An active learning budget-based oversampling approach for partially labeled multi-class imbalanced data streams

Gabriel J. Aguiar
Virginia Commonwealth University
Richmond, Virginia
aguiargj@vcu.edu

Alberto Cano
Virginia Commonwealth University
Richmond, Virginia
acano@vcu.edu

ABSTRACT

Learning classification models from multi-class imbalanced data streams is a challenging task in machine learning. Moreover, there is a common assumption that all instances are labeled and available for the training phase. However, this is not realistic in real-world scenarios when learning from partially labeled data. In this work, we propose an active learning method based on labeling budget that can tackle multi-class imbalance data, concept drift, and limited access to labels. The proposed method combines information from budget constraints and dynamic class ratios to generate new relevant instances. We performed experiments on 18 real-world data streams and 11 semi-synthetic data streams, under different labeling budgets, in order to evaluate the performance of the proposed method under a varied set of scenarios. The experimental study showed that our oversampling method was able to improve the performance of state-of-the-art classifiers for multi-class imbalanced data streams under strict budgets and outperforms previously proposed oversampling methods in the domain.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**;

KEYWORDS

Machine Learning, Data Streams, Imbalanced Learning, Active Learning

ACM Reference Format:
Gabriel J. Aguiar and Alberto Cano. 2023. An active learning budget-based oversampling approach for partially labeled multi-class imbalanced data streams. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27 - March 31, 2023, Tallinn, Estonia.* ACM, Tallinn, Estonia, Article 4, 8 pages. https://doi.org/10.1145/3555776.3577624

1 INTRODUCTION

Our ability to collect and analyze data has immensely increased in recent years. This growth has created challenges for traditional machine learning methods originally designed to learn from static data. On the other hand, modern data sources are characterized by producing continuous data in high volume and velocity. Such a scenario is known as a data stream [2, 13, 17] which can be defined as an ordered sequence of instances arriving at a learning system.

The online and evolving nature of data streams makes it essential for the classifiers to adapt and learn from new concepts that emerge over time. This phenomenon is known as concept drift [15]. If not detected and tackled correctly, it will inevitably lead to poorer predictive performance, as knowledge learned from older concepts may not be useful anymore for recent instances.

Besides concept drift, another big challenge lies in the need for a learning algorithm to display robustness to class imbalance. Solutions that rely on fixed data properties cannot be applied successfully in the streaming scenario since streams may oscillate between different degrees of imbalance, definitions of classes can change, and class roles may switch. Imbalanced streams can also display other data difficulties such as small sample sizes, borderline and rare instances, overlapping among classes, or noisy labels. Imbalanced data streams are handled via class resampling, algorithm adaptation, or ensembles [1].

Furthermore, since data streams are potentially unbounded, it is impossible to provide ground truth for every new instance [34]. Therefore, we must work with sparsely labeled data streams in real-world scenarios. The lack of access to class labels will likely lead to underfitting. Hence the classifier has to use a wise approach to query the label of the instances. The most popular approach to select instances to label is Active Learning [23, 30, 34] while considering the presence of concept drift. Active Learning focuses on finding valuable data instances that should be labeled to improve the fitting of the model [24, 27]. There are several ways of defining useful instances. It can be based on the uncertainty of a classifier, randomly, the error of a classifier, etc. However, using Active Learning alone for highly limited budgets can be a problem since the selected instances may still not be enough to learn due to dynamic class boundaries. Also, if we consider high imbalance scenarios, the probability of choosing an example from the majority class is higher, leading to poorer predictive performance of the minority classes.

In the literature, there are recent works to handle imbalanced data streams [1, 22]. However, they mainly focus on fully-labeled binary data streams [3, 21, 29]. On the other hand, not much research has been done in the context of imbalanced multi-class data streams, which are much more interesting since relationships among the classes may vary over time, new classes can emerge, and old ones disappear. Some researchers tackle this problem by decomposing the multi-class problem into many binary subproblems; however, since the relationships among classes are not well-defined, inevitably, valuable information may be lost [22, 32]. The majority of algorithms dedicated to multi-class imbalanced streams assume fully-labeled

SAC '23, March 27 - March 31, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03.
https://doi.org/10.1145/3555776.3577624

data. They either focus on changing class ratios without concept drift or handling concept drift with static imbalance ratios [11, 31].

Regarding methods that account for limited access to labels, two robust ensembles have been proposed to tackle problems related to network traffic data streams [25, 26], which have their intrinsic difficulties and, therefore, may not be suitable for general-purpose data streams. These ensembles were not designed using one specific resampling and combining different active learning approaches. Korycki and Krawczyk [20] proposed an oversampling technique to tackle this problem. It combines active learning with online oversampling and uses the current imbalance ratio in the stream and the classifier error to generate meaningful instances. However, the central core of the algorithm is based on measurements of the G-Mean metric. Since it is blind to the imbalance ratio, it can be deceived in multi-class scenarios in the presence of class imbalance and be very dissimilar to other metrics, such as Kappa. Moreover, according to the literature [1, 12, 18, 19], Kappa is more suitable for understanding the behavior of classifiers in multi-class scenarios in the presence of a high imbalance ratio.

One primary constraint to learning from sparsely labeled streams is the labeling budget, and none of the previous methods take into account the labeling budget when performing oversampling decisions. It is intuitive to think that the lower the budget, the higher the oversampling factor should be. With this in mind, in this work, we propose an online oversampling method based on labeling budgets for partially labeled non-stationary and imbalanced data streams that addresses all the previously mentioned problems in partially-labeled multi-class imbalanced data streams.

The main contributions of this paper can be summarized as:

- A novel active learning oversampling method (classifier agnostic) based on the labeling budget and dynamic imbalance ratio for multi-class sparsely labeled data streams.
- Evaluation of the proposed oversampling method with state-of-art data streams classifiers for imbalanced data.
- Comparison with another oversampling strategy for sparsely labeled multi-class streaming data.

This paper is organized as follows. Section 2 introduces the proposed oversampling method. Section 3 provides the experimental setup and evaluation methodology. Section 4 presents and analyzes the results of our study. Finally, Section 5 discusses the conclusions and future work.

