IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 April 2023, accepted 3 May 2023, date of publication 8 May 2023, date of current version 16 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3274481

RESEARCH ARTICLE

Incremental Fault Diagnosis Method Based on Metric Feature Distillation and Improved Sample Memory

QILANG MIN, JUAN-JUAN HE, (Member, IEEE), PIAOYAO YU, AND YUE FU
College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081, China
Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan 430065, China

Corresponding author: Juan-Juan He (hejuanjuan@wust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62272355, Grant 61702383, and Grant 62176191.

ABSTRACT Incremental learning-based fault diagnosis systems (IFD) are widely used because of their ability to handle constantly updated fault data and types. However, the catastrophic forgetting problem remains the most crucial contemporary challenge facing IFD. This paper proposes an incremental fault diagnosis method based on metric feature distillation (MFD) and improved sample memory to solve this problem. First, the metric feature distillation is designed with metric learning methods and feature distillation. It uses distillation and triplet loss to constrain the network parameters of old and new tasks in the same feasible region, effectively alleviating catastrophic forgetting. Then, for a small amount of data that can be stored scenario, an improved sample memory strategy is introduced to reduce catastrophic forgetting further, called the center and hard sample memory (CAHM). It can better preserve the global information of the data, reducing the forgetting of old data information that needs to be preserved during the training process. Experimental results on CWRU and MFPT datasets verify the proposed method's effectiveness.

INDEX TERMS Fault diagnosis, incremental learning, signal processing, knowledge distillation.

I. INTRODUCTION

In many scenarios in fault diagnosis, not all fault data is available simultaneously. New fault data and fault types will emerge as diagnostic techniques advance. This requires the fault diagnosis system to process continuous data streams and update the diagnosis. However, conventional intelligent diagnostic systems are trained in a batch-learning setting. When new fault data and fault types appear, the model will be retrained using old and new fault data, resulting in a cumulative increase in time cost and computing resources. The fault diagnosis system based on incremental learning [1], i.e., the incremental fault diagnosis system [2] (IFD), overcomes this shortcoming and has been widely used. IFD requires a fault diagnosis classifier capable of diagnosing new fault types while maintaining the ability to analyze old ones.

Catastrophic forgetting [3] is a key problem in enabling incremental fault diagnosis systems to have continuous diagnostic capabilities. Recent research has made progress on how to resist catastrophic forgetting. This further promotes the development of deep learning [4]. However, most existing incremental learning methods typically use SoftMax Classifier, which requires the network to add additional parameters to new classes during incremental training continuously. It will lead to a shift in the parameter distribution towards the new class and network budget increase. Metric learning (embedding network) is an effective solution. For example, SDC [5] applies the metric learning method to incremental learning and proposes semantic drift compensation to resist forgetting. Zhou et al. [6] introduces metric learning to achieve few-shot class incremental learning by compressing the embedding of known classes and reserving new ones. Dou et al. [7] proposes a new Metric Learning framework to accurately capture the relationships among images. These methods effectively address the problem of

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Moinul Hossain.

increasing network parameters and the distribution drift in traditional classification networks during incremental learning by introducing metric learning. Moreover, these methods also demonstrate that metric learning networks have significant advantages over classification networks for similarity-based classifiers. However, there needs to be more application of metric learning to incremental fault diagnosis. The problem of catastrophic forgetting in incremental learning networks based on metric learning must also be investigated in depth. Inspired by the above ideas, this paper proposes and applies a new forgetting resistance strategy to fault diagnosis.

This paper proposes an incremental fault diagnosis framework based on metric feature distillation and improved sample memory to solve the catastrophic forgetting problem facing IFD from two aspects. i) The metric feature distillation (MFD) is designed with metric learning methods and feature distillation. Wavelet transform is used to convert the 1D signals into 2D images as the input to the neural network. MFD adds the embedding layer after the fully connected layer of ResNet18, and the network output is a 512-dimensional feature vector. The NCM classifier is used instead of the SoftMax classifier to avoid adding new output heads for new fault types. Moreover, MFD applies absolute knowledge distillation that calculates the absolute distance of new data in the teacher and student networks as the distillation loss function. It uses distillation and triplet loss to constrain the network parameters of old and new tasks in the same feasible region, effectively alleviating catastrophic forgetting. ii) For a small amount of data that can be stored scenario, this paper proposes an improved sample memory strategy to reduce catastrophic forgetting further, called the center and hard sample memory (CAHM). CAHM solves the problem of missing local information in most current sample memory strategies. CAHM calculates the distance of each sample to the other samples. Then, the samples with the largest distance are selected for storage. MFD is combined with CAHM (MMFD), to improve the forgetting resistance of the model synergistically. In summary, the major contributions of this paper are:

- This paper proposes an incremental fault diagnosis framework based on metric feature distillation (MFD) to constrain the network parameters of old and new tasks in the same feasible region, effectively alleviating catastrophic forgetting.
- For a small amount of data that can be stored scenario, this paper proposes an improved sample memory strategy to further enhance the resistance to forgetting.
- The signal is processed with wavelet transform to convert the 1D dataset of CWRU and MFPT into a 2D image dataset. The normalized images are input to train an incremental learning-based fault diagnosis model.

The paper is structured as follows: Some relevant knowledge and motivation are introduced in Section II. The proposed method is detailed in Section III. In Section IV, experimental results are analyzed to verify the effectiveness and efficiency of the proposed method. The impact of each part of the proposed method on the performance is discussed in Section V. Finally, the paper concludes with a summary of the proposed method in Section VI.

II. RELATED WORK AND MOTIVATION

A. INCREMENTAL FAULT DIAGNOSIS METHOD

In practical applications, the data and categories that need fault diagnosis are continuously updated and cumulatively increased. However, most traditional intelligent diagnosis methods are founded on batch learning. These methods are not suitable for handling continuously updated data streams. Recently, incremental learning-based fault diagnosis systems, called Incremental Fault Diagnosis systems, have been widely studied to deal with incremental fault diagnosis problems. Yu and Zhao [8] designed a broad convolutional neural network based on incremental learning to extract both fault tendency and nonlinear structure from the obtained data matrix. The model's ability to capture fault features is effectively enhanced using broad convolutional neural networks. The method allows the model to incrementally diagnoses by adding additional features for new fault data and types. In [9], an online fault diagnosis method is proposed to learn new fault types quickly. Oriented towards non-iid data, it solves the problem of data distribution drift under nonstationary industrial processes. The way extends the incremental learning assumption of iid of data to non-iid states to fit changing environments. Arunthavanathan et al. [10] use one-class SVM and neural network Permutation algorithms to complete self-learning and automatic diagnosis. The support vector machines detect unmarked faults to achieve self-updating of the fault database. The parameter contributions of the fault diagnosis model under the new fault database are extracted through a neural network permutation algorithm, and then the diagnosis model is self-updating.

