Machine Learning
https://doi.org/10.1007/s10994-023-06353-6

A survey on learning from imbalanced data streams: taxonomy, challenges, empirical study, and reproducible experimental framework

Gabriel Aguiar[1] · **Bartosz Krawczyk**[2] · **Alberto Cano**[3]

Received: 7 April 2022 / Revised: 20 April 2023 / Accepted: 16 May 2023

Abstract
Class imbalance poses new challenges when it comes to classifying data streams. Many algorithms recently proposed in the literature tackle this problem using a variety of data-level, algorithm-level, and ensemble approaches. However, there is a lack of standardized and agreed-upon procedures and benchmarks on how to evaluate these algorithms. This work proposes a standardized, exhaustive, and comprehensive experimental framework to evaluate algorithms in a collection of diverse and challenging imbalanced data stream scenarios. The experimental study evaluates 24 state-of-the-art data streams algorithms on 515 imbalanced data streams that combine static and dynamic class imbalance ratios, instance-level difficulties, concept drift, real-world and semi-synthetic datasets in binary and multi-class scenarios. This leads to a large-scale experimental study comparing state-of-the-art classifiers in the data stream mining domain. We discuss the advantages and disadvantages of state-of-the-art classifiers in each of these scenarios and we provide general recommendations to end-users for selecting the best algorithms for imbalanced data streams. Additionally, we formulate open challenges and future directions for this domain. Our experimental framework is fully reproducible and easy to extend with new methods. This way, we propose a standardized approach to conducting experiments in imbalanced data streams that can be used by other researchers to create complete, trustworthy, and fair evaluation of newly proposed methods. Our experimental framework can be downloaded from https://github.com/canoalberto/imbalanced-streams.

Keywords Machine learning · Data stream mining · Class imbalance · Concept drift · Reproducible research

Editor: Nuno Moniz, Paula Branco, Luís Torgo, Nathalie Japkowicz, Michal Wozniak, Shuo Wang.

Extended author information available on the last page of the article

1 Introduction

Recent advancements in our ability to collect, integrate, store, and analyze big amounts of data led to the emergence of new challenges for machine learning methods. Traditional algorithms were designed to discover knowledge from static datasets. Contrary, contemporary data sources generate information characterized by both volume and velocity. Such a scenario is known as data streams (Gama, 2010; Bahri et al., 2021; Read and Žliobaitė, 2023) and traditional methods lack the speed, adaptability, and robustness to succeed.

One of the biggest challenges, when compared to learning from static data, lies in the need of adapting to the evolving nature of data, where concepts are non-stationary and may change over time. This phenomenon is called concept drift (Krawczyk et al., 2017; Khamassi et al., 2018) and leads to degradation of the classifier, as knowledge learned on previous concepts may not be useful anymore for the recent instances. Recovering from concept drift requires either the presence of explicit detectors or implicit adaptation mechanisms.

Another vital challenge in data stream mining lies in the need for algorithms to display robustness to class imbalance (Krawczyk, 2016; Fernández et al., 2018a). Despite almost three decades of research, handling skewed class distributions is still a crucial domain of machine learning. This becomes even more challenging in the streaming scenario, where imbalance happens simultaneously with concept drift. Not only do the definitions of classes change but also the imbalance ratio becomes dynamic and class roles may switch. Solutions that assume fixed data properties cannot be applied here, as streams may oscillate between varying degrees of imbalance and periods of balance among classes. Furthermore, imbalanced streams can have other underlying difficulties, such as small sample size, borderline and rare instances, overlapping among classes, or noisy labels (Santos et al., 2022). Imbalanced data streams are usually handled via class resampling (Korycki & Krawczyk, 2020; Bernardo et al., 2020b; Bernardo & Della Valle, 2021a), algorithm adaptation mechanism (Loezer et al., 2020; Lu et al., 2020), or ensembles (Zyblewski et al., 2021; Cano & Krawczyk, 2022). This problem is motivated by a plethora of real-world problems where data is both streaming and skewed, such as Twitter streams (Shah & Dunn, 2022), fraud detection (Bourdonnaye & Daniel, 2022), abuse and hate speech detection (Marwa et al., 2021), Internet of Things (Sudharsan et al., 2021), or intelligent manufacturing (Lee, 2018). While there are several works on how to handle imbalanced data streams, there are no agreed-upon standards, benchmarks, or good practices that are necessary for fully reproducible, transparent, and impactful research.

Research goal. To create a standardized, exhaustive, and informative experimental framework for binary and multi-class imbalanced data streams, and conduct an extensive comparison of state-of-the-art classifiers.

Motivation. While there are many algorithms for drifting and imbalanced data streams in the literature, there is a lack of standardized procedures and benchmarks on how to evaluate these algorithms holistically. Existing studies are often limited to a selection of algorithms and data difficulties, typically only considering binary class data, and do not provide insights into what aspects of imbalanced data streams must be considered and translated into meaningful benchmark problems. There is a need for a unified and holistic evaluation framework for imbalanced data streams that could be used as a template for researchers to evaluate their newly proposed algorithms against the relevant methods in the literature. Additionally, in-depth experimental comparison of state-of-the-art methods would allow to gain valuable insights into what classifiers and learning mechanisms work under different conditions. Therefore, we propose an evaluation framework and perform a large-scale

empirical study to obtain insights into the performance of the methods under an extensive and varied set of data difficulties.

Overview and contributions. This paper proposes a complete and holistic framework for benchmarking and evaluating classifiers for imbalanced data streams. We summarize existing works and organize them according to established taxonomies dedicated to skewed and streaming problems. We distill the most crucial and insightful problems that appear in this domain and use them to design a set of benchmark problems that capture distinctive learning difficulties and challenges. We compile these benchmarks into a framework embedding various metrics, statistical tests, and visualization tools. Finally, we showcase our framework by comparing 24 state-of-the-art algorithms, which allows us to choose the best-performing ones, discover in what specific areas they excel and formulate recommendations for end-users. The main contributions of the paper are summarized as follows:

- **Taxonomy of algorithms for imbalanced data streams.** We organize the methods in the state of the art according to established taxonomies that summarize recent progress in learning from imbalanced data streams and provide a survey of the most important contributions.
- **Holistic and reproducible evaluation framework.** We propose a complete and holistic framework for evaluating classifiers for two-class and multi-class imbalanced data streams that standardizes metrics, statistical tests, and visualization tools to be used for transparent and reproducible research.
- **Diverse benchmark problems.** We formulate a set of benchmark problems to be used within our framework. We capture the most vital and challenging problems that are present in imbalance data streams, such as dynamic imbalance ratio, instance-level difficulties (borderline, rare, and subconcepts), or number of classes. Furthermore, we include real-world and semi-synthetic imbalanced problems, leading to a total of 515 data stream benchmarks.
- **Comparison among state-of-the-art classifiers.** We conduct an extensive, comprehensive, and reproducible comparative study among 24 state-of-the-art stream mining algorithms based on the proposed framework and 515 benchmark problems.
- **Recommendations and open challenges.** Based on the results from the exhaustive experimental study, we formulate recommendations for end-users that will allow to understand the strengths and weaknesses of the best-performing classifiers. Furthermore, we formulate open challenges in learning from imbalanced data streams that should be addressed by researchers in the years to come.

Comparison with most related experimental works. In recent years, several survey papers and works with large experimental studies touching on joint areas of class imbalance and data streams were published. Therefore, it is important to understand the key differences between them and this work, as well as how our survey provides new insights into this topic that were not touched upon in the previous works. Wang et al. (2018) proposed an overview of several existing techniques, both drift detectors and adaptive classifiers, and experimentally compared their predictive accuracy. While being the first dedicated study in this area, it was limited by not evaluating computational complexities of compared algorithms, using a very small selection of datasets (7 benchmarks), and investigating only limited properties of imbalanced data streams (not touching upon instance-level characteristics or multi-class problems). Brzeziński et al. (2021) proposed a follow-up study that focused on data-level properties of imbalanced streams, such as instance difficulties (borderline and rare instances) and the presence of subconcepts. However, the study was done

Table 1 Comparison of the number of algorithms and benchmarks evaluated in most related works

	Wang et al. (2018)	Brzeziński et al. (2021)	Bernardo et al. (2021)	Cano and Krawczyk (2022)	This paper
Algorithms					
General purpose	×	1	2	21	5
Imbalanced specific	10	4	9	9	19
Benchmarks					
Binary class generators	4	385	232	99	406
Binary class datasets	3	4	3	16	19
Multi-class generators	×	×	×	×	72
Multi-class datasets	×	×	×	×	18
Total algorithms	10	5	11	30	24
Total benchmarks	7	389	235	115	515

for a limited number of algorithms (5 classifiers) and focused only on two-class problems. Bernardo et al. (2021) proposed an experimental comparison of methods for imbalanced data streams. They extended Brzeziński et al. (2021) benchmarks using different levels of imbalance ratio and three drift speeds. However, their study analyzed a limited number of algorithms (11 classifiers) and only three real-world datasets. Cano and Krawczyk (2022) presented a large comparison of 30 algorithms focusing on ensemble approaches but 21 of them were general-purpose ensembles rather than imbalanced specific classifiers. These four works address only binary class imbalanced data streams. This paper extends the benchmark evaluation from all previous studies, proposes new benchmark scenarios, extends the number of real-world datasets, and evaluates both two-class and multi-class imbalanced data streams. We also extend the comparison to 24 classifiers, 19 of them specifically designed for imbalanced data streams. Table 1 summarizes the main differences in the experimental evaluations of these works. This allows us to conclude that while these works are an important first step, there is a need for a unified, comprehensive, and holistic study of learning from imbalanced data streams that could be used as a template for researchers to evaluate their newly proposed algorithms.

This paper is organized as follows. Section 2 provides a background on data streams. Section 3 discusses the main challenges of imbalanced data. Section 4 presents the specific difficulties of imbalanced streams. Section 5 describes the approaches for tackling imbalanced steams with ensembles. Section 6 introduces the experimental setup and methodology. Section 7 presents and analyzes the results of our study. Section 8 summarizes the lessons learned. Section 9 formulates recommendations to end-users for selecting the best algorithms for imbalanced data streams. Section 10 discusses the open challenges and future directions. Finally, Sect. 11 covers the conclusions.

2 Data streams

In this section we present the preliminaries of data stream characteristics, learning approaches, and the concept drift properties.

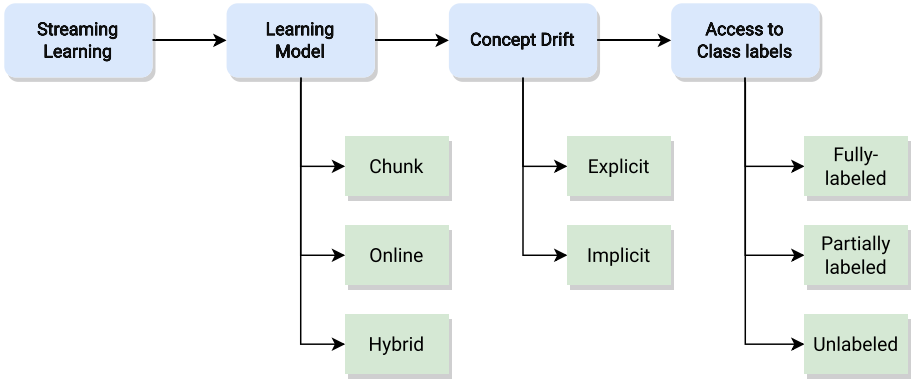

Fig. 1 Streaming learning taxonomy

2.1 Data stream characteristics

The main characteristics of data streams can be summarized as follows (Gama, 2010; Krempl et al., 2014; Bahri et al., 2021):

- **Volume.** Streams are potentially unbounded collections of data that constantly flood the system and thus they are impossible to be stored and must be processed incrementally. The volume also imposes limitations on the computational resources, which are magnitudes smaller than the actual size of data would call for.
- **Velocity.** Streaming data sources are in constant motion. New data is being generated continuously and often in rapid bursts, leading to high-speed data streams. These force learning systems to work in real-time, must be analyzed and incorporated into the learning system to model the current state of the stream.
- **Non-stationarity.** Data streams are subject to change over time, which is known as concept drift. This phenomenon may affect feature distributions, class boundaries, but also lead to changes in class proportions, or emergence of new classes (or disappearance of old ones).
- **Veracity.** Data arriving from the stream can be uncertain and affected by various problems, such as noise, injection of adversarial patterns, or missing values. Having access to fully labeled stream is often impossible due to cost and time requirements, leading to need for learning from weakly labeled instances.

We can define a stream S as a sequence $< s_1, s_2, s_3, \ldots, s_\infty >$. We consider a supervised scenario $s_i = (X, y)$, where $X = [x_1, x_2, \ldots, x_f]$ with f as the dimensionality of the feature space, and y as the target variable, which may or may not be available on arrival. Each instance in the stream is independent and randomly drawn from a stationary probability distribution. Figure 1 illustrates the workflow to learn from data streams and approaches to tackle related challenges (Gama, 2012; Nguyen et al., 2015; Ditzler et al., 2015; Wares et al., 2019).

2.2 Learning model

Due to both the volume and velocity of data streams, algorithms need to be capable of incremental processing of the continuously arriving information. Instances from the data stream are provided either online, or in the form of data chunks (portions, blocks).

- **Online.** Algorithms will process each single instance one by one. The main advantage of this approach is a low response time and adaptivity to changes in the stream. The main drawback lies in their limited view of the current state of the stream, as a single instance can be either a poor representation of a larger concept or may be susceptible to noise.
- **Chunk.** Instances are processed in windows called data chunks or blocks. Chunk-based approaches offer a better estimation of the current concept due to a larger training sample size. The main drawback is the delayed response to changes in some settings because the construction, evaluation, or updating of classifiers is done when all instances from a new block are available. Additionally, in case of rapid changes chunks may consist of instances coming from multiple concepts, further harming the adaptation capabilities.
- **Hybrid.** Hybrid approaches can combine the previous methodologies to address their shortcomings. One of the most popular approaches is to use online learning, while maintaining chunks of data to extract statistics and useful knowledge about the stream for additional periodical classifier updates.

2.3 Concept drift

Data streams are subject to a phenomenon called concept drift (Krawczyk et al., 2017; Lu et al., 2018). Each instance arrives at a time t and is generated by a probabilistic distribution $\Phi^t(X, y)$ where X corresponds to the feature vector and y to the class label. If the same probability distribution generates all instances in the stream, data is stationary, i.e., originating from the same concept. On the other hand, if two separate instances, arriving at times t and $t + C$, are generated by $\Phi^t(X, y)$ and $\Phi^{t+C}(X, y)$. If $\Phi^t \neq \Phi^{t+C}$, then a concept drift occurred. When analyzing and understanding concept drift, following factors are considered:

- **Influence of the decision boundaries.** Here we distinguish: (i) virtual; and (ii) real types of drift. Virtual drift can be defined as a change in the unconditional probability distribution $P(x)$, meaning it does not affect the learned decision boundaries. Such drift, while not having a deteriorating influence on learning models, must be detected as it may trigger false alarms and force unnecessary, yet costly adaptation. Real concept drift affects the decision boundaries, making them worthless to the current concept. Detecting it and adapting to new distribution is crucial for maintaining predictive performance.
- **Speed of change.** Here we can distinguish three types of concept drift (Webb et al., 2016): (i) incremental; (ii) gradual; and (iii) sudden. Incremental drift generates a sequence of intermediate states between the old and new concept that are often. This requires detection of the stabilization moment when new concept becomes fully formed and relevant. Gradual drift oscillates between instances coming from both old and new

concepts, with new concept becoming more and more frequent over time. Sudden drift instantaneously switches between old and new concept, leading to an instant degradation of the underlying learning algorithm.
- **Recurrence**. Changes in the stream can be either unique or recurring. In the latter case the previously seen concept may reemerge over time, allowing us to recycle previously learned knowledge. This calls for having a repository of models that can be utilized for faster adaptation to previously seen changes. With more relaxed assumptions, one can extend recurrence to appearance of concepts similar to the ones seen in the past. Here, the past knowledge can be used as initialization point for the drift recovery.

There are two strategies to tackle concept drift: explicit and implicit (Lu et al., 2018; Han et al., 2022):

- **Explicit**. Here drift adaptation is managed by an external tool, called drift detector (Barros & Santos, 2018). They are used for continuous monitoring of the stream properties (e.g. statistics) or classifier performance (e.g. error rates). Drift detectors raise a warning signal when there are signs of upcoming drift, and alarm signal when the concept drift has taken place. When drift is detected, the classifier is replaced with a new one trained on recent instances. The pitfall of drift detectors is the need for labeled instances (semi-supervised and unsupervised detectors also exist but are less accurate) and false alarms that replaces competent classifiers.
- **Implicit**. Here drift adaptation is managed by learning mechanisms embedded in the classifier, assuming that it can adjust itself to new instances from the latest concept and gradually forget outdated information (Ditzler et al., 2015; da Costa et al., 2018). This requires establishing proper learning and forgetting rates, use of adaptive sliding windows, or continual hyperparameter tuning.

2.4 Access to labels

Obtaining the ground truth (e.g. class labels) in a data stream setting relates to significant time and cost requirements. As instances arrive continuously and in large volumes, domain experts may not be able to label a significant portion of the data or may not be able to provide labels fast enough. In the case of applications where labels can be obtained at no cost (e.g. weather prediction), a significant delay between instance and label arrival must be considered. Data streams can be divided into three groups concerning ground truth availability:

- **Fully-labeled**. For every instance x in the stream the label y is known and can be used for training. This scenario assumes no need for explicit label query and is the most common one for evaluating stream learning algorithms. However, the assumption of a fully labeled stream may not be feasible for many real-world applications.
- **Partially labeled**. Only a subset of instances in the stream are labeled on arrival. The ratio between labeled and unlabeled instances can change overtime. This scenario requires either active learning for selecting most valuable instances for labelling (Žliobaitė et al., 2013) or semi-supervised mechanisms for extending the knowledge from labeled instances unto unlabeled ones (Bhowmick & Narvekar, 2022; Gomes et al., 2022).

- **Unlabeled**. Every instance arrives without label and one cannot obtain it upon request, or it will arrive with a significant delay. This forces approximation mechanisms that can either generate pseudo-labels, look for evolving structures in data, or use delayed labels to approximate future concepts.

In this work, only fully labeled streams were used, but some of the algorithms evaluated possess mechanisms to deal with partially labeled or unlabeled streams.

3 Imbalanced data

In this section we will discuss shortly the main challenges present when learning from imbalanced data. Almost three decades of developments in this field allowed us to gain deeper insights into what inhibits the performance of classifier training procedures under skewed distributions (Fernández et al., 2018a).

- **Imbalance ratio.** The most obvious and well-studied property of imbalanced datasets is their imbalance ratio, i.e., the disproportion between majority and minority classes. It is commonly assumed that the higher the imbalance ratio, the more difficulty it poses to a classifier. This is justified by the fact that most classifier training procedures are driven by 0-1 loss functions that assume uniform importance of every instance. Therefore, the more predominant the majority class is, the more classifier becomes biased towards it. However, many recent studies have pointed out that the imbalance ratio is not the sole source of learning difficulties (He & Ma, 2013). As long as classes are well-separated and sufficiently represented in the training set, even very high imbalance ratio will not significantly impair the classifier. Therefore, we must look into instance-level properties to find other sources of classifier bias.
- **Small sample size.** The imbalance ratio is often accompanied by the fact that minority class is appearing infrequently and collecting sufficient number of instances may be costly, time-consuming, or simply impossible. This leads to an issue of small sample size, where minority class does not have big enough training set to allow classifiers to correctly capture its characteristics (Wasikowski & Chen, 2010). This, combined with high imbalance ratio, can significantly affect the training procedure, leading to poor generalization capabilities and classification bias. Furthermore, small sample size cannot guarantee that the training set is representative of the actual distribution - problem known as data shift (Rabanser et al., 2019).
- **Class overlapping.** Another challenge in imbalanced learning comes from the topology of classes, as often minority and majority classes overlap significantly. Class overlap poses difficulty for standard machine learning problems (Galar et al., 2014), while presence of skewed distribution makes it even more challenging (Vuttipittayamongkol et al., 2021). Overlapping regions can be seen as uncertainty regions for classifiers. In such case, the majority class will dominate the training procedure, leading to decision boundary ignoring the minority class in the overlapping area. This problem becomes even more difficult when dealing with multiple classes overlapping with each other.
- **Instance-level difficulties.** The problem of class overlapping points out to the importance of analyzing the properties of minority class instances and their individual difficulties. Minority classes often form small disjuncts, creating subconcepts that further reduce the minority class sample size in given area (García et al., 2015). When looking

at individual properties of each instance, one can analyze its neighborhood in order to determine how challenging it will be for the classifier. A popular taxonomy divides minority instances into safe, borderline, rare, and outliers based on how homogeneous are the class labels of their nearest neighbors (Napierala & Stefanowski, 2016). This information can be utilized to either obtain more effective resampling approaches or guide the classifier training procedure.

4 Imbalanced data streams

Class imbalance is one of the most vital problems in contemporary machine learning (Fernández et al., 2018a; Wang et al., 2019). It deals with a disproportion among the number of instances in each class, where some of the classes are significantly underrepresented. As most classifiers are driven by 0-1 loss, they get biased towards the easier to model majority classes. The underrepresented minority classes are usually the more important ones, thus one needs to alter either the dataset or learning procedure to create balanced decision boundaries that do not favor any of the classes.

Class imbalance is a common problem in the data stream mining domain (Wu et al., 2014; Aminian et al., 2019). Here streams can have a fixed imbalance ratio, or it may evolve over time (Komorniczak et al., 2021). Furthermore, class imbalance combined with concept drift poses novel and unique challenges (Brzeziński & Stefanowski, 2017; Sun et al., 2021). Class roles may switch (majority becomes the minority and vice versa), several classes may change (new classes appearing or old disappearing), or instance level difficulties may emerge (evolving class overlapping or clusters/sub-concepts) (Krawczyk, 2016). Changes in the imbalance ratio can be independent or connected with concept drift, where class definitions ($P(y \mid x)$) will change over time (Wang & Minku, 2020). Henceforth, monitoring each class for changes in its properties is not enough, as one also needs to track per-class frequencies of arriving new instances.

In most real-life scenarios streams are not predefined as balanced or imbalanced and they may become imbalanced only temporarily (Wang et al., 2018). Users' interests over time (where new topics emerge and old ones lose relevance) (Wang et al., 2014), social media analysis (Liu et al., 2020), or medical data streams (Al-Shammari et al., 2019) are examples of such cases. Therefore, a robust data stream mining algorithm should display high predictive performance regardless of the underlying class distributions (Fernández et al., 2018a). Most algorithms dedicated to imbalanced data streams do not perform as well on balanced problems as their canonical counterparts (Cano & Krawczyk, 2020). On the other hand, these canonical algorithms display low robustness to high imbalance ratios. There exist but few algorithms that can handle both scenarios with satisfactory performance (Cano & Krawczyk, 2020, 2022).

There are two main approaches dedicated to handling imbalanced data:

- **Data-level approaches.** These methods focus on the alteration of the underlying dataset to make it balanced (e.g. by oversampling or undersampling), thus being classifier-agnostic approaches. They focus on resampling or learning more robust representations.
- **Algorithm-level approaches.** These methods focus on modifying the training approach to make classifiers robust to skewed distributions. They are dedicated to specific learning models, being often more specialized, but less flexible than their data-level counter-

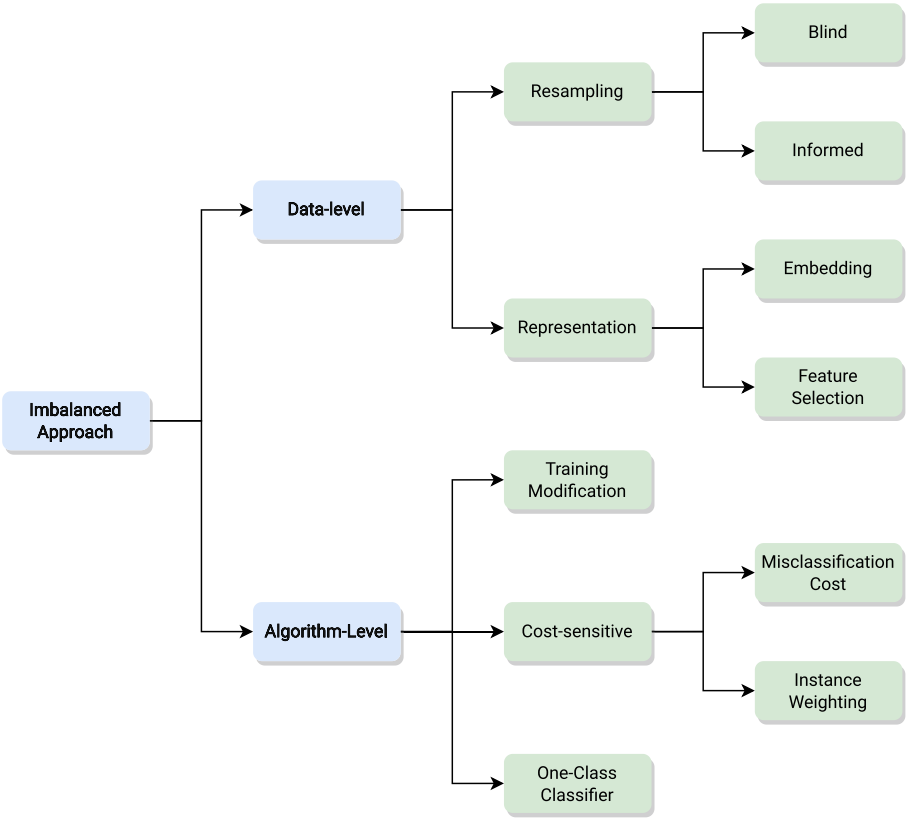

Fig. 2 Taxonomy of approaches for imbalanced data streams

parts. Algorithm-level modifications focus on identifying mechanisms that suffer from class imbalance, cost-sensitive learning, or one-class classification.

Figure 2 presents a taxonomy (He & Garcia, 2009; Branco et al., 2016; Krawczyk, 2016; Fernández et al., 2018a) of approaches for tackling the class imbalance problem. The specific details are discussed in the following subsections.

4.1 Data-level approaches

While resampling techniques are very popular for static imbalanced problems (Fernández et al., 2018a; Aminian et al., 2021), they cannot be directly used in the streaming scenario. Concept drift may render resampled data obsolete or even harming to the current state of the stream (e.g. when classes switch roles and resampling starts to empower further the new majority). This calls for dedicated strategies for keeping track of which classes should be resampled at a given moment, as well as for mechanisms capable of dealing with drift by forgetting outdated artificial instances (Fernández et al., 2018a).

Resampling algorithms can be categorized as either blind or informed (utilizing information about minority class properties to at least some degree). While blind approaches

can be effectively combined with ensembles due to their low computational cost, they do not perform well on their own. Therefore, most resampling methods for data streams are informed and based on a very popular SMOTE (Synthetic Minority Over-sampling Technique) algorithm (Fernández et al., 2018b). Those versions focus on keeping track of changes in the stream by employing either adaptive windows (Korycki & Krawczyk, 2020) or data sketches (Bernardo & Della Valle, 2021a; Bernardo & Della Valle, 2021b). This allows them to generate relevant artificial instances for the current concept and display good reactivity to sudden changes in the stream. It is important to note that the streaming version of SMOTE presented in (Korycki & Krawczyk, 2020) can work with any number of classes, as well as under extremely limited access to class labels. Incremental Oversampling for Data Streams (IOSDS) (Anupama & Jena, 2019) focuses on replicating instances that are not identified as noisy or overlapping. Clustering of data chunks can be used to identify the most relevant instances to resample (Czarnowski, 2021). Undersampling via Selection-Based Resampling (SRE) (Ren et al., 2019) iteratively removes the safe instances from the majority class without introducing reverse bias towards the minority class. Some works present the usefulness of combining over and under sampling together to obtain a more diverse representation of the minority class (Bobowska et al., 2019). When handling multi-class imbalanced data streams, resampling can be either conducted using information about all of classes (Korycki & Krawczyk, 2020; Sadeghi & Viktor, 2021) or by applying binarization schemes and pairwise resampling (Mohammed et al., 2020a). Active learning techniques such as dynamic budgets (Aguiar & Cano, 2023) and Racing Algorithms (Nguyen et al., 2018) are also combined with resampling techniques to overcome class imbalance (Mohammed et al., 2020b). Disadvantages of data-level methods lie in their high memory use (when oversampling), or the possibility of under-representation of older concepts that are still relevant (when undersampling).

A study by Korycki and Krawczyk (2021b) discusses an alternative data-level approach to resampling. They propose to create dynamic and low-dimensional embeddings that use information about the class imbalance ratio and separability to find highly discriminative projections. A well-defined low-dimensional embedding may offer better class separability and thus make resampling obsolete, especially when dealing with high-dimensional and difficult imbalanced data streams.

4.2 Algorithm-level approaches

Among training modifications, the most popular one is the combination of Hoeffding Decision Trees with Hellinger splitting criteria to make skew-insensitive (Lyon et al., 2014). Ksieniewicz (2021) proposed a method to modify predictions of a base classifier on-the-fly, aiming at modifying prior probabilities based on the frequency of each class. A new loss function was proposed to make neural networks able to handle imbalanced streams in an online setting (Ghazikhani et al., 2014). A combination of online active learning, siamese networks, and multi-queue memory was introduced by (Malialis et al., 2022). Various modifications of the popular Nearest Neighbors classifier have been adapted to tackling imbalanced data streams by using either dedicated memory formation or skew-insensitive distance metrics (Vaquet & Hammer, 2020; Roseberry et al., 2019; Abolfazli & Ntoutsi, 2020). Genetic programming has been successfully used for induction of robust classifiers from the stream (Jedrzejowicz & Jedrzejowicz, 2020), as well as increasing skew-insensitive rule interpretability and recovery speed from concept drift (Cano & Krawczyk, 2019).

Cost-sensitive methods have been applied to streaming decision trees. Krawczyk and Skryjomski (2017) proposed replacing leaves with perceptrons that use cost-sensitive threshold adjustment of class-based outputs. Their cost matrix is adapted in an online fashion to the evolving imbalance ratio, while multiple expositions of difficult instances are used to improve adaptation. Alternatively, Gaussian cost-sensitive decision trees combine cost and accuracy into a hybrid criterion during their training (Guo et al., 2013). Another approach uses Online Multiple Cost-Sensitive Learning (OMCSL) (Yan et al., 2017) where cost matrices for all classes are adjusted incrementally according to a sliding window. The recent framework proposed two-stage cost-sensitive learning, where a cost matrix is used for both online feature selection and classification (Sun et al., 2020). Finally, cost-sensitive approaches have been combined with Extreme Learning Machine algorithms via weighting matrices and misclassification costs (Li-wen et al., 1994).

One-class classification is an interesting solution to class imbalance, where one uses these class-specific models to either describe minority class or all the classes (achieving a one-class decomposition of multi-class problems) (Krawczyk et al., 2018). One-class classifiers can be used for data stream mining scenarios and display good reactivity to concept drift (Krawczyk & Wozniak, 2015). One can use adaptive online one-class Support Vector Machines to track minority classes and their changes over time (Klikowski & Woźniak, 2020). One can combine one-class classification with ensembles, over-sampling, and instance selection (Czarnowski, 2022). One-class classifiers can be combined with active learning to select the most informative instances from the stream to be used for class modeling (Gao, 2015). Anomaly detection, similar in its assumptions to one-class classifiers can also be used to identify minority and majority instances in the stream (Liang et al., 2021).

4.3 Similar domains

When talking about learning from imbalanced data streams, it is necessary to mention to similar domains in contemporary machine learning, namely continual learning and long-tailed recognition.

Similarities to continual learning. It is important to mention that data stream mining can often be viewed as task-free continual learning (Krawczyk, 2021). While imbalanced problems have not been yet discussed widely in this setup, there are some works noticing the importance of handling skewed class distributions for continual deep learning (Chrysakis & Moens, 2020; Kim et al., 2020; Arya & Hanumat Sastry, 2022; Priya & Uthra, 2021).

Similarities to long-tailed recognition. The extreme case of multi-class imbalance is known as long-tailed recognition (Yang et al., 2022). It deals with situations, where we have hundreds or thousands of classes, with progressively increasing imbalance ratio and smallest classes being extremely imbalanced compared to the majority ones (hence long-tailed class-based distribution of instances). This problem is mainly discussed in the context of deep learning, where various decomposition strategies (Zhu et al., 2022), loss functions (Zhao et al., 2022), or cost-sensitive solutions (Peng et al., 2022) are being utilized. Currently, there are but few works that discuss the combined challenge of continual learning from long-tailed distributions (Kim et al., 2020).

5 Ensembles for imbalanced data streams

Combining multiple classifiers into an ensemble is one of the most powerful approaches in modern machine learning, leading to improved predictive performance, generalization capabilities, and robustness. Ensembles have proven themselves to be highly effective for data streams, as they offer unique ways of managing concept drift and class imbalance (Krawczyk et al., 2017). The former can be achieved by adding new classifiers or updating the existing ones, while the latter is achieved by combining classifiers with different skew-insensitive approaches (Brzeziński & Stefanowski, 2018; Grzyb et al., 2021; Du et al., 2021).

Ensembles for data streams can be categorized by the following design choices:

- **Classifier pool generation.** There are two major approaches for generating a pool of classifiers for forming an ensemble: heterogeneous and homogeneous (Bian & Wang, 2007). Heterogeneous solutions assume that we ensure diversity of the pool by using different classifier models, aiming at exploiting their individual strengths at forming decision boundaries. Homogeneous solutions assume that we select a specific type of classifier (e.g., popular choice are decision trees) and then ensure diversity among them by modification of the training set. This is usually achieved by one of two popular solutions: bagging and boosting. Bagging (bootstrap aggregating) trains multiple independent base learners in parallel and combines their predictions using an aggregation function (e.g. by simple average or simple majority vote). Boosting trains the base learners in a sequential way. Each model in the sequence is fitted giving more importance to observations in the dataset that were poorly handled by the previous models. Predictions are combined using a deterministic strategy (e.g. weighted majority voting). It is worthwhile noting that while the majority of the methods are based on either heterogeneous pool or homogeneous weak learners, there exist alternative approaches, such as generating hybrid pools (using multiple types of models, but also generating multiple learners for each of them) (Luong et al., 2020) and using projections (Korycki & Krawczyk, 2021b).
- **Feature space modification.** This defines what feature space input is being used by base classifiers. They can either be trained on full feature space (here their diversity must be ensured in another way), feature subspaces, or completely new feature embeddings (e.g. creating artificial feature spaces).
- **Ensemble line-up.** This defines how ensembles are managed during the continual learning from streams. Voting procedures can be used for dynamical adjustment of base learners' importance. Ensembles can be fixed, meaning that each base learner is continuously updated, but never removed. Alternatively, one can use a dynamic setup, where worst classifiers are pruned and replaced by new ones trained on more recent instances. Finally, all of these mentioned techniques can be combined to create hybrid architectures, capable of better responsiveness to concept drift.

For imbalanced data streams, ensembles are usually combined with techniques mentioned in the previous section. Figure 3 presents a taxonomy (Krawczyk et al., 2017; Gomes et al., 2017a) based on how ensembles are built for data streams and how this can be connected with the previously discussed approaches to handle drifting and imbalanced streams.

The most popular approach lies in combining resampling techniques with Online Bagging (Wang et al., 2015, 2016; Wang & Pineau, 2016). Similar strategies can be applied to

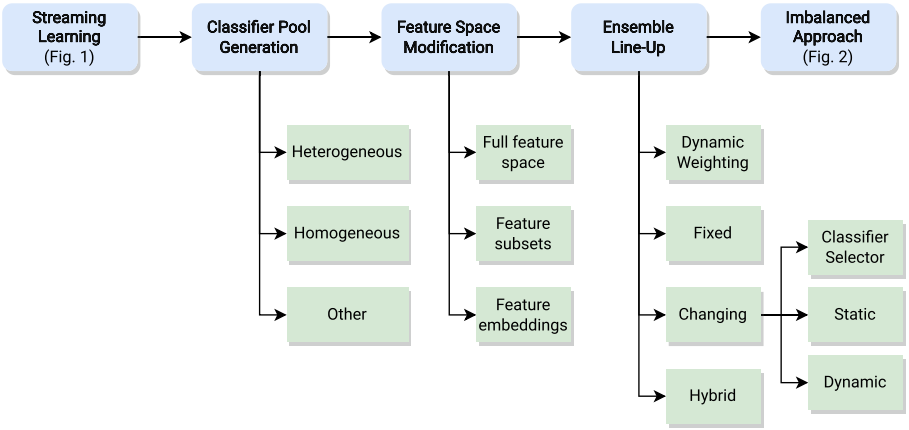

Fig. 3 Taxonomy of ensemble definition for imbalanced data streams

Adaptive Random Forest (Gomes et al., 2017b), Online Boosting (Klikowski & Woźniak, 2019; Gomes et al., 2019), Dynamic Weighted Majority (Lu et al., 2017), Dynamic Feature Selection (Wu et al., 2014), Adaptive Random Forest with resampling (Ferreira et al., 2019), Kappa Updated Ensemble (Cano & Krawczyk, 2020), Robust Online Self-Adjusting Ensemble (Cano & Krawczyk, 2022), Deterministic Sampling Classifier with weighted Bagging (Klikowski & Wozniak, 2022), Dynamic Ensemble Selection (Jiao et al., 2022; Han et al., 2023) or any ensemble that can incrementally update its base learners (Ancy & Paulraj, 2020; Li et al., 2020). It is interesting to note that preprocessing approaches enhance diversity among base classifiers (Zyblewski et al., 2019). Alternatively, cost-sensitive solutions can be used together with ensembles such as Adaptive Random Forest (Loezer et al., 2020).

The effectiveness of ensembles for imbalanced data streams can be further increased by using dedicated combination schemes or adaptive chunk-based learning (Lu et al., 2020). Weights assigned for each base classifier can be continuously updated to reflect their current competencies on minority classes (Ren et al., 2018). A reinforcement learning mechanism can be used to increase the weights of the base classifiers that perform better on the minority class (Zhang et al., 2019). One can use a hybrid approach that combines resampling minority instances with dynamic weighting base classifiers based on their predictive performance on sliding windows of minority samples (Yan et al., 2022). Dynamic selection of classifiers and their related preprocessing techniques can be a very effective tool for handling concept drift, as it offers exploitation of diversity among base classifiers (Zyblewski et al., 2021; Zyblewski & Woźniak, 2021). Alternatively, classifier selection balances subsets of the incoming stream. Cost-sensitive neural networks can be initialized using different random weights and then incrementally improved with new instances (Ghazikhani et al., 2013). OSELM (Li-wen et al., 1994) classifiers can be combined using diverse initialization to generate a more robust compound classifier (Wang et al., 2021).

Finally, ensembles found their applications in imbalanced data streams with limited access to class labels. CALMID is a robust framework to deal with limited label access, concept drift, and class imbalance by dynamically inducing new base classifiers with the weighting of the most relevant instances (Liu et al., 2021). Another approach uses reinforcement learning (Zhang et al., 2022) to select instances for updating the ensemble

under labeling constraints. In multi-class imbalance settings, self-training semi-supervised (Vafaie et al., 2020) methods were applied to self-labeling driven by a small subset of labeled instances. It can be realized by an abstaining mechanism temporarily removing uncertain classifiers, with dynamically adjusting the abstaining criterion in favor of minority classes (Korycki et al., 2019).

While the vast majority of mentioned ensembles use Hoeffding Decision Trees (or their variants) as base classifiers, there are several skew-insensitive ensembles dedicated to neural networks. ESOS-ELM (Mirza et al., 2015) maintains randomized neural networks that are trained on balanced subsets of the incoming stream. Cost-sensitive neural networks can be initialized using random weights and then incrementally improved with new instances (Ghazikhani et al., 2013). OSELM (Li-wen et al., 1994) classifiers can be combined using diverse initialization to generate a more robust compound classifier (Wang et al., 2021).

Finally, ensembles found their applications in imbalanced data streams with limited access to class labels. CALMID is a robust framework to deal with limited label access, concept drift, and class imbalance by dynamically inducing new base classifiers with the weighting of the most relevant instances (Liu et al., 2021). Another approach uses reinforcement learning (Zhang et al., 2022) to select instances for updating the ensemble under labeling constraints. In multi-class imbalance settings, self-training semi-supervised (Vafaie et al., 2020) methods were applied to self-labeling driven by a small subset of labeled instances.

6 Experimental setup

The experimental study was designed to evaluate the performance of data stream mining algorithms under varied imbalanced scenarios and difficulties. We aim at gaining a better understanding of the data difficulties and how they impact the classifiers. We address the following research questions (RQ):

- **RQ1**: How do different levels of class imbalance ratio affect the algorithms?
- **RQ2**: How do static versus dynamic imbalance ratios influence the classifiers?
- **RQ3**: How do instance-level difficulties impact the classifiers?
- **RQ4**: How do algorithms adapt to simultaneous concept drift and imbalance ratio changes?
- **RQ5**: Are there differences on the performance between imbalanced generators and real-world streams?
- **RQ6**: Is there trade-off between the accuracy and the computational and memory complexities?
- **RQ7**: What are the lessons learned? Which algorithm should I use in my dataset?

To answer these questions, we formulate a set of benchmark problems building on experiments proposed in previous studies and new ones to assess additional data difficulties in two-class and multi-class imbalanced data streams. One of the major issues in this research area is the lack of standardized and agreed-upon procedures and benchmarks on how to evaluate these algorithms holistically. Therefore, we evaluate a comprehensive set of benchmark problems which includes an exhaustive list of data difficulties in imbalanced data streams. The experimental study in Sect. 7 is divided into the following experiments whereas Sect. 8 discusses the lessons learned and recommendations.

Table 2 Data stream algorithms and their taxonomy

Algorithm	Imbalanced approach	Learning method	Concept drift	Multi class	BL agnostic
IRL (Bernardo et al., 2020a)	RI	Online	Explicit	×	✓
C-SMOTE (Bernardo et al., 2020b)	RI	Online	Explicit	×	✓
VFC-SMOTE (Bernardo & Della Valle, 2021b)	RI	Online	Explicit	×	✓
CSARF (Loezer et al., 2020)	CS	Online	Explicit	✓	×
GHVFDT (Lyon et al., 2014)	TM	Online	Implicit	✓	×
HDVFDT (Cieslak & Chawla, 2008)	TM	Online	Implicit	✓	×
ARF (Gomes et al., 2017b)	×	Online	Explicit	✓	×
KUE (Cano & Krawczyk, 2020)	H	Hybrid	Explicit	✓	✓
LB (Bifet et al., 2010a)	×	Online	Explicit	✓	✓
OBA (Bifet et al., 2009)	×	Online	Explicit	✓	✓
SRP (Gomes et al., 2019)	×	Online	Explicit	✓	✓
ESOS-ELM (Mirza et al., 2015)	RB	Chunk	Explicit	×	×
CALMID (Liu et al., 2021)	H	Hybrid	Explicit	✓	✓
MICFOAL (Liu et al., 2021)	H	Online	Explicit	✓	✓
ROSE	H	Hybrid	Explicit	✓	✓
OADA (Wang & Pineau, 2016)	×	Online	Explicit	✓	✓
OADAC2 (Wang & Pineau, 2016)	CS	Online	Explicit	×	✓
ARFR (Ferreira et al., 2019)	RI	Online	Explicit	✓	×
SMOTE-OB (Bernardo & Della Valle, 2021a)	RI	Online	Explicit	×	✓
OSMOTE (Wang & Pineau, 2016)	RI	Online	Explicit	×	✓
OOB (Wang et al., 2016)	ROB	Online	Implicit	✓	✓
UOB (Wang et al., 2016)	RUB	Online	Implicit	✓	✓
ORUS (Wang & Pineau, 2016)	RUB	Online	Explicit	×	✓
OUOB (Wang & Pineau, 2016)	RB	Online	Explicit	×	✓

CS: cost-sensitive, TM: training modification, RI: informed resampling R(O U)B: blind (over under) resampling, BL: base-learner, H: hybrid

Table 3 Ensemble algorithms and their taxonomy

Ensemble	Meta-algorithm	Feature space	Line-up
ARF	Bagging	Feature subsets	Fixed
KUE	Bagging	Feature subsets	Dynamic weighting
LB	Bagging	Full feature space	Fixed
OBA	Boosting	Full feature space	Fixed
SRP	Bagging	Feature subsets	Change dynamic
ESOS-ELM	Other	Full feature space	Hybrid
CALMID	Other	Full feature space	Change dynamic
MICFOAL	Other	Full feature space	Change dynamic
ROSE	Bagging	Feature subsets	Dynamic weighting
OADA	Boosting	Full feature space	Fixed
OADAC2	Boosting	Full feature space	Fixed
ARFR	Bagging	Feature subsets	Fixed
SMOTE-OB	Bagging	Feature subsets	Fixed
OSMOTE	Bagging	Feature subsets	Fixed
OOB	Bagging	Full feature space	Fixed
UOB	Bagging	Full feature space	Fixed
ORUS	Boosting	Feature subsets	Fixed
OUOB	Bagging	Feature subsets	Fixed

6.1 Algorithms

The experiments comprise 24 state-of-the-art algorithms for data streams, including best-performing general-purpose ensembles and algorithms specifically designed for imbalanced streams. Algorithms are presented in Table 2 with their characteristics according to the established taxonomies. Specific properties of the ensemble models are presented in Table 3. All algorithms are implemented in MOA (Bifet et al., 2010b). The source code of the algorithms and the experiments are publicly available on GitHub to facilitate the transparency and reproducibility of this research.[1] All results, interactive plots and tables are available on the website.[2] Algorithms were run on a cluster with 2300 AMD EPYC2 cores, 12 TB RAM, and Centos 7. No individual hyperparameter optimization was conducted for any algorithm. All algorithms use the parameter settings recommended by their authors on their respective implementations. All ensembles are evaluated with the same parameter settings of 10 base classifiers using Hoeffding tree as the base learner. We acknowledge that algorithms often depend on parameters that may have a significant impact on the results obtained. Some methods use random generators which require an initial random seed. Different seeds will produce different results and multiple seeds should be run when the number of benchmarks is small due to the central limited theorem. Other methods have parameters that affect the classifier learning (e.g. the split confidence of the Hoeffding tree) that should be more carefully chosen when fitting a particular dataset. Due to the large number

[1] Source code, experiments, and results are available at https://github.com/canoalberto/imbalanced-streams.
[2] Interactive plots and tables for all experiments are available at https://people.vcu.edu/~acano/imbalanced-streams.

Table 4 Specifications of the data stream generators

Generator	Attributes	Classes	Concept drift
Agrawal	10	2	✓ Functions
Asset	5	2	✓ Functions
Brzeziński	N (5)	2	✓ Centroids
Hyperplane	12	N (2 binary class, 5 multi-class)	✓ Features
Mixed	4	2	✓ Function
RandomRBF	N (10)	N (2 binary class, 5 multi-class)	✓ Centroids
RandomTree	N (10)	N (2 binary class, 5 multi-class)	✓ Trees
SEA	3	2	✓ Function
Sine	3	2	✓ Function
Text	100	2	×

of benchmarks, experiments, and data size, the results reported on the paper are the median for 5 runs (5 seeds). Complete results to facilitate future comparisons and detailed information about the specific parameter configuration are available on the GitHub repository.

6.2 Generators

To evaluate the classifiers in specific and controlled scenarios, we prepared data streams generators under different imbalanced and drifting settings. Nine generators in MOA (Bifet et al., 2010b) plus one generator proposed by Brzeziński et al. (2021) were used. Those generators are presented in Table 4, with their number of attributes, classes, and whether they can generate internal concept drifts. All generators are evaluated on a stream of 200,000 instances. For generators where it is possible to use a configurable number of attributes, the default value on the table was used. The number of classes was adjusted according to the experiment (2 for binary class experiments and 5 for multi-class experiments).

6.3 Performance evaluation

The algorithms were evaluated using the test-then-train model, where each instance is first used to test then update the classifier in an online manner (instance by instance). We measured seven performance metrics (Accuracy, Kappa, G-Mean, AUC, PMAUC, WMAUC, and EWMAUC). Complete results are available on the website https://people.vcu.edu/~acano/imbalanced-streams. However, due to the limitations of space in the manuscript, we show results for Kappa, G-Mean, and the Area Under the Curve (AUC). They are calculated over a sliding window of 500 instances. We also acknowledge that there are different schools of thought regarding the best selection of performance metrics for imbalanced data. Our argument is that in order to have a comprehensive evaluation of the classifier performance on imbalanced datasets, one should not use only one metric, whichever the metric is, since all metrics have biases one way or another, and focus on assessing different aspects. Therefore, in our study, we report pairs of metrics that we have observed they exhibit complementary behaviors.

Kappa is often used to evaluate classifiers in imbalanced settings (Japkowicz, 2013; Brzeziński et al., 2018, 2019). It evaluates the classifier performance by computing the inter-rater agreement between the successful predictions and the statistical distribution of

the data classes, correcting agreements that occur by mere statistical chance. Kappa values range from −100 (total disagreement) through 0 (default probabilistic classification) to 100 (total agreement) as Eq. 1.

$$Kappa = \frac{n \sum_{i=1}^{c} x_{ii} - \sum_{i=1}^{c} x_{i.} x_{.i}}{n^2 - \sum_{i=1}^{c} x_{i.} x_{.i}} \cdot 100 \tag{1}$$

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. However, some authors recommend to avoid Kappa as Kappa's values vary depending not only on the performance of the model in question, but also on the level of class imbalance in the data, which can make the analyses difficult (Luque et al., 2019).

To tackle a balance between the performance of classifiers on the majority and minority classes, many researchers consider null-bias metrics such as sensitivity and specificity (Brzeziński & Stefanowski, 2018). These metrics are based on the confusion matrix: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Sensitivity, also called recall, is the ratio of correctly classified instances from the minority class (true positive rate) defined in Eq. 2. Specificity is the ratio of instances correctly classified from the majority class (true negative rate) defined in Eq. 3. The geometric mean (G-Mean) is the product of the two metrics as defined in Eq. 4. This measure tries to maximize the accuracy of each of the classes while keeping these accuracies balanced. G-Mean is a recommended null-bias metric for class imbalance (Luque et al., 2019). For multi-class data, the geometric mean is the square root of the product of class-wise sensitivity. However, this introduces the problem that as soon as the recall for one class is 0 the product of the whole geometric mean becomes 0. Therefore, it is much more complicated to use in multi-class experiments with a large number of classes and consequently, AUC would be preferred.

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$G{-}Mean = \sqrt{Sensitivity \times Specificity} \tag{4}$$

The Area Under the Curve (AUC) is invariant to changes in class distribution and provides a statistical interpretation for scoring classifiers. However, to measure the ranking ability of the classifiers the AUC needs to sort the data and iterate through each example. We employ the prequential AUC formulation proposed by Brzeziński and Stefanowski (2017) which uses a sorted tree structure with a sliding window. The AUC formulation was extended by (Wang & Minku, 2020) for multi-class problems defining the Prequential Multi-Class (PMAUC) as Eq. 5.

$$PMAUC = \frac{1}{C(C-1)} \cdot \sum_{i \neq j} A(i \mid j) \tag{5}$$

where $A(i \mid j)$ is pairwise AUC when treating class i as the positive class and class j as negative, and C is the number of classes. Extensions of the PMAUC calculation include Weighted Multi-class AUC (WMAUC) and Equal Weighted Multi-class AUC (EWMAUC) (Wang & Minku, 2020).

Both AUC and G-Mean are blind regarding the level of the class imbalance, while Kappa takes into account the class distribution but makes it more difficult to understand. Therefore, in cases of extreme imbalance ratios, the Kappa metric can be very dissimilar to the G-Mean and AUC, which means a classifier can have a high value of AUC, but a very low Kappa value. This is very useful to understand the behavior of a classifier under high imbalance ratios and how different metrics exhibit complementary facets of the classification performance. Therefore, it is important to evaluate the algorithms using both metrics in other to counterbalance overestimation. Henceforth, in our experiments presented in the manuscript, we evaluated the classifiers with G-Mean and Kappa for binary class scenarios and PMAUC and Kappa for multi-class scenarios. Metrics were calculated prequentially (Gama et al., 2013) using a sliding window of 500 examples. Complete results for all metrics (Accuracy, Kappa, G-Mean, AUC, PMAUC, WMAUC, and EWMAUC) are available on the website https://people.vcu.edu/~acano/imbalanced-streams for analysis and comparison with future works.

7 Results

This section presents the experimental results from the set of benchmarks proposed to answer the research questions. Section 7.1 shows the experiments on binary class imbalanced streams. Section 7.2 shows the experiments on multi-class imbalanced streams. Finally, Sect. 7.3 shows overall results and an aggregated comparison of all algorithms.

Due to the very large number of experiments conducted in this work, we present in the manuscript a selection of the most representative results. The experiments are organized to show three levels of detail in the results. First, a more detailed comparison of the top five methods. Second, an aggregated comparison of the top ten methods. Third, a summary of the comparison among all methods. Complete results for all experiments on all algorithms, datasets/generators, and metrics are available on the website.[3]

7.1 Binary class experiments

The first set of experiments focuses on binary class problems with a positive minority class and a negative majority class. These experiments include static imbalance ratio, dynamic imbalance ratio, instance-level difficulties, concept drift and static imbalance ratio, concept drift and dynamic imbalance ratio, and real-world binary class imbalanced datasets.

[3] Complete results for all experiments are available at https://people.vcu.edu/~acano/imbalanced-streams.

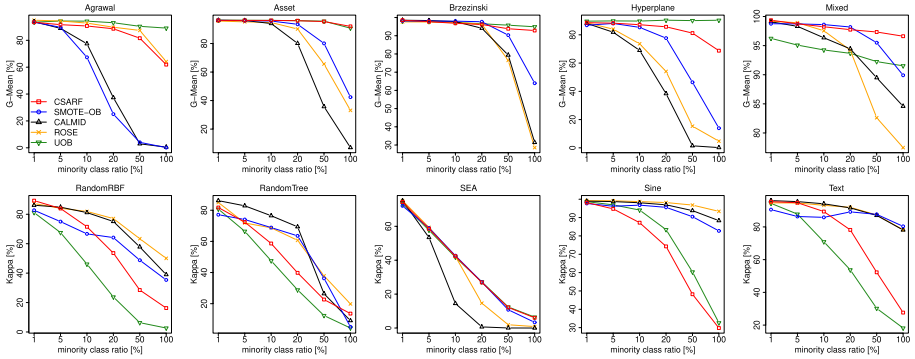

Fig. 4 Robustness to different levels of static class imbalance ratio (G-Mean and Kappa)

Table 5 G-Mean and Kappa averages of all 10 streams on static class imbalance ratio

IR	CSARF	ARF	KUE	LB	CALMID	ROSE	ARFR	SMOTE-OB	OOB	UOB
G-Mean										
1	94.54	94.54	94.10	**95.09**	94.68	94.65	94.57	93.30	94.15	93.77
5	**94.12**	86.79	90.28	92.10	91.87	92.76	93.68	93.44	93.92	93.07
10	**93.48**	72.95	79.20	80.38	83.09	89.87	88.59	90.55	92.98	92.16
20	**92.21**	57.21	69.44	70.35	69.05	78.75	79.08	83.64	90.39	91.12
50	88.95	39.55	49.49	47.50	49.14	63.83	47.55	68.44	73.97	**89.22**
100	82.75	30.14	36.55	36.41	35.17	46.82	14.50	45.10	60.99	**86.78**
Kappa										
1	89.13	89.12	88.76	**90.24**	89.42	89.36	89.20	86.75	88.36	87.63
5	82.12	79.38	82.83	**84.94**	84.18	83.86	84.33	80.84	83.77	78.68
10	70.92	65.64	72.43	74.63	74.88	**78.00**	72.95	72.72	77.54	65.44
20	55.19	52.42	62.79	64.10	61.53	65.95	57.38	62.52	**69.64**	48.24
50	31.91	36.15	45.92	44.34	44.34	54.15	27.43	47.90	**57.14**	26.77
100	17.84	28.39	34.55	34.48	32.84	41.09	5.62	35.18	**46.93**	13.92
Avg. G-Mean	91.01	63.53	69.85	70.30	70.50	77.78	69.66	79.08	84.40	**91.02**
Avg. Kappa	57.85	58.52	64.55	65.45	64.53	68.73	56.15	64.32	**70.56**	53.45
Rank G-Mean	**3.08**	8.08	7.87	5.69	6.32	5.15	5.93	5.25	3.51	4.12
Rank Kappa	6.40	6.49	6.00	4.11	4.77	4.33	6.32	6.00	**3.75**	6.83

Bold font highlights the best result

7.1.1 Static imbalance ratio

Goal of the experiment. This experiment was designed to address **RQ1** and evaluate the robustness of the classifiers under different levels of static class imbalance without concept drift. It is expected that classifiers that were designed to tackle class imbalance will present better robustness to different levels of imbalance, i.e., to achieve a stable performance regardless of the imbalance ratio. To evaluate this, we prepared the generators presented in Table 4 with static imbalance ratios (ratio of the size of the majority class to the minority

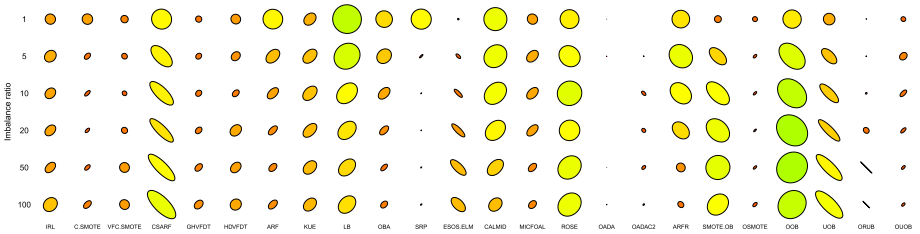

Fig. 5 Comparison of all 24 algorithms for different levels of static class imbalance ratio. The axes of the ellipse represent G-Mean and Kappa metrics. The bigger the axes the better rank of the algorithm on the metrics. The more rounded the ellipse the more agreement between the metrics. The color gradient represents the product of both metrics' ranks (Color figure online)

class as defined by Zhu et al. (2020)) of {5, 10, 20, 50, 100}. This allows us to assess how each classifier performs under specific levels of class imbalance. Figure 4 illustrates the performance of five selected algorithms with increasing levels of static imbalance ratio. Table 5 presents the average G-Mean and Kappa for the top 10 classifiers for each of the evaluated imbalance ratios and the overall rank of the algorithms. Figure 5 provides a comparison of all algorithms for each level of imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. The bigger the axes the better rank of the algorithm on the metrics. The more rounded the ellipse the more agreement between the metrics. Finally, the color represents a gradient of the product of the two metrics' ranks—red (worse) to green (better).

Discussion

Impact of approach to class imbalance. First, we will analyze the impact of different skew-insensitive mechanisms used by analyzed ensembles on their robustness to various levels of static imbalance under stationary assumptions. Looking at resampling-based methods we can observe a clear distinction between methods based on blind and informative approaches. Ensembles utilizing blind approaches usually drop their performance with an increasing imbalance ratio. Taking UOB as an example, one can see discrepancies between G-mean and Kappa metrics. For G-mean UOB maintains its predictive performance, to the point that for very high imbalance ratios it outperforms other approaches. However, for the Kappa metric, we can see that performance of UOB deteriorates significantly with each increase in class disproportions. This shows that UOB produces a good true positive ratio but proportionally a larger number of false positives. We can explain that by the limitations of undersampling to extreme class imbalance, as to balance the current distribution one must aggressively discard majority instances. In static problems the higher the disproportion between classes, the higher chance of discarding relevant majority examples. However, in a streaming setting, we analyze the imbalance ratio in an online manner, thus UOB is not able to counter the bias towards the majority class accumulated over time by undersampling incoming instances one by one. Its counterpart OOB shows the opposite behavior, returning best results for Kappa metric. Additionally, for high imbalance ratios OOB starts displaying balanced performance on both metrics. This shows that blind oversampling in online scenarios are capable of better and faster countering of bias accumulated over time. From informative resampling methods, we can observe that only SMOTE-OB returns satisfactory performance. For the Kappa metric, it can outperform UOB but does not fare well against OOB. All other algorithms that use SMOTE-based resampling perform even worse. This allows us to conclude that blind oversampling performs best from all data-level mechanisms in terms of robustness to static imbalance.

Among the algorithm-level solutions, CSARF displays best results for the G-mean metric, outperforming all reference methods. However, it does not hold its performance when evaluated using Kappa. This is another striking example of discrepancies between those metrics and how they highlight different aspects of imbalanced classification. Alternative algorithm-level approaches, such as ROSE and CALMID, while performing worse on G-mean, offer a more balanced performance on both metrics at once. Additionally, they display good robustness to increasing imbalance ratios. Therefore, algorithm selection for data streams with static imbalance is far from trivial, as one must choose between methods that perform very well only on one of the metrics, or choose a well-rounded method that, while not exceeding on any single metric, offers more even performance.

Finally, out of standard ensembles with no skew-insensitive mechanisms, LB returned the best predictive performance, outperforming several methods dedicated to imbalanced data streams. This did not hold for other methods, such as SRP or ARF that displayed no robustness to increasing imbalance ratios.

Impact of ensemble architecture. When we look at the overall best-performing methods in every scenario, we can see a dominance of ensembles based on bagging or hybrid architectures. Bagging offers an easy and effective way of maintaining instance-based diversity among base learners that benefits both data and algorithm-level approaches and leads to high robustness under various levels of class imbalance. Within bagging methods, only OUOB can be seen as an outlier. We can explain this using our observations from the previous paragraph—that undersampling and oversampling offer contrary performance (one favoring G-mean and the other one Kappa). Therefore, by combining those two approaches we obtain an ensemble that is driven by two conflicting mechanisms. Boosting-based ensembles are usually the worst performing ones. We can explain this by the fact that boosting mechanism focuses on correcting the errors of the previous classifier in a chain. When dealing with high imbalance ratios the errors are driven by a small number of minority instances, leading to too small sample sizes to effectively improve the performance. As usually minority instances are misclassified, assigning high weights to them will lead to high error on the majority class by increasing the number of false positives. In the end, boosting-based ensembles will consist of classifiers biased towards one of the classes. Without proper selection or weighting mechanisms, it is impossible to maintain robustness to high imbalance ratios with such classifiers in the ensemble pool.

7.1.2 Dynamic imbalance ratio

Goal of the experiment. This experiment was designed to address **RQ2** and to evaluate how classifiers behave under dynamic imbalance ratios. Even though many existing methods were designed to deal with static imbalance ratio, they lack mechanisms that allow adaptation to time-varying changes in the imbalance ratio. To evaluate this, we prepared four scenarios: (i) increasing the imbalance ratio {1, 5, 10, 20, 50, 100}, (ii) increasing then decreasing the imbalance ratio {1, 5, 10, 20, 50, 100, 50, 20, 10, 5, 1}, (iii) flipping the imbalance ratio, in which the majority becomes the minority class and vice versa {100, 50, 20, 10, 5, 1, 0.2, 0.1, 0.05, 0.02, 0.01}, and (iv) flipping then repflipping the imbalance ratio, in which the majority becomes the minority class and then flips back to become the majority and vice versa {100, 50, 20, 10, 5, 1, 0.2, 0.1, 0.05, 0.02, 0.01, 0.02, 0.05, 0.1, 0.2, 1, 5, 10, 20, 50, 100}. In this experiment, we also evaluated two types of drift: gradual and sudden. This allows us to analyze how the classifiers can cope with dynamic imbalance ratio changes and how they can adapt when majority and minority change roles. Figures 6,

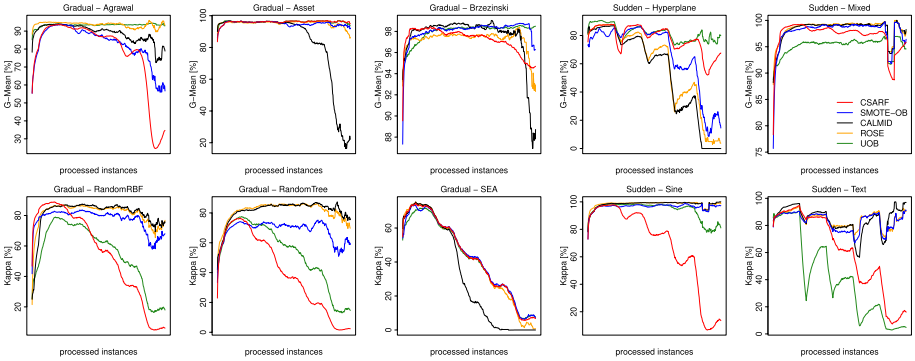

Fig. 6 G-Mean and Kappa on increasing class imbalance ratio with gradual and sudden drift

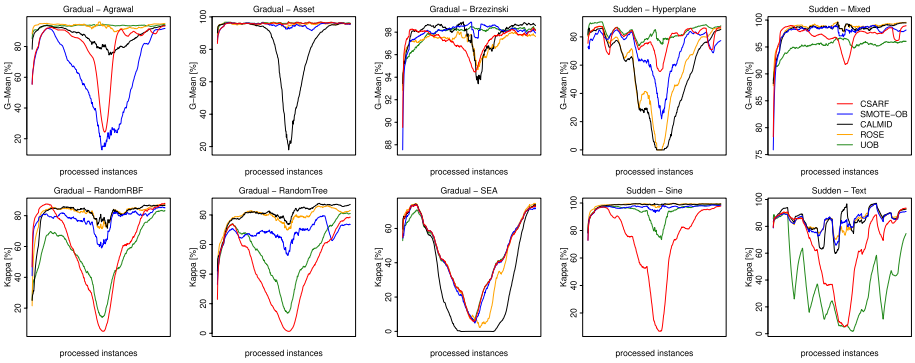

Fig. 7 G-Mean and Kappa on increasing decreasing class imbalance ratio with gradual and sudden drift

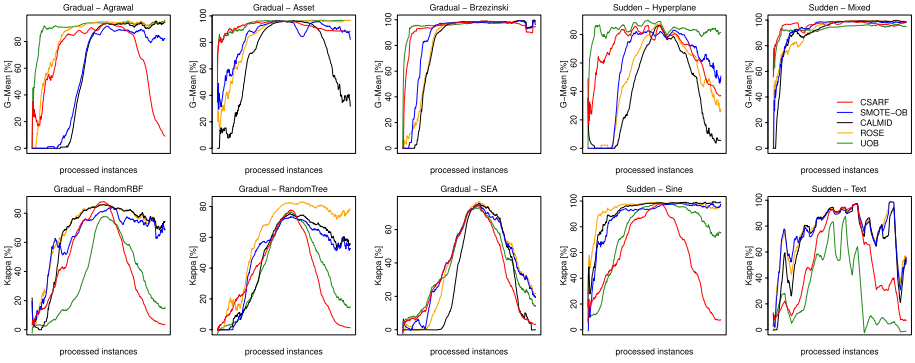

Fig. 8 G-Mean and Kappa on flipping class imbalance ratio with gradual and sudden drift

7, 8, 9 present the G-Mean and Kappa over time for the five selected classifiers for the generators and for both types of drift over (i) increasing imbalance ratio, (ii) increasing then decreasing imbalance ratio, (iii) flipping imbalance ratio, and (iv) flipping then reflipping

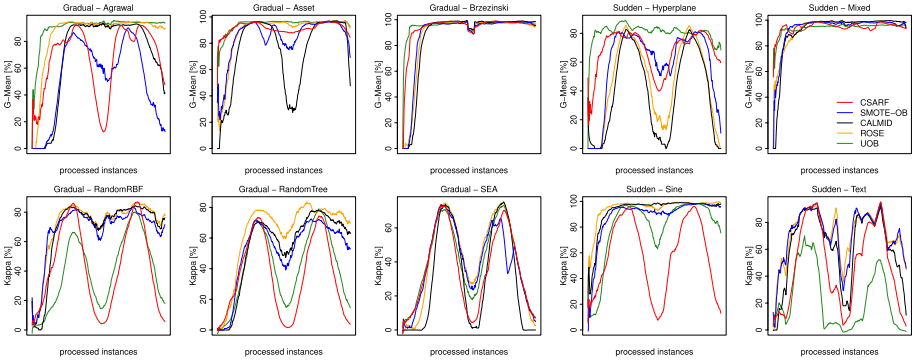

Fig. 9 G-Mean and Kappa on flipping and reflipping class imbalance ratio with gradual and sudden drift

Table 6 G-Mean and Kappa averages of all 10 streams on dynamic class imbalance ratio

IR	CSARF	ARF	KUE	LB	CALMID	ROSE	ARFR	SMOTE-OB	OOB	UOB
G-Mean										
Gradual										
Increasing	89.21	80.87	83.60	82.52	82.44	87.83	85.62	88.42	**90.88**	90.76
Inc. then dec	89.57	76.18	80.57	82.08	81.49	87.33	83.11	85.61	89.05	**90.17**
Flipping	84.59	64.34	71.08	69.73	68.01	78.32	65.67	75.83	80.33	**87.78**
Reflipling	84.28	64.72	75.11	72.77	70.23	80.39	64.88	75.23	82.30	**87.35**
Sudden										
Increasing	89.16	81.35	83.75	83.80	82.36	88.00	84.40	85.61	90.89	**90.78**
Inc. then dec	89.88	77.16	81.82	83.10	82.32	87.96	83.41	86.62	89.30	**90.47**
Flipping	84.78	65.41	72.15	71.35	69.09	79.61	66.66	77.52	80.40	**87.89**
Reflipling	84.69	66.12	76.10	74.09	71.74	81.37	66.23	77.26	82.22	**87.41**
Kappa										
Gradual										
Increasing	54.08	71.49	72.98	72.68	73.38	**73.96**	73.61	71.47	68.44	58.68
Inc. then dec	59.21	67.50	71.08	73.14	72.95	**74.98**	71.47	69.89	69.87	61.81
Flipping	45.14	56.97	62.00	61.40	60.20	**66.97**	50.51	61.61	61.73	51.76
Reflipling	43.43	56.80	61.91	62.61	61.36	**67.64**	52.78	60.64	60.86	52.43
Sudden										
Increasing	54.48	72.50	73.53	74.45	73.98	**74.46**	73.87	69.93	68.26	58.46
Inc. then dec	59.87	68.85	72.47	74.70	74.39	**75.82**	72.43	71.44	69.51	61.50
Flipping	45.16	58.50	63.31	63.51	61.54	**68.52**	51.93	63.23	62.09	51.09
Reflipling	43.82	58.28	62.91	64.47	62.96	**69.19**	54.16	62.61	60.66	51.81
Avg. G-Mean	87.02	72.02	78.02	77.43	75.96	83.85	75.00	81.51	85.67	**89.08**
Avg. Kappa	50.65	63.86	67.53	68.37	67.59	**71.44**	62.59	66.35	65.18	55.94
Rank G-Mean	3.83	8.62	7.19	5.83	6.85	4.11	6.66	5.11	3.80	**3.01**
Rank Kappa	8.58	6.31	5.14	4.41	4.68	**2.80**	6.41	4.93	4.64	7.12

Bold font highlights the best result

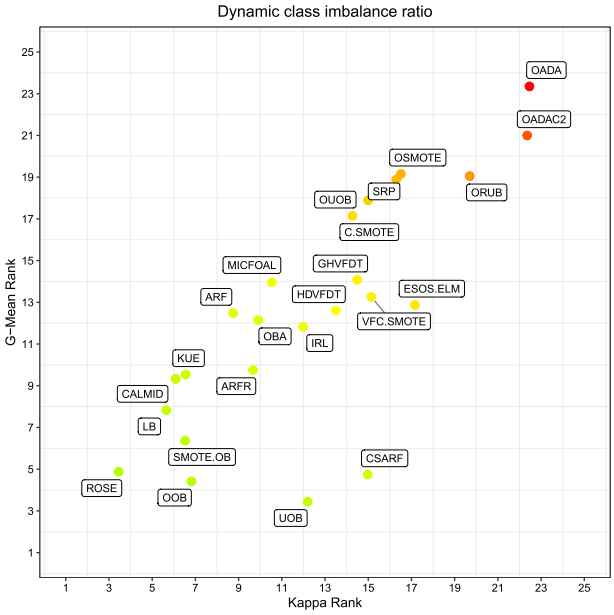

Fig. 10 Comparison of all 24 algorithms for dynamic class imbalance ratio. Color gradient represents the product of both metrics (Color figure online)

imbalance ratio. To increase readability, the line plots were smoothed using a moving average of 20 data points. Table 6 presents the average G-Mean and Kappa for the top 10 classifiers for each of the evaluated dynamic scenarios and the overall rank of the algorithms. Figure 10 provides an overall comparison among all algorithms.

Discussion

Impact of approach to class imbalance. In our second experiment, it is interesting to note that best-performing classifiers are similar to the ones in the static scenario with their difference relying on how quickly they can recover from the imbalance drift. Regarding data-level approaches, UOB and OOB achieve good results, even without having explicit mechanisms for handling concept drift in imbalance ratio. OUOB once again did not display satisfactory results, mainly because of its inability to switch between different resampling approaches that lead to a slower response to changes. SMOTE based methods had diverging performances. C-SMOTE and OSMOTE cannot handle increasing imbalance ratio, losing their performance over time and not being able to cope with the increasing disproportion among classes. This can be explained by the fact that increasing imbalance ratio leads to a lower number of minority instances that could be used for the generation of relevant and diverse artificial instances. SMOTE-OB was among the best-performing classifiers. This can be explained by SMOTE-OB undersampling together with oversampling, leading to smaller disproportions between classes and more homogeneous k-nearest neighborhoods used for instance generation.

For algorithms modification methods, CSARF is the best performing one according to G-Mean but suffers under Kappa metric. When dynamic changes are introduced, ROSE presented the most balanced results according to both metrics. In the previous experiments, ROSE was also one of the best classifiers, but its underlying change adaptation mechanisms and usage of dynamic sliding windows lead to significant improvements for non-stationary imbalance, especially when dealing with high imbalance ratios and flipping role of classes.

Impact of ensemble architecture. Experiments with the dynamic imbalance ratio confirm our previous observations regarding the most robust architecture choice for ensembles. Boosting-based methods return even worse performance when dealing with an evolving disproportion between classes. We can explain this by the fact that each classifier in boosting change may be built using different class ratios, thus further reinforcing the small sample size problem for minority classes observed for static imbalance. This allows us to conclude that boosting-based ensembles are not best suited for handling difficult imbalanced streams. Bagging-based and hybrid architectures perform significantly better, with bagging being a dominating solution. It is very interesting to see that regardless of the used skew-insensitive mechanism bagging-based ensembles (or hybrid architectures like ROSE that utilize bagging) deliver superior performance. This can be explained by the diversity among base classifiers that allow for anticipating different local characteristics of decision boundaries. Therefore, with increasing or decreasing of the imbalance ratio there is a high chance that some of the base classifiers (and thus a subset of instances that they use) offer better generalization and faster adaptation to evolving disproportions between classes.

Impact of drift speed in class imbalance. We can observe that most of the examined algorithms offer similar performance on all types of drifts. Some of the methods do not have explicit mechanisms for change adaptation and this leads to their slower recovery from changes. However, in the long run, there were no significant differences between sudden and gradual drift adaptations for all methods on G-Mean or Kappa. However, the third analyzed scenario with flipping the majority and minority classes has a major impact on analyzed classifiers. ARF significantly suffers on both metrics, showing that its adaptation mechanisms are not suitable for settings where classes can change roles over time. The same observation holds for CSARF that displays an increasing gap between performances on majority and minority classes, as it is not able to effectively adapt its cost matrix to such changes and penalizes wrong class over time. Considering only G-Mean, flipping classes did not impact much UOB demonstrating that undersampling displays potential robustness to switching minority class. However, the same cannot be said for the Kappa metric, leading us to conclude that UOB tends to prioritize one class even when their roles flip. ROSE was the most robust and stable classifier for all possible changes in class distribution, being capable of avoiding huge drops of performance for the metrics due to its per-class sliding window adaptation.

7.1.3 Instance-level difficulties

Goal of the experiment. This experiment addresses **RQ3** and evaluates the robustness of the classifiers to instance-level difficulties (Brzeziński et al., 2021). We evaluated the Brzeziński generator with borderline or rare instances, and combining both at the same time. The ratio for difficult instances for scenarios where there are only rare or borderline instances are {0%, 20%, 40%, 60%, 80%, 100%}. In the combined scenario they represent {0%, 20%, 40%} of rare and borderline instances, e.g. 20% means there are 20% rare instances and 20% borderline instances. Difficult instances were created for the minority class to present a challenging scenario for the classifier. We evaluated the influence on classifiers combined with static and dynamic imbalance ratios. Borderline instances pose a challenge to the classifier because they lie in the uncertainty area of the decision space and strongly impact the induction of the classification boundaries. Rare instances are

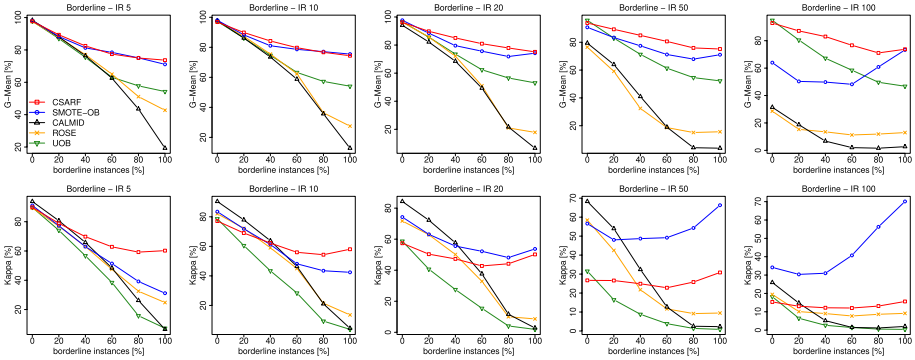

Fig. 11 Robustness to borderline instances for static imbalance ratio (G-Mean and Kappa)

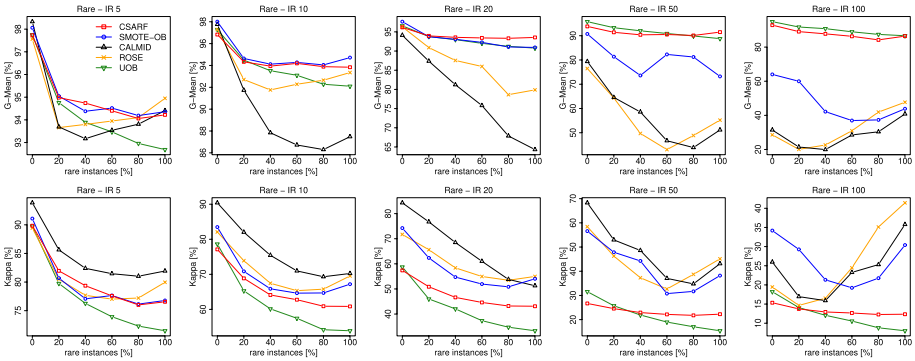

Fig. 12 Robustness to rare instances for static imbalance ratio (G-Mean and Kappa)

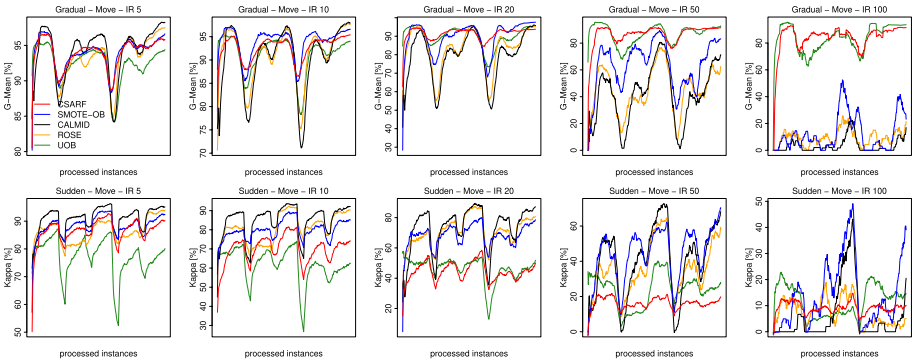

Fig. 13 G-Mean and Kappa on moving minority clusters for static imbalance ratio

overlapping with the majority class. Also, instances of minority classes are distributed in clusters. Moving, splitting and merging these clusters leads to new challenges for the classifiers since the decision boundary moves accordingly.

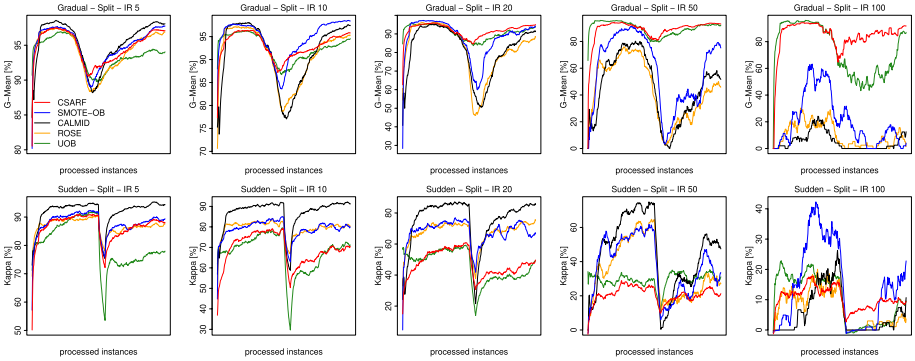

Fig. 14 G-Mean and Kappa on splitting minority clusters for static imbalance ratio

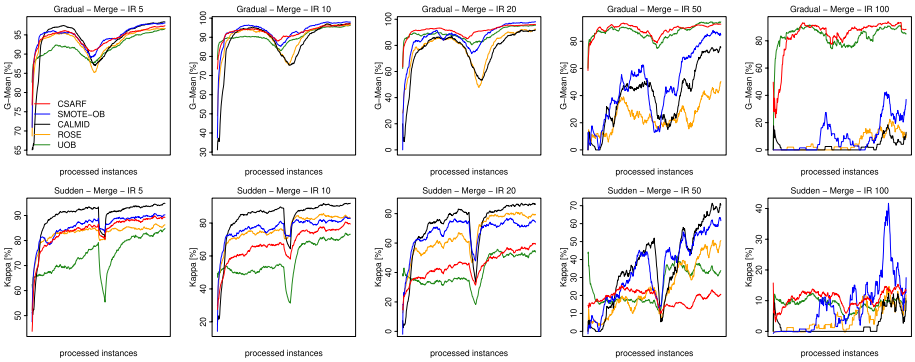

Fig. 15 G-Mean and Kappa on merging minority clusters for static imbalance ratio

Figures 11 and 12 present the performance of the five selected classifiers with the increasing presence of borderline and rare instances respectively under static imbalance ratio. Figures 13, 14, 15 illustrate the performance of the same classifiers with changes in the spatial distribution of minority class instances under static imbalance. Table 7 presents the average G-Mean and Kappa for the top 10 classifiers for each imbalance ratio and a given instance-level difficulty, and their average ranking. The overall performance of all classifiers regarding each of the instance difficulties is presented in Figs. 16, 17,18, 19, 20, in which axes of the ellipse represent G-Mean and Kappa metrics, the more rounded the better, and the color represents the product of both metrics.

Considering the instance-level difficulties combined with dynamic imbalance ratio, Fig. 21 illustrates the performance of the classifiers with increasing imbalance ratio. Table 8 presents the average G-Mean and Kappa for the top 10 classifiers for increasing imbalance ratio in the presence of instance-level difficulties, and their overall ranking. To summarize, Fig. 22 shows the overall performance of all classifiers for each instance-level difficulty.

Discussion

Impact of approach to class imbalance. First, let us look on how different mechanisms for ensuring robustness to class imbalance tend to perform under diverse data-level

Table 7 G-Mean and Kappa averages on borderline, rare, moving, splitting, merging minority clusters for static imbalance ratio

Instance-level diff.			CSARF	ARF	KUE	LB	CALMID	ROSE	ARFR	SMOTE-OB	OOB	UOB
G-Mean	IR 5	Borderline	95.02	93.05	92.93	93.95	94.49	94.67	**95.27**	95.09	95.04	94.25
		Rare	**82.62**	61.15	59.84	61.64	64.69	70.21	74.15	82.05	67.86	72.57
		Moving	95.45	94.81	92.75	94.99	95.70	95.24	**96.32**	95.70	93.51	92.69
		Splitting	95.93	95.06	94.34	95.25	96.45	95.51	**96.86**	96.20	95.37	94.13
		Merging	95.47	94.06	92.57	94.55	95.57	94.44	**96.39**	95.86	94.28	92.55
	IR 100	Borderline	87.78	16.83	18.57	24.24	28.75	31.98	14.27	47.38	64.69	**90.09**
		Rare	**80.76**	2.55	3.03	7.74	10.53	15.59	10.10	57.74	39.45	66.26
		Moving	**85.32**	0.06	5.84	3.81	9.53	14.20	10.50	18.86	53.82	84.30
		Splitting	**86.32**	1.05	4.27	3.49	9.06	15.84	11.19	24.81	61.12	73.18
		Merging	**86.14**	0.88	4.26	3.64	5.35	9.62	6.18	11.47	51.49	84.44
Kappa	IR 5	Borderline	80.22	84.16	82.71	84.19	**84.38**	80.30	81.52	79.91	80.70	77.26
		Rare	**70.14**	51.37	50.18	51.64	53.56	55.70	61.70	58.85	52.83	47.03
		Moving	82.56	**89.00**	84.87	88.27	88.70	84.55	86.94	83.77	79.74	76.18
		Splitting	84.29	**90.40**	87.49	88.97	90.34	85.03	88.67	85.76	83.99	79.91
		Merging	82.18	**88.99**	84.75	88.03	88.59	81.57	86.74	84.65	80.74	73.56
	IR 100	Borderline	13.22	13.52	15.08	20.03	23.86	25.27	2.51	26.03	**38.09**	11.97
		Rare	13.56	1.96	2.16	5.94	8.43	10.68	3.00	**43.78**	24.51	4.96
		Moving	9.19	0.05	4.14	2.80	7.37	9.31	1.79	11.19	**31.14**	10.98
		Splitting	10.44	0.83	3.09	2.62	6.98	9.98	1.92	13.20	**36.78**	9.94
		Merging	10.92	0.65	3.27	2.66	4.05	6.36	1.33	8.15	**34.58**	8.94
All IRs	Avg. G-Mean		**88.40**	52.60	53.32	57.81	62.98	65.06	64.94	77.21	75.87	83.10
	Avg. Kappa		50.01	46.06	45.16	49.79	53.68	51.26	47.05	**56.75**	54.67	41.29
All IRs	Rank G-Mean		**2.06**	8.13	9.17	7.10	5.69	5.78	4.42	3.61	4.34	4.71
	Rank Kappa		5.65	5.63	7.06	4.49	**2.85**	5.29	5.99	4.52	4.87	8.64

Bold font highlights the best result

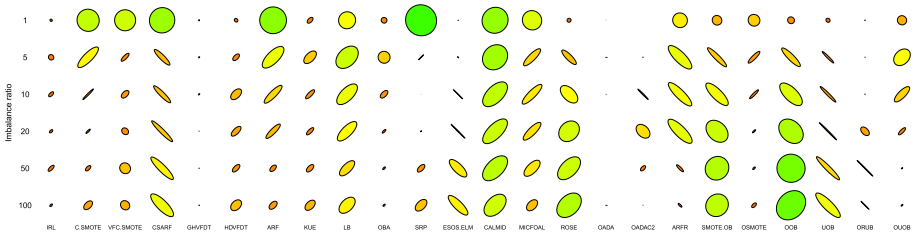

Fig. 16 Comparison of all 24 algorithms for borderline instances on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics (Color figure online)

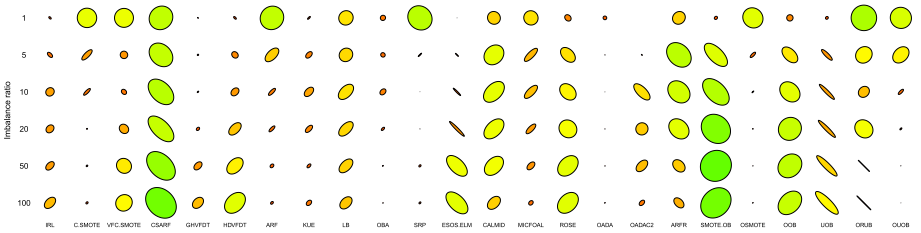

Fig. 17 Comparison of all 24 algorithms for rare instances on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics (Color figure online)

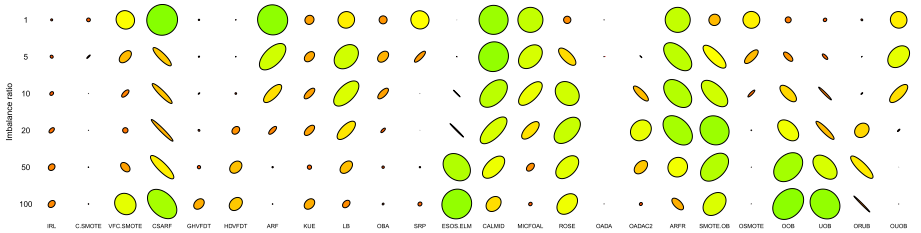

Fig. 18 Comparison of all 24 algorithms for moving minority clusters on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics (Color figure online)

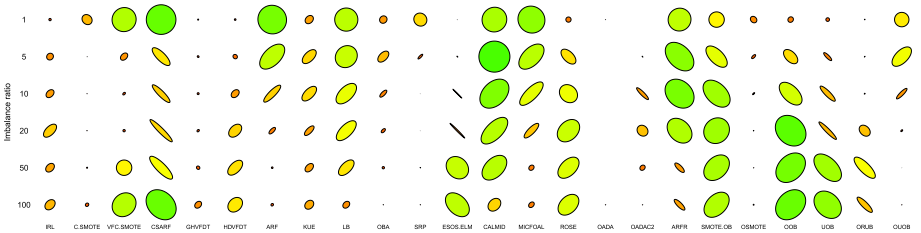

Fig. 19 Comparison of all 24 algorithms for splitting minority clusters on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics (Color figure online)

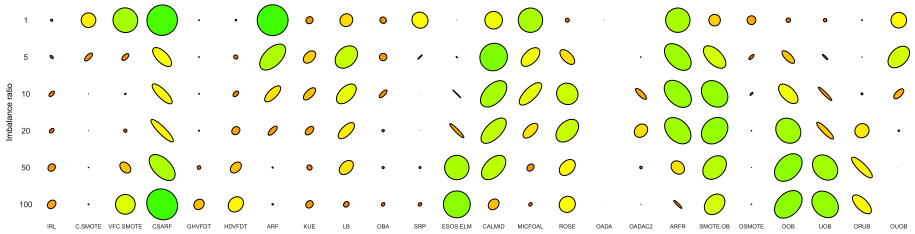

Fig. 20 Comparison of all 24 algorithms for merging minority clusters on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics (Color figure online)

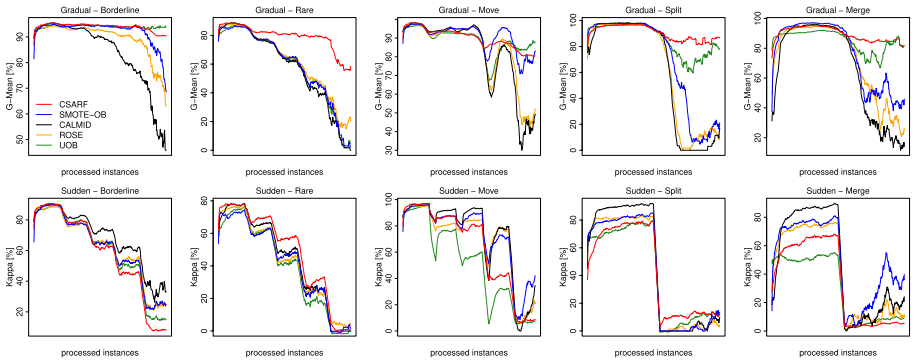

Fig. 21 G-Mean and Kappa on increasing borderline, rare, moving, splitting, and merging minority clusters and increasing imbalance ratio

difficulties. While guided resampling solutions usually perform better than their blind counterparts, here we can see that most of approaches based on SMOTE tend to fail. This can be explained by reliance of SMOTE on the neighborhood. Borderline and rare instances create non-homogeneous neighborhoods that are characterized by high overlapping and classification uncertainty. Oversampling such areas will lead to enhancing these undesirable qualities, instead of simplifying the classification task. This is especially visible for rare instances, where SMOTE-based methods deliver the worst performance. Rare instances do not form homogeneous neighborhoods and in streaming setup may appear infrequently, leading to lack of both spatial and temporal coherencies. This undermines the basic assumptions of SMOTE-based algorithms and renders them ineffective. Only SMOTE-OB can handle both types of minority instances, as well as various minority clusters. This can be explained by using bags of instances (subsets) for training, which may offer better separation of instances and impose partial coherence on artificially generated instances. Blind oversampling employed by OOB also performs well under data-level difficulties, as it does not rely on the neighborhood analysis. However, when dealing with rare instances, OOB tends to significantly fail, as it amplifies these often-scattered instances, leading to overfitted decision boundaries. Where OOB and UOB excel is for minority class clusters, as due to their online nature they can swiftly adapt to changes in cluster structures and resample even small drifts to make them viable for their base learners. Algorithms based on training modifications, such as HDVFDT, ROSE, or CALMID can handle well all types of difficulties. Their robustness lies in their data manipulation and usage of modified

Table 8 G-Mean and Kappa averages on borderline, rare, moving, splitting, merging minority clusters and increasing imbalance ratio

Instance-level diff.	CSARF	ARF	KUE	LB	CALMID	ROSE	ARFR	SMOTE-OB	OOB	UOB
G-Mean										
Borderline	93.73	81.55	80.95	82.08	82.22	88.58	89.19	91.89	93.81	**93.97**
Rare	**77.98**	55.37	54.81	55.94	55.42	60.65	55.85	56.78	56.58	56.34
Moving	**92.00**	66.93	69.21	71.70	80.45	81.54	79.93	87.91	89.35	87.61
Splitting	**90.01**	51.00	53.83	51.63	55.09	57.65	56.19	59.40	80.39	78.81
Merging	**88.10**	52.07	50.19	60.06	60.64	63.47	57.99	69.83	81.59	82.82
Kappa										
Borderline	57.38	68.43	67.06	67.23	**68.56**	61.26	65.30	61.88	61.92	59.73
Rare	**47.66**	44.43	43.16	44.07	44.87	42.83	43.34	41.70	41.94	39.34
Moving	56.13	60.61	60.35	64.81	**71.25**	68.12	70.10	69.19	62.31	50.86
Splitting	42.51	46.48	46.84	46.36	48.25	45.38	45.78	45.53	**50.90**	42.41
Merging	34.26	44.70	42.53	50.28	51.19	46.32	43.12	**51.75**	50.50	30.08
Avg. G-Mean	**87.76**	62.28	62.61	65.06	66.88	70.73	68.54	73.43	79.71	79.55
Avg. Kappa	47.83	53.15	52.28	54.55	**56.63**	52.30	53.46	53.60	53.06	45.40
Rank G-Mean	**1.42**	9.17	9.58	7.42	7.25	4.75	6.25	3.75	2.75	2.67
Rank Kappa	8.00	4.17	5.67	3.75	**1.71**	6.50	5.08	5.63	5.17	9.33

Bold font highlights the best result

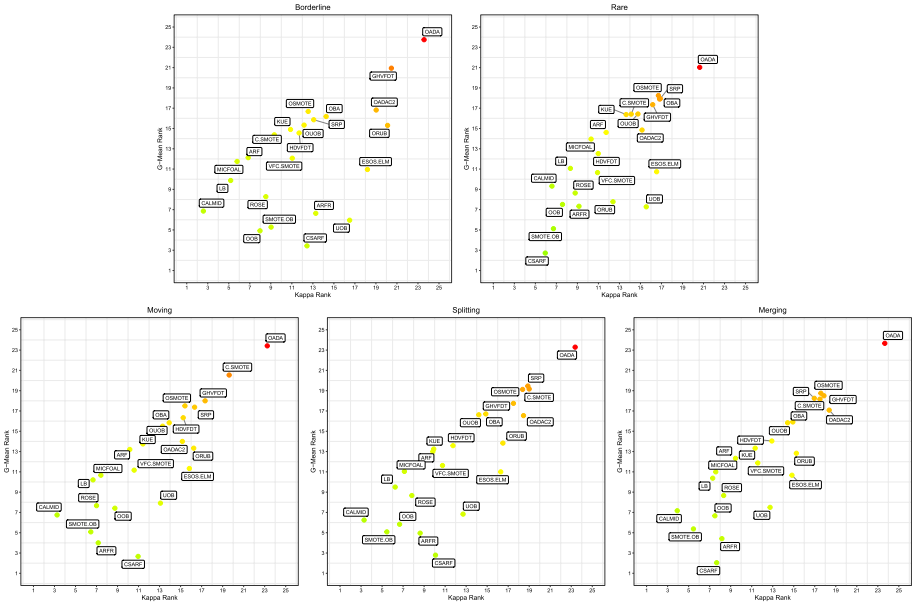

Fig. 22 Comparison of all 24 algorithms on borderline, rare, moving, splitting, merging minority clusters and increasing imbalance ratio. Color gradient represents the product of both metrics (Color figure online)

mechanisms for training, which can display all-around, yet implicit, robustness to such challenges. Especially for Kappa metric, ROSE and CALMID can be seen as good choices for these scenarios. Their offshoot, a cost-sensitive approach of CSARF, displays best performance on G-Mean metric, yet suffers under Kappa evaluation. This shows that CSARF strongly focuses on the minority class, but this hits back in the form of an increased number of false positives. So, while it is capable of handling minority difficulties and clusters by cost penalty-lead training, it increases the false positives in these overlapping or uncertain regions.

Impact of ensemble architecture. We can observe that all best-performing methods for various types of data-level difficulties are based either on bagging or hybrid architectures. All boosting methods are among the worst performing ones. This can be explained by the nature of boosting, as it focuses on correcting the mistakes of the previous classifier in the ensemble. Rare, borderline, or clustered minority instances will always introduce a high uncertainty into the training procedure. This may significantly destabilize boosting, as by focusing on correcting errors on those uncertain instances it will be continuously introducing other errors, locking itself in a cycle of never reducing the overall error. Bagging methods offer natural partitioning of instances, allowing to break difficult neighborhood or clusters and introduce more instance-level diversity into base classifiers. This aids the used mechanisms for handling class imbalance, making bagging methods more robust to scenarios where learning difficulties lie in spatial characteristics of data.

Comparison with standard ensembles. Interestingly, general-purpose ensembles display better robustness to various instance-level difficulties than over half of the classifiers dedicated to imbalanced data streams. LB, ARF, and KUE can relatively effectively handle both types of difficult instances, as well as various types of evolving clusters within the minority class. They always significantly outperform all methods based on boosting and

most of approaches using informative oversampling (except for SMOTE-OB). Of course, we can observe a drop in their performance with the increase of imbalance ratios, yet even for IR = 100 they can perform better than several dedicated approaches. Their robustness to instance types can be explained by the fact that all three mentioned ensembles use instance subsets for training their base classifiers. Therefore, such subsampling may implicitly lead to more sparse neighborhoods (reducing overlapping and uncertainty) and thus to reduction of difficulty levels for certain instances. When analyzing the robustness to evolving clusters, one can explain this by concept drift adaptation mechanisms employed by LB, ARF, and KUE. Changes in minority class clusters can be picked up by their drift detectors, leading to adaptation to the current state of the minority class. Therefore, any splitting or merging of clusters will be picked up as changes in data distributions and managed by simple online adaptation to the most recent instances. Finding that such general-purpose ensembles display significant robustness to data-level difficulties stands as a testament to how well designed those methods are. However, to excel when dealing with such challenging learning scenario, highly specialized ensemble models enhance with skew-insensitive mechanisms can deliver much better performance.

Relationships between instance-level difficulties and imbalance ratios. When analyzing algorithms for their robustness to data-level difficulties, we must understand the relationship between them and the class imbalance ratio. Ideally, we are looking for a method that will be insensitive to changing imbalance ratios and will display stable robustness to data-level difficulties. Most of the existing algorithms do not possess this quality, displaying either drops in the performance with increasing imbalance ratio (e.g. MICFOAL), or lack of any stability (e.g. VFC-SMOTE). The most reliable methods are CSARF, SMOTE-OB, OOB, and ROSE that offer stable, or improving, robustness with increasing imbalance. It is important to note that CSARF performance is skewed towards G-Mean, while the remaining methods tend to perform well on both metrics.

What difficulties are the most challenging. While analyzing the performance of the methods, we can see significant drops in the performance in two scenarios: when dealing with rare instances and splitting/merging clusters. Rare instances are one of the biggest challenges for any imbalanced algorithms, as they combine small sample size, class overlapping, and potential presence of noise. With increasing ratio of rare instances, the minority class in the stream starts losing any coherent structure, converging towards a collection of sparsely distributed and spatially uncorrelated instances, more akin to a cloud of points than any structure. This makes the formulation of decision boundaries especially difficult and requires dedicated mechanisms that can either learn under small sample size or can create more coherent representations (either via resampling like OOB, or via instance buffers like ROSE). Minority clusters pose even bigger challenge, as they force classifiers to track sub-concepts in minority classes (each cluster should be treated as a sub-concept). Both cases require fast adaptation and are strongly aided by a presence of underlying drift detector. With splitting clusters, previously learned decision boundaries become to general and are not able to capture the emergence of sub-concepts in minority class. With merging clusters, we are left with too complex decision boundaries that are not able to generalize well over the current state of the stream.

7.1.4 Concept drift and static imbalance ratio

Goal of the experiment. This experiment aims to address **RQ4** and to evaluate the robustness of the data stream classifiers to the static imbalance in the presence of concept drift.

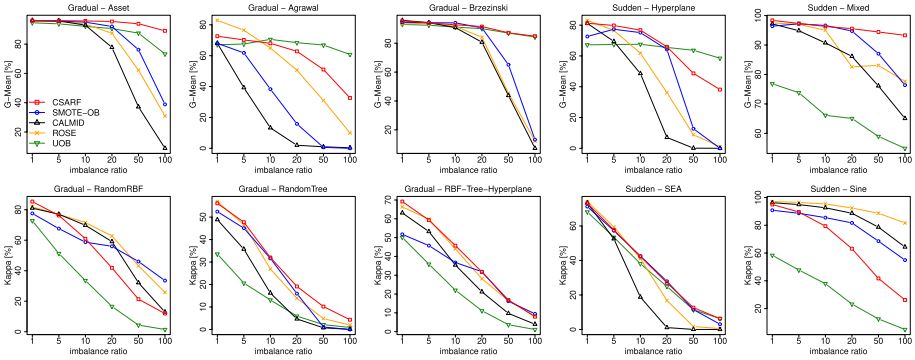

Fig. 23 Robustness to concept drift with static class imbalance ratio (G-Mean and Kappa)

Table 9 G-Mean and Kappa averages of all 10 streams for concept drift with static class imbalance ratio

IR	CSARF	ARF	KUE	LB	CALMID	ROSE	ARFR	SMOTE-OB	OOB	UOB
G-Mean										
1	87.83	87.82	85.44	87.99	86.31	**88.70**	87.71	83.74	78.80	77.90
5	**86.62**	67.56	71.74	75.41	77.08	84.74	83.96	84.60	76.62	76.35
10	**84.64**	46.23	52.48	54.02	61.48	75.57	75.44	80.31	71.15	74.96
20	**81.06**	31.49	38.67	38.94	41.65	59.10	57.25	70.20	60.33	72.97
50	**73.97**	15.43	19.40	20.63	23.71	37.90	28.59	42.82	36.64	68.99
100	**64.86**	9.93	12.11	13.16	11.71	21.80	9.53	21.54	24.81	63.78
Kappa										
1	75.76	75.76	71.67	76.10	72.73	**77.52**	75.59	69.71	57.93	56.23
5	66.58	56.51	58.24	64.02	64.12	**70.99**	68.15	62.24	52.71	45.30
10	53.18	39.00	42.57	45.61	50.06	**60.09**	54.30	53.17	45.58	33.72
20	37.11	26.41	30.96	32.65	34.34	**45.03**	33.83	42.97	36.70	22.06
50	19.26	12.69	15.89	17.31	19.04	**28.36**	13.39	26.54	22.33	10.14
100	10.54	8.65	10.42	11.70	10.15	**17.25**	2.82	15.11	15.79	4.52
Avg. G-Mean	**79.83**	43.08	46.64	48.36	50.32	61.30	57.08	63.87	58.06	72.49
Avg. Kappa	43.74	36.50	38.29	41.23	41.74	**49.87**	41.35	44.96	38.51	28.66
Rank G-Mean	**1.83**	7.95	7.54	6.69	6.50	4.21	4.93	4.27	5.99	5.08
Rank Kappa	3.95	7.05	6.68	5.50	5.40	**2.92**	4.88	4.72	5.93	7.97

Bold font highlights the best result

Even though the classifiers are designed to deal with imbalance ratios, they also have mechanisms to deal with concept changes. Concept drift affects decision boundaries, thus leading to a more challenging skewed learning scenario with a higher degree of overlap between classes. To evaluate this, we prepared the same generators used in experiment Sect. 7.1.1 with two types of concept drift: gradual and sudden. They were combined with the static imbalance ratio examined in experiment Sect. 7.1.1. Figure 23 illustrates the G-Mean and Kappa over time for the five selected classifiers with static imbalance ratio under the presence of concept drift. Table 9 presents the G-Mean and Kappa for the top 10

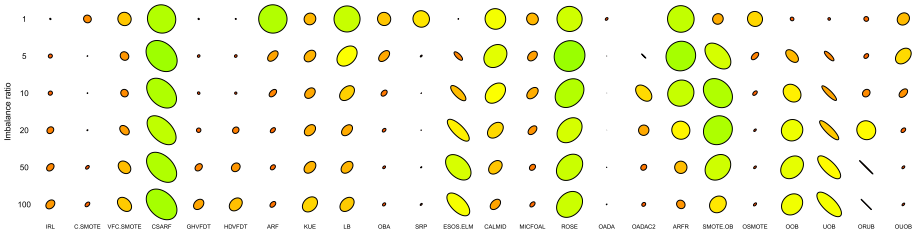

Fig. 24 Comparison of all 24 algorithms for concept drift with static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics (Color figure online)

classifiers and each imbalanced ratio, and their overall ranking as well. Figure 24 summarizes the overall performance of all classifiers in this scenario.

Discussion

Impact of approach to class imbalance. In this experiment, we extend the problem of analyzing the robustness of classifiers to various imbalance ratios by adding concept drift affecting the decision boundaries. It is important to notice that the drift did not influence the disproportion between classes. OOB and UOB, two methods that offered excellent performance for stationary and imbalanced streams suffer from a significant drop in performance when handling non-stationary problems. OOB had the biggest performance drop under concept drift for higher imbalance ratios. This is expected since both methods do not have mechanisms to deal with changes in feature distribution. While the changes in the imbalance ratio could be tackled by resampling approaches, they do not allow for any efficient adaptation to evolving decision boundaries. Classifiers based on informed resampling, such as C-SMOTE, OSMOTE and VFC-SMOTE offered only a slightly better performance than the mentioned blind resampling ensembles. This shows that under the presence of concept drift, adaptation mechanisms play a more important role than the solutions used to tackle class imbalance. For algorithm-level methods, CSARF demonstrated the best results, thanks to its underlying implicit mechanisms for handling non-stationary data. While, similarly to previous experiments CSARF suffered under Kappa evaluation, this time it was the second-best regarding this metric. ROSE remained as the most balanced classifier displaying robustness to changes since it can adapt both to concept drift and imbalance ratio. ARF, ARFR, LB and SRP achieved decent results in a scenario with concept drift, however, their performance drops as the imbalance ratio increases.

Impact of ensembles architecture. As observed in the previous experiments, boosting-based methods deliver the worst performance among all ensemble architectures. This can be explained by drift destabilizing boosting classifier chains, as errors made by previous classifiers may no longer be meaningful for the updating of their follow-ups. There is a need to improve drift adaptation procedures for boosting-based ensembles so they can become competitive with their bagging peers. While bagging-based architectures are still the core of the best-performing methods, we can see the increasing dominance of hybrid architectures for concept drift scenarios. While all of them use bagging, they combine it with the dynamic weighting of the base classifier and dynamic line-up, demonstrating that a combination of several mechanisms is necessary to tackle class imbalance and concept drift at the same time.

Relationship between concept drift and imbalance ratios. In the context of this experiment, it is crucial to analyze and understand the interplay between the concept drift impacting the class boundaries and static imbalance ratios affecting the disproportion

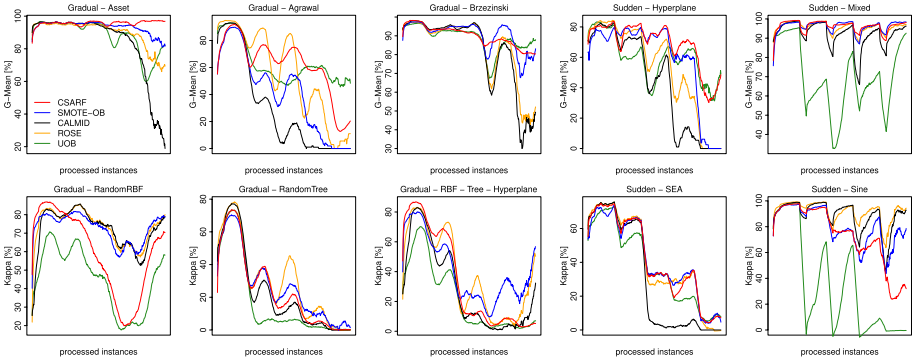

Fig. 25 G-Mean and Kappa on concept drift with increasing imbalance ratio

Table 10 G-Mean and Kappa averages of all 10 streams for concept drift with increasing class imbalance ratio

Drift	CSARF	ARF	KUE	LB	CALMID	ROSE	ARFR	SMOTE-OB	OOB	UOB
G-Mean										
Sudden	**85.96**	55.56	62.45	62.61	64.19	74.05	60.64	75.79	72.91	71.00
Gradual	**77.90**	45.73	53.47	48.12	53.78	64.60	50.70	70.29	66.23	70.32
Kappa										
Sudden	55.13	48.88	52.29	55.39	55.50	**63.23**	52.70	57.48	48.48	33.09
Gradual	41.57	36.44	40.59	39.22	42.33	**48.45**	39.77	45.67	39.12	29.69
Avg. G-Mean	**81.93**	50.64	57.96	55.36	58.98	69.33	55.67	73.04	69.57	70.66
Avg. Kappa	48.35	42.66	46.44	47.31	48.91	**55.84**	46.23	51.57	43.80	31.39
Rank G-Mean	**1.10**	9.30	7.20	7.70	6.45	4.10	7.35	3.30	4.35	4.15
Rank Kappa	4.75	8.00	6.35	5.90	4.70	**1.75**	5.65	3.10	5.60	9.20

Bold font highlights the best result

between them. While focusing on how the classifiers try to tackle concept drift, we do not see significant differences between the ones utilizing implicit or explicit drift detection. This shows that there is no obvious choice for adaptation mechanisms and that the classifier performance for drifting and imbalance streams is a product of their learning architecture, drift adaptation mechanism, and approach to tackling class imbalance. We can see that popular classifier for drifting data streams, such as ARF, LB, or SRP cannot handle increasing imbalance ratios. At the same time, solutions dedicated to online learning from imbalanced data streams, such as UOB or OOB cannot deal with the non-stationary nature of data streams. Best performing methods, such as ROSE, CSARF and SMOTE-OB combine adaptation and skew-insensitive mechanisms for all-round robustness.

7.1.5 Concept drift and dynamic imbalance ratio

Goal of the experiment. This experiment was designed to complement the previous experiment, and completely address **RQ2** and **RQ4**, examining the classifiers in the presence of concept drift combined with dynamic imbalance ratio. Combining concept

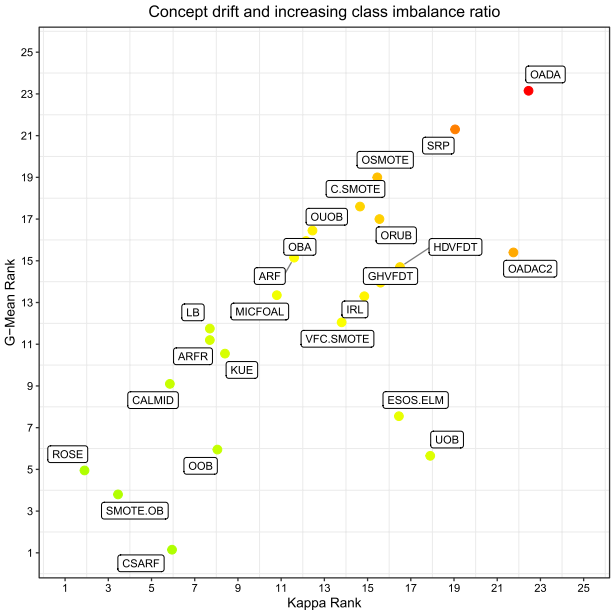

Fig. 26 Comparison of all 24 algorithms for concept drift with increasing class imbalance ratio. Color gradient represents the product of both metrics (Color figure online)

drift at the same time with changes in the class imbalance poses a complex challenge to classifiers. To evaluate this, we prepared the same generators in experiment Sect. 7.1.4 with gradual and sudden concept drift, and combined them with the dynamic increasing imbalance ratio proposed in experiment Sect. 7.1.2. Figure 25 illustrates the performance of the selected classifiers with dynamic increasing imbalance ratio under the presence of concept drift. Table 10 presents the G-Mean and Kappa for the top 10 classifiers for each type of concept drift and the average ranking for each evaluated metric. Figure 26 provides an overall comparison of all classifiers in the proposed scenario.

Discussion

Impact of approach to class imbalance. Let us focus on changes in the behavior of classifier as compared with the previous case of evolving class imbalance without explicit concept drift. All methods based on blind resampling display drops in performance, as usually they lack explicit mechanisms for handling concept drift, leading to their deterioration over time. SMOTE based methods followed the behavior experienced in previous experiments, mainly because concept drift and increasing imbalance ratio may lead to temporal incoherence which can enhance problems of oversampling. Only SMOTE-OB displayed satisfactory robustness to simultaneously evolving imbalance ratio and concept drift, while additionally achieving good balance between Kappa and G-Mean metrics.

Classifiers based on training modifications, such as ROSE and CALMID displayed robustness to concept drift and dynamic imbalance ratio, especially for the G-mean metric. Their training procedures provide reliability in a scenario where multiple changes happen simultaneously. The cost-sensitive approach of CSARF presents outstanding results regarding the G-Mean metric, with almost 1 as the average rank. Nevertheless, when analyzing the Kappa metric, we can see shortcomings of CSARF, where it ranks the third. This shows that the CSARF adaptation to evolving data characteristics is not balanced over both classes. ROSE displays balanced performance on both metrics, which can be explained by

the combination of concept drift detector with balanced buffers for each of classes, allowing for equalized performance on the majority and minority classes.

Impact of ensemble architecture. This highly difficult scenario further shows that bagging-based and hybrid architectures are the only ones capable of handling drifting and evolving class imbalance. Their superiority over boosting methods becomes even more evident in these experiments. However, another interesting observation is the increasing gap between dynamic and static ensemble line-ups. Here we can see that most of the best performing methods use dynamic replacement of the ensemble members. This can be explained by the fact that when concept drift is combined with evolving class imbalance, especially under rapid changes, it is more efficient to train a new classifier from scratch and replace the weakest one, instead of trying to adapt the existing members to a vastly different new concept.

Impact of concept drift speed. As mentioned in the previous observation, the speed of changes (velocity of concept drift) significantly impacts the classifiers. We observed that all of the classifiers tend to react worse to gradual drift, while displaying better robustness on sudden drift. While this observation can be surprising, we can explain it by taking a deeper look on how the adaptation mechanisms work in these ensembles. Under sudden concept drift, we cn observe a rapid deterioration of the ensemble performance. However, new instances coming from a stable concept are readily available, allowing for a recovery and adaptation with sufficient sample size. When dealing with gradual drift, classifiers do not see the new, fully formed concept so quickly. Therefore, the adaptation process becomes more tedious, as the sample size from the new concept may not be big enough. This may mislead some pruning or weighting mechanisms, forcing costly false adaptations. While in case of gradual drift we do not observe one single drop of performance, the negative impact of change is prolonged over time and thus may sum up to a bigger challenge for the classifier in the long run.

Relationship between concept drift and increasing class imbalance. In this scenario each classifier must be able to simultaneously handle concept drift (impacting the decision boundaries) and evolving class imbalance ratio (impacting both skew-insensitive mechanisms and decision boundaries). This creates a trade-off, with classifiers displaying different behavior patterns. Some methods, like LB or KUE display high adaptability to concept drift. Others, like OOB focus on robustness to evolving class imbalance. The most balanced method, offering best trade-off between those two factors is ROSE, followed by SMOTE-OB and CSARF.

7.1.6 Real-world binary class imbalanced datasets

Goal of the experiment. This experiment was designed to address **RQ5** and to evaluate the performance of the classifiers on 19 real-world imbalanced data streams. The previous experiments focused on analyzing how the classifiers cope with various learning difficulties present in imbalanced data streams using synthetic generators, allowing us to inspect how the classifiers behave in specific and controlled scenarios. Meanwhile, real-world datasets pose specific challenges to classifiers, as they are not generated in a controlled environment. They are characterized by a combination of various learning difficulties that appear with varying intensity or frequency. Their imbalance ratio changes over time, while concept drift may oscillate among different types with varying speed. Therefore, assessing the performance of all classifiers on real-world data is a major step towards evaluation. The real-world data streams employed in the experiments are popular benchmarks

Table 11 Real-world binary datasets specifications

Dataset	Instances	Features
adult	45,222	14
amazon	8000	30
amazon-emp	32,769	9
census	299,284	41
coil2000	9,822	85
covtype	267,001	54
creditcard	284,807	30
electricity	45,312	8
gmsc	150,000	10
hepatitis	1,000,000	19
internet-ads	3,279	1,558
kddcup	494,021	41
nomao	34,465	118
pakdd	50,000	27
poker	359,999	10
spam	9,324	499
tripadvisor	18,569	30
twitter	9,090	30
weather	18,159	8

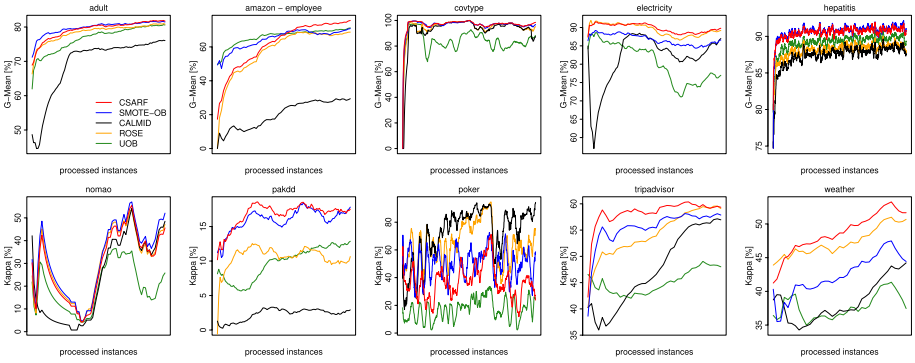

Fig. 27 G-Mean and Kappa on binary class imbalanced datasets

for imbalanced data streams classifiers, and their specifications are presented at Table 11. Figure 27 illustrates the performance of the five selected classifiers in the real-world datasets. Table 12 presents the performance for the top 10 classifier on each dataset. Figure 28 summarizes the overall performance of all classifiers in the real-world datasets scenario.

Characteristics of real-world imbalanced data streams. Before analyzing the classifiers' performance in real-world datasets, it is important to point up the difference between artificial and real-world imbalanced data streams. Generators are probabilistic and base the generation of instances on prior probability taken from the parametric imbalance ratio. Their appearance in the stream is dictated strictly by these priors, leading to bounded

Table 12 G-Mean and Kappa on binary class imbalanced datasets

Dataset	CSARF	ARF	KUE	LB	CALMID	ROSE	ARFR	SMOTE-OB	OOB	UOB
G-Mean										
adult	80.09	72.58	69.56	72.65	72.15	79.20	**80.61**	80.40	79.28	77.81
amazon	67.19	52.05	32.57	25.17	24.35	62.38	50.54	67.18	58.15	**68.11**
amazon-emp	**23.15**	6.63	0.00	4.08	5.40	20.77	18.42	23.04	11.00	18.78
census	**48.67**	27.79	22.97	29.11	32.45	39.29	41.84	44.01	36.57	45.59
coil2000	56.59	4.74	4.51	6.77	2.86	18.58	17.94	50.89	19.60	**61.83**
covtype	**97.22**	94.35	90.39	93.98	93.73	95.39	93.40	96.99	94.08	84.57
creditcard	**32.47**	27.37	16.02	25.07	26.46	28.34	27.54	28.25	29.99	32.17
electricity	**89.79**	89.42	70.49	88.71	83.11	89.14	89.45	86.67	81.67	79.67
gmsc	**76.58**	30.55	32.02	30.31	43.89	58.33	65.77	71.65	71.35	70.65
hepatitis	90.56	85.66	85.37	86.36	87.45	88.33	90.39	**90.75**	89.71	89.25
internet-ads	**13.90**	13.54	0.00	12.98	12.98	13.36	0.00	**13.90**	13.36	13.36
kddcup	7.71	7.26	4.19	7.44	7.61	7.83	7.27	7.52	7.65	**7.86**
nomao	42.91	37.38	30.35	34.89	34.24	40.09	35.68	42.13	40.04	**43.73**
pakdd	**59.48**	2.09	12.82	4.37	15.66	39.71	57.06	54.60	54.66	55.06
poker	**79.61**	45.27	53.38	57.52	76.98	72.81	49.88	62.31	73.39	72.45
spam	**80.63**	74.24	67.13	73.13	72.15	77.44	79.37	80.31	75.80	72.86
tripadvisor	64.11	18.66	14.21	22.54	21.83	45.74	57.10	64.06	60.92	**64.55**
twitter	**79.61**	78.53	58.03	76.34	55.57	77.14	77.77	78.95	71.39	70.39
weather	**76.45**	68.02	59.75	67.38	67.86	74.28	74.19	73.53	69.68	67.19
Kappa										
adult	53.17	53.80	52.31	54.05	52.80	**55.68**	53.60	53.68	52.72	47.99
amazon	29.30	31.99	12.99	7.78	5.38	**33.07**	17.84	26.25	25.85	13.00
amazon-emp	1.91	0.35	0.00	0.13	0.19	**2.42**	0.88	1.21	0.59	1.15
census	20.87	19.72	14.66	20.47	22.13	25.95	23.46	**26.78**	24.20	22.75
coil2000	12.53	1.24	0.74	2.23	0.41	6.22	2.65	**13.20**	5.82	9.49
covtype	85.26	**91.22**	83.43	89.94	88.60	90.13	83.64	89.51	83.21	49.74
creditcard	17.96	26.43	15.45	24.30	25.56	27.30	22.61	27.33	**28.44**	6.71
electricity	79.89	**79.95**	50.14	78.44	68.81	78.64	79.70	73.91	65.57	62.13
gmsc	29.09	15.98	17.02	15.74	26.26	35.23	31.53	**35.75**	35.22	23.95
hepatitis	72.92	76.83	75.32	76.62	**77.30**	74.26	74.73	75.33	74.17	69.47
internet-ads	**13.53**	13.12	0.00	12.48	12.48	12.91	0.00	**13.53**	12.91	12.91
kddcup	7.04	7.14	3.56	7.31	7.45	**7.49**	6.99	7.25	7.27	6.26
nomao	32.56	32.70	18.71	29.71	28.11	33.24	29.11	**36.02**	26.73	21.56
pakdd	**17.32**	0.29	1.29	0.52	2.77	11.06	15.88	16.52	15.08	10.38
poker	36.84	37.95	43.85	50.82	**72.67**	63.49	30.14	50.72	54.39	17.05
spam	**58.74**	58.33	49.72	54.56	51.84	56.45	58.50	56.68	49.81	45.43
tripadvisor	**24.97**	6.15	2.55	7.51	6.51	18.12	23.67	24.12	23.53	22.25
twitter	73.73	75.21	50.07	70.89	50.43	72.39	71.53	**75.24**	60.97	58.81
weather	**49.81**	46.77	34.79	44.00	41.19	48.32	45.29	43.08	40.32	36.19
Avg. G-Mean	**61.41**	44.01	38.09	43.10	44.04	54.11	53.38	58.80	54.65	57.68
Avg. Kappa	37.76	35.53	27.72	34.08	33.73	**39.60**	35.36	39.27	36.15	28.27
Rank G-Mean	**1.50**	6.95	9.39	7.61	7.61	4.37	5.13	3.24	4.74	4.47
Rank Kappa	4.08	4.84	8.87	6.03	6.34	3.11	5.39	**3.03**	5.63	7.68

Bold font highlights the best result

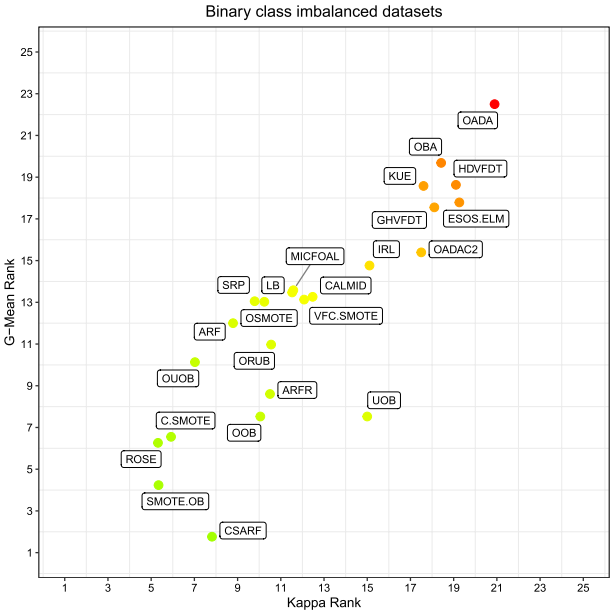

Fig. 28 Comparison of all 24 algorithms for binary class imbalanced datasets. Color gradient represents the product of both metrics (Color figure online)

windows in which minority and majority instances appear. In real-world datasets, this does not happen, since they were collected to model specific phenomenon observations and does not respect such clear probabilistic mechanisms. All of this poses unique challenges to classifiers, such as the latency with which instances from a specific class arrive, or long periods when instances from only one class appear. This configuration of data streams presents many more challenges for streaming classifiers. Such benchmarks allow us to gain insights about the classifiers examining them under unique and challenging conditions.

Discussion.

Impact of approach to class imbalance. First, it is interesting to note than on average all examined methods displayed much better Kappa than G-mean performance. We can observe that ensembles utilizing blind resampling, such as OOB and UOB, returned poor performance over real-world data streams. We can explain this by their purely online nature paired up with catastrophic forgetting, as these ensembles adapt their resampling strategy to the newest arriving instances, and thus are not being able to retain any memory of previously seen concepts. As in real-world scenarios instances do not arrive in stratified windows, one of classes may disappear for a while. This confuses such ensembles and leads to high skewness towards one class that is very difficult to overcome via online blind resampling. The methods based on informed resampling, such as C-SMOTE and SMOTE-OB displayed satisfactory results, showing that their learning mechanisms are robust to various characteristics of real-world streams. Also, SMOTE-OB achieved balanced results regarding both metrics, demonstrating a high stability and reliability in real-world cases. Interestingly, C-SMOTE was underperforming in previous synthetic cases, showing discrepancies between artificial and real-world domains. This allows us to conclude that there is a need for further research in real-world imbalanced streams and capturing more realistic benchmarks that reflect various learning difficulties.

When analyzing algorithm-level modification classifiers, ROSE displayed the best results, especially for the Kappa metric. While for synthetic datasets ROSE was

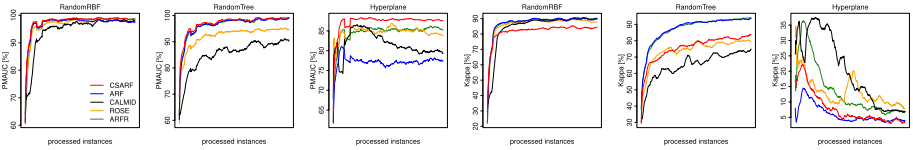

Fig. 29 PMAUC and Kappa on multi-class static imbalance ratio

consistently among the best methods, for real-world cases we can see that its robustness to a variety of learning difficulties allowed it to demonstrate its potential. ROSE stores buffers for each class independently contributing to scenarios with high latency of instances. CSARF remained as one of the best-performing classifier, displaying the best results on G-Mean, and being among the best regarding Kappa.

The worst-performing algorithms in the real-world scenario differ from what we saw in previous scenarios (with exception of OADA still being the weakest classifier). Algorithms that achieved average to good performance in other experiments such as KUE, HDVFDT and OBA did not maintain their performance over real-world datasets.

It is interesting to see that ARF which was not among the best-performing classifiers in the experiment with synthetic data, was the third-best classifier regarding Kappa. This shows that in the used real-world datasets the impact of concept drift was much more significant than the impact of class imbalance, allowing for a method focusing purely on adaptation to changes to rank so high.

Impact of ensemble architecture. Real-world datasets allow us to evaluate how each type of ensemble architecture deals with streams under multiple difficulties appearing at the same time. We can see that all ensemble-based methods display much better performance on average than in the previous experiments. This is especially true of boosting-based methods that reduced the gap in their performance when compared to top-performing algorithms. However, bagging-based and hybrid ensembles still are the superior choices. This shows how these architectures offer better robustness in scenarios where data does not follow uniform characteristics over extended periods.

7.2 Multi-class experiments

The second set of experiments focuses on multi-class problems where the relationships among the many classes may vary over time (Lango & Stefanowski, 2022). Multi-class imbalanced data is more difficult and less frequently studied than its binary counterpart. There are relative imbalance ratios among classes and overlapping of the minority and majority classes becomes a greater issue (Santos et al., 2023; Stefanowski, 2021; Lipska & Stefanowski, 2022). These experiments include static imbalance ratio, dynamic imbalance ratio, concept drift and static imbalance ratio, concept drift and dynamic imbalance ratio, analysis on the impact of the number of classes, real-world multi-class datasets, and semi-synthetic multi-class imbalanced datasets. The number of examined algorithms in this set of experiments is reduced to 15 following their multi-class capabilities shown in Table 2.

7.2.1 Static imbalance ratio

Goal of the experiment. This experiment was designed to address **RQ1** and to evaluate the performance and robustness of the classifiers to the static class imbalance in a scenario

Table 13 PMAUC and Kappa on multi-class static imbalance ratio

Generator	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC										
Hyperplane	**87.48**	77.30	82.75	76.81	76.24	81.53	75.59	85.21	84.91	87.83
RandomRBF	**97.97**	97.32	95.48	96.36	97.43	95.17	96.59	96.96	97.81	95.28
RandomTree	**97.11**	96.57	93.31	84.52	94.99	85.69	91.07	92.57	96.65	92.14
Kappa										
Hyperplane	7.63	6.13	**21.63**	7.10	2.06	20.15	6.37	15.69	13.86	21.39
RandomRBF	82.59	**88.27**	82.82	86.04	87.63	85.57	87.73	86.97	88.11	79.89
RandomTree	76.98	**88.55**	74.99	57.06	71.89	65.27	80.32	75.26	88.43	68.63
Avg. PMAUC	**94.19**	90.40	90.51	85.89	89.55	87.46	87.75	91.58	93.13	91.75
Avg. Kappa	55.73	60.98	59.81	50.06	53.86	57.00	58.14	59.30	**63.47**	56.64
Rank PMAUC	**1.33**	4.67	6.00	8.33	5.33	8.33	8.00	4.67	2.67	5.67
Rank Kappa	6.33	3.67	5.00	7.67	7.00	6.33	4.67	4.67	**3.00**	6.67

Bold font highlights the best result

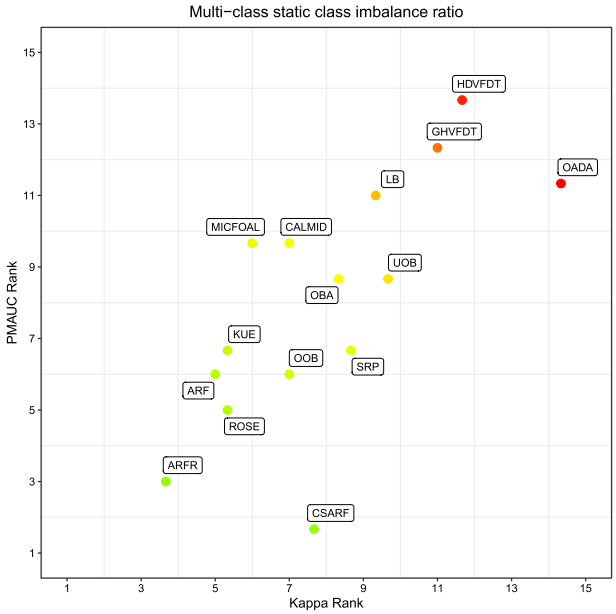

Fig. 30 Comparison of all 15 algorithms for multi-class static class imbalance ratio. Color gradient represents the product of both metrics (Color figure online)

with multiple classes. In multi-class settings, the class imbalance can be even more challenging than in binary settings, since now multiple classes can be underrepresented. Also, relations among the classes are no longer obvious, since one class may be a majority when compared to some other classes, but a minority for the rest of them. This allows us to analyze how each classifier behaves under specific class distributions. To evaluate this, we prepared three multi-class generators {Hyperplane, RandomRBF, and RandomTree}, all of them with 5 classes using the class distribution {50, 20, 10, 5, 1}. Figure 29 illustrates the performance of the five selected algorithms classifiers for each multi-class stream.

Table 13 summarizes the performance of the top 10 classifiers for each generator and their average ranking regarding each metric. For overall comparison, Fig. 30 presents the overall aggregated performance of all classifiers. Axes of the ellipse represent PMAUC and Kappa metrics, the more rounded the better, and the color represents the product of both metrics.

Discussion

Impact of class imbalance approach. First, we need to observe that the performance of the algorithms in multi-class problems significantly differs from the binary problems. This shows that multi-class imbalance data streams pose a series of unique challenges and thus this requires developing specific mechanisms dedicated to tackling more than two classes. Simple adaptation of binary mechanisms tends to fail and underperform, especially when dealing with a large number of classes.

For blind resampling methods, we can see a drop in their performance. OOB returns mediocre results, much below the ranks observed in binary scenarios. UOB becomes completely unusable in multi-class problems, failing to achieve any acceptable predictive power. This shows that when dealing with multiple distributions, blind resampling methods cannot capture complex relationships among classes and tend to further increase the difficulty factors (such as class overlapping or noise). This happens because blind resampling approaches consider only a single class, thus discarding valuable information about other classes. There is a need to develop novel resampling algorithms dedicated specifically to multi-class data streams.

ARFR was the best algorithm regarding the Kappa metric and the second-best regarding PMAUC. Its weighing mechanism led to good robustness to multi-class imbalance ratio, as it assigns importance to every tree in the ensemble based on the class distribution (independently of the number of classes. The cost-sensitive CSARF displays the best performance on the G-Mean metric, yet suffers under Kappa evaluation. This shows that CSARF focuses on the minority classes but at the cost of suffering a larger number of false positives. The best three classifiers are based on ARF, showing that it is very reliable in multi-class scenarios. Among classifiers based on training modifications, only ROSE achieved good results, demonstrating that keeping buffers for each class is a good choice for this scenario. On the other hand, HDVFDT and GHVFDT were among the worst. It is worth mentioning that CALMID and MICFOAL were not able to outperform the mentioned classifiers, despite being specifically designed for multi-class imbalanced data streams.

Impact of ensemble architecture. Ensembles once again are predominant among the best performing methods for multi-class imbalanced streams. Within bagging-based methods, only LB underperformed. However, LB is a general-purpose ensemble, therefore it was expected not to display robustness on pair with dedicated skew-insensitive solutions. KUE and SRP could satisfactory handle static multi-class imbalance. Also, it is interesting to note that most bagging methods displayed balanced performance considering Kappa and PMAUC, demonstrating that their natural partitioning of instances contributes to a balanced performance among all classes. We have much less information on boosting-based ensembles, as only one of the examined classifiers were suitable for multi-class problems. However, this single case performed poorly, allowing us to assume that the performance of boosting-based methods will follow trends from binary scenarios.

7.2.2 Dynamic imbalance ratio

Goal of the experiment. This experiment was designed to complement the previous experiment and address **RQ2** to evaluate the robustness of the classifiers to dynamic

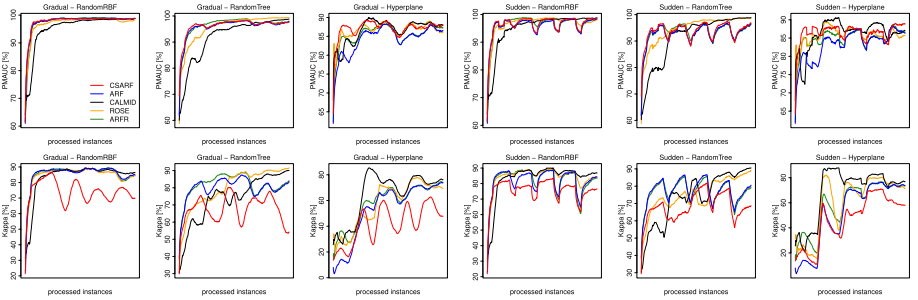

Fig. 31 PMAUC and Kappa on multi-class shifting imbalance ratio

Table 14 PMAUC and Kappa on multi-class shifting imbalance ratio

Drift	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC										
Sudden	92.32	90.95	**92.58**	92.29	90.44	91.38	89.93	91.65	91.35	92.49
Gradual	**93.85**	92.87	93.04	92.79	91.88	91.86	90.94	93.56	93.26	93.04
Kappa										
Sudden	63.49	70.99	72.60	73.04	65.73	**76.87**	70.13	72.24	71.91	68.33
Gradual	63.81	73.42	72.42	71.55	68.26	74.36	70.61	**75.11**	74.97	70.47
Avg. PMAUC	**93.09**	91.91	92.81	92.54	91.16	91.62	90.43	92.61	92.31	92.76
Avg. Kappa	63.65	72.20	72.51	72.29	66.99	**75.62**	70.37	73.68	73.44	69.40
Rank PMAUC	**2.58**	5.58	4.42	4.50	7.42	7.50	7.92	4.83	4.92	5.33
Rank Kappa	9.50	4.50	5.00	4.67	8.00	**3.08**	5.58	4.00	3.92	6.75

Bold font highlights the best result

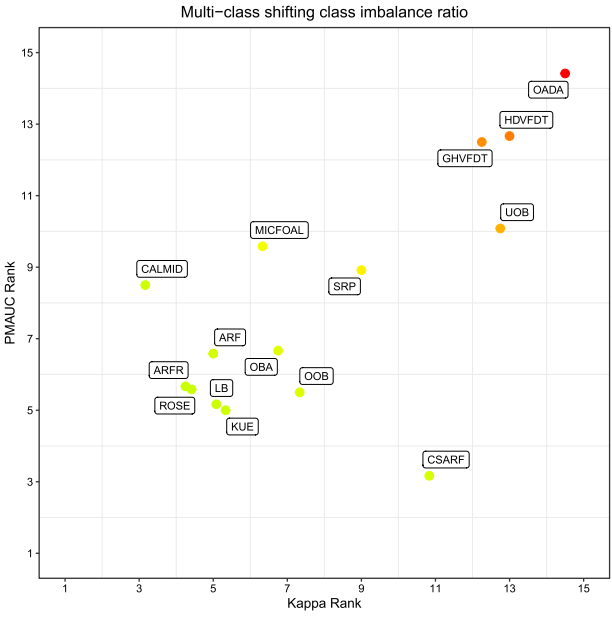

Fig. 32 Comparison of all 15 algorithms for multi-class shifting class imbalance ratio. Color gradient represents the product of both metrics (Color figure online)

changes in imbalance ratio with multiple classes. To evaluate this, we prepared three multi-class generators {Hyperplane, RandomRBF and RandomTree}, all of them with 5 classes shifting imbalance ratio through the following distributions: {{50, 20, 10, 5, 1}, {20, 10, 5, 1, 50}, {10, 5, 1, 50, 20}, {5, 1, 50, 20, 10}, {1, 50, 20, 10, 5}}. The speed of the changes was evaluated both sudden and gradual. This allows us to analyze how classifiers are able to cope with dynamic imbalance ratio changes and how they are able to adapt. Figure 31 illustrates the prequential PMAUC and Kappa for each generator over time for the selected classifiers. Table 14 presents the performance for the top 10 classifiers, and their average ranking. To summarize, Fig. 32 shows the overall performance of all classifiers

Discussion

Impact of class imbalance approach. Interestingly, the average performance of all evaluated classifiers is higher under the dynamic imbalance than under the static skewness ratio. We can explain that by the fact that evolving imbalance and class roles lead to each of the classes being the majority class for a given period of time, thus allowing for a better exposure of it to the classier, as well as countering the small sample size problem (which is a big challenge for multi-class imbalance data).

Blind resampling methods repeated the trends observed in the previous experiment, with OOB returning acceptable performance and UOB failing to deliver predictive power. Undersampling, by reducing the size of majority class, can be enhancing the small sample size difficulty, instead of temporarily alleviating it. This prevents UOB from capitalizing on stats of the stream when a minority class transforms to majority one.

Ensembles based on ARF maintain their very good performance and robustness to evolving imbalance ratio. ARFR displayed one of the best performances, showing that adding a level of resampling really enhanced the robustness to drifting class imbalance. CSARF exhibited great performance on the PMAUC metric, but again failed to return satisfactory Kappa. Algorithms based on training modifications like ROSE and CALMID showed better robustness and ability to handle drifting imbalance ratios. CALMID exceeds on Kappa metric, but drops several ranks under PMAUC. ROSE presented the most balanced results in this scenario regarding both metrics, therefore it can be seen as the most reliable and trustworthy choice for a multi-class imbalanced data stream.

Impact of ensemble architecture. As we saw in previous experiments, bagging methods are among the best-performing, and it can be seen easily on the overall figure (Fig. 32), where 7 classifiers form a cluster in the bottom-left side of the distribution, all of them being bagging-based ensembles. Hybrid architectures presented by CALMID and MICFOAL that were designed specifically for multi-class imbalanced streams significantly improve their performance when dealing with evolving imbalance ratios.

Impact of drift speed in class imbalance. Considering the speed of changes in class imbalance ratios, we can notice that the impact of speed is marginal on most of the classifiers. Some methods, mainly the ones without any drift handling mechanisms, have slower responses, but it did not translate to significant changes over predictive performance. PMAUC metric seems to be more sensitive to differentiation between gradual and sudden changes, while Kappa values are similar for both speeds. We can explain that by the fact that PMAUC does not consider the class imbalance ratios, thus responding differently to varying speed of changes. Kappa offers a more stable monitoring of the stream changes, not affected by the velocity of imbalance ratio evolution.

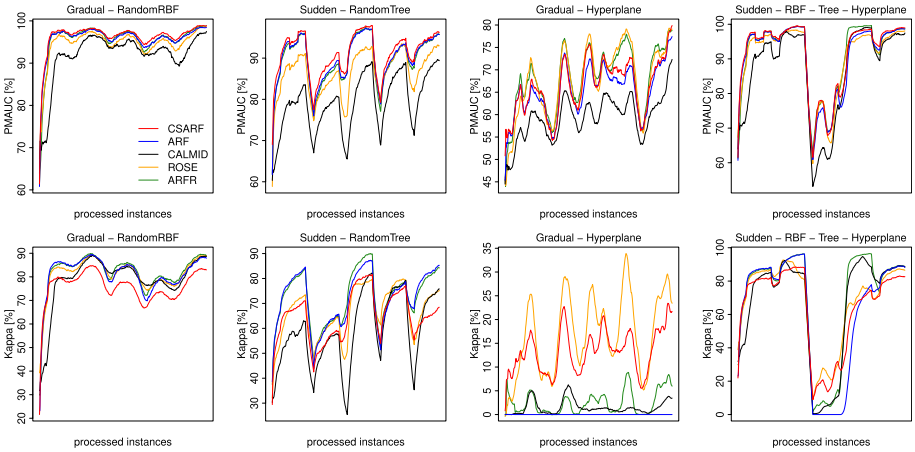

Fig. 33 PMAUC and Kappa on concept drift and multi-class static imbalance ratio

Table 15 PMAUC and Kappa on concept drift and multi-class static imbalance ratio

Drift	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC										
Sudden	**87.02**	86.26	77.49	83.35	84.82	80.19	81.02	85.37	86.94	77.42
Gradual	84.02	83.29	76.84	81.12	82.42	78.06	79.00	83.41	**84.22**	77.01
Kappa										
Sudden	56.12	54.82	46.27	52.28	49.92	52.30	49.49	**59.32**	58.12	41.93
Gradual	49.05	46.12	42.37	47.00	42.97	46.65	43.16	**55.05**	50.32	39.94
Avg. PMAUC	85.52	84.77	77.16	82.24	83.62	79.12	80.01	84.39	**85.58**	77.22
Avg. Kappa	52.58	50.47	44.32	49.64	46.45	49.48	46.32	**57.18**	54.22	40.93
Rank PMAUC	**1.75**	3.25	8.75	6.00	4.25	8.75	7.50	3.88	2.00	8.88
Rank Kappa	4.75	5.00	7.75	5.38	7.25	5.25	7.00	**2.13**	2.25	8.25

Bold font highlights the best result

7.2.3 Concept drift and static imbalance ratio

Goal of the experiment. This experiment was designed to complement previous experiments and address **RQ2** and **RQ4** and to evaluate the behavior of the classifiers in a scenario with multiple classes in the presence of concept drift and static imbalance ratio. Concept drift leads to changes in decision boundaries, creating a challenge for classifiers to cope with and react to change. To evaluate this, we prepared three streams generators similarly to experiment Sect. 7.2.1, plus a concatenation of all three streams, and introduced concept drifts along the stream gradually or suddenly. Figure 33 presents the performance of the five selected classifiers for each evaluated drifting stream. Table 15 provides the PMAUC and Kappa for the top 10 classifiers for both types of drift and their average value and ranking. Figure 34 illustrates the overall performance for all classifiers.

Discussion

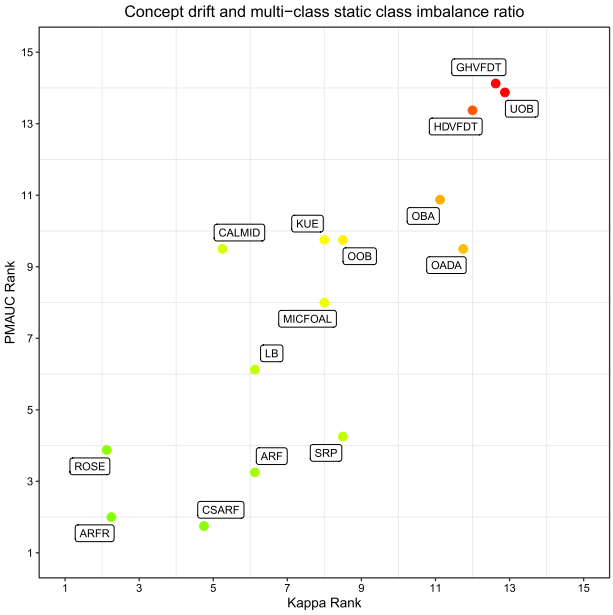

Fig. 34 Comparison of all 15 algorithms on concept drift and multi-class static class imbalance ratio. Color gradient represents the product of both metrics

Impact of class imbalance approach. Concept drift poses an increased difficulty in multi-class scenarios, as it changes complex relationships among classes. Multi-class problems tend to have much more complex decision boundaries than their binary counterparts and thus adaptation to drift requires more training instances or increased amount of time.

When analyzing resampling-based approaches, we can observe a significant drop in predictive power for both OOB and UOB. We already established that UOB is incapable of handling multi-class problems, but the additional presence of concept drift positions it among the worst performing classifiers. This follows our observations from the binary experiments, where we showed that lack of explicit or implicit drift adaptation mechanisms in OOB and UOB inhibits their learning capabilities from non-stationary data. ARFR returned best results among resampling-based algorithms, being at the same time competitive with other top performing classifiers.

The basic version of ARF displayed a loss of performance, showing that this algorithm cannot handle well changes appearing in multiple classes at once, especially when these classes are skewed. Its cost-sensitive modification maintained the very good performance observed in previous experiments, additionally improving under Kappa metric. This shows that CSARF is capable of an efficient adaptation to concept drift. CALMID and MICFOAL displayed good results, being methods natively designed for multi-class scenarios. ROSE was among the best performing algorithms, without relying on resampling or cost-sensitive modifications. This shows that ROSE mechanisms, mainly effective classifier replacement and class-based buffers, allow for an improved robustness in drifting and imbalanced multi-class scenarios.

Impact of ensemble architecture. Once again bagging-based and hybrid architectures tend to dominate the experimental study. Even methods such as LB and SRP returned decent results, despite their lack of skew-insensitive mechanisms. This shows that well-designed drift adaptation goes a long way in every streaming scenario and that bagging-based architectures can utilize their diversity to better anticipate the drift occurrence. Two

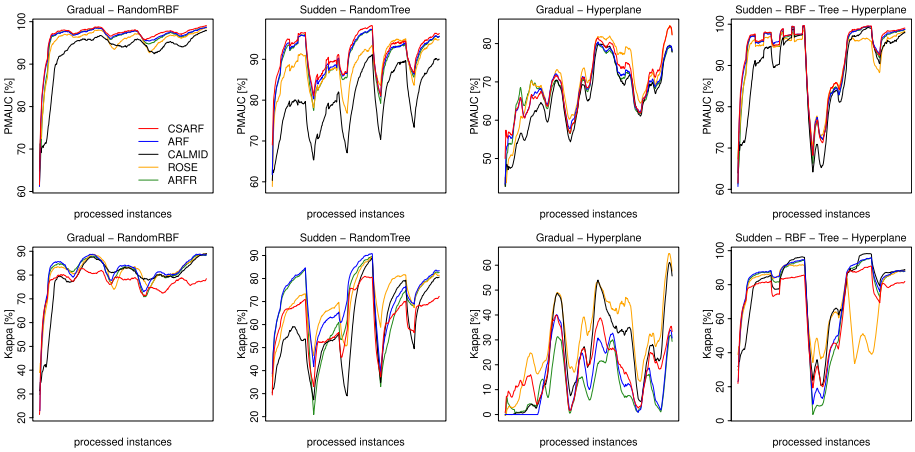

Fig. 35 PMAUC and Kappa on concept drift and multi-class shifting imbalance ratio

Table 16 PMAUC and Kappa on concept drift and multi-class shifting imbalance ratio

Drift	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC										
Sudden	82.02	81.97	77.82	78.66	**82.40**	77.60	78.75	81.80	81.80	74.14
Gradual	80.83	81.54	80.04	79.14	**82.05**	78.25	79.37	81.93	81.43	76.56
Kappa										
Sudden	48.85	54.31	48.93	50.05	51.95	50.46	51.54	**55.57**	52.81	37.27
Gradual	43.35	49.18	46.40	46.91	49.01	45.83	47.19	**51.68**	48.26	37.07
Avg. PMAUC	81.43	81.75	78.93	78.90	**82.23**	77.92	79.06	81.87	81.61	75.35
Avg. Kappa	46.10	51.75	47.66	48.48	50.48	48.15	49.37	**53.62**	50.54	37.17
Rank PMAUC	3.55	3.40	7.13	6.28	**2.93**	7.75	6.93	3.90	3.88	9.28
Rank Kappa	7.30	3.70	6.45	5.18	4.15	5.30	5.70	**2.93**	4.88	9.43

Bold font highlights the best result

exceptions to this rule are KUE, which as we observed in binary class cannot perform well with concept drift and imbalance, and UOB that does not adapt well to multi-class imbalance.

Impact of concept drift speed. We can see that the speed of concept drift does not significantly affect the results of individual classifiers. However, we can see different behavior of the metrics as compared to the previous experiment. Here Kappa reacts differently to gradual and sudden drifts, showing that the speed of evolution of class boundaries can be picked up by Kappa analysis. This allows us to conclude that when concept drift is combined with imbalance, both PMAUC and Kappa become sensitive to speed of changes.

7.2.4 Concept drift and dynamic imbalance ratio

Goal of the experiment. This experiment was designed to complement previous experiments and address **RQ4** to evaluate the behavior of the classifiers in a scenario with

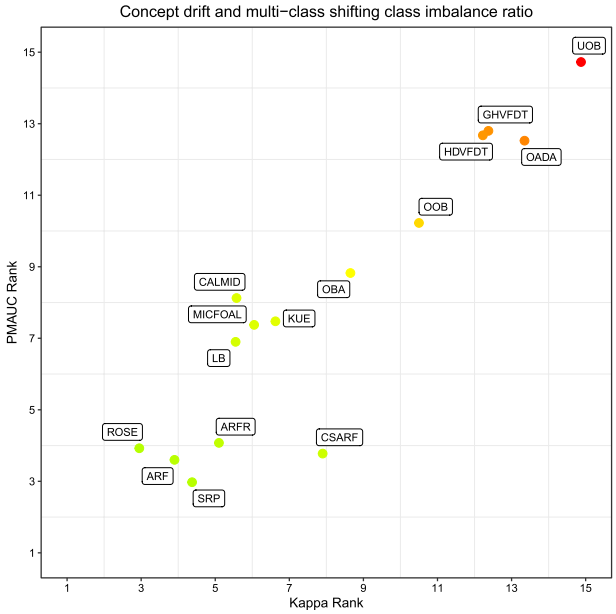

Fig. 36 Comparison of all 15 algorithms on concept drift and multi-class shifting class imbalance ratio. Color gradient represents the product of both metrics (Color figure online)

multiple classes in the presence of concept drift and dynamic imbalance ratio. Besides concept drift, changes in the imbalance ratio poses obstacles for classifiers that have to deal with multiple changes in data distribution. To evaluate this, we prepared three streams generators similarly to experiment Sect. 7.2.2, but introducing concept drifts along the stream gradually and suddenly. Figure 35 illustrates the PMAUC and Kappa metrics of the selected classifiers for each evaluated drifting stream. Table 16 presents the PMAUC and Kappa for the top 10 classifiers for both types of drift and their average value and ranking. Figure 36 provides the overall performance for all classifiers.

Discussion

Impact of class imbalance approach. Regarding blind resampling methods, we can see that the combination of concept drift and evolving imbalance ratios led to significant deterioration of OOB results, showing that the blind oversampling cannot adapt well to changes happening in both feature space and class characteristics. UOB was impacted even more significantly, making it the worst classifier in this scenario. ARFR is still among the best performing methods, however we can see small drop in the performance compared to the previous experiment. This shows that informed resampling techniques still require more work regarding the adaptation to both drifting and evolving imbalance ratios, as especially class role switching became challenging for ARFR.

When analyzing algorithm-level solutions we can see that CSARF, while still performing well on PMAUC, displayed reduced performance on Kappa. This shows that it cannot handle evolving imbalance ratios and class roles well, having high bias towards the initial role of classes. CALMID and MICFOAL improved their relative ranking regarding previous experiments, showing that they are resilient enough to handle both challenges at the same time.

ROSE is a clear winner in this scenario, showing the best robustness to multiple types of changes affecting the data stream. Its adaptation and skew-insensitive

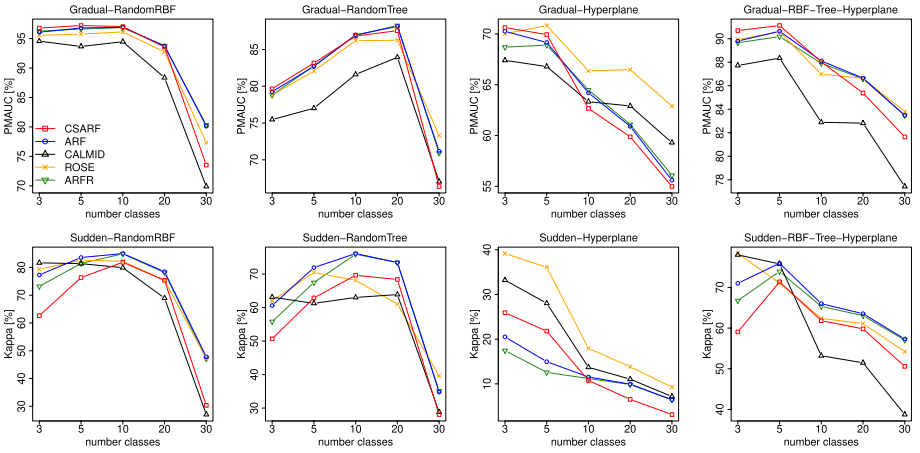

Fig. 37 Impact of the number of classes on PMAUC and Kappa under concept drift and multi-class shifting imbalance ratio

mechanisms allow it to efficiently handle the combination of concept drift and dynamic class imbalance, easily adapting to the new incoming concepts, even with changed class roles.

Impact of ensemble architecture. Once again we can see a clear dominance of bagging-based and hybrid architectures. However, this difficult learning scenario gives us a very unexpected insight. We can see that SRP and LB are able to outperform CALMID and MICFOAL. This is highly surprising, as the former methods are general-purpose classifiers, while the latter ones were specifically designed to handle imbalanced multi-class streams. Additionally, KUE achieved similar performance to dedicated skew-insensitive ensembles. This allows us to conclude that combination of bagging-based or hybrid architecture with an effective drift adaptation mechanism is a leading factor in the performance of ensemble classifiers for drifting and dynamically imbalanced streams. Therefore, it is crucial for future researchers not to focus solely on how to handle class imbalance, but firstly how to handle non-stationary characteristics, and then make this adaptation mechanism skew-insensitive.

Impact of concept drift speed. Once again, we are unable to see a clear relationship between the speed of concept drift and classifier performance. Even under sudden drifts, most of the examined methods were able to quickly recover and return to their performance before the change. Therefore, end results are similar for any speed of change. The differences can be observed very locally during the drift occurrence, but they did not have a long-lasting effect on any classifier.

Relationship between concept drift and shifting imbalance ratio. This scenario combines two types of changes, creating a more realistic and challenging scenario. Therefore, we need to understand the impact of each of these types of changes on the underlying classifier. Analyzing the results, we can see that most of existing algorithms are characterized by a trade-off: either focusing on adaptation to changes or on robustness to class imbalance. Only ROSE and SRP displayed a balanced performance on both tasks. This supports our previous conclusion that there is a need to design novel methods where both adaptation and skew-insensitiveness will be solved as a joint problem.

Table 17 PMAUC and Kappa averages on the number of classes under concept drift and multi-class shifting imbalance ratio

Classes	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC										
3	**85.81**	84.88	80.07	83.19	84.06	81.26	81.62	83.99	84.30	78.45
5	**86.65**	86.10	81.27	84.52	86.08	81.89	82.68	85.74	85.86	78.57
10	84.49	84.69	80.84	83.10	**84.98**	80.74	81.18	84.37	84.68	77.33
20	81.48	81.95	80.27	79.91	**82.92**	78.89	79.11	82.32	81.99	75.96
30	68.71	71.13	72.19	63.77	**73.09**	66.84	70.71	72.93	71.23	66.45
Kappa										
3	48.01	55.72	54.00	60.04	44.70	60.92	54.10	**62.20**	52.48	43.66
5	55.22	58.39	53.72	58.47	54.46	58.83	54.81	**63.08**	56.11	44.43
10	53.11	56.71	49.11	53.30	**57.07**	50.48	51.77	55.88	56.36	38.51
20	49.52	53.83	47.57	48.28	**56.11**	46.62	49.06	51.76	53.66	36.79
30	24.65	34.08	33.92	22.33	**40.06**	23.88	37.08	35.19	34.06	22.43
Avg. PMAUC	81.43	81.75	78.93	78.90	**82.23**	77.92	79.06	81.87	81.61	75.35
Avg. Kappa	46.10	51.75	47.66	48.48	50.48	48.15	49.37	**53.62**	50.54	37.17
Rank PMAUC	3.55	3.40	7.13	6.28	**2.93**	7.75	6.93	3.90	3.88	9.28
Rank Kappa	7.30	3.70	6.45	5.18	4.15	5.30	5.70	**2.93**	4.88	9.43

Bold font highlights the best result

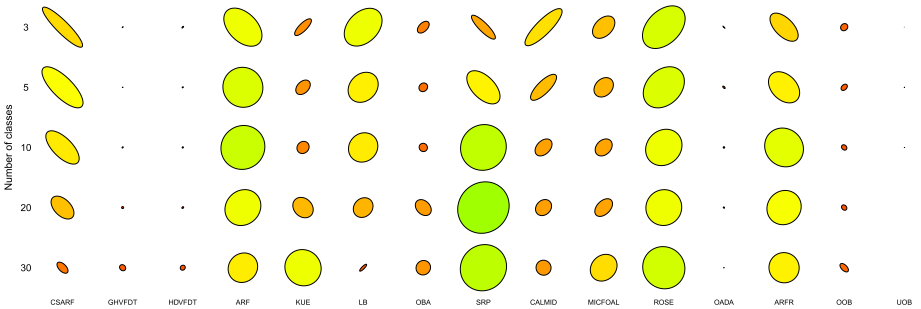

Fig. 38 Comparison of all 15 algorithms on the number of classes under concept drift and multi-class shifting imbalance ratio. Axes of the ellipse represent PMAUC and Kappa metrics. Color gradient represents the product of both metrics (Color figure online)

7.2.5 Impact of the number of classes

Goal of the experiment. This experiment was designed to evaluate the robustness of the classifiers to different number of classes under the presence of concept drift and dynamic imbalance ratio. Combining those learning difficulties with different number of classes allow us to evaluate how classifiers deal with higher number of classes and examine if does affect their learning mechanisms or not. To evaluate this, we used the generators in experiment Sect. 7.2.2. All these generators were evaluated with the following number of classes {3, 5, 10, 20, 30}. Figure 37 illustrates the performance of

five selected algorithms classifier for each number of classes. Table 17 summarizes the performance of the top 10 classifiers for each number of classes and their average ranking regarding each metric. For overall comparison, Fig. 38 presents the overall aggregated performance of all classifiers. Axes of the ellipse represent PMAUC and Kappa metrics, the more rounded the better, and the color represents the product of both metrics.

Discussion

Impact of class imbalance approach. We can see that high number of classes pose a significant challenge for most of the examined methods. For resampling-based approaches, we observe that OOB and UOB cannot handle any higher number of classes, returning the worst performance of the same rank as single tree classifiers (GHVFDT and HDVFDT). ARFR maintains its performance with the increase in the number of classes, showing that the combination of informed resampling with ARF-based architecture allows for the memorization of more complex decision boundaries, while combating bias using well-placed artificial instances.

When looking at the algorithm-level modifications we can see that CSARF, previously one of the best algorithms, displays no robustness to increasing number of classes. This shows the limitations of cost-sensitive approaches, as with the increased number of classes the cost matrix needs to grow. Large cost matrices lead to loss of meaning behind the penalties and their reduced influence on learning process and no effect on bias towards majority classes. We can conclude that existing cost-sensitive methods are not suitable for handling multi-class imbalanced streams with a high number of classes. CALMID, despite being designed for multi-class problems, cannot handle increasing number of classes and returns performance similar to ensembles based on blind resampling. ROSE and MICFOAL displayed the best robustness to high number of classes. ROSE, especially for the Kappa metric, is a safe choice for scenarios with elevated number of classes.

Impact of ensemble architecture. Our analysis of the ensembles under increasing number of classes showed once again the dominance of bagging-based and hybrid architectures. In this scenario, hybrid approaches became dominant, with ROSE displaying best results due to its combination of working on both instance and feature subspaces, combined with per-class memory buffers for balanced class representations.

Impact of the high number of classes. With the increasing number of classes, we can see a clear break point when the number of classes is > 20. This shows that all classifiers could handle the increasing number of classes up to a certain point, after which their capabilities for memorizing new concepts and generalizing over all classes begin to rapidly deteriorate. We can see that for scenarios with 30 classes most of the methods start returning highly unsatisfactory results. Interestingly, for these cases we can observe a very good performance of standard classifiers, such as SRP. When analyzing ranks, we can see that SRP and ROSE are two best performing ensembles when handling high number of classes. While we provided the explanation for the better performance of ROSE, it is very surprising to see that SRP performs on par with it. We can explain it by the fact that both ROSE and SRP use feature subspaces, which can be seen as lower dimensional projections of a difficult learning task. In such a lower dimensional subspaces the decision boundaries among classes may be simplified, leading to better generalization capabilities. This follows observation made in Korycki and Krawczyk (2021b), where it was postulated that low-dimensional representations can overcome class imbalance without any dedicated skew-insensitive mechanisms.

Table 18 Real-world multi-class datasets specifications

Dataset	Instances	Features	Classes
activity	5418	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	7200	21	3
zoo	1,000,000	17	7

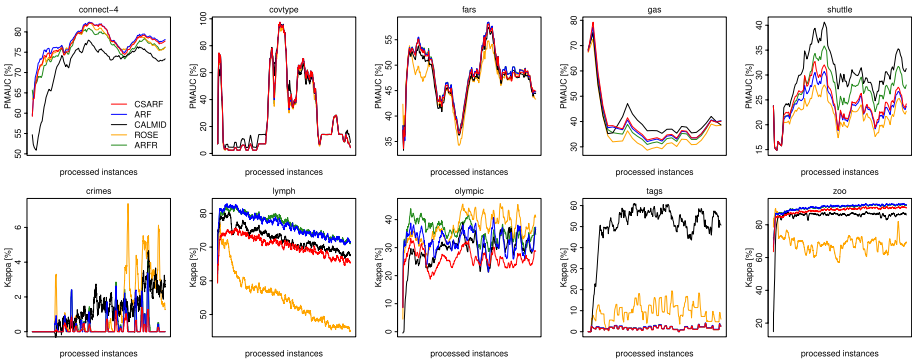

Fig. 39 PMAUC and Kappa on multi-class imbalanced datasets

7.2.6 Real-world multi-class imbalanced datasets

Goