Article

A Novel Capsule Network Based on Wide Convolution and Multi-Scale Convolution for Fault Diagnosis

Yu Wang ^{1,2}, Dejun Ning ^{1,*} and Songlin Feng ¹

¹ Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China; wangyu02@sari.ac.cn (Y.W.); fengsl@sari.ac.cn (S.F.)

² University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: ningdj@sari.ac.cn; Tel.: +86-186-1653-6063

Received: 3 April 2020; Accepted: 18 May 2020; Published: 25 May 2020

Abstract: In the prognostics health management (PHM) of rotating machinery, the accurate identification of bearing fault is critical. In recent years, various deep learning methods can well identify bearing fault based on monitoring data. However, facing changing operating conditions and noise pollution, the accuracy of these algorithms decreases significantly, which makes the algorithms difficult in practical applications. To solve this problem, a novel capsule network based on wide convolution and multi-scale convolution (WMSCCN) is proposed for fault diagnosis. The proposed WMSCCN algorithm takes one-dimensional vibration signal as an input and no additional manual processing is required. In addition, the adaptive batch normalization (AdaBN) algorithm is introduced to further enhance the adaptability of WMSCCN under noise pollution and load changes. On generalization experiments under different loads, the proposed WMSCCN and WMSCCN-AdaBN algorithms achieve average accuracy rates of 96.44% and 97.44%, respectively, which is superior to other advanced algorithms. In the noise resistance experiment, the proposed WMSCCN-AdaBN can maintain a 92.3% diagnostic accuracy in a strong noise environment with a signal to noise ratio (SNR) = −4 dB, showing a very strong anti-noise ability. When the SNR exceeds 4 dB, the accuracy reaches 100%, indicating that the proposed algorithm has a very good accuracy at low noise levels. Two experiments have effectively verified the validity and generalizability of the proposed model.

Keywords: multi-scale convolution; capsule network; fault diagnosis; adaptive batch normalization

1. Introduction

The large-scale rise of the industrial Internet has brought global industry into a new era of technological innovation and change. Industrial machinery systems are developing toward complexity, precision, and integration [1]. At the same time, the operation safety of equipment is facing greater challenges. As the core component of rotating machinery, the normal operation of rolling bearings is critical. Once a bearing fails, it will not only cause performance degradation and productivity reduction [2] but also lead to safety accidents in serious cases, resulting in huge property losses and serious casualties. Therefore, research on intelligent fault diagnosis algorithms for bearings is very important [3–5].

In recent years, as the most mainstream algorithms of artificial intelligence, deep learning algorithms have achieved excellent results in intelligent applications such as target recognition, speech recognition, and machine translation, etc. [6–8]. At the same time, deep learning has also received more and more attention in the field of fault identification. On the one hand, fault features do not

need to be designed manually but are automatically and efficiently extracted through neural networks, which reduces dependence on expert experience. On the other hand, the deep network structure can establish complex non-linear mapping relationships between monitoring signals and faults, which is critical to improve the diagnosis accuracy of complex mechanical equipment. Therefore, deep learning algorithms have gradually replaced traditional fault diagnosis algorithms and become mainstream algorithms for fault diagnosis.

In the current research, deep belief network (DBN) and its variants are widely used in fault diagnosis. DBN consists of multiple stacked restricted Boltzmann machines (RBMs) [9], which can be used for both supervised and unsupervised learning. Unsupervised layer-by-layer greedy pre-training is the first step in DBN training. Next, the parameters of the neural network are fine-tuned using the back-propagation (BP) algorithm based on class labels. Compared with the traditional neural network only relying on the BP algorithm for model training, DBN's training method can achieve a higher performance in pattern recognition [10]. Gan et al. [11] combined two layers of DBN to form a new hierarchical diagnosis network (HDN) which identified the types of faults and the degree of faults hierarchically. Shen et al. [12] proposed an improved hierarchical adaptive DBN. The input of the network is frequency domain information and it is optimized using Nesterov momentum. This method achieved a high performance in the diagnosis of bearing fault types and damage levels. The training process of stacked autoencoder (SAE) is similar to DBN, except that the basic module of SAE is autoencoder (AE). Lu et al. [13] used stacked denoising autoencoder (SDA) for fault diagnosis and pointed out that the algorithm is robust to noise pollution and changing working conditions. Based on the data collected by multiple accelerometers, Chen et al. [14] combined SAE and DBN to diagnose bearing faults. Specifically, this method first extracts time-domain and frequency-domain features based on multiple sensor data. On this basis, SAE is used for feature fusion and DBN is trained to identify bearing faults. Li et al. [15] proposed an unsupervised diagnosis model that uses normalized frequency-domain signal as an input and integrates sparse autoencoder, DBN, and binary processor.

The above SAE and DBN methods have achieved good fault diagnosis results to a certain extent, but on the network structure, they adopt a fully connected mode between different layers. The above method performs poorly when the feature has translation or scaling [16]. The convolutional neural network (CNN) is the first image processing method proposed by Yann LeCun [17], and has opened a new era of computer vision [18]. CNN is generally constructed by alternately stacking convolutional layers and pooling layers and finally adding a fully connected (FC) layer. CNN has the characteristics of local connection, weight sharing, and pooling mechanism. This makes CNN well able to cope with the features of translation and scaling. On the other hand, it reduces the network parameters to be trained and reduces the computational complexity. Based on the classic CNN network structure LeNet-5, Wen et al. [19] proposed an innovative CNN algorithm for bearing fault diagnosis. The algorithm converts vibration signals into two-dimensional grayscale images by means of signal stacking. Han et al. [20] proposed a new diagnostic framework, namely the spatiotemporal convolutional neural network (ST-CNN) framework, which includes a spatiotemporal pattern network (STPN) for spatiotemporal feature learning and CNN for condition classification. In [21–23], the authors used 1D-CNN for fault diagnosis and used raw data as the input. Among them, Zhang et al. [22] first analyzed the characteristics of the vibration signal and proposed a deep CNN method with a wide convolution kernel for the first layer. The algorithm directly inputs the original one-dimensional vibration signal into the model and achieves good results in generalization. At the same time, they also show that in a low signal to noise ratio (SNR) environment, a wide first-layer convolution kernel can more effectively extract low-frequency features and prevent noise interference.

