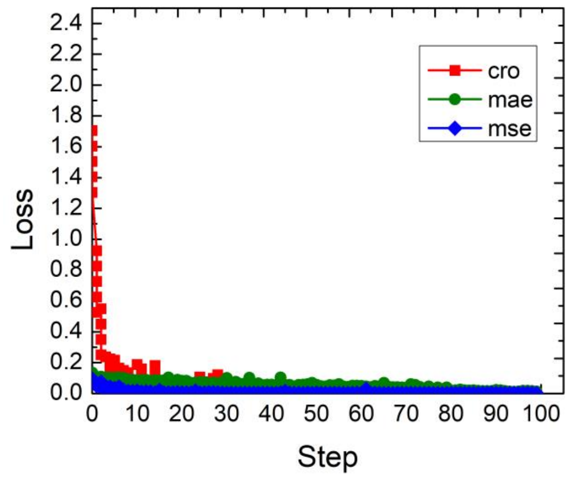
(**a**)



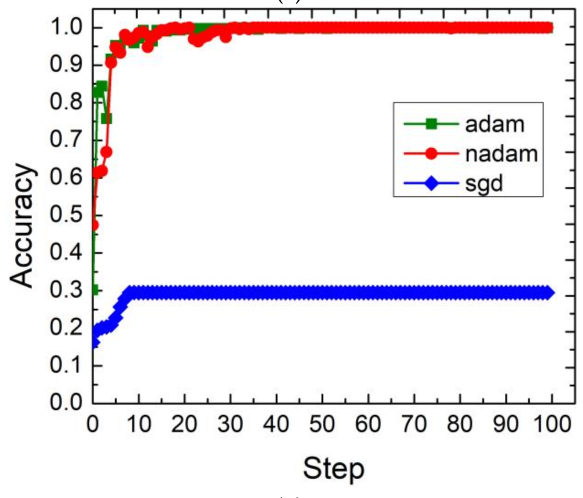
(**b**)



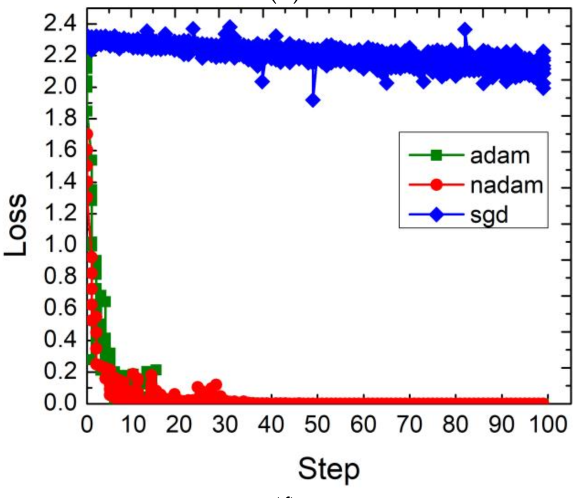
(**c**)



(**d**)



(**e**)



(**f**)

Figure 8. TFRCNN training accuracy and loss: (**a**) training accuracy with the learning rate; (**b**) training loss with the learning rate (cross entropy, nadam); (**c**) training accuracy with the loss function; (**d**) training loss with the loss function (lr = 10^{-4}, nadam); (**e**) training accuracy with the optimizer; (**f**) training loss with the optimizer (lr = 10^{-4}, cross entropy). nadam (nesterov accelerated gradient and adaptive moment estimation); adam (adaptive moment estimation); sgd (stochastic gradient descent); mse (mean squared error); mae (mean absolute error).

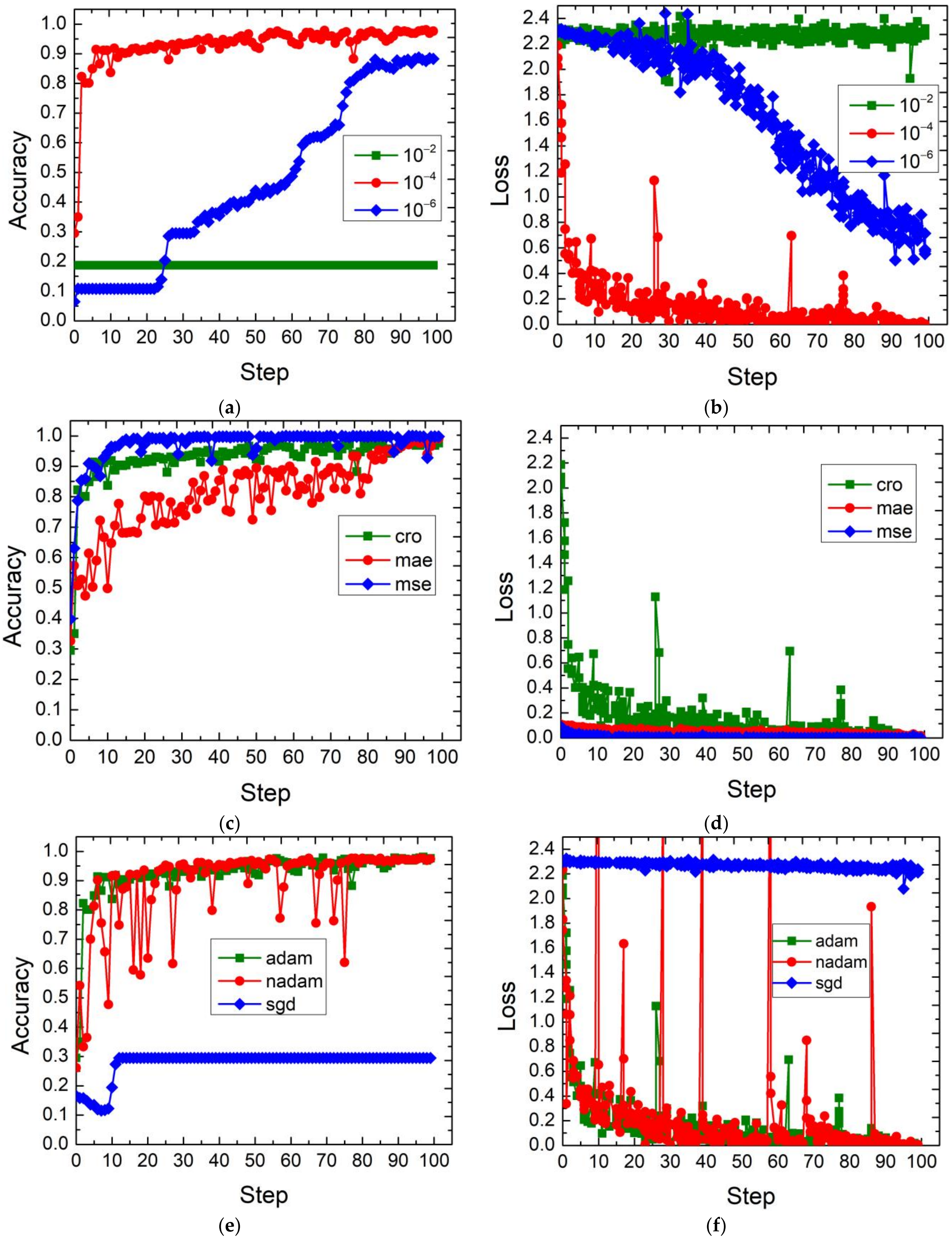


Figure 9. TSRCNN training accuracy and loss: (**a**) training accuracy with the learning rate; (**b**) training loss with the learning rate (cross entropy, nadam); (**c**) training accuracy with the loss function; (**d**) training loss with the loss function (lr = 10^{-4}, nadam); (**e**) training accuracy with the optimizer; (**f**) training loss with the optimizer (lr = 10^{-4}, cross entropy).