Applying incremental learning methods to fault diagnosis has solved problems such as feature extraction and classification diagnosis in the case of incremental data types. However, there needs to be further research into the issue of catastrophic forgetting in incremental learning methods. This is the main problem to be solved to improve the model's online diagnostic capability and to adapt it to continuously updated fault types.

B. RECENT RESEARCH ON RESISTING CATASTROPHIC FORGETTING

There are three main approaches [11] to dealing with catastrophic forgetting: Replay methods [12], [13], Regularization based methods [14], [15], and Parameter isolation methods [16], [17]. Replay methods expect a joint training-like effect by replaying old data or pseudo-data when learning a new task. Experience replay [18] takes random samples from the old experience when learning a new task and uses them for training along with the new samples. This method

prevents the latest samples from overwriting old ones, thus protecting the old knowledge. Regularization-based methods consolidate old knowledge by introducing an additional Regularization term. Rannen et al. [19] train an autoencoder that prevents feature reconstruction from changing when training a new task, thus preserving the primary information from the old task. Minimizing all task losses by sharing the autoencoder, the task-specific decoder is trained to minimize task-specific losses. Parameter isolation methods assign different model parameters to each task or use compressed storage to free up network capacity for new tasks to prevent forgetting. In PathNet [20], each layer of the network can select K modules among N candidate modules and then randomly connect these modules to form multiple paths. These paths are shared by multiple users, thus enabling parameter reuse and thus resisting forgetting.

These methods yield better results, particularly Parameter isolation methods achieve incredibly high performance. However, the memory size increases linearly with the number of tasks. Regularization-based strategies overcome this problem. Nevertheless, the soft penalty introduced is insufficient to restrict the optimization process to stay in the feasible region of previous tasks [21], which sometimes leads to an increase in forgetting the previous task [11]. Replay methods generally perform better than Regularization based.

C. SAMPLE MEMORY STRATEGY

A series of different ways of storage strategies have arisen with data replay methods. They contribute significantly to solving the problem of catastrophic forgetting. Random selection is the simplest way to select samples to add to memory. The reservoir sampling algorithm [22] is used in Random selection to generate random numbers from all training samples to ensure that each sample has the same probability of being randomly selected. It maintains the distribution of the original data as much as possible. Herding [12] iteratively chooses a subset of samples for each class by calculating the distance between each sample and class mean and chooses the subset closest to the class mean in the learned feature space. Distance [23] selects samples by calculating the inverse distance from the sample to the class mean, approximating the distance from the sample to the decision boundary instead. Entropy [23] calculates the entropy of the SoftMax output and selects the samples with higher entropy. Since the SoftMax layer has been removed from our network, we artificially add a SoftMax output that does not affect the current network when calculating the entropy. In [24], the inverse form of Distance and Entropy is proposed. It is worth noting that Inverse Distance is remarkably similar to Herding but differs in that Inverse distance iteratively selects the single sample closest to the class mean. In contrast, Herding determines the sample set most comparable to the class mean.

D. MOTIVATION

Existing incremental fault diagnosis methods leave much to be desired. Firstly, SoftMax classifiers require constant modification of the fully connected layers to accommodate adding new classes during incremental learning. It not only causes the accumulation of additional parameters but also leads to parameter distribution drift toward the new task. Then, the regularization-based incremental learning approach is challenging to constrain the loss optimization for new tasks. Incremental learning methods based on stored memory need to find suitable sampling strategies.

Moreover, these strategies are only partially adaptable to fault diagnosis datasets. When applying incremental learning methods directly to fault diagnosis, some processing of data and networks is required in advance. Therefore, a practical framework must be investigated for the incremental fault diagnosis problem.

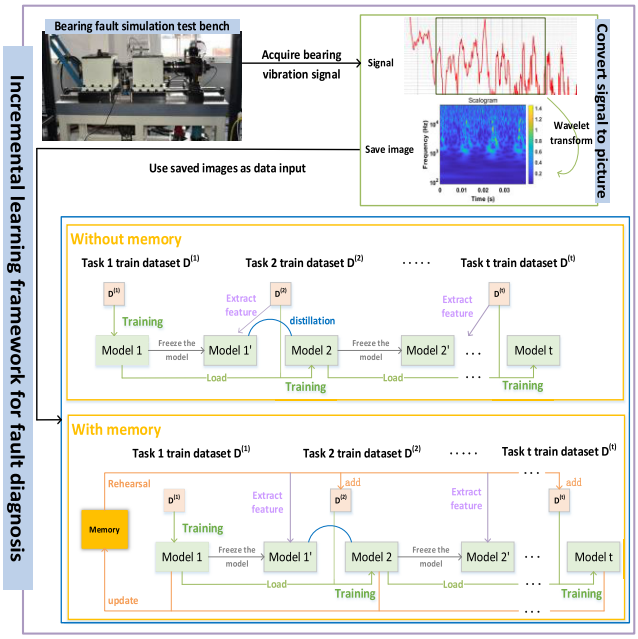

FIGURE 1. The general procedure of the proposed fault diagnosis method.

III. PROPOSED METHOD

In this section, two incremental fault diagnosis methods based on metric feature distillation and improved sample memory are developed. The general process of the proposed method is shown in Fig. 1. First, a 2D picture dataset D is constructed with wavelet transform by overlapping sampling from the signal. Secondly, the pictures are divided into t incremental task datasets D^t. Then, D^t is used as input to the incremental neural network, and MFD is built to be trained. For the scenarios where samples can be stored, the model's performance is further improved with the improved sampling strategy CAHM added. Finally, the fault classes of the test data are identified by the obtained incremental network model connected to the NCM classifier.

Without memory: When $t = 1$, the initial model is trained. When $t > 1$, first, the parameters of model t are passed to t' and the parameters of model t' are frozen. Second, when

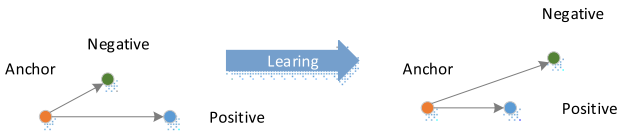

FIGURE 2. Triplet loss learning process. The input is a triplet including A (Anchor example), P (Positive example: samples that are in the same class as A), and N (Negative example: samples that are in a different class from A).