2 PROPOSED METHOD

This section presents the oversampling budget-based (BB) method that can tackle multi-class imbalance while learning with strict labeling budget constraints. We designed this method in order to try mimic the behavior of a fully labeled and balanced stream and present it to the classifier. Algorithm 1 presents our method, which is divided into three main parts. Firstly, we need to decide when to require the ground truth label, and that should be done in a way that we also can dynamically estimate class imbalance. After that, a balancing strategy is used to compute how many instances will be synthetically generated. Finally, we generate the instances and use them to train the classifier. All of these parts are discussed in the following.

Algorithm 1: Proposed active learning method to handle class imbalance in multi-class scenarios.

Data: Data Stream s, Labeling Budget B, Budget Spending $\hat{b}$, Class Proportions cp, Uniform Distribution υ, Balancing Ratio BR, Synthetic Instances SI
Result: Classifier C
Initialization: $\hat{b} \leftarrow 0$, $cp \leftarrow []$, $SI \leftarrow []$;
while $x \leftarrow s.nextInstance()$ **do**
  $rand \leftarrow \upsilon(0, 1)$;
  if $rand \leq B$ **then**
    request true label y of instance x;
    update cp;
    train classifier C with x;
    $BR \leftarrow$ `BalancingStrategy` (y, cp, B);
    $SI \leftarrow$ `InstanceGeneration` (x, BR);
    for $i \leftarrow 1$ **to** *SI.size()* **do**
      train classifier C with $(SI[i], c)$

2.1 Label acquisition

Since the data instances are sparsely labeled, it is difficult to estimate the actual class proportions in the stream. To estimate class imbalance while respecting the labeling budget constraint, we cannot use uncertainty strategies to decide when we should request a label. This is because uncertainty strategies are based on the uncertainty of the classifier in predicting the instance. By doing so, we would have access only to more uncertain instances, thus estimating the class proportions in the area of uncertainty. However, this would not provide us with a complete perspective of the global class proportions in the whole of the stream. Therefore, we decided to use a random selection strategy based on a uniform distribution to randomly sample the class label of the instance; if the value is smaller than the budget, the classifier will request the label for the instance. Using this strategy, the estimated class proportions among selected instances are statistically representative of the complete stream proportions in the long run.

One of the main issues with data streams is their non-stationary behavior, which means that after a drift, old information about class proportions should not be considered for newly arrived instances. To estimate dynamic class proportions, when a given instance (x, y) arrives, we update class proportions by using Algorithm 2 in a prequential fashion, where $\theta \in [0,1]$ controls how much from the past will be remembered when learning for a new instance. The lower θ is, the higher the weight for new instances.

Algorithm 2: Prequential dynamic class imbalance.

Data: Number of classes c, Class Proportions cp, Decay Factor θ
for $i \leftarrow 1$ **to** c **do**
  if $i = y$ **then**
    $cp[i] \leftarrow \theta * cp[i] + (1 - \theta)$;
  else
    $cp[i] \leftarrow \theta * cp[i]$;

2.2 Balancing strategies

This module is responsible for defining how many instances will be generated to train the classifier. Here we want to define a function that will consider the imbalance ratios and be aware of the labeling budget constraints. Suppose we have an instance from the majority class but are on a strict budget. In that case, it is necessary to have a higher level of oversampling due to the number of instances that were not labeled and, therefore, the model ignored. Therefore we propose the following function:

$$f(c, cp, B) = \frac{1}{B} * (1 - cp[c]) \quad (1)$$

where c is the class of a given instance, cp is the previously computed class proportions and B is the labeling budget. With this function which is inversely proportional to the budget, we can mimic the behavior of a fully labeled stream. Moreover, when combined with the current imbalance ratio, we can increase the number of samples generated for minority classes dynamically.

2.3 Instance generation

Given a balancing ratio BR computed in the previous module, we employed two generative methods to create BR additional instances. They are based on oversampling methods proposed for fully labeled imbalanced data streams.

Resampling (BB-RE) - an online approach that presents BR times the same instance x to the classifier. This method was inspired by state-of-the-art ensembles that uses resampling in order to tackle imbalanced streams.

SMOTE (BB-SM) - new instances are generated using SMOTE [8] from a sliding window of the w latest instances for each class. For a given instance x, we join all windows, find its k nearest neighbors and generate synthetic samples using the neighbors from the same class. This procedure is described in Algorithm 3.

Algorithm 3: Algorithm to generate artificial instances.

```
Data: Instance (x, y), Class Window w_c, Uniform
      Distribution v, Nearest Neighbors nn, Number of
      neighbors k, Synthetic Sample ss
Initialization: k ← 10, nn ← [] ;
W ← w_1 ∪ w_2 ··· ∪ w_c;
nn ← k-Nearest Neighbors (x, W, k);
for i ← 1 to BR do
    randomIndex ← v(1, k);
    neighbor ← nn[randomIndex];
    gap ← v(0, 1);
    for attr ← 1 to x.numAttributes() do
        diff ← x[attr] − neighbor[attr];
        ss[attr] ← x[attr] + gap * diff;
    train classifier C with ss;
```

3 EXPERIMENTAL SETUP

The experiments were designed to evaluate the performance of the proposed method under varied multi-class imbalanced scenarios and difficulties. We aim to understand in which scenarios our oversampling technique would improve the performance of the classifiers. The following research questions (RQ) were addressed:

- **RQ1**: Is the proposed oversampling strategy able to improve the performance of classifiers under different budgets?
- **RQ2**: Does our framework works for classifiers with different learning mechanisms?
- **RQ3**: Is the proposed framework competitive, in terms of classification performance with previously proposed methods for multi-class sparsely labeled streams?

3.1 Algorithms

Since our method is classifier agnostic, we selected 8 algorithms for data streams in order to compare the performance with and without the proposed method. We selected top performing methods in the state of the art with internal mechanisms to deal with imbalanced streams, general-purpose ensembles and tree-based algorithms. The algorithms and their references are presented in Table 1. All algorithms are implemented in MOA [6]. The source code of the algorithms and the experiments are publicly available on GitHub to facilitate the reproducibility of this research. All algorithms use the parameter settings recommended by their authors. Detailed information about the specific parameters configuration is available on the GitHub repository[1].

