

# Research on Incremental Learning Methods Based on Sample and Category Prototype Playback

Jiamin Zhi, Yong Liu

Henan University of Science and Technology, Luoyang 471023, China.

## ABSTRACT

Machine learning technology has been successfully applied to computer vision, natural language processing and other fields, most of the existing machine learning models are deployed in actual production, because their categories and parameters are fixed, so they can only be applied to the categories that appear in the training set, and cannot incrementally learn the new categories that appear in the practical application. However, the advent of incremental learning has solved this problem. Therefore, this paper proposes an incremental learning method based on sample and category prototype replay, which aims to solve the forgetting problem in incremental learning while maintaining high accuracy and low computational complexity. Our method consists of two steps: sample playback and category prototype playback. In the sample playback stage, we record and store the historical data of the recent period, and in order to further reduce the category imbalance problem during playback, we propose a SMOTE algorithm based on global sample rationality, so that the model can learn the latest trends and changes. In the category prototype playback stage, we consider the importance of the representativeness of different samples and distinguish the important factors of different samples according to their dataset abundance, which makes it more accurate to calculate the prototype. The experimental results show that our method has good performance in both forgetting and prediction, and also has low computational complexity.

## KEYWORDS

Incremental Learning; Catastrophic Oblivion; Sample Playback; Category Prototype Playback; SMOTE Algorithm Based on Global Sample Rationality

## 1. INTRODUCTION

With the continuous growth of data and the continuous change of application scenarios, traditional machine learning methods face the problem of forgetting and scalability. While traditional machine learning methods are basically based on static datasets for training and testing[1], incremental learning can better respond to changing data and requirements by using new samples to update the model [2]. In incremental learning, models can improve prediction performance and applicability by learning on new data, adaptively updating. However, one of the main challenges of incremental learning is the catastrophic forgetting problem that arises during the learning process, i.e., the problem that updating the model with new data causes the results of previously received data to become inaccurate or no longer applicable. Solving the problem of forgetting is a core task of incremental learning [3], which has attracted extensive research interest in recent years.

The playback mechanism is a common method to solve the forgetting problem in incremental learning[4], and its main idea is to save historical data and model state for use in subsequent learning. Depending on the type of historical data used and the method of preservation, playback mechanisms are classified into three types: instance-based playback[5]feature-based playback[6], and prototype-

based playback[7]. Although these methods can solve the problem of forgetting, there are still some challenges, such as insufficient knowledge[8] and sparsity of representatives[9].

Therefore, this paper proposes an incremental learning method based on sample and class prototype replay, which can effectively solve the problem of catastrophic forgetting and maintain the high accuracy and low computational complexity of the model. The contribution of this article is to:

(1) In the sample playback, the historical data of the recent period is retained, and in order to further reduce the category imbalance problem in the playback, an SMOTE algorithm based on global rationality is proposed, so that the model can learn the latest trends and changes.

(2) In the playback of the category prototype, some representative samples are retained, and in order to measure the representativeness of different samples when calculating the prototypes, we propose a category prototype playback method based on the important factors of the abundance of the dataset. This facilitates the classification of new data so that the model can learn a wider distribution of data.

(3) The method used in this paper is evaluated through experiments, and the advantages in the prediction stage and algorithm complexity are demonstrated.

Our approach is applicable to different fields and scenarios, such as computer vision, natural language processing, and recommender systems, among others. We expect that this kind of incremental learning based on sample and category prototype replay has a wide range of application prospects in practice, which can provide more incremental learning schemes for different application scenarios and stimulate more research, so as to better adapt to practical application scenarios and solve more complex incremental learning problems.

The structure of this paper is as follows: Section 2 introduces the relevant research work; Section 3 describes our method in detail, including a description of sample and category prototype playback, an improved SMOTE algorithm based on global plausibility, and a method for measuring the representativeness importance of different samples in category prototype playback. Section 4 presents our protocol and results; Section 5 discusses the limitations and future work of this study, and finally, Section 6 concludes that summarizes the work of this paper.

## **2. RELATED WORK**

In the field of incremental learning, there has been a lot of research on forgetting. The traditional solution to the forgetting problem is to retrain the entire model, but this method requires re-training with all historical data, which is expensive to compute and store, which limits its feasibility in practical applications. Therefore, some incremental learning methods based on playback mechanism have been proposed and widely studied.

### **2.1. Playback-based incremental learning strategy**

In incremental learning, a basic assumption is that the model can't get old task data while learning a new task. However, some work has eased this restriction to allow the model to save a portion of the old category data  $D_1$ . The old category data in this section will be used together with the new category data ( $D_1D_2$ ) for the update of the current model. Among the incremental learning methods based on the replay mechanism, the common method is the playback based on the original instance, the generated instance, and the category prototype.

