sensors

MDPI

Article

Method for Diagnosing Bearing Faults in Electromechanical Equipment Based on Improved Prototypical Networks

Zilong Wang [1,2,3], Honghai Shen [1,2,3,*], Wenzhuo Xiong [2,3,*], Xueming Zhang [2,3] and Jinghua Hou [4]

[1] Key Laboratory of Airborne Optical Imaging and Measurement, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[3] University of Chinese Academy of Sciences, Beijing 100039, China
[4] Jiuquan Satellite Launch Centre, Jiuquan 732750, China
* Correspondence: shenhh@ciomp.ac.cn (H.S.); xiongwenzhuo@ciomp.ac.cn (W.X.)

Abstract: Due to the complexity of electromechanical equipment and the difficulties in obtaining large-scale health monitoring datasets, as well as the long-tailed distribution of data, existing methods ignore certain characteristics of health monitoring data. In order to solve these problems, this paper proposes a method for the fault diagnosis of rolling bearings in electromechanical equipment based on an improved prototypical network—the weight prototypical networks (WPorNet). The main contributions of this paper are as follows: (1) the prototypical networks, which perform well on small-sample classification tasks, were improved by calculating the different levels of influence of support sample distributions in order to achieve the prototypical calculation. The change in sample influence was calculated using the Kullback–Leibler divergence of the sample distribution. The influence change in a specific sample can be measured by assessing how much the distribution changes in the absence of that sample; and (2) The Gramian Angular Field (GAF) algorithm was used to transform one-dimensional time series into two-dimensional vibration images, which greatly improved the application effect of the 2D convolutional neural network (CNN). Through experiments on MAFAULDA and CWRU bearing datasets, it was shown that this network effectively solves the shortcomings of a small number of valid samples and a long-tail distribution in health monitoring data, it enhances the dependency between the samples and the global data, it improves the model's feature extraction ability, and it enhances the accuracy of model classification. Compared with the prototypical network, the improved network model increased the performance of the 2-way 10-shot, 2-way 20-shot, and 2-way 50-shot classification tasks by 5.23%, 5.74%, and 4.37%, respectively, and it increased the performance of the 4-way 10-shot, 4-way 20-shot, and 4-way 50-shot classification tasks by 12.02%, 10.47%, and 4.66%, respectively. Experimental results show that the improved prototypical network model has higher sample classification accuracy and stronger anti-interference ability compared with traditional small-sample classification models.

Keywords: fault diagnosis; meta-learning; Kullback–Leibler divergence; Gramian angular field

check for updates

Citation: Wang, Z.; Shen, H.; Xiong, W.; Zhang, X.; Hou, J. Method for Diagnosing Bearing Faults in Electromechanical Equipment Based on Improved Prototypical Networks. *Sensors* **2023**, *23*, 4485. https://doi.org/10.3390/s23094485

Academic Editor: Ruqiang Yan

Received: 14 March 2023
Revised: 20 April 2023
Accepted: 25 April 2023
Published: 4 May 2023

1. Introduction

Currently, electromechanical equipment plays a crucial role in various industries, from manufacturing to healthcare. It is essential to ensure their regular operation and lifespan to ensure operational continuity and to avoid costly inoperative periods [1]. Prognostics health management (PHM) [2,3] and Remaining Useful Life (RUL) [4] are processes that involve collecting and analyzing data from equipment to detect potential faults and take preventive measures. The PHM process typically involves data collection, processing and preparation, feature selection, and model construction. Data can be collected from various sources, including sensors, telemetry systems, and manual inspection. Researchers can use machine learning algorithms such as regression, decision trees, and neural networks to analyze the data and build models for predicting faults. PHM research in industry is extensive, and this article mainly focuses on the application of PHM in diagnosing failures of rotating bearings in electromechanical equipment. Accurately locating the cause of failures in electromechanical equipment is often difficult, as although the forms of failure may appear simple, the underlying causes are relatively complex. According to the statistics, up to 45% of motor failures are caused by rolling bearings, thus making rolling bearing failures one of the main failures of rotating motors [5].

Traditional PHM research methods are based on physical and mathematical models which require complex modeling, a large number of parameters, and a wealth of expert experience. These methods are primarily used in systems with simple and clear physical models. It is challenging to implement these methods given that modern equipment operates under relatively complex conditions, and thus, they need better portability. For example, Kirubarajan [6] et al. predicted the RUL of the shaft using Forman's crack growth law; Shen et al. [7] proposed a virtual simulation-based prediction method for the typical failure mode of gear pumps; and Chen [8] et al. established a dynamic system model using two different depths of tooth crack in a sun gear and internal gear, extracting faults via fast spectral kurtosis (FSK).

In data-driven PHM applications, traditional signal processing methods use shallow models, and it is not easy to extract the implicit relationships between different data features in the input dataset; therefore, the diagnostic effect could be better. Yin et al. [9] used SVM for fault diagnosis, Zhao [10] et al. used the Hidden Markov Model (HMM) to diagnose equipment, and Zhang [11] et al. used the Back Propagation Neural Network (BPNN) for equipment fault diagnosis. Deep learning [12–14] is a learning method that establishes multi-level representations, converts data features into more abstract modules, learns layer by layer by combining simple, but non-linear, modules, and it shows representations of each layer [15,16]. High-dimensional non-linear features can be transformed into low-dimensional features by using enough hidden layers, thus effectively capturing the hidden information in the data and achieving very complex learning functions [17]. Eren et al. [18] used a convolutional neural network (CNN) classifier for bearing fault diagnoses; Zhang et al. [19] proposed a fault diagnosis method that combines time–frequency feature oversampling (TFFO) with CNN; Yan [20] proposed a fault diagnosis model, MTF-ResNet, based on the Markov transition field and deep residual network; Liu et al. [21] proposed a way of using RNN as an autoencoder for bearing fault diagnosis; Gao et al. [22] optimized fault feature extraction by improving the WDCNN network; and Chen et al. [23] proposed a novel Wide Residual Relation Network (WRRN) for RM intelligent fault diagnoses.

Due to the long-term operation of mechanical equipment under complex working conditions, most of the data collected by the monitoring system is repetitive, and there are few samples consisting of effective and marked fault characteristic information, thus making it difficult to provide a large amount of sample training data for neural networks to train; this causes great difficulties during the application of neural network models. In recent years, the use of small-sample data classification methods when solving the problem of mechanical and electrical equipment fault diagnosis has attracted the attention of many researchers. Xiao et al. [24] used Domain Adaptive (DA) technology and deep transfer learning for fault diagnoses, thus improving performance through knowledge transfer. Chen et al. [25] proposed an adaptive CNN network diagnostic model. Yue et al. [26] extracted rich features using a multi-scale wavelet convolution module and then performed fault diagnosis using a meta-learning module. Li et al. [27] improved the MAML algorithm for a small sample bearing a fault diagnosis. Based on meta-learning, small-sample classification methods have gradually gained the attention of researchers due to their efficient and accurate advantages.

Prototypical network [28] is a meta-learning-based method for small-sample learning; it has performed best when dealing with small-sample classification problems among various models. It uses a support set, S, to extract prototypical vectors for each category, and it classifies query vectors inputted into the query set based on their distance from the prototypical vectors. However, due to the diverse types of faults in mechanical and electrical equipment, such as complex operating conditions, strong time-varying characteristics, and strong long-tail effects in data, the prototypical network structure is prone to ignoring the capturing of feature data, thus making it difficult to quickly capture signal features. This seriously affects its application effectiveness in mechanical and electrical equipment fault diagnosis [29,30].

In this paper, an improved prototypical network model, WPortNet, is proposed and compared with other machine learning models using the same rolling bearing dataset in a comparative experiment, yielding satisfactory results. Firstly, the model uses the Gramian Angular Field (GAF) algorithm to transform one-dimensional time series into two-dimensional vibration images, thus effectively improving the feature extraction effect of the 2D CNN network. Then, classification is performed by calculating the differences between the levels of influence of the supporting sample distribution, and the influence of specific samples can be calculated using the KL divergence in the absence of the sample. The network effectively solves the shortcomings of a small number of effective samples and long-tail distribution in health monitoring data, it enhances the dependence relationship between samples and global data, improves the model's feature extraction ability, and enhances the accuracy of model classification. Through experiments, it has been demonstrated that the improved prototypical network model has a higher sample classification accuracy and a high level of robustness. The main contributions of this paper are summarized as follows:

(1) The prototypical network, which performs well on small-sample classification tasks, was improved by calculating the differences between the influence of the support sample distributions in order to achieve the prototypical calculation. The change in sample influence was calculated using the Kullback–Leibler divergence of the sample distribution. The influence change of a specific sample can be measured by assessing how much the distribution changes in the absence of that sample.

