BEDCOE: Borderline Enhanced Disjunct Cluster Based Oversampling Ensemble for Online Multi-class Imbalance Learning

Shuxian Li[a,b,c], **Liyan Song**[a,b], **Yiu-ming Cheung**[c;*] **and Xin Yao**[a,b;**]

[a]Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology (SUSTech), Shenzhen, China.
[b]Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen, China.
[c]Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China.

Abstract. Multi-class imbalance learning usually confronts more challenges especially when learning from streaming data. Most existing methods focus on manipulating class imbalance ratios, disregarding other data properties such as the borderline and the disjunct. Recent studies have shown non-negligible impact of disregarding these properties on deteriorating predictive performance. Online multi-class imbalance would further exacerbate such negative impact. To abridge the research gap of online multi-class imbalance learning, we propose to enhance the number of training times of borderline samples based on the disjunct class-wise clusters that are adaptively constructed over time for each class individually. Specifically, we propose a borderline enhanced strategy for ensemble aiming to increase the number of training times of samples neighboring to borderline areas of different classes. We also propose to generate synthetic samples for training based on the adaptively learned disjunct clusters that are maintained for each class individually online, catering for online multi-class imbalance problem directly. These two components construct the **B**orderline **E**nhanced **D**isjunct **C**luster Based **O**versampling **E**nsemble (BEDCOE). Experimental studies are conducted and demonstrate the effectiveness of BEDCOE and each of its components in dealing with online multi-class imbalance.

1 Introduction

Learning with streaming data is very common in real-world applications which usually exposes to more severe challenges due to the limited time and memory [20]. Online multi-class imbalance learning is one of the popular topics in the area of data stream learning, for which some classes (minorities) own much fewer samples than others (majorities). This issue can potentially cause performance deterioration, especially for minority classes [22, 35, 20].

There have been only a few existing approaches to deal with the online multi-class imbalance issue, which can be grouped into three categories [35]: data level approaches, algorithm level approaches, and ensemble methods. Sampling methods are typically on the data level, which balance data samples of different classes by oversampling the minorities or/and undersampling the majorities in an online manner [32, 28]. Approaches of this category usually need to maintain a sliding window reserving recent training samples, increasing the memory burden. Cost-sensitive methods are representative of the algorithm level approaches, which handle online class imbalance by setting different weights to training samples of different classes adaptively with time [26, 27]. Class imbalance ratio is usually utilized to set the weights, so that samples from the same class will usually be treated with the same weight values.

The third category of methods is the online ensemble, which have shown to perform well for dealing with online multi-class imbalance learning [35, 20]. The proposed method in this paper belongs to this category. Methods of this category usually cooperate with sampling approaches to directly cope with class imbalance issue [34, 32, 8]. Training samples are typically used multiple times to update the model in sequence with the number of training times being derived based on latest class imbalance ratios. However, duplicating a single training sample multiple times would potentially overfit the model especially in the neighboring data area of this duplication [17, 22]. Related studies have shown that data properties such as the borderline and the disjunct would have non-negligible impact on predictive performance of online learning approaches [24, 6, 8], which should be especially taken into consideration when conducting the online multi-class imbalance learning process.

To abridge the research gap of online multi-class imbalance learning, we propose the **B**orderline **E**nhanced **D**isjunct **C**luster Based **O**versampling **E**nsemble (BEDCOE) method. First, the borderline degree of each sample is defined based on the probabilities that this sample can be classified into different classes. In this sense, samples with higher borderline degrees and lower class imbalance ratios would be assigned with larger number of training times, contributing to the borderline enhanced strategy. Then, disjunct clusters for each class individually are constructed and traced by an online clustering algorithm to capture the disjunct property of data space. Multiple synthetic samples can be produced based the combination among this training sample and cluster centers of this class, which are used for ensemble model adaptation, contributing to the disjunct cluster based oversampling. The overall training procedures of the proposed BEDCOE will be presented in Section 3.1 and in Algorithm 1. The main contributions of this paper are listed below:

* Corresponding Author. Email: ymc@comp.hkbu.edu.hk.
** Corresponding Author. Email: xiny@sustech.edu.cn.

1. We propose a novel borderline enhanced strategy for the online ensemble that can derive the number of training times for each sample individually so that training samples being closer to the borderline with a lower class imbalance ratio would be emphasized, alleviating the online multi-class imbalance issue.

2. We propose a novel disjunct cluster based oversampling method that is embedded into the online ensemble learning process to adaptively generate synthetic samples surrounding the training sample for model update, further alleviating the potential overfitting problem and catering for online multi-class imbalance.

3. We investigate the effectiveness of our BEDCOE and each of its two components for dealing with online multi-class imbalance problem experimentally based on a wide range of synthetic and real-world data sets.

The remainder of the paper is organized as follows. Section 2 presents related work. Section 3 presents our proposed BEDCOE. The experimental setup and results are discussed in Section 4. The paper is concluded in Section 5.

2 Related Work

Methods for dealing with online multi-class imbalance problem can generally be classified into three categories: data level approaches, algorithm level approaches, and ensemble-based methods [35].

2.1 Data Level Approaches

Sampling methods are popular in dealing with class imbalance problem in the offline scenario [17, 10, 16], but they cannot be directly adapted to the data stream learning. Revisions would be required for adapting to the online learning scenario.

A sliding window reserving the most recent training samples usually needs to be maintained for oversampling such as online SMOTE [32], generating synthetic samples using the training samples saved in the sliding window. C-SMOTE [4] is a variant of SMOTE for dealing with the binary imbalance streaming data, which actively detected the concept drift by ADWIN [5] and applied SMOTE on minority class samples stored in the most up-to-date sliding window. For online binary classification, SRE [28] proposes a selection-based resampling method to do oversampling or undersampling based on the data property of saved recent samples. And IOSDS [3] replicates samples that are not identified as noisy or borderline.

These methods mostly targeted on binary class imbalance problem and need to maintain a sliding window to reserve relevant training samples, increasing the memory burden.

2.2 Algorithm Level Approaches

Cost-sensitive methods are representative of this category. Setting different weights for different classes is the key idea, where the weights of minority classes are usually larger than that of majority classes [17].

WOS-ELM [26] is the cost-sensitive version of the OS-ELM algorithm [23], for which the class weights are set according to the class sizes. VWOS-ELM [25] trains a series of WOS-ELMs as base learners, and the final prediction is decided based on the weighted majority voting. Also based on OS-ELM, WOS-ELMK [12] uses kernel to avoid the non-optimal hidden node problem associated with OS-ELM methods, PBG [31] uses G-mean performance to monitor the concept drift and optimize the model in the learning process. Class sizes are used in these methods to determine the sample weights. AI-WSELM [27] is one of the state-of-the-art methods that can solve the multi-class imbalance problem on data streams, which proposes an improved weighting strategy based on the class sizes of recent data samples.

Class sizes are the most widely used data property to determine sample weights, for which samples of the same class are usually treated equally. This approach cannot reflect other data properties such as the borderline or the disjunct.