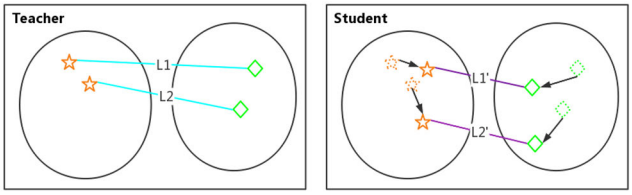

FIGURE 3. Knowledge distillation model. The same points mean the same class. The dashed graph represents the relative position of this sample in the teacher's feature space. L1 and L2 are the distances from the sample to other class samples.

training model t using new data, the output obtained from the new data passing through model t' is used to constrain the parameter distribution bias of model t. Finally, the parameters of the received new model t are passed to model t'. The knowledge distillation is marked with a blue line in Fig. 1.

With memory: For samples that can be stored, the model performance is further improved using a suitable sample management strategy. Update the samples in the buffer before the start of the next task. The old and new samples are fed into the network simultaneously during training. Moreover, unlike the situation without a buffer, the sample set from the buffer is used instead of the current dataset during distillation.

A. SIGNAL PROCESSING WITH WAVELET TRANSFORM

Before the incremental neural network process, sensor signals must be converted into images. The wavelet transform replaces the basis of the Fourier transforms with a wavelet basis that can decay. This way, the frequency is obtained while the time can be localized. The original signal is sampled and converted into 2D images. The transformed images are normalized and transmitted to the incremental neural network. The general form of the wavelet transform is:

$$\psi_{a,\tau}(t) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) * \psi(\frac{t-\tau}{a}) dt \tag{1}$$

where a is the scale, τ is the translation, t is the time, and $\psi(\cdot)$ is the wavelet basis function. $f(t)$ is the original signal.

B. METRIC FEATURE DISTILLATION

SDC [5] first added metric learning (embedding networks) to incremental learning. However, it only focuses on computational class centers and neglects the retention of old knowledge. This paper develops the idea of SDC by introducing knowledge distillation in the embedding network and improving the computation. Moreover, traditional convolutional neural networks use the SoftMax classifier. Nevertheless, the SoftMax classifier must keep adding additional parameters for new classes during incremental training. Besides, the network is vulnerable to class imbalance as the structure of the fully connected layer is changed. This paper adds an embedding layer to the last layer of Resnet18 to map the data features to a low-dimensional feature space. The loss function consists of a classification loss function and an anti-forgetting regularization term. MFD uses the triplet loss instead of the traditional cross-entropy loss function. When adding new classes, MFD needs no change in structure and learns the relationship between samples better.

The structure of triplet loss is explained in Fig. 2. The final optimization goal is to shorten the distance between A and P and increase the distance between A and N. The calculation formula is as follows:

$$d_+ = ||f(x_i^a) - f(x_i^p)||_2^2 \tag{2}$$

$$d_- = ||f(x_i^a - f(x_i^n))||_2^2 \tag{3}$$

$$L_{\mathrm{ML}} = \max(0, d_+ - d_- + \alpha) \tag{4}$$

where d_+ is the Euclidean distance between $f(x_i^a)$ and $f(x_i^p)$. d_- is the Euclidean distance between $f(x_i^a)$ and $f(x_i^n)$, and $f(x_i)$ is the feature vector of the sample x_i. The hyperparameter α determines the metric scale.

The old network is the teacher, and the current network is the student, as shown in Fig. 3. Traditional knowledge distillation uses a cross-entropy loss function to constrain the parameter optimization of the new task, thus resisting catastrophic forgetting. This paper uses absolute knowledge distillation to constrain the network parameters of the old and new tasks to the same feasible region. In other words, absolute knowledge distillation aims to reduce the distance of new samples in the old and new networks.

$$L_{\mathrm{KD}}^{\mathrm{abs}} = ||f^S(x) - f^T(x)|| \tag{5}$$

where $\|\cdot\|$ refers to the Frobenius norm. $f^S(x)$ is the feature vector of the current training sample x in the student network, and $f^T(x)$ is the feature vector of the sample x in the teacher network.

The parameters are updated jointly using the triplet loss and the distillation loss to prevent forgetting the embedded network.

$$L = L_{\mathrm{ML}} + \lambda L_{\mathrm{KD}}^{\mathrm{abs}} \tag{6}$$

where λ is the reconciling factor of the two loss functions.

Instead of the SoftMax classifier, the nearest class mean classifier (NCM) is used.

$$y* = \arg\min_{c=1,\ldots,t} \mathrm{dist}(f(x), U_C) \tag{7}$$

where $\mathrm{dist}(\cdot,\cdot)$ is the Euclidean distance, and $f(x)$ is the feature vector of the sample x. U_C is the class mean of class C.

Algorithm 1

input $X_s, \ldots, X_t$ // training samples
input M // total memory size
require θ // model parameters
require $P = (P_1, \ldots, P_{s-1})$ // current sample sets
$\theta \longleftarrow \text{Train}(X_s, \ldots, X_t; P, \theta)$
$m \longleftarrow M/(t * 2)$ // memory size of samples per class
for $y = 1, \ldots, s-1$ **do**
  $P_y \leftarrow \text{delete\_subsample}(P_y, m)$
end for
for $y = s, ..., t$ **do**
  $f \leftarrow \text{Extract\_features}(X_s, \ldots, X_t, \theta)$
  $P_y \leftarrow \text{add\_center\_data}(X_y, m, f)$
  dis_ap, dis_an $\leftarrow f$
  $P_y \leftarrow \text{add\_hard\_data}(X_y, m, \text{dis\_ap}, \text{dis\_an})$
end for
$P \leftarrow (P_1, \ldots, P_t)$ // new exemplar sets

C. IMPROVED SAMPLE MEMORY

The selected examples stored in memory should not only represent their corresponding classes but also make a significant contribution when they are replayed. The sample near the center of the samples is the most representative, which can provide higher accuracy when calculating the class mean required for the test. The hard samples are fuzzy, so they will be more challenging to train when replayed. To meet these two characteristics, this paper proposes a memory management strategy (center and hard memory, CAHM). This strategy consists of two parts:

- Half of the fixed memory stores samples near the sample center to keep the sample set's original characteristics.
- The other half of the fixed memory is used to store hard samples to improve the classification accuracy of fuzzy samples during training.

CAHM is detailed and explained in Algorithm 1. The hard samples are obtained using the triplet loss calculation method. First, CAHM gets the features of all candidate samples through the network. Secondly, these features calculate the distance between each sample and other samples. The distance *dis_ap* of similar samples and the distance *dis_an* of different class samples are obtained. CAHM calculates the distance of each sample to similar samples and other samples. The sample with the largest difference between them is selected for storage. The order in which the samples are chosen is recorded in decreasing order of importance. If fixed memory is used and some memory must be freed to make room for new samples, the samples with lower priority are removed first.