Although CNN and its variants have achieved good results in identifying bearing faults, the existence of the pooling layer causes the valuable spatial information between the layers to be lost. The capsule network with dynamic routing mechanism proposed by Sabour et al. [24] perfectly solves this problem. In the capsule network, the capsule replaces some neurons and becomes the “basic unit” of the neural network. The low-layer capsule is transmitted to the high-layer capsule

through dynamic routing. Unlike max-pooling, the dynamic routing mechanism enables transmission between layers without discarding information about the precise location of entities in the area. Recently, some researches have applied capsule networks to the field of fault diagnosis. Wang et al. [25] first obtained the time–frequency diagram of the fault through a wavelet time–frequency analysis and then combined the capsule network with the Xception module for intelligent fault diagnosis. This algorithm has a better fault classification ability and high reliability through experimental verification. Zhu et al. [26] introduced the inception module and a regression branch for diagnosing fault damage based on the classic capsule network. In the data pre-processing phase, they converted the original vibration signal into time–frequency graphs by means of short-time Fourier transform. The algorithm was tested on different bearing datasets, and the experimental results verify the strong generalization of the algorithm.

In general, the above-mentioned deep learning algorithms have been widely used in bearing fault diagnosis and have achieved a high recognition accuracy in specific tasks. However, these algorithms have some shortcomings that need further improvement. First of all, most algorithms require the preprocessing of the original signal. Taking the DBN and SAE algorithms as examples, the input is generally time-domain features or frequency-domain features or both. In the current proposed capsule network for fault diagnosis, the input is generally a pre-processed time–frequency graph [25,26]. These preprocessing operations rely on expert experience, which hinders the further promotion of the algorithm. In addition, many algorithms have poor generalization, which is manifested in the fact that when a classifier trained by data under one working condition is used to classify data under other working conditions, the accuracy rate drops significantly. The changing conditions of the actual industrial site (such as load and speed changes) require the algorithm to further improve its generalization. Finally, few algorithms can perform well in strong noise environments, as can be clearly seen in the experimental results in Section 4.3. This makes it difficult to apply the algorithm in actual factories. In view of the above shortcomings, a novel capsule network based on wide convolution and multi-scale convolution (WMSCCN) for fault diagnosis is proposed. The algorithm has the following innovations.

- (1) A novel capsule network for fault diagnosis is proposed which takes the original signal as input and does not require any time-consuming manual feature extraction processes.
- (2) The proposed WMSCCN algorithm has a high diagnostic accuracy in different working conditions and is superior to other advanced models, such as Deep Convolutional Neural Networks with Wide First-Layer Kernels (WDCNN) [22].
- (3) By adding noise to the test set of data to simulate noise pollution in the industrial environment, the proposed model still achieves a higher accuracy when compared to other algorithms and has a better anti-noise ability.

The rest of the paper is organized into four sections. The theory of CNN and capsule network is briefly introduced in Section 2, and Section 3 describes the technical details of the proposed WMSCCN algorithm. Two comparative experiments are carried out and some results are visualized in Section 4. In the last section, this paper presents the conclusions of the study.

2. Basic Theory of CNN and Capsule Network

2.1. Convolutional Layer

The convolutional layer is composed of multiple feature maps, and each feature map is composed of multiple neurons. Each neuron of the feature map is connected to the local area of the feature map of the previous layer through the convolution kernel. The convolutional layer performs convolution operations on the input signal through different convolution kernels to extract corresponding features. It is generally believed that a convolution kernel defines a certain pattern. The convolution operation is used to calculate the degree of similarity between each position and the pattern. The more similar

the current position to the pattern, the stronger the response. The convolution operation formula is shown below:

$$\mathbf{y}_i^{l+1}(j) = \mathbf{K}_i^l \times \mathbf{x}^l(j) + b_i^l, \quad (1)$$

where $\mathbf{K}_i^l$ and b_i^l represent the weights and biases of the i -th kernel in the l -th layer, respectively, and $\mathbf{x}^l(j)$ denotes the j -th local region in the l -th layer. $\mathbf{y}_i^{l+1}(j)$ is output value of the convolutional operation.

Two major features of convolutional layers are local connection and weight sharing. Local connection means that each time the convolutional layer performs a convolution operation, it only connects and operates with some nodes in the feature map of the previous layer. At the same time, the weight sharing reduces the network parameters of the convolutional layer greatly, which further reduces the risk of overfitting. In practice, a correlation operation is often used instead of a convolution operation.

In the classic CNN structure, the convolutional layer is usually followed by the pooling layer. The pooling layer is to sample the feature map of the previous layer to reduce the size of the feature map. Common pooling functions are maximum pooling and mean pooling. By alternately stacking the convolutional layer and the pooling layer, the features in the input signal can be effectively extracted [27]. As the number of network layers deepens, the features learned by the network gradually change from simple features to complex and abstract features.

2.2. Capsule Network

Traditional CNNs mainly extract the main features to the next layer through the pooling mechanism—that is, selecting the regional maximum or regional average. However, this operation leads to the loss of valuable spatial information between layers, which further reduces the network recognition rate. To solve the above problems, Sabour et al. [24] proposed a novel network-capsule network, and this model used dynamic routing mechanism to support transmission between different capsule layers.