(2) The Gramian Angular Field algorithm was used to transform a one-dimensional time series into two-dimensional vibration images, thus greatly improving the application effect of the 2D convolutional neural network.

The rest of the paper is organized as follows: preliminary knowledge is introduced in section two; in section three, the structural design of the proposed model is introduced; some experiments are carried out, and the experimental results are analyzed to evaluate our method against other methods in section four; section five presents the results of the experiments; and finally, we draw conclusions in section six.

2. Preliminary Knowledge

Machine learning and prototypical networks have been widely and successfully applied to image classification tasks; however, due to the complex operating conditions, the various types of faults in electromechanical equipment, substantial time variability, and complex frequency components of vibration signals, the prototypical network structure is prone to ignoring the internal correlation of the spectrum. It is severely affected by a large amount of noise, which seriously interferes with the fault characteristics of the signal. In this section, we propose a method using KL divergence to create a new prototypical network, WPorNet, which utilizes the ability of the prototypical network, in terms of image classification, for fault diagnoses of electromechanical equipment.

2.1. Meta Learning and Prototypical Networks

Meta Learning, also known as learning to learn, is a crucial approach in machine learning. Its goal is to adapt to new, unseen tasks by learning general learning strategies from a large number of tasks.

Meta learning can be divided into two categories: model-based meta learning and data-based meta learning. Model-based meta learning refers to the learning algorithm, which is a learnable model that can be trained to produce different models for different learning tasks. Common model-based meta-learning algorithms include Model-Agnostic Meta-Learning (MAML) [31] and Reptile [32]. Data-based meta-learning refers to model learning from the training data of multiple tasks so that connections can be found between them and so that it is possible to quickly adapt to new ones. Representative algorithms include prototypical networks, matching networks [33], and relation networks [34].

Prototypical networks are a type of data-driven meta-learning method that achieves data classification tasks by learning prototypical networks from the sample data, and it has the advantage of being simple and efficient when handling small-sample classification problems. It assumes that the samples of each category are clustered around a featured center, and it maps the input onto an embedding vector via a neural network. By calculating the average embedding vector of all support instances in the support set category, the obtained vector is used as the prototypical center of the category, which then realizes the feature extraction of the category; this center is also a parameter of network learning. Then, classification is achieved by measuring the distance between the sample and the center.

A prototypical network divides the training dataset into two parts: the support set and the query set. The support set contains a dataset with K categories and a total of N labeled samples $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$, where $x_i \in R^D$ is a D-dimensional feature vector and $y_i \in \{1, 2, \cdots, K\}$ is a sample label. The query set contains H labeled data. First, the data in each category in the support set is inputted into the feature extractor of the prototypical network to learn the nonlinear mapping relationship between the support set and the metric space, and to obtain the M-dimensional feature embedding of each sample in the metric space. S_k is a sample set from category k. Then, the feature representation of each category's health status in the metric space is obtained using the average value of each class's sample feature embedding, as follows:

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i) \tag{1}$$

where $f_\phi(x_i)$ is the feature embedding function of vector x_i, which represents the feature of vector x_i, then, the data of query set Q is inputted to obtain the M-dimensional feature embedding of $c \in R^M$. The Euclidean distance d_k between c and K prototypicals is calculated and then transformed into the probability of each class using Softmax, represented as:

$$\mathrm{p}_\phi(y = k|\hat{x}) = \frac{\exp(-d(f_\phi(\hat{x}), c_k))}{\sum_k \exp(-d(f_\phi(\hat{x}), c_k))} \tag{2}$$

2.2. Gramian Angular Field Transformation

Gramian Angular Field (GAF) can be represented by a Gramian matrix. By using matrix values as pixels in an image, time series can be encoded as images. Gramian Angular Field Transformation is shown in Figure 1.

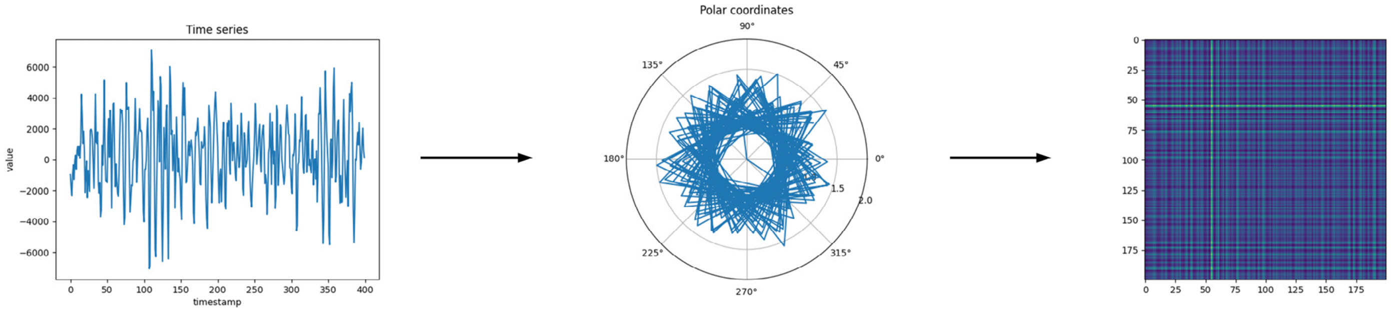


Figure 1. Gramian Angular Field Transformation.

Wang [35] proposed a method for transforming one-dimensional time series into two-dimensional images using the GAF algorithm. When given a time series, $A = \{a_1, a_2, \cdots, a_n\}$ with n observations, first, we normalized A so that all values in time series A are within the range of $[-1, 1]$ or $[0, 1]$:

$$\widetilde{a}_{-1} = \frac{(a_i - \max(A) + (a_i - \min(A)))}{\max(A) - \min(A)} \tag{3}$$

$$\text{or } \widetilde{a}_0 = \frac{a_i - \min(A)}{\max(A) - \min(A)} \tag{4}$$

Next, the time series was transformed into polar coordinates:

$$\phi_i = \arccos(\widetilde{a}_i), -1 \leq \widetilde{a}_i \leq 1, \widetilde{a}_i \in \widetilde{A}r_i = \frac{t_i}{N}, t_i \in N \tag{5}$$

where t_i is the timestamp and N is a constant factor that is used to adjust the range of the polar coordinate system. The angle value ϕ_i in the polar coordinate system is the inverse cosine of the time series $\widetilde{a}_i$, and the radius in the polar coordinate system is the ratio of the timestamp to the constant factor. This achieves a transformation from the amplitude variation of the time series to the angle variation in the polar coordinate image. Then, the Gramian Field is generated by defining the sum of the trigonometric functions of each point in the interval:

$$GAF = \begin{pmatrix} \cos(\phi_1 + \phi_1) & \ldots & \cos(\phi_1 + \phi_N) \\ \vdots & \ddots & \vdots \\ \cos(\phi_N + \phi_1) & \cdots & \cos(\phi_N + \phi_N) \end{pmatrix} \tag{6}$$

It is a bijection, and when $\phi \in [0, \pi]$, $\cos(\phi)$ is monotonically increasing. The value of GAF not only depends on the time stamp interval but also on the absolute position of the time series.

3. Fault Diagnosis Method Based on WproNet

We calculated the distributional influence of the support-set samples and incorporated it into the network for reevaluation. The differences between the levels of distributional influence among samples can be measured by how much the distribution changes when the sample is removed, unlike in the prototypical network, where all samples are treated equally. This enhances the dependence between the samples and the global data, thus effectively addressing the long-tail distribution characteristics of health monitoring data and significantly improving the accuracy of model classification.