#### **2.1.1. Playback method based on the original instance**

Playback methods based on the original instance typically use buffers to store previous samples[10]. When a new sample is received, the model is updated with old data in the buffer, and although this approach usually solves the forgetting problem, storing all historical data has high computational and storage costs[11]. In the field of image classification, Rebuffi et al.[12] were the first to propose a

strategy to learn both classifiers and feature representations in the process of class incremental learning, called Representation-Based Learning Incremental Classifier (iCARL), which uses a small number of samples that are representative of the old class stored and samples from the new task for training[13]. However, this method retains some of the old samples and all the data of the new task for training, which has certain requirements for storage space and computing power. The category incremental learning method based on real data playback, retaining the old category data is essential to maintain the old knowledge. Therefore, it is important to consider which samples are more conducive to maintaining the knowledge that exists[14]. The Rwalk (Riemannian walk) method stores samples with high softmax output probability entropy or a small distance to the decision boundary. MIR (Maximal interfered retrieval)[15] selects samples with significantly increased losses. Wang et al.[16] proposed data replay based on data compression to reduce storage costs and increase the number of samples saved.

### 2.1.2. Playback method based on generated instances

The generative instance-based playback approach is an alternative to the memory-based approach, which replaces the memory buffer with a generation module that can reproduce relevant information about previous tasks. In general, most instance-based approaches only generate previous data instances and their corresponding labels. Shin[17] proposed a deep generative replay (DGR) method that uses a GAN model to mimic the pseudo-data generated by previous training examples to sequentially learn multiple tasks and pair them with corresponding labels. But the flaw[18] is that the effectiveness of the algorithm depends largely on the quality of the generator. Zhang et al.[19] proposed deep model consolidation (DMC), which first uses labeled data to train a separate model for the new class, and then uses publicly available unlabeled auxiliary data to combine the old and new models, overcoming the difficulties caused by the lack of access to legacy data

### 2.1.3. Playback method based on category prototypes

The category prototype-based playback method stores the prototypes in the old model and uses the prototypes to help the new samples map into the category space in the old model when new data arrives to retain information related to the original task [20]. Currently, the prototype-based playback method is one of the most widely used. Yu et al.[21] proposed that for each old class, only the category prototype in the feature space should be saved for the joint classification of the old and new categories, in order to cope with the prototype drift problem caused by the update of the feature extractor (that is, the saved prototype cannot represent the old class information after the model is updated). In Ref.[22], a prototype selection mechanism was proposed to reduce feature confusion during knowledge distillation. According to the similarity between the feature representation of the new class in the feature space and the prototype of the old class, the new class sample is divided into similar and non-similar parts. Similar samples are distilled to maintain the discriminative characteristics of the old class. Dissimilar samples are mainly involved in the classification loss used for new class learning.

Recent studies have proposed some new playback methods, such as centralized sample playback and key sample playback. The sample set-up replay method provides an alternative to storing all historical data, solving the forgetting problem by running the model on a scaled-down sample[23], and the critical sample replay method uses importance weights to select "important" historical samples to preserve historical knowledge with greater precision[24]. In addition, there are incremental learning methods that use generative models or methodological losses, which can capture the data distribution of the model and the relevant background knowledge to improve the efficiency and robustness of incremental learning[25].

### 3. INCREMENTAL LEARNING METHOD BASED ON SAMPLE AND CATEGORY PROTOTYPE PLAYBACK

Incremental learning is a key machine learning strategy that can improve the scalability and generalization performance of models, however, traditional incremental learning often forgets the knowledge of the learned tasks when faced with new tasks, and a common solution is to use replay techniques to preserve the knowledge of old tasks by retraining the model. In this paper, we propose a new incremental learning method that preserves information from old tasks based on the playback technique of sample and category prototypes. Specifically, historical trends over the most recent period are preserved through sample playback so that the model can learn the latest trends and changes. A representative sample is preserved through the playback of the category prototype so that the model can learn from a wider distribution of data.

#### 3.1. Method based on improved sample playback

In traditional incremental learning for sample replay, in order to avoid forgetting the old data, the old task data is usually stored in an empirical cache, and the replayed old data is introduced in the training of the new task, but this method leads to an imbalance in the category distribution in the replay samples. This is because in incremental learning, the number of categories for each task tends to be different, and some categories appear more frequently than others. Therefore, when replaying the old task samples in the empirical cache, it will lead to the problem of class imbalance, which makes the model more likely to favor the classes with higher occurrence frequency and ignore those that occur less frequently, which will affect the generalization performance of the model. In order to solve the problem of category imbalance in the replay samples, we propose a SMOTE algorithm based on global rationality improvement to alleviate this dilemma, and balance the dataset by synthesizing new class samples.