In capsule networks, capsules partially replace neurons in traditional neural networks to more effectively identify features. Specifically, a capsule is a vector composed of multiple neurons, and its length indicates the probability of entity existence. Each neuron in the capsule represents various properties of a specific entity, such as the pose (position, size, direction), deformation, speed, albedo, tone, texture, and other characteristics of the object [24].

Just as different layers of deep CNN learn different semantic attributes of images, capsules can also be organized into different layers. The low-layer capsules learn low-level characteristics and output to the “more similar” high-layer capsules through dynamic routing.

The flow of the dynamic routing algorithm is shown below. Among them, b_{ij} represents the intermediate weight coefficient of the capsule i in layer l and capsule j in layer $l + 1$, which is set to 0 before the first iteration; $\hat{\mathbf{u}}_{j|i}$ represents the intermediate prediction vectors of the capsule i in layer l and capsule j in layer $l + 1$, which is obtained by multiplying the output of capsule $\mathbf{u}_i$ and the corresponding weight matrix $\mathbf{W}_{ij}$. $\mathbf{W}_{ij}$ encodes very important spatial relationship between low-level features and high-level features. Its size is $\mathbf{W}_{ij} = (D_i, D_j)$, where D_c represents the dimension of capsule c . $\mathbf{W}_{ij}$ is calculated by a BP algorithm, like the parameters of a traditional neural network. During the first iteration, since b_{ij} is set to 0, all the prediction vectors $\hat{\mathbf{u}}_{j|i}$ have the same weight coefficient c_{ij} . With the process of iteration, the prediction vector $\hat{\mathbf{u}}_{j|i}$, which is more similar to the high layer capsule j , has a larger dot product result, which increases the corresponding weight coefficient c_{ij} and finally makes the low-layer capsules i tend to output to more similar high-layer capsules j . The process of dynamic routing is similar to the idea of automatic clustering. Through dynamic routing, the automatic aggregation of low-level features is completed.

Procedure 1 Dynamic routing algorithm.

```

1: procedure ROUTING( $\hat{\mathbf{u}}_{j|i}$ ,  $r$ ,  $l$ )
2:   for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$ :  $b_{ij} \leftarrow 0$ .
3:   for  $r$  iterations do
4:     for all capsule  $i$  in layer  $l$ :  $\mathbf{c}_i \leftarrow \text{softmax}(\mathbf{b}_i)$ 
5:     for all capsule  $j$  in layer  $(l + 1)$ :  $\mathbf{v}_j \leftarrow \text{squash}(\sum_i c_{ij} \hat{\mathbf{u}}_{j|i})$ 
6:     for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$ :  $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j$ 
   return  $\mathbf{v}_j$ 

```

3. Proposed WMSCCN for Fault Diagnosis

In order to meet the challenges of variable working conditions and noise pollution and further improve the generalization and anti-noise performance of the algorithm, a novel capsule network based on wide convolution and multi-scale convolution (WMSCCN) for fault diagnosis is proposed. Figure 1 shows the network structure of the proposed WMSCCN algorithm, which includes wide convolutional layers, a multi-scale convolutional layer, a primary capsule layer, and a digit capsule layer. In addition, a 1D vibration signal with the size of 2048 is the input of the proposed WMSCCN algorithm, without any manual feature extraction process. In the proposed WMSCCN algorithm, the wide convolutional layers use the larger size convolution kernels, and the multi-scale convolutional layer uses several convolution kernels of different sizes to learn the features of different time scales. In the multi-scale convolutional layer, in order to ensure that the feature maps of convolution kernels with different sizes have the same size, the zero-padding technique is used. In addition, Batch Normalization (BN) technology is used to speed up the training process, and dropout technology is used to reduce the risk of overfitting. On this basis, the adaptive batch normalization (AdaBN) algorithm is further introduced to enhance the adaptability of WMSCCN under noise and load changes. The loss function is used to determine the error between the predicted value of the algorithm and the true value. In the proposed WMSCCN algorithm, Margin Loss is selected as the loss function.

Figure 1. Framework of the novel capsule network based on wide convolution and multi-scale convolution (WMSCCN). The $a @ b$ indicates that the number of current feature maps is a and the size is b . BN stands for Batch Normalization technology, and Dim denotes the dimension of the capsule. Conv1 and Conv2 denote the first and second convolutional layers, respectively.

3.1. Wide Convolution and Multi-Scale Convolution

As the first layer of the proposed WMSCCN algorithm, the wide convolutional layers can be called the noise reduction layer in function. The core of the layers is to suppress high-frequency noise through larger-sized convolution kernels, thereby achieving the goal of improving the algorithm's noise resistance. Generally speaking, wide convolution kernels have a larger reception field than

narrow convolution kernels and can effectively capture low-frequency features. The wide convolution kernel can act as a low-pass filter, so it can better suppress high-frequency noise, which has been confirmed in [22]. In the proposed WMSCNN algorithm, the wide convolutional layers contain two convolutional layers, and the sizes of the convolution kernels are 64 and 16, respectively.

After the wide convolutional layer, the multi-scale convolutional layer is accessed. The precision of the fault feature extracted by convolutional layer depends on the size of convolution kernel. Small convolution kernels can extract fine features, while large convolution kernels can extract coarse-grained features. If a single layer only uses convolution kernels of the same scale, it is easy to ignore features of other precisions, resulting in incomplete information on the extracted features [28]. In order to enhance the fault feature extraction ability under different working conditions and strengthen the generalization of the algorithm, this paper introduces the idea of multi-scale convolution. The multi-scale convolutional layer consists of eight convolution kernels of different scales. The size of the convolution kernels is i ($i = 1, 2, 3, \dots, 8$). The stride denotes how much we move the convolution kernel at each step, and it is set to 1. The filter denotes the number of convolution kernels, and it is set to 16.