2.3 Ensemble Methods

Ensemble approaches have been shown to perform well for online class imbalance learning [35, 20]. Popular examples include MOOB and MUOB proposed in [34] to conduct online multi-class imbalance learning. They employ a time-decay class size to continuously derive class imbalance ratios, based on which the training times of each training data can be derived. Another popular example is Online SMOTE Bagging [32], which needs to maintain a sliding window for each class individually so that relevant training samples can be reserved for synthetic data generation.

Later on, Cano et al. [7] proposes Kappa Updated Ensemble (KUE) to gain better diversity for base learners, in an attempt to further improve prediction ability of online ensemble. After that, [8] proposes an advanced method called ROSE to improve the robustness of KUE. To directly deal with (binary- or multi-) class imbalance, ROSE computes the imbalance ratio of each class based on the recent samples to derive the training times of each sample for online ensemble learning.

We can see that existing online ensemble methods have only relied on class imbalance ratios to balance the multi-class imbalance continuously over time; however, other data properties such as the borderline and the disjunct have not been specifically dealt with [6, 8, 30, 21]. Recent studies have shown that the borderline and the disjunct have non-negligible impact on predictive performance of the online model, and class imbalance would possibly exacerbate such negative impact [6, 8, 30, 21]. Another potential issue for these online ensemble methods is that they usually duplicate the same training sample multiple times, potentially causing overfitting [17, 22].

3 Borderline Enhanced Disjunct Cluster Based Oversampling Ensemble (BEDCOE)

This section proposes **B**orderline **E**nhanced **D**isjunct **C**luster Based **O**versampling **E**nsemble (BEDCOE) to specifically deal with the online multi-class imbalance, abridging the research gap in this area.

3.1 Overall Test-then-Train Process of BEDCOE

Given a data stream $\{X_t\}$ where $t \in \{1, 2, \cdots\}$ is the test time step and $X_t \in \mathbb{R}^d$ denotes the d-dimensional data feature arrived at time step t, we use y_t to denote the true label of X_t and $y_t \in \{1, \cdots, c\}$ for $c \geq 2$. We follow the conventional "test-then-train" online learning process to proceed BEDCOE where test sample arrives one by one [14]. Specifically, given X_t arrived at t, the aim is to predict its label with the latest model as $\widehat{y}_t = \mathcal{H}_{t-1}(X_t)$. Then, one can obtain the true label y_t before $t+1$, and the new training sample (X_t, y_t) is used to update model $\mathcal{H}_{t-1}(\cdot)$ to $\mathcal{H}_t(\cdot)$.

Algorithm 1 presents the training procedures of BEDCOE at a given time step t. As shown in Line 1, we adopt the time-decay class

Algorithm 1 Training Procedures of the Proposed BEDCOE.

Inputs:
(1) ensemble $\mathcal{H}_{t-1}(\cdot)$ with base learners $f_m(\cdot)$ for $m = 1, \cdots, M$;
(2) class size $\Omega_{t-1} = \{\omega_{t-1}^{(1)}, \cdots, \omega_{t-1}^{(c)}\}$;
(3) class-wise clusters $\{D_{t-1}^{(1)}, \cdots, D_{t-1}^{(c)}\}$ for $c \geq 2$;
(4) a new training sample (X_t, y_t).

1: Update class size $\Omega_{t-1} \rightarrow \Omega_t$ according to Eqn. 1.
2: Compute class imbalance ratio $\{\lambda_t^{(1)}, \cdots, \lambda_t^{(c)}\}$ by Eqn. 2.
3: Update clusters $D_{t-1}^{(y_t)} \rightarrow D_t^{(y_t)}$ with (X_t, y_t).
4: **for** each base learner $f_m(\cdot)$ **do**
5: Derive training times $K_{t,m}$ of sample (X_t, y_t) by Alg. 2 for which the class imbalance ratio $\lambda = \lambda_t^{(y_t)}$.
6: **if** $K_{t,m} \geq 1$ **then**
7: Update $f_m(\cdot)$ with (X_t, y_t).
8: Get cluster center(s) $\{C_{y_t}^{(1)}, \cdots, C_{y_t}^{(n_{y_t})}\}$ from $D_t^{(y_t)}$, where n_{y_t} is the number of clusters for class y_t.
9: Generate synthetic training samples $\{(\widehat{X}_t^{(s)}, y_t)\}$ for $s = 1, \cdots, K_{t,m} - 1$, following Alg. 3.
10: **for** each $(\widehat{X}_t^{(s)}, y_t)$ **do**
11: Derive training times $\widehat{K}_{t,m}^{(s)}$ of sample $(\widehat{X}_t^{(s)}, y_t)$ by Alg. 2 for which $\lambda = 1$.
12: Update $f_m(\cdot)$ number of $\widehat{K}_{t,m}^{(s)}$ times using $(\widehat{X}_t^{(s)}, y_t)$.
13: **end for**
14: **end if**
15: **end for**

sizes $\Omega_t = \{\omega_t^{(1)}, \cdots, \omega_t^{(c)}\}$ proposed in [34] to trace the class imbalance status continuously over time. They were used to quantify the occurrence probability of each class and were updated as:

$$\omega_t^{(k)} = \theta \cdot \omega_{t-1}^{(k)} + (1 - \theta) \cdot [(X_t, k)], \tag{1}$$

where $[(X_t, k)]$ is the indicator function equaling 1 if X_t belongs to class k or equaling 0 if otherwise, and θ is the time-decay factor that is set to 0.9 in our paper following [34]. As shown in Line 2, based on the time decay class size, multi-class imbalance ratios $\{\lambda_t^{(1)}, \cdots, \lambda_t^{(c)}\}$ are defined following [34] as:

$$\lambda_t^{(k)} = \omega_{max} / \omega_t^{(k)}, \tag{2}$$

where $\omega_{max} = \max_{k=1}^{c} \omega_t^{(k)}$. As shown in Line 3, clusters $D_{t-1}^{(y_t)}$ that have been exclusively constructed for class y_t are updated with (X_t, y_t). This learning process can be proceeded by any online clustering algorithm such as CluStream [1], DenStream [9], and DBStream [15]. This paper opts for DenStream to construct clusters for each class individually.

As shown in Line 5, the borderline degree $\rho_m(X_t, y_t)$ for data (X_t, y_t) is derived following Alg. 2 and is then used to decide the training times $K_{t,m}$, contributing to the borderline enhanced strategy. When $K_{t,m} \geq 1$, this training sample is used to update base learner $f_m(\cdot)$ (Line 7); then $K_{t,m} - 1$ synthetic samples are generated by the latest clusters of the class y_t (Line 8∼9), contributing to the disjunct cluster based oversampling method in Alg. 3. Learning with synthetic samples follows the same procedures of the borderline enhanced strategy as shown in Line 11∼12. We detail the borderline enhanced strategy and the disjunct cluster based oversampling in the subsequent subsections individually.