3.1. Improved Model Architecture

The model is mainly composed of a preprocessing layer, encoder layer, and distribution-prototypical layer. The raw signal is transformed by GAF to obtain the vibration image in the preprocessing layer. Then, for each category, the vibration image is divided into support and query sets. In the encoder layer, a four-layer CNN embedding function f_ϕ is used to extract features and generate a prototypical representation for each category. After the features extracted from the two channels are passed through a convolutional block, the distributional weight of each support sample is calculated to obtain the new probability of the category. Next, the distribution–prototypical layer maps onto the feature space to calculate the distributional distance between the sample and the prototypical layer, and the probability of the sample in each category is calculated using Softmax. The model's structure is shown in Figure 2.

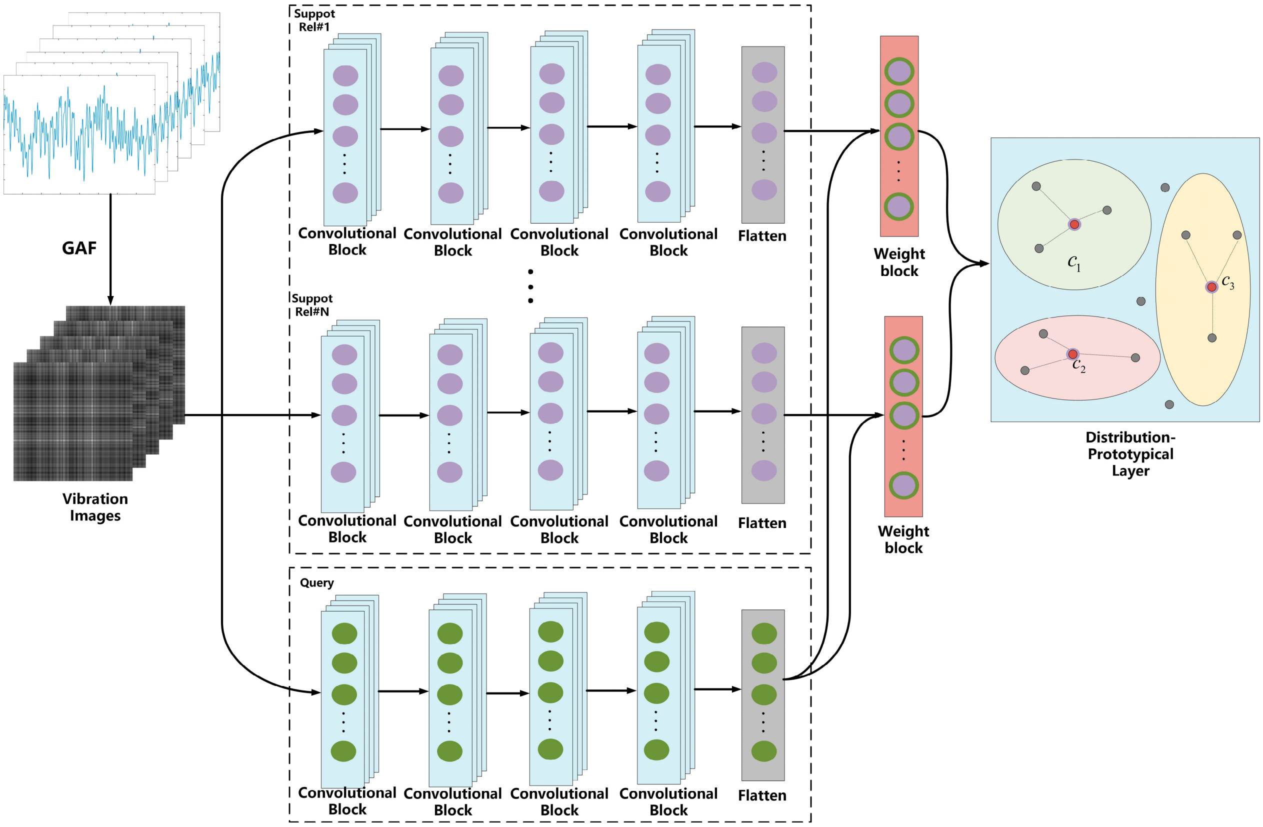


Figure 2. Structure of the Improved Model.

3.2. Encoder Layer

WproNet uses a four-layer convolutional neural network (CNN) as its encoder. The structure of the encoder is shown in Figure 3. Each convolutional block is composed of four identical encoder modules, each of which contains a convolutional layer (Conv2D), a batch normalization layer (BatchNorm2D), a linear rectification function layer (ReLU), and a maximum pooling layer (MaxPool2D). Finally, the Flatten layer is introduced to flatten the feature vectors of each sample into a one-dimensional vector input for the loss function. The network automatically learns the features of the signals from the grayscale vibration images to train the embedding function f_ϕ, and it then prepares for the next operation.

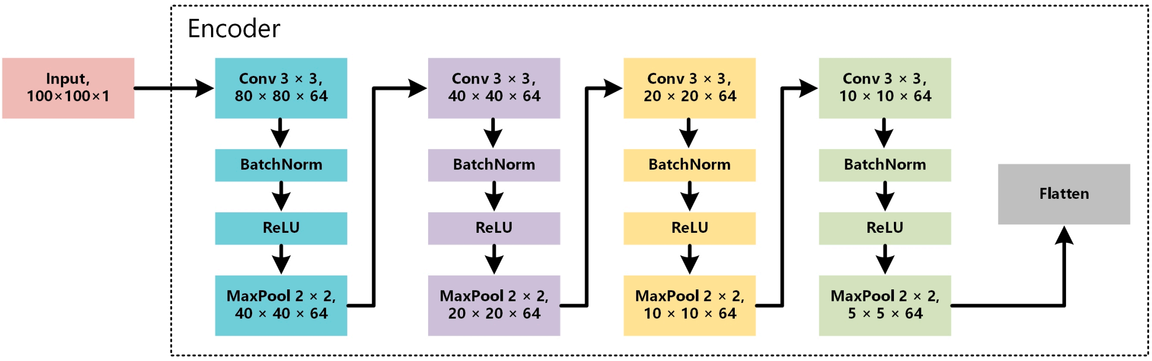


Figure 3. Structure of Encoder.

The convolutional layer extracts features by sliding a convolution kernel over an input vector with a certain window size. When performing convolution, the width of the convolution kernel should be the same as the dimension of the embedding vector output of the embedding layer. Here, a convolution kernel of size m is used for the convolution operation, and the hidden representation of each feature in the image is obtained via calculations.

$$h_i = CNN(s_{i-\frac{m-1}{2}}, \cdots, s_{i+\frac{m-1}{2}}) \tag{7}$$

where $CNN()$ represents the convolutional operation and the image after undergoing convolutional operation can be represented as:

$$x = (h_1, h_2 \cdot \cdot \cdot, h_n) \tag{8}$$

After the convolution operation, a ReLU activation function is added:

$$\mathrm{Re}LU = \max(0, a) \tag{9}$$

3.3. Distribution–Prototypical Layer

3.3.1. K–L Divergence

The Kullback–Leibler Divergence (KL divergence), also known as relative entropy, is generally used to measure the similarity or degree of difference between two probability distribution functions. Given the two distribution functions, $P(X)$ and $Q(X)$, their KL divergence is defined as follows:

$$D_{KL}[P(X)||Q(X)] = \sum_{x \in X}\left[P(x)\log\frac{P(x)}{Q(x)}\right] = E_{x\sim P(x)}\left[\log\frac{P(x)}{Q(x)}\right] \tag{10}$$

Thus, given the two datasets with distributions A and, respectively, their KL divergence is:

$$D_{KL}[A||B] = \sum_i\left[A\log\frac{A}{B}\right] \tag{11}$$

where ϕ represents the function that maps onto the latent space. When KL divergence is equal to 0, $A = B$ indicates that the distribution of A and B is highly consistent. Samples with a high KL divergence represent deviations from the distribution.