Zero padding is used in the multi-scale convolutional layers to keep the output features mapped on the same scale. It allows the convolution kernel to slide from start to end in the input features. Therefore, it can extract boundary features more efficiently. For specific positions filled with zero, PL represents the number of zero padding on the left and PR represents the number of zero padding on the right, as shown in the following formula. It is worth noting that the following formula is only applicable when the stride is 1.

$$PL = \text{ceil}\left(\frac{k-1}{4}\right) + \text{floor}\left(\frac{k-1}{4}\right), \quad (2)$$

$$PR = k - 1 - PL, \quad (3)$$

where k indicates the convolution kernel size. Function $y = \text{ceil}(x)$ means that for the floating-point number x in the positive infinite direction takes the nearest integer. For example, $\text{ceil}(6.5) = 7$. Function $y = \text{floor}(x)$ means that floating-point number x in the negative infinity direction takes the nearest integer. For example, $\text{floor}(6.5) = 6$. Following the above formula, if the total number of zeros to be filled is even, then $PL = PR$. If the total number of zeros to be filled is odd, then PL is always odd and PR is always even and the difference between them is 1.

3.2. Capsule Network Structure

After the multi-scale convolutional layer, we access the capsule network, which includes the primary capsule layer and the digit capsule layer. Among them, the primary capsule layer converts the extracted fault features into capsule form and transmits them to the digit capsule through dynamic routing. Each digit capsule represents a specific fault type and its length represents the probability of the specific fault type.

Specifically, first of all, two convolutional layers with a size of 3×1 are used to transform fault features into primary capsules with dimensions of 8. Further, the primary capsule i is transmitted to the digit capsule j through the dynamic routing mechanism, and this process retains all the spatial information of the primary capsule. In the model of this paper, the dimension of the digit capsule is 10. The number of iterations of dynamic routing is consistent with the original literature [24], which is 3.

The activation function in traditional neural networks is usually used to non-linearly activate the output of the network layer and only works on scalars. In the capsule network, the squashing function is a special activation function to normalize vectors. This activation function shrinks short vectors to almost zero length and long vectors to lengths slightly less than 1 and keeps its orientation unchanged.

It is worth noting that the length of the vector mentioned here refers to its L2 norm. The formula is as follows.

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}, \quad (4)$$

where $\mathbf{v}_j$ is the vector output of capsule j and $\mathbf{s}_j$ is its total input.

3.3. AdaBN Algorithm

AdaBN [29] is a BN-based domain adaptive algorithm. The algorithm is based on the following assumptions: the information related to the sample category label is determined by the weight of each layer and the information related to the sample domain label is represented by the BN layer statistics. AdaBN transforms the traditional BN to make the source and target domain statistics independent at the BN layer, and the remaining network parameters are still shared between the source domain and target domain.

The AdaBN algorithm was further introduced to enhance the adaptability of WMSCCN in the case of noise and load changes. The WMSCCN algorithm based on AdaBN first trains the WMSCCN model with training samples until the training is completed. If the field distributions of the training samples and the test samples are inconsistent, some parameters of the model need to be adjusted. Specifically, the mean and variance of all the BN layers are replaced with the mean and variance of the test set, and other network parameters remain unchanged. The fault diagnosis of the test set is carried out on the improved model. The WMSCCN algorithm based on AdaBN is shown as Algorithm 1, where γ_i and β_i are the scaling and translation parameters that have been trained for the neuron i on the BN layer in the WMSCCN model.

Algorithm 1 WMSCCN algorithm based on AdaBN

```

for neuron  $i$  of BN layer in WMSCCN do
    Concatenate neuron response on target domain  $k$ :  $\mathbf{x}_i = [\dots, x_i(m), \dots]$ 
    Compute the mean of the target domain  $k$ :  $\mu_i^k = \mathbf{E}(\mathbf{x}_i^k)$ ,
    Compute the standard deviation of the target domain  $k$ :  $\sigma_i^k = \sqrt{\mathbf{Var}(\mathbf{x}_i^k)}$ 
end for
for neuron  $i$  of BN layer in WMSCCN, testing sample  $m$  in target domain  $k$  do
    Compute BN output  $y_i(m) = \gamma_i \frac{x_i(m) - \mu_i^k}{\sigma_i^k} + \beta_i$ 
end for

```

3.4. Model Parameters of WMSCCN

Margin Loss is the loss function in the proposed WMSCCN algorithm. On the one hand, the process of reducing loss through training makes the digit capsules c , representing the current fault type, tend to become longer. On the other hand, Margin Loss supports the output of multi-classification results, which is of great significance for the future popularization of the model on composite faults. The corresponding formula is shown below.

$$L_c = T_c \max(0, m^+ - \|\mathbf{v}_c\|)^2 + \lambda(1 - T_c) \max(0, \|\mathbf{v}_c\| - m^-)^2, \quad (5)$$

where L_c represents the Margin Loss of digit capsule c and $\mathbf{v}_c$ represents the output of digit capsule c . The λ down-weighting of the loss for absent digit classes stops the initial learning from shrinking the lengths of the activity vectors of all the digit capsules. In the experiment, λ is set to 0.5. The T_c indicates whether the fault c exists. When the fault c exists, $T_c = 1$ and the absence c is 0. The m^+ is 0.9, which can punish false positives. When the fault c is present but the prediction result is non-existent, it will result in a large loss value. The m^- is 0.1, which can punish false negatives. When the fault c does not exist but the prediction result is present, it will also lead to large loss values.

