symmetry

MDPI

Article

Intelligent Fault Diagnosis Method Based on Neural Network Compression for Rolling Bearings

Xinren Wang [1], Dongming Hu [1], Xueqi Fan [1], Huiyi Liu [2] and Chenbin Yang [2,*]

[1] Jiangsu Special Equipment Safety Supervision and Inspection Institute, Wuxi 214174, China
[2] College of Computer and Soft, Hohai University, Nanjing 211100, China
* Correspondence: yangchenbin@hhu.edu.cn

Abstract: Rolling bearings are often exposed to high speeds and pressures, leading to the symmetry in their rotating structure being disrupted, which can lead to serious failures. Intelligent rolling bearing fault diagnosis is a critical part of ensuring operation of machinery, and it has been facilitated by the growing popularity of convolutional neural networks (CNNs). The outstanding performance of fault diagnosis CNNs results from complex and redundant network structures and parameters, resulting in huge storage and computational requirements, which makes it challenging to implement these models in resource-limited industrial devices. This study aims to address this problem by proposing a comprehensive compression method for CNNs that is applied to intelligent fault diagnosis. It involves several different compression methods, including tensor train decomposition, parameter quantization, and knowledge distillation for deep network compression. This results in a significant decrease in redundancy and speeding up the training of CNN models. Firstly, tensor train decomposition is applied to reduce redundant connections in both convolutional and fully connected layers. The next step is to perform parameter quantization to minimize the bits needed for parameter representation and storage. Finally, knowledge distillation is used to restore accuracy to the compressed model. The effectiveness of the proposed approach is confirmed by an experiment and ablation study with different models on several datasets. The results show that it can significantly reduce redundant information and floating-point operations with little degradation in accuracy. Notably, on the CWRU dataset, with about 60% parameter reduction, there is no degradation in our model's accuracy. The proposed approach is a new attempt at the intelligent fault diagnosis of rolling bearings in industrial equipment.

Keywords: intelligent fault diagnosis; convolutional neural networks; network compression; tensor train decomposition; parameter quantization; knowledge distillation

Citation: Wang, X.; Hu, D.; Fan, X.; Liu, H.; Yang, C. Intelligent Fault Diagnosis Method Based on Neural Network Compression for Rolling Bearings. *Symmetry* **2024**, *16*, 1461. https://doi.org/10.3390/sym16111461

Academic Editor: Sergei Odintsov

Received: 29 September 2024
Revised: 27 October 2024
Accepted: 28 October 2024
Published: 4 November 2024

1. Introduction

Due to its structural symmetry and rotation mode, rolling bearings are a crucial part of rotating devices. They often experience increasingly severe wear and tear over time, leading to bearing failures and impact on the operation of the entire device, resulting in serious accidents and economic losses. Consequently, studying rolling bearing fault diagnosis in real time is vital to preserve the equipment's steady performance.

As artificial intelligence (AI) technology advances quickly, neural networks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), have been extensively applied in fault diagnosis. Among them, CNNs are the most widely utilized and have achieved impressive success.

CNN-based diagnostic methods mainly improve diagnostic performance through more complex and redundant structures and parameters in networks, which results in high hardware computational requirements for the process of training and inference. This factor significantly imposes limitations on the practical application of such intelligent diagnostic methods in industrial situations with restricted resources. Studies have shown that there is

usually a significant amount of redundancy in CNNs, and the impact of this redundant information on diagnostic performance is small or even negligible. Therefore, removing the redundancy in the model can be considered to achieve efficient compression of neural networks while taking into account their diagnostic accuracy. Deep compression techniques for neural networks are well suited for industrial applications to improve the performance of online monitoring and diagnostics.

The aim of neural network compression is to compress CNNs with a large amount of redundancy so that they can be deployed on edge industrial devices with restricted computational and storage capacity. Compression methods of CNNs have been widely studied and have achieved significant success in related fields, such as image classification and recognition. The primary methods of CNN compression include network pruning [1,2], parameter quantization [3,4], tensor decomposition [5,6], and knowledge distillation [7,8]. Tensor decomposition is one of the most widely utilized compression methods to compress both convolutional and fully connected layers. Among all the decomposition methods, tensor train decomposition has the highest decomposition efficiency [9]. Parameter quantization converts floating-point numbers to low-bit representations, which helps to improve the efficiency of the deployment of networks. Knowledge distillation enables knowledge from large networks to be extracted and transferred to small networks to enhance the performance of small models on devices with limited resources.

In order to enhance diagnostic accuracy, the current CNN-based bearing fault diagnosis methods rely on sophisticated models. However, the deployment of the models requires a significant amount of storage and computing resources due to the large parameters and FLOPs, which results in challenges in online fault diagnosis. This significantly impairs the practicality of intelligent fault diagnosis methods. To tackle the above problems, this paper presents a multiple compression approach that combines tensor decomposition, parameter quantization, and knowledge distillation. It decomposes both the convolutional and fully connected layers of the network and then further reduces the required storage by quantizing its parameters. Finally, it can restore accuracy by training the compressed network as a student model using the original uncompressed network by knowledge distillation.

Our main contributions are as follows:

1. A new multiple compression approach is introduced for CNN-based fault diagnosis for the first time, which greatly reduces the parameters and FLOPs of CNNs and maintains accuracy at the same time. This indicates that the proposed approach can compress the fault diagnostic network multiple times while maintaining performance.
2. The proposed approach is combined with tensor train decomposition, parameter quantization, and knowledge distillation, and their compression effects are investigated separately. Tensor train decomposition can drastically reduce storage and computing requirements, and the parameter quantization can contribute to further reducing the storage. The knowledge distillation can restore the accuracy of the compressed network.
3. The effectiveness of the comprehensive compression method proposed in this paper has been demonstrated through comparative experiments and ablation studies. The results indicate that our approach has achieved relatively advanced performance.
4. The proposed method enhances the efficiency of the diagnostic model based on CNNs. It gives future studies a theoretical foundation and offers novel insights for the implementation of intelligent fault diagnosis in industrial equipment.

2. Related Work

For more than 30 years, intelligent bearing fault diagnosis has garnered widespread interest in the industry. The various kinds of diagnostic approaches can generally be categorized into two types: traditional methods [10] and methods based on neural networks [11].

The first type of fault diagnosis comprises traditional methods, which mainly focus on signal feature extraction and pattern classification. Typical signal processing analyzes the input signal with different techniques in the time or frequency domain [12], such as fast

Fourier transform [13], wavelet transform [14], and empirical mode decomposition [15]. Xu et al. [16] introduced a technique that involves inverse short-time Fourier transform (ISTFT) to diagnose failures of rotating equipment. In the study of [17], the diagnosis of failures was introduced using the discrete wavelet transform (DWT) method. Fan et al. [18] applied empirical mode decomposition (EMD) to diagnose the different faults of rotating machinery. Gu et al. [19] used ensemble empirical mode decomposition (EEMD) to obtain rolling bearing fault signal information. Rai et al. [20] used the Hilbert transform (HT) to analyze bearing vibration signals. Traditional fault diagnosis approaches often require manual design of suitable fault features and the signal process requires a great deal of a priori knowledge [21].

The second type of method is based on neural networks. The rapid advancement of AI technology has resulted in the success of fault diagnosis approaches based on CNNs. CNNs have excellent abilities in feature extraction and classification [22]. By using the original monitoring signals as direct inputs, the model provides efficient and fast end-to-end fault analysis [23]. The initial application of CNNs for fault diagnosis was by Feng et al. [24]. It uses a five-layer autoencoder that can adaptively extract features from the monitoring data. Liang et al. [25] used the residual connection to enhance the feature extraction capability of a one-dimensional dilated CNN in the diagnosis process. In the study of [26], a five-layer WDCNN model was introduced to diagnose the fault of bearings. The model is capable of achieving high accuracy on large datasets compared to other methods. Che et al. [27] introduced a deep belief network (DBN) to solve the issue of the lack of label samples for fault diagnosis under the new operating circumstances. In the work of [28], a data-driven timed failure propagation graph (TFPG) construction method was proposed using a new type of tree-structured LSTM model for fault diagnosis. Wang et al. [29] introduced a CNN structure automated search technique utilizing reinforcement learning for fault diagnosis.

Nevertheless, fault diagnosis based on CNNs has the following issues: the current fault diagnostic networks contain numerous parameters that put higher and higher demands on the capability and memory of the device, making them challenging to incorporate in embedded systems [30]. There is a growing focus on neural network compression methods to optimize neural network size and meet performance requirements simultaneously.