Table 1: Classifiers evaluated in the experiments.

Algorithm	Reference
GHVFDT	Lyon et al. [28]
HDVFDT	Cieslak and Chawla [9]
ROSE	Cano and Krawczyk [7]
OzaBag	Bifet et al. [5]
LB	Bifet et al. [4]
ARF	Gomes et al. [16]
MOOB	Wang et al. [33]
MUOB	Wang et al. [33]

3.2 Baseline

As baselines we use the selected classifiers without oversampling and applying two active learning (AL) techniques: (i) Random Selection and (ii) Random Variable Uncertainty. Besides that, we used a method that uses a fixed weight for resampling (FR). Those baselines will help us understand where our resampling method works better or when our weighting methodology may overestimate the number of samples to train the model, and also whether it is possible to understand when only the selection of the most suitable instance will work better than resampling.

We compared our results with another oversampling technique for sparsely labeled data streams, namely OSAMP [20]. We used the publicly available implementation on Github [2] relying on its

[1]https://github.com/canoalberto/budgetbasedoversampling
[2]https://github.com/lkorycki/osamp-moa

Table 2: Real-world multi-class datasets specifications.

Dataset	Instances	Features	Classes	Drift
activity	5,418	45	6	✓
connect-4	67,557	42	3	-
cov-pok-elec	1,455,525	72	10	✓
covtype	581,012	54	7	-
crimes	878,049	3	39	-
fars	100,968	29	8	-
gas	13,910	128	6	✓
hypothyroid	1,000,000	29	4	-
kddcup	4,898,431	41	23	✓
kr-vs-k	28,056	6	18	✓
lymph	1,000,000	18	4	✓
olympic	271,116	7	4	-
poker	829,201	10	10	✓
sensor	2,219,803	5	57	✓
shuttle	57,999	9	7	✓
tags	164,860	4	11	-
thyroid	7,200	21	3	-
zoo	1,000,000	17	7	✓

default hyperparameters recommended by its authors. OSAMP was chosen due to their clearly similarity with our method, being an oversampling method based on synthetic samples and error rates.

3.3 Experimental configuration

In our experiments we evaluated all the algorithms under 8 different budget sizes: {50%, 20%, 10%, 5%, 1%, 0.5% and 0.1%}. Regarding the proposed method, two hyperparameters need to be configured: (i) number of neighbors (k) and (ii) sliding window size (w). For k we used 10 for every budget size, but for w we varied the window size according to the budget size, *i.e.*, $w \in \{500, 200, 100, 50, 10, 10, 10\}$. The reported hyperparameters were chosen after empirical evaluation that can be verified in the github repository. Algorithms were run on a GNU/Linux cluster with 192 Intel Xeon cores, 6 TB RAM, and Centos 7.

3.4 Datasets

To address the research questions we selected 18 real datasets widely used in the data streams domain to evaluate classifiers in the multi-class scenario. Their charactericts and specifications are presented in Table 2. Datasets without the check-mark are not necessarily stationary streams but it is unknown about drift. We decided to evaluate in real-world datasets because they present unique and challenging conditions, such as not well defined relationship among classes, that help us to gain insights about classifiers under those conditions. These datasets were collected in order to model a specific behavior and do not hold clear probabilistic mechanisms such as stream generators.

Moreover, to evaluate our method in specific scenarios such as dynamic imbalance ratio and concept drift, we used the 11 synthetic data streams provided by Korycki and Krawczyk [20]. Figure 1 shows the imbalance ratio in the stream over time. The semi-synthetic streams pose different challenges and are even harder than only the real ones, because they represent the most difficult scenarios we can encounter, *i.e.*, dynamic class ratios with concept drift at the same time.

3.5 Evaluation

In order to evaluate the classifiers we followed the recommendations for multi-class scenarios and used Kappa as our evaluation metric. In binary settings, the G-Mean is widely used for evaluation, however for multi-class scenarios, it introduces the problem that as soon as the recall for one class is 0 the product of the whole geometric mean becomes 0, therefore, Kappa is more suitable. Kappa is used to evaluate classifiers in imbalanced setting. It evaluates the classifier performance by computing the inter-rater agreement between the successful predictions and the statistical distribution of the data classes, as Eq. 2.

$$Kappa = \frac{n\sum_{i=1}^{c} x_{ii} - \sum_{i=1}^{c} x_{i.}x_{.i}}{n^2 - \sum_{i=1}^{c} x_{i.}x_{.i}} \qquad (2)$$

where x_{ii} is the count of cases in the main diagonal of the confusion matrix, n is the number of examples, c is the number of classes, and $x_{.i}$, $x_{i.}$ are the column and row total counts, respectively. Kappa punishes homogeneous predictions, which is very important to detect in imbalanced scenarios but can be too drastic in penalizing misclassifications on difficult data. Moreover, Kappa provides better insights in detecting changes in the distribution of classes in multi-class imbalanced data. Metrics were computed prequentially [14] using a sliding window of 500 examples.

4 RESULTS AND DISCUSSION

The results were organized by comparing the results in both types of data streams and asesssing the performance of a classifier with and without our oversampling methods. Afterwards, we analyze how the internal learning mechanisms of the base learners affect how our proposed method improves (or does not) their performance. Then, we compare our results with OSAMP in order to evaluate how competitive our method is regarding previously proposed oversampling methods. Finally, we analyze how our method affects the runtime of each base-learner.

Improving base learners. The average results for all algorithms are presented in Tables 3 and 4. Looking at the overall results, we can see that the Kappa metric for almost every base learner is higher for the oversampling version than for the base learner with only an active learning method.