The model structure parameters of WMSCCN are shown in Table 1. It is worth noting that the multi-scale convolutional layer uses the zero-padding technique introduced in Section 3.1 to ensure the consistent size of the feature maps. The Adam optimization algorithm [30] is used in the training process. It uses the first moment estimation and second moment estimation of the gradient to dynamically adjust the learning rate of each parameter. The Adam algorithm has the advantages of a high computational efficiency and lower memory requirements. Therefore, it is suitable for the training of neural networks with a large number of parameters. The training parameters of the proposed WMSCCN algorithm are set as follows: the batch size is 100 and the learning rate is 0.001.

Table 1. Model structure parameters.

No.	Layer Name	Kernel Size/Strides/Filters	Parameters	Output Shape	Padding
1	Conv1	64/16/32	2080	(125,32)	No
2	Conv2	16/3/64	32,832	(37,64)	No
3	Multi-scale convolution	$i (i = 1, 2, 3, \dots, 8)/1/16$	36,992	(37,128)	Yes
4	Primary capsule	3/(1,2)/32	15,424	(68,8)	No
5	Digit capsule	-	54,400	(10,10)	-

In addition, dropout is usually used as a trick for training deep networks and is also used in the proposed WMSCCN algorithm to reduce the risk of overfitting. Dropout technology refers to that in the forward propagation of each training batch; some neurons are ignored with probability p —that is, some node values of hidden layers are 0. After the training, it is equivalent to obtain an integrated model composed of multiple neural networks with different network structures, which can effectively reduce the risk of overfitting. In the proposed WMSCCN algorithm, dropout is used between the multi-scale convolutional layer and the primary capsule layer with a drop rate of 0.3—that is, 30% of the nodes in the multi-scale convolutional layer have a value of 0.

4. Experimental Analysis

In order to promote the practical industrial application, the proposed WMSCCN algorithm is tested under different working conditions and noise environments. The results of two comparative experiments well verify the generalization and noise immunity of the WMSCCN algorithm. The model was developed in Python 3.5 through the deep learning library keras 2.2.4, and experiments were performed using the CentOS 7.6 operating system and hardware with Intel (R) Xeon (R) Gold 5120 CPU and NVIDIA GeForce GTX 1080 Ti GPU.

4.1. Dataset Introduction

The Case Western Reserve University (CWRU) dataset [31] is the data source of this paper. The test bench is shown in Figure 2, which includes a motor, torque transducer/encoder, dynamometer, etc. Electrical discharge machining (EDM) is used to implant single point faults of different diameters into different positions in the Svenska Kullager-Fabriken (SKF) bearing—that is, the inner raceway, balls, and outer raceway. There are three kinds of fault sizes; the smallest is 0.007 inches and the largest is 0.014 inches.

Figure 2. Fault bearing test bench.

The above faulty bearings are installed into the motor shown in Figure 2, and the accelerometer is used to collect vibration data. Speed and horsepower data are collected using the torque transducer/encoder. This paper uses the acceleration data of the driving end as an input. The dataset contains three different fault locations and three different fault sizes, corresponding to nine damage states. In addition to the health state, the final number of digit capsules in our fault diagnosis model is set to 10.

The accelerometer collects vibration signals for 10 s and the sampling frequency is 12 kHz, so each fault sample contains 120,000 points. In addition, the experiment also set at three different working conditions—that is, the motor load is 1hp, 2hp, and 3hp, respectively. In the experiment, 2048 data points were used for fault diagnosis at a time. To facilitate the training of the network, each signal is standardized and the formula is shown below.

$$x^* = \frac{x - \bar{x}}{\sigma}, \quad (6)$$

where $\bar{x}$ and σ represent the mean and standard deviation of the original signal, respectively.

Further, data augmentation technology is applied to augment the original data during the preprocessing stage. As shown in Figure 3, the original fault information is divided into two parts; the first half contains 60,000 points as the training set and the second half contains about 60,000 points as the test set. Data enhancement is performed on the training set—that is, overlap sampling is performed with an offset of 100. It is worth noting that the test samples do not overlap. After processing the dataset, the training set has 17,400 samples and the test set has 870 samples.

Figure 3. Data enhancement diagram.

4.2. Case Study 1: Generalization Experiment under Different Working Conditions

The generalization of the proposed WMSCCN algorithm under different loads is tested in this experiment. According to different operating conditions, such as rotational speed, loads, etc., the experimental data is divided into three different datasets, as shown in Table 2. The specific

experimental steps are as follows: first, use one of the three datasets A, B, and C to train the WMSCCN model; then, test the trained model on a non-training dataset. For example, if you use dataset A to train the WMSCCN model, you can choose the B or C dataset for the model testing.

Table 2. Dataset details under different operating conditions.

Dataset Name	Rotational Speed (rpm)	Loads (hp)	Damage Size (in.)
A	1772	1	0.007, 0.014, 0.021
B	1750	2	0.007, 0.014, 0.021
C	1730	3	0.007, 0.014, 0.021

The proposed WMSCCN algorithm takes the vibration data as an input and requires no other preprocessing. It is an end-to-end intelligent fault diagnosis algorithm. Two classic algorithms and five advanced algorithms are developed for comparative experiments: artificial neural network (ANN), support vector machine (SVM) [32], WDCNN [22], CNN based on vibration images (VI-CNN) [33], New CNN based on LeNet-5 [19], multiscale CNN (MSCNN) [34] and adaptive weighted multiscale CNN (AWMSCNN) [35]. The architecture of the last five models implemented in this paper is exactly the same as the references. The input of ANN and SVM is the amplitude at different frequencies obtained by fast Fourier transform (FFT) transformation. Among them, the radial basis function (RBF) is selected as the kernel function of the SVM and the network structure of ANN is 1024-500-200-10. Both VI-CNN and the New CNN based on LeNet-5 convert vibration signals into image data as an input. For specific conversion methods, refer to literature [33] and literature [19], respectively. The input of WDCNN, MSCNN, and AWMSCNN is consistent with the input of the proposed WMSCCN algorithm, and both are original signals. Therefore, the types of input data of the comparison algorithm include frequency domain data, image data, and original data, which are more comprehensive. In this paper, the accuracy rate is selected as the evaluation index. Figure 4 shows the diagnostic accuracy under different settings, where A→B indicates that the WMSCCN model is trained on dataset A and tested on dataset B.