#### 3.2. SMOTE algorithm based on global sample rationality

The SMOTE algorithm is an oversampling method for synthesizing minority samples, which is an improved scheme based on the random oversampling method. The basic idea of the algorithm is to analyze minority samples and artificially synthesize new samples based on minority samples to add to the dataset, so as to alleviate the problems caused by data imbalance. The specific process of the algorithm is as follows: for each sample  $x$  in the minority class, the distance from it to all samples in the minority sample set is calculated using the Euclidean distance as the standard to obtain  $K$  nearest neighbors. A sampling ratio is set according to the sample imbalance ratio to determine the sampling magnification  $N$ , and for each minority sample  $x$ , a number of samples are randomly selected from its  $K$  neighbors, assuming that the selected neighbors are, and the original samples are respectively constructed with the following formula.

$$x_{i1} = x_i + \xi \times (x_{i(n)} - x_i) \quad (1)$$

Based on the perspective of global sample rationality, the improved SMOTE algorithm mainly takes into account the distribution of sample points. The synthesis operation is carried out differently according to the different positions of the samples to be synthesized in the world. When the SMOTE algorithm selects the sample points for synthesis, the sample points are judged firstly, and the position of the sample points in the whole is in the minority dense area and far away from the majority dense area, in the minority dense area and close to the majority dense area, away from the minority dense area and as far away as possible from the majority dense area, away from the minority dense area and away from the majority dense area. Synthetic operations are performed separately for each of these four cases. The synthesis operations for samples are close to this point, expanding the interpolation range, moving away from this point, and expanding the interpolation range. The key question is how to determine where the point is. The method we use is to first calculate the average distance between

minority sample points, and between minority and majority samples, and for the selected sample take, calculate the distance between it and the surrounding points, and compare it with the average distance to determine the relative position of the sample point.

### 3.3. The specific idea is as follows:

In the minority sample set T, select a sample point arbitrarily, take the K minority sample set with its nearest neighbor calculate the distance between the sample point and K minority sample points, and find the average value, and perform the calculation for each minority sample point, and find the average distance between the minority samples that the average distance of T is the average distance between minority samples.

The average distance of the sample points with respect to the nearest neighbor sample can be expressed as equation (2):

$$d_{12} = \sqrt[p]{\sum_{k=1}^n |x_{1k} - x_{2k}|^p} \quad (2)$$

where denotes the distance between and . Generally speaking, Euclidean distance, Manhattan distance, and Chebyshev distance can be used as distance measures. In this paper, the Euclidean distance is used to calculate.

The average distance of all sample points in the sample set T with respect to the nearest neighbor sample can be expressed as equation (3):

$$S_{1,2} = \frac{1}{m} \sum_{i=1}^m d(x_i) \quad (3)$$

where represents the average distance of the sample point from the nearest neighbor sample. m in the formula indicates the number of sample points contained in the minority sample set.

For the minority sample set T, select any local point, obtain K majority class sample points closest to it, calculate the distance from these K sample points, and find the average value, and perform such calculations for each minority sample point, and find the average value of this T distance as the average distance between the minority samples and the majority class samples.

The average distance between minority and majority samples can also be expressed as equation (3): in this case, the distance between the sample point and its j-th nearest neighbor majority sample point in equation (2) is described. In equation (3), the average distance between the sample points and their corresponding nearest majority sample points is described, and m is the number of sample points contained in the minority sample set.

Secondly, the average distance between the selected sample points and their nearest neighbors of K minority samples and the distance between K majority samples were calculated respectively, and the average values were recorded as and respectively.

According to the different positions of the selected sample points, the random number between 0 and 1 in the original SMOTE algorithm is modified.

If the average distance between the selected sample point and K nearest neighbor minority samples is less than the distance between minority samples, and the average distance between the sample point and K nearest neighbor majority samples is greater than the average distance between minority samples and majority samples. In this case, the coefficient  $\xi$  obtains equation (4):

$$\xi = rand(0,1) \quad (4)$$

If the average distance between the selected sample point and K nearest neighbor minority samples is less than the distance between minority samples, and the average distance between the sample point and K nearest neighbor majority samples is less than the average distance between minority samples and majority samples. In this case, the coefficient  $\xi$  obtains the equation (5):

$$\xi = \frac{1}{2} \times rand(0,1) \quad (5)$$

If the average distance between the selected sample point and K nearest neighbor minority samples is greater than the distance between minority samples, and the average distance between the sample point and K nearest neighbor majority samples is less than the average distance between minority samples and majority samples. In this case, the coefficient  $\xi$  is obtained by equation (6), where the value range is set between 0.5 and 1:

$$\xi = \vartheta + (1 - \vartheta) \times rand(0,1) \quad (6)$$

If the average distance between the selected sample point and K nearest neighbor minority samples is greater than the distance between minority samples, and the average distance between the sample point and K nearest neighbor majority samples is greater than the average distance between minority samples and majority samples. In this case, the coefficient  $\xi$  obtains equation (7):

$$\xi = \frac{1}{2} + \frac{1}{2} \times rand(0,1) \quad (7)$$

The pseudocode of the improved SMOTE algorithm based on the perspective of global sample rationality is as follows:

---

Algorithm 1:MD-SMOTE

---

Input:

- minority\_data
- majority\_data
- k\_neighbors
- rand\_seed
- N%

Output:

- Synthesized\_minority\_data
1. S1, S2= mean distance (minority\_data and k\_neighbors in majority, k\_neighbors in majority\_data)
  2. repeat
    - for each sample in minority\_data:
      - d1, d2 = mean distance (sample and nbrs\_minority, nbrs\_majority)
      - If d1 < S1 and d2 > S2:
        - $\xi$ = random number between 0 and 1
        - Synthetic\_data.append(sample+ $\xi$  × mean(nbrs\_minority - sample))
      - elif
        - if d1 < S1 and d2 < S2:
          - $\xi$  = random number between 0 and 0.5
-

---

```

Synthetic_data.append(sample+ $\xi$  × mean(nbrs_majority - sample))
elif
if d1 > S1 and d2 < S2:
     $\partial$  = random number between 0.5 and 1
     $\xi$  = random number between  $\partial$  and 1
    Synthetic_data.append(sample+ $\xi$  × mean(nbrs_majority - sample))
elif
if d1 > S1 and d2 > S2:
     $\xi$  = random number between 0.5 and 1
    Synthetic_data.append(sample+ $\xi$  × (Synthesized_minority_data[closest
point to sample] - sample))
append synthetic_data to Synthesized_minority_data
return Synthesized_minority_data

```

---

The above pseudocode briefly illustrates the specificity of the synthesis operation on a small number of samples. The constraints of global rationality are introduced into the samples to be synthesized, and the suitability of the generated new samples in the overall sample is better maintained by introducing constraints. In conclusion, the SMOTE algorithm proposed in this paper is an improvement of the traditional SMOTE algorithm based on the rationality of the global sample, which can better cope with the problems of a large amount of noise data by determining the location of the sample and specifically synthesizing.

### 3.4. A playback method based on a category prototype

In the playback method of incremental learning, class-based archetype-based playback is a common strategy that can maintain class-based archetypes to reduce the excessive influence of replay data on the mode. However, the direct calculation of the prototypes of each category will lead to the overfitting of new samples and the problem of high computational complexity, due to the small number of samples in each category, there will be uncertainty in the calculation of class prototypes, and when the samples appear outliers, the calculated class prototypes will deviate from the real prototypes. Therefore, in this paper, we use the important factor strategy of dataset abundance to calculate different category prototypes in a method based on category prototype replay.

#### 3.4.1. Category prototype playback method based on the important factor strategy of dataset abundance

Traditional class-based archetype-based playback is to apply the same attention to each sample, for example, for each class of samples, the prototype network algorithm uses the mean to calculate the archetype of that class, which essentially assigns the same weight to each sample and then weights it. However, in the actual calculation of the category prototype on the dataset, the importance of each sample is different, so this paper considers the representativeness of different samples in the world. First of all, consider that the number of samples in a fixed range of each sample is mostly different, that is, the more samples in this area, the higher the abundance of the data, which means that the more representative the sample. On the contrary, the smaller the number of samples in a fixed region, the lower the abundance of the data, which means that the sample is weaker. Based on the above strategy, a playback method of important factor class prototype based on dataset abundance is proposed.

In classification tasks, vector averaging is a common method for calculating prototypes. In this paper, the prototype of the important factor categories of the abundance of the dataset is completed on the

basis of the vector averaging method. For each category, the feature vectors of all the samples in the category can be weighted average to obtain an archetype representing the category, and the process of calculating the prototype can be calculated using equation (8):

$$P_k = \frac{1}{n_k} \times \sum_{i=1}^{n_k} x_i^{(k)} \quad (8)$$

where represents the prototype vector of the kth category, the number of samples in the kth category, the weight of the ith sample in the kth category, and the eigenvector of the ith sample in the kth category.

In this paper, the Parzen window method is used to calculate the abundance of the dataset, and different important factors are given through the abundance of the dataset.