3.3.2. Distribution–Prototypical Layer Design

Unlike the prototypical network, after mapping the samples onto the feature space using the feature extractor, we used the K–L divergence to measure the consistency between the test samples and their corresponding dataset distribution. The weight of the sample can be measured by how much the distribution changes when the sample is not present in the dataset:

$$D_{KL}(x_i) = D_{KL(\phi)}[S||S - x_i] = \sum_i\left[S\log\frac{S}{S - x_i}\right] \tag{12}$$

We define the sample weight W by normalizing the K–L divergence of the sample, as follows:

$$W(f_\phi(x_i)) = D_{KL}(\widetilde{x}_i) \tag{13}$$

Therefore, the new feature representation of each category is:

$$c_k = \frac{\sum_{i=1}^{|S_k|} W(f_\phi(x_i)) f_\phi(x_i)}{\sum_{i=1}^{|S_k|} W(f_\phi(x_i))} \tag{14}$$

Finally, we calculated the loss function:

$$J(x, c_k) = -\log\frac{1}{q}\sum_{i=1}^{q}\frac{\exp(-d(f_\phi(\hat{x}), c_k))}{\sum_k \exp(-d(f_\phi(\hat{x}), c_k))} \tag{15}$$

where q denotes the number of query instances.

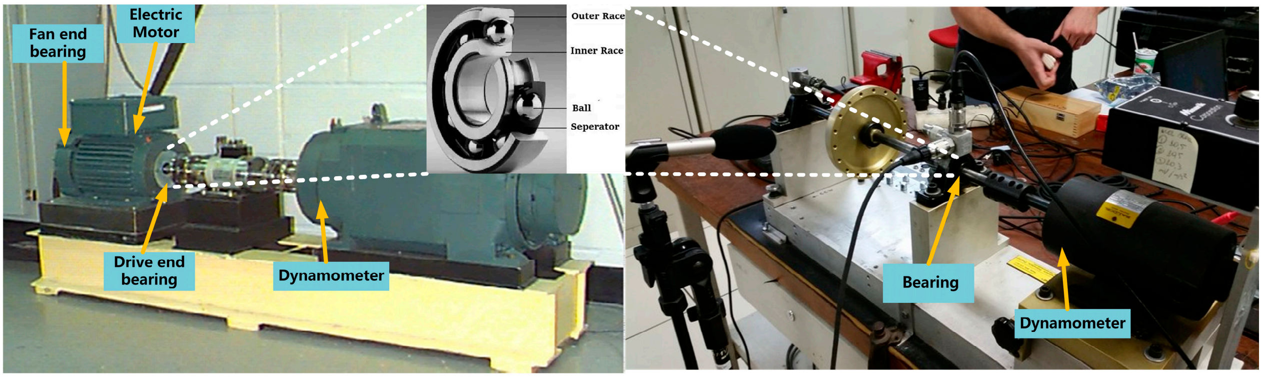
When a loss function is generated, the stochastic gradient descent (SGD) is used to train the network. Through multiple updates, optimal model parameters ϕ and prototypical networks of various health states are obtained. By calculating the sample weights, new probabilities for each class are obtained, and the model's prediction result is obtained.

3.4. Fault Diagnosis Process

The improved prototypical network is applied during the fault diagnosis process concerning mechanical and electrical equipment bearings, which is divided into three parts: data preprocessing, network model training, and model testing.

3.4.1. Dataset

To verify the effectiveness of the proposed method, we conducted experiments using two real-world open datasets from Case Western Reserve University (CWRU) [36] and the Machinery Fault Database (MAFAULDA) [37]. The experimental device is shown in Figure 4.



(**a**) (**b**)

Figure 4. Experimental device. (**a**) CWRU; (**b**) MAFAULDA.

The CWRU dataset includes one normal condition type and three fault types: (i) inner race fault, (ii) outer race fault, and (iii) ball fault. These correspond with the four load types (0HP, 1HP, 2HP, and 3HP) and three diameter types (0.007″, 0.014″, and 0.021″). The dataset is divided into four categories: 48k Hz baseline, 48k Hz drive-end fault, 12k Hz drive-end fault, and 12k Hz fan-end fault. We only selected the dataset from the 12k Hz drive-end fault (corresponding with a sampling rate of 12 KHz) and from a unidirectional accelerometer installed at the drive-end bearing position in the CWRU dataset.

The MAFAULDA dataset is composed of a time series with five fault conditions—(i) horizontal misalignment, (ii) vertical misalignment, (iii) imbalance, (iv) underhang bearing fault (outer race fault, ball fault, and cage fault), and (v) overhang bearing fault (outer race fault, ball fault, and cage fault). Only the data from the underhang bearing fault, which were collected by three accelerometers on the radial, axial, and tangential directions, were selected here.

3.4.2. Data Preprocessing

The vibration signal of the electromechanical equipment bearing is a one-dimensional time series, and we needed to generate vibration images before use. More specifically, we divided each group of data into groups of 400 data points. Then, we used the Gramian Angular Field (GAF) method to encode the values of the time series as angular cosines and the timestamps as radii in order to represent the time series in polar coordinates. This process can preserve time dependencies and then calculate its GAF image in accordance with the defined GAF formula. This achieves a transformation from time series to image, thus obtaining the vibration images. The vibration signals of various fault types, and the vibration images generated from them, are shown in Figure 5.

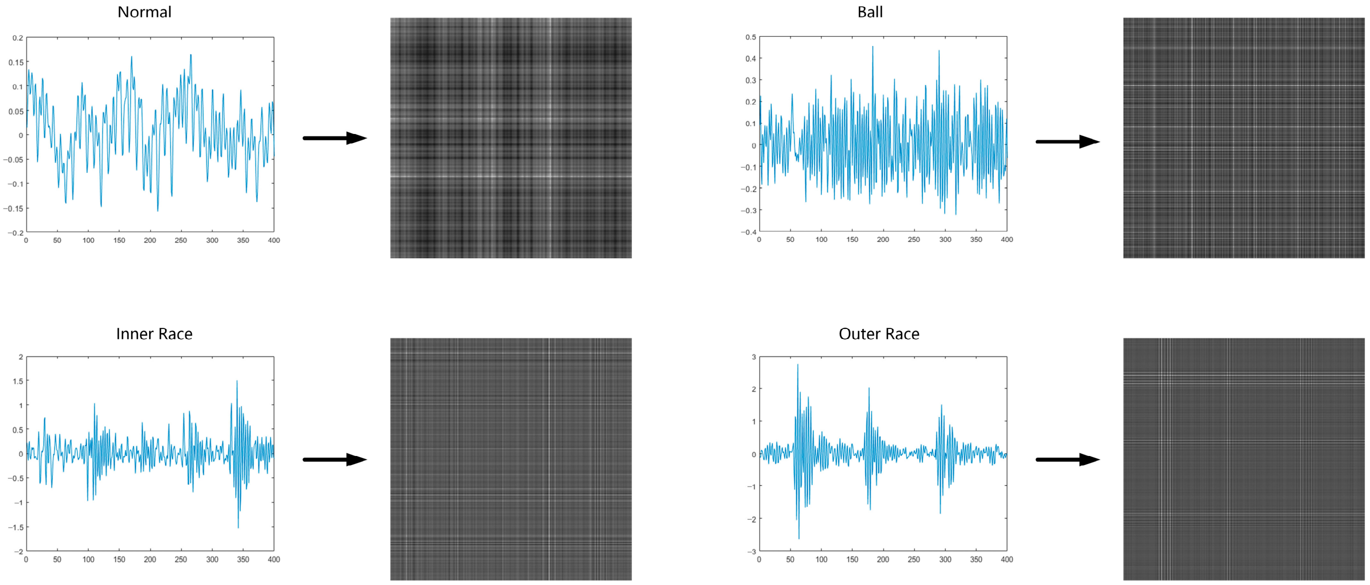


Figure 5. Vibration signals of various fault types and their generated vibration images.

3.4.3. Training and Testing of Models

1. The model parameters and various prototypes were trained using support and query set data. During the training process, the stochastic gradient descent (SGD) method was used to optimize and adjust parameters ϕ until a better performance was achieved. The training process of the WproNet is shown in Algorithm 1.