Figure 4. Comparison of diagnostic performance under different loads.

From the experimental results, SVM + FFT has the worst fault diagnosis performance under different loads, with an average accuracy of only 63.36%. New CNN based on LeNet-5 (referred to as New CNN) and VI-CNN, which take image as an input type, cannot effectively identify fault features under different loads because they directly stack the original data into images and input them into the variation in CNN. As a result, the diagnostic accuracy decreased, with average accuracy rates of 80.86% and 84.99%, respectively. Surprisingly, the ANN + FFT algorithm achieved a relatively good performance. Compared with the above four algorithms, the average accuracy of AWMSCNN, MSCNN, WDCNN, and the proposed WMSCCN algorithm with the original signal as input exceeds 90% and the generalization is better. The improvement in performance may be because the original time series data is more effective for fault feature extraction under variable operating conditions than frequency data and image data. In the case of the same input, the proposed WMSCCN algorithm shows a strong generalization. Although under the conditions of A→B and B→C, the accuracy of the proposed WMSCCN algorithm is 2.93% and 0.18% lower than WDCNN, respectively. However, under other conditions, the proposed WMSCCN algorithm has the highest accuracy. In terms of average accuracy, the proposed WMSCCN algorithm is the highest at 96.44%, exceeding 92.76% of WDCNN. Such comparison results fully demonstrate that the proposed WMSCCN algorithm can effectively identify fault characteristics under different loads, and the generalization performance is good. WMSCCN-AdaBN refers to the application of the AdaBN algorithm on the basis of the WMSCCN algorithm to further strengthen the model's domain adaptation and generalization capabilities. Experimental results indicate that after applying the AdaBN algorithm, the average accuracy of the model has improved by 1%.

4.3. Case Study 2: Noise Resistance Experiment under Different Levels of Noise

The noise resistance performance of the proposed WMSCCN algorithm is tested in this section. In the choice of noise, many researchers choose Gaussian white noise to simulate noise pollution in industrial environments [16,22,35] and further test the anti-noise performance of the algorithm. Therefore, in this section Gaussian white noise is used for the noise immunity experiments. Specifically, different levels of Gaussian white noise are injected into the test set to simulate the noise pollution in industrial environments. It is worth noting that the training set in this experiment still uses the original signal.

SNR is a standard to measure the noise level in a signal. Its formula is as follows. Larger SNR values indicate a better signal quality and lower noise levels in the signal; otherwise, it is the opposite. When the SNR value is 0, the signal and noise power are equal.

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right), \quad (7)$$

where P_{signal} and P_{noise} indicate the power of the test sample and noise, respectively. Add different degrees of additive Gaussian white noise to the test sample according to the following formula.

$$x = x_s + x_n \times P_{noise}, \quad (8)$$

where x_s is the original signal in the test set and its power is P_{signal} . The x_n represents a random array with the same size of x_s , which is sampled from the standard Gaussian distribution. In order to make Equation (8) consistent with Equation (7), the power of x_n is 1.

The dataset used in the experiments in this section is the dataset described in Section 4.1—that is, the training set has 17,400 samples and the test set has 870 samples. Gaussian white noise of different degrees is added to the test set, and its SNR range is from −4 dB to 10 dB.

Figure 5 shows the results of the noise immunity experiment. Experimental results indicate that as the SNR increases, the recognition accuracy of all the algorithms is significantly improved. When the SNR = −4 dB, the original signal is submerged in the noise information due to the influence of strong

noise, which significantly reduces the fault recognition accuracy of all the algorithms. Among them, the ANN + FFT algorithm has the worst recognition performance—only 33.56%. The SVM + FFT algorithm has a better recognition performance of 67.82%. WMSCCN-AdaBN uses domain adaptive technology, so it can better identify the fault characteristics in the strong noise environment; it finally reaches an accuracy of 92.30%, far exceeding other algorithms. When the SNR = −2 dB, as the noise weakens, the accuracy of the algorithm improves significantly. The proposed WMSCCN algorithm has the biggest improvement; compared with the −4 dB case, it improves the accuracy by 24.48%. When the SNR = 0 dB, the power of the original signal is the same as the noise. At this time, the algorithms with a diagnostic accuracy exceeding 90% include SVM + FFT, WMSCCN, and WMSCCN-AdaBN; the diagnostic accuracy of the WMSCCN-AdaBN algorithm is still the highest, at 98.39%. When the SNR = 10 dB, the power of the noise is much smaller than the signal power, and the diagnostic accuracy of all the algorithms exceeds 97%. As can be seen from the figure, WMSCCN-AdaBN can maintain a 92.3% diagnostic accuracy in a strong noise environment with a SNR = −4 dB, showing a very strong anti-noise ability. When the SNR exceeds 4 dB, the accuracy reaches 100%, indicating that the proposed algorithm has a very good accuracy at low noise levels.

Figure 5. Comparison of diagnostic performance under different degrees of noise.

4.4. Analysis

In order to detect the internal efforts made by the capsule network in the generalization of the WMSCCN algorithm, two similar algorithms were established and compared with WMSCCN. The first half of the two algorithms is completely consistent with WMSCCN, including wide convolutional layers and a multi-scale convolutional layer. Based on the above layers, an algorithm connects three small convolutional layers (the size of the convolution kernel is 3) and two fully connected layers with 50 and 10 nodes, called WMSCNN-I. Based on the wide convolutional layers and the multi-scale convolutional layer, another algorithm accesses three fully connected layers with 500, 50, and 10 nodes which are

called WMSCNN-II. This section takes the influence of the capsule network on the generalization of WMSCNN as the research object. Therefore, the experimental data is consistent with Section 4.2.

