The Journal of Supercomputing (2022) 78:5394–5419
https://doi.org/10.1007/s11227-021-04084-w

Dynamic weighted selective ensemble learning algorithm for imbalanced data streams

Zhang Yan[1,2] · Du Hongle[1,2] · Ke Gang[3] · Zhang Lin[1,2] · Yeh-Cheng Chen[4]

Accepted: 11 September 2021 / Published online: 23 September 2021

Abstract

Data stream mining is one of the hot topics in data mining. Most existing algorithms assume that data stream with concept drift is balanced. However, in real-world, the data streams are imbalanced with concept drift. The learning algorithm will be more complex for the imbalanced data stream with concept drift. In online learning algorithm, the oversampling method is used to select a small number of samples from the previous data block through a certain strategy and add them into the current data block to amplify the current minority class. However, in this method, the number of stored samples, the method of oversampling and the weight calculation of base-classifier all affect the classification performance of ensemble classifier. This paper proposes a dynamic weighted selective ensemble (DWSE) learning algorithm for imbalanced data stream with concept drift. On the one hand, through resampling the minority samples in previous data block, the minority samples of the current data block can be amplified, and the information in the previous data block can be absorbed into building a classifier to reduce the impact of concept drift. The calculation method of information content of every sample is defined, and the resampling method and updating method of the minority samples are given in this paper. On the other hand, because of concept drift, the performance of the base-classifier will be degraded, and the decay factor is usually used to describe the performance degradation of base-classifier. However, the static decay factor cannot accurately describe the performance degradation of the base-classifier with the concept drift. The calculation method of dynamic decay factor of the base-classifier is defined in DWSE algorithm to select sub-classifiers to eliminate according to the attenuation situation, which makes the algorithm better deal with concept drift. Compared with other algorithms, the results show that the DWSE algorithm has better classification performance for majority class samples and minority samples.

Keywords Concept drift · Imbalanced data stream · Data stream mining · Ensemble learning

Ke Gang, Zhang Lin and Yeh-Cheng Chen these authors contributed equally to this work.

Extended author information available on the last page of the article

1 Introduction

In practical application, the space distribution of data sets changes with time, so it is impossible to obtain a complete training data set at one time. This requires that the classifier can constantly improve itself. The traditional batch learning algorithm [1, 2] for static scene construction model is not competent. For example, network intrusion detection, online shopping, online social networking and so on, unlimited data are coming, the method of storing data and training model is not feasible. This needs the ability of real-time processing and analysis of learning algorithm.

Stream data classification [3, 4] is an important branch of machine learning and one of the important contents of stream data mining. The goal is to build incrementally learning model based on the continuous stream data, so as to accurately classify the samples arriving at any time. Traditional batch learning algorithms, such as support vector machine algorithm [5], decision tree algorithm [6, 7], etc., need to scan the data set for many times to obtain useful information to build the classification model. After the training, even when the new data with unknown information comes, the old model can't update the original model incrementally [2, 8], but can only retrain the model. In contrast, online learning can incrementally update the model [9–13], and does not need to store training data.

In practical application, data stream can be divided into stable data stream [8, 9] and unstable data stream [10–15]. Stable data stream has the characteristics of independent and identically distributed, while the unstable data stream changes with time, the distribution of data flow will change constantly, that is, concept drift occurs [10]. The concept drift of data stream includes gradual concept drift [11], abrupt concept drift [12, 13], and recurrent concept drift [14]. Concept drift exists in most data stream, and the speed of concept drift is different, which increases the difficulty of online learning algorithm.

In addition, the data in practical application are often imbalanced, and the learning algorithm for imbalanced data is also a hot spot in the field of machine learning [15–17], such as medical diagnosis [15], fault detection [16], network intrusion detection [17], etc. The model construction method for data imbalance can be divided into data level and algorithm level method. At the data level method, we usually increase the number of minority samples or reduce the number of majority samples through a certain strategy. This method is often called oversampling, undersampling or mixed sampling.

In practical applications, there are not only concept drift, but also data stream is imbalanced. This poses a new challenge to the online learning algorithm. The online learning algorithm of imbalanced data stream with concept drift mainly solves these problems: (1) Does the minority samples in the previous data block affect the construction of the current base-classifier?. How to evaluate the impact on the current classifier?. It is necessary to determine whether which previous data block and which minority samples are selected to join the current training data set. (2) The calculation method of sample weight, how to calculate the weight of every sample?. In other words, how to calculate the sampling probability, especially for minority samples in the previous data block, the sampling probability reflects the importance of the sample. (3) The weight calculation of the base-classifier. Every base-classifier needs to calculate the corresponding weight. The size of the weight affects the selection of which base-classifiers are eliminated, which classifiers are used for the final prediction of the weighted voting. (4) For updating the weight of base-classifier, the weight of base-classifier is usually calculated by classification accuracy. However, in imbalanced data stream, the accuracy cannot accurately describe the performance of base-classifier.

This paper also proposes a dynamic weighted selection ensemble learning algorithm from the above aspects, which takes into account the accuracy of a single sub-classifier and the diversity of multiple sub-classifiers. The minority samples are oversampled to obtain a balanced training subset, and then the weight of each sub-classifier is dynamically adjusted to improve the classification performance of the ensemble classifier. According to the generalization error, the sub-classifiers with poor classification performance are eliminated to improve the performance of the final ensemble classifier.

2 Related works

2.1 Concept drifts

Machine learning algorithms for data stream rely heavily on the distribution of training data [18, 19]. Data stream is dynamic, irreversible and fast. This requires that the construction of classification model can gradually deal with the coming data stream, so that the classifier has better accuracy for new data stream. Whether the distribution of data stream has changed or not, that is, whether there is concept drift, has a great impact on the classification model, which needs to be able to dynamically detect whether there is concept drift. The most representative method to solve the concept drift problem is dynamic weighted major (DWM) [3] that is the ensemble learning algorithm. DWM maintains a certain number of base-classifiers. The ensemble model is determined by weighted voting of multiple base-classifiers. In the process of new data prediction, if the prediction is correct, the weight of the base-classifier is increased, otherwise, the weight of the base-classifier is reduced. If the weight of the base-classifier is lower than the given threshold, the base-classifier is removed and a new base-classifier is created.

In practical application, concept drift can be divided into gradual concept drift, abrupt concept drift and recurrent concept drift [20]. Online ensemble learning algorithm is a common method to solve concept drift. In the ensemble classifiers, the diversity among the base-classifiers has greatly impact on the performance of the ensemble classifier. In order to improve the diversity of the base-classifiers, diversified dynamic weighted majority(DDWM) [19] adopts two sets of ensemble systems with different diversity evaluation criteria. The base-classifier is updated and removed according to the accuracy of base-classifier. According to the global prediction accuracy, a new base-classifier is added to improve the performance of online ensemble classifier. However, concept drift changes in different ways and different algorithm is adopted to deal with the concept drift. In order to make the DWM method suitable for the recurrent concept drift, Sidhu et al. [14] improve the DWM algorithm and propose the recurring dynamic weighted majority (RDWM) algorithm. RDWM algorithm adopts two ensembles for current concept and previous concept. And the corresponding pruning strategy is used to remove redundant or old classifiers, so as to better introduce new concepts into the ensemble classifier and improve the performance of the classifier. In order to make the classifier have better classification performance in the face of gradual concept drift, Ren et al. [21] proposed the gradual resampling ensemble (GRE) algorithm. According to the distance relationship between the previous minority samples and the majority and minority samples in the current data block, the previous minority samples are oversampled, and then the training subset is formed together with the current data block. In addition, the weight of the base-classifier is calculated by the mean square error. Then the base-classifier is selected to eliminate according to the mean square error which can improve the performance of the final ensemble classifier. For gradual concept drift and abrupt concept drift, Sun et al. [11] proposed class-based ensemble for class evolution (CBCE) algorithm. CBCE algorithm uses undersampling method to construct base-classifier. Then, the weight of the base-classifier is calculated and updated according to the prior probability. This method can quickly adjust the base-classifier, so that the ensemble classifier can have a high accuracy on the new data stream.

2.2 Online ensemble learning

Ensemble learning [5–7] is to combine multiple weak learning models in order to get a stronger learning model with better performance. The basic idea of ensemble learning is that even if one weak classifier gets the wrong prediction, other weak classifiers can correct the error. In order to apply ensemble learning algorithm to online data stream learning, oza [22] improved ensemble learning algorithm to obtain the online ensemble learning algorithm. The algorithm does not need to store data. The static finite data set obeys binomial distribution. In the data stream, the number of samples will increase unlimited, which can be regarded as Poisson distribution. Online boosting and online bagging algorithms are based on this idea. The algorithm constructs a new base-classifier according to the new samples, so as to realize online ensemble learning.