### 3.4.2. Calculate the important factors based on the Parzen window method

In this method, the density is evaluated by establishing a kernel function model and calculating the number of samples around the sample point. Specifically, for each sample point, a window with radius  $r$  is set around the point, the number of samples in the window is calculated, and finally the density of the sample point is obtained by dividing the number of samples by the window volume. The density calculation formula for the Parzen window method is (9):

$$P_k = \frac{1}{n_k} \times (\sum_{i=1}^{n_k} w_i^{(k)} \times x_i^{(k)}) \quad (9)$$

where the sample density in a hypersphere centered on sample  $x$  and radius  $h$  is described.  $N$  represents the total number of samples in the dataset. Represents the first sample point in the dataset.  $D$  represents the sample dimension.  $H$  indicates the density of the Parzen window. For the kernel function, the Gaussian kernel function is generally selected, and its formula is (10):

$$\rho(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h^d} \varphi\left(\frac{\|x - x_i\|}{h}\right) \quad (10)$$

By controlling the size of the window width  $h$ , you can adjust the size of the window to affect the result of the density calculation. If the window  $h$  is too small, it may result in inaccurate sample point density. If the window  $h$  is too large, it may lead to a misunderstanding that sample points of different densities are divided into the same class.

where  $n$  represents the total number of samples in the dataset. The core idea of the important factor calculation formula is that the sum of the important factors of all samples is 1, and the ratio of the abundance of each sample to its total density can be used as an important factor for that sample. This calculation method makes the higher the abundance of the sample, the greater the point importance factor, which can better play the influence of the high-abundance sample on the category prototype, so as to better distinguish different categories.

The pseudocode that uses the Parzen window method to calculate sample density and important factors is as follows:

---

**Algorithm 2: VA-Prototype**

---

Input:

- dataset;
- features;
- labels;

Output:

- Prototypes;
1. for c in labels:
    - samples\_c = []
    - If label == c:
      - samples\_c = [vector]
      - Num\_samples = samples\_c
      - Prototype\_c = []
    - weights = []
  2. for sample in samples\_c:
    - calculate weight
    - weights.append(weight)
    - prototype\_c.append(sample × weight)
    - total\_weight = sum(weights)
    - Prototype\_c = prototype\_c / total\_weight
    - Prototypes.append(prototype\_c)
- 

The above pseudocode briefly describes how well different samples are represented in the global context when calculating class archetypes. By calculating the abundance of data and assigning appropriate important factors to different sample points, the influence of highly representative sample points can be emphasized, so as to improve the classification performance of the model, and improve the classification accuracy and robustness.

## 4. EXPERIMENT

In this section, we first cover the experimental setup, including the dataset and implementation details in Section 4.1. Then, in Section 4.2 we conducted extensive experiments on three standard datasets and compared our approach with relevant studies in the field of incremental learning. In section 4.3, we performed the relevant ablation experiments and gave the experimental results. The results of the experiment are described in detail in Section 4.4.

### 4.1. Experimental Setup

#### 4.1.1. Datasets

In this paper, four UCI public datasets are used for experiments. Table 1 summarizes the information on these datasets. They are IDD, GSADD MSD. In the Gas Sensor Array Drift Dataset at Different Concentrations, the sensor dataset may drift over time and have a serious impact on the performance

and reliability of the sensor, resulting in the generation of new chemicals that cannot be predicted by previous models, so when we need to add new types of substances, treat them as a new classification, Use incremental learning to continuously adjust existing models. In the Meteorological Stations dataset, new extreme weather conditions need to be treated as a new category, and the model needs to be adjusted and learned to better identify and predict the occurrence of these new weather conditions. In the Individual household electric power consumption dataset, when a new electricity usage pattern emerges, it is obvious that the previously trained model cannot meet the new needs, and the data needs to be incrementally trained to adapt to the new pattern. The above four datasets are suitable for categorical incremental learning research.

#### 4.1.2. Implementation details

We compare the proposed incremental learning method based on sample and category prototype replay with the state-of-the-art incremental learning algorithms.

In order to ensure the reproducibility and validity of the experiment, the experimental configuration and experimental environment in this paper are as follows: in the hardware configuration, the computer is equipped with 16GB RAM, 2TB solid state drive, and the processor is 12th Gen Intel(R) Core(TM) i5-12400F. In the software configuration, use Python version 3.7.0 and Jupyter Notebook version 6.0.3. The deep learning framework is Pytorch 1.13.1 and uses Pandas 1.3.0, Numpy 1.19.5, and Sklearn 0.22.0.

### 4.2. Comparison with state-of-the-art methods

In order to evaluate the performance of the incremental learning method based on sample and category prototype replay, it is compared with several popular and classical incremental learning methods in this field described in Section 4.1.2. Figure 1 shows the accuracy of the first two datasets compared to the six methods used in this paper. From the graph, we can see that the method used in this paper outperforms the performance of the comparison method in most cases in terms of accuracy.