Currently, there are several main types of neural network compression methods, including pruning, low-rank tensor decomposition, quantization, compact network design, knowledge distillation, and neural architecture search, whose common goal is to reduce the size of the CNN in different ways without losing performance, in order to achieve balance between the performance of the network and the simplification of the structure. Zhu et al. [31] claimed a diagnosis approach with a new stacked pruning sparse denoising autoencoder (sPSDAE) to enhance the inference efficiency of the compressed CNN. Ding et al. [32] introduced an adaptive pruning approach for removing useless structures when training the fault diagnosis model. Yao et al. [33] introduced a stacked inverted residual CNN with rapid training efficiency and high robustness to reduce the hardware demands for diagnosis performance. Deng et al. [34] applied the knowledge distillation method for the first time in fault diagnosis, which can reduce the harmful impacts of unbalanced data and enhance the model's performance. The study [35] applied the distillation quantization approach to minimize the size of the CNN model deployed on edge hardware for fault diagnosis. Li et al. [36] applied Neural Architecture Search to fault diagnosis for the first time, which can automatically design the most appropriate neural network. In the work of [37], a new differentiable neural structure search method was proposed to generate subnetworks with lower complexity and computational cost. In general, neural network compression techniques are still less researched and applied in intelligent diagnostic systems. Although these attempts to compress the diagnostic network have achieved good results, they are still not sufficiently effective. For example, combining different compression techniques to improve performance has not been considered.

3. The Proposed Method

In industrial applications, CNN-based intelligent diagnosis of bearing faults has many limitations, including parameter redundancy and significant computing and storage costs. To resolve these issues, a hybrid compression approach is proposed that combines tensor train decomposition, parameter quantization, and knowledge distillation. It is possible to achieve greater model compression efficiency by combining these approaches while maintaining a high-performance standard for the compressed model.

3.1. Tensor Train Decomposition

In recent years, tensor train (TT) decomposition has become a more prevalent choice as an advanced method for tensor decomposition [38]. This method can achieve very high compression ratios, and its great compression advantage has been demonstrated and validated in the application of recurrent neural network (RNN) compression in video recognition tasks [9].

Given a tensor $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, it can be decomposed into 3-order tensors using TT decomposition:

$$\begin{aligned} \boldsymbol{A}_{(i_1, i_2, \cdots, i_d)} &= \boldsymbol{G}_{1(:, i_1, :)} \boldsymbol{G}_{2(:, i_2, :)} \cdots \boldsymbol{G}_{d_{(:, i_d, :)}} \\ &= \sum_{r_0, r_1, \cdots r_d}^{\alpha_0, \alpha_1, \cdots \alpha_d} \boldsymbol{G}_{1(\alpha_0, i_1, \alpha_1)} \boldsymbol{G}_{2(\alpha_1, i_2, \alpha_2)} \ldots \boldsymbol{G}_{d(\alpha_{d-1}, i_d, \alpha_d)}, \end{aligned} \tag{1}$$

where $\boldsymbol{G}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k} (k = 1, 2, \cdots, d)$ are TT cores, and $\boldsymbol{r} = [r_0, r_1, \cdots, r_d]$, $r_0 = r_d = 1$ are TT ranks, which indicate the compression rate of the compressed TT-format tensor. The process of TT decomposition is displayed in Figure 1.

For a fully connected layer, its weight matrix is $w \in \mathbb{R}^{M \times N}$, where $M = \prod_{k=1}^{d} m_k$ and $N = \prod_{k=1}^{d} n_k$. The output $y \in \mathbb{R}^M$ can be obtained from the input $x \in \mathbb{R}^N$ by $y = wx$. To convert this fully connected layer into TT format, we first reshape and order-transpose the weight matrix w to a weight tensor $\boldsymbol{W} \in \mathbb{R}^{(m_1 \times n_1) \times \cdots \times (m_d \times n_d)}$. Then, $\boldsymbol{W}$ can be decomposed as

$$\boldsymbol{W}_{((i_1, j_1), \cdots, (i_d, j_d))} = \boldsymbol{G}_{1(:, i_1, j_1, :)} \cdots \boldsymbol{G}_{d(:, i_d, j_d, :)}. \tag{2}$$

Here, each TT core $\boldsymbol{G}_k \in \mathbb{R}^{r_{k-1} \times m_k \times n_k \times r_k}$ is a 4-order tensor. Consequently, the tensor format representation of the forward propagation of this layer is as follows:

$$\boldsymbol{Y}_{(i_1, \cdots, i_d)} = \sum_{j_1, \cdots, j_d} \boldsymbol{G}_{1(:, i_1, j_1, :)} \cdots \boldsymbol{G}_{d(:, i_d, j_d, :)} \boldsymbol{X}_{(j_1, \cdots, j_d)}, \tag{3}$$

where $\boldsymbol{X} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ and $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ represent the tensorized input and output for x and y, respectively.

For a standard convolutional layer, it generates an output tensor $\widetilde{\boldsymbol{Y}} \in \mathbb{R}^{(W-K+1) \times (H-K+1) \times M}$ by applying convolution between an input tensor $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{W \times H \times N}$ and a weight tensor $\widetilde{\boldsymbol{W}} \in \mathbb{R}^{K \times K \times M \times N}$. To transform this convolutional layer into TT format, the input tensor $\widetilde{\boldsymbol{X}}$ can be reshaped to $\boldsymbol{X} \in \mathbb{R}^{W \times H \times n_1 \times \cdots \times n_d}$, and the weight tensor $\widetilde{\boldsymbol{W}}$ can be reshaped and transposed to $\boldsymbol{W} \in \mathbb{R}^{(K \times K) \times (m_1 \times n_1) \times \cdots \times (m_d \times n_d)}$, and then it can be decomposed as

$$\boldsymbol{W}_{((k_1, k_2), (i_1, j_1), \cdots, (i_d, j_d))} = \boldsymbol{G}_{0(k_1, k_2)} \boldsymbol{G}_{1(:, i_1, j_1, :)} \cdots \boldsymbol{G}_{d(:, i_d, j_d, :)}, \tag{4}$$

where $M = \prod_{k=1}^{d} m_k$ and $N = \prod_{k=1}^{d} n_k$, and $\boldsymbol{G}_k \in \mathbb{R}^{r_{k-1} \times m_k \times n_k \times r_k}$ is a 4-order tensor with the exception of $\boldsymbol{G}_0 \in \mathbb{R}^{K \times K}$. Then, the new output tensor is $\boldsymbol{Y} \in \mathbb{R}^{(W-K+1) \times (H-K+1) \times m_1 \cdots \times m_d}$, which can be achieved as follows:

$$\boldsymbol{Y}_{(w, h, i_1, \cdots, i_d)} = \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \sum_{j_1, \cdots, j_d} \boldsymbol{X}_{(k_1+w-1, k_2+h-1, j_1, \cdots, j_d)} \boldsymbol{G}_{0(k_1, k_2)} \boldsymbol{G}_{1(:, i_1, j_1, :)} \cdots \boldsymbol{G}_{d(:, i_d, j_d, :)}. \tag{5}$$

The associated forward propagation is created using the TT-format fully connected and convolutional layer. Then, the stochastic gradient descent (SGD) technique is utilized for refreshing the TT cores with a target compression ratio, which depends on the rank set r.

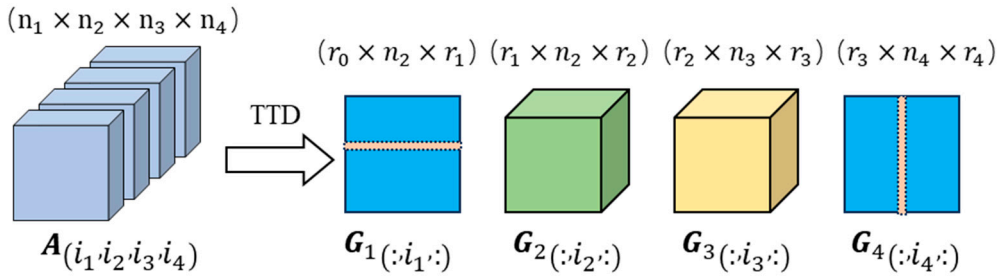


Figure 1. Tensor train decomposition (TTD) of a 4-order tensor, where the dimension of r_0 and r_4 are always set to 1.

3.2. Parameter Quantization

Parameter quantization has made significant developments in the field of CNN compression. However, not much work has been carried out in the domain of applying the parameter quantization approach to fault diagnosis. The purpose of parameter quantization is to accomplish network compression by minimizing the bits needed for representing and storing each parameter. Weight-sharing and low-bit representation are two basic quantization methods. The quantization of the parameters can significantly decrease the storage of the network. It is highly significant for the deployment of CNNs on embedded equipment.

The weight-sharing method uses K-mean clustering for each layer's weight matrices to obtain the corresponding cluster centers. Then, the clustering center is used to replace all the weights in the cluster. Finally, just the clustering center and index should be saved. Figure 2 depicts the whole procedure, with the same cluster denoted by the same color.

The weights are converted from their initial 32-bit floating-point representation to a clustering center (32-bit) and a clustering index (2-bit). This conversion dramatically decreases the amount of storage. If the clustering group is indicated as k, the total amount of indexes required is $\log_2(k)$ bits. In a CNN model with n weights, where each weight is indicated by b bits, the compression rate R is usually expressed as

$$R = \frac{nb}{n\log_2(k) + kb} \tag{6}$$

Then, we consider the low-bit representation of the clustering center. This can convert 32-bit floating-point value clustering centers to integers.