Because the distribution of the actual data stream will change constantly, it is very important to determine whether the parameters λ of Poisson distribution in this method. λ will directly affect the learning effect of the algorithm on the new data stream. Many scholars have improved the calculation method of the parameter. The Poisson distribution $m \leftarrow \text{possion}(1)$ does not reflect the dynamic change of data stream. The Poisson distribution $m \leftarrow \text{possion}(\lambda)$ is used by Bifet et al. [23] to process the samples in the data stream. The dynamic adaptive window is used to test the data. Then, calculate and update the weight of the base-classifier. When Poisson distribution is used, 37% of the samples will be not selected. In order to add more samples to the training, Gomes [24] proposed adaptive random forests (ARF) algorithm.

Through calculation and experimental verification, it is considered that the Poisson distribution $m \leftarrow$ possion($\lambda = 6$) is used to process the samples in the data stream.

Zhou et al. [25] proposed the concept of selective ensemble and proved that the performance of ensemble classifier obtained by selecting partial base-classifiers is better than that by using all base-classifiers. The new question is which base-classifiers to choose for ensemble?. How to evaluate the quality of base-classifier?. The key of selective ensemble is the evaluation criterion for base-classifier, which directly affects the performance of selective ensemble learning algorithm. Common methods include the classification accuracy of base-classifiers, such as AUE and AUE2 [26] algorithms. AUE and AUE2 use mean square error to evaluate the performance of base-classifier, and mean square error is used to express the weight of base-classifier. Finally, multiple base-classifiers use weighted voting to predict. Zhou et al. [27] pointed out that the greater the difference among the base-classifiers and the better the classification accuracy of each base-classifier, the better the performance of the ensemble classifier. Subsequently, many experts and scholars [27–31] have improved the accuracy and difference of the base-classifier, which is actually to optimize the selection method of the base-classifier, so that the ensemble classifier has better generalization performance.

2.3 Imbalanced data stream

In practical applications, the data stream is usually imbalanced, such as there are a lot of normal behavior data and a small amount of intrusion behavior data in the network data stream [28]. However, the traditional online learning algorithm is designed for balanced data stream [32]. That is to say, the same Poisson distribution $m \leftarrow$ possion(1) is used for the two classes of samples, which leads to the sample is still imbalanced data after sampling. This makes the poor performance of each base-classifier. It is obvious that affects the generalization performance of the final ensemble classifier.

In order to better adapt to the classification under imbalanced data stream, Wang et al. [29] improved the online bagging algorithm and proposed OOB algorithm based on oversampling and UOB algorithm based on mixed sampling. OOB and UOB algorithm apply the undersampling and oversampling methods of block learning algorithm to online ensemble learning algorithm. Online Bagging algorithm resamples the samples in the data stream according to Poisson distribution $m \leftarrow$ possion(1). OOB algorithm resamples the samples in the data stream according to Poisson distribution $m \leftarrow$ possion(λ). For different classes of samples, λ take different values to make the sampled data, approximately balanced. The UOB algorithm uses different penalty coefficients for different class of data, which makes the sampling different. In these two methods, Poisson distribution parameters λ are calculated based on the initial training data. However, with the continuous arrival of the data stream, majority class and minority class may change constantly. It is not accurate to use static values λ to describe the imbalance degree of class samples. For this reason, Wang also improved OOB and UOB algorithms [31], and dynamically calculated Poisson distribution parameters λ according to the ratio of various sample numbers in the data stream. Subsequently, many experts and scholars continuously optimize and improve the algorithm [26, 28, 33, 34], mainly from the aspects of sample resampling in data stream, weight calculation of base-classifier, elimination strategy of base-classifier, ensemble method, etc. Literatures [29, 33] used oversampling for minority samples, [26, 28] used undersampling for majority samples, [34] used mixed sampling. Then, these methods use some evaluation metric to evaluate the performance of the base-classifier, dynamically adjust the weight of the base-classifier. Then design a certain elimination strategy according to the performance of the base-classifier, remove the old and poor base-classifier, and add a new classifier, so that the ensemble classifier can deal with the problem of concept drift.

3 Dynamic weighted selective ensemble learning algorithm

This section mainly introduces the imbalanced data stream processing mechanism, the weight calculation and update of the base-classifier, the ensemble method of the base-classifier and so on, and then gives the online learning algorithm of dynamic weighted selective ensemble for the imbalanced data stream. The algorithm uses the sliding window mechanism to segment the data. The following parts are introduced in detail.

3.1 Imbalanced data stream

The imbalanced data processing methods include data level method and algorithm level method. The data level method is to resample the data through a certain strategy. In the online learning algorithms based on data block, most of the processing methods are resampling the samples in the current data block [12, 21, 32–35]. Gao et al. [32] proposed the uncorrelated bagging (UB) algorithm based on data block. UB stores the minority samples in all previous data blocks, and is used to oversampling the minority samples in the current data block, and then the balanced data block is divided into several sub blocks. In order to ensure the diversity, majority class samples are randomly added to each sub block.

In fact, when creating a new base-classifier based on the current data block, the selection of some minority samples in the previous data block is more consistent with the original data distribution than the artificial synthesis of minority samples, which is more conducive to the equalization of the training data set [35]. Chen et al. [35] used Mahalanobis distance to calculate the distance between the minority samples in the previous data block and the minority samples in the current data block. Then, the similarity between the previous minority samples and the samples in the current data block are judged according to the distance value. Then, according to the similarity, the minority samples are selected to add into the training subset of the current data block. In the case of concept drift, minority samples in the previous data block are not necessarily good samples. In order to adapt to the learning under concept bias, Sheng et al. [36] proposed a selective recurrent approach (SERA) algorithm. The SERA algorithm selects some samples which are similar to the distribution of the current data block from the minority samples of the previous data block. In order to evaluate whether a sample is selected, Mahalanobis distance is used to calculate the distance between each previous minority sample and the majority sample in the current data block. Recursive ensemble approach (REA) [34]. REA improves the SERA algorithm and uses K-nearest neighbor to evaluate the similarity of samples, and selects samples according to the similarity.

The above method does not consider the influence of outliers and noise data. For this reason, Ren et al. [33] proposed the Gradual Resampling Ensemble (GRE) algorithm. GRE cluster the majority and minority samples in the current data block. Then, the distances between the minority sample in the previous data block and the minority cluster center and the majority cluster center are calculated, respectively. Because the number of cluster centers is less than the number of samples, the computing cost can be reduced. In order to reduce the impact of outliers and noise data, density-based spatial clustering of applications with noise (DBSCAN) [35] is used to comprehensively consider the samples that are close to the center of minority clusters and far away from the majority samples.

In the GRE algorithm, as time goes on, the storage space and the cost of computing similarity will increase. In addition, if the imbalance degree in the data stream is small, that is, the number of minority samples is large, and then the cost of storage and computation complexity of similarity will be greatly increased.

In order to solve this problem, this algorithm uses two windows (*Win1* and *Win2*) to process the data in the data stream. Window *Win1* represents the current data block to be processed, and window *Win2* stores the previous minority samples according to a certain policy. *Win1* and *Win2* can be equal or unequal in size. *Win1* and *Win2* can be fixed or dynamically changed. *Win1* and *Win2* are equal and fixed values in DWSE algorithm in this paper. How to select the previous minority samples to store in *Win2* has a great impact on the performance of the DWSE algorithm. For example, the previous minority samples can be stored in *Win2* according to the First In First Out method, or the similarity of samples can be calculated according to the method in the literature [33, 35], and the samples can be temporarily stored in *Win2* according to the similarity. In this paper, the sample selection method in *Win2* is based on the assumption that the base-classifier predicts the sample correctly, which indicates that the base-classifier has learned the information of the sample. On the contrary, it indicates that the base-classifier has not learned (or has not learned enough) the information contained in the sample. Therefore, the number of base-classifiers with misclassified for each minority sample is counted. The more the number is, the more priority will be put into *Win2*. If the number of misclassified base-classifiers of samples are equal, the samples will be stored in *Win2* in chronological order. In addition, when the samples in *Win2* are full, the sample updating strategy in *Win2* is also needed. The detailed sample selection method is shown in algorithm 1.