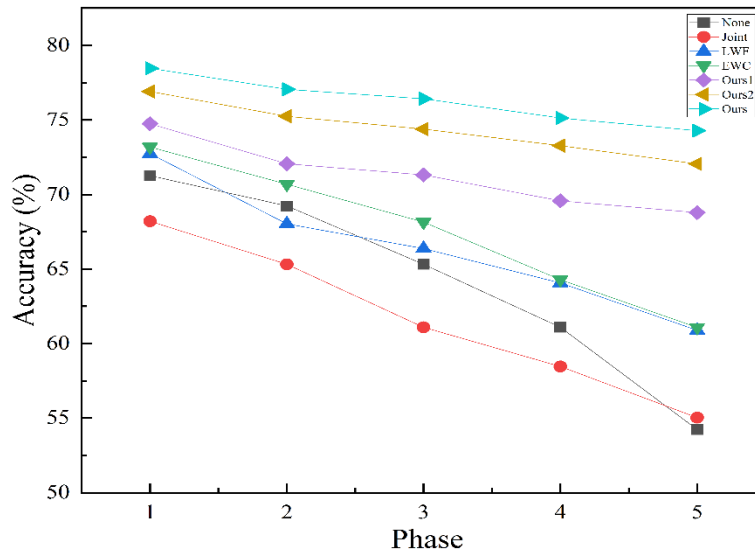


Fig. 1 The accuracy of the algorithm in each incremental stage on the IDD dataset

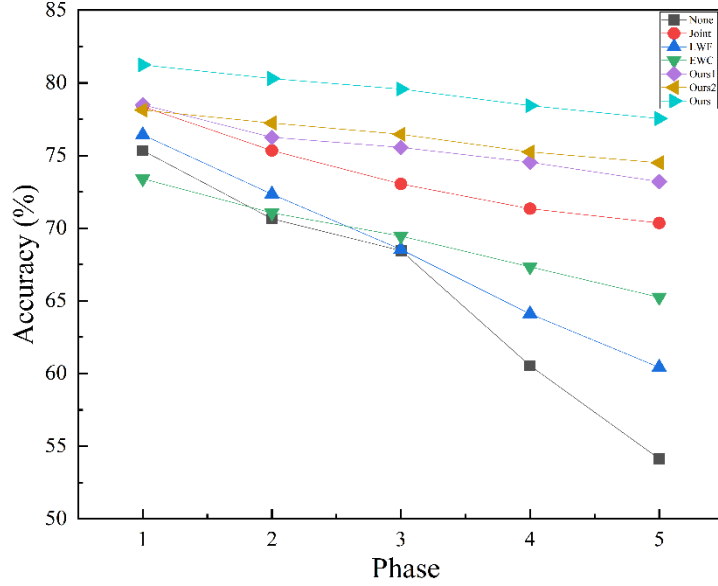


Fig. 2 The accuracy of the algorithm in each incremental stage on the GSADD dataset

The experimental results of the sample replay group, the category prototype replay group and the class-based prototype-assisted sample playback group were higher than those of other comparison algorithms, and the experimental effect of the class-based prototype-assisted sample playback group was the best, and the change trend of the accuracy in each dataset depicted the performance of the overall model with the continuous expansion of new categories.

In order to verify the effectiveness of the proposed method in the long-term incremental stage, different stages of training were carried out, and different categories were added in each incremental stage, and the incremental data sample size used by each algorithm in each stage was the same, as can be seen from Figure 2, when different important factors were applied to different samples, only the method showed good performance in the first three stages, and the average accuracy of the first two stages was 80.73%. The average accuracy of the latter two stages was 77.38%. This is a decrease of 3.35 percentage points. From the experimental results, it can be seen that the advantages of this method are weakening with the increase of categories in the dataset at different training stages, but compared with other algorithms, the adaptability of the proposed method in the face of new samples is still commendable.

The ablation experiment was set up as follows: firstly, the SMOTE algorithm based on the improved sample distribution during sample sampling was used as the method Ours1. A sample density-based class prototype replay method was used as method Ours2. The combination of the two methods is the final method of Ours. The experimental effects of four comparison algorithms, None, Joint, LWF, EWC and three component algorithms Ours1, Ours2 and Ours, were compared, and the comparison experiments were carried out in the same experimental environment.

In order to verify the forgetting degree of each algorithm in different incremental stages under different datasets, it can be seen that the average forgetting rate of evaluation indicators is the most accurate and fair when evaluating the forgetting degree of old knowledge in incremental learning. Therefore, the forgetting rates of each experimental algorithm in each incremental stage test set on the three datasets are show respectively.