This paper combines CAHM with MFD to synergistically improve the model's performance. Equation (5) is modified as follows:

$$L_{\text{KD}}^{\text{abs}} = ||f^S(x_m) - f^T(x_m)|| \tag{8}$$

where x_m is the sample set of the buffer.

TABLE 1. Description of the ten bearing conditions of CWRU.

Bearing conditions	Fault diameter	Outer race fault orientation	Label
Ball fault	7	n/a	1
Inner race fault	7	n/a	2
Outer race fault	7	6 o'clock	3
Ball fault	14	n/a	4
Outer race fault	14	6 o'clock	5
Ball fault	21	n/a	6
Inner race fault	21	n/a	7
Outer race fault	21	6 o'clock	8
Ball fault	28	n/a	9
Normal	0	n/a	10

TABLE 2. Description of the eighteen bearing conditions of MFPT.

Bearing conditions	Load(lb)	Time(s)	Label
Inner race fault	0	3	1
Inner race fault	50	3	2
Inner race fault	100	3	3
Inner race fault	150	3	4
Inner race fault	200	3	5
Inner race fault	250	3	6
Inner race fault	300	3	7
Outer race fault	270	6	8
Outer race fault	270	6	9
Outer race fault	270	6	10
Outer race fault	25	3	11
Outer race fault	50	3	12
Outer race fault	100	3	13
Outer race fault	150	3	14
Outer race fault	200	3	15
Outer race fault	250	3	16
Outer race fault	300	3	17
Baseline	270	6	18

IV. EXPERIMENTAL VERIFICATION

A. EXPERIMENTAL SETUP

Two well-known public experimental datasets are used to verify the proposed method's validity. One is from the Case Western Reserve University (CWRU) Bearing Data Center [25], and the other is Machinery Failure Prevention Technology (MFPT) [26] society rolling element vibrational data set. The data are shown in Table 1 and Table 2.

The CWRU dataset consists of a 1.5KW (2 horsepower) motor under different working conditions (including 2 sampling frequencies, 4 rotation speeds and load). It uses EDM technology to simulate different fault severity on other bearings (including 3 fault types, 5 fault sizes, and 3 fault directions) vibration signal collection results. The two sampling frequencies are 12Khz and 48Khz. The four driving motor speeds are 1730, 1750, 1772, and 1797 rpms, respectively. The three types of failures are inner ring failure (IRD), outer ring failure (ORD), and ball failure (BD). The five types of faults are respectively 7, 14, 21, 28 mils, and 0.04 inches in diameter. The first three fault diameters are SKF bearings with the latter two fault diameters equivalent. NTN bearings. The three failure directions are the faults at 3 o'clock (directly located in the loaded area), 6 o'clock (orthogonal to the loaded area), and 12 o'clock in the v outer bearing ring of the drive and fan ends. This paper selects 10 vibration signals

TABLE 3. Experimental parameter setting.

Types	CWRU Value	MFPT Value
Network	ResNet18	ResNet18
Epoch	10	30
Batch Size	64	32
Learning Rate	(1e-4)/9	1e-5
Weight Decay	2e-4	2e-4
Optimizer	Adam	Adam
Dimension of Embedding Space	512	512
Tradeoff Hyperparameter λ	0.5	0.5

with a sampling rate of 12khz, a driving motor speed of 1797rpms, and a load of 0hp. One is a normal vibration signal, and 9 are vibration signals with different faults.

The MFPT dataset is provided by the Mechanical Failure Prevention Technology Association. Acceleration data were collected for 6 seconds under baseline conditions with a load of 270 lb and a 97,656 Hz sampling rate. In total, 10 outer and 7 inner race failures were tracked. These included 3 outer raceway failures, including a 270 lb load and a 97,656 Hz sampling rate for 6 seconds. Seven additional outer race failures are evaluated at various loads: 25, 50, 100, 150, 200, 250, and 300 lb. Faults are sampled at a rate of 48,828 Hz for 3 seconds. Seven inner ring faults are analyzed at loads of 0, 50, 100, 150, 200, 250, and 300 pounds. The sampling rate for the inner ring failures is 48,848 Hz for 3 s. The MFPT is classified as follows: normal baseline (N), inner ring failure (IR), and outer ring failure (OR). The raw data consisted of the following data points: N is 1757808 data points, IR is 1025388 data points, and OR is 2782196 data points. The raw data consisted of the following data points: N is 1757808 data points, IR is 1025388 data points, and OR is 2782196 data points.

The specific parameter settings are shown in Table 3. This paper uses ResNet18, an 18-layer convolutional neural network with 5 residual blocks, as the basic network for incremental fault diagnosis. An Embedding layer is added to the output layer of ResNet18, and the network's output is a 512-dimensional embedded feature vector. Instead of the traditional SoftMax classifier, the NCM classifier is used. The Adam optimizer is used instead of SGD.

This paper runs all experiments on a desktop computer with a GTX 1065, an Intel Core i7-9750H, and 8G RAM. The codes for this paper and the comparison method are written in PyTorch 1.4, and Python 3.7.

B. PREPROCESS OF ORIGINAL DATA

Since the sensor can only capture the vibration signal of the rolling bearing, it is necessary to process the original vibration signal and convert it into images. The wavelet transform can amplify the details of the data and can extract the features better, so this paper converts the vibration signal into pictures by wavelet transform. In the CWRU dataset, each bearing condition contains 2400 samples, among which 2000 samples are randomly selected as training samples, and 400 samples

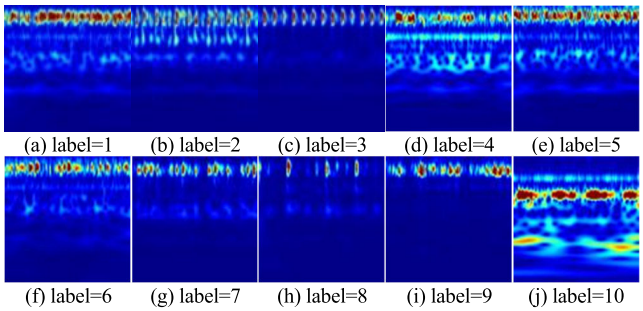

FIGURE 4. Examples of CWRU with wavelet transform.