3.2 Borderline Enhanced Strategy

This subsection presents the borderline enhanced strategy listed in Line 5 and Line 11 of Alg. 1 to decide the training times of a given sample, based on which the online ensemble is updated multiple times. Particularly, the number of training times for a training sample depends on how close it locates to the borderline being the area samples of different classes overlap [6, 30].

Algorithm 2 Borderline Enhanced Strategy.

Inputs:
(1) base learner $f_m(\cdot)$
(2) sample (X, y);
(3) class imbalance ratio λ.

1: Compute borderline degree $\rho_m(X, y)$ by Eqn. 3.
2: Derive the training times K by Eqn. 4.

Algorithm 2 presents the borderline enhanced strategy. We use $p_m(X, k)$ to denote the prediction probability that data X is classified to class k by base learner $f_m(\cdot)$. The borderline degree of a given sample (X, y) is formulated as:

$$\rho_m(X, y) = \exp\left(\max_{k \neq y} p_m(X, k) - p_m(X, y) + 1\right), \tag{3}$$

where $\max_{k \neq y} p_m(X, k)$ quantifies the maximal probability that $f_m(\cdot)$ wrongly predicts X and larger value indicates higher misclassification possibility; $p_m(X, y)$ is the prediction probability that $f_m(\cdot)$ correctly predicts X, and lower $p_m(X, y)$ indicates that this sample is more difficult to be correctly predicted by this base learner. In this sense, a larger borderline degree $\rho_m(X, y)$ of training sample (X, y) estimates a higher possibility that this data would locate closer to the borderline, thus is anticipated to derive a higher training times to facilitate the model to be more adapted to the neighboring area of this training sample.

Given class imbalance ratio λ and base learner $f_m(\cdot)$, the borderline enhanced training time of sample (X, y) is formulated as a random variable being sampled from Poisson distribution as:

$$K \sim \mathcal{Z}(\rho_m(X, y) \cdot Poisson(\lambda)), \tag{4}$$

where $\mathcal{Z}(\cdot)$ is the rounding function to derive an integer. When $Poisson(\lambda)$ samples out 0, $K = 0$ occurs meaning that (X, y) is not used for training. Ultimately, each base learner $f_m(\cdot)$ would be updated K times based on (X, y).

3.3 Disjunct Cluster Based Oversampling

Given training sample (X_t, y_t) and the training times $K_{t,m}$ derived by Alg. 2, when $K_{t,m} \geq 1$, base learner $f_m(\cdot)$ is firstly updated with training sample (X_t, y_t), consuming 1 training time (Line 7 of Algo. 1). This base learner is then updated with another $K_{t,m} - 1$ synthetic variants of (X_t, y_t) (Line 8∼9 of Alg. 1). The reason for not duplicating this training sample $K_{t,m}$ times is to alleviate the potential issue of overfitting. Clusters $D_t^{(k)}$ for $k = 1, \cdots, c$ are constructed to trace the disjunct within each class individually. Based on these notations, we propose the disjunct cluster based oversampling method in Algorithm 3.

We generate n_{y_t} temporary samples $\{X^{(i)}\}$ for $i = 1, \cdots, n_{y_t}$ based on (X_t, y_t) and cluster center $C_{y_t}^{(i)}$ as:

$$X^{(i)} = X_t + \alpha \cdot (C_{y_t}^{(i)} - X_t), \tag{5}$$

where $\alpha \sim U(0, 1)$ and $U(0, 1)$ denotes the uniform distribution ranging from 0 and 1. Based on them, we produce a synthetic variant $\widehat{X}_t$ of training sample X_t by linearly incorporating (X_t, y_t) and those temporary samples as:

$$\widehat{X}_t = \frac{m}{M} \cdot X_t + (1 - \frac{m}{M}) \cdot \sum_{i=1}^{n_{y_t}} \gamma^{(i)} \cdot X^{(i)}, \tag{6}$$

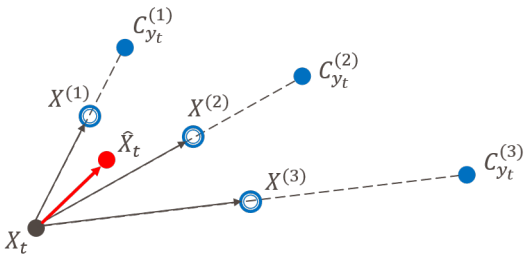


Figure 1. Illustration of the disjunct cluster based sampling method. Given training sample (X_t, y_t), we use $\{C_{y_t}^{(i)}\}$ for $i = 1, 2, 3$ to denote cluster centers corresponding to class y_t. Temporary samples $\{X^{(i)}\}$ are generated by Eqn. 5. A synthetic variant $\widehat{X}_t$ is finally produced by Eqn. 6.

Algorithm 3 Disjunct Cluster Based Oversampling Method.

Inputs:
(1) index m of $f_m(\cdot)$ and overall number M of base learners;
(2) training sample (X_t, y_t);
(3) center(s) $\{C_{y_t}^{(1)}, \cdots, C_{y_t}^{(n_{y_t})}\}$ of cluster(s) in class y_t, where n_{y_t} is the number of learned cluster(s) in class y_t.

1: Produce temporary samples $X^{(i)}$ for $i = 1, \cdots, n_{y_t}$ by Eqn. 5.
2: Produce a synthetic variant $\widehat{X}_t$ of X_t by Eqn. 6.

where we formulate the weight of temporary sample $X^{(i)}$ by

$$\gamma^{(i)} = \frac{1/dist(X_t, C_{y_t}^{(i)})}{\sum_{j=1}^{n_{y_t}} 1/dist(X_t, C_{y_t}^{(j)})}, \tag{7}$$

and $dist(X_t, C_{y_t}^{(i)})$ is the Euclidean distance between X_t and $C_{y_t}^{(i)}$. The heuristic m/M for each base leaner $f_m(\cdot)$ in Eqn. 6 is deliberately designed to weight real data X_t against temporary data captured in clusters. As a result, a base learner in a latter sequential order (larger m/M) would tend to better adapt to the real data generation status compared to the earlier one, producing the multi-scale simulation of the data space. This would potentially help with predictive performance by improving the diversity of online ensemble learning. Algorithm 3 explains the disjunct cluster based oversampling method aiming for generating a synthetic sample based on cluster centers of the same class. Figure 1 illustrates the generation process given three cluster centers.

Overall, such oversampling procedure repeats $(K_{t,m} - 1)$ times to generate $(K_{t,m} - 1)$ synthetic samples $\{(\widehat{X}_t^{(s)}, y_t)\}$ for $s = 1, \cdots, K_{t,m} - 1$, based on which base learner $f_m(\cdot)$ is updated sequentially (see Line 10∼13 of Algorithm 1). Specifically, each synthetic sample $(\widehat{X}_t^{(s)}, y_t)$ is used to update base learner $f_m(\cdot)$ the number of $\widehat{K}_{t,m}^{(s)}$ times based on the borderline enhanced strategy, where $\widehat{K}_{t,m}^{(s)}$ is derived by Alg. 2 with the input $\lambda = 1$. The reason for setting class imbalance ratio $\lambda = 1$ in this scenario is because multi-class imbalance has been dealt with while deriving training times $K_{t,m}$ for the real data (X_t, y_t) and thus it is unnecessary to overemphasize this issue for training on synthetic data.