Algorithm 1 Selectively Resampling method

```
Input: $B_i$: the current block data
    $R_i$: the Imbalanced degree between the majority class and minority class in the current data block $B_i$.
    $M1$: the dataset containing minority samples of the current block data $B_i$.
    $M2$: the dataset containing majority samples of the current block data $B_i$.
    $a$: the number of misclassified base-classifiers. And $a$ initializes to 1.
    $Win2$: the dataset containing the selected minority samples from the previous block data.
    $P$: the probability distribution of minority samples in $Win2$.
    $C$: the classifier set containing the $K$ base-classifiers.
Output: the training dataset $T_i$
 1: If (|Win2|≥|M2|-|M1|)
 2:     for i=1:K
 3:         for j=1:|Win2|
 4:             if (f(xj) ≠$y_j$)
 5:                 Add $x_j$ into M1.
 6:             else
 7:                 if ($f_i(x_j)$≠$y_j$) then
 8:                     $a_j$=$a_j$+1;
 9:                 end if
10:             end if
11:         end for
12:     end for
13:     if (|M2|≥|M1|)
14:         if (a>0)
15:             Let's sort a in descending order. And the first |M2|-|M1| samples are taken as the M'.
16:         else
17:             Take the |M2|-|M1| samples from Win2 with the Bootstrap method as M' according to
18:             the sampling probability P.
19:         end if
20:         $T_i = M' \bigcup \mathrm{M1} \bigcup \mathrm{M2}$
21:     else
22:         $T_i = \mathrm{M1} \bigcup \mathrm{M2}$
23:     end if
24: else
25:     Take the |M2|-|M1| samples from Win2 with the Bootstrap method as M'.
26:     $T_i = M' \bigcup \mathrm{M1} \bigcup \mathrm{M2}$
27: end if
```

If the number of minority samples stored in *Win2* is greater than the difference (Line1 of Algorithm 1) between the number of majority samples and the number of minority samples in the current data block: then call the current base-classifiers to predict each sample x_i in *Win2*. The number of misclassified base-classifiers (lines 4, 5 and 6 of Algorithm 1) for each sample was counted. If the remaining samples are correctly classified by all the base-classifiers (line 14 of algorith*M1*), | *M2* |—| *M1* | samples (line 17 of algorith*M1*) are taken from the remaining minority samples with bootstrap method. Otherwise, it is arranged in descending order according to the number of misclassified basis classifiers, and the first | *M2* |—| *M1* | samples (line 15 of algorithm 1) are taken into current training dataset. If the number of misclassified base-classifiers is equal, the sample is selected based on the arrival time, and the latest arrival takes precedence. If the number of misclassified minority samples plus the number of minority samples in the current block is greater than | *M2* | (line 20 of algorith*M1*), and then the obtained training set is the union of *M1* and *M2* (line 21 of algorith*M1*). At the beginning, the number of samples in *Win2* was small. If the number of samples in *Win2* was smaller than |*M2*|—|*M1*| (line 23 of algorith*M1*), that is to add all the samples in *Win2* into the current data block, the data are still not balanced, then use bootstrap method to take |*M2*|—|*M1*| samples from *Win2* and minority samples in current data block to be add into the training set (lines 24 and 25 of algorithm 1).

3.2 Calculate the weight

The weight of base-classifier has a great influence on the final ensemble classifier. Weight also determines which base-classifier will be eliminated when new base-classifiers are added in. Therefore, weight calculation is very important for online ensemble learning. AWE algorithm [37] uses the classification accuracy of the base-classifier on the current block as the weight of the base-classifier. However, this method only considers the classification performance on current data block, and ignores the classification performance on the previous data block. In order to improve the classifier gradually, it is necessary to integrate previous data and current data blocks. AUE2 algorithm [26] is still tested on the current data block, and the weight of the base-classifier is calculated by the mean square error of the test results. And the current data block is used for incremental learning of each base-classifier, so that the previous data are also taken into account, so that the classifier has better performance.

Due to the concept drift in the data stream, the impact of historical data on the current classifier will be smaller and smaller over time. For example, in the literature [23, 32], the weight of each sample is inversely proportional to the arrival time, that is, the earlier the arrival time is, the smaller the weight of the sample is. The newly arrived samples have the largest weight, which reduces the influence of early samples on base-classifier and ensemble classifier. In reference [3, 15], decay factor $\beta(0 < \beta < 1)$ is used to represent the weight change of base-classifier. When the base-classifier misclassifies the samples in the current data block, the weight of base-classifier is set to $W_{ti} = \beta * W_{(t-1)i}$. That is to say, when concept drift occurs, the weight of early ensemble classifier is reduced. That is, the voting weight in ensemble classifier is reduced. How to determine whether the decay factor β is the key of this method. However, fixed decay factor cannot be accurately described the weight changes. In fact, when concept drift occurs, the weight change of base-classifier is related to the degree of concept drift. And the degree of concept drift can be expressed by the classification performance on the current data block. For example, in reference [26], the mean square error is used to calculate the weight of the base-classifier, which can be understood as the mean square error to describe the degree of concept drift. This dynamic method is better than static decay factor. Therefore, combining these two methods, this paper proposes a dynamic weight calculation method of base-classifier, and the weight calculation formula is shown in (1).

$$W_{ti} = (1 - E_{ti})W_{(t-1)i} \tag{1}$$

W_{ti} represents the weight of ith base-classifier after the tth update. E_{ti} represents the classification performance of the ith base-classifier on the tth block data. It can be classified error rate, weighted classification error rate, g-mean, AUC (area under curve), etc. That is to say, all the indictors describing the classification performance of the classifier can be used to calculate this parameter E_{ti}. Because the proposed algorithm is oriented to imbalanced data stream, we use the evaluation metric of classification performance for imbalanced data. In this paper, we use the weighted classification error rate to represent the classification performance of the base-classifier, which can be expressed as formula (2).

$$E_{ti} = \sum_{j=1}^{n} d_{tj} * I(h_{ti}(x_j) \neq y_j) \tag{2}$$

d_{tj} represents the weight of the jth sample in the tth data block. $I(\cdot)$ is the indicator function. If the condition is true, the function value is 1, otherwise the function value is 0. In order to reflect the difference between the minority samples and majority samples, different weights are used for the samples in two classes. Assuming that the misclassification cost of samples in two classes, the sample weight can be calculated with formula (3) as follow.

$$d_{tj} = \begin{cases} \frac{1}{2N_{maj}} & x_j \in \text{majority class} \\ \frac{1}{2N_{\min}} & x_j \in \text{minority class} \end{cases} \tag{3}$$

where N_{maj} and $N_{\min}$ represent the number of majority and minority samples in the current data block, respectively.

3.3 Tow windows

In block-based online learning, data block can be represented by window [34–36]. The size of window has a great influence on detecting concept drift [2, 29, 31, 38]. Du et al. [38] use adaptive window, which can automatically adjust the window size according to the concept drift degree in the data stream. In this paper, we use double windows. Window *Win1* represents the size of the data block, and window *Win2* is used to store the minority samples in the previous data block. The two windows size can be fixed or dynamically adjusted. The sizes of the two windows can be the same, or they can be set according to a certain proportion. However, the window size has an impact on the algorithm performance. In this paper, the two windows size are equal and fixed. The current data block is equalized with the minority samples from the previous data block. Literatures [33–36] keep all the minority samples, and then use a certain strategy to select some minority samples to equalize the current data block. As time goes on, more and more minority samples will be saved in this method, which will increase the storage space and computing time.

Therefore, the algorithm in this paper selectively keeps some minority samples and stores the selected minority samples in window *Win2*. Now the question is which minority samples to save in *Win2*? When new samples come, which samples are removed from *Win2*, and which minority samples are added into *Win2*? The sample selection method in this paper is based on two hypotheses: (1) The misclassified sample (excluding noise data and outliers) indicates that the information contained in the sample is not fully learned by the classifier and should be selected first. (2) The more base-classifiers are misclassified, the more attention should be paid to the sample and the more priority should be given to it. Therefore, the samples that are misclassified by the ensemble classifier are selected to join *Win2* first. If the ensemble classifier classifies correctly, the more base-classifiers are misclassified, the more priority will be added into *Win2*. In addition, the sample weight represents the sampling probability in *Win2*. The weight is determined by the number of misclassified base-classifiers. In summary, the update process of *Win2* is shown in algorithm 2:

Algorithm 2 Data update algorithm in Windows（DUAW）

Input: *Bm*: the minority samples in the current block data B_i;

Win2: the dataset containing the selected minority samples from the previous block data.

a: the number of misclassified base-classifiers of every minority sample in Win2.

P: the probability distribution of minority samples in *Win2*.

C: the classifier set containing the *K* base-classifiers.

h_t: current classifier on block data B_i.