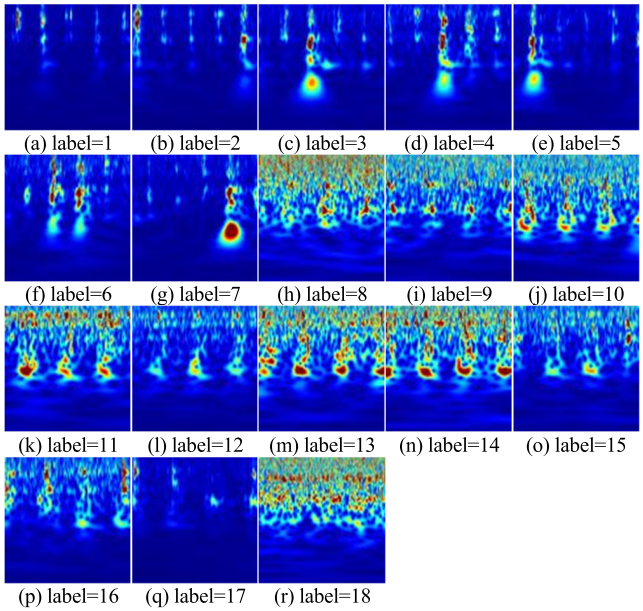

FIGURE 5. Examples of CWRU with wavelet transform.

are test samples. Each sample has 4096 sampling data points. In the MFPT dataset, the data overlap method is used because of the small number of images composed of data points. The fault samples with a sampling rate of 6 seconds overlapped 1953 data points. For the sample with a sampling rate of 3 seconds, 651 data points overlapped. The total images generated from the dataset include 3480 images for the training sample and 838 images for the test sample. The transformed images are shown in Fig. 4 and Fig. 5. Further, the constructed image is processed by MFD and MMFD.

C. MFD WITHOUT MEMORY

This section compares MFD with current classical incremental learning methods (all without memory) on CRWU and MFPT. These incremental learning methods include the classical LWF [27], EWC [28], SI [29], and SDC. This paper reports the diagnostic accuracy of these methods on two fault datasets using a two-increment format. Ten experiments are performed to calculate the average accuracy to avoid the specificity and chance of the results. The time cost in Fig. 8 is the result of 10 epoch runs.

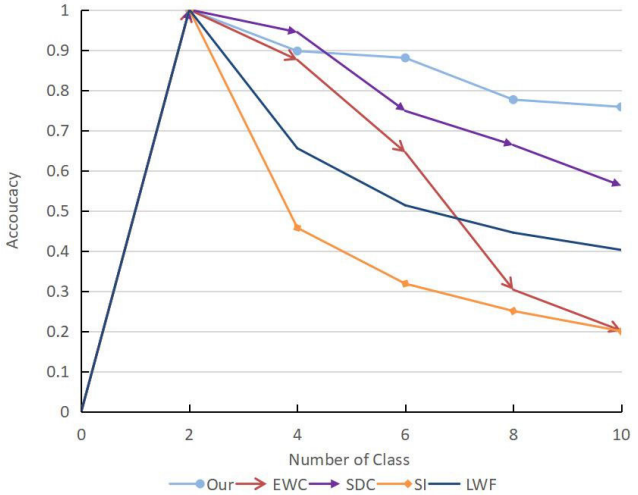

FIGURE 6. Experimental results of different methods for incremental class training on CWRU.

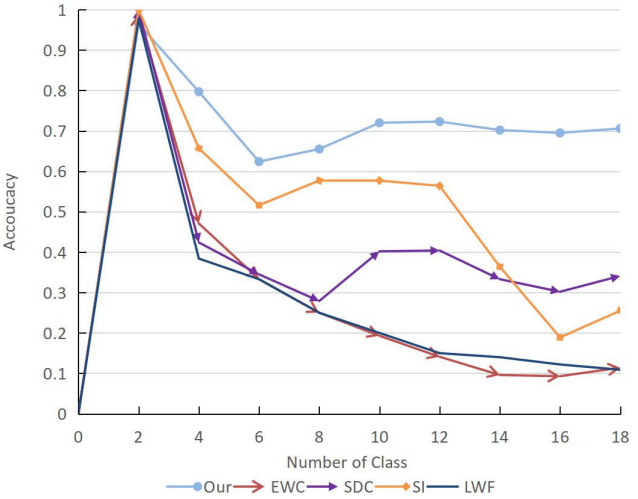

FIGURE 7. Experimental results of different methods for incremental class training on MFPT.

Experimental results in Fig. 6 show that our method (MFD) significantly outperforms LWF, EWC, SDC, and SI on the CWRU. Although MFD performs slightly worse than SDC in the second diagnostic classification, it outperforms SDC after that. Compared to SDC, MFD improves the model accuracy by about 18.6%. Experimental results in Fig. 7 show that MFD still achieves excellent diagnostic results (about 40% improvement over SDC) when the number of classification tasks increases. The data in Fig. 8 shows that MFD and LWF have similar time costs and are lower than other algorithms because both algorithms have the same time complexity.

D. MFD WITH MEMORY

MMFD is compared with classical sample memory-based methods for the scenario where samples can be stored. Experience Replay (ER) [18] merges memory samples with current samples for training. Meta-Experience Replay

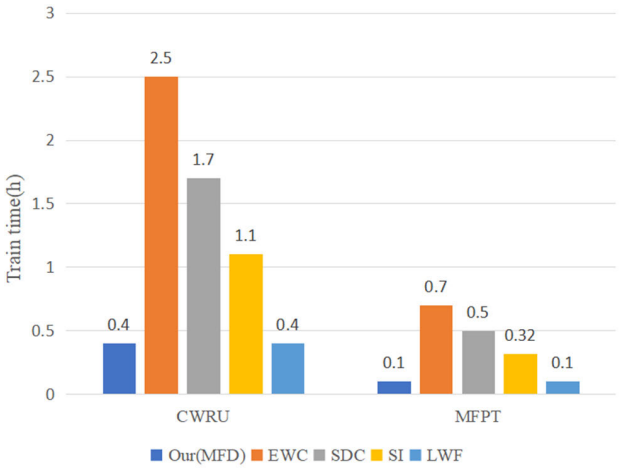

FIGURE 8. Time cost of MFD and other regularization-based methods on CWRU and MFPT.

TABLE 4. Average accuracy on CWRU and MFPT for different rehearsal method (memory size=200).

Method	CWRU (time: h)	MFPT (time: h)
MMFD	**90.4±0.9(1.1)**	**91.5±0.7(0.3)**
FDR	60.0±1.8(18.0)	85.4±4.8(2.0)
GSS	73.1±3.4(11.7)	67.4±3.2(1.9)
iCaRL	10.0±0.1(4.5)	19.1±4.8(1.8)
HAL	12.7±1.8(28.1)	10.7±5.1(6.8)
MER	**91.7±1.3(30.0)**	**92.2±1.8(5.4)**
GEM	12.4±2.2(140.0)	15.0±5.4(32.0)
JOINT	98.4±0.4(1.0)	98.9±0.2(0.2)
Finetuning	20.0±0.3(0.4)	11.6±0.4(0.1)

(MER) [30] uses a Meta-learning approach to replay past data. Gradient-based Sample Selection (GSS) [31] finds a constrained subset that maximizes the sample diversity. Hindsight Anchor Learning (HAL) [32] introduces "anchoring" to complement experience replay and uses bilevel optimization to update knowledge. iCaRL combines replay and knowledge distillation strategies. Unlike the parameter L2 distance optimization method, FDR [33] applies the function L2 distance to optimize. Gradient Episodic Memory (GEM) [34] leverages old training data to build optimization constraints to optimize the gradient loss while training new tasks.