Regarding the real data streams, we can see that the only classifier that was not improved in any scenario was MUOB; the reasons for it will be discussed further in this section. Besides that, ROSE with a labeling budget lower than 0.5% and MOOB for $B = 20\%$ and $B = 10\%$ were the other scenarios that the proposed method was unable to improve the base classifier. Considering all possible configurations, we can see that only for a budget lower than 0.5%, the best overall result was not achieved by either BB-SM or BB-RE. In this case, the best result was achieved by ROSE, displaying its excellent ability to deal with imbalanced streams. Also, it is worth mentioning that some real data streams do not have a dynamic imbalance ratio or concept drift, which helps classifiers to learn without the oversampling method.

When we look at the results for the semi-synthetic streams, in which we know explicitly that there are presence of concept drift and dynamic imbalance ratios. We can see clearly how these factors deteriorate the performance of the classifiers, which displayed lower

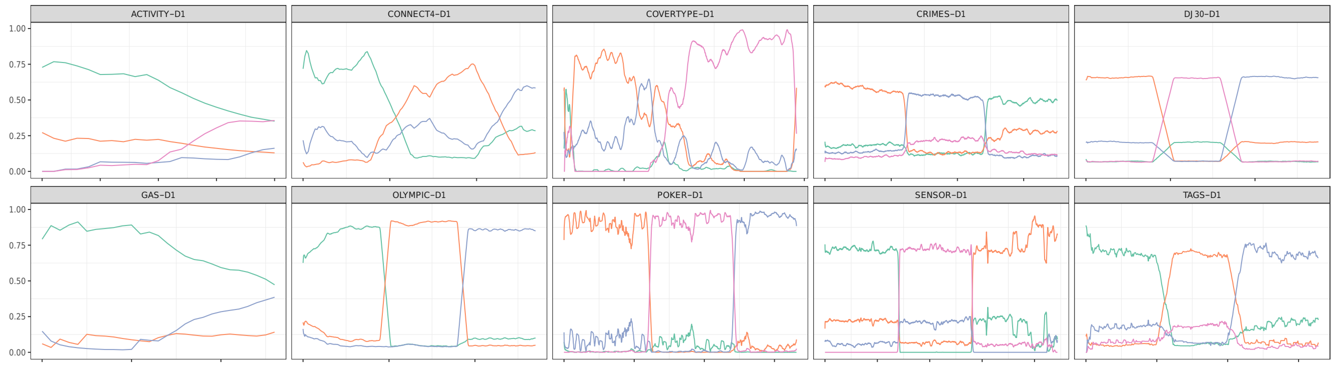

Figure 1: Dynamic class ratios for the multi-class semi-synthetic data streams. Each color represents a different class.

Table 3: Kappa averages of all real streams for all base learners under different budgets. The best values for the same base learners are highlighted and best values for each budget value is underlined.

Classifier	Strategy	50%	20%	10%	5%	1%	0.5%	0.1%
GHVFDT	AL	0.2733	0.2605	0.2352	0.2553	0.1595	0.1301	0.0953
	BB-SM	**0.2940**	**0.2943**	**0.2641**	0.2592	0.1577	0.1293	0.0791
	BB-RE	0.2741	0.2588	0.2266	**0.2604**	**0.1727**	**0.1387**	**0.1012**
	FR	0.2756	0.2178	0.2158	0.2283	0.1512	0.1278	0.0918
HDVFDT	AL	0.2502	0.2270	0.2275	0.2457	0.1520	0.1245	0.0953
	BB-SM	**0.2679**	**0.2679**	**0.2499**	**0.2522**	0.1534	0.1213	0.0699
	BB-RE	0.2537	0.2303	0.2187	0.2460	**0.1640**	**0.1343**	**0.1012**
	FR	0.2575	0.2097	0.2165	0.2327	0.1446	0.1195	0.0885
ROSE	AL	0.2843	0.2695	0.2591	0.2388	0.1601	**0.1654**	**0.1061**
	BB-SM	**0.2890**	**0.2971**	**0.2884**	**0.2530**	0.1432	0.0935	0.0362
	BB-RE	0.2539	0.2798	0.2636	0.2191	0.1573	0.1417	0.1035
	FR	0.2510	0.2570	0.2537	0.2385	**0.1715**	0.1398	0.1026
OzaBag	AL	0.3116	0.3173	0.2887	0.2666	0.1591	0.1184	0.0753
	BB-SM	**0.3419**	**0.3424**	**0.3078**	**0.2843**	0.1378	0.1084	0.0450
	BB-RE	0.3152	0.3117	0.3058	0.2787	**0.1654**	0.1289	**0.0816**
	FR	0.2797	0.2443	0.2408	0.2523	0.1581	**0.1290**	0.0737
LB	AL	0.3121	0.2936	0.2715	0.2843	0.1621	0.1215	0.0713
	BB-SM	**0.3226**	**0.3089**	**0.2756**	0.2642	0.1285	0.1016	0.0395
	BB-RE	0.3060	0.2949	0.2699	**0.2780**	**0.1694**	**0.1389**	**0.0789**
	FR	0.2850	0.2724	0.2537	0.2478	0.1616	0.1341	0.0724
ARF	AL	0.3164	0.2849	0.2678	0.2596	0.1601	0.1377	0.0956
	BB-SM	**0.3340**	**0.3238**	**0.2827**	**0.2654**	0.1476	0.1042	0.0612
	BB-RE	0.3090	0.2842	0.2724	0.2601	**0.1663**	**0.1465**	**0.0991**
	FR	0.2825	0.2589	0.2327	0.2378	0.1500	0.1343	0.0987
MOOB	AL	**0.3022**	**0.3187**	**0.3043**	0.2811	0.1681	0.1298	**0.0791**
	BB-SM	0.3010	0.3048	0.3022	0.2870	0.1636	0.1297	0.0614
	BB-RE	0.3008	0.2770	0.2635	**0.2912**	**0.1777**	**0.1410**	0.0779
	FR	0.2950	0.2616	0.2519	0.2487	0.1657	0.1306	0.0750
MUOB	AL	**0.1444**	**0.1427**	**0.1258**	**0.1104**	**0.0892**	**0.0851**	**0.0655**
	BB-SM	0.1407	0.1331	0.1171	0.1067	0.0718	0.0380	0.0004
	BB-RE	0.1315	0.1230	0.1112	0.0920	0.0776	0.0842	0.0505
	FR	0.1077	0.0918	0.0833	0.0913	0.0773	0.0532	0.0309
OSAMP	SM-DHR	0.0850	0.0823	0.0756	0.0651	0.0553	0.0180	0.0060

Kappa values on average than with real data streams. MUOB still did not improve when combined with our oversampling methods. Also, ROSE did display better results without oversampling when B = 20%, 10% and 5%. Besides those particular cases, all the other base learners presented better results when combined with oversampling. Moreover, the best performance by each budget when classifying the semi-synthetic streams were achieved by combining a base learner and the proposed method. ARF combined with BB-SM displayed the highest Kappa values for budgets between 1% and 50%. In contrast, ROSE with BB-SE achieved the best results with more strict budgets (0.5% and 0.1%).