Output: Updated dataset Win2 and the number of misclassification classifiers *a* and probability distribution *p*

```
Count $N$ the number of samples in win2;
If ($N$+|$Bm$|<|$Win2$|)
    $Win2 = Win2 \cup Bm$
else
    for $x_i$ in $Win2$
        If ($h_t(x_i) \neq y_i$)
            $a_i$=$a_i$+1;
        end if
        if($a_i$>$K$+5)
            remove the sample $x_i$ from $Win2$;
        end if
    end for
    for $x_i$ in $Bm$
        if($h_t(x_i) \neq y_i$)
            Replace the sample with the smallest $a$ value in $Win2$ with $x_i$;
            Set the corresponding $a$ value of this sample to $K$;
            Remove the $x_i$ from $Bm$;
        end if
    end for
    for $h_i$ in $C$
        for $x_j$ in $Bm$
            if ($h_i(x_j) \neq y_j$)
                $temp_j$=$temp_j$+1;
            end if
        end for
    end for
    for i=1|Win2|
        for j=1:|temp|
            if ($temp_j \geqslant a_i$)
                Replacing $x_i$ in $Win2$ with $x_j$ in $Bm$;
            End if
        End for
    End for
End if
Update the sampling probability $P$ using Eq.(4).
```

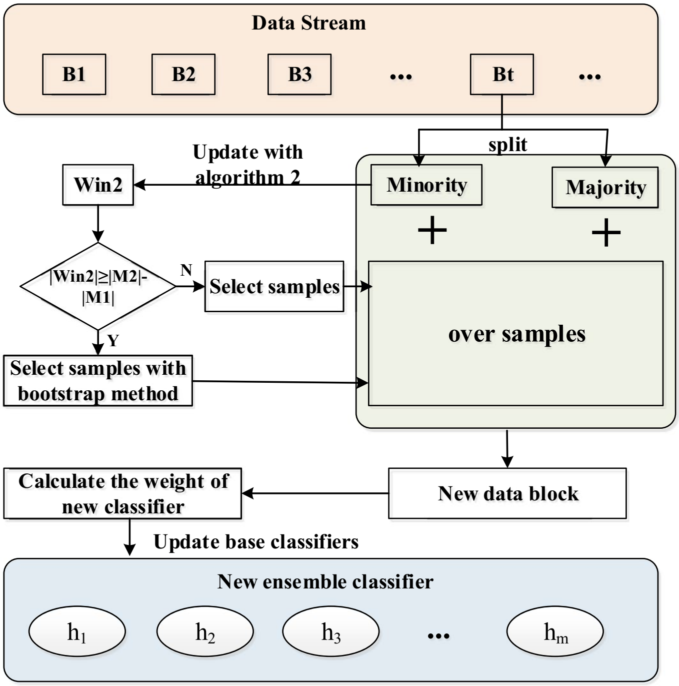

Fig. 1 Flow chart of the proposed algorithm

Minority samples in some previous data blocks are reserved for oversampling. The key of this method is to keep which samples. Algorithm 2 is used to update the samples in *Win2*. In order to prevent the continuous influence of noise data on the classifier, the corresponding samples (lines 9–11 of algorithm *2*) are deleted when the frequency of misclassified reaches the threshold. In this paper, if $K+5$ misclassified for one sample, we think that the sample is noise data. The parameters can get the best value through parameter optimization. In this paper, we have done many experiments, the effect of this value is better.

For minority samples in the current data block, if the current classifier misclassified, it will be added into *Win2* (lines 13–19 of Algorithm 2) and the number of base-classifiers that misclassified the sample is set to K (the actual number of base-classifiers is not necessarily K) (line 16 of Algorithm 2). The purpose of this is to prevent the samples with concept drift from being learned insufficiently. When selecting the samples from *Win2* to construct the balanced training set with the current data block, the sampling with replacement method is randomly sampled according to the sampling probability. The sampling probability in this method is based on the number of (lines 5–13 of Algorithm 2) misclassified classifiers. Then, the remaining samples in B_m are predicted by the previous

K base-classifiers (lines 20–26 of Algorithm 2). Then update the samples in *Win2* according to the number of the misclassification base-classifiers (lines 27–33 of Algorithm 2). Finally, each sample in *Win2* is updated (line 35 of Algorithm 2), and the sampling probability of each sample is calculated by formula (4).

$$p_i = \frac{a_i}{\sum\limits_{j=1}^{N} a_j} \tag{4}$$

where *N* is the number of samples in *Win2* and a_i is the number of base-classifiers of misclassified samples X_i.

4 Algorithm description

In algorithm 3, the pseudo code of DWSE algorithm is shown. DWSE balance the current data block by adding some minority samples from the old data block. In order to reduce space complexity and time complexity, DWSE is implemented with double windows. A window is used to store some minority samples in the previous data block, and then use these minority samples to take the current data block for equalization. See algorithm 1 for the detailed process. The algorithm 2 is the update process of stored minority samples (Fig. 1).

For first data block, because of the imbalanced data stream, if the classifier is constructed directly on the data block, the performance of the classifier will be poor. In order to reduce the influence of imbalanced data, the method in the literature [28, 30] was adopted in DWSE algorithm. The bootstrap method is used to select |*M1*| samples from *M2*, and the training subset sub_i is formed with *M1*. The base-classifier is obtained by training on sub_i. Repeat this process *K* times to get *K* base-classifiers. This can reduce the impact of initial imbalanced data, and make the initial base-classifier have better accuracy, and also ensure the diversity of base-classifiers.

Because the classification accuracy cannot accurately describe the classification performance of the classifier under the imbalanced data, the misclassified cost is used to describe the classification performance of the base-classifier. Therefore, it is necessary to calculate the weight of each sample in the current data block (line 11 of algorithm 3). If the imbalance degree of the current data block reaches the threshold, algorithm 1 is called for resampling to obtain the balanced training set B'_t of the current data block. Then we train h_t on the balanced data set B'_t. In the literature [33–36], the classifier obtained from current data block is the most important for concept drift, and the initial weight is set to 1. This method can simplify the calculation, but it is not accurate enough. In this paper, the weight of the current classifier is calculated by using the prediction accuracy of the classifier on the current data block. According to formula (1), the initial weight W_t (line 17 of algorithm 3) of h_t is calculated. For the classifier obtained on the new data block, most of the existing methods directly replace the worst performance [15, 16, 28, 36] of the existing base-classifiers. The key problem of these methods is how to evaluate the performance of base-classifier. Accuracy is the most commonly method to evaluate the classification performance. But for imbalanced data, the evaluating indicator commonly is used to evaluate the classification performance include accuracy, F-mean, G-mean, AUC, etc., which can more accurately evaluate the classification performance of imbalanced data. Therefore, this algorithm uses the weighted accuracy, that is, use the classification cost to evaluate the classification performance of the base-classifier. According to the classification cost, we decide whether to replace the base-classifier and which base-classifier to be replaced (lines 19–21 of algorithm 3). Next, calculate the weight of each base-classifier using the formula (1), and update the minority samples and the corresponding sampling probability in *Win2*. Finally, the ensemble classifier is obtained.

Table 1 Experimental dataset

Dataset	Number	Attribute	Classes	Noise	Ratio	Type
SEAs	100,000	3	2	10%	19:1	Sudden
SEA_G	100,000	3	2	10%	19:1	Gradual
Hyper	100,000	10	2	5%	19:1	Incremental
RBFb	100,000	20	2	0%	19:1	Sudden
RBFgr	100,000	20	2	0%	19:1	Gradual recurrent

Algorithm 3 Dynamic weighted selective ensemble(DWSE)

```
Input: S:data stream
    $t$:the current time stamp
    $B_t$: the current block data
    $K$:Maximum number of the base classifiers
    $Len$: the windows size of $Win1$ and $Win2$
    $W$: the weight of the base-classifiers.
    $C$: the classifier set containing the base-classifiers.
Output: the ensemble classifier $H(x) = \arg\max_{y \in Y} \sum_{i=1}^{K} W_i I(h_i(x) = y_i)$
For each block data $B_t$
    Split $B_t$ into $M1$( the minority samples) and $M2$(the majority samples);
    If (t==1)
        For i=1:K
            Select $|M1|$samples from $M2$ using the bootstrap method to get dataset
            $M2'$,set the $sub_i = M1 \bigcup M2'$;
            Obtain the $h_i$ base-classifiers on Sub$_i$;
            Calculate the $W_i$ of classifier $h_i$ on block data $B_t$ using the Eq.(1);
        End for
    else
        Calculate the weight $SW$ of every class samples in $B_t$ using the Eq.(3);
        If ($|M2|$>$|M1|$)
            Call the selective resampling algorithm in Algorithm 1 and obtain current
            training dataset $B'_t$;
        End if
        Train and obtain the new classifier $h_t$ on the $B'_t$;
        Calculate the $W_t$ of classifier $h_t$ on block data $B_t$ using the Eq.(1);
        Let $Wmin$ be the minimum of $W$
        If (Wt>$Wmin$)
            Use $h_t$ to replace the base classifier with the smallest weight;
        End if
        Update the weight of every base-classifier using Eq.(1);
        Update the $Win2$ and the probability distribution of minority samples in $Win2$
        using Algorithm 2;
    End if

    Obtain the final ensemble classifier: $H(x) = \arg\max_{y \in Y} \sum_{i=1}^{K} W_i I(h_i(x) = y_i)$
End for
```

According to the probability distribution (That is, the sampling probability.) of the sample, sample is selected from *Win2* every time. The samples in *Win2* are constantly changing, and the sampling probability is also constantly changing. At the same time, the resampling process has certain randomness, even if the samples in *Win2* remain unchanged, the samples selected each time are not the same. Therefore, when a new data block arrives, selecting samples from *Win2* can ensure that there is a big difference in each selected sample. In this way, the difference degree of constructing classifiers can be improved, and the performance of ensemble classifiers can be improved.