Experimental results in Table 4 show that MMFD performs well on both datasets due to the improved sample management strategy and the superiority of the overall algorithm. FDR performs well on the MFPT dataset but less on the CWRU dataset. Like HAL and GEM, iCaRL shows unsatisfactory results, even lower than Finetuning (without memory and distillation). This paper compares the performance of iCaRL, HAL, and GEM on classical datasets (e.g., CIFAR10) and discusses why these methods do not apply to CWRU and MFPT. These methods are not suitable for incremental learning classes with high-class similarity. The high-class similarity makes training for the current task significantly less efficient. Combining this and catastrophic forgetting prevents

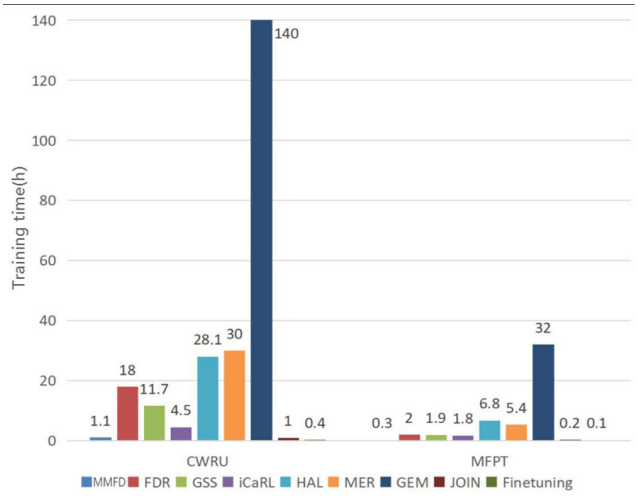

FIGURE 9. Time cost of MMFD and other replay-based methods on CWRU and MFPT.

TABLE 5. Average accuracy on CWRU and MFPT for independent method (memory size=200).

Method	CWRU	MFPT
Finetuning	20.0±0.3	11.6±0.4
MFD	75.8±0.2	70.6±0.3
CAHM	87.7±1.2	88.9±1.1
CAHM & MFD (MMFD)	90.4±0.9	91.5±0.7

obtaining good classification results in all tasks. When comparing MMFD with MER on similar time overhead (reducing the number of the running epoch of MER to shorten the time), the results show that MMFD outperforms MER. However, when the number of calendar elements of MER is increased, it is found that the accuracy of MER also increases and even exceeds the accuracy of our algorithm, as shown in Table 4. It is worth noting that the time cost of MER is enormous due to the high complexity of the algorithm. The data in Fig. 9 show that the time cost of MER is about 30 times higher than that of MMFD on CWRU and about 18 times higher than that of MMFD on MFPT. This problem has to be considered in industrial applications. In addition, FDR and GSS do not perform better with increased time and memory costs. MER, GSS, HAL, and GEM experienced an intractable training time under the same setting.

V. DISCUSSION

A. EFFECT OF CAHM ON MFD

To independently evaluate the benefit of our proposed MFD and CAHM, this paper performs an ablation study using the CWRU and MFPT datasets. Three scenarios are considered: 1) with MFD and CAHM, 2) only MFD, 3) only CAHM, and 4) without both (Finetuning).

Experimental results show that both components contribute to the algorithm's performance, as evidenced by the accuracy tests in Table 5. The accuracy without Distillation and CAHM is extremely low, attributed to catastrophic

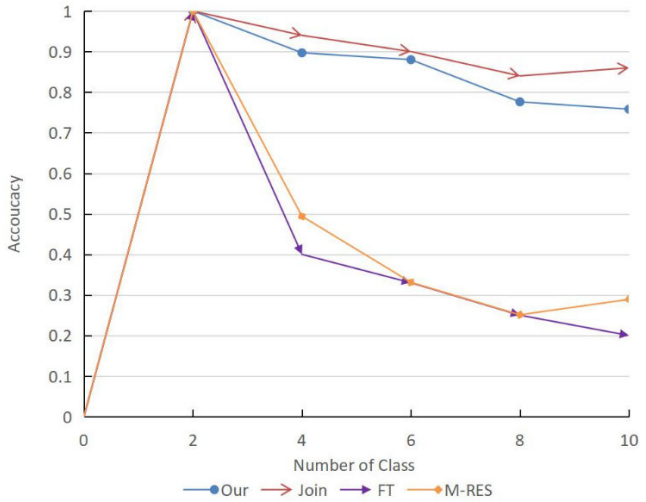

FIGURE 10. Experimental results on CWRU (base=2, task=5).

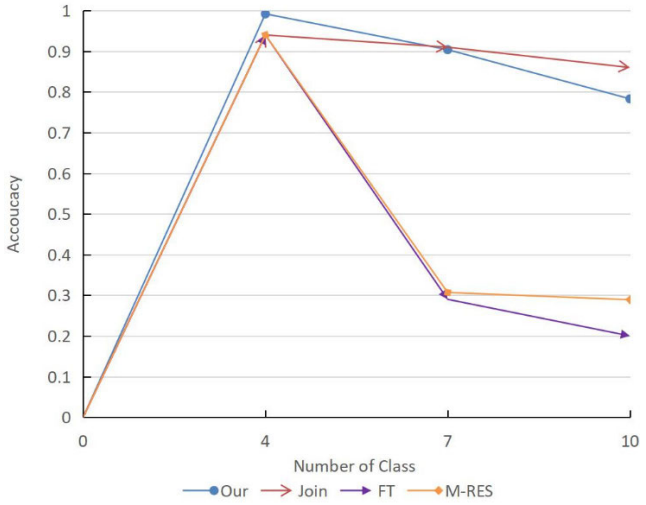

FIGURE 11. Experimental results of on CWRU (base=4, task=3).

forgetting. The combination of Distillation and CAHM provides a better model, although they both work well alone in resisting catastrophic forgetting.