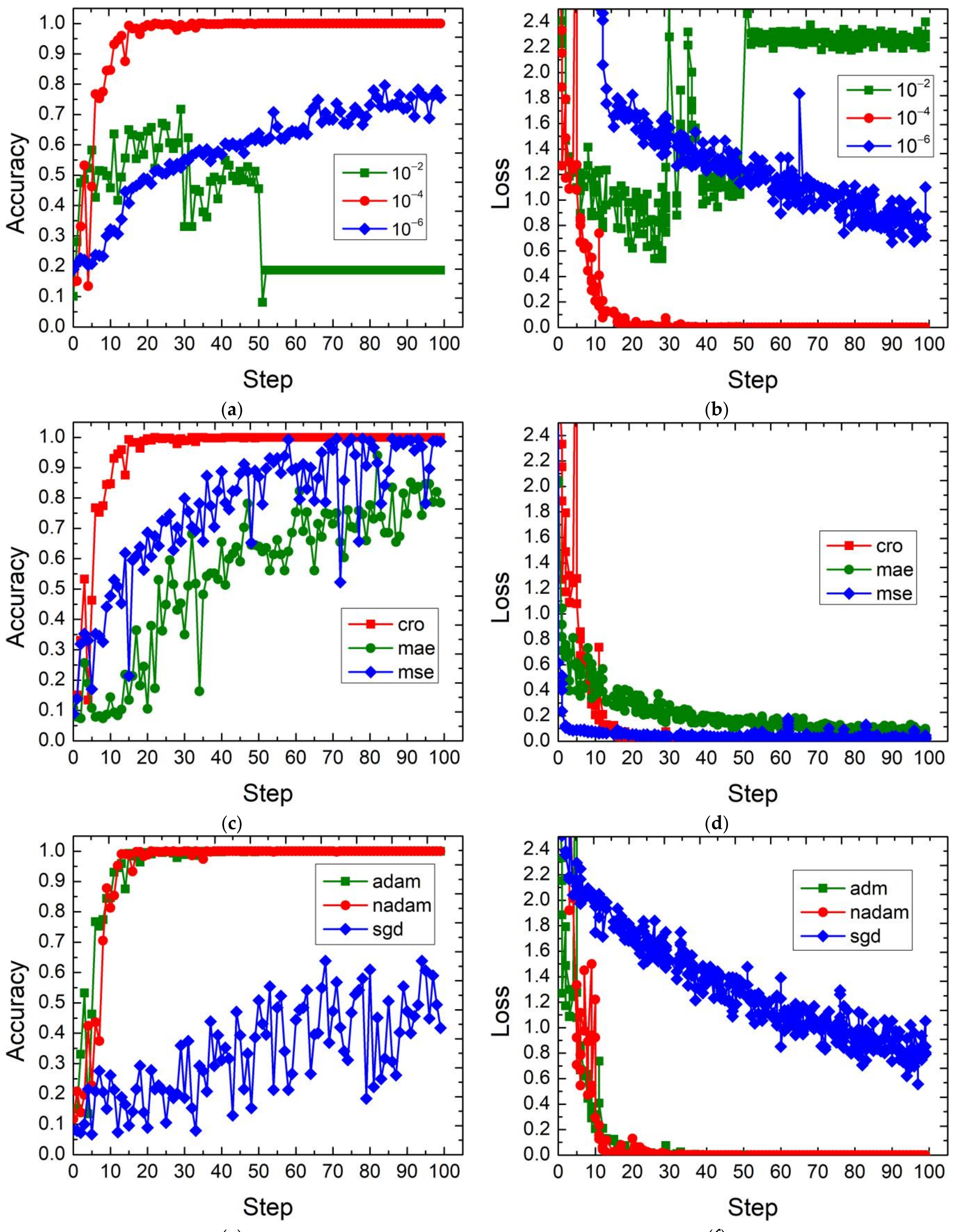
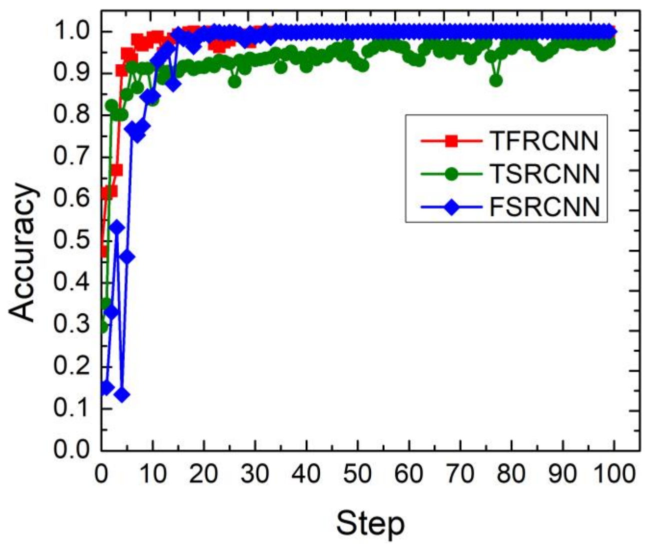
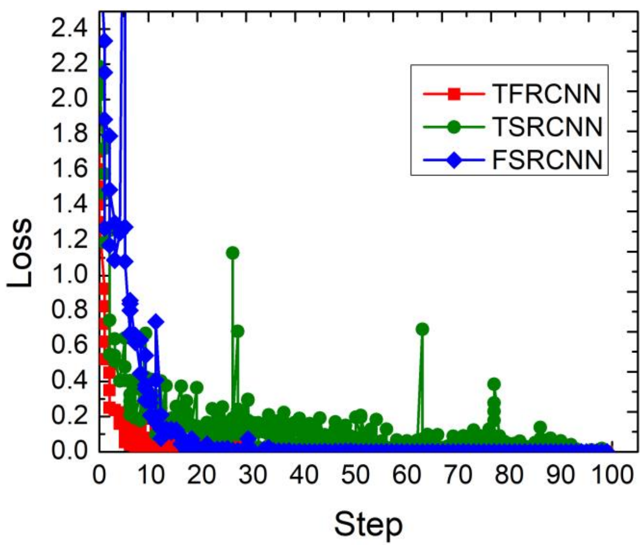


Figure 10. FSRCNN training accuracy and loss curves: (**a**) training accuracy with the learning rate; (**b**) training loss with the learning rate (cross entropy, nadam); (**c**) training accuracy with the loss function; (**d**) training loss with the loss function (lr = 10^{-4}, nadam); (**e**) training accuracy with the optimizer; (**f**) training loss with the optimizer (lr = 10^{-4}, cross entropy).



(**a**)



(**b**)

Figure 11. Optimized training accuracy and loss curves for TFRCNN, TSRCNN, and FSRCNN: (**a**) optimized training accuracy curves; (**b**) optimized training loss curves.

Table 4. Steps and training time for optimization.

Model	Iterations	Time (s)
TFRCNN	20	80
TSRCNN	150	250
FSRCNN	50	60

The initial learning rate, optimizer, loss function, and other parameters of the models are outlined in Table 5.

Table 5. Model settings.

Model	TFRCNN	TSRCNN	FSRCNN
Net parameter	15,167,754	3,844,234	11,323,520
Preprocessing	normalization, FFT	normalization	FFT
Batch sample	64	64	64
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
Decay coefficient	0.99	0.99	0.99
Optimizer	nadam	adam	adam
Loss function	cross entropy	cross entropy	cross entropy
Step	300	300	300

3.4. Model Validation

3.4.1. Prediction of Model Trained under a Specific Load for the Load Case

TFRCNN was trained with the training sets from DS0 toDS4, as shown in Table 3, and was tested with the respective test sets. The confusion matrices of the test results are provided in Figure 12a–e.

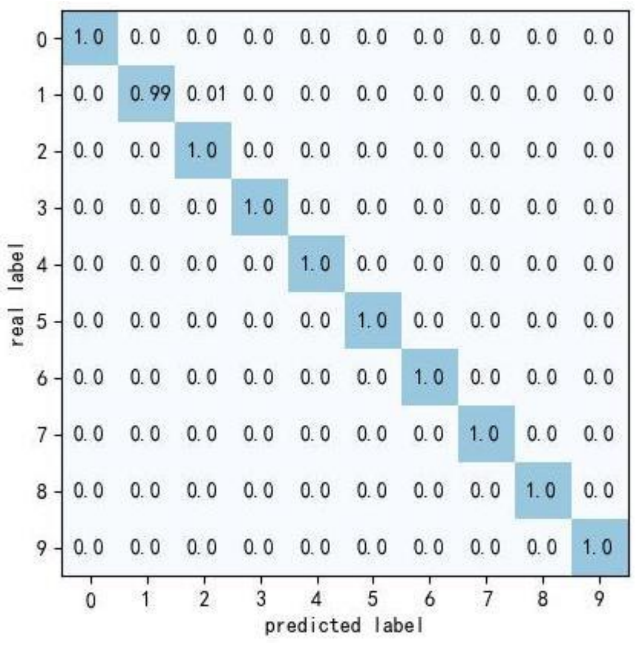
From the confusion matrices, it can be concluded that the prediction accuracy of TFRCNN is as high as 99.9% under each load condition; namely, only a sample out of 1000 was misjudged. For example, under condition of load 0 in Figure 12a, only label 1 out of 1000 samples was misjudged as label 2. TSRCNN and FSRCNN were trained and tested in the same manner as TFRCNN.

Figure 13 shows the prediction accuracy of the three models for 10 repeated tests with their own test sets after being trained with different load datasets, from DS0 to DS4. The mean accuracy and standard deviation are shown in Table 6. Figure 14 shows the mean accuracy and corresponding error of the three models.