Algorithm 1 Training process of the WproNet

```
Input: Training Dataset $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$
Output: The trained network parameter $f_\phi$
Begin:
For $i$ in $[1, N]$ do:
    $D_i \leftarrow GAF(D_i)$
End For
For epoch to set value, do:
    For epoch to set value, do:
        For $i$in $[1, y_N]$ do:
            Randomly take $N_s$ samples $\{(x_1, y_1), \cdots, (x_{N_s}, y_{N_s})\}$ from $D$ as the support set $S$
            Randomly take $N_q$ samples $\left\{(x_1, y_1), \cdots, (x_{N_q}, y_{N_q})\right\}$ from $D/S$ as the query set $Q$
            For $i$in $[1, N_s]$ do:
                Compute distribution changes of the samples: $D_{KL}(x_i) = \sum_i \left[S \log \frac{S}{S - x_i}\right]$
                Normalization as weight: $W(f_\phi(x_i)) \leftarrow D_{KL}(\widetilde{x}_i)$
                Compute prototypical of the samples: $c_k = \frac{\sum_{i=1}^{|S_k|} W(f_\phi(x_i)) f_\phi(x_i)}{\sum_{i=1}^{|S_k|} W(f_\phi(x_i))}$
            End For
        End For
        $J \leftarrow 0$
        For $i$ in$[1, N_s]$do:
            For $[x, y_j]$ in $Q_{N_q}$do:
                Update loss: $J \leftarrow J + J(x, c_k)$
                Update the parameter $\phi$ via the SGD method
            End For
        End For
    End For
End For
```

2. After the network model was trained and various prototypes were obtained, the validation set was used to perform fault diagnosis testing. Figure 6 shows the flowchart of our model. The hyperparameters and parameter settings of the WproNet are shown in Tables 1 and 2.

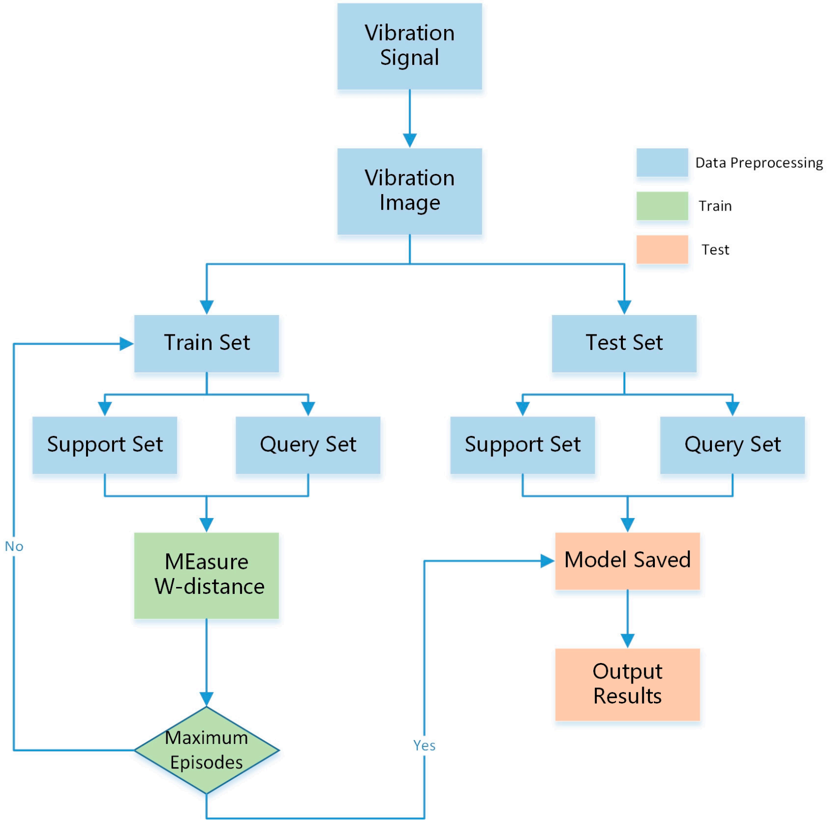


Figure 6. Flowchart of our model.

Table 1. Hyperparameter settings of the WproNet.

Name	Channels	Kernel Size	Stride	Input Size	Output Size	Activation Function
convolutional block1	64	3 × 3	1 × 1	80 × 80 × 64	40 × 40 × 64	Relu
convolutional block2	64	3 × 3	1 × 1	40 × 40 × 64	20 × 20 × 64	Relu
convolutional block3	64	3 × 3	1 × 1	20 × 20 × 64	10 × 10 × 64	Relu
convolutional block4	64	3 × 3	1 × 1	10 × 10 × 64	5 × 5 × 64	Relu
Distribution-Prototypical Layer	/	/	/	1600	1600	Softmax

Table 2. Parameter settings of the WproNet.

Parameter Type	Parameter Value
Optimizer	SGD
Initial learning rate	0.0005
Learning rate decay period	2000 episodes

4. Results

4.1. Comparative Experiments

In order to verify the outstanding performance of the proposed model, experiments were conducted using the same bearing dataset from the CWRU and MAFAULDA. The experiments were compared with representative classification algorithms based on statistics (SVM), CNNs (WDCNN), meta-learning (matching networks), and the original prototypical network. The settings for the parameters and hyperparameters are shown in Table 1.

1. SVM [9]: The Support Vector Machine is a supervised learning algorithm for classification and regression analysis. It is a binary classification model that finds the optimal hyperplane to achieve classification. SVM can handle non-linear classification problems well;
2. WDCNN [22]: Deep Convolutional Neural Networks with Wide First-layer Kernels is a traditional machine learning model based on deep convolutional neural networks (DCNNs). Its main feature is that it uses wide convolutional kernels to increase the number of features and reduce network depth. It requires a large number of samples for training;
3. Matching Networks [33]: Matching Networks is a meta-learning method that uses an attention-based approach to compare input samples with samples in the support set, thus enabling rapid model adaptation.
4. Prototypical network [28];
5. DSN—Conv4 [38]: Discriminative Deep Subspace Networks, the backbone of which is composed of Conv4;
6. PNMD [39]: Prototypical network based on the Manhattan distance.

The network model is built using the PyTorch 1.10.2 deep learning framework, and it is accelerated by CUDA 11.3 and CUDNN 8.3.2. The system environment is Windows 10, and the processor is Intel® Core™ i7-9700K CPU @ 3.60GHz, 16.0GB RAM, and the Nvidia GeForce RTX 3090 graphics card.

Accuracy is used as the primary evaluation metric.

We set different parameters for the N-way K-shot task in the experiment, selecting two and four categories and randomly selecting 10/20/50 images from each class to create a support set and ten images were selected for the query set. Training and fine-tuning were conducted for 2-way 10-shot/2-way 20-shot/2-way 50-shot and 4-way 10-shot/4-way 20-shot/4-way 50-shot, respectively, with episodes set to 100. Each epoch contained 100 episodes, and the epochs were set to 1000. We calculated the accuracy using the method of repeating the same scene 20 times and taking the average value. The results are shown in Tables 3 and 4.

Table 3. Average accuracies of the results for CWRU (%).

Model	**2-Way**			**4-Way**		
	10-Shot	**20-Shot**	**50-Shot**	**10-Shot**	**20-Shot**	**50-Shot**
SVM	74.78 ± 1.59	77.89 ± 1.41	82.85 ± 1.03	50.67 ± 2.62	58.21 ± 2.87	74.06 ± 1.53
WDCNN	42.35 ± 2.32	47.12 ± 2.17	59.64 ± 1.52	33.42 ± 2.89	38.80 ± 2.21	43.71 ± 1.98
Match Networks	72.93 ± 1.39	74.86 ± 1.31	80.31 ± 0.92	48.56 ± 2.28	57.78 ± 2.49	77.21 ± 1.67
Prototypical Networks	85.14 ± 1.13	85.68 ± 0.96	91.87 ± 0.70	59.77 ± 1.71	67.67 ± 1.82	85.02 ± 1.33
DSN- Conv4	88.39 ± 0.66	89.61 ± 0.53	95.52 ± 0.34	71.08 ± 0.70	72.36 ± 0.54	88.67 ± 0.30
PNMD	85.54 ± 1.61	85.80 ± 1.13	92.27 ± 0.69	61.37 ± 1.85	68.19 ± 1.35	87.05 ± 0.92
WProNet(ours)	**90.37 ± 0.83**	**91.42 ± 0.77**	**96.24 ± 0.75**	**71.79 ± 0.91**	**78.14 ± 1.14**	**89.68 ± 0.96**

Table 4. Average accuracies of the results for MAFAULDA (%).