B. DISCUSSION OF THE MFD COMPONENTS

This section investigates fault diagnosis accuracy under different classes of incremental conditions. Experiments are performed to verify the effectiveness of each component of the MFD. This paper compares the following cases: 1) FT: Resnet18 and cross-entropy loss function without measures to prevent catastrophic forgetting (lower band). 2) M-RES: Resnet18 network with metric learning added (FC layer deleted and embedding layer added) and the triplet loss function. 3) Join: in the Resnet18 classification network, each time a new category is added, all the categories that have been acquired are fully trained and are the fault diagnosis results (upper band). 4) our: the MFD proposed in this paper.

TABLE 6. Specific incremental learning results on the CWRU (accuracy percentage).

Method	model	Task 1	Task 2	Task 3	Task 4	Task 5	Average accuracy
MFD	Model 1	1.000					1.000
	Model 2	1.000	0.794				0.897
	Model 3	0.940	0.787	0.914			0.880
	Model 4	0.863	0.669	0.871	0.703		0.776
	Model 5	0.866	0.593	0.841	0.650	0.840	0.758
EWC	Model 1	1.000					1.000
	Model 2	0.750	1.000				0.875
	Model 3	0.301	0.636	0.998			0.645
	Model 4	0.003	0.111	0.108	0.995		0.303
	Model 5	0.020	0.000	0.000	0.020	0.798	0.201
SDC	Model 1	1.000					1.000
	Model 2	0.991	0.896				0.944
	Model 3	0.789	0.490	0.960			0.748
	Model 4	0.627	0.204	0.912	0.910		0.663
	Model 5	0.169	0.076	0.720	0.887	0.958	0.562
SI	Model 1	1.000					1.000
	Model 2	0.000	0.914				0.457
	Model 3	0.000	0.000	0.955			0.318
	Model 4	0.000	0.000	0.000	1.000		0.250
	Model 5	0.000	0.000	0.000	0.000	1.000	0.200
LWF	Model 1	0.972					0.972
	Model 2	0.836	0.474				0.655
	Model 3	0.792	0.273	0.473			0.513
	Model 4	0.729	0.198	0.344	0.509		0.445
	Model 5	0.546	0.208	0.344	0.498	0.415	0.402

TABLE 7. Comparison with different sampling strategies for three episodic memory sizes 200, 500, and 1000 on CWRU and MFPT.

Method	CWRU			MFPT		
	200	500	1000	200	500	1000
CAHM	**87.7±1.2**	**91.2±0.5**	92.6±0.4	**88.9±1.1**	**93.3±0.4**	94.5±0.3
Random	78.6±1.7	87.8±0.7	91.8±0.5	81.3±2.4	91.7±0.4	94.0±0.5
Herd	83.6±0.9	89.7±1.1	92.3±0.4	88.3±2.5	92.3±0.9	94.3±0.9
Distance	65.9±2.1	88.3±0.6	**93.2±0.2**	84.3±0.6	93.0±1.0	**95.6±0.3**
Entropy	67.2±2.5	79.2±1.9	85.7±1.1	83.7±0.2	91.8±0.8	93.5±0.3
Inverse Distance	78.4±1.5	88.4±0.6	91.7±0.3	86.3±1.6	92.5±0.4	93.7±0.2
Inverse Entropy	62.6±2.6	77.3±1.5	83.6±0,7	87.8±0.5	92.4±0.4	95.2±0.6

This paper conducted 10 experiments and took the average diagnostic accuracy of each for the study. Experimental results are shown in Fig. 10 and Fig. 11. (Base: the number of categories trained for the first time. Task: the number of tasks.)

Experimental results from Fig. 10 and Fig. 11 show that the diagnostic results of the embedding network perform better than those of the classification network. The final diagnostic accuracy of M-RES improves by about 9% compared with FT, which shows that Triple Loss function works better than the Cross-Entropy function for different classes of incremental conditions. Fig. 8 shows that the results of MFD are almost the same as Join of the first 3 diagnoses. Fig. 11 indicates that the results of MFD are nearly the same as Join of the first 2 diagnoses. This illustrates the efficiency of MFD. FT always performs the worst, emphasizing the impact of catastrophic forgetting in incremental learning.

To better illustrate the effectiveness of different methods for incremental fault diagnosis, Table 6 lists the results of each task on CWRU. After training a new model, this paper tests the samples from the previously trained task separately for each task. The aim is to show how well the old data retain knowledge on the new model and to illustrate better the effectiveness of each method in resisting forgetting.

C. COMPARISON OF CAHM WITH OTHER SAMPLING METHODS

There are different sampling strategies for the scenario where sample memory can be stored, as introduced in Section II. Table 7 compares the performance of the various sampling strategies for 200, 500, and 1000 memory sizes.

Experimental results show that CAHM performs better than other strategies on CWRU. As the memory size increases, all methods perform better. From the comparison between distance and inverse distance, it can be seen that distance performs better below 1000 memory sizes. This gives us an insight: if the samples in the buffer are sufficient to represent the characteristics of the original data, the samples with highly uncertain decision boundaries will contribute more to the training model. The same conclusion can be drawn from entropy and inverse entropy. Whereas the memory size is small, our method is superior to other methods. It can be inferred that hard samples are more suitable for few-shot incremental learning. The random process has less cost, and when high accuracy is not required, and computational cost is minimized, random is a better choice. Entropy and inverse entropy have always maintained low accuracy and high cost, which is unsuitable for our overall mechanism and may have unexpected effects on other algorithms.

On the other hand, due to the different sizes of the two datasets, the data trend is slightly different, and the accuracy of the various methods on the MFPT datasets is generally higher than that on CWRU. CWRU contains 20,000 training samples and 4,000 test samples, a total of 10 classes, while MFPT contains 3,480 training samples and 838 test samples, 18 classes. The data in Table 7 show that the effect of entropy and inverse entropy in MFPT is better than in CWRU. The possible reason is that the data set of MFPT is smaller on the one hand, and the gap between MFPT classes is smaller on the other. Keeping a large buffer for each class in a practical application is challenging. Therefore, our method will be a good choice.

VI. CONCLUSION

To solve the catastrophic forgetting problem facing incremental fault diagnosis, this paper designs two incremental fault diagnosis frameworks that apply to two scenarios. First, inspired by metric learning and knowledge distillation, this paper proposes an incremental fault diagnosis framework based on metric feature distillation (MFD). Second, for scenarios where a small amount of fault data can be stored, this paper proposes a CAHM sample memory strategy. Combining MFD, this paper designs an incremental fault diagnosis framework based on MFD and CAHM (MMFD). Experimental results on CWRU and MFPT show that the proposed

framework can continuously maintain the diagnostic capability with high accuracy with increasing fault data and types.