4 Experimental Studies

This section aims to investigate the proposed BEDCOE from two perspectives: performance comparisons against state-of-art methods for online multi-class imbalance learning and the effectiveness of two proposed components of our BEDCOE. Code is available online [1].

[1] Code: https://github.com/shuxian-li/BEDCOE

Table 1. Overview of the data set. "#Data" denotes the total number of samples within this data set, "#Initial" denotes the number of samples used for data normalization and model initialization, #Fea denotes the number of features, #Class denotes the number of classes, and IR denotes the overall static imbalance ratio being computed as the ratio between the largest class size and the smallest class size.

Data set	#Data	#Initial	#Fea	#Class	IR
Gaussian	20000	500	2	5	10.00
Abrupt	20000	500	33	6	15.70
Gradual	20000	3500	33	6	108.19
Incremental	20000	3500	33	6	39.95
Incremental-Abrupt	20000	3500	33	6	23.19
Incremental-Reoccurring	20000	3500	33	6	22.99
Elec	20000	500	8	2	1.29
Luxembourg	1901	200	31	2	1.06
NOAA	18159	500	8	2	2.19
Ozone	2534	200	72	2	14.84
Ecoli	332	150	7	6	28.60
Dermatology	358	200	34	6	5.55
Pageblocks	545	400	10	4	61.50
Thyroid	720	500	21	3	39.18
Yeast	1484	1000	8	10	92.60
Chess	533	200	5	3	10.17
Keystroke	1600	200	10	4	1.00
Outdoor	4000	1300	21	40	1.00
Powersupply	29928	500	2	24	1.00
Rialto	20000	500	27	10	1.00

4.1 Experimental Setup

This paper adopts 20 data sets to conduct experimental studies, which include 6 synthetic data sets (Gaussian, Abrupt, Gradual, Incremental, Incremental-Abrupt, and Incremental-Reoccurring), 4 real-world binary data sets (Elec, Luxembourg, NOAA, and Ozone), and 10 real-world multi-class data sets (Ecoli, Dermatology, Pageblocks, Thyroid, Yeast, Chess, Keystroke, Outdoor, Powersupply, and Rialto), covering a wide range of data properties for online multi-class imbalance learning. Table 1 summarize their information. The initial number of each data set is decided based on the requirement that each class needs to contain training samples. The overall static imbalance ratio of each data set outlines the class imbalance severity disregarding the fact that the class imbalance ratio of each class is actually varying throughout the data stream in the online learning scenario. Data set Gaussian is produced based on multiple Gaussian distributions synthetically; Ecoli, Dermatology, Pageblocks, Thyroid, and Yeast are available in the Keel repository [2]; Elec, Luxembourg, NOAA, Ozone, Chess, Keystroke, Outdoor, Powersupply, and Rialto are available in the USP-DS repository [29].

We compare the proposed BEDCOE against 5 state-of-the-art online multi-class imbalance learning approaches, including MOOB, MUOB [34], Online SMOTE Bagging ("SmoteOB") [32], AI-WSELM [27], and ROSE [8]. Except for AI-WSELM [23], Hoeffding trees [19] are adopted as base learners of online ensemble approaches, for yielding generally good predictive performance and being robust to various circumstances [33, 34, 32]. The total number of base learners is set to 10, following the previous studies [34, 8]. For Online SMOTE Bagging [32], the size of the sliding window is set to 100 for each class. All methods follow the strict online learning setup as explained in Sec. 3.1.

Prequential G-mean and balanced accuracy with fading factor 0.99 are chosen as the performance metrics for being popularly used in online multi-class imbalance learning [35, 31]. Predictive performance is evaluated based on the remaining samples after the initialization number. Mean performance across 10 runs is conducted for comparisons.

We perform Friedman tests [11] for statistical comparisons between competing methods across data sets. The null hypothesis (H0) states that all models are equivalent in terms of the predictive performance metric. The alternative hypothesis (H1) states that at least one pair of methods differ significantly. When H0 is rejected, Holm-Bonferroni correction [18] is conducted as the post-hoc test.

4.2 Performance Comparison

4.2.1 Overall Performance Across Time

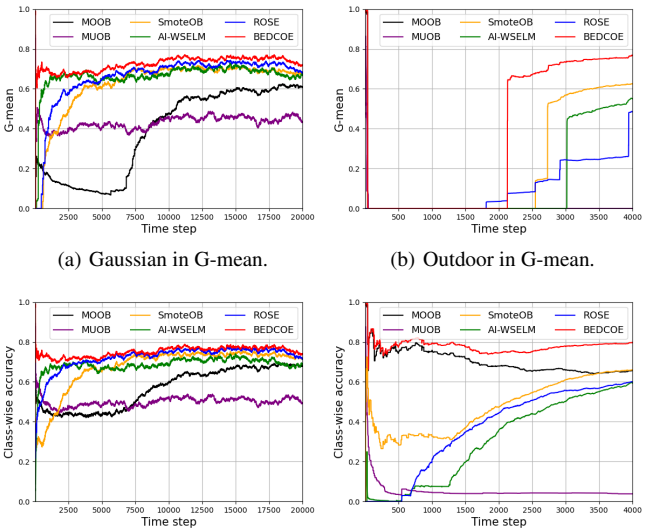
We can see from Table 2(a) that in terms of G-mean, our proposed BEDCOE performs the best in 9 out of 20 data sets, and the 2nd best in 7 data sets. Friedman tests at the significance level 0.05 reject H0 with the p-value 1.50×10^{-4}, showing that there is significant difference between methods. Average ranks ("avgRank") across data sets are reported in the last row of this table to show how well each method performs compared to others across data sets. Average rank of BEDCOE is 1.9, being the best (lowest) among all methods, indicating that our method can generally perform the best across different kinds of data streams. Post-hoc tests are then conducted to detect whether BEDCOE has significant difference from the competitors, for which BEDCOE is chosen as the control method. Post-hoc comparisons show that the proposed BEDCOE can significantly outperform all of the competitors.

It is worth noting that methods such as MOOB, MUOB, AI-WESELM and ROSE report zero G-mean in some data sets such as Yeast, Outdoor, and Powersupply. Further inspection found that those "terrible" data sets have much fewer samples per class compared to others. For example, the average sample size is 100 in Outdoor compared to 400 in Keystroke. This issue becomes worse when confronting online multi-class imbalance, which is the case of these data sets. Such issue possibly results in terrible prediction performance in terms of recall on minority classes especially when the method cannot perform well. Whenever the method attains a very tiny (even zero) recall on any of these minority classes, this would result in zero G-mean. We can see shortly that predictive performance in terms of balanced accuracy does not contain zero performance anymore.