The experimental results are shown in Table 3. It can be seen that the WMSCNN-I and WMSCNN-II algorithms have achieved a relatively good generalization on the basis of wide convolutional layers and the multi-scale convolutional layer, and their average accuracy is close to or exceeds 90%. The structure of the capsule network effectively improves the generalization of the algorithm, as can be clearly seen from Table 3. Especially in the case of $A \rightarrow C$ and $C \rightarrow B$, the addition of the capsule network improves the accuracy of the algorithm by about 13% and 10%, respectively. Compared with the two similar algorithms WMSCNN-I and WMSCNN-II, the proposed WMSCCN algorithm improves the average accuracy by 5.58% and 8.48%, respectively. This result fully shows that the addition of the capsule network effectively improves the generalization of the algorithm.

Table 3. Experimental results of the generalization of the capsule network.

Algorithms	A→B	A→C	B→A	B→C	C→A	C→B	Average
WMSCNN-I	94.48%	83.79%	95.86%	93.10%	88.10%	89.83%	90.86%
WMSCNN-II	94.31%	83.97%	91.72%	88.45%	82.93%	86.38%	87.96%
WMSCCN	95.17%	97.24%	98.79%	96.55%	93.10%	97.76%	96.44%

4.5. Visualization of Results

This section visualizes the experimental results of $B \rightarrow A$ in the generalization experiments. Specifically, for all samples of the dataset A, the feature expressions of all the convolutional layers and capsule layers are reduced to a two-dimensional distribution by means of t-distributed stochastic neighbor embedding (t-SNE), as shown in Figure 6.

Figure 6. Visualization of all the samples of dataset A in each layer of WMSCCN after t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction. (a) Visualization of all the samples of dataset A in conv1 layer after t-SNE. (b) Visualization of all the samples of dataset A in conv2 layer after t-SNE. (c) Visualization of all the samples of dataset A in the multi-scale convolutional layer after t-SNE. (d) Visualization of all the samples of dataset A in the primary capsule layer after t-SNE. (e) Visualization of all the samples of dataset A in the digit capsule layer after t-SNE.

Experimental results show that as the network structure deepens, test samples of the same fault gradually gather and eventually become independent clusters. In the expression of the multi-scale convolutional layer, the expressions of OuterRace-0.021 fault, InnerRace-0.014 fault, and InnerRace-0.007 fault are linearly inseparable at this layer; after the primary capsule layer, the characteristic expressions of the three types of fault are linearly separable. This shows that the model's non-linear expression ability is constantly increasing. In summary, the WMSCCN model maps inseparable features to a non-linear separable space, effectively identifying different fault types.

5. Conclusions

In order to meet the challenges of variable working conditions and noise pollution and further improve the generalization and anti-noise performance of the algorithm, a novel capsule network model based on wide convolution and multi-scale convolution (WMSCCN) for bearing fault diagnosis is proposed. The proposed WMSCCN algorithm takes a one-dimensional vibration signal as input and no additional manual processing is required. In addition, the AdaBN algorithm further improves the robustness of the model, and in particular obtains a higher fault recognition rate under noisy conditions. In generalization experiments under different loads, the proposed WMSCCN and WMSCCN-AdaBN algorithms achieve average accuracy rates of 96.44% and 97.44%, respectively, which is superior to other advanced algorithms. In the noise resistance experiment, WMSCCN-AdaBN could maintain a 92.3% diagnostic accuracy in a strong noise environment with an SNR = −4 dB, showing a very strong anti-noise ability. When the SNR exceeds 4 dB, the accuracy reaches 100%, indicating that the proposed algorithm has a very good accuracy at low noise levels. Two experiments have effectively verified the effectiveness and generalizability of the proposed WMSCCN algorithm.

Author Contributions: Data collection, data analysis, python programming, manuscript writing, Y.W.; manuscript review, comparative experiment design, funding acquisition, D.N. and S.F. This version of the manuscript has been read, agreed on, and substantially contributed to by all the authors. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the green manufacturing system integration project of the Ministry of Industry and Information Technology of the People's Republic of China—GE Green Supply Chain Innovation Project.

Acknowledgments: The authors would like to thank the Shanghai Economic and Information Technology Commission Industrial Internet Innovation Development Special Fund Project-the security and operation and maintenance service platform for industrial key basic information facilities and the Shanghai Industrial Field (Steel Manufacturing) Big Data Joint Innovation Laboratory for their support.

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

References

1. Chen, Z.; Gryllias, K.; Li, W. Mechanical fault diagnosis using Convolutional Neural Networks and Extreme Learning Machine. *Mech. Syst. Signal Process.* **2019**, *133*, 1–21. [\[CrossRef\]](#)
2. Rai, A.; Upadhyay, S.H. A review on signal processing techniques utilized in the fault diagnosis of rolling element bearings. *Tribol. Int.* **2016**, *96*, 289–306. [\[CrossRef\]](#)
3. Shao, H.; Jiang, H.; Lin, Y.; Li, X. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. *Mech. Syst. Signal Process.* **2018**, *102*, 278–297. [\[CrossRef\]](#)
4. Wei, Z.; Wang, Y.; He, S.; Bao, J. A novel intelligent method for bearing fault diagnosis based on affinity propagation clustering and adaptive feature selection. *Knowl.-Based Syst.* **2017**, *116*, 1–12. [\[CrossRef\]](#)
5. Li, X.; Zhang, W.; Ding, Q. Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. *Signal Process.* **2019**, *161*, 136–154. [\[CrossRef\]](#)
6. Yang, J.; Li, X.; Jiang, Y.; Qiu, G.; Buckdahn, S. Target recognition system of dynamic scene based on artificial intelligence vision. *J. Intell. Fuzzy Syst.* **2018**, *35*, 4373–4383. [\[CrossRef\]](#)
7. Khalil, R.A.; Jones, E.; Babar, M.I.; Jan, T.; Zafar, M.H.; Alhussain, T. Speech emotion recognition using deep learning techniques: A review. *IEEE ACCESS* **2019**, *7*, 117327–117345. [\[CrossRef\]](#)