Table 4: Kappa averages of all semi-synthetic streams for all base learners under different budgets. The best values for the same base learners are highlighted and best values for each budget value is underlined.

Classifier	Strategy	50%	20%	10%	5%	1%	0.5%	0.1%
GHVFDT	AL	0.1821	0.1444	0.1420	0.1167	0.0991	0.0672	0.0131
	BB-SM	0.2033	0.1611	0.1608	**0.1401**	0.0927	0.0784	**0.0192**
	BB-RE	**0.2262**	0.1827	0.1559	0.1347	**0.0951**	**0.0886**	0.0151
	FR	0.2201	**0.1954**	**0.1732**	0.1140	0.0876	0.0816	0.0060
HDVFDT	AL	0.1673	0.1506	0.1384	0.1118	0.0969	0.0831	0.0136
	BB-SM	0.1755	0.1645	0.1743	**0.1462**	**0.1092**	0.0795	**0.0355**
	BB-RE	0.2093	0.1636	0.1622	0.1212	0.1001	**0.0931**	0.0161
	FR	**0.2249**	**0.1851**	**0.1762**	0.1264	0.0863	0.0675	0.0048
ROSE	AL	0.3553	**0.3314**	**0.2933**	**0.2578**	0.1447	0.1289	0.0248
	BB-SM	**0.3737**	0.3221	0.2852	0.2393	0.1560	0.0913	0.0283
	BB-RE	0.3545	0.3184	0.2853	0.2533	0.1546	**0.1479**	**0.0467**
	FR	0.3268	0.2836	0.2703	0.2314	**0.1621**	0.1290	0.0241
OzaBag	AL	0.2461	0.2034	0.1884	0.1555	0.1050	0.0893	0.0076
	BB-SM	**0.2879**	**0.2454**	**0.2437**	**0.2180**	**0.1391**	**0.1088**	**0.0375**
	BB-RE	0.2843	0.2358	0.2101	0.1777	0.1233	0.0925	0.0192
	FR	0.2770	0.2150	0.2030	0.1780	0.0995	0.0807	0.0070
LB	AL	0.3337	0.2713	0.2236	0.1774	0.1166	0.0959	0.0215
	BB-SM	**0.3505**	0.2896	**0.2667**	**0.2385**	**0.1477**	**0.1102**	**0.0380**
	BB-RE	0.3459	**0.3124**	0.2619	0.2107	0.1365	0.1046	0.0301
	FR	0.3293	0.2688	0.2583	0.1981	0.1135	0.0824	0.0166
ARF	AL	0.3735	0.3109	0.2718	0.2065	0.1226	0.1077	0.0359
	BB-SM	**0.4102**	**0.3767**	**0.3398**	**0.2864**	**0.1626**	0.1161	0.0347
	BB-RE	0.3898	0.3441	0.2999	0.2581	0.1358	**0.1285**	**0.0359**
	FR	0.3573	0.3135	0.2722	0.2328	0.1448	0.1046	0.0284
MOOB	AL	0.2531	0.1871	0.1794	0.1433	0.0809	0.0747	0.0194
	BB-SM	0.2577	0.1956	0.1862	0.1525	**0.1056**	**0.0834**	**0.0433**
	BB-RE	0.3094	0.2161	0.1932	**0.1599**	0.0869	0.0723	0.0196
	FR	**0.3312**	**0.2606**	**0.2081**	0.1480	0.0820	0.0628	0.0396
MUOB	AL	**0.0691**	**0.0856**	**0.1016**	0.0887	**0.0431**	**0.0272**	0.0004
	BB-SM	0.0675	0.0651	0.0704	**0.0948**	0.0292	0.0064	0.0015
	BB-RE	0.0612	0.0723	0.0942	0.0746	0.0378	0.0193	-0.0012
	FR	0.0535	0.0595	0.0695	0.0735	0.0324	0.0102	**0.0038**
OSAMP	SM-DHR	0.1755	0.1638	0.1487	0.1238	0.0627	0.0678	0.0328

All things considered, it was possible to address **RQ1** and state that our proposed method could improve the performance of the classifiers under different budgets when learning from drifting and imbalanced data streams.

Learning mechanisms. After addressing **RQ1**, we could see that some base learners had a greater improvement than others, while MUOB which did not have any improvement when combined with our method. Figures 2 and 3 present the relative performance of our proposed method relative to its version without oversampling, and this difference among base learners is clear.

HDVFDT is a tree-based online classifier without any resampling mechanisms to deal with imbalanced streams, and it experienced an improvement of 25% on average, and BB-SM made it 160% better for $B = 0.01\%$ in semi-synthetic streams. ARF, a general-purpose ensemble, also significantly improved when combined with our oversampling method. However, ROSE, which is also an ensemble, but that was designed to tackle class imbalance in general, does not have a better performance with the oversampling mechanisms. This happens because ROSE does resampling internally; therefore, as we create new samples, ROSE learning mechanisms will not behave the same way, which is why the performance with and without oversampling is very similar. Only when the budget is very strict (0.1%), our oversampling method could improve ROSE performance, which is mainly related to the number of samples that the classifier without oversampling would have access to.