5 Experimental results and analysis

In this part, we mainly verify the effectiveness of the proposed algorithm. On five datasets, we compare the performance of DWSE algorithm and other five algorithms on imbalanced data streams with concept drift.

Table 2 Experimental results of different windows (accuracy)

Dataset	200	500	800	1000	1200	1500
SEAs	0.8946 ± 0.0292	0.8992 ± 0.0236	0.9088 ± 0.0212	**0.9117 ± 0.0346**	0.9104 ± 0.0312	0.9184 ± 0.0284
SEA$_{G}$	0.8714 ± 0.0310	0.8769 ± 0.0284	0.8835 ± 0.0261	0.8891 ± 0.0328	**0.8921 ± 0.0273**	0.8843 ± 0.0246
Hyper	0.9361 ± 0.0434	0.9397 ± 0.0382	0.9381 ± 0.0417	0.9275 ± 0.0313	**0.9431 ± 0.0364**	0.9384 ± 0.0372
RBFb	0.9076 ± 0.0439	0.9145 ± 0.0412	**0.9276 ± 0.0386**	0.9241 ± 0.0141	0.9187 ± 0.0349	0.9207 ± 0.0321
RBFgr	0.9116 ± 0.0386	0.9172 ± 0.0364	0.9205 ± 0.0371	**0.9678 ± 0.0108**	0.9216 ± 0.0244	0.9166 ± 0.0281

The results in bold are the best results

Table 3 Experimental results of different windows (F-mean)

Dataset	200	500	800	1000	1200	1500
SEAs	0.4875 ± 0.0291	0.4931 ± 0.0214	**0.5081 ± 0.0214**	0.5013 ± 0.0166	0.4968 ± 0.0185	0.4915 ± 0.0189
SEA$_{G}$	0.4513 ± 0.0237	0.4548 ± 0.0193	0.4594 ± 0.0242	**0.4627 ± 0.0192**	0.4573 ± 0.0187	0.4549 ± 0.0265
Hyper	0.6078 ± 0.0384	0.6097 ± 0.0226	0.6127 ± 0.0304	**0.6194 ± 0.0197**	0.6107 ± 0.0231	0.6124 ± 0.0282
RBFb	0.8106 ± 0.0401	0.8176 ± 0.0387	0.8189 ± 0.0354	**0.8273 ± 0.0249**	0.8216 ± 0.0116	0.8314 ± 0.0327
RBFgr	0.8285 ± 0.0328	0.8314 ± 0.0286	0.8361 ± 0.0261	0.8411 ± 0.0118	**0.8428 ± 0.0217**	0.8348 ± 0.0184

The results in bold are the best results

Table 4 Experimental results of different windows (G-mean)

Dataset	200	500	800	1000	1200	1500
SEAs	0.8014 ± 0.0108	0.8063 ± 0.0174	0.8142 ± 0.0168	**0.8198 ± 0.0147**	0.8135 ± 0.0158	0.8167 ± 0.0171
SEA$_{G}$	0.7943 ± 0.0237	0.7981 ± 0.0239	0.8069 ± 0.0237	**0.8195 ± 0.0208**	0.8037 ± 0.0268	0.8011 ± 0.0237
Hyper	0.9016 ± 0.0489	0.9028 ± 0.0415	0.9051 ± 0.0391	0.9074 ± 0.0181	**0.9098 ± 0.0207**	0.9043 ± 0.0214
RBFb	0.8914 ± 0.0431	0.8937 ± 0.0251	0.8985 ± 0.0274	0.9007 ± 0.0241	0.8976 ± 0.0244	**0.9015 ± 0.0278**
RBFgr	0.8995 ± 0.0372	0.9084 ± 0.0316	0.9131 ± 0.0324	**0.9118 ± 0.0185**	0.9108 ± 0.0264	0.9096 ± 0.0233

The results in bold are the best results

Table 5 Experimental results of different windows (AUC)

Dataset	200	500	800	1000	1200	1500
SEAs	0.8619 ± 0.0174	0.8668 ± 0.0194	**0.8746 ± 0.0214**	0.8716 ± 0.0135	0.8709 ± 0.0201	0.8651 ± 0.0347
SEA$_G$	0.8796 ± 0.0238	0.8824 ± 0.0251	0.8897 ± 0.0288	**0.8947 ± 0.0337**	0.8873 ± 0.0417	0.8818 ± 0.0492
Hyper	0.9366 ± 0.0519	0.9417 ± 0.0483	0.9503 ± 0.0428	**0.9544 ± 0.0335**	0.9357 ± 0.0398	0.9302 ± 0.0439
RBFb	0.8811 ± 0.0451	0.8856 ± 0.0408	0.8841 ± 0.0385	0.9079 ± 0.0282	**0.9101 ± 0.0293**	0.8945 ± 0.0315
RBFgr	0.9385 ± 0.0382	0.9421 ± 0.0217	0.9591 ± 0.0357	**0.9787 ± 0.0139**	0.9496 ± 0.0278	0.9427 ± 0.0329

The results in bold are the best results

5.1 Synthetic dataset

This part mainly introduces the data set used in the experiment. Detailed experimental results on five datasets with six algorithms are shown in Table 1. Because DWSE algorithm is oriented to the classification of two classes of problems, all data sets are of two classes. The following describes each data set of data set.

Massive Online Analysis (MOA) [39] is an open source framework for data stream mining. This framework provides a variety of data stream generators and online learning algorithms. The artificial data used in this experiment are generated by MOA.

SEA data: A set of data sets with sudden concept drift generated by SEA generator [13]. It includes three attributes, in which one attribute is redundant. Two other attributes determine the category of the sample. The dataset consists of 100,000 samples, and the dataset contains 10 percent of the noise data. The SEAs contains five mutation concept drift, and SEA$_G$ contains 10 gradual concept drift. Each 100 samples contain 5 minority samples, that is, the data flow is imbalanced.

Hyper data: Hyperplane is also a common generator in data flow learning. It is an incremental concept drift to change hyperplane smoothly by adjusting direction and position. Hyper data includes 100,000 samples, each sample includes 10 attributes, and the weight change of each sample is 0.1. At the same time, the data set contains 5% noise data, and each 100 samples contains 5 minority samples, in the other word, the imbalance rate of the two classes of samples is 19:1.

RBF data: the random Radial Basis Function (RBF) generator generates two data sets RBF$_b$ and RBF$_{gr}$. The data set contains 20 attributes, and RBF$_b$ contains 4 mutation concepts, RBF$_{gr}$ contains four gradual recurrent drifts. The same data set contains 5 minority samples in every 100 samples.

5.2 Experiment settings

Experiment results on five state-of-the-art block-based algorithms are used to compare the performance with DWSE. All the experiments were completed on the MOA framework [39]. The five algorithms used for comparison are described as follows.

SERA [36], The SERA algorithm selects some samples which are similar to the distribution of the current data block from the minority samples of the previous data