Despite the promising experimental results, the problems of incremental fault diagnosis still need to be fully solved. First, the CWRU and MFPT datasets used in this paper are both rolling bearing fault datasets, and the experimental results under other datasets have yet to be discovered. Second, the catastrophic forgetting problem must be fully solved at this stage, which makes the incremental fault diagnosis fail to achieve the joint training results.

In future work, we plan to solve both problems. We will first use the proposed framework for more complex fault diagnosis datasets, testing and optimizing the algorithmic framework. Moreover, we will further investigate how to solve the catastrophic forgetting problem and apply it to more domains. Some deep learning models have been used in various fields to improve the performance of the models, for example, medical diagnosis [35], musculoskeletal modeling [36], genetic engineering [37], and so on. These models, however, have to face the problem of model growth or memory growth under continuous data streams, and the fundamental solution to them is to overcome catastrophic forgetting.

REFERENCES

[1] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16071–16080.

[2] X. Gu, Y. Zhao, G. Yang, and L. Li, "An imbalance modified convolutional neural network with incremental learning for chemical fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 3630–3639, Jun. 2022.

[3] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, pp. 1–9.

[4] J. Zhang, Y. Li, W. Xiao, and Z. Zhang, "Non-iterative and fast deep learning: Multilayer extreme learning machines," *J. Franklin Inst.*, vol. 357, no. 13, pp. 8925–8955, 2020.

[5] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. Van De Weijer, "Semantic drift compensation for class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6982–6991.

[6] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, L. Ma, S. Pu, and D.-C. Zhan, "Forward compatible few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9046–9056.

[7] J. X. Dou, L. Luo, and R. M. Yang, "An optimal transport approach to deep metric learning (student abstract)," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 11, pp. 12935–12936.

[8] W. Yu and C. Zhao, "Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability," *IEEE Trans. Ind. Electron.*, vol. 67, no. 6, pp. 5081–5091, Jun. 2020.

[9] H. Zhou, H. Yin, D. Zhao, and L. Cai, "Incremental learning and conditional drift adaptation for nonstationary industrial process fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 5935–5944, Apr. 2023.

[10] R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, "Autonomous fault diagnosis and root cause analysis for the processing system using one-class SVM and NN permutation algorithm," *Ind. Eng. Chem. Res.*, vol. 61, no. 3, pp. 1408–1422, Jan. 2022.

[11] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[12] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2001–2010.

[13] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 1–8.

[14] S. Tang, P. Su, D. Chen, and W. Ouyang, "Gradient regularized contrastive learning for continual domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 3, pp. 2665–2673.

[15] K. J. Joseph, S. Khan, F. S. Khan, R. M. Anwer, and V. N. Balasubramanian, "Energy-based latent aligner for incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7452–7461.

[16] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016, *arXiv:1606.04671*.

[17] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3366–3375.

[18] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[19] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, "Encoder based lifelong learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1320–1328.

[20] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "PathNet: Evolution channels gradient descent in super neural networks," 2017, *arXiv:1701.08734*.

[21] S. Farquhar and Y. Gal, "Towards robust evaluations of continual learning," 2018, *arXiv:1805.09733*.

[22] R. Jayaram, G. Sharma, S. Tirthapura, and D. P. Woodruff, "Weighted reservoir sampling from distributed streams," in *Proc. 38th ACM SIGMOD-SIGACT-SIGAI Symp. Princ. Database Syst.*, Jun. 2019, pp. 218–235.

[23] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 532–547.

[24] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. Van De Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5513–5533, May 2023.

[25] K. Loparo. *Case Western Reserve University Bearing Data Center*. Accessed: 2022. [Online]. Available: http://csegroups.case.edu/bearingdatacenter/home

[26] E. Bechhoefer. (2013). *Condition Based Maintenance Fault Database for Testing Diagnostics and Prognostic Algorithms*. [Online]. Available: https://www.mfpt.org/fault-data-sets/

[27] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[28] K. James, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, and D. Hassabis, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[29] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 3987–3995.

[30] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," 2018, *arXiv:1810.11910*.

[31] F. Wiewel and B. Yang, "Entropy-based sample selection for online continual learning," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 1477–1481.

[32] A. Chaudhry, A. Gordo, P. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proc. AAAI Conf. Artif. Intell.*, May 202, vol. 35, no. 8, pp. 6993–7001.

[33] A. S. Benjamin, D. Rolnick, and K. Kording, "Measuring and regularizing networks in function space," 2018, *arXiv:1805.08289*.

[34] S. Lee, M. Weerakoon, J. Choi, M. Zhang, D. Wang, and M. Jeon, "CarM: Hierarchical episodic memory for continual learning," in *Proc. 59th ACM/IEEE Design Autom. Conf.*, Jul. 2022, pp. 1147–1152.

[35] R. Aggarwal, V. Sounderajah, G. Martin, D. S. W. Ting, A. Karthikesalingam, D. King, H. Ashrafian, and A. Darzi, "Diagnostic accuracy of deep learning in medical imaging: A systematic review and meta-analysis," *Npj Digit. Med.*, vol. 4, no. 1, p. 65, Apr. 2021.

[36] J. Zhang, Y. Zhao, F. Shone, Z. Li, A. F. Frangi, S. Q. Xie, and Z.-Q. Zhang, "Physics-informed deep learning for musculoskeletal modeling: Predicting muscle forces and joint kinematics from surface EMG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 484–493, 2023.

[37] R. S. Piecyk, L. Schlegel, and F. Johannes, "Predicting 3D chromatin interactions from DNA sequence using deep learning," *Comput. Struct. Biotechnol. J.*, vol. 20, pp. 3439–3448, Jan. 2022.

QILANG MIN received the B.E. degree from the Wuhan University of Science and Technology, Wuhan, China, in 2020, where he is currently pursuing the M.E. degree with the School of Computer Science and Technology. His research interests include machine learning, computer vision, and fault diagnosis.

JUAN-JUAN HE (Member, IEEE) received the Ph.D. degree in engineering from the School of Automation, Huazhong University of Science and Technology, in 2014. She was a Visiting Professor with the Department of Computer Science, Western University, for 24 months. She is currently an Associate Professor of computer science with the Wuhan University of Science and Technology. Her research interests include computational intelligence, machine learning, membrane computing, and various application domains.

PIAOYAO YU received the B.E. degree in computer science and technology from Wuhan Donghu University, Wuhan, China, in 2019, and the M.E. degree in computer technology from the Wuhan University of Science and Technology, in 2022. Her research interests include incremental learning, computer vision, and these applications in industry.

YUE FU received the B.E. degree in network engineering from Xiangtan University, Xiangtan, China, in 2020. She is currently pursuing the M.E. degree with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. Her research interests include incremental learning and fault diagnosis.