To represent the floating-point value in a low-bit integers, the following steps are taken:

1. Find the range of values for the weights that need to be quantified among all the weights in the uncompressed model and set the maximum and minimum values to be x_{max} and x_{min}.
2. Determine the interval $[y_{min}, y_{max}]$ of values after quantization of all weights. An asymmetric quantization approach is utilized with the quantization range selected as [0, 255].
3. Determine the scaling factor S and the zero point Z as follows:

$$\begin{aligned} S &= \tfrac{x_{max} - x_{min}}{255 - 0} \\ Z &= \text{round}\left(255 - \tfrac{x_{max}}{S}\right) \end{aligned} \tag{7}$$

4. For a weight x represented by a floating-point number, quantize the floating-point value to an integer y with the following formulation:

$$y = \text{round}\left(\frac{x}{S} + Z\right) \tag{8}$$

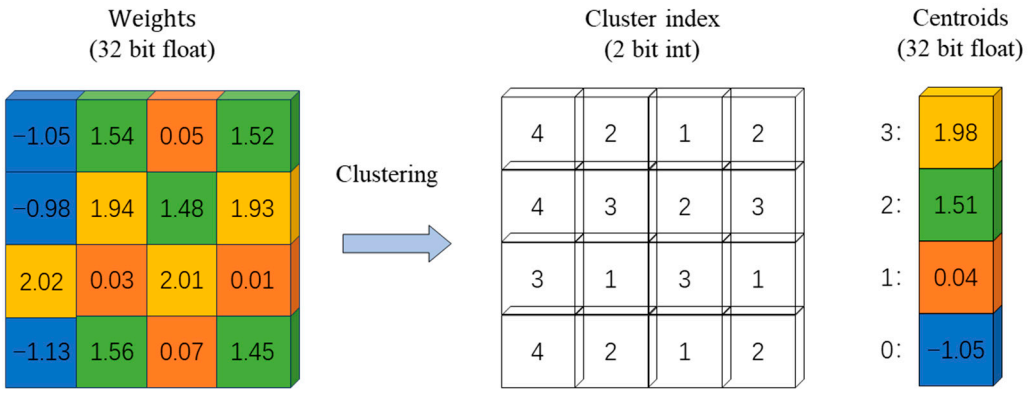


Figure 2. The process of quantization by weight-sharing clustering. Assuming that the weights in a convolutional layer are a 4×4 matrix, the weights are quantized into four categories, each represented by different colors, and all the weights in the same category have the same value. Therefore, each weight only requires a tiny 2-bit clustering index.

3.3. Knowledge Distillation

Knowledge distillation belongs to a kind of transfer learning method. It is obvious that the tensor train decomposition and parameter quantization mainly delete the redundant parameters and reduce the storage of the network to achieve network compression. On the other hand, knowledge distillation focuses on achieving lightweight networks by directly training small networks. It uses small networks as student models and large networks with superior performance as teacher models and obtains small networks with good performance through training. Tensor train decomposition and parameter quantization can simplify the CNN. However, with the reduction in network parameters, the model's accuracy drops dramatically, and the knowledge distillation approach can be implemented to enhance the model's accuracy. So, the original neural network before compression is used as the teacher model, and the compressed model after tensor train decomposition and parameter quantization is used as the student model, which implements knowledge distillation to achieve the purpose of restoring the network accuracy. Knowledge distillation, tensor train decomposition and parameter quantization do not conflict with each other, and their combination can better balance the purpose of streamlining network structure and obtaining similar performance.

The knowledge distillation approach applies soft labels produced by the teacher model to control the training of the student model [39]. When the output of the teacher model is utilized as the training target (soft labels) for the student model, and the original data of the student model are used as the hard labels, the student model's accuracy will be close to that of the teacher model. Obviously, soft labels carry more knowledge than hard labels. Through knowledge distillation, the teacher model can teach knowledge to the student model. The student model can gain a wealth of knowledge while being very small.

To effectively transfer knowledge, the loss functions of the teacher and student model must be optimized. Therefore, the softmax function of the model can be defined as

$$q_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}, \tag{9}$$

where q_i is the soft label for student models; z_i and z_j are the logits of the student and teacher models, respectively; and T, also known as "temperature", is the parameter used to control the distillation process. A higher T value produces a softer and smoother probability distribution on the output, and if T equals infinity, it is a uniform distribution. When T goes toward 0, the probability distribution of the softmax function becomes more concentrated.

Both soft and hard label loss are components of the loss function used to train the student model. The hard label loss is the cross-entropy between the output of the student model and the labels. Then, the soft label loss can minimize the probability distributions of the teacher and the student model by K-L divergence, which enables the probability distribution of the teacher model to be simulated by the student model. So, the hard and soft label loss of the loss function are as follows:

$$\begin{aligned} Loss_{hard} &= -\sum_{x} q_S(x)\log(p_S(x)) \\ Loss_{soft} &= A\sum_{x}[-q_T(x)\log(q_S(x))] \end{aligned} \tag{10}$$

where $q_S(x)$ and $q_T(x)$ are the estimated distributions of the student model and the teacher model, $p_S(x)$ denotes the target distribution, and A is the adjustment coefficient.

With Equations (9) and (10), the gradient of $Loss_{soft}$ can be denoted as

$$\frac{\partial Loss_{soft}}{\partial z_i} = \frac{1}{T}(q_{Ti} - q_{Si}) = \frac{1}{T}\left(\frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)} - \frac{\exp\left(\frac{v_i}{T}\right)}{\sum_j \exp\left(\frac{v_j}{T}\right)}\right) \tag{11}$$

where v_i is the output of the teacher model.

When $T \to 0$, Equation (11) can be approximated and simplified as

$$\begin{aligned} \frac{\partial Loss_{soft}}{\partial z_i} &\approx \frac{1}{T}\left(\frac{1+\frac{z_i}{T}}{\sum_j\left(1+\frac{z_j}{T}\right)} - \frac{1+\frac{v_i}{T}}{\sum_j\left(1+\frac{v_j}{T}\right)}\right) \\ &= \frac{1}{T}\left(\frac{1+\frac{z_i}{T}}{N+\sum_j \frac{z_j}{T}} - \frac{1+\frac{v_i}{Tz}}{N+\sum_j \frac{v_j}{T}}\right) \end{aligned} \tag{12}$$

If every sample is subjected to a zero average with the logits function, then $\sum_j z_j = \sum_j v_j = 0$. Therefore, the gradient of the $Loss_{soft}$ can be simplified as

$$\begin{aligned} \frac{\partial Loss_{soft}}{\partial z_i} &\approx \frac{1}{T}\left(\frac{1+\frac{z_i}{T}}{N} - \frac{1+\frac{v_i}{T}}{N}\right) \\ &\approx \frac{1}{NT^2}(z_i - v_i) \end{aligned} \tag{13}$$

Finally, the student model's overall loss function is

$$\text{Loss}_{student} = \quad (1-\alpha)\sum_{x} q_S(x)\log(p_S(x)) + \alpha * T^2\sum_{x}[-q_T(x)\log(q_S(x))] \tag{14}$$

3.4. Strategy of the Compressing Method

The proposed approach is intended to deal with the following issue: CNNs for fault diagnosis are hard to implement in embedded industrial devices due to constrained computing and storage resources. Therefore, a hybrid CNN compression approach is proposed, which combines the tensor train decomposition, parameter quantization, and knowledge distillation.

Firstly, the irrelevant and redundant structures and parameters in the convolutional and fully connected layer are removed by tensor train decomposition. It reduces the FLOPs and speeds up model training. Secondly, the remainder of the parameters are quantized to decrease their storage requirements through weight-sharing and low-bit representation. Finally, the compressed model (student model) is trained with knowledge distillation. The student model's loss function is optimized throughout the training stage. The loss function

combines both the soft loss from the original uncompressed model (teacher model) and the hard loss obtained from the original data of the student model. By applying this loss function, the convergence of the student model can be accelerated, and its accuracy can also be enhanced. So, the student model's whole training process can be facilitated. A flowchart of the approach in this section is shown in Figure 3.