We can also see from Table 3(a) that in terms of balanced accuracy, the proposed BEDCOE performs the best in 9 out of 20 data sets, and the second best in 9 out of 20 data sets. Friedman tests at the significance level 0.05 reject H0 with the p-value 5.19×10^{-6}, showing that there is significant difference between methods. Average rank of BEDCOE is 1.7, being the best (lowest) among all methods, indicating that our method can generally perform the best across different kinds of data streams. Post-hoc tests are then conducted to investigate whether BEDCOE has significant difference from the competitors, for which BEDCOE is chosen as the control method. Post-hoc comparisons show that the proposed BEDCOE can significantly outperform all of the competitors.

4.2.2 Continuous Performance Throughout Time

Figure 2 shows performance comparisons of the proposed method against the competitors throughout time steps on two representative data sets in terms of G-mean and balanced accuracy, respectively. Results of other data sets show similar patterns and were omitted for the space reason. We can see that the proposed BEDCOE can usually outperform most of the other methods constantly at most time steps in terms of both G-mean and balanced accuracy, respectively, demonstrating the effectiveness of our approach in helping with the performance improvement continuously over time.



(a) Gaussian in G-mean. (b) Outdoor in G-mean.

(c) Gaussian in class-wise acc. (d) Outdoor in class-wise acc.

Figure 2. Continuous performance comparison throughout time on representative data sets in terms of G-mean and balanced accuracy.

4.3 Ablation Studies

This section aims to study the effectiveness of each of the proposed components of BEDCOE, including the borderline enhanced strategy proposed in Sec. 3.2 and the disjunct cluster based oversampling method proposed in Sec. 3.3. To this end, we respectively remove each of the two components from BEDCOE, leading to BEDCOE eliminating the borderline enhanced strategy ("BEDCOE-BE") and BEDCOE eliminating the disjunct cluster based oversampling method ("BEDCOE-DC"), which would be individually compared to BEDCOE. If predictive performance declined significantly after eliminating one component, we can conclude that this component is crucial in dealing with online class imbalance.

Table 2(b) and Table 3(b) report predictive performances of the two variants in terms of G-mean and balanced accuracy, respectively. BEDCOE is chosen as the control method and is compared with BEDCOE-BE and BEDCOE-DC, individually. Wilcoxon signed rank tests [36] are conducted to detect whether BEDCOE has significant difference from each of the variants.

4.3.1 Effectiveness of Borderline Enhanced Strategy

Effectiveness of borderline enhanced strategy with respect to dealing with online multi-class imbalance is studied via performance comparison between BEDCOE and BEDCOE-BE. Specifically, the borderline enhanced strategy is eliminated by replacing $K_{t,m} \sim \mathcal{Z}(\rho_m(X_t, y_t) \cdot Poisson(\lambda_t^{y_t}))$ with $K_{t,m} \sim Poisson(\lambda_t^{y_t})$ (Line 5 of Alg. 1) and replacing $\widehat{K}_{t,m}^{(s)} \sim \mathcal{Z}(\rho_m(X_t, y_t) \cdot Poisson(1))$ with $\widehat{K}_{t,m}^{(s)} \sim Poisson(1)$ (Line 11 of Alg. 1) individually.

We can see from the first column of Table 2(b) that in terms of G-means, BEDCOE-BE performs worse than BEDCOE in 15 out of 20 data sets. Wilcoxon signed rank test rejects H0 with p-value 0.01, meaning there is significant difference between BEDCOE and BEDCOE-BE. Average rank of BEDCOE-BE (1.75) is worse than that of BEDCOE (1.25), meaning that BEDCOE-BE is significantly inferior to BEDCOE. This indicates that eliminating the borderline enhanced strategy would result in significant decline in predictive performance in terms of G-mean, showing the effectiveness of the

Table 2. Performance comparison in terms of G-mean (%). Each entry is the mean±std performance across 10 runs. The best performance on each data set is highlighted in bold, and the 2nd best performance is highlighted in italic. The last row lists the average ranks (avgRank) of each model across data sets. Significant difference against BEDCOE is highlighted in yellow. Part (b) reports the ablation results of the proposed BEDCOE against its variants.

(a) Performance Comparison

Data set	MOOB	MUOB	SmoteOB	AI-WSELM	ROSE	BEDCOE
Gaussian	39.02±4.39	43.11±1.48	63.97±0.54	67.40±0.42	*67.41±1.04*	**72.66±0.53**
Abrupt	59.56±0.26	56.88±0.53	53.11±2.25	**66.52±0.46**	1.66±0.12	*59.68±1.16*
Gradual	*40.58±4.96*	36.20±2.82	5.06±9.87	**58.17±0.98**	24.76±8.32	25.98±9.20
Incremental	44.06±0.68	40.98±0.99	33.75±0.85	41.10±0.61	*48.77±1.30*	**54.16±0.85**
Incremental-Abrupt	40.15±0.62	45.44±1.40	31.11±0.69	43.14±0.84	*47.71±0.88*	**58.10±0.37**
Incremental-Reoccurring	42.70±0.32	46.62±1.20	37.64±4.02	43.74±0.82	*49.97±0.72*	**56.00±0.54**
Elec	86.71±0.46	84.07±0.60	84.15±0.45	71.10±7.32	**91.04±0.13**	*90.84±0.15*
Luxembourg	*99.99±0.02*	98.45±0.36	99.96±0.04	90.24±0.84	99.75±0.02	**100.00±0.00**
NOAA	70.84±0.37	69.01±0.80	70.10±0.17	**79.17±0.10**	*74.46±0.22*	72.58±0.42
Ozone	50.47±5.66	**78.18±0.74**	*76.54±0.14*	70.73±1.63	62.63±1.21	66.61±2.63
Ecoli	**82.73±0.83**	5.79±11.80	76.53±3.63	74.47±1.36	50.94±2.24	*81.89±0.74*
Dermatology	90.87±1.55	*92.29±1.09*	**94.28±1.37**	92.15±0.95	79.51±0.40	91.66±1.30
Pageblocks	*80.88±0.94*	24.85±18.19	0.64±1.28	69.55±1.25	67.98±0.16	**86.12±1.61**
Thyroid	**92.31±0.64**	54.61±20.84	52.45±6.77	54.81±3.07	61.20±4.76	*91.57±5.38*
Yeast	0.00±0.00	0.00±0.00	12.17±8.88	**47.43±0.77**	8.39±12.82	*16.89±21.97*
Chess	55.79±2.73	52.93±2.87	52.43±5.41	*56.27±1.16*	26.46±9.98	**61.66±3.56**
Keystroke	82.29±1.92	82.61±2.34	81.94±1.42	**92.34±0.20**	86.78±0.84	*90.60±0.97*
Outdoor	0.00±0.00	0.00±0.00	*28.75±1.57*	18.07±0.15	14.03±13.66	**49.82±1.10**
Powersupply	0.00±0.00	0.00±0.00	**8.32±1.48**	0.00±0.00	0.00±0.00	*3.92±0.92*
Rialto	11.53±0.42	12.35±0.76	*62.90±0.72*	48.48±0.29	46.33±2.14	**63.09±1.14**
avgRank	3.875	4.325	4.1	3.075	3.725	1.9