Table 6 Comparison of experimental results

Dataset	Algorithm	Evaluation Indicators (%)				
		Accuracy	AUC	F-mean	G-mean	Recall
SEAs	AWE	**0.9489 ± 0.0093**	0.8704 ± 0.0121	0.0801 ± 0.0738	0.1685 ± 0.0689	0.0461 ± 0.0706
	SERA	0.9085 ± 0.0378	0.7009 ± 0.0687	0.2728 ± 0.0516	0.4962 ± 0.0501	0.2936 ± 0.0556
	MuSeRA	0.9347 ± 0.0217	0.8533 ± 0.0226	0.3171 ± 0.0210	0.5356 ± 0.0412	0.3203 ± 0.0517
	UB	0.1690 ± 0.0728	0.8524 ± 0.0587	0.1227 ± 0.0449	0.2311 ± 0.0595	**0.9737 ± 0.0113**
	OOB	0.8860 ± 0.0608	0.8591 ± 0.0319	0.4281 ± 0.0175	0.8048 ± 0.0248	0.7244 ± 0.0263
	DWSE	0.9117 ± 0.0346	**0.8716 ± 0.0135**	**0.5013 ± 0.0166**	**0.8198 ± 0.0147**	0.7377 ± 0.0213
SEA_G	AWE	**0.9491 ± 0.0105**	0.8851 ± 0.0347	0.0415 ± 0.0702	0.0924 ± 0.0641	0.0318 ± 0.0464
	SERA	0.9255 ± 0.0331	0.7118 ± 0.0658	0.2614 ± 0.0537	0.5017 ± 0.0526	0.2949 ± 0.0521
	MuSeRA	0.9283 ± 0.0227	0.8724 ± 0.0481	0.2786 ± 0.0254	0.5241 ± 0.0418	0.3011 ± 0.0476
	UB	0.2753 ± 0.0465	0.8786 ± 0.0414	0.1437 ± 0.0575	0.3595 ± 0.0574	**0.9677 ± 0.0143**
	OOB	0.8607 ± 0.0512	0.8871 ± 0.0257	0.3782 ± 0.0245	0.8073 ± 0.0243	0.8269 ± 0.0275
	DWSE	0.8891 ± 0.0328	**0.8947 ± 0.0337**	**0.4627 ± 0.0192**	**0.8195 ± 0.0208**	0.7811 ± 0.0317
Hyper	AWE	**0.9548 ± 0.0084**	0.9085 ± 0.0218	0.2011 ± 0.0724	0.3163 ± 0.0411	0.1232 ± 0.0664
	SERA	0.9287 ± 0.0326	0.7272 ± 0.0535	0.2355 ± 0.0546	0.5669 ± 0.0358	0.3668 ± 0.0482
	MuSeRA	0.9374 ± 0.0231	0.8471 ± 0.0252	0.3976 ± 0.0379	0.6197 ± 0.0194	0.4179 ± 0.0489
	UB	0.8810 ± 0.0725	0.9438 ± 0.0351	0.4764 ± 0.0471	0.8973 ± 0.0216	0.9361 ± 0.0148
	OOB	0.9183 ± 0.0334	**0.9652 ± 0.0145**	0.5996 ± 0.0284	0.8991 ± 0.0250	0.8835 ± 0.0244
	DWSE	0.9275 ± 0.0313	0.9544 ± 0.0335	**0.6194 ± 0.0197**	**0.9074 ± 0.0181**	**0.9416 ± 0.0289**
RBFb	AWE	**0.9316 ± 0.0159**	0.8873 ± 0.0228	0.5143 ± 0.0712	0.5837 ± 0.0614	0.4087 ± 0.0623
	SERA	0.8975 ± 0.0258	0.6927 ± 0.0448	0.3688 ± 0.0507	0.4977 ± 0.0671	0.2845 ± 0.0522
	MuSeRA	0.9218 ± 0.0215	0.7816 ± 0.0341	0.1123 ± 0.0303	0.2098 ± 0.0395	0.0809 ± 0.0542
	UB	0.6745 ± 0.0653	0.8998 ± 0.0559	0.2764 ± 0.0342	0.8075 ± 0.0515	**0.9115 ± 0.0137**
	OOB	0.8659 ± 0.0229	**0.9485 ± 0.0311**	0.8198 ± 0.0235	0.8938 ± 0.0255	0.8744 ± 0.0276
	DWSE	0.9241 ± 0.0141	0.9079 ± 0.0282	**0.8273 ± 0.0249**	**0.9007 ± 0.0241**	0.9077 ± 0.0281
RBFgr	AWE	**0.9687 ± 0.0081**	0.9092 ± 0.0285	0.5377 ± 0.0694	0.5947 ± 0.0513	0.4188 ± 0.0708
	SERA	0.9219 ± 0.0368	0.7001 ± 0.0561	0.3821 ± 0.0205	0.5022 ± 0.0437	0.2953 ± 0.0596
	MuSeRA	0.9533 ± 0.0262	0.7914 ± 0.0309	0.1041 ± 0.0219	0.2139 ± 0.0332	0.0841 ± 0.0339
	UB	0.7018 ± 0.0749	0.90434 ± 0.0142	0.2836 ± 0.0518	0.8168 ± 0.0554	**0.9212 ± 0.0079**
	OOB	0.9632 ± 0.0125	0.9704 ± 0.0206	0.8284 ± 0.0224	0.9028 ± 0.0203	0.8824 ± 0.0297
	DWSE	0.9678 ± 0.0108	**0.9787 ± 0.0139**	**0.8411 ± 0.0118**	**0.9118 ± 0.0185**	0.9014 ± 0.0251

The results in bold are the best results

block. The distance between each minority sample in previous data block and the samples in the current data block is calculated with the Mahalanobis distance to evaluate whether the sample is used to balance the current data block. The parameter *f* is set to 0.5. It includes 11 base-classifiers (K = 11).

AWE [37], AWE is a block-based online learning algorithm for concept drift. Each base-classifier is based on a data block, and the ensemble classifier includes 11 base-classifiers.

MuSeRA[40], MuSeRA algorithm is an online integrated learning algorithm for unbalanced data flow with concept drift. Compared with sera algorithm, MuSeRA algorithm uses the information in previous data blocks, that is, uses the data information in previous data blocks to construct each base-classifier. The parameter f is set to 0.5.

OOB [31], OOB uses oversampling and attenuation function to solve concept drift in imbalanced data stream. The number of base-classifiers is 11($K=11$). And the decay factor is equal to 0.9.

UB [32], UB algorithm adds all the previous minority samples to the current data block. He thinks that the data distribution of all previous minority samples is similar to that of the current data block. The ensemble members is 11($K=11$).

In order to make the experimental results more comparable, VFDT is used to construct the base-classifier [33], and all parameters of VFDT adopt the default values in MOA. The data block size of all algorithms is 1000. In our method, the Win2 of minority samples is also 1000. In the other words, parameter Len is set to 1000. The number of base-classifiers is set to $K=11$.

The data stream is divided into several equal size data blocks in the experiment, and then half of the data in each data block is used for model training, and the other half is used for testing.

In this section, we will analyze the influence of the window size and the number of base-classifier on the classification performance of the algorithm. In AWE and MuSeRA algorithm based on the block learning, the size of the window has a great influence on the classification performance. If the window size is too large and there is concept drift in the data stream, the concept drift may be in one window, which is not conducive to the performance of the base-classifier. If the window is too small to lead to the amount of information included in each window is insufficient. This leads to the performance of the base-classifier from every data block is poor. This

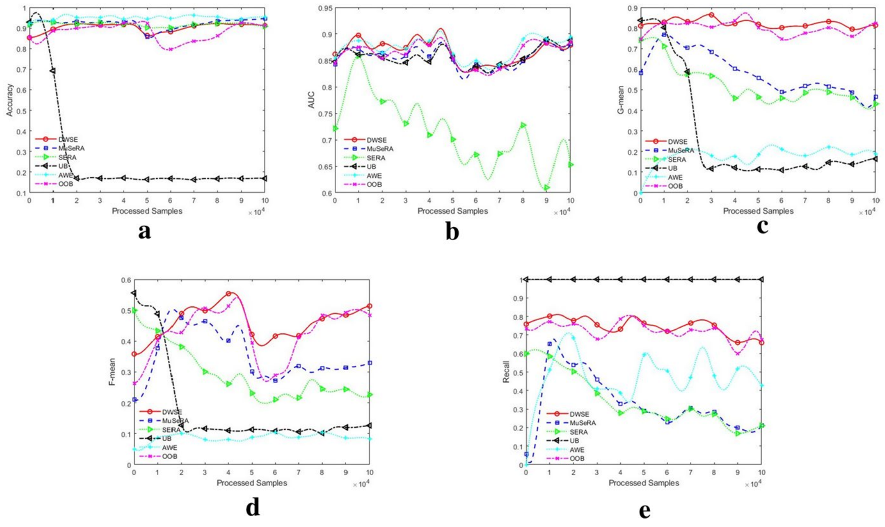

Fig. 2 Six algorithms comparison with SEA$_b$. **a** Accuracy, **b** AUC, **c** G-mean, **d** F-measure, and **e** Recall

affects the performance of the final ensemble classifier, especially when the number of minority samples is small.

DWSE algorithm uses some minority samples from the previous data block to balance the current data block, and constructs a balanced training subset. Therefore, the size of the window has little effect on the performance of the algorithm even under imbalanced data stream. Tables 2, 3, 4, 5 show the experimental results of accuracy, AUC, G-mean, and F-mean when the window of DWSE algorithm is 200, 500, 800, 1000, 1200 and 1500, respectively. Due to the randomness of DWSE algorithm, the average of 20 independent experiments is used in all experimental results. The data in the table are mean and standard deviation. The best experimental result is highlighted with bold-face type in Tables 2, 3, 4, 5. In Tables 2, 3, 4, 5, detailed experimental results are shown. From detailed experimental results of every evaluation indicator, it can be seen that the difference of values in different windows is small in each data set. That is to say, the experimental results of each index do not completely depend on the size of the window. Considering the experimental results of each index, it is found that when the window value is 1000, the experimental results of each index are relatively good, so in the later experiments of this paper, the window value is 1000.