Tab. 1 The forgetting rate of each incremental stage of the algorithm on the IDD dataset

Method-FOR(%)	2-phase	4-phase	5-phase	Entire-phase
None	24.375	27.564	30.872	27.503
Joint	18.364	24.578	29.315	24.085
LWF	21.035	22.340	24.958	22.787
EWC	20.255	21.349	26.541	22.715
Ours1	14.384	16.311	21.379	17.358
Ours2	17.328	18.024	24.346	19.899
Ours	13.372	15.356	20.324	16.351

Tab. 2 The forgetting rate of each incremental stage of the algorithm on the GSADD dataset

Method-FOR(%)	2-phase	4-phase	5-phase	Entire-phase
None	21.431	20.217	31.532	24.393
Joint	18.349	25.423	28.147	23.973
LWF	12.375	18.254	21.659	17.429
EWC	23.998	21.349	26.541	23.962
Ours1	17.523	19.371	23.258	20.051
Ours2	13.461	17.582	22.349	17.797
Ours	12.453	15.756	20.437	16.150

Tab. 3 The forgetting rate of each incremental stage of the algorithm on the MSD dataset

Method-FOR(%)	2-phase	4-phase	5-phase	Entire-phase
None	17.753	21.369	28.425	22.525
Joint	14.452	16.637	24.357	18.482
LWF	12.084	17.741	22.852	17.559
EWC	16.311	20.159	28.753	21.741
Ours1	13.463	17.487	19.451	16.800
Ours2	12.149	16.306	23.547	17.334
Ours	11.418	16.354	20.571	16.114

Observing the experimental results of the average forgetting rate of the above algorithms in the 2nd, 4th, 5th and the whole incremental stage of different datasets, it can be seen that the experimental effect in the sample replay group and the category prototype replay group is slightly better than that of the other four comparison algorithms, and the lowest forgetting rate in the second incremental stage is the method of combining sample and category prototype replay, and the average difference in forgetting rate is about 14 percentage points compared with that of other algorithms. In the fifth incremental stage, the forgetting rates were the most different, about 16 percentage points and 18 percentage points, respectively. From Tables 1,2 and 3, it can be seen that the method based on the

assisted sample playback of the category prototype has the best experimental effect among all methods, and the average forgetting rates in the five incremental stages of the three datasets are 16.35, 16.150 and 16.114, respectively. In the dataset IDD, the proposed method and other methods have the highest average forgetting rate of 11.152 in the whole incremental stage, and similarly, the average forgetting rate between GSADD and MSD is 8.243 and 6.411. Therefore, among the three public datasets IDD, GSADD, and MSD, the best experimental effect is the combination of sample playback and category prototype playback in the ablation experiment.

## 5. LIMITATIONS OF THIS STUDY AND FUTURE WORK

The limitation of this study is that the SMOTE algorithm is a technique that balances the dataset by synthesizing new samples, which synthesizes new samples by interpolating the minority class samples so that they are comparable to the number of majority class samples. This method effectively solves the problem of category imbalance in traditional machine learning tasks, but in incremental learning, because the incremental learning model needs to continuously adapt to new data, it needs to make incremental adjustments to the data in the empirical cache every time, and the category ratio will also change accordingly. Therefore, the number of classes in the first few rounds of training is relatively balanced, and over time, the minority class samples in the dataset will become fewer, which increases the difficulty of the SMOTE algorithm to synthesize new samples. Secondly, the influence of the important factor strategy based on the abundance of datasets on outliers is not weakened, and the classification accuracy is not high.

In incremental learning, future research work is that solving the problem of category imbalance in sample playback requires more detailed methods and operations. For example, in the replay cache, you can use Category Weighted Playback to balance the distribution of data in the empirical cache, or to reconstruct past data by keeping a prototype of the class to avoid the impact on the model if the category distribution is very unbalanced. Category-based prototype playback needs to be further improved to improve the classification accuracy in practical applications.

## 6. SUMMARY

In order to overcome the problem of sample class imbalance, this paper proposes an improved SMOTE algorithm based on global sample rationality to synthesize new samples according to the different specificities of the sample position by using a combination of sample replay and category prototype replay to overcome the sample class imbalance problem. In the replay method based on category prototypes, the degree of representation of different samples in the global is considered when calculating category prototypes. By calculating the abundance of data and assigning appropriate important factors to different sample points, the influence of highly representative sample points can be emphasized, so as to improve the classification performance and accuracy of the model. The effectiveness of the method was verified by experiments. In summary, the playback mechanism is an effective method for incremental learning to solve the forgetting problem, and the incremental learning method based on sample and category prototype playback has great prospects in practical application. However, the choice of a specific application scenario depends on the nature of the problem and the characteristics of the data, so further research and exploration are needed.