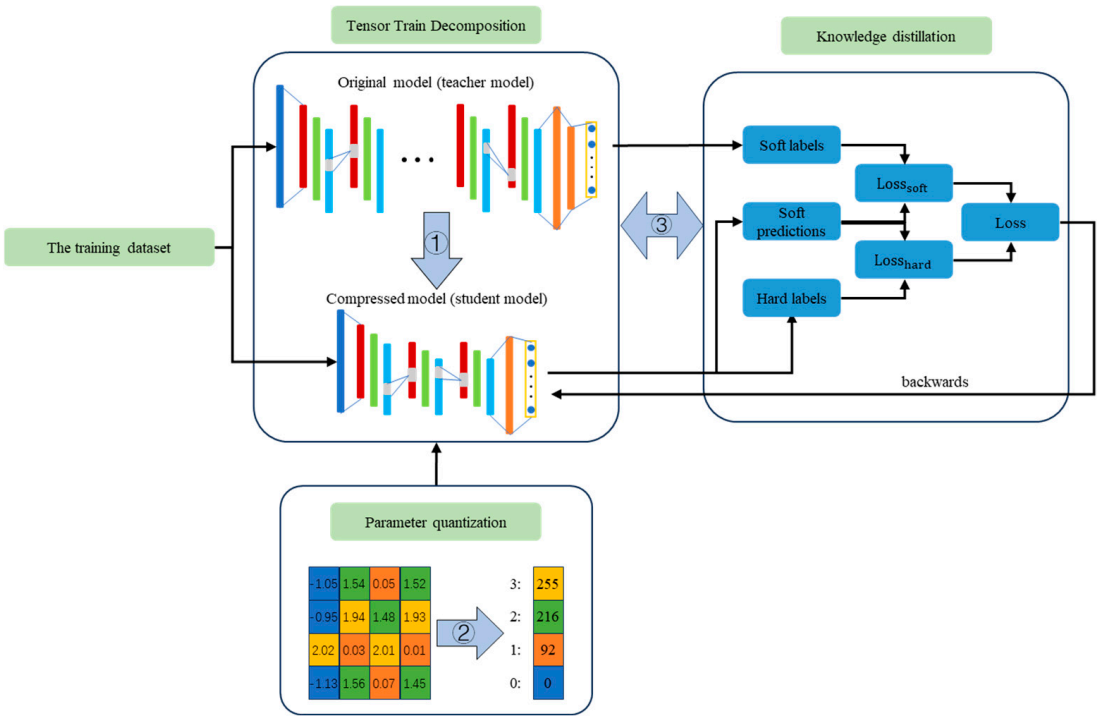


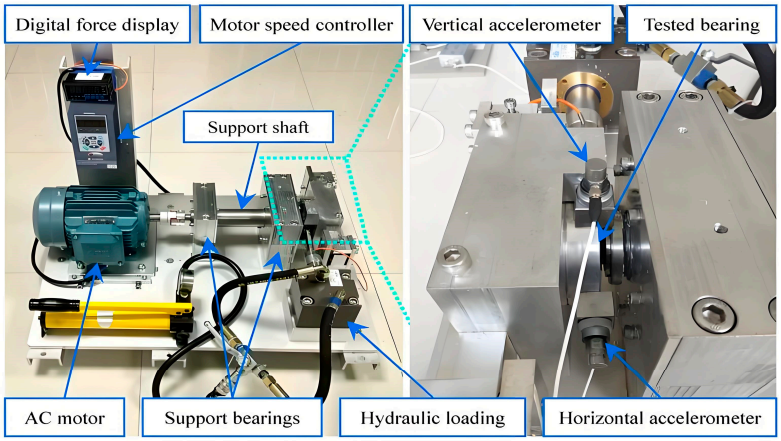
Figure 3. The flowchart of the compression method for the bearing fault diagnosis in this section.

4. Experimental Results

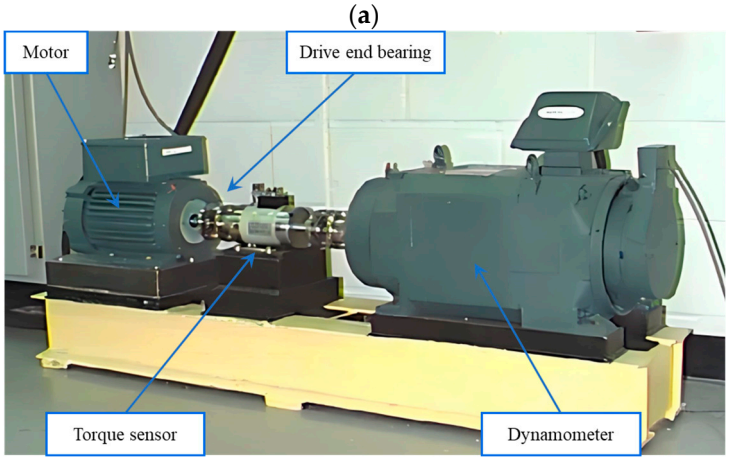
4.1. Experimental Setup

Datasets. Firstly, the effectiveness of the proposed approach was confirmed with the XJTU-SY bearings dataset, which was provided by Xi'an Jiaotong University. The study of [40] presents a complete description of the experimental setup. The experimental platform for obtaining the bearing dataset is shown in Figure 4a. The tested bearings failed due to a variety of problems, including inner race wear, cage fracture, outer race wear, outer race fracture, etc. The data samples were separated into three operating conditions at different rotational speeds and radial forces, each containing a different type of bearing failure. The details of the dataset are presented in Table 1.

The second dataset was from the Bearing Data Centre at Case Western Reserve University (CWRU) [41]. The experimental platform for obtaining the bearing dataset is shown in Figure 4b. The dataset sample was taken from SKF 6205 rolling bearings with three types of faults: inner race, outer race, and rolling body. Each type of fault is separated into three situations: fault diameter of 0.18 mm, 0.36 mm, and 0.54 mm. The details of the dataset are presented in Table 2.



(**a**)



(**b**)

Figure 4. The experimental platform of two bearing datasets: (**a**) Experimental equipment of XJTU-SY dataset [40]. (**b**) Experimental equipment of CWRU dataset [41].

Table 1. Details of XJTU-SY dataset.

Operation Condition	Bearing Dataset	Fault Types	Labels
Condition 1	Bearing1_1	Outer race	0
	Bearing1_4	Cage	1
	Bearing1_5	Inner and outer race	2
Condition 2	Bearing2_1	Inner race	3
	Bearing2_2	Outer race	4
	Bearing2_3	Cage	5
Condition 3	Bearing3_1	Outer race	6
	Bearing3_2	Cage; inner and outer race	7
	Bearing3_3	Inner race	8

Model structure. To demonstrate the proposed fault diagnosis neural network compression approach, two classical models with various depths [42,43] were chosen to test and assess the compression abilities of the approach. The specific situations of each layer involved in the two CNN models used in this section are presented in Table 3. The first CNN model, denoted as CNN-1 in the table, comprises two convolutional and BN layers, two fully connected layers, and a softmax layer. The second model, denoted as CNN-2 in the table, with deeper network depth, has two more convolutional layers.

Table 2. Details of CWRU dataset.

Fault Diameter	Fault Types	Labels
-	Normal	0
0.18 mm	Inner race	1
	Outer race	2
	Rolling body	3
0.36 mm	Inner race	4
	Outer race	5
	Rolling body	6
0.54 mm	Inner race	7
	Outer race	8
	Rolling body	9

Table 3. Details of structures of two test models.

Model	Layer	Layer Type	Kernel Number	Kernel Size
CNN-1	1	Conv	32	5×5
	2	Maxpool	/	2×2
	3	Conv	64	5×5
	4	Maxpool	/	2×2
	5	FC1	1024	/
	6	Dropout	/	/
	7	FC2	256	/
	8	Dropout	/	/
	9	Softmax	/	/
CNN-2	1	Conv	32	5×5
	2	Maxpool	/	2×2
	3	Conv	64	3×3
	4	Maxpool	/	2×2
	5	Conv	128	3×3
	6	Maxpool	/	2×2
	7	Conv	256	3×3
	8	Maxpool	/	2×2
	9	FC1	2560	/
	10	Dropout	/	/
	11	FC2	768	/
	12	Dropout	/	/
	13	Softmax	/	/

Data processing. At present, for 1D bearing vibration signals that require fault diagnosis, neural network models cannot achieve excellent end-to-end performance. Therefore, it is necessary to preprocess the input signals. In this study, to better process the input signal, we effectively transformed the original 1D signal into a 2D input signal. Firstly, we truncated the original 1D data at specific intervals M^2, and then obtained the results $L(n), n \in \{1, 2, \ldots, M^2\}$, and denoted the value of the n-th point in the truncation of the original input. Finally, we obtained the following conversion formula:

$$P(i,j) = \text{round}\left\{\frac{L((i-1) \times M + \text{j}) - \text{Min}(L)}{\text{Max}(L) - \text{Min}(L)} \times 255\right\}, \tag{15}$$

where $P(i, j), i, j \in \{1, 2, \ldots, M\}$ denotes the intensity at point (i, j). With the conversion, the original 1D signal with length M^2 is transformed into a 2D greyscale map of size $\mathbb{R}^{M \times M}$ with scale [0, 255].

Evaluation metrics. The evaluation metrics of the CNN compression method usually involve accuracy, the number of parameters, and FLOPs. The accuracy quantifies the loss of diagnostic ability of the model before and after compression; the number of parameters

denotes the storage that the model occupies when it is operating; and FLOPs evaluate the amount of computation required by the model during the fault diagnosis process.

4.2. The Results of Compression Performance

In this section, CNN-1 and CNN-2 are compressed on XJTU-SY datasets and CWRU datasets, respectively, and the classification accuracy reduction, parameter reduction, and FLOP reduction before and after compression are obtained. The results are presented in Table 4.

Table 4. Compression performance of the proposed approach in different models and datasets.