Model	**2-Way**			**4-Way**		
	10-Shot	**20-Shot**	**50-Shot**	**10-Shot**	**20-Shot**	**50-Shot**
SVM	76.49 ± 1.64	80.16 ± 1.32	83.37 ± 0.88	51.67 ± 2.47	59.04 ± 2.20	77.43 ± 1.43
WDCNN	44.61 ± 2.09	46.12 ± 2.19	61.36 ± 1.87	34.12 ± 3.05	41.06 ± 2.51	47.26 ± 2.14
Match Networks	75.80 ± 1.40	78.20 ± 1.41	83.32 ± 0.85	50.61 ± 2.09	61.88 ± 1.89	82.64 ± 1.55
Prototypical Networks	86.02 ± 1.09	88.64 ± 1.05	93.50 ± 0.65	62.28 ± 1.38	70.07 ± 2.06	88.16 ± 0.83
DSN- Conv4	89.03 ± 0.51	91.15 ± 0.60	94.76 ± 0.38	70.84 ± 0.40	77.76 ± 0.45	90.33 ± 0.41
PNMD	85.79 ± 0.86	88.31 ± 0.96	93.97 ± 0.59	62.16 ± 1.66	70.19 ± 1.39	89.25 ± 0.86
WProNet(ours)	**91.70 ± 0.85**	**93.71 ± 0.70**	**96.45 ± 0.69**	**75.35 ± 1.32**	**80.73 ± 1.16**	**91.93 ± 0.94**

The experimental results show that for small-sample classification tasks, the performance of WProNet is better than traditional machine learning models and meta-learning methods. When different numbers of categories and sample sizes were used, the method proposed in this paper, WProNet, achieved the highest accuracy rate among the five fault diagnosis methods. This is because traditional machine learning models and CNN models have serious overfitting problems in small-sample classification tasks which require a large number of samples for training to achieve satisfactory results. In particular, the WDCNN network has poor accuracy when there are only ten samples. The accuracy rates for the two classification scenarios were only 42.35% and 33.42% with regard to the experiment concerning the CWRU dataset, respectively. With the increase in the number of samples, the overfitting of each network is alleviated, and the accuracy rate is improved. The SVM model shows unexpectedly high accuracy rates, achieving performance comparable to the matching network in various classification experiments. In the comparison experiment with the prototypical network, the improvement in accuracy of the four-way classification task is significantly better than that of the two-way classification task, and the improvement effect is most obvious in the 4-way 10-shot scenario, with an improvement of 12.02%. As the number of categories decreased and the number of samples increased, the gap between the two models gradually narrowed.

4.1.1. Ablation Experiment

We conducted ablation experiments on two datasets to demonstrate the effectiveness of the improved algorithm, as shown in Tables 5 and 6. These experimental results indicate that the prototypical network was calculated with the added distribution difference weight, which significantly outperformed the original prototypical network. The model with added weights enhanced the dependent relationship between the samples and global data, improved the model's feature extraction ability, and increased the accuracy of the model classification. All experimental metrics were improved, thus demonstrating the effectiveness of the model.

Table 5. Comparison of ablated model structures for CWRU (%).

Model	**2-Way**			**4-Way**		
	10-Shot	**20-Shot**	**50-Shot**	**10-Shot**	**20-Shot**	**50-Shot**
ProNet	85.14 ± 1.13	85.68 ± 0.96	91.87 ± 0.70	59.77 ± 1.71	67.67 ± 1.82	85.02 ± 1.33
W+ProNet	90.37 ± 0.83	91.42 ± 0.77	96.24 ± 0.75	71.79 ± 0.91	78.14 ± 1.14	89.68 ± 0.96

Table 6. Comparison of ablated model structures for MAFAULDA (%).

Model	**2-Way**			**4-Way**		
	10-Shot	**20-Shot**	**50-Shot**	**10-Shot**	**20-Shot**	**50-Shot**
ProNet	86.02 ± 1.09	88.64 ± 1.05	93.50 ± 0.65	62.28 ± 1.38	70.07 ± 2.06	88.16 ± 0.83
W+ProNet	91.70 ± 0.85	93.71 ± 0.70	96.45 ± 0.69	75.35 ± 1.32	80.73 ± 1.16	91.93 ± 0.94

4.1.2. Training Time Analysis

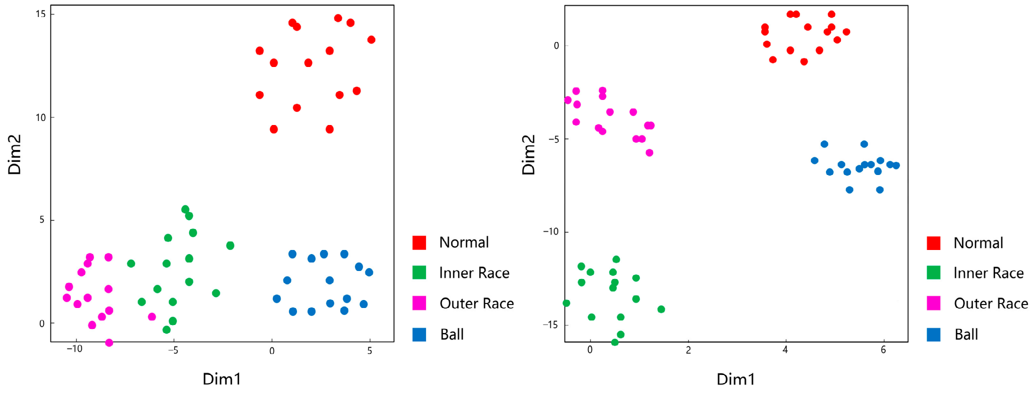
The training runtime of experimental models and our model, with regard to both datasets, CWRU and MAFAULDA, are shown in Table 7. To ensure a fair comparison, the hyperparameters of all the methods follow their original settings. Our training runtime is nearly the same as that of the match networks' runtime, and slightly greater than that of ProtoNet and PNMD. It is evident that even among the various models based on the same Conv-4 backbone networks, our model is highly efficient. Our approach achieves the best performance with only a modest increase in total training time.

Table 7. Training runtime comparison of models on CWRU and MAFAULDA datasets, under 2-way n-shot and 4-way n-shot classification scenarios.

Model	**Training Tasks**	**2-Way**			**4-Way**		
		10-Shot	**20-Shot**	**50-Shot**	**10-Shot**	**20-Shot**	**50-Shot**
SVM	100,000	0.5 h	0.8 h	1.2 h	1.3 h	1.8 h	2.3 h
WDCNN	100,000	1 h	1.3 h	2 h	2.6 h	3 h	4.3 h
Match Networks	100,000	1.7 h	2 h	2.2 h	2.5 h	4 h	6 h
Prototypical Networks	100,000	1.5 h	1.8 h	2.4 h	2.5 h	3.8 h	5 h
DSN-Conv4	100,000	8 h	13 h	19 h	15 h	21 h	35 h
PNMD	100,000	1 h	1.2 h	2 h	1.5 h	1.7 h	2.5 h
WProNet(ours)	100,000	2 h	2.3 h	3.5 h	2.4 h	3.3 h	4.4 h

4.1.3. Visualization Analysis

We used t-Distributed Stochastic Neighbor Embedding (t-SNE) [40] to visualize the performance of WProNet and prototypical networks on the test set, regarding 4-way 50-shot. The results are shown in Figure 7. It is evident that both WProNet and prototypical networks achieve a very good performance, but the prototypical networks' ability to generalize is much weaker than that of the WProNet.



(**a**) (**b**)

Figure 7. Visualization of the features extracted by the Prototypical Networks and the weight prototypical networks (WProNet) via t-SNE. (**a**) Prototypical Networks; (**b**) WProNet.

4.1.4. Comparison of WProNet with Several Other Models

To observe the performance of the WProNet in various categories, we plotted the confusion matrix of a random test result on the validation set with 50 samples, as shown in Figure 8. The vertical axis represents the true label of the validation set, the horizontal axis represents the predicted label of the validation set, and the main diagonal axis represents the number of correctly predicted samples as per the network model.

The WProNet model incorporates the influence of the sample distribution into the network and reevaluates it, instead of treating all samples equally as in traditional prototypical networks. This effectively solves the characteristics of the long-tail distribution of health monitoring data, and it enhances the dependence relationship between samples and global data, thus effectively improving the accuracy of model classification.