(b) Ablation Studies

BEDCOE-BE	BEDCOE-DC
39.04±2.80	68.40±1.91
61.25±0.45	**61.29±1.27**
37.77±3.28	**35.08±0.74**
45.83±0.32	51.78±1.24
42.97±0.40	52.67±0.32
44.80±0.48	53.36±0.34
85.61±0.57	**91.51±0.23**
99.97±0.06	**100.00±0.00**
70.57±0.32	**72.92±0.49**
77.22±0.81	64.66±4.12
82.42±1.03	**83.15±0.81**
93.83±1.33	88.74±1.02
83.43±1.64	76.69±1.39
71.14±15.24	**93.37±2.24**
0.00±0.00	0.00±0.00
54.30±0.67	60.34±2.27
84.01±2.16	86.63±1.80
0.00±0.00	0.60±0.60
0.00±0.00	1.26±0.50
11.44±0.51	44.85±2.06
1.25 vs 1.75	1.325 vs 1.675

Table 3. Performance comparison in terms of balanced accuracy (%). Each entry is the mean±std performance across 10 runs. The best performance on each data set is highlighted in bold, and the second best performance is highlighted by in italic. The last row lists the average ranks (avgRank) of each model across data sets. Significant difference against BEDCOE is highlighted in yellow. Part (b) reports the ablation results of the proposed BEDCOE against its variants.

(a) Performance Comparison

Data set	MOOB	MUOB	SmoteOB	AI-WSELM	ROSE	BEDCOE
Gaussian	56.95±1.96	49.84±0.77	68.58±0.35	69.18±0.35	*72.83±0.47*	**75.15±0.40**
Abrupt	64.22±0.22	62.01±0.40	62.01±0.73	**69.32±0.25**	16.59±0.08	*65.01±0.49*
Gradual	58.46±0.60	49.69±1.24	58.28±0.36	**66.36±0.73**	59.39±0.81	*63.03±0.46*
Incremental	52.20±0.39	47.37±0.61	50.80±0.52	54.57±0.37	*55.39±0.86*	**57.79±0.64**
Incremental-Abrupt	50.58±0.38	51.51±0.96	50.94±0.51	53.40±0.39	*56.20±0.50*	**61.40±0.36**
Incremental-Reoccurring	51.47±0.19	51.68±0.89	51.29±0.36	53.63±0.48	*56.97±0.74*	**60.19±0.37**
Elec	87.20±0.41	84.94±0.46	84.92±0.38	75.66±2.71	**91.15±0.12**	*91.03±0.14*
Luxembourg	*99.99±0.02*	98.45±0.36	99.96±0.04	90.26±0.84	99.75±0.02	**100.00±0.00**
NOAA	72.00±0.29	70.55±0.43	71.01±0.16	**79.36±0.10**	*75.00±0.21*	73.40±0.38
Ozone	63.18±2.31	**78.74±0.73**	*77.74±0.13*	72.20±1.29	68.43±0.72	70.77±1.78
Ecoli	**83.26±0.78**	45.70±8.51	79.53±1.85	75.80±1.56	63.00±2.74	*82.51±0.69*
Dermatology	92.16±1.04	92.57±1.10	**94.74±1.12**	92.36±0.91	80.49±0.37	*92.70±0.89*
Pageblocks	*83.16±0.68*	47.92±11.24	55.44±1.68	71.27±1.23	70.43±0.16	**86.98±1.33**
Thyroid	**92.49±0.61**	64.74±10.09	61.04±4.72	59.78±2.38	69.14±2.19	*92.11±4.72*
Yeast	*55.64±0.67*	10.22±0.60	35.88±3.05	51.82±1.20	41.76±1.72	**58.04±1.18**
Chess	*57.94±2.17*	55.24±2.17	55.51±3.14	57.56±1.02	50.86±1.56	**63.59±3.14**
Keystroke	82.76±1.72	83.13±2.23	82.91±1.26	**92.41±0.20**	87.01±0.79	*90.70±0.96*
Outdoor	*66.91±0.37*	4.16±0.46	54.40±0.56	43.05±0.31	49.76±1.38	**76.99±0.63**
Powersupply	16.07±0.08	16.15±0.08	**17.78±0.12**	15.35±0.11	14.28±0.32	*16.32±0.13*
Rialto	27.39±0.22	27.90±0.49	**68.56±0.60**	51.57±0.34	54.20±1.37	*67.00±0.98*
avgRank	3.65	4.75	3.85	3.4	3.65	1.7

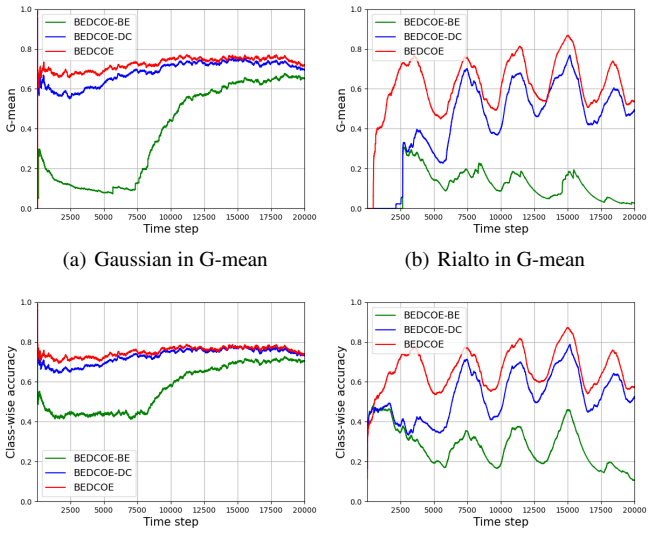
(b) Ablation Studies

BEDCOE-BE	BEDCOE-DC
57.01±1.22	72.96±1.19
64.75±0.36	63.96±0.72
60.52±0.54	62.19±0.63
52.72±0.26	56.28±0.86
51.31±0.28	58.40±0.27
52.08±0.35	58.71±0.22
86.31±0.40	**91.62±0.23**
99.97±0.06	**100.00±0.00**
71.75±0.29	73.36±0.45
77.96±0.90	69.10±2.86
83.15±0.79	**83.54±0.80**
94.34±1.04	90.79±0.63
85.35±1.17	80.17±1.01
74.85±12.00	**93.54±2.08**
55.56±0.57	54.88±0.50
57.29±0.50	62.44±1.59
84.34±2.00	86.87±1.70
67.60±0.25	71.10±0.35
16.06±0.07	15.91±0.05
27.50±0.37	53.23±1.79
1.15 vs 1.85	1.175 vs 1.825

borderline-enhanced strategy in dealing with online multi-class imbalance.