Datasets	Models	Accuracy Reduction (%)	Parameter Reduction (%)	FLOP Reduction (%)
XJTU-SY	CNN-1	0.00	59.6	56.3
		1.51	82.0	88.4
	CNN-2	0.00	65.2	69.9
		1.66	85.4	90.6
CWRU	CNN-1	0.00	59.8	60.7
		2.06	87.5	87.9
	CNN-2	0.00	69.2	75.4
		2.33	87.8	89.5

Using our method on the XJTU-SY dataset, the parameters of CNN-1 can be reduced by 59.6% compared to the original model, and FLOPs can be reduced by 56.3% of the original model with no impact on accuracy. Meanwhile, in the case of a 1.51% decrease in accuracy, parameters and FLOP reduction can be up to 82.0% and 88.4%, respectively. For CNN-2, when the parameter and FLOP reductions are 65.2% and 69.9%, respectively, the accuracy of the compressed model has not decreased yet. When the parameters and FLOPs drop by 85.4 and 90.6%, the accuracy of the compressed model only decrease by 1.66%. Similar results can also be observed in the CWRU dataset. These results indicate that our method can achieve excellent compression performance for neural networks. By adopting our method, the storage for parameters in industrial devices can be substantially reduced, and the computation operation of the model can be considerably accelerated.

4.3. Comparison with Other Methods

The proposed compression method was compared with several common compression methods. Most of these methods use a single type of compression method, such as Pruning Filters [44] or HRank [45], which only use pruning to achieve network compression; Tucker decomposition [6] and CP decomposition [46] also only use tensor decomposition methods to compress the network. In addition, there is a Quantized CNN method [47] to achieve quantization for compression. Table 5 lists the experimental results of these methods on CNN-1 and CNN-2 models, and the evaluation criteria for the comparison include three metrics: accuracy reduction, parameter reduction, and FLOP reduction.

From Table 5, for both models compressed on the XJTU-SY dataset, the results of our approach are considerably higher than the other five methods regarding parameter reduction and FLOP reduction. At the same time, it has the lowest drop in accuracy. This indicates that the performance of our approach surpasses that of other approaches.

Table 5. Compression performance with different approaches on the XJTU-SY dataset.

Datasets	Models	Methods	Accuracy Reduction (%)	Parameter Reduction (%)	FLOP Reduction (%)
XJTU-SY	CNN-1	Pruning Filters	7.74	43.1	48.5
		HRank	6.55	49.2	53.7
		Tucker	3.89	63.5	65.6
		CP	3.69	69.8	75.5
		Quantized CNN	4.33	55.1	54.9
		Ours	1.51	82.0	88.4
	CNN-2	Pruning Filters	8.48	48.3	50.0
		HRank	8.35	52.2	52.7
		Tucker	4.68	65.4	64.0
		CP	4.56	73.8	74.1
		Quantized CNN	5.44	60.1	59.9
		Ours	1.66	85.4	90.6

Specifically, for the compression of the CNN-1 model, compared to the Pruning Filters and HRank method, Tucker and CP decomposition based on tensor decomposition can achieve a lower accuracy drop under a higher parameter compression rate, indicating that tensor decomposition methods can achieve better compression efficiency compared to simple pruning methods. On the contrary, the quantization-based compression methods do not achieve better compression results, which suggests that there is no advantage in using quantization alone to compress the model, and that using it in conjunction with other methods is a better choice. Finally, our approach achieves the best compression results, with the compressed model achieving 82.0% and 88.4% compression of the model parameter and FLOPs, respectively, with a slight decrease in accuracy (1.51%). In brief, our method acquired the best overall performance. The same conclusion can be obtained from the CNN-2 model. The difference is that since the CNN-2 model has deeper layers, a higher compression rate of parameters and FLOPs can be achieved with a low drop in accuracy. Our compression method can eliminate 85% of the parameters, and the compression of FLOPs can reach more than 90%.

This section also tests the response time of the CNN-1 and CNN-2 with the compression methods of Pruning Filters, HRank, Tucker, CP and our method, respectively. The results are shown in Figure 5. For both models CNN-1 and CNN-2, our proposed method significantly reduces the response time of the compressed model compared to other algorithms. Therefore, our method can make the compressed model have better real-time diagnosis ability.

We also conducted a comparison between our approach and traditional machine learning techniques to demonstrate its efficacy in fault diagnosis. To confirm the advantages of the model compressed by our method compared to traditional machine learning methods, we compared the accuracy and the parameters of the compressed model obtained by our method with traditional machine learning models on the CWRU dataset. Some traditional machine learning methods have provided promising results in fault diagnosis, such as support vector machine (SVM) [48], k-nearest neighbor (KNN) [49], stacked autoencoder (SAE) [50], and so on.

From Table 6, the performance of CNN model compressed by our method is better than the traditional machine learning model regarding diagnostic accuracy, and the model parameters are lower. This indicates that under the same conditions, the model built with our method can provide better diagnostic results with significant performance, while our model requires the least storage resources.

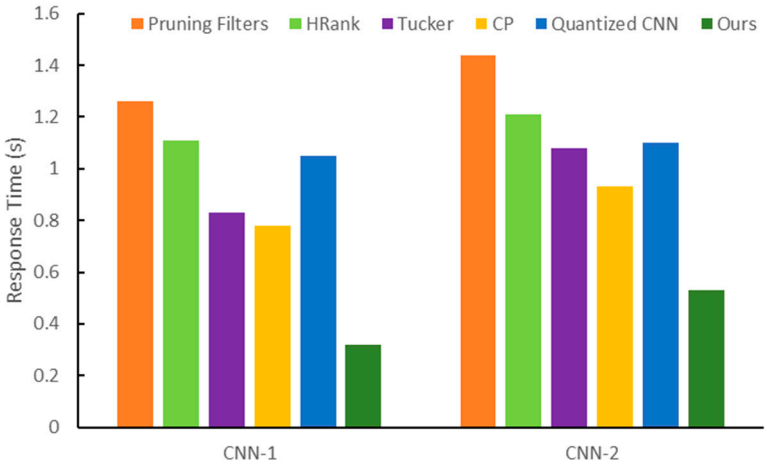


Figure 5. Comparison of response time with different methods on the XJTU-SY dataset.

Table 6. Performance with different approaches on the CWRU dataset.

Methods	Accuracy (%)	Parameters (M)
SVM	80.16	44.37
KNN	89.32	86.46
SAE	93.03	23.18
Ours—CNN-1	96.41	10.20
Ours—CNN-2	98.05	14.24

4.4. The Ablation Study

We performed sufficient ablation studies to test our compression method's effectiveness. For brevity, we performed all experiments on the XJTU-SY dataset. Similar observations can be seen for other datasets.

The first ablation study aimed to verify how well the knowledge distillation method works. Firstly, tensor train decomposition and quantization of the model were performed with different compression rates, and the accuracy curves of the compressed model at this point were obtained. Then, knowledge distillation was applied to the compressed model to restore its accuracy and obtain a new accuracy curve. Finally, as a comparison, the traditional fine-tuning-based method for accuracy recovery of the compressed model was used to observe the effect comparison.

The results can be observed in Figure 6, where the accuracy of the compressed model after knowledge distillation is far superior to that of the model before knowledge distillation under different compression rate conditions. Meanwhile, knowledge distillation can also achieve higher model accuracy compared to fine-tuning methods. As can be seen, with the rise in compression ratio, the accuracy gap between the models obtained by the two methods further widens. However, when the compression rate reaches 90%, the accuracy of the model obtained by the knowledge distillation process shows a rapid decline, which indicates that the number of parameters in the model at this point is too low to adequately extract and express the fault features in intelligent diagnosis.

Figure 7 illustrates the accuracy and loss curves of two CNN models for two different accuracy recovery methods. From Figure 7a,c, the accuracy of the model obtained by knowledge distillation is higher than that of the common fine-tuning method in the beginning phases of the training process, and when the training reaches a later phase, there is still a significant gap between two accuracy curves. From Figure 7b,d, the model from the fine-tuning method has the greatest loss during the training process, and the model with the proposed method achieves the least training loss.

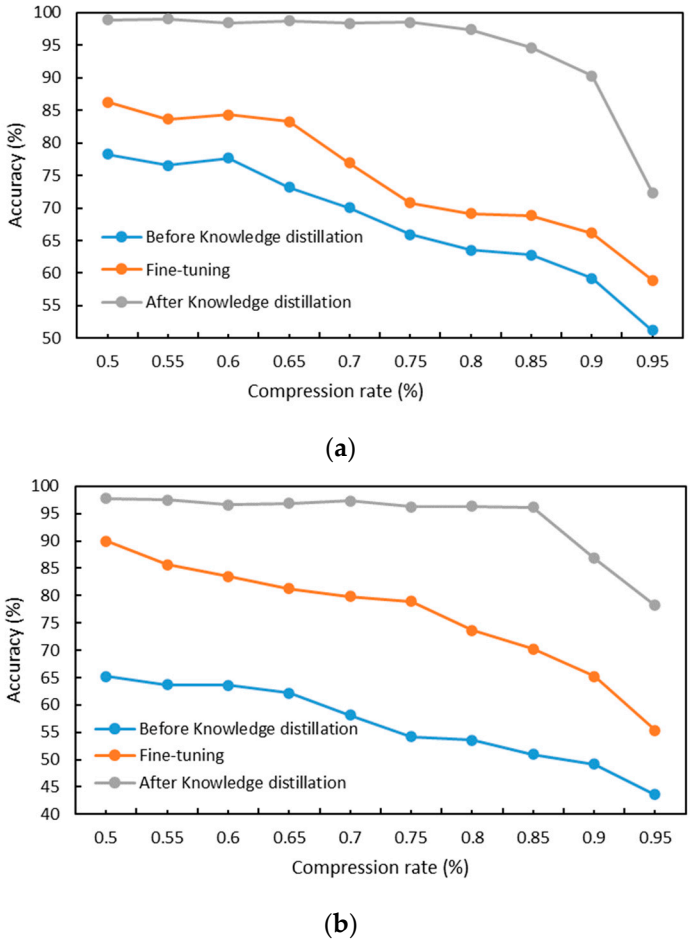


Figure 6. The accuracy curve before and after distillation of knowledge with different compression rates on the XJTU-SY dataset: (**a**) Accuracy curve of CNN-1; (**b**) Accuracy curve of CNN-2.