The last two ensembles are similar, with the main difference in how they deal with the imbalance problem. MOOB had a behavior similar to ROSE. When combined with the proposed oversampling method, the performance was not significantly better due to their built-in oversampling mechanism. On the other hand, MUOB had the worst relative performance and also the worst performance overall. This poor performance is due to combining two concurrent methods without any balance. On one side, our proposed method was oversampling based on a budget and class imbalance, while internally, the base learner is undersampling the majority class based on the imbalance ratio, which was based on synthetic samples. This may generate a lot of noise and degenerate the classifier's overall performance.

In summary, we can conclude that the proposed oversampling method works better with base learners that are not explicitly designed for imbalanced scenarios, *i.e.*, general-purpose ensembles, tree-based classifiers, and so on. This can also be seen in the overall results, where ARF and OzaBag are among the best classifiers. It can also improve the performance of classifiers designed for the specific scenario, but not significantly. Finally, we could analyze how classifiers with different learning mechanisms behave and answer **RQ2**.

Oversampling methods. Besides comparing our method with only the base learner without oversampling, we evaluated how our method performs relative to OSAMP, another oversampling method for multi-class sparsely labeled streams.

Figures 2 and 3 also display the relative performance of our method relative to OSAMP. Regarding the real data streams, the performance gap is big, with even MUOB, which did not present good results relative to the plain base learner, achieving more than 60% of improvement. For stringent budgets, *i.e.*, 0.05% and 0.01%, the performance of our proposed method was more than seven times better than OSAMP for all evaluated base learners. It is important to mention that the OSAMP method was designed mainly to deal with scenarios presented in the semi-synthetic streams.

Regarding the semi-synthetic streams, for budgets bigger than 0.5%, all base learners but MUOB could achieve better performance than OSAMP. HDVFDT with BB-SE got its better result with $B = 1\%$ and performing 56% better than OSAMP. ROSE and ARF got the best relative performance up to 150% of improvement. MUOB displayed the worst relative performance in this scenario. One may notice that when $B = 0.5\%$ and 0.1%, the performance gap decreases; this is related to the number of generated samples. The OSAMP method generates more additional samples than our proposed method. Therefore, more samples when the budget is more limited may help the classifier, but still, BB-SM performs better than OSAMP for 6 out of 8 base learners.

Finally, to address **RQ3**, we confirm the superiority of BB-SM and BB-SE regarding the baselines by statistical tests. We used the Friedman test, with a significance level of $\alpha = 0.05$. The null hypothesis is that the proposed method and the baselines are similar. Anytime the null hypothesis is rejected, the Nemenyi post hoc test can be applied, stating that the performance of the two approaches is significantly different if their corresponding average ranks differ by at least a Critical Difference (CD) value. When multiple algorithms are compared, a graphic representation can be used to represent the results with the CD diagram, as proposed by Demšar [10].

BB-RE and BB-SM were compared to Active Learning and OSAMP over all the datasets under all different budgets. This analysis is shown in Figure 4, using the results from the Nemenyi test. In this diagram, if the lines are connected, it means that they are similar, and there is no statistical difference. We can see that the performance of all base-leaners, except for MUOB, the oversampling methods, or simply the plain base learner performed better than OSAMP, supporting the hypothesis that our method performs better than OSAMP. Also, we can see that for HDVFDT and ARF, and there is a statistical difference between our method and Active Learning. At the same time, ROSE and MOOB are similar, supporting that our method works better for general-purpose classifiers.

Time consumption. One of the main constraints when dealing with data streams is time consumption. Table 5 presents the average running time of each base learner under different budgets for all datasets. It was expected that our method would increase the running time since we are generating new samples and training the models with them. As we can see in the results for high budget values as 50% or 20%, the proposed method is 24 and 7, respectively, times slower than the plain classifier for both generation methods. However, if we are looking for a more real-world scenario, where labeling 50% of our samples is not reliable, we should look at more limited budgets, and for budgets smaller than 10%, BB-SM is on average only 2.5 times slower, and BB-RE 1.5 times slower, therefore, if we consider that the proposed oversampling method can improve the performance of base classifiers. We are dealing with sparsely labeled streams. Consequently, we will have a bigger interval between training steps, and sacrificing update time to improve performance, is a valid choice.

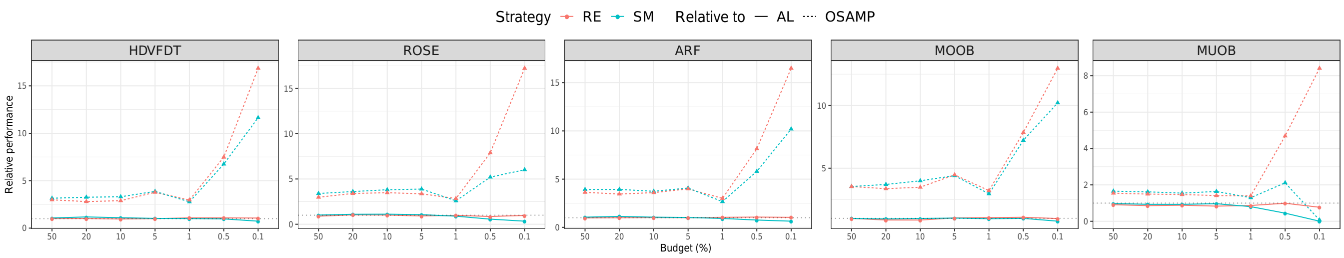

Figure 2: Relative performance comparison between our proposed methods and the defined baselines for a given budget for real data streams. Values greater than 1 indicates better relative performance.