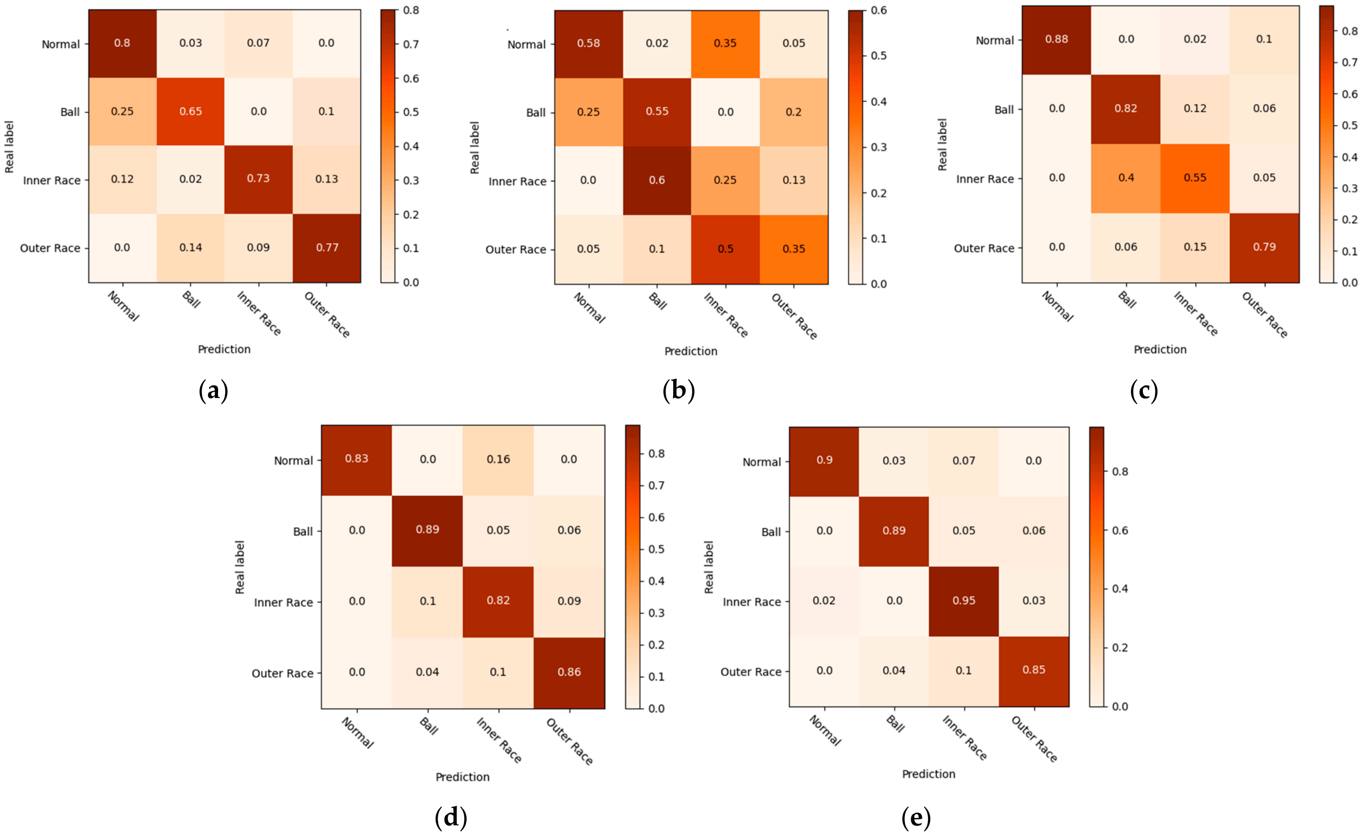


Figure 8. Confusion matrix of Models. (**a**) SVM; (**b**) WDCNN; (**c**) Match Networks; (**d**) Prototypical Networks; (**e**) WProNet (ours).

5. Discussion

When setting the initial learning rate, we tested the prototypical network and our model from 0.0001 to 0.001 in increments of 0.0001. As shown in Figure 9, we found that the best effect was achieved when the initial learning rate was set to 0.0005, and if it was set to other values, the effect would be reduced. Learning rates that are either too small or too large can lead to a decrease in model accuracy.

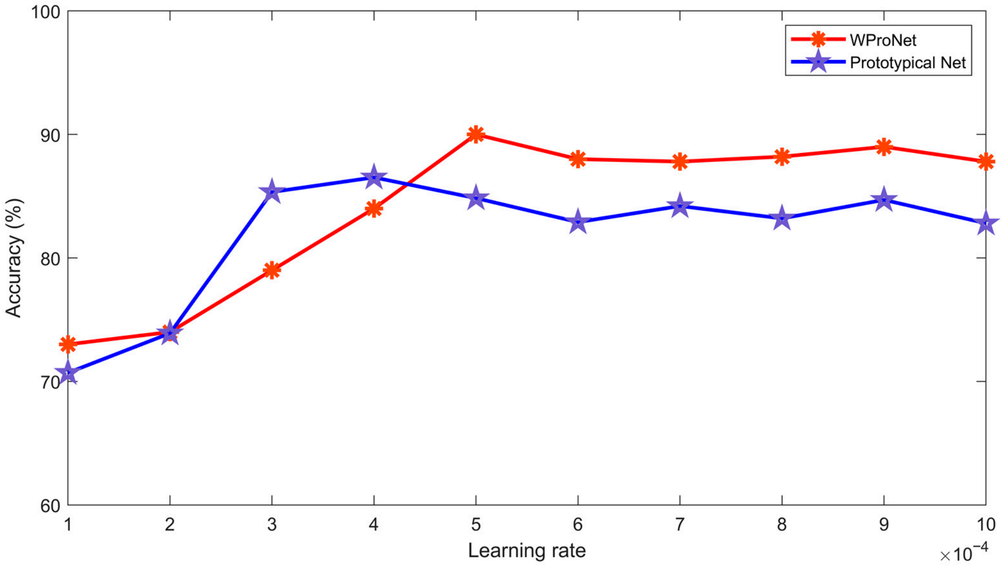


Figure 9. The 0.0001–0.001 interval for the learning rate (0.0001 is the step size) to test the Prototypical Network and our model.

When selecting the number of CNN layers, we tested 3, 4, 5, and 6 layers separately. The results of the experiment show that only when there are four layers does the model have the highest accuracy, and other situations will lead to a certain degree of underfitting and overfitting, which affects the accuracy of the model. It also shows that simply increasing the number of CNN layers will not increase the performance of the model.

Improving the feature optimization learning ability of machine learning methods is an important means to solve complex mechanical vibration signal problems. The WProNet model, which is based on the prototypical network proposed in this paper, is used to detect mechanical and electrical equipment bearing faults, incorporate the influence of sample distribution into the network, and allocate weights based on the proportion of sample distribution in the dataset, thus effectively solving the defects of health monitoring data regarding long-tail distributions, and enhancing the dependence relationship between samples and global data. By establishing feature prototypical networks of various health states in the measurement space, and then classifying and querying the health status of samples through distance metrics, the overfitting problem can be effectively alleviated, and the accuracy of fault diagnoses in small samples can be improved.

6. Conclusions

This article proposes a prototypical network based on the differences between computed sample distributions in order to identify the healthy state of rotating machinery with few samples. To test the performance of the proposed model, we conducted comparative experiments on the bearing dataset at CWRU and MAFAULDA, with other machine learning models such as SVM, WDCNN, and a prototypical network. The improved network model that was used on the CWRU dataset, increased the performance quality by 5.23%, 5.74%, and 4.37% in the 2-way 10-shot, 2-way 20-shot, and 2-way 50-shot classification tasks, respectively. It also increased the performance quality by 12.02%, 10.47%, and 4.66% in the 4-way 10-shot, 4-way 20-shot, and 4-way 50-shot classification tasks, compared with the prototypical network. The results from the MAFAULDA dataset also showed significant improvement.

The experimental results have shown that this method demonstrates great potential for complex signal classification and recognition tasks. It can learn data features from the minimum number of samples, thus indicating that the model is feasible and effective for classifying and recognizing complex signals and time-series data. Future studies will focus on applying this model to other industrial applications to develop real-time fault detection for different types of machines. Additionally, recognizing and denoising complex signals will also be topics of interest for future research.