5.3 Comparative with other algorithms

This section compares the experimental results of DWSE algorithm and five algorithms on five datasets. Since the data stream is imbalanced, this section compares the performance of the six algorithms on five evaluation indicators that is used to measure the classification performance of the algorithms under imbalanced data. Those evaluation indicators consist of accuracy, f-mean, g-mean, recall and AUC. Among them, the algorithms SERA, MuSeRA and UB are the learning methods of imbalanced data stream based on blocks. AWE and OOB are data stream oriented online learning algorithms. In Table 6, detailed experimental results are shown.

Table 6 shows average and standard deviation of each evaluation metric with the various algorithms on all datasets. Due to the good classification performance of majority samples, AWE algorithm has better mean and standard deviation on accuracy. However, the performance of AWE algorithm on recall, F-mean and G-mean is very poor, because AWE algorithm does not consider the impact of imbalanced data stream. For example, in extreme cases, AWE algorithm regards all samples as majority samples, and the accuracy is 95%.

In all artificial datasets, UB algorithm has the best performance in recall index. This is because UB algorithm equalizes the current data block by oversampling minority samples in all previous data blocks. With the continuous arrival of data stream, the number of all minority samples in previous data block will exceed the number of majority samples in the current data block. After oversampling, the original minority samples become new majority class samples. The recall of minority class is calculated after oversampling, so UB algorithm has the best recall on all data sets. However, the performance of these indexes accuracy, F-mean, G-mean and AUC are very poor. DWSE algorithm selectively retains

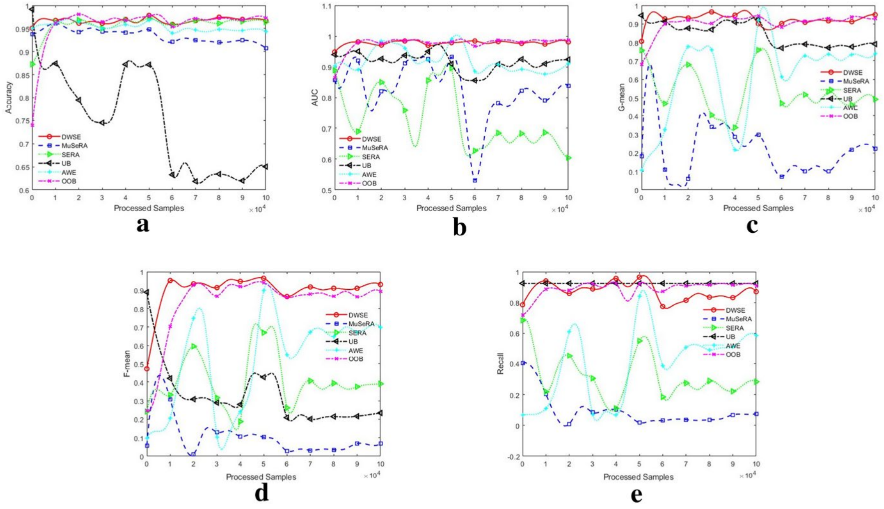

Fig. 3 Six algorithms comparison with $\mathrm{RBF}_{\mathrm{GR}}$. **a** Accuracy, **b** AUC, **c** G-mean, **d** F-measure, and **e** Recall

some valuable minority samples in the previous data block, and the number of selected sample is the difference between the number of majority samples and the number of minority samples in the current data block, which means that the training data set constructed after oversampling is balanced.

In MuSeRA algorithm, the classifiers obtained from all data blocks are stored, that is, the information in all data streams is retained. In contrast, in SERA algorithm, all the classifiers are constructed on the current data block. Therefore, compared with the SERA algorithm, the MuSeRA algorithm has better performance in accuracy due to its higher accuracy for majority class samples. Because MuSeRA algorithm selects good performance sub-classifiers from all sub-classifiers to integrate each time, the performance of MuSeRA algorithm will be better in the case of recurrent concept drift.

Through oversampling and attenuation factor, OOB algorithm has better performance on AUC, F-mean, G-mean and recall, but it improves the performance of minority classification at the expense of accuracy.

DWSE algorithm is a balance in the index of accuracy, AUC, F-mean, G-mean and recall. On the performance indicators AUC, F-mean and G-mean, the best performance index of DWSE algorithm is obtained. At the same time, the two performance indicators of accuracy and recall are kept at the second or third level. DWSE algorithm selectively selects valuable minority samples from historical data blocks for oversampling, taking into account the imbalance of data stream and concept drift. Therefore, the sub-classifier has high classification accuracy for minority samples. In addition, DWSE algorithm constantly updates the base-classifiers in order to make the ensemble classifier can deal with the concept drift. That is to say, DWSE can improve the classification performance of minority samples without affecting the classification performance of majority samples.

Figure 2 is the experimental result on the data set SEA. We can get these conclusions from Fig. 2. First, the performance of most algorithms declines when the abrupt concept drift occurs. Secondly, with the arrival of data stream, SERA algorithm has little change in accuracy, while other algorithms fluctuate with the concept drift accuracy. This is because the SERA algorithm oversamples the current data block with minority samples from the previous data block, without considering the knowledge of majority samples from the previous data block. Third, compared with other algorithms, AWE algorithm keeps a good level in accuracy. This is because AWE algorithm keeps high classification accuracy for majority samples. AWE algorithm improves accuracy at the expense of recall, G-mean and F-mean. The reason for this phenomenon is that AWE algorithm does not consider the impact of imbalanced data stream. UB algorithm has the best performance in recall indictor, but it is poor in other evaluation indictors, especially in accuracy. Compared with other algorithms, DWSE algorithm has good performance on F-mean and G-mean. The DWSE algorithm can recover quickly and maintain a good level when the abrupt concept drift occurs. At the same time, DWSE algorithm keeps the second level in AUC and the third level in recall.

Similar results can be obtained on the data set $\mathrm{SEA_G}$ with gradual concept drift. AWE algorithm has the best accuracy because it has good classification ability for majority class samples without considering the impact of imbalanced data stream. Therefore, the performance of AWE algorithm on F-mean, G-mean and recall is very poor. The performance of OOB algorithm in each index is average, which is in the middle level. AUC, F-mean and G-mean of DWSE algorithm is the best performance in six algorithms, and recall is in the third level on six algorithms. This is because DWSE algorithm through resampling, so that the data set that is used to obtain each base-classifier is balanced, which fully considering the knowledge contained in the historical data. So the overall performance of DWSE algorithm is optimal.

For incremental drift data Hyper, DWSE algorithm has the best performance on F-mean, G-mean and recall, while accuracy and AUC are the second among all six algorithms. The same as the previous performance, UB algorithm is the best in recall, but its performance is poor in other evaluation indexes. AWE algorithm has the best accuracy, but it has poor performance in the evaluation indicators AUC, F-mean and G-mean.

On the data set $\mathrm{RBF_b}$, the experimental performance is very similar to the previous test results. DWSE algorithm has the best performance in the evaluation indicators F-mean and G-mean, and the second in the evaluation indicators accuracy, AUC and recall. In terms of accuracy, AWE algorithm is still the best, while UB algorithm is the best in recall. The reason has been analyzed before.

Figure 3 is the experimental result on the data set $\mathrm{RBF_{GR}}$. In the data set $\mathrm{RBF_{GR}}$, there is a recurrent concept drift. The algorithm can improve the performance of the current classifier by introducing the knowledge from the previous data blocks. Because Ub and SERA algorithms build classifiers based on current data blocks, the performance of these two algorithms is poor in the case of recurrent concept drift. This is because they don’t make full use of historical data. UB algorithm is still the best in the evaluation index recall. DWSE algorithm performs best in AUC, F-mean and G-mean, and is still in the second level in accuracy and recall. This is because DWSE algorithm retains a large amount of information samples, so that the current base-classifier can learn the knowledge contained in the historical block. At the same time, DWSE algorithm avoids the impact of imbalanced data stream by resampling.

6 Conclusion

In practical application, there are two kinds of problems in data stream, concept drift and data stream imbalance. Many research results have been made in the research on these two kinds of problems. But at the same time, the research on these two kinds of problems has attracted the attention of experts and scholars in recent years. Combined with the existing research, this paper proposes DWSE algorithm. The main contribution of the algorithm is in these three aspects. (1) Two windows are used, one is used to keep minority samples with rich knowledge, and another is used to represent the size of the current data block. The resampling algorithm of the minority samples in the previous data block is given, and updating method of minority samples in the window Win2 is given. (2) In view of the deficiency of static attenuation factor, a dynamic calculation method of attenuation factor is proposed, which can more accurately calculate the change of base-classifier and update base-classifier and ensemble classifier. (3) Weighted accuracy is used to measure the classification performance of base-classifier, which can reduce the impact of imbalanced data stream on the algorithm. But this algorithm is for binary classification problems, how to expand the multi class problem will be the main work in the next stage.