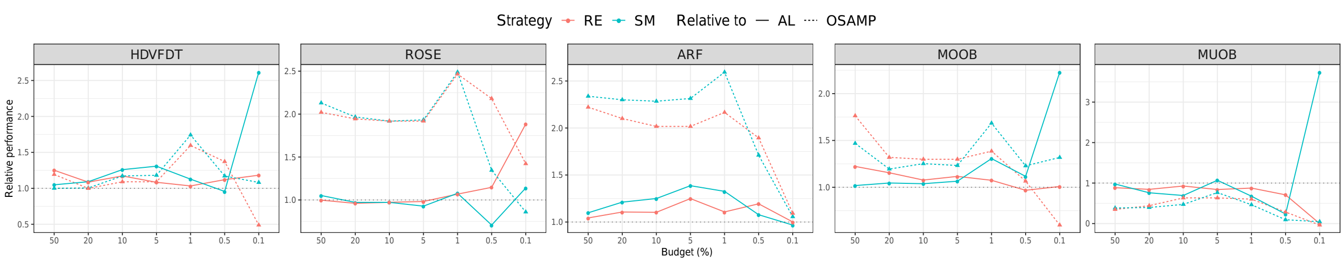

Figure 3: Relative performance comparison between our proposed methods and the defined baselines for a given budget for semi-synthetic data streams. Values greater than 1 indicates better relative performance.

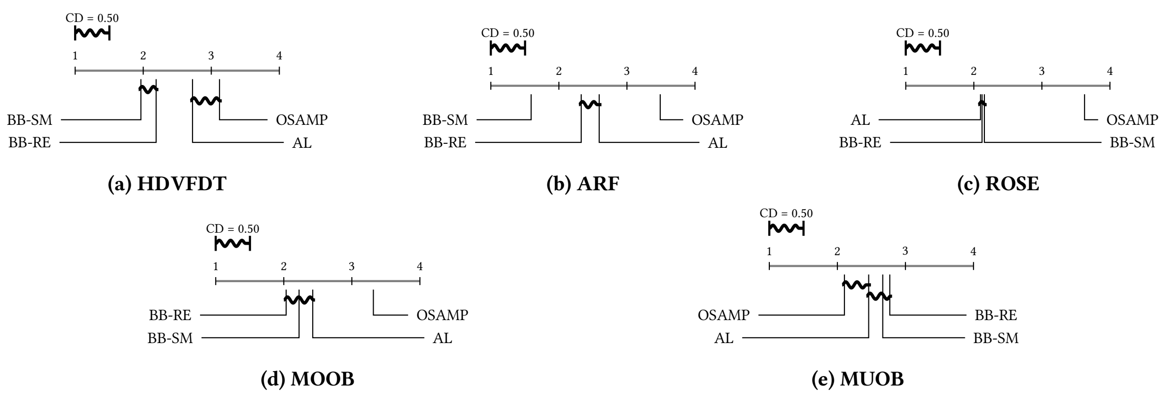

Figure 4: Comparison of the Kappa values obtained by each model (and methodology) according to the Nemenyi test. Groups of algorithms that are not significantly different ($\alpha = 0.05$ and $CD = 0.50$) are connected.

5 CONCLUSION

In this paper, we presented an oversampling method based on labeling budget and imbalance ratio that is classifier agnostic to address the problem of learning from multi-class imbalanced data streams under labeling budget restrictions combined with dynamic class ratios and concept drift. We performed experiments under a wide range of budgets, from very loose to very tight, in a plethora of real-world and semi-synthetic data streams. We compared classifiers with different types of learning mechanisms in order to evaluate how our proposed method would work when combined with state-of-the-art classifiers.

Our experiments demonstrated that the proposed method could improve the performance of general-purpose streaming classifiers and ensembles while being competitive in terms of running time. Also, it dealt with very restricted budgets achieving up to 100% of improvement under budgets of 0.1%. Moreover, compared to other oversampling methods for multi-class data streams, the performance gap was even more significant, around 1500%, on stringent budgets.

Finally, in future work, we intend to improve our oversampling mechanism to detect which samples are more valuable to oversample than others to improve the performance of classifiers in the multi-class scenario. Besides that, we also plan to propose a robust ensemble framework that merges all the knowledge acquired from this paper

Table 5: Average running time (ms) of all base learners and their oversampling variations under different budgets.

Classifier	Strategy	50%	20%	10%	5%	1	0.5%	0.1%
GHVFDT	AL	1.91	1.38	1.20	1.10	0.97	0.93	0.90
	BB-SM	93.18	16.95	5.78	1.67	1.37	1.12	1.04
	BB-SE	92.30	17.92	6.12	1.70	1.32	1.04	0.97
HDVFDT	AL	1.94	1.37	1.19	1.12	0.94	0.92	0.91
	BB-SM	90.91	16.99	5.76	1.70	1.33	1.11	1.06
	BB-SE	91.48	17.81	5.99	1.71	1.30	1.03	0.97
ROSE	AL	44.23	19.21	11.30	6.55	2.40	1.70	1.17
	BB-SM	144.02	44.15	24.16	15.81	10.23	8.34	4.55
	BB-SE	183.72	55.71	26.54	13.27	4.19	2.61	1.42
OzaBag	AL	9.13	4.31	2.60	1.80	1.17	1.06	1.00
	BB-SM	103.86	21.80	8.84	3.82	2.63	2.20	1.77
	BB-SE	111.45	24.33	9.36	3.20	1.64	1.25	1.07
LB	AL	17.30	8.20	4.75	3.20	1.40	1.17	1.01
	BB-SM	108.78	27.44	12.70	6.97	4.26	3.42	2.37
	BB-SE	121.94	30.21	12.85	5.39	2.17	1.52	1.11
ARF	AL	274.12	114.16	62.46	34.08	8.61	5.11	1.99
	BB-SM	456.36	209.58	139.19	101.84	48.90	38.57	18.35
	BB-SE	639.68	248.36	130.67	69.78	16.54	9.19	2.93
MOOB	AL	8.29	4.02	2.68	1.87	1.14	1.05	0.98
	BB-SM	97.22	20.55	8.20	3.37	2.35	1.90	1.64
	BB-SE	106.88	23.06	8.94	3.23	1.59	1.19	1.05
MUOB	AL	1.41	1.09	1.01	0.97	0.95	0.95	0.92
	BB-SM	89.90	17.57	6.08	1.80	1.25	0.98	0.89
	BB-SE	92.56	18.07	6.19	1.71	1.29	1.01	0.94

and combines with new Active Learning methods for instance selection.

ACKNOWLEDGMENTS

High Performance Computing resources provided by the High Performance Research Computing (HPRC) Core Facility at Virginia Commonwealth University (https://hprc.vcu.edu) were used for conducting the research reported in this work.