We can also see from the first column of Table 3(b) that in terms of balanced accuracy, BEDCOE-BE performs worse than BEDCOE in 17 out of 20 data sets. Wilcoxon signed rank test rejects H0 with p-value 0.0025, meaning there is significant difference between BEDCOE and BEDCOE-BE. Average rank of BEDCOE-BE (1.85) is worse than that of BEDCOE (1.15), meaning that BEDCOE-BE is significantly inferior to BEDCOE. This indicates that eliminating the borderline enhanced strategy would result in significant decline in predictive performance in terms of balanced accuracy, showing the effectiveness of the borderline enhanced strategy in dealing with online multi-class imbalance.

Moreover, we illustrate continuous performance throughout time in Figure 3 in terms of G-mean and balanced accuracy, respectively. We can observe similar results that eliminating the borderline enhanced strategy would result in the performance decline at most test steps constantly, demonstrating the effectiveness of borderline enhanced strategy. Therefore, we conclude that the borderline enhanced strategy plays an important role in dealing with online multi-class imbalance and should not be eliminated from BEDCOE.



(a) Gaussian in G-mean (b) Rialto in G-mean

(c) Gaussian in class-wise acc. (d) Rialto in class-wise acc.

Figure 3. Continuous performance comparison throughout time on representative data sets in terms of G-mean and balanced accuracy.

4.3.2 *Effectiveness of Disjunct Cluster Based Oversampling*

Effectiveness of disjunct cluster based oversampling method with respect to dealing with online multi-class imbalance is analyzed via the performance comparison between BEDCOE vs BEDCOE-DC. Specifically, the disjunct cluster based oversampling is eliminated by replacing Line 7∼13 of Alg. 1 by the procedure that updates base learner $f_m(\cdot)$ with (X_t, y_t) the number of $K_{t,m}$ times.

We can see from the second column of Table 2(b) that in terms of G-mean, BEDCOE-DC performs worse than BEDCOE in 13 out of 20 data sets. Wilcoxon signed rank test rejects H0 with p-value 0.0112, meaning that there is significant difference between BEDCOE and BEDCOE-DC. Average rank of BEDCOE-DC (1.675) is worse than that of BEDCOE (1.325), meaning that BEDCOE-DC is significantly inferior to BEDCOE. This indicates that eliminating the disjunct cluster based oversampling would result in significant performance decline in terms of G-mean, showing the effectiveness of the disjunct cluster based oversampling method in dealing with online multi-class imbalance.

We can also see from the second column of Table 3(b) that in terms of balanced accuracy, BEDCOE-DC performs worse than BEDCOE in 16 out of 20 data sets. Wilcoxon signed rank test rejects H0 with p-value 0.00148, meaning there is significant difference between BEDCOE and BEDCOE-DC. Average rank of BEDCOE-DC (1.825) is worse than that of BEDCOE (1.175), meaning that BEDCOE-DC is significantly inferior to BEDCOE. This indicates that eliminating the disjunct cluster based oversampling would result in significant performance decline in terms of balanced accuracy, showing the effectiveness of the disjunct cluster based oversampling in dealing with online multi-class imbalance.

As shown in Figure 3, we can observe similar results that eliminating the disjunct cluster based oversampling would result in the performance decline at most test steps constantly, demonstrating the effectiveness of disjunct cluster based oversampling method. Therefore, we conclude that the disjunct cluster based oversampling plays an important role in dealing with online multi-class imbalance and should not be eliminated from BEDCOE.

4.4 *Further Discussion*

Computational and space complexity: The computational complexity of our proposed BEDCOE is $\mathcal{O}(M \times N)$, where M denotes the number of base learners and N denotes the data set size, being the same to MOOB (the most popular work) [34] and ROSE (the most recent work) [8].

We would like to highlight that, the same as MOOB, our BEDCOE follows the strict online learning scenario where no previous data is allowed to store; in contrast, ROSE requested the storage of a sliding window of data of size w. Therefore, the storage complexity is $\mathcal{O}(d)$ BEDCOE and MOOB, and is $\mathcal{O}(w \times d)$ for ROSE, where d denotes the number of data features. Overall, our BEDCOE is computationally efficient with low requirement of storage accumulation, being competitive to current online multi-class imbalance methods. Considering the generally better predictive performance, it is recommended to adopt BEDCOE for dealing with online multi-class imbalance problem.

Potential limitation of BEDCOE: Considering the overall static imbalance ratio (IR) in Table 1, we further analyze the Spearman correlation[13] between the prediction performance of BEDCOE and IRs across data sets, being -0.2289 (weak) in terms of G-Means and -0.3184 (weak) in terms of balanced accuracy. This indicates there is a weak negative correlation between the prediction performance of our proposed method and the overall static imbalance ratio of a data set. This means that our BEDCOE, similar to many other existing online multi-class imbalance methods, may confront the limitation of predictive performance for data sets with higher overall static imbalance ratios.

5 Conclusions

This paper proposed **B**orderline **E**nhanced **D**isjunct **C**luster-based **O**versampling **E**nsemble (BEDCOE) method to deal with the online multi-class imbalance issue. Specifically, the borderline enhanced strategy was proposed to increase the number of training times of the sample that is closer to the borderline, alleviating the online multi-class imbalance issue; the disjunct cluster based oversampling method was proposed to produce multiple synthetic samples for the real training data based on the clusters especially constructed for this class, alleviating the potential issue of overfitting compared to learning with the duplication of the same training sample multiple times.

We conducted systematic experimental investigation on the proposed BEDCOE based on 20 synthetic and real-world data sets. Experimental results demonstrated the superiority of BEDCOE over state-of-the-art online multi-class imbalance methods in achieving significantly better predictive performance in terms of both G-mean and balanced accuracy, respectively. Continuous predictive performance throughout time also showed the superior performance of BEDCOE against competitors at most test steps constantly. By eliminating each of the proposed components from BEDCOE, we induced two degraded variants, namely BEDCOE-BE and BEDCOE-DC. Experimental analyses demonstrated the effectiveness of each of the proposed components in dealing with the online multi-class imbalance issue.

Future work includes further investigating how label noise, missing data, and high feature dimension affect the performance of our proposed method and extending BEDCOE to better adapt to varying conditions of data sets.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) under Grant No. 62002148 and Grant No. 62250710682, Guangdong Provincial Key Laboratory under Grant No. 2020B121201001, the Program for Guangdong Introducing Innovative and Enterpreneurial Teams under Grant No. 2017ZT07X386, and Research Institute of Trustworthy Autonomous Systems (RITAS).

References

[1] Charu C Aggarwal, S Yu Philip, Jiawei Han, and Jianyong Wang, 'A framework for clustering evolving data streams', in *Proceedings 2003 VLDB conference*, pp. 81–92. Elsevier, (2003).