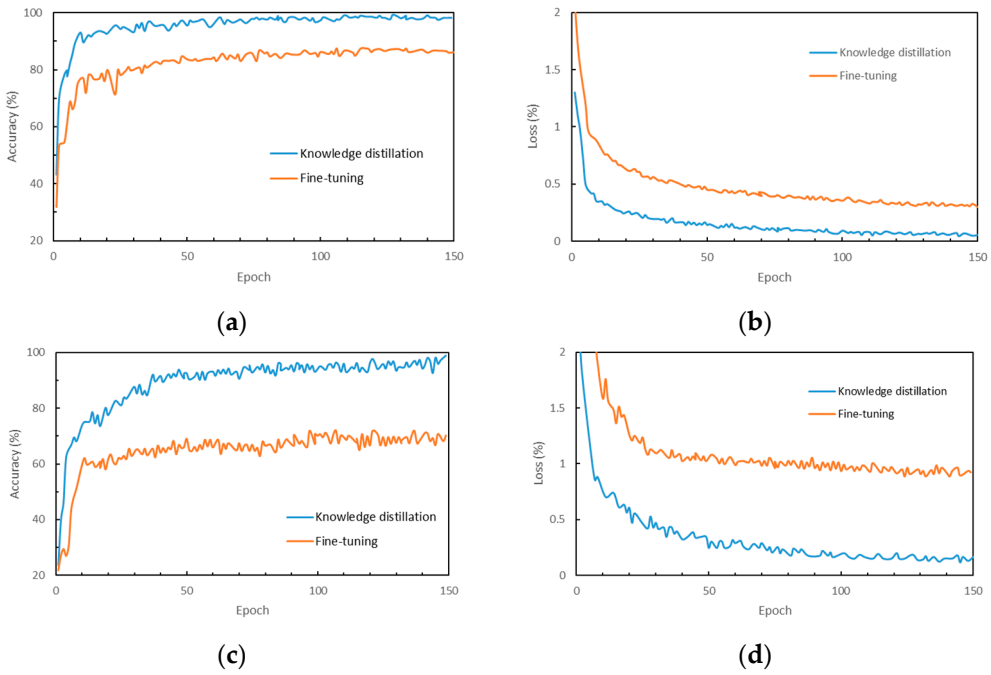


Figure 7. The accuracy and loss curves with different retraining methods on the XJTU-SY dataset: (**a**) accuracy curve of CNN-1; (**b**) loss curve of CNN-1; (**c**) accuracy curve of CNN-2; (**d**) loss curve of CNN-2.

It can be demonstrated through Figures 6 and 7 that knowledge distillation can significantly enhance the performance of the model after compression. This is attributed to the introduction of a soft label loss into the compressed model's loss function during the knowledge distillation procedure. It differs from the conventional training process for model fine-tuning. The soft label loss is obtained from the output of the compressed model (student model) and the soft labels, which incorporate more knowledge from the uncompressed original model (teacher models). Therefore, the compressed model using the knowledge distillation process can achieve superior accuracy.

Figure 8 illustrates the classification confusion matrix of the CNN-1 and CNN-2 before and after compression, respectively. The compression ratio of both models is set to 60%. The y-labels are predicted labels, x-labels are true labels, and the diagonal values are the overlap between them. The darker blue hue indicates the higher value. From Figure 8, it can be concluded that there are some fluctuations in the diagnostic performance of the model during the compression process. The accuracy of the compressed model without knowledge distillation has a significant decrease, whereas with knowledge distillation, the compressed model can basically reach the level before compression. This indicates that the proposed method for the fault diagnosis neural network can provide excellent performance, achieving a good balance between model compression and diagnostic accuracy.

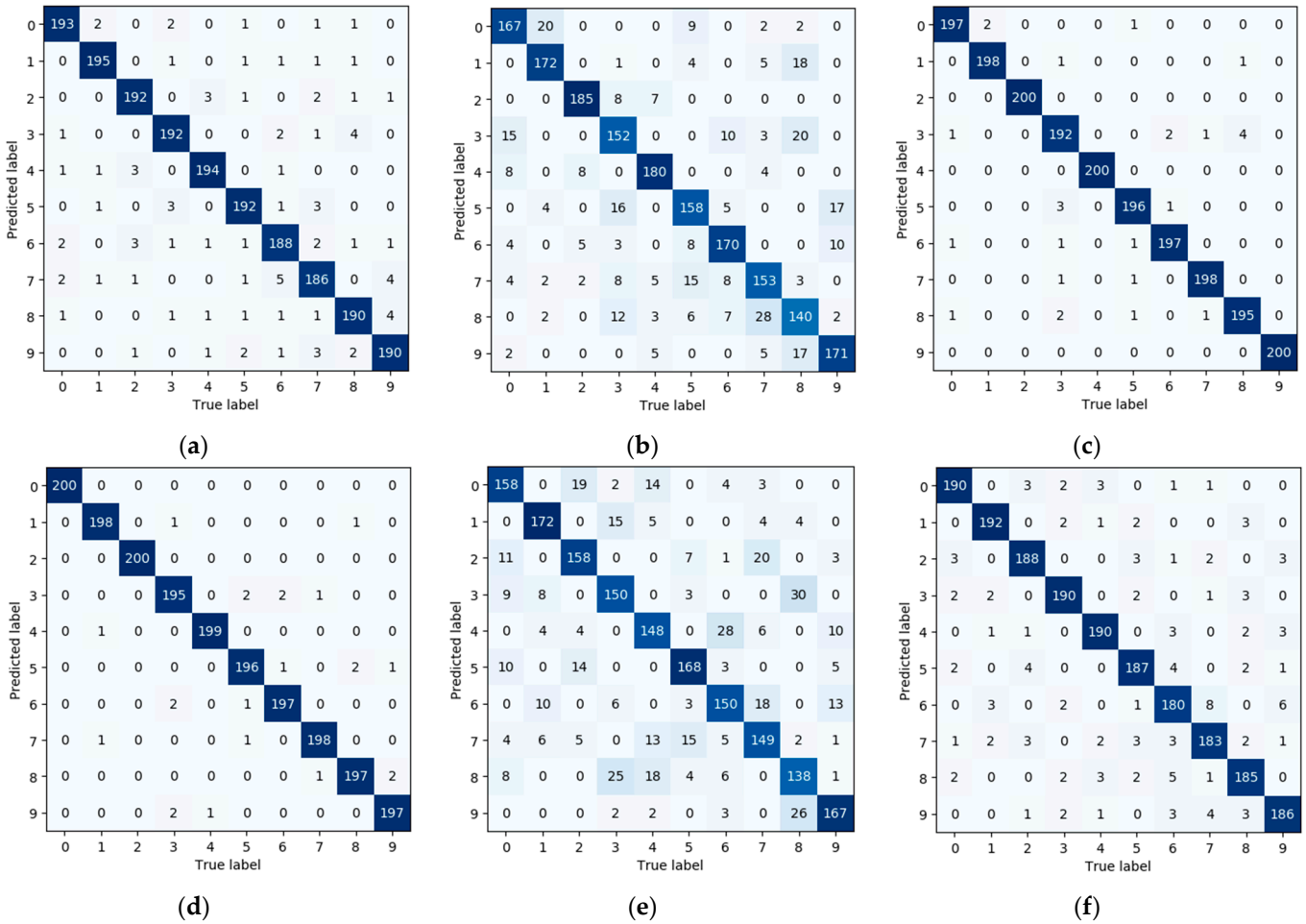


Figure 8. The confusion matrices of CNN-1 and CNN-2 on the XJTU-SY dataset: (**a**) confusion matrices of original CNN-1 before compression; (**b**) confusion matrices of compressed CNN-1 before knowledge distillation; (**c**) confusion matrices of compressed CNN-1 after knowledge distillation; (**d**) confusion matrices of original CNN-2 before compression; (**e**) confusion matrices of compressed CNN-2 before knowledge distillation; (**f**) confusion matrices of compressed CNN-2 after knowledge distillation.