REFERENCES

[1] Gabriel Aguiar, Bartosz Krawczyk, and Alberto Cano. 2022. A survey on learning from imbalanced data streams: taxonomy, challenges, empirical study, and reproducible experimental framework. *arXiv preprint arXiv:2204.03719* (2022).

[2] Maroua Bahri, Albert Bifet, João Gama, Heitor Murilo Gomes, and Silviu Maniu. 2021. Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11, 3 (2021).

[3] Alessio Bernardo and Emanuele Della Valle. 2022. An extensive study of C-SMOTE, a Continuous Synthetic Minority Oversampling Technique for Evolving Data Streams. *Expert Systems with Applications* 196 (2022), 116630.

[4] Albert Bifet, Geoff Holmes, and Bernhard Pfahringer. 2010. Leveraging Bagging for Evolving Data Streams. In *Machine Learning and Knowledge Discovery in Databases*. 135–150.

[5] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Richard Kirkby, and Ricard Gavaldà. 2009. New Ensemble Methods for Evolving Data Streams. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 139–148.

[6] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, and Thomas Seidl. 2010. MOA: Massive online analysis, a framework for stream classification and clustering. In *Workshop on Applications of Pattern Analysis*. 44–50.

[7] Alberto Cano and Bartosz Krawczyk. 2022. ROSE: robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Machine Learning* 111 (2022), 2561–2599.

[8] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* (2002), 321–357.

[9] David Cieslak and Nitesh Chawla. 2008. Learning decision trees for unbalanced data. In *European Conference on Machine Learning and Knowledge Discovery in Databases*. 241–256.

[10] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research* 7 (2006), 1–30.

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

[12] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. 2018. *Learning from Imbalanced Data Sets*. Springer.

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

[14] Joao Gama, Raquel Sebastiao, and Pedro Pereira Rodrigues. 2013. On evaluating stream learning algorithms. *Machine learning* 90, 3 (2013), 317–346.

[15] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 1–37.

[16] Heitor M Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfharinger, Geoff Holmes, and Talel Abdessalem. 2017. Adaptive random forests for evolving data stream classification. *Machine Learning* 106, 9 (2017), 1469–1495.

[17] Heitor Murilo Gomes, Jesse Read, Albert Bifet, Jean Paul Barddal, and João Gama. 2019. Machine learning for streaming data: state of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter* 21, 2 (2019), 6–22.

[18] Nathalie Japkowicz. 2013. Assessment metrics for imbalanced learning. *Imbalanced learning: Foundations, algorithms, and applications* (2013), 187–206.

[19] László Jeni, Jeffrey Cohn, and Fernando De la Torre. 2013. Facing Imbalanced Data - Recommendations for the Use of Performance Metrics. *Humaine Association Conference on Affective Computing and Intelligent Interaction* 2013.

[20] Łukasz Korycki and Bartosz Krawczyk. 2020. Online oversampling for sparsely labeled imbalanced and non-stationary data streams. In *International Joint Conference on Neural Networks (IJCNN)*. 1–8.

[21] Łukasz Korycki, Alberto Cano, and Bartosz Krawczyk. 2019. Active Learning with Abstaining Classifiers for Imbalanced Drifting Data Streams. In *2019 IEEE International Conference on Big Data (Big Data)*. 2334–2343.

[22] Bartosz Krawczyk. 2016. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence* 5, 4 (2016), 221–232.

[23] B. Krawczyk and A. Cano. 2019. Adaptive ensemble active learning for drifting data stream mining. In *International Joint Conference on Artificial Intelligence*. 2763–2771.

[24] Sanmin Liu, Shan Xue, Jia Wu, Chuan Zhou, Jian Yang, Zhao Li, and Jie Cao. 2021. Online active learning for drifting data streams. *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[25] Weike Liu, Hang Zhang, Zhaoyun Ding, Qingbao Liu, and Cheng Zhu. 2021. A comprehensive active learning method for multiclass imbalanced data streams with concept drift. *Knowledge-Based Systems* 215 (2021), 106778.

[26] Weike Liu, Chen Zhu, Zhaoyun Ding, Hang Zhang, and Qingbao Liu. 2022. Multiclass Imbalanced and Concept Drift Network Traffic Classification Framework Based on Online Active Learning. *SSRN* (2022), 4114383.

[27] Edwin Lughofer. 2017. On-line active learning: A new paradigm to improve practical useability of data stream modeling methods. *Information Sciences* 415 (2017), 356–376.

[28] Robert Lyon, J.M. Brooke, J.D. Knowles, and B.W. Stappers. 2014. Hellinger Distance Trees for Imbalanced Streams. In *2014 22nd International Conference on Pattern Recognition*. 1969–1974.

[29] Kleanthis Malialis, Christos Panayiotou, and Marios Polycarpou. 2022. Nonstationary Data Stream Classification with Online Active Learning and Siamese Neural Networks. *Neurocomputing* (2022).

[30] Saad Mohamad, Abdelhamid Bouchachia, and Moamar Sayed-Mouchaweh. 2016. A bi-criteria active learning algorithm for dynamic data streams. *IEEE transactions on neural networks and learning systems* 29, 1 (2016), 74–86.

[31] Farnaz Sadeghi and Herna L. Viktor. 2021. Online-MC-Queue: Learning from Imbalanced Multi-Class Streams. In *International Workshop on Learning with Imbalanced Domains: Theory and Applications*, Vol. 154. 21–34.

[32] Yu Sun, Ke Tang, Leandro L Minku, Shuo Wang, and Xin Yao. 2016. Online ensemble learning of data streams with gradually evolved classes. *IEEE Transactions on Knowledge and Data Engineering* 28, 6 (2016), 1532–1545.

[33] Shuo Wang, Leandro L. Minku, and Xin Yao. 2016. Dealing with Multiple Classes in Online Class Imbalance Learning. In *International Joint Conference on Artificial Intelligence*. 2118–2124.

[34] Indrė Žliobaitė, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes. 2013. Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems* 25, 1 (2013), 27–39.