[2] Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, and Francisco Herrera, 'KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework', *Journal of Multiple-Valued Logic and Soft Computing*, **17**(2-3), 255–287, (2011).

[3] N Anupama and Sudarson Jena, 'A novel approach using incremental oversampling for data stream mining', *Evolving Systems*, **10**(3), 351–362, (2019).

[4] Alessio Bernardo, Heitor Murilo Gomes, Jacob Montiel, Bernhard Pfahringer, Albert Bifet, and Emanuele Della Valle, 'C-SMOTE: continuous synthetic minority oversampling for evolving data streams', in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 483–492, (2020).

[5] Albert Bifet and Ricard Gavalda, 'Learning from time-changing data with adaptive windowing', in *Proceedings of the 2007 SIAM international conference on data mining*, pp. 443–448. SIAM, (2007).

[6] Dariusz Brzezinski, Leandro L Minku, Tomasz Pewinski, Jerzy Stefanowski, and Artur Szumaczuk, 'The impact of data difficulty factors on classification of imbalanced and concept drifting data streams', *Knowledge and Information Systems*, **63**(6), 1429–1469, (2021).

[7] Alberto Cano and Bartosz Krawczyk, 'Kappa Updated Ensemble for drifting data stream mining', *Machine Learning*, **109**(1), 175–218, (2020).

[8] Alberto Cano and Bartosz Krawczyk, 'ROSE: robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams', *Machine Learning*, 1–39, (2022).

[9] Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou, 'Density-based clustering over an evolving data stream with noise', in *Proceedings of the 2006 SIAM international conference on data mining*, pp. 328–339. SIAM, (2006).

[10] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer, 'SMOTE: synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research*, **16**(1), 321–357, (2002).

[11] Janez Demšar, 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine Learning Research*, **7**, 1–30, (2006).

[12] Shuya Ding, Bilal Mirza, Zhiping Lin, Jiuwen Cao, Xiaoping Lai, Tam V Nguyen, and Jose Sepulveda, 'Kernel based online learning for imbalance multiclass classification', *Neurocomputing*, **277**, 139–148, (2018).

[13] Edgar C Fieller, Herman O Hartley, and Egon S Pearson, 'Tests for rank correlation coefficients. i', *Biometrika*, **44**(3/4), 470–481, (1957).

[14] Joao Gama, *Knowledge discovery from data streams*, CRC Press, 2010.

[15] Michael Hahsler and Matthew Bolaños, 'Clustering data streams based on shared density between micro-clusters', *IEEE Transactions on Knowledge and Data Engineering*, **28**(6), 1449–1461, (2016).

[16] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li, 'ADASYN: Adaptive synthetic sampling approach for imbalanced learning', in *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328. IEEE, (2008).

[17] Haibo He and Edwardo A. Garcia, 'Learning from imbalanced data', *IEEE Transactions on Knowledge and Data Engineering*, **21**(9), 1263–1284, (2009).

[18] Sture Holm, 'A simple sequentially rejective multiple test procedure', *Scandinavian Journal of Statistics*, **6**(2), 65–70, (1979).

[19] Geoff Hulten, Laurie Spencer, and Pedro Domingos, 'Mining time-changing data streams', in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 97–106, (2001).

[20] Varsha S Khandekar and Pravin Srinath, 'Non-stationary data stream analysis: state-of-the-art challenges and solutions', in *Proceeding of International Conference on Computational Science and Applications*, pp. 67–80. Springer, (2020).

[21] Mateusz Lango and Jerzy Stefanowski, 'What makes multi-class imbalanced problems difficult? An experimental study', *Expert Systems with Applications*, **199**, 116962, (2022).

[22] Joffrey L Leevy, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya, 'A survey on addressing high-class imbalance in big data', *Journal of Big Data*, **5**(1), 1–30, (2018).

[23] Nan-Ying Liang, Guang-Bin Huang, Paramasivan Saratchandran, and Narasimhan Sundararajan, 'A fast and accurate online sequential learning algorithm for feedforward networks', *IEEE Transactions on neural networks*, **17**(6), 1411–1423, (2006).

[24] Minlong Lin, Ke Tang, and Xin Yao, 'Dynamic sampling approach to training neural networks for multiclass imbalance classification', *IEEE Transactions on Neural Networks and Learning Systems*, **24**(4), 647–660, (2013).

[25] Bilal Mirza, Zhiping Lin, Jiuwen Cao, and Xiaoping Lai, 'Voting based weighted online sequential extreme learning machine for imbalance multi-class classification', in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 565–568, (2015).

[26] Bilal Mirza, Zhiping Lin, and Kar-Ann Toh, 'Weighted online sequential extreme learning machine for class imbalance learning', *Neural processing letters*, **38**(3), 465–486, (2013).

[27] Jiongming Qin, Cong Wang, Qinhong Zou, Yubin Sun, and Bin Chen, 'Active learning with extreme learning machine for online imbalanced multiclass classification', *Knowledge-Based Systems*, **231**, 107385, (2021).

[28] Siqi Ren, Wen Zhu, Bo Liao, Zeng Li, Peng Wang, Keqin Li, Min Chen, and Zejun Li, 'Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning', *Knowledge-Based Systems*, **163**, 705–722, (2019).

[29] V. M. A. Souza, D. M. Reis, A. G. Maletzke, and G. E. A. P. A. Batista, 'Challenges in benchmarking stream learning algorithms with real-world data', *Data Mining and Knowledge Discovery*, **34**, 1805–1858, (2020).

[30] Jerzy Stefanowski, 'Classification of multi-class imbalanced data: Data difficulty factors and selected methods for improving classifiers', in *International Joint Conference on Rough Sets*, pp. 57–72. Springer International Publishing, (2021).

[31] Chi-Man Vong, Jie Du, Chi-Man Wong, and Jiu-Wen Cao, 'Postboosting using extended G-mean for online sequential multiclass imbalance learning', *IEEE Transactions on Neural Networks and Learning Systems*, **29**(12), 6163–6177, (2018).

[32] Boyu Wang and Joelle Pineau, 'Online bagging and boosting for imbalanced data streams', *IEEE Transactions on Knowledge and Data Engineering*, **28**(12), 3353–3366, (2016).

[33] Shuo Wang, Leandro L. Minku, and Xin Yao, 'Resampling-based ensemble methods for online class imbalance learning', *IEEE Transactions on Knowledge and Data Engineering*, **27**(5), 1356–1368, (2015).

[34] Shuo Wang, Leandro L Minku, and Xin Yao, 'Dealing with multiple classes in online class imbalance learning.', in *IJCAI*, pp. 2118–2124, (2016).

[35] Shuo Wang, Leandro L. Minku, and Xin Yao, 'A systematic study of online class imbalance learning with concept drift', *IEEE Transactions on Neural Networks and Learning Systems*, **29**(10), 4802–4821, (2018).

[36] Frank Wilcoxon, 'Individual comparisons by ranking methods', in *Breakthroughs in statistics*, 196–202, Springer, (1992).