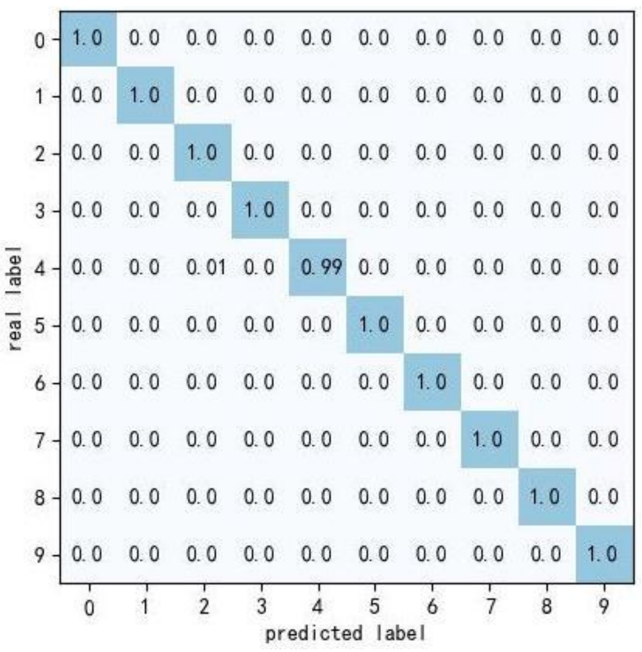
Table 6. Mean accuracy and standard deviation of models under a specific load.

Dataset	Mean (%)			Standard Deviation		
	TFRCNN	TSRCNN	FSRCNN	TFRCNN	TSRCNN	FSRCNN
DS0	99.96	99.92	98.50	0	0	0
DS1	99.86	99.89	99.38	0.001187715	0.00028	0
DS2	99.94	99.81	99.67	0.000632456	0.000193	0
DS3	99.93	99.70	99.57	0.000790569	0.000169	0
DS4	99.52	97.88	99.31	0.004389685	0.001053	0

On the whole, the model with the highest prediction capability was TFRCNN, followed by FSRCNN and TSRCNN. The prediction accuracy of TFRCNN stood at 99.52–99.96%, higher than 97.88–99.92% for TSRCNN, and 98.50–99.67% for FSRCNN, as seen in Table 6, along with the high stability measured by the standard deviations. This indicates that TFRCNN was not sensitive to the load changes, which demonstrates that TFRCNN has strong robustness, high prediction accuracy, and wide generalization for all types of loads. Therefore, TFRCNN was better than the other two models. The reason for this is that TFRCNN adopts a double-branch structure with time and frequency domains, which can fully extract complex and diverse bearing fault characteristics, and deeply fuse and efficiently identify bearing faults. TFRCNN has better performance than FSRCNN and TSRCNN, which have a single structure.

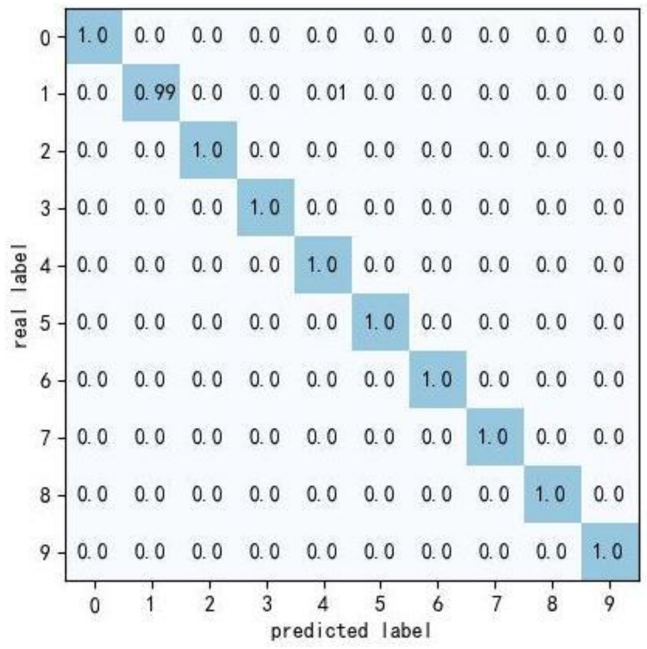


(**a**)

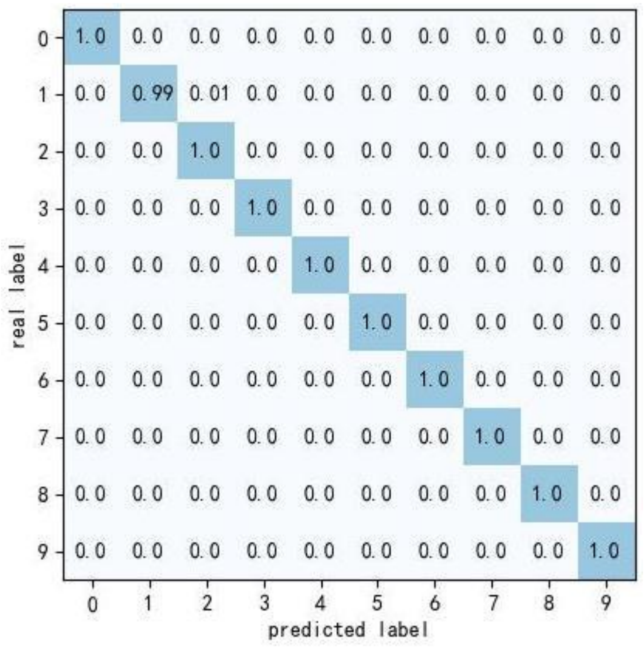


(**b**)

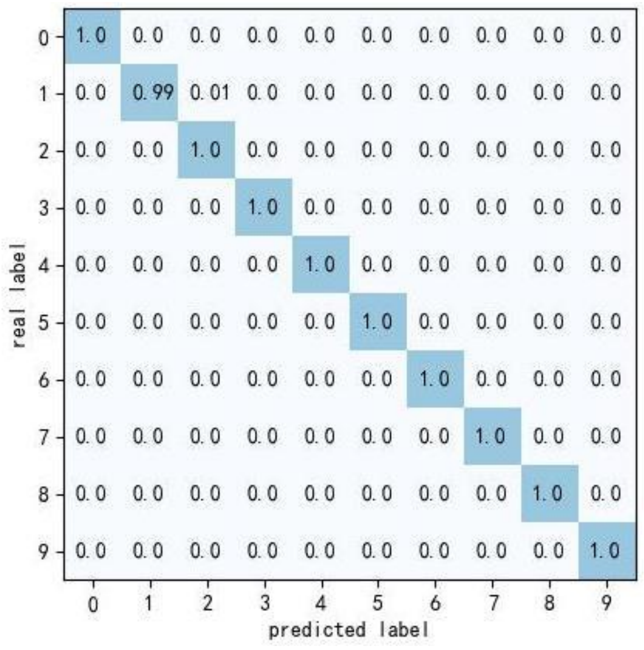
Figure 12. *Cont.*



(**c**)

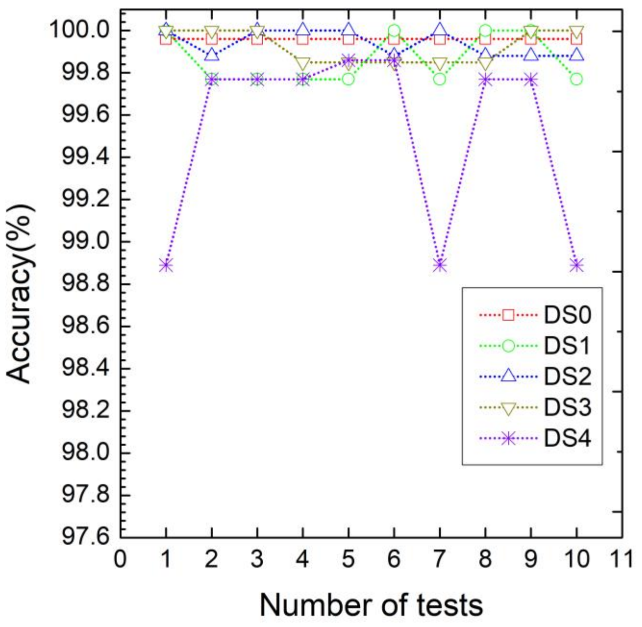


(**d**)

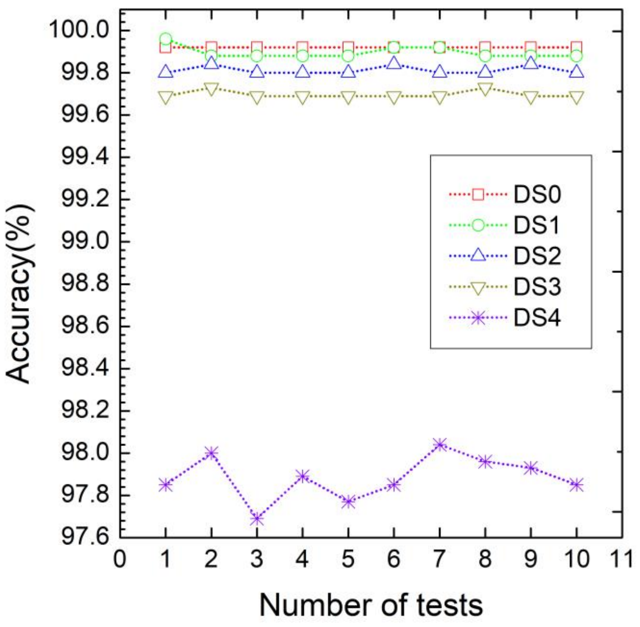


(**e**)

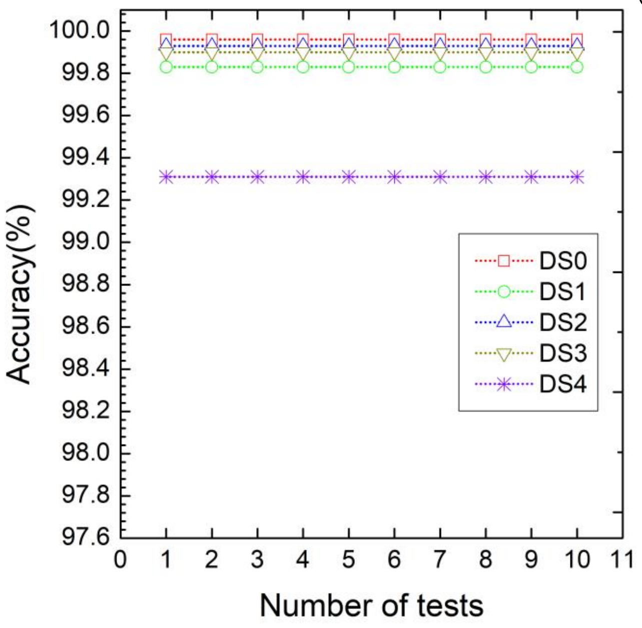
Figure 12. Confusion matrices for the test set: (**a**) DS0; (**b**) DS1; (**c**) DS2; (**d**) DS3; (**e**) DS4.



(**a**)



(**b**)



(**c**)

Figure 13. Accuracy of the models for 10 repeated tests under a specific load: (**a**) TFRCNN; (**b**) TSRCNN; (**c**) FSRCNN.

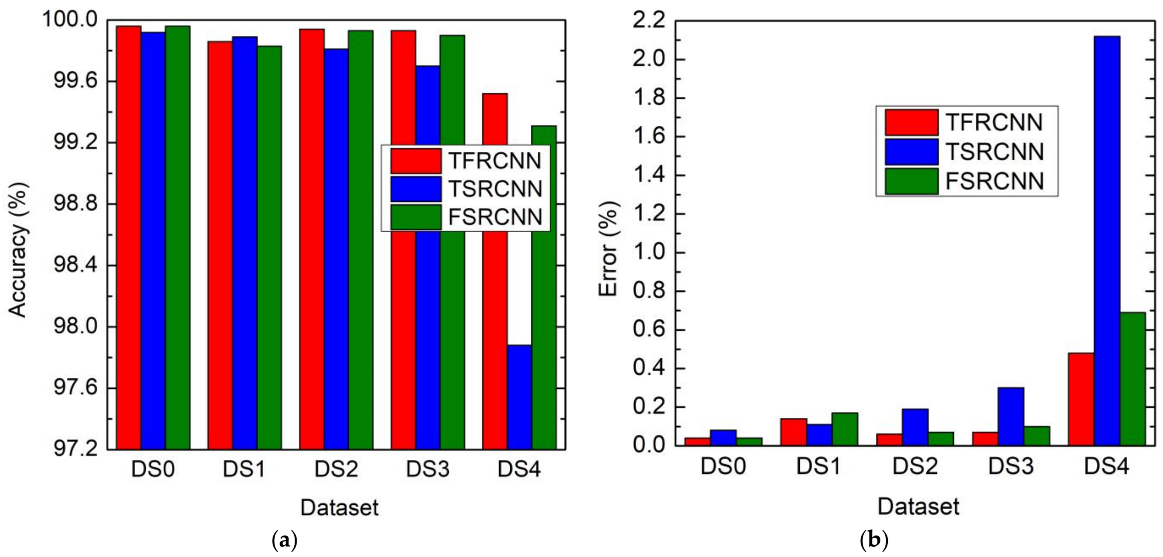


Figure 14. Mean accuracy and error of the models under a specific load: (**a**) mean accuracy; (**b**) mean error.

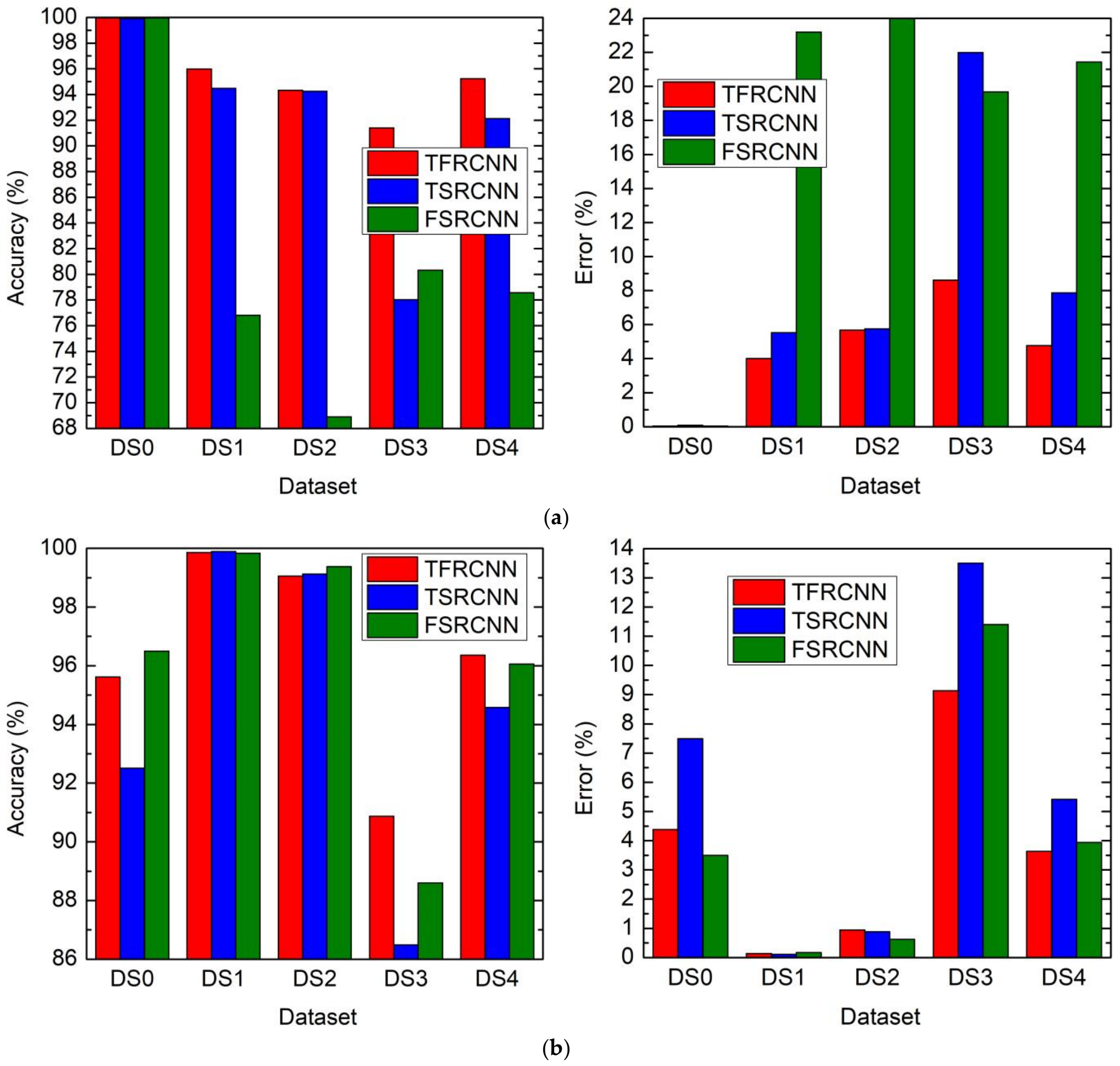
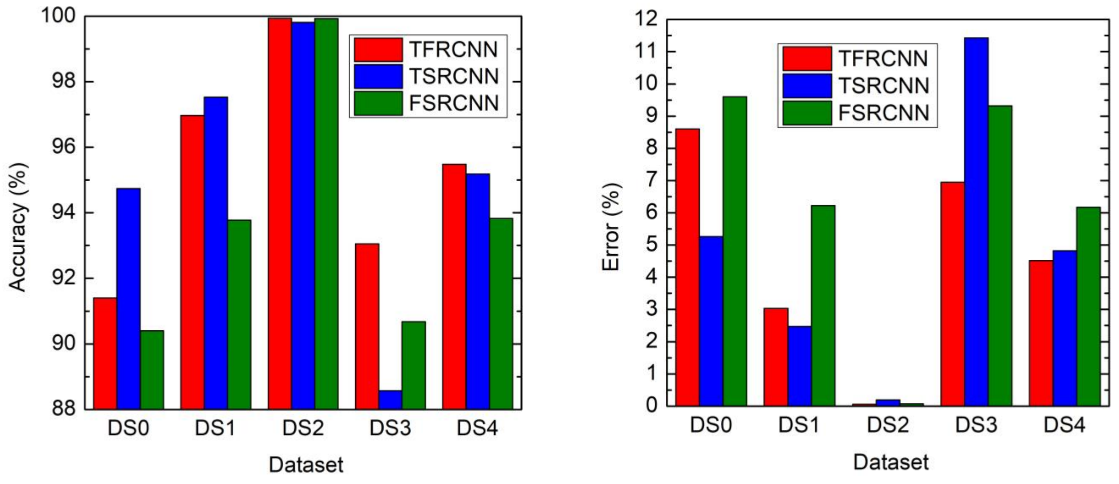
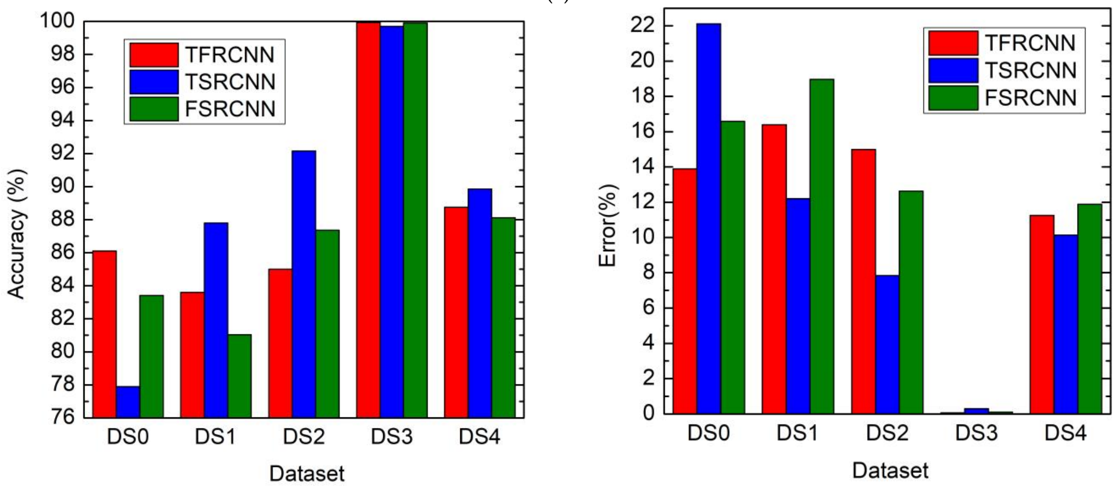


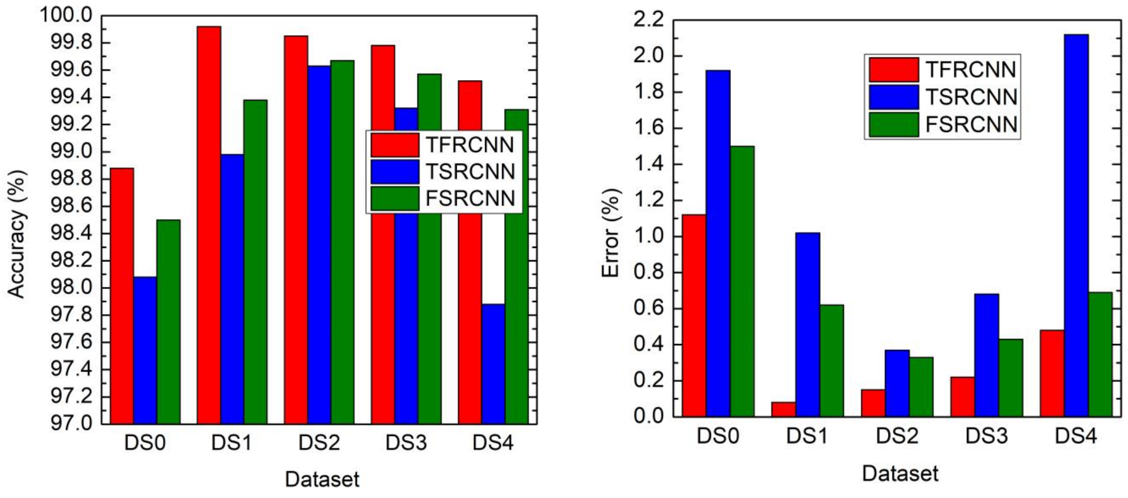
Figure 15. *Cont.*



(**c**)



(**d**)



(**e**)

Figure 15. Mean accuracy and error of the models trained with the following training sets: (**a**) DS0; (**b**) DS1; (**c**) DS2; (**d**) DS3; (**e**) DS4.

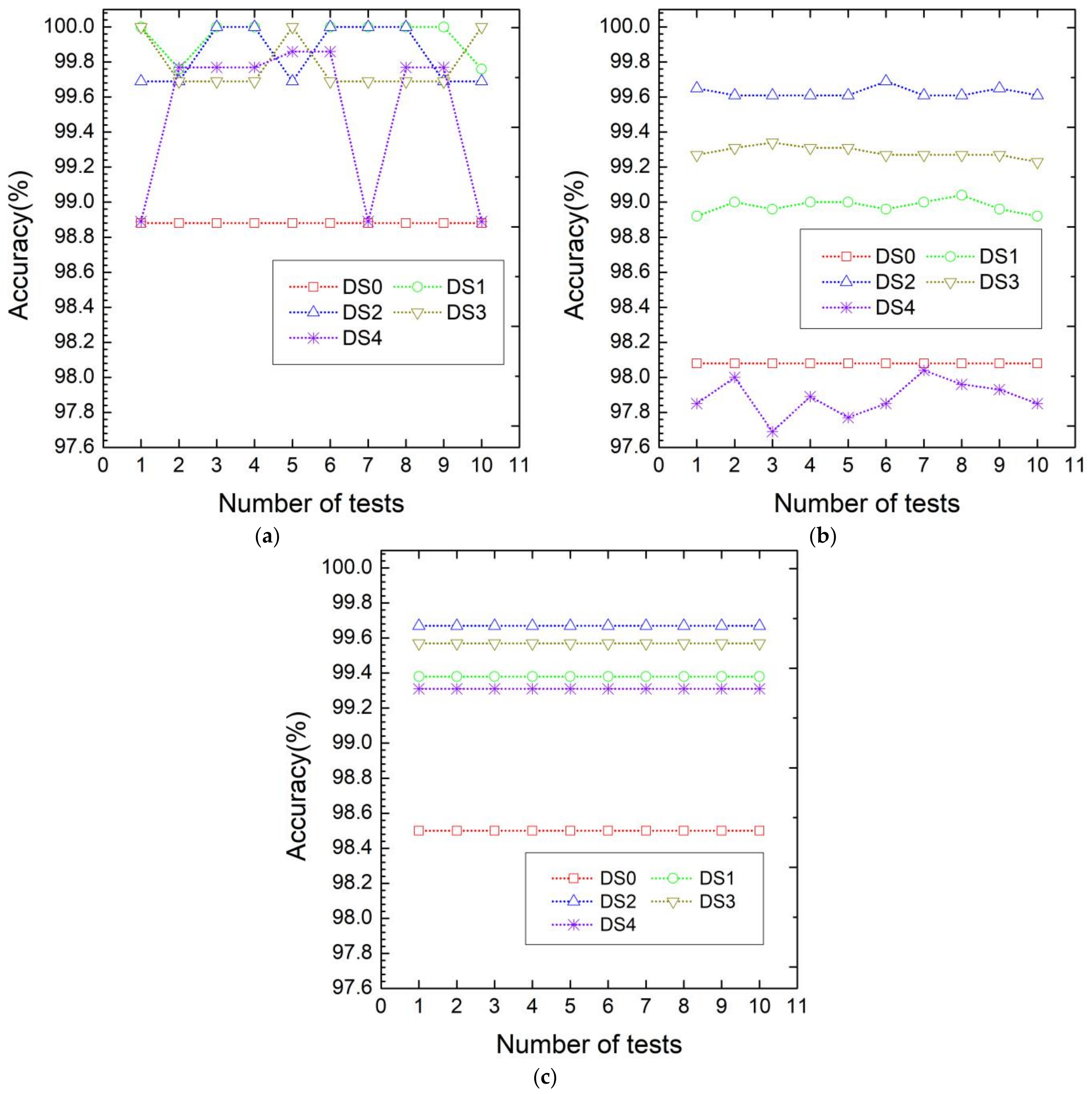


Figure 16. Accuracy of the models trained with DS4 for 10 repeated tests: (**a**) TFRCNN; (**b**) TSRCNN; (**c**) FSRCNN.

Table 7. Mean accuracy and standard deviation of the models trained with DS4.

Dataset	Mean (%)				Standard Deviation			
	TFRCNN	TSRCNN	FSRCNN	VGG13	TFRCNN	TSRCNN	FSRCNN	VGG13
DS0	98.88	98.08	98.50	97.73	0	0	0	0
DS1	99.92	98.98	99.38	99.25	0.001159	0.000386	0	0
DS2	99.85	99.63	99.67	99.25	0.001634	0.00028	0	0
DS3	99.78	99.32	99.57	99.61	0.001497	0.000317	0	0
DS4	99.52	97.88	99.31	99.01	0.00439	0.001053	0	0

From Figure 15a–d, it can be seen that when the load increased, the prediction capability of the models trained with a single load dataset decreased. For models trained with DS0, DS1, and DS2, the prediction accuracy of TFRCNN remained above 90.87% (see Figure 15b), which was superior to FSRCNN and TSRCNN. The prediction accuracy of the three models trained with DS3 was not high but had a higher fluctuation. Since a big impact and noise under heavy loads aggravate the risk of data contamination and result in specificity enhancement and lack of sparsity of the sample data, the models trained with such data have overfitting and narrow generalization ability, thus reducing the prediction accuracy for the other load cases.

In Figure 15e, TFRCNN, FSRCNN, and TSRCNN trained with mixing load data DS4 had better overall prediction capability for each load dataset than when trained with a single load dataset. Among the three models, TFRCNN had the highest prediction accuracy, ranging from 98.88 to 99.92%, with small fluctuation and good stability measured via standard deviations (see Table 7). The strong robustness and wide generalization of TFRCNN were confirmed again, and the validity of the double-branch structure and global average pooling design of TFRCNN was further verified as well.

Hence, from the aspects of prediction accuracy and generalization, TFRCNN trained with mixing load data is the best model.

3.4.2. TFRCNN Compared with VGG

As mentioned above, TFRCNN was optimized with the nadam optimizer, cross-entropy loss function, and an initial learning rate of 10^{-4}. Similarly, VGG13 was optimized using the adam optimizer, mae loss function, and an initial learning rate of 10^{-4}. The optimized training and loss curves for TFRCNN and VGG13 are shown in Figure 17. The convergence rate of TFRCNN was faster than that of VGG13. TFRCNN converged after 20 steps and 80 s, while VGG13 required 150 steps and 180 s.

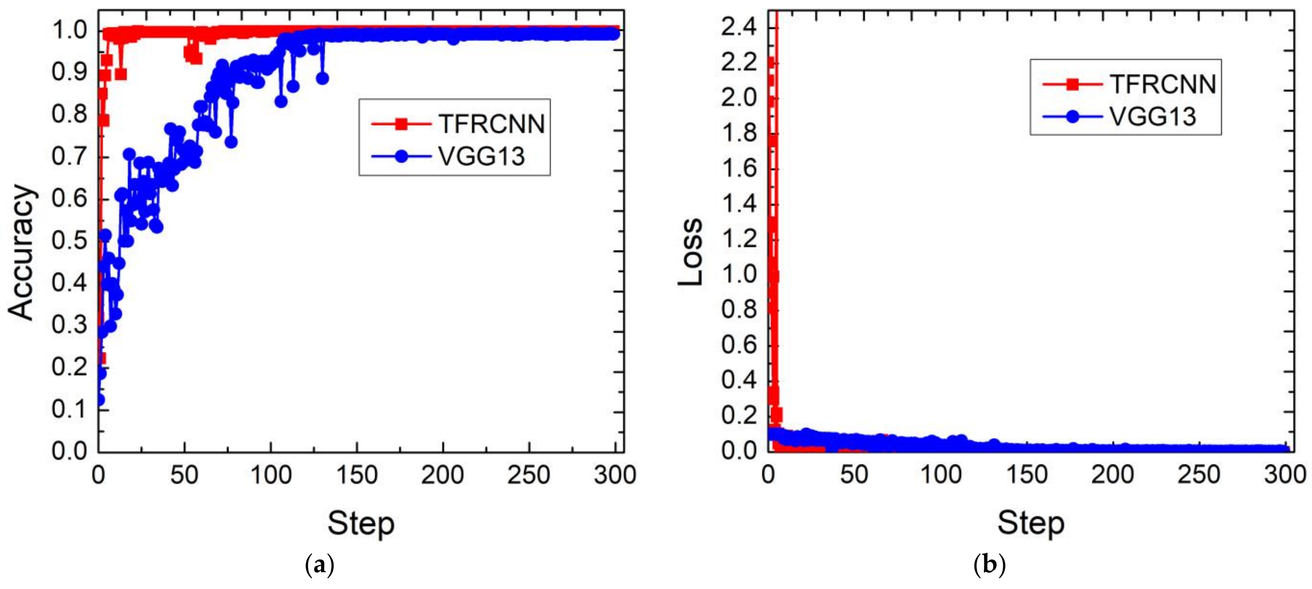


Figure 17. Optimized training accuracy and loss curves for TFRCNN and VGG13: (**a**) optimized training accuracy curves; (**b**) optimized training loss curves.

Figure 18 shows the prediction accuracy of VGG13 trained with DS4 for 10 repeated tests. The mean accuracy and standard deviation are shown in Table 7. Figure 19 shows the mean prediction accuracy of TFRCNN and VGG13 trained with DS4 for each load case.

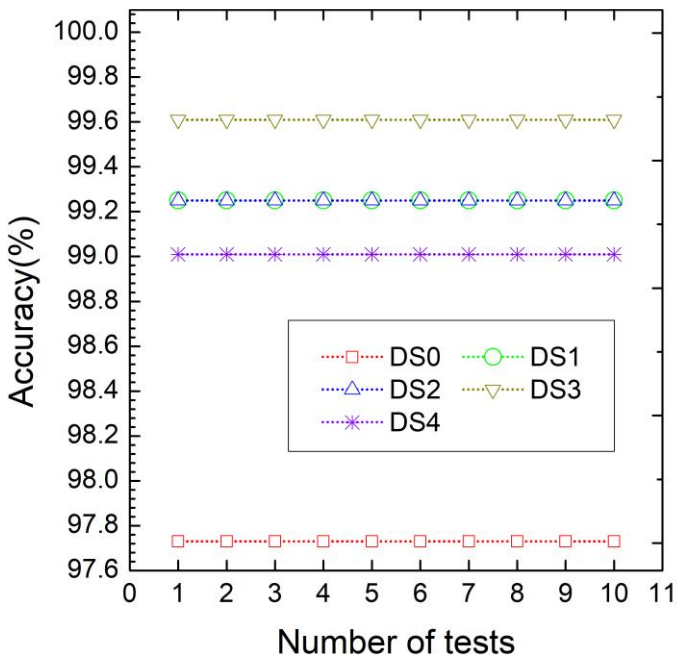
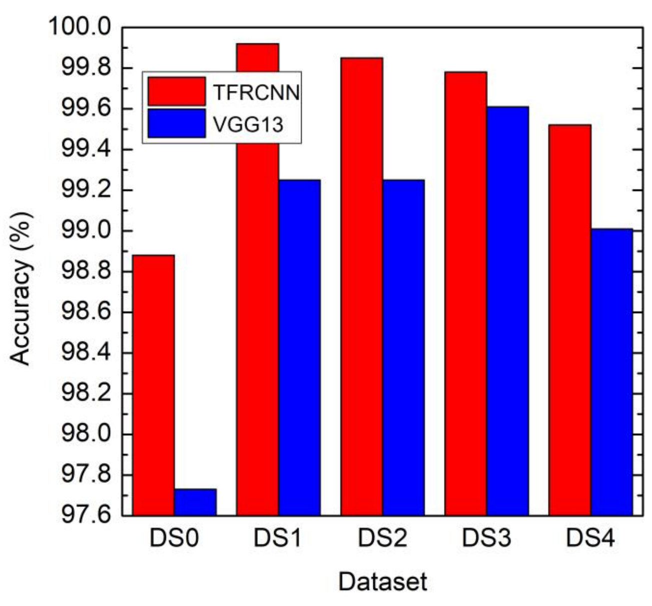
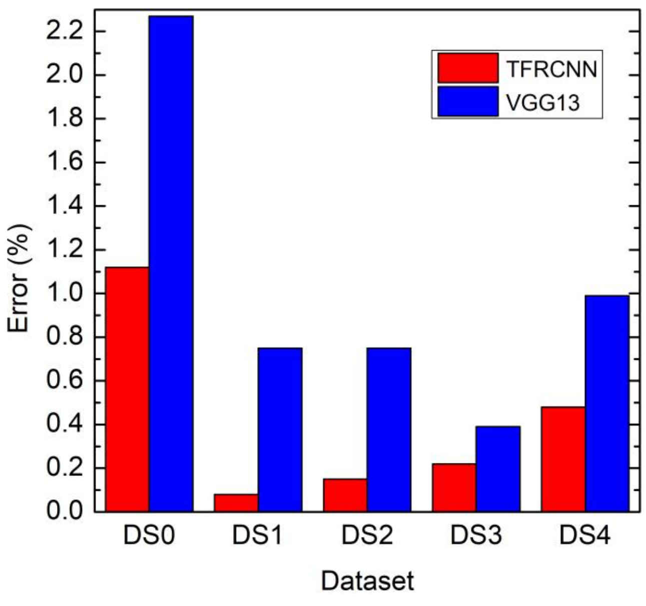


Figure 18. Accuracy of VGG13 trained with DS4 for 10 repeated tests.



(**a**)



(**b**)

Figure 19. Mean accuracy and error of TFRCNN and VGG13: (**a**) mean accuracy; (**b**) mean error.

It is obvious that the prediction capability of TFRCNN comprehensively exceeded that of VGG13, once again verifying the high accuracy and stability of TFRCNN.

3.4.3. Validation with the IMS Dataset

In order to verify the universality of the models, another experiment dataset from the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati, was added to evaluate their performance. Like the CWRU bearing database, the IMS bearing datasets are also widely used for verifying bearing fault diagnosis methods. These samples were acquired from the motor at 2000 rpm under a radial load of 26,695 N with a sampling frequency of 20 KHz. Table 8 lists the sample size, category, and label of the IMS dataset. Figure 20 shows the prediction accuracy of the models for 10 repeated tests. The mean accuracy and standard deviation are shown in Table 9. The experimental results once again demonstrate that the model had high prediction accuracy and stability for different data sources.

Table 8. IMS dataset specification.

Sample Size	Fault Location	Source	Label
200 × 512	Ball	Channel 5, Bearing 3	0
200 × 512	Inner race	Channel 7, Bearing 4	1
200 × 512	Outer race	Channel 1, Bearing 1	2
200 × 512	Normal	Channel 5, Bearing 3	3

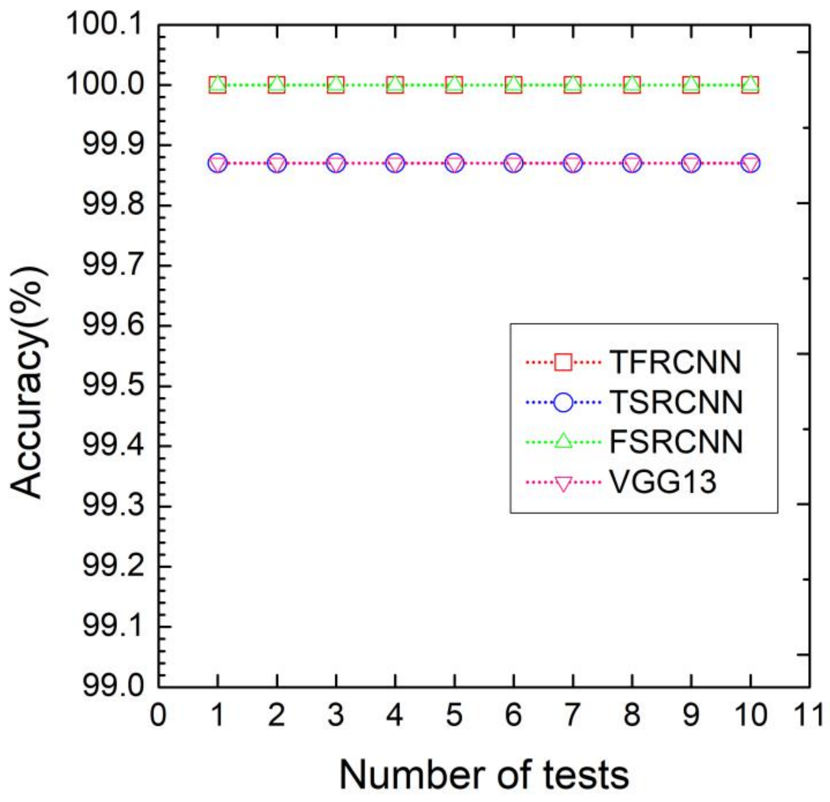


Figure 20. Accuracy of the models for 10 repeated tests.

Table 9. Mean accuracy and standard deviation.

Model	Mean (%)	Standard Deviation
TFRCNN	100	0
TSRCNN	99.87	0
FSRCNN	100	0
VGG13	99.87	0

Table 10 lists the bearing fault diagnosis methods used in this work alongside other classical and CNN methods for comparison.

Table 10. Comparison of different diagnostic methods.

Study	Method	Accuracy (%)
[29]	Zero-crossing + ANN	91.5–97.1
[30]	CWT + CTKSVM	96.23
[31]	CWT + SVD + KMCSVM	95.34
[32]	Wavelet + K * classifier	92.50
[33]	TF + MDAE + SAMB + NN	96.65
[34]	SSL + CNN	96.00
Present work	TSRCNN	98.08–99.63 *
	FSRCNN	98.50–99.67 *
	VGG13	97.73–99.61 *
	TFRCNN	98.88–99.92 *

* Prediction accuracy of models trained with mixing dataset DS4.

The prediction accuracy of TFRCNN, FSRCNN, TSRCNN, and VGG13 was significantly higher than that of the other methods [29–34], due to the ability of deep convolution and pooling to adaptively extract the bearing fault features and achieve efficient recognition through deep integration with a classifier. However, for the other methods, the feature selection required professional signal processing technology and was subjected to personal experience preference, insufficient feature extraction of shallow networks, overfitting of fully connected networks, etc.; thus, the capability of diagnostic accuracy, robustness, stability, and generalization was restricted.

Among them, TFRCNN with a double branch had the highest prediction accuracy, ranging from 98.88 to 99.92%, demonstrating good stability and generalization with a small fluctuation, while FSRCNN, TSRCNN, and VGG13 with single branches had room for further improvement in performance.

Based on the above comparison and analysis, the results of TFRCNN are clarified in detail as follows:

1. The prediction accuracy of the models under specific load (see Figure 14) indicates that TFRCNN with a double-branch structure of time and frequency domains has better performance of robustness, generalization, high prediction accuracy, and stability than FSRCNN and TSRCNN with a single structure.
2. The prediction accuracy of the models trained with training set DS4 (see Figures 15e and 19) shows that TFRCNN is further proven to have stronger robustness, wider generalization, and higher prediction accuracy than FSRCNN, TSRCNN, and VGG13 trained with the same mixing load data.
3. Compared with the classical methods (see Table 10), TFRCNN is once more verified to have significantly higher prediction accuracy, with little fluctuation and high stability.

4. Conclusions

The proposed TFRCNN aimed to take advantage of CNN in the integration of deep feature extraction for big data and fault classification, enhance anti-interference ability by extracting the time and frequency domain features of the bearing state signals, and use residual structure to prevent the degradation of the deep network, so as to improve the diagnosis ability.

The study mainly discusses the effectiveness of TFRCNN under five changeable loads and two different kinds of dataset sources. TFRCNN trained with mixing loads had better prediction accuracy and generalization ability than the other conditions. It is expected that TFRCNN can ultimately be transplanted to different factory environments through further research and improvement.

The major conclusions are drawn as follows:

1. The experimental results show that the prediction accuracy of TFRCNN reached 98.88–99.92%, which is higher than that of the other methods.

2. Compared with single-branch FSRCNN, TSRCNN, and VGG13, the double-branch structure of TFRCNN for extracting time and frequency features can enrich and improve feature expression from the raw data sources, thereby ensuring the prediction effect.
3. The residual structure of TFRCNN can resolve the contradiction between the increasing network layers and learning degradation, and the global average pooling improves the sparsity of the model as well.
4. Compared to the other models, TFRCNN has more advantages in prediction accuracy, convergence, robustness, sparsity, and generalization ability.

With the increasing demands of big data feature extraction for rotating machinery, the findings in this paper certainly promote the exploration and innovation of deep learning methods such as CNN to achieve multi-perspective deep feature extraction and highly efficient diagnosis.

Author Contributions: Conceptualization, C.W. and R.J.; methodology, C.W. and R.J.; software, C.W., X.W., C.Z. and C.H.; validation, C.W., R.J., X.W. and C.Z.; formal analysis, C.W., R.J., X.W. and C.Z.; data curation, C.W.; writing—original draft preparation, C.W.; writing—review and editing, C.W.; visualization, C.W.; supervision, R.J.; project administration, C.W. and R.J.; funding acquisition, C.W., X.W., C.Z. and C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Scientific Research Fund of Hunan Provincial Education Department, grant numbers 20A107 and 20A109; and Hunan Provincial Natural Science Foundation of China, grant numbers 2021JJ50112, 2023JJ40215 and 2021JJ30182.

Data Availability Statement: The research work in this paper uses two publicly available datasets: CWRU and IMS, which can be accessed, respectively, at: https://engineering.case.edu/bearingdatacenter/download-data-file or https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/ (accessed on 8 May 2023).

Acknowledgments: The authors would like to thank all the reviewers for giving valuable suggestions for this paper. The authors also thank the bearing data center of the Case Western Reserve University and the Center for Intelligent Maintenance Systems of Cincinnati University for using the bearing data.

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

References

1. Tan, H.; Xie, S.; Zhou, H.; Ma, W.; Yang, C.; Zhang, J. Sensible multiscale symbol dynamic entropy for fault diagnosis of bearing. *Int. J. Mech. Sci.* **2023**, *256*, 108509. [CrossRef]
2. Buchaiah, S.; Shakya, P. Bearing fault diagnosis and prognosis using data fusion based feature extraction and feature selection. *Measurement* **2022**, *188*, 110506. [CrossRef]
3. Bastami, A.R.; Vahid, S. Estimating the size of naturally generated defects in the outer ring and roller of a tapered roller bearing based on autoregressive model combined with envelope analysis and discrete wavelet transform. *Measurement* **2020**, *159*, 107767. [CrossRef]
4. Gao, S.; Li, T.; Zhang, Y.; Pei, Z. Fault diagnosis method of rolling bearings based on adaptive modified CEEMD and 1DCNN model. *ISA Trans.* **2023**, *140*, 309–330. [CrossRef] [PubMed]
5. Rikam, L.E.; Bitjoka, L.; Nketsa, A. Quaternion Fourier Transform spectral analysis of electrical currents for bearing faults detection and diagnosis. *Mech. Syst. Signal Process.* **2022**, *168*, 108656. [CrossRef]
6. Yan, R.; Shang, Z.; Xu, H.; Wen, J.; Zhao, Z.; Chen, X.; Gao, R.X. Wavelet transform for rotary machine fault diagnosis:10 years revisited. *Mech. Syst. Signal Process.* **2023**, *200*, 110545. [CrossRef]
7. Chauhan, S.; Singh, M.; Aggarwal, A.K. Bearing defect identification via evolutionary algorithm with adaptive wavelet mutation strategy. *Measurement* **2021**, *179*, 109445. [CrossRef]
8. Xu, L.; Chatterton, S.; Pennacchi, P. Rolling element bearing diagnosis based on singular value decomposition and composite squared envelope spectrum. *Mech. Syst. Signal Process.* **2021**, *148*, 107174. [CrossRef]
9. Chaleshtori, A.E.; Aghaie, A. A novel bearing fault diagnosis approach using the Gaussian mixture model and the weighted principal component analysis. *Reliab. Eng. Syst. Saf.* **2024**, *242*, 109720. [CrossRef]
10. Yin, J.; Zhuang, X.; Sui, W.; Sheng, Y. Manifold learning and Lempel-Ziv complexity-based fault severity recognition method for bearing. *Measurement* **2023**, *213*, 112714. [CrossRef]

11. Li, B.; Delpha, C.; Diallo, D.; Migan-Dubois, A. Application of Artificial Neural Networks to photovoltaic fault detection and diagnosis: A review. *Renew. Sustain. Energy Rev.* **2021**, *138*, 110512. [CrossRef]
12. Meserkhani, A.; Jafari, S.; Rahi, A. Experimental comparison of acoustic emission sensors in the detection of outer race defect of angular contact ball bearings by artificial neural network. *Measurement* **2021**, *168*, 108198. [CrossRef]
13. Li, J.; Yao, X.; Wang, X.; Yu, Q.; Zhang, Y. Multiscale local features learning based on BP neural network for rolling bearing intelligent fault diagnosis. *Measurement* **2020**, *153*, 107419. [CrossRef]
14. Wang, B.; Zhang, X.; Xing, S.; Sun, C.; Chen, X. Sparse representation theory for support vector machine kernel function selection and its application in high-speed bearing fault diagnosis. *ISA Trans.* **2021**, *118*, 207–218. [CrossRef] [PubMed]
15. Jha, R.K.; Swami, P.D. Fault diagnosis and severity analysis of rolling bearings using vibration image texture enhancement and multiclass support vector machines. *Appl. Acoust.* **2021**, *182*, 108243. [CrossRef]
16. Islam, M.M.; Prosvirin, A.E.; Kim, J.-M. Data-driven prognostic scheme for rolling-element bearings using a new health index and variants of least-square support vector machines. *Mech. Syst. Signal Process.* **2021**, *160*, 107853. [CrossRef]
17. Wang, J.; Liu, P.; Lu, S.; Zhou, M.; Chen, X. Decentralized plant-wide monitoring based on mutual information-Louvain de-composition and support vector data description diagnosis. *ISA Trans.* **2023**, *133*, 42–52. [CrossRef]
18. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [CrossRef]
19. Yu, J.; Yan, X. Data-feature-driven nonlinear process monitoring based on joint deep learning models with dual-scale. *Inf. Sci.* **2022**, *591*, 381–399. [CrossRef]
20. Chen, L.; Ma, Y.; Hu, H.; Khan, U.S. An effective fault diagnosis approach for bearing using stacked de-noising auto-encoder with structure adaptive adjustment. *Measurement* **2023**, *214*, 112774. [CrossRef]
21. Tang, J.; Wu, J.; Hu, B.; Liu, J. Towards a fault diagnosis method for rolling bearing with Bi-directional deep belief network. *Appl. Acoust.* **2022**, *192*, 108727. [CrossRef]
22. Yan, X.; She, D.; Xu, Y. Deep order-wavelet convolutional variational autoencoder for fault identification of rolling bearing under fluctuating speed conditions. *Expert Syst. Appl.* **2023**, *216*, 119479. [CrossRef]
23. An, Z.; Li, S.; Wang, J.; Jiang, X. A novel bearing intelligent fault diagnosis framework under time-varying working conditions using recurrent neural network. *ISA Trans.* **2020**, *100*, 155–170. [CrossRef] [PubMed]
24. Yan, W.; Wang, J.; Lu, S.; Zhou, M.; Peng, X. A Review of Real-Time Fault Diagnosis Methods for Industrial Smart Manufacturing. *Processes* **2023**, *11*, 369. [CrossRef]
25. Jiao, J.; Zhao, M.; Lin, J.; Liang, K. A comprehensive review on convolutional neural network in machine fault diagnosis. *Neuro-computing* **2020**, *417*, 36–63. [CrossRef]
26. Chen, Z.; Mauricio, A.; Li, W.; Gryllias, K. A deep learning method for bearing fault diagnosis based on Cyclic Spectral Coherence and Convolutional Neural Networks. *Mech. Syst. Signal Pract.* **2020**, *140*, 106683. [CrossRef]
27. Jia, L.; Chow, T.; Yuan, Y. GTFE-Net: A Gramian Time Frequency Enhancement CNN for bearing fault diagnosis. *Eng. Appl. Artif. Intel.* **2023**, *119*, 105794. [CrossRef]
28. Li, F.; Wang, L.; Wang, D.; Wu, J.; Zhao, H. An adaptive multiscale fully convolutional network for bearing fault diagnosis under noisy environments. *Measurement* **2023**, *216*, 112993. [CrossRef]
29. William, P.; Hoffman, M. Identification of bearing faults using time domain zero-crossings. *Mech. Syst. Signal Process.* **2011**, *25*, 3078–3088. [CrossRef]
30. Wu, C.; Chen, T.; Jiang, R.; Ning, L.; Jiang, Z. A novel approach to wavelet selection and tree kernel construction for diagnosis of rolling element bearing fault. *J. Intell. Manuf.* **2017**, *28*, 1847–1858. [CrossRef]
31. Wu, C.; Chen, T.; Jiang, R. Bearing fault diagnosis via kernel matrix construction based support vector machine. *J. Vibroeng.* **2017**, *19*, 3445–3461. [CrossRef]
32. Ravikumar, K.N.; Aralikatti, S.S.; Kumar, H.; Kumar, G.N.; Gangadharan, K.V. Fault diagnosis of antifriction bearing in internal combustion engine gearbox using data mining techniques. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 1121–1134. [CrossRef]
33. 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**, *34*, 1–12. [CrossRef] [PubMed]
34. Yu, K.; Lin, T.R.; Ma, H.; Li, X.; Li, X. A multi-stage semi-supervised learning approach for intelligent fault diagnosis of rolling bearing using data augmentation and metric learning. *Mech. Syst. Signal Process.* **2021**, *146*, 107043. [CrossRef]