## REFERENCES

- [1] CHANG LIU, YI ZHANG, ZHONGJUN DING, et al. Active Incremental Learning for Health State Assessment of Dynamic Systems With Unknown Scenarios[J]. IEEE transactions on industrial informatics, 2023 , 19(2) : 1863-1873 .

- [2] HAN ZHOU, HONGPENG YIN, DANDAN ZHAO, et al. Incremental Learning and Conditional Drift Adaptation for Nonstationary Industrial Process Fault Diagnosis[J]. IEEE transactions on industrial informatics, 2023 , 19(4) : 5935-5944 . DOI:10.1109/TII.2022.3179423.
- [3] Kim D, Han B. On the Stability-Plasticity Dilemma of Class-Incremental Learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 20196-20204.
- [4] Smith, J. et al. Incremental Bayesian for Classifier for Streaming Data with Concept Drift[J]. Journal of Machine Learning, 2022 , 45(3) : 45-60 .
- [5] Belilovsky E, Caccia M, Lin M, et al. Online continual learning with maximally interfered retrieval[J]. vol, 2019, 32 : 11849-11860 .
- [6] Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the national academy of sciences, 2017 , 114(13) : 3521-3526 .
- [7] Smith J S, Seymour Z, Chiu H P. Incremental learning with differentiable architecture and forgetting search[C]//2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022 : 01-08 .
- [8] Yan S, Xie J, He X. Der: Dynamically expandable representation for class incremental learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 3014-3023 .
- [9] Lee A, Gomes H M, Zhang Y. Balancing the Stability-Plasticity Dilemma with Online Stability Tuning for Continual Learning[C]//2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022: 1-8.
- [10] Lopez-Paz D, Ranzato M. Gradient episodic memory for continual learning[C]//Proceedings of the Annual Conference on Neural Information Processing Systems. Long Beach, USA, 2017 : 6467-6476 .
- [11] Verwimp E, De Lange M, Tuytelaars T. Rehearsal revealed: The limits and merits of revisiting samples in continual learning[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 9385-9394 .
- [12] Shen G, Zhang S, Chen X, et al. Generative feature replay with orthogonal weight modification for continual learning[C]//2021 International Joint Conference on Neural Networks (IJCNN). IEEE, 2021 : 1-8 .
- [13] Iscen A, Zhang J, Lazebnik S, et al. Memory-efficient incremental learning through feature adaptation[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. Springer International Publishing, 2020 : 699-715 .
- [14] Pellegrini L, Graffieti G, Lomonaco V, et al. Latent replay for real-time continual learning[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020 : 10203-10209 .
- [15] Zhu F, Zhang X Y, Wang C, et al. Prototype augmentation and self-supervision for incremental learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 5871-5880 .
- [16] Yang H M, Zhang X Y, Yin F, et al. Convolutional prototype network for open set recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020 , 44(5) : 2358-2370 .
- [17] K. Zhu, w. Zhai, w. Y. Cao, B.J. Luo, and J.Z.Zha, "Self-sustaining presentation expansion for non-exemplar class-incremental learning, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, USA: IEEE, 2022, pp.92869295.
- [18] G. Yang, M. Li, Y. Li, and Y. Ke, "An Improved Incremental Learning based on Gradient Vector Field," , IEEE Access, vol. 9, pp.33955-33967,2021.
- [19] Y. Dong, Y. Zhang, and X. Wang, "A novel incremental learning frame-work based on class prototype reconstruction and transfer learning Applied Sciences, vol. 10, no. 8, p.2871,2020.
- [20] J. Bang and J. Hui, "Rainbow Memory: Continual Learning with a Memory of Diverse Samples, Journal of Computer Research and Development, vol.58, no.7, pp.1337-1348, 2021.
- [21] Z.D. Mai, and R. W. Li, "Supervised Contrastive Replay: Revisiting the Nearest Class Mean Classifier in Online Class-Incremental Continual Learning,, Journal of Computer Research and Development, vol. 58, no.4, pp.811-820,2021
- [22] W. Zhu, J. He, and Z.Y. You, "Class-incremental learning with prototype container", Journal of Huazhong University of Science and Technology (Natural Science Edition), vol. 49, no.1, pp. 103-109, 2021
- [23] X. H. Li, Y. N. Xu, F. Y Li, and Y.F. Liang, "Learning with out Forgetting," , Computational Intelligence, vol. 34, no.4, pp. 970-987,2018.
- [24] T. Y. Li, W. J. Zhao, and Y. F. Liu, "Experience Replay for Continual Learning," , Acta Electronica Sinica, vol.49,no.2, pp. 238-243, 2021.
- [25] J. Yoon, E. Yang, and J. Lee, "Lifelong Learning with Dynamically Expandable Networks, In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 6140-6147.