8. Xia, Y. Research on statistical machine translation model based on deep neural network. *Computing* **2020**, *102*, 643–661. [\[CrossRef\]](#)
9. Mohamed, A.; Dahl, G.E.; Hinton, G. Acoustic modeling using deep belief networks. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 14–22. [\[CrossRef\]](#)
10. Sarikaya, R.; Hinton, G.E.; Deoras, A. Application of deep belief networks for natural language understanding. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 778–784. [\[CrossRef\]](#)
11. Gan, M.; Wang, C.; Zhu, C. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mech. Syst. Signal Process.* **2016**, *72–73*, 92–104. [\[CrossRef\]](#)
12. Shen, C.; Xie, J.; Wang, D.; Jiang, X.; Shi, J.; Zhu, Z. Improved hierarchical adaptive deep belief network for bearing fault diagnosis. *Appl. Sci.* **2019**, *9*, 3374. [\[CrossRef\]](#)
13. Lu, C.; Wang, Z.Y.; Qin, W.L.; Ma, J. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Process.* **2017**, *130*, 377–388. [\[CrossRef\]](#)
14. Chen, Z.; Li, W. Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 1693–1702. [\[CrossRef\]](#)
15. Li, J.; Li, X.; He, D.; Qu, Y. Unsupervised rotating machinery fault diagnosis method based on integrated SAE-DBN and a binary processor. *J. Intell. Manuf.* **2020**. [\[CrossRef\]](#)
16. Chen, T.; Wang, Z.; Yang, X.; Jiang, K. A deep capsule neural network with stochastic delta rule for bearing fault diagnosis on raw vibration signals. *Measurement* **2019**, *148*, 106857. [\[CrossRef\]](#)
17. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* **1990**, *2*, 396–404.
18. Yang, X.; Ye, Y.; Li, X.; Lau, R.Y.K.; Zhang, X.; Huang, X. Hyperspectral image classification with deep learning models. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5408–5423. [\[CrossRef\]](#)
19. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5990–5998. [\[CrossRef\]](#)
20. Han, T.; Liu, C.; Wu, L.; Sarkar, S. An adaptive spatiotemporal feature learning approach for fault diagnosis in complex systems. *Mech. Syst. Signal Process.* **2018**, *117*, 170–187. [\[CrossRef\]](#)
21. Li, Y.; Zou, L.; Jiang, L.; Zhou, X. Fault diagnosis of rotating machinery based on combination of deep belief network and one-dimensional convolutional neural network. *IEEE Access* **2019**, *7*, 165710–165723. [\[CrossRef\]](#)
22. Zhang, W.; Peng, G.; Li, C.; Chen, Y.; Zhang, Z. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors* **2017**, *17*, 425. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Ma, Y.; Jia, X.; Bai, H.; Liu, G.; Wang, G.; Guo, C.; Wang, S. A new fault diagnosis method based on convolutional neural network and compressive sensing. *J. Mech. Sci. Technol.* **2019**, *33*, 5177–5188. [\[CrossRef\]](#)
24. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 3859–3869.
25. Wang, Z.; Zheng, L.; Du, W.; Cai, W.; Zhou, J.; Wang, J.; Han, X.; He, G. A novel method for intelligent fault diagnosis of bearing based on capsule neural network. *Complexity* **2019**, *2019*, 1–17. [\[CrossRef\]](#)
26. Zhu, Z.; Peng, G.; Chen, Y.; Gao, H. A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis. *Neurocomputing* **2018**, *323*, 62–75. [\[CrossRef\]](#)
27. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [\[CrossRef\]](#)
28. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9. [\[CrossRef\]](#)
29. Li, Y.; Wang, N.; Shi, J.; Hou, X.; Liu, J. Adaptive Batch Normalization for practical domain adaptation. *Pattern Recognit.* **2018**, *80*, 109–117. [\[CrossRef\]](#)
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Case Western Reserve University Bearing Data Center Website. Available online: <https://csegroups.case.edu/bearingdatacenter/home> (accessed on 10 January 2019).
32. Yadavar Nikraves, S.M.; Rezaie, H.; Kilpatrik, M.; Taheri, H. Intelligent fault diagnosis of bearings based on energy levels in frequency bands using wavelet and support vector machines (SVM). *J. Manuf. Mater. Process.* **2019**, *3*, 11. [\[CrossRef\]](#)

33. Hoang, D.T.; Kang, H.J. Rolling element bearing fault diagnosis using convolutional neural network and vibration image. *Cogn. Syst. Res.* **2018**, *53*, 42–50. [[CrossRef](#)]
34. Jiang, G.; He, H.; Yan, J.; Xie, P. Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox. *IEEE Trans. Ind. Electron.* **2019**, *66*, 3196–3207. [[CrossRef](#)]
35. Qiao, H.; Wang, T.; Wang, P.; Zhang, L.; Xu, M. An adaptive weighted multiscale convolutional neural network for rotating machinery fault diagnosis under variable operating conditions. *IEEE Access* **2019**, *7*, 118954–118964. [[CrossRef](#)]

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