The second ablation study tested the effects of tensor train decomposition and parameter quantization separately. Figure 9 shows the change in parameters and accuracy in different compression stages for the CNN-1 model on the XJTU-SY dataset. Different compression stages were applied sequentially to the CNN-1 model, and parameters and accuracy were recorded. As can be seen in Figure 9, the tensor train decomposition and parameter quantification methods can effectively reduce the number of parameters in the CNN-1 model. At the same time, they also adversely affect model accuracy. Specifically, the model has the most significant parameter reduction (over 40%) in the tensor train decomposition stage, which also has a significant impact on accuracy, with an accuracy loss of over 30%. On the other hand, the compression effect of parameter quantization is not as substantial as tensor decomposition. So, it is reasonable to use both together and use parameter quantization after tensor decomposition. In addition, the knowledge distillation method can successfully restore the accuracy of the compressed model in three different situations, which also proves the effectiveness of this method. In summary, the proposed multiple compression technique achieves high accuracy while significantly reducing the parameters.

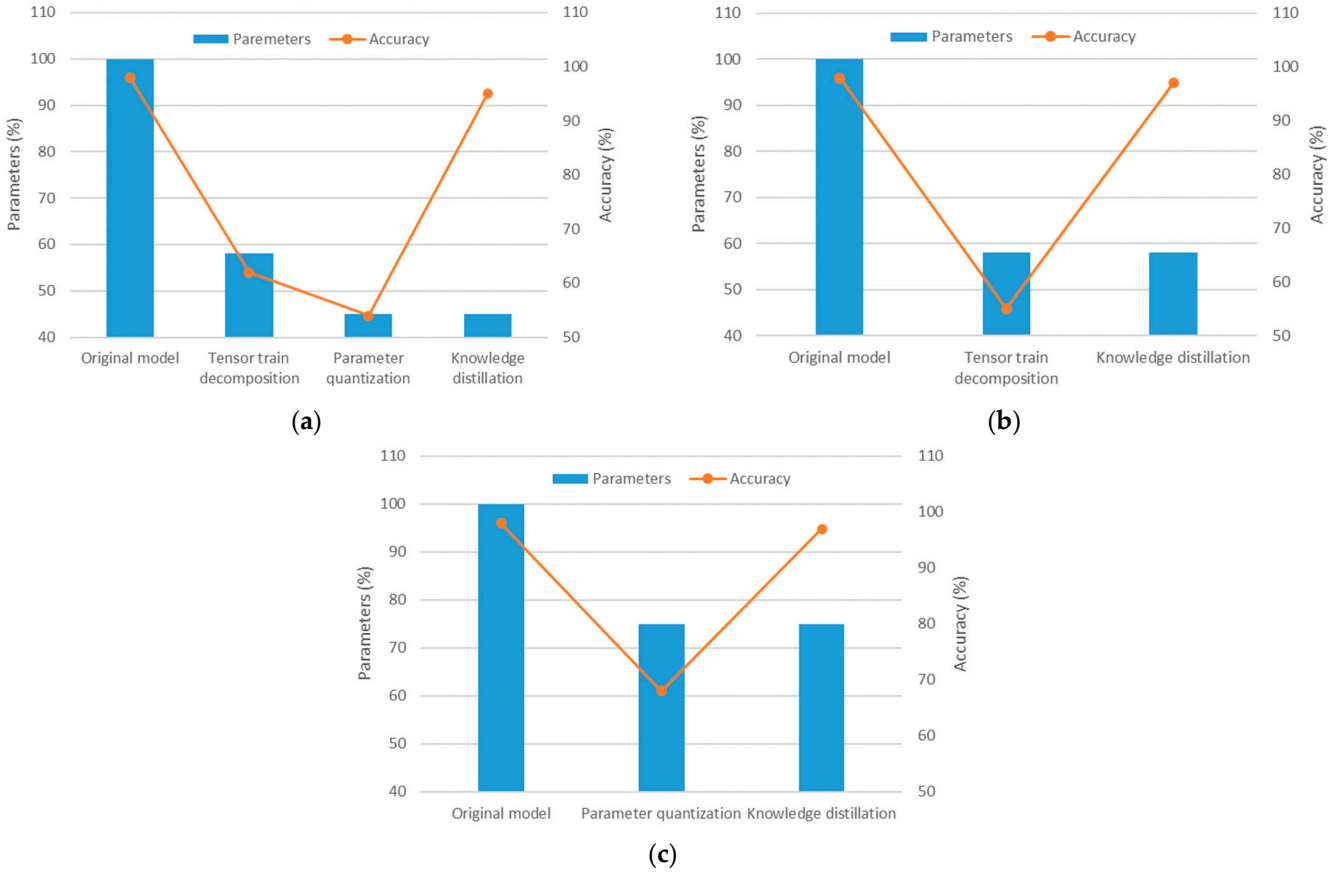


Figure 9. The accuracy curve and parameters in different compression stages of the CNN-1 model on the XJTU-SY dataset: (**a**) The whole compression process. (**b**) The compression process without parameter quantization. (**c**) The compression process without tensor train decomposition.

5. Conclusions

Bearing fault diagnosis based on CNNs has shown excellent performance. Unfortunately, the approach often suffers from drawbacks such as the excessive number of redundant parameters and computational complexity, which makes it challenging to implement in industrial devices with limited resources. To address this problem, model compression techniques provide an efficient remedy for this issue and generate outstanding results. In this study, a deep hybrid compression method is introduced to achieve neural network compression by combining tensor train decomposition, quantization, and

knowledge distillation. Based on the experimental results, the number of parameters of CNN-1 and CNN-2 is reduced by 87.5% and 87.8%, and the FLOPs are reduced by 87.9% and 89.5%, respectively. The approach introduced in this paper can effectively decrease the number of parameters and FLOPs substantially without compromising the accuracy of diagnostic models across various depths. Through ablation studies, it can be found that tensor train decomposition and parameter quantization can compress the size of diagnostic neural networks by more than 50% with 45% accuracy loss, and knowledge distillation can restore the accuracy lost from compression models. In addition, compared with existing CNN-based compression approaches and conventional machine learning techniques, this approach maintains greater diagnostic accuracy at reduced storage capacity. This can promote the application of intelligent bearing fault diagnosis based on CNNs in practical industrial equipment. Our future work will implement the practical deployment of neural networks for bearing fault diagnosis on embedded platforms, such as the FPGA and smart chips.

Author Contributions: Conceptualization, X.W., C.Y., and H.L.; methodology, X.W. and C.Y.; investigation, D.H. and C.Y.; resources, X.W. and C.Y.; writing—original draft preparation, D.H. and C.Y.; writing—review and editing, X.F. and C.Y.; supervision, X.F. and H.L.; project administration, D.H. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities, grant number B230205019, and the National Key Research and Development Program of China, grant number 2019YFE0105200.

Data Availability Statement: Datas are contained within the article.

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

References

1. He, Y.; Lin, J.; Liu, Z.; Wang, H.; Li, L.-J.; Han, S. AMC: AutoML for Model Compression and Acceleration on Mobile Devices. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 784–800.
2. He, Y.; Kang, G.; Dong, X.; Fu, Y.; Yang, Y. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; AAAI Press: Stockholm, Sweden, 2018; pp. 2234–2240.
3. Gong, R.; Liu, X.; Jiang, S.; Li, T.; Hu, P.; Lin, J.; Yu, F.; Yan, J. Differentiable Soft Quantization: Bridging Full-Precision and Low-Bit Neural Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4852–4861.
4. Dong, Z.; Yao, Z.; Arfeen, D.; Gholami, A.; Mahoney, M.W.; Keutzer, K. HAWQ-V2: Hessian Aware Trace-Weighted Quantization of Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 18518–18529.
5. Kossaifi, J.; Toisoul, A.; Bulat, A.; Panagakis, Y.; Hospedales, T.M.; Pantic, M. Factorized Higher-Order CNNs with an Application to Spatio-Temporal Emotion Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6060–6069.
6. Kim, Y.-D.; Park, E.; Yoo, S.; Choi, T.; Yang, L.; Shin, D. Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications. *arXiv* **2016**, arXiv:1511.06530.
7. Li, X.; Li, S.; Omar, B.; Wu, F.; Li, X. ResKD: Residual-Guided Knowledge Distillation. *IEEE Trans. Image Process.* **2021**, *30*, 4735–4746. [CrossRef] [PubMed]
8. Ji, M.; Peng, G.; Li, S.; Cheng, F.; Chen, Z.; Li, Z.; Du, H. A Neural Network Compression Method Based on Knowledge-Distillation and Parameter Quantization for the Bearing Fault Diagnosis. *Appl. Soft Comput.* **2022**, *127*, 109331. [CrossRef]
9. Yang, Y.; Krompass, D.; Tresp, V. Tensor-Train Recurrent Neural Networks for Video Classification. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Precup, D., Teh, Y.W., Eds.; PMLR: Breckenridge, CO, USA, 2017; Volume 70, pp. 3891–3900.
10. Huang, W.; Sun, H.; Liu, Y.; Wang, W. Feature Extraction for Rolling Element Bearing Faults Using Resonance Sparse Signal Decomposition. *Exp. Tech.* **2017**, *41*, 251–265. [CrossRef]
11. Tang, S.; Yuan, S.; Zhu, Y. Convolutional Neural Network in Intelligent Fault Diagnosis Toward Rotatory Machinery. *IEEE Access* **2020**, *8*, 86510–86519. [CrossRef]
12. Neupane, D.; Seok, J. Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches: A Review. *IEEE Access* **2020**, *8*, 93155–93178. [CrossRef]

13. Duhamel, P.; Vetterli, M. Fast Fourier Transforms: A Tutorial Review and a State of the Art. *Signal Process.* **1990**, *19*, 259–299. [CrossRef]
14. McDonnell, J.T.E.; Bentley, P.M. Wavelet Transforms: An Introduction. *Electron. Commun. Eng. J.* **1994**, *6*, 175–186. [CrossRef]
15. Zeiler, A.; Faltermeier, R.; Keck, I.R.; Tomé, A.M.; Puntonet, C.G.; Lang, E.W. Empirical Mode Decomposition—An Introduction. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–8.
16. Xu, L.; Chatterton, S.; Pennacchi, P.; Liu, C. A Tacholess Order Tracking Method Based on Inverse Short Time Fourier Transform and Singular Value Decomposition for Bearing Fault Diagnosis. *Sensors* **2020**, *20*, 6924. [CrossRef]
17. Prabhakar, S.; Mohanty, A.R.; Sekhar, A.S. Application of Discrete Wavelet Transform for Detection of Ball Bearing Race Faults. *Tribol. Int.* **2002**, *35*, 793–800. [CrossRef]
18. Fan, H.; Shao, S.; Zhang, X.; Wan, X.; Cao, X.; Ma, H. Intelligent Fault Diagnosis of Rolling Bearing Using FCM Clustering of EMD-PWVD Vibration Images. *IEEE Access* **2020**, *8*, 145194–145206. [CrossRef]
19. Gu, J.; Peng, Y. An Improved Complementary Ensemble Empirical Mode Decomposition Method and Its Application in Rolling Bearing Fault Diagnosis. *Digit. Signal Process.* **2021**, *113*, 103050. [CrossRef]
20. Rai, V.K.; Mohanty, A.R. Bearing Fault Diagnosis Using FFT of Intrinsic Mode Functions in Hilbert–Huang Transform. *Mech. Syst. Signal Process.* **2007**, *21*, 2607–2615. [CrossRef]
21. Yang, J.; Yang, C.; Zhuang, X.; Liu, H.; Wang, Z. Unknown Bearing Fault Diagnosis under Time-Varying Speed Conditions and Strong Noise Background. *Nonlinear Dyn.* **2022**, *107*, 2177–2193. [CrossRef]
22. AlShorman, O.; Irfan, M.; Saad, N.; Zhen, D.; Haider, N.; Glowacz, A.; AlShorman, A. A Review of Artificial Intelligence Methods for Condition Monitoring and Fault Diagnosis of Rolling Element Bearings for Induction Motor. *Shock Vib.* **2020**, *2020*, 8843759. [CrossRef]
23. Xie, J.; Du, G.; Shen, C.; Chen, N.; Chen, L.; Zhu, Z. An End-to-End Model Based on Improved Adaptive Deep Belief Network and Its Application to Bearing Fault Diagnosis. *IEEE Access* **2018**, *6*, 63584–63596. [CrossRef]
24. Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N. Deep Neural Networks: A Promising Tool for Fault Characteristic Mining and Intelligent Diagnosis of Rotating Machinery with Massive Data. *Mech. Syst. Signal Process.* **2016**, *72–73*, 303–315. [CrossRef]
25. Liang, H.; Zhao, X. Rolling Bearing Fault Diagnosis Based on One-Dimensional Dilated Convolution Network With Residual Connection. *IEEE Access* **2021**, *9*, 31078–31091. [CrossRef]
26. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A Deep Convolutional Neural Network with New Training Methods for Bearing Fault Diagnosis under Noisy Environment and Different Working Load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [CrossRef]
27. Che, C.; Wang, H.; Ni, X.; Fu, Q. Domain Adaptive Deep Belief Network for Rolling Bearing Fault Diagnosis. *Comput. Ind. Eng.* **2020**, *143*, 106427. [CrossRef]
28. Chen, G. Timed Failure Propagation Graph Construction with Supremal Language Guided Tree-LSTM and Its Application to Interpretable Fault Diagnosis. *Appl. Intell.* **2022**, *52*, 12990–13005. [CrossRef]
29. Wang, R.; Jiang, H.; Li, X.; Liu, S. A Reinforcement Neural Architecture Search Method for Rolling Bearing Fault Diagnosis. *Measurement* **2020**, *154*, 107417. [CrossRef]
30. Zhang, S.; Zhang, S.; Wang, B.; Habetler, T.G. Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review. *IEEE Access* **2020**, *8*, 29857–29881. [CrossRef]
31. Zhu, H.; Cheng, J.; Zhang, C.; Wu, J.; Shao, X. Stacked Pruning Sparse Denoising Autoencoder Based Intelligent Fault Diagnosis of Rolling Bearings. *Appl. Soft Comput.* **2020**, *88*, 106060. [CrossRef]
32. Ding, A.; Qin, Y.; Wang, B.; Jia, L.; Cheng, X. Lightweight Multiscale Convolutional Networks With Adaptive Pruning for Intelligent Fault Diagnosis of Train Bogie Bearings in Edge Computing Scenarios. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 3502813. [CrossRef]
33. Yao, D.; Liu, H.; Yang, J.; Li, X. A Lightweight Neural Network with Strong Robustness for Bearing Fault Diagnosis. *Measurement* **2020**, *159*, 107756. [CrossRef]
34. Deng, J.; Jiang, W.; Zhang, Y.; Wang, G.; Li, S.; Fang, H. HS-KDNet: A Lightweight Network Based on Hierarchical-Split Block and Knowledge Distillation for Fault Diagnosis With Extremely Imbalanced Data. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3521109. [CrossRef]
35. Fu, L.; Yan, K.; Zhang, Y.; Chen, R.; Ma, Z.; Xu, F.; Zhu, T. EdgeCog: A Real-Time Bearing Fault Diagnosis System Based on Lightweight Edge Computing. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2521711. [CrossRef]
36. Li, X.; Hu, Y.; Zheng, J.; Li, M. Neural Architecture Search For Fault Diagnosis. *arXiv* **2020**, arXiv:2002.07997.
37. Zhang, K.; Chen, J.; He, S.; Xu, E.; Li, F.; Zhou, Z. Differentiable Neural Architecture Search Augmented with Pruning and Multi-Objective Optimization for Time-Efficient Intelligent Fault Diagnosis of Machinery. *Mech. Syst. Signal Process.* **2021**, *158*, 107773. [CrossRef]
38. Garipov, T.; Podoprikhin, D.; Novikov, A.; Vetrov, D. Ultimate Tensorization: Compressing Convolutional and Fc Layers Alike. *arXiv* **2016**, arXiv:1611.03214.
39. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
40. Wang, B.; Lei, Y.; Li, N.; Li, N. A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings. *IEEE Trans. Reliab.* **2020**, *69*, 401–412. [CrossRef]
41. Smith, W.A.; Randall, R.B. Rolling Element Bearing Diagnostics Using the Case Western Reserve University Data: A Benchmark Study. *Mech. Syst. Signal Process.* **2015**, *64–65*, 100–131. [CrossRef]

42. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5990–5998. [CrossRef]
43. Zhao, J.; Yang, S.; Li, Q.; Liu, Y.; Gu, X.; Liu, W. A New Bearing Fault Diagnosis Method Based on Signal-to-Image Mapping and Convolutional Neural Network. *Measurement* **2021**, *176*, 109088. [CrossRef]
44. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning Filters for Efficient ConvNets. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017; pp. 1–13.
45. Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; Shao, L. HRank: Filter Pruning Using High-Rank Feature Map. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE Computer Society: Los Alamitos, CA, USA, 2020; pp. 1529–1538.
46. Phan, A.-H.; Sobolev, K.; Sozykin, K.; Ermilov, D.; Gusak, J.; Tichavský, P.; Glukhov, V.; Oseledets, I.; Cichocki, A. Stable Low-Rank Tensor Decomposition for Compression of Convolutional Neural Network. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 522–539.
47. Cheng, J.; Wu, J.; Leng, C.; Wang, Y.; Hu, Q. Quantized CNN: A Unified Approach to Accelerate and Compress Convolutional Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 4730–4743. [CrossRef]
48. Li, X.; Yang, Y.; Pan, H.; Cheng, J.; Cheng, J. A Novel Deep Stacking Least Squares Support Vector Machine for Rolling Bearing Fault Diagnosis. *Comput. Ind.* **2019**, *110*, 36–47. [CrossRef]
49. Qian, W.; Li, S.; Lu, J. Adaptive Nearest Neighbor Reconstruction with Deep Contractive Sparse Filtering for Fault Diagnosis of Roller Bearings. *Eng. Appl. Artif. Intell.* **2022**, *111*, 104749. [CrossRef]
50. Yu, J.; Yan, X. A New Deep Model Based on the Stacked Autoencoder with Intensified Iterative Learning Style for Industrial Fault Detection. *Process Saf. Environ. Prot.* **2021**, *153*, 47–59. [CrossRef]