Acknowledgements This work is supported by the Natural Science Foundation Research Project of Shaanxi Province of China (No.2020KR*M1*56); Shaanxi Provincial Education Department Scientific Research Program Foundation of China (No.15JK1218); Science and Technology Plan Project of Shangluo City of China (No. SK2019-84); Science and Technology Research Project of Shangluo University of China (No.17SKY003). Science and Technology Innovation Team Building Project of Shangluo University of China (No.18SCX002).

References

1. Sun Z, Song Q, Zhu X, Sun H, Xu B, Zhou Y (2015) A novel ensemble method for classifying imbalanced data. Pattern Recogn 48(5):1623–1637
2. Tingting Z, Yang G, Junwu Z (2020) Survey of online learning algorithms for streaming data classification. Ruan Jian Xue Bao J Softw 31(4):912–993. https://doi.org/10.13328/j.cnki.jos.005916 (**in Chinese with English abstract**)
3. Kolter JZ, Maloof MA (2007) Dynamic weighted majority: an ensemble method for drifting concepts. J Mach Learn Res 8(12):2755–2790
4. Luong AV, Nguyen TT, Liew AW, Wang S (2020) Heterogeneous ensemble selection for evolving data streams-sciencedirect. Pattern Recognit. https://doi.org/10.1016/j.patcog.2020.107743
5. Aburomman AA, Reaz M (2016) A novel SVM-KNN-PSO ensemble method for intrusion detection system. Appl Soft Comput 38(C):360–372
6. Hido S, Kashima H (2008) Roughly Balanced Bagging for Imbalanced Data. Proceedings of the SIAM International Conference on Data Mining, SDM 2008, April 24–26, 2008, Atlanta, Georgia, USA
7. Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A (2010) Rusboost: a hybrid approach to alleviating class imbalance. IEEE Trans Syst Man Cybern Part A Syst Hum 40(1):185–197

8. Wang B, Pineau J (2016) Online bagging and boosting for imbalanced data streams. IEEE Trans Knowl Data Eng 28:3353–3366
9. Ferdowsi Z, Ghani R, Settimi R (2013) Online Active Learning with Imbalanced Classes. IEEE International Conference on Data Mining. IEEE.1, pp 1043–1048
10. Wang S, Minku LL, Xin Y (2013) A learning framework for online class imbalance learning. 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), 2013: 36–45, https://doi.org/10.1109/CIEL.2013.6613138
11. Yu S, Ke T, Minku LL, Wang S, Xin Y (2016) Online ensemble learning of data streams with gradually evolved classes. IEEE Trans Knowl Data Eng 28(6):1532–1545
12. Ke W, Edwards A, Wei F, Jing G, Zhang K (2014) Classifying imbalanced data streams via dynamic feature group weighting with importance sampling. International Conference on Data Mining. SIAM International Conference on Data Mining. 2014 Apr; 2014:722–730. https://doi.org/10.1137/1.9781611973440.83
13. Street WN, Kim Y (2001) A streaming ensemble algorithm (sea) for large-scale classification. Proc Seventh ACM SIGKDD Int Conf Knowl Discov Data Min ACM 2001:377–382
14. Sidhu P, Bhatia M (2019) A two ensemble system to handle concept drifting data streams: recurring dynamic weighted majority. Int J Mach Learn Cybern 10(3):563–578
15. Dhaliwal P, Kumar A, Chaudhary P (2020) An approach for concept drifting streams: early dynamic weighted majority. Proc Comput Sci 167:2653–2661
16. Brzezinski D, Stefanowski J (2014) Combining block-based and online methods in learning ensembles from concept drifting data streams. Inf Sci 265:50–67
17. Sidhu P, Bhatia M (2015) A novel online ensemble approach to handle concept drifting data streams: diversified dynamic weighted majority. Int J Mach Learn Cyber 9:37–61. https://doi.org/10.1007/s13042-015-0333-x
18. Kamala VR, MaryGladence L (2015) An optimal approach for social data analysis in Big Data. In 2015 international conference on computation of power, energy, information and communication (ICCPEIC), pp. 0205–0208. IEEE
19. Shan J, Zhang H, Liu W, Liu Q (2018) Online active learning ensemble framework for drifted data streams. IEEE Trans Neural Netw Learn Syst 30:486–498
20. Yang Z, Al-Dahidi S, Baraldi P et al (2020) A novel concept drift detection method for incremental learning in nonstationary environments[J]. IEEE Trans Neural Netw Learn Syst 31(1):309–320
21. He H, Yang B, Garcia EA, Li S (2008) ADASYN: Adaptive synthetic sampling approach for imbalanced learning. Neural Networks, 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328, https://doi.org/10.1109/IJCNN.2008.4633969
22. Oza NC, Russell S (2005) Online Bagging and Boosting. 2005 IEEE International Conference on Systems, Man and Cybernetics, pp. 2340–2345 Vol. 3, https://doi.org/10.1109/ICSMC.2005.1571498
23. Bifet A, Holmes G, Pfahringer B (2010) Leveraging Bagging for Evolving Data Streams. Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases. ECML PKDD 2010. Lecture Notes in Computer Science, vol 6321. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15880-3_15
24. Gomes HM, Bifet A, Read J et al (2017) Adaptive random forests for evolving data stream classification. Mach Learn 106(9):1469–1495
25. Zhou ZH, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all [J]. Artif Intell 137(1–2):239–263
26. Brzezinski D, Stefanowski J (2013) Reacting to different types of concept drift: the accuracy updated ensemble algorithm [J]. IEEE Trans Nerrual Netw Learn Syst 25(1):81–94. https://doi.org/10.1109/TNNLS.2013.2251352
27. Zhou JX, Zhou ZH, Shen XH et al (2000) A selective constructing approach to neural network ensemble[J]. J Calc Res Dev 37(9):1039–1044
28. Du H, Zhang Y, Gang K, Zhang L, Chen YC (2021) Online ensemble learning algorithm for imbalanced data stream. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2021.107378
29. Wang Q, Luo ZH, Huang JC, Feng YH, Zhong L (2017) A novel ensemble method for imbalanced data learning: bagging of extrapolation-SMOTE SVM. Comput Intell Neurosci 2017:1827016. https://doi.org/10.1155/2017/1827016
30. Du H, Zhang Y (2020) Network anomaly detection based on selective ensemble algorithm [J]. J Supercomput. https://doi.org/10.1007/s11227-020-03374-z

31. Wang S, Minku LL, Yao X (2015) (2015) Resampling-based ensemble methods for online class imbalance learning [J]. Knowl Data Eng IEEE Trans 27(5):1356–1368
32. Gao J, Fan W, Han J, Philip SY (2007) A general framework for mining concept drifting data streams with skewed distributions. In: Proceedings of the 2007 SIAM International Conference on Data Mining (SDM), SIAM, 2007, pp. 3–14
33. Ren S, Liao B, Zhu W et al (2018) The gradual resampling ensemble for mining imbalanced data streams with concept drift. Neurocomputing 286:150–166
34. Sheng C, He H (2011) Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach. Evol Syst 2(1):35–50
35. Ester M, Kriegel H-P, Sander J, Xu X et al (1996) (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. Proc Sec Int Conf Knowl Discov Data Min (KDD) 96:226–231
36. Sheng C, He H (2009) SERA: Selectively recursive approach towards nonstationary imbalanced stream data mining. IEEE Press. In Proceedings of the 2009 international joint conference on Neural Networks (IJCNN'09). IEEE Press, 2053–2060, https://doi.org/10.1109/IJCNN.2009.5178874
37. Wang H, Fan W, Yu PS, Han J (2003) Mining concept-drifting data streams using ensemble classifiers. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03). Association for Computing Machinery, New York, NY, USA, 226–235. https://doi.org/10.1145/956750.956778
38. Du L, Song Q, Jia X (2014) detecting concept drift: an information entropy based method using an adaptive sliding window. Intell Data Anal 18(3):337–364
39. Bifet A, Holmes G, Kirkby R, Pfahringer B (2010) MOA: massive online analysis. J Mach Learn Res 11:1601–1604
40. Chen S, He H, Li K, Desai S (2010) Musera: multiple selectively recursive approach towards imbalanced stream data mining, In: Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), IEEE, 2010, pp. 1–8

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Zhang Yan[1,2] · Du Hongle[1,2] · Ke Gang[3] · Zhang Lin[1,2] · Yeh-Cheng Chen[4]

✉ Du Hongle
Dhl5597@163.com

Ke Gang
kegang95@126.com

Zhang Lin
zlin002@163.com

[1] School of Mathematics and Computer Application, Shangluo University, Shangluo City, Shaanxi Province, China

[2] Shangluo Public Big Data Research Center, Shangluo City, Shaanxi Province, China

[3] Department of Computer Engineering, Dongguan Polytechnic, Dongguan City, Guangdong Province, China

[4] Department of Computer Science, University of California, Davis, CA, USA