Author Contributions: Conceptualization, methodology, formal analysis, and editing, Z.W.; supervision, H.S.; visualization, W.X.; provided the experiment guidance and revised the manuscript, J.H.; participated in the design of the prototype, X.Z.; validation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Program of China (Grant No. 2022YFB3902300).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: I would like to thank Mingyu Yang, Hanyu Wang and Kaiqi Lang for their invaluable assistance and support. Their work, insights, and ideas were invaluable and greatly appreciated.

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

References

1. Pecht, M.G. *Prognostics and Health Management of Electronics*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
2. Mishra, S.; Pecht, M.; Goodman, D.L. In-situ sensors for product reliability monitoring. In *Design, Test, Integration, and Packaging of MEMS/MOEMS 2002*; SPIE: New York, NY, USA, 2002; Volume 4755, pp. 10–19.
3. Kelkar, N.; Dasgupta, A.; Pecht, M.; Knowles, I.; Hawley, M.; Jennings, D. 'Smart'electronic Systems for Condition-Based Health Management. *Qual. Reliab. Eng. Int.* **1997**, *13*, 3–8. [CrossRef]
4. Engel, S.J.; Gilmartin, B.J.; Bongort, K.; Hess, A. Prognostics, the real issues involved with predicting life remaining. In Proceedings of the 2000 IEEE Aerospace Conference, (Cat. No. 00TH8484), Big Sky, MT, USA, 25 March 2000; Volume 6, pp. 457–469.
5. Hamadache, M.; Jung, J.H.; Park, J.; Youn, B.D. A comprehensive review of artificial intelligence-based approaches for rolling element bearing PHM: Shallow and deep learning. *JMST Adv.* **2019**, *1*, 125–151. [CrossRef]
6. Kirubarajan, T. Physically based diagnosis and prognosis of cracked rotor shafts. Proceedings of SPIE—The International Society for Optical Engineering; SPIE: New York, NY, USA, 2002; p. 4733.
7. Shen, H.; Li, Z.; Qi, L.; Qiao, L. A method for gear fatigue life prediction considering the internal flow field of the gear pump. *Mech. Syst. Signal Process.* **2018**, *99*, 921–929. [CrossRef]
8. Chen, L.; Zhang, X.; Wang, L. Research on Root Strain Response Characteristics of Inner Ring of Planetary Gear Transmission System with Crack Fault. *Sensors* **2023**, *23*, 253. [CrossRef] [PubMed]
9. Yin, Z.; Hou, J. Recent advances on SVM based fault diagnosis and process monitoring in complicated industrial processes. *Neurocomputing* **2016**, *174*, 643–650. [CrossRef]
10. Zhao, W.; Shi, T.; Wang, L. Fault diagnosis and prognosis of bearing based on hidden Markov model with multi-features. *Appl. Math. Nonlinear Sci.* **2020**, *5*, 71–84. [CrossRef]
11. Zhang, D.; Li, W.; Wu, X.; Lv, X. Application of simulated annealing genetic algorithm-optimized back propagation (BP) neural network in fault diagnosis. *Int. J. Model. Simul. Sci. Comput.* **2019**, *10*, 1950024. [CrossRef]
12. Graves, G.; Wayne, M.; Reynolds, T.; Harley, I.; Danihelka, A.; Wayne, M.; Reynolds, T.; Harley, I.; Danihelka, A.; Grabska-Barwińska, A.P. Badia, Hybrid computing using a neural network with dynamic external memory. *Nature* **2016**, *538*, 471.
13. Zbontar, J.; LeCun, Y. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.* **2016**, *17*, 1–32.
14. Jin, K.H.; McCann, M.T.; Froustey, E.; Unser, M. Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.* **2017**, *26*, 4509–4522. [CrossRef]
15. Anderson, M.A.; Burda, J.E.; Ren, Y.; Ao, Y.; O'Shea, T.M.; Kawaguchi, R.; Sofroniew, M.V. Astrocyte scar formation aids central nervous system axon regeneration. *Nature* **2016**, *532*, 195–200. [CrossRef] [PubMed]
16. Seixas, A.I.; Azevedo, M.M.; de Faria, J.P.; Fernandes, D.; Pinto, I.M.; Relvas, J.B. Evolvability of the actin cytoskeleton in oligodendrocytes during central nervous system development and aging. *Cell Mol. Life Sci.* **2019**, *76*, 1–11. [CrossRef] [PubMed]
17. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
18. Eren, L.; Ince, T.; Kiranyaz, S. A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *J. Signal Process. Syst.* **2019**, *91*, 179–189. [CrossRef]
19. Zhang, L.; Liu, Y.; Zhou, J.; Luo, M.; Pu, S.; Yang, X. An Imbalanced Fault Diagnosis Method Based on TFFO and CNN for Rotating Machinery. *Sensors* **2022**, *22*, 8749. [CrossRef] [PubMed]
20. Yan, J.; Kan, J.; Luo, H. Rolling bearing fault diagnosis based on Markov transition field and residual network. *Sensors* **2022**, *22*, 3936. [CrossRef]
21. Liu, H.; Zhou, J.; Zheng, Y.; Jiang, W.; Zhang, Y. Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. *ISA Trans.* **2018**, *77*, 167–178. [CrossRef]
22. Gao, Y.; Kim, C.H.; Kim, J.M. A novel hybrid deep learning method for fault diagnosis of rotating machinery based on extended WDCNN and long short-term memory. *Sensors* **2021**, *21*, 6614. [CrossRef]
23. Chen, Z.; Wang, Y.; Wu, J.; Deng, C.; Jiang, W. Wide Residual Relation Network-Based Intelligent Fault Diagnosis of Rotating Machines with Small Samples. *Sensors* **2022**, *22*, 4161. [CrossRef]
24. Xiao, D.; Huang, Y.; Zhao, L.; Qin, C. Domain adaptive motor fault diagnosis using deep transfer learning. *IEEE Access* **2019**, *7*, 80937–80949. [CrossRef]
25. Wang, C.; Qiao, Z.; Huang, Z.; Xu, J.; Fang, S.; Zhang, C.; Liu, J.; Zhu, R.; Lai, Z. Research on a Bearing Fault Enhancement Diagnosis Method with Convolutional Neural Network Based on Adaptive Stochastic Resonance. *Sensors* **2022**, *22*, 8730. [CrossRef] [PubMed]
26. Yue, K.; Li, J.; Chen, J.; Huang, R.; Li, W. Multiscale Wavelet Prototypical Network for Cross-Component Few-Shot Intelligent Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2022**, *72*, 1–11. [CrossRef]
27. Li, C.; Li, S.; Zhang, A.; He, Q.; Liao, Z.; Hu, J. Meta-learning for few-shot bearing fault diagnosis under complex working conditions. *Neurocomputing* **2021**, *439*, 197–211. [CrossRef]
28. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-shot Learning. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 4077–4087.
29. Jing, L.; Zhao, M.; Li, P.; Xu, X. A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. *Measurement* **2017**, *111*, 1–10. [CrossRef]

30. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
31. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
32. Nichol, A.; Achiam, J.; Schulman, J. On first-order meta-learning algorithms. *arXiv* **2018**, arXiv:1803.02999.
33. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; p. 29.
34. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; Hospedales, T.H. Learning to compare: Relation network for few-shot learning. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.
35. Wang, Z.; Oates, T. Imaging time-series to improve classification and imputation. *arXiv* **2015**, arXiv:1506.00327.
36. Case Western Reserve University. Bearing Data Center Web-Site: Bearing Data Center Seeded Fault Test Data. Available online: https://csegroups.case.edu/bearingdatacenter/pages/download-data-file (accessed on 27 November 2007).
37. Machinery Fault Database. Available online: http://www02.smt.ufrj.br/~offshore/mfs/page_01.html (accessed on 25 May 2021).
38. Simon, C.; Koniusz, P.; Nock, R.; Harandi, M. Adaptive Subspaces for Few-Shot Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 4136–4145.
39. Yu, Z.; Wang, K.; Xie, S.; Zhong, Y.; Lv, Z. Prototypical Network Based on Manhattan Distance. *Cmes-Comput. Model. Eng. Sci.* **2022**, *131*, 655–675. [CrossRef]
40. Maaten, L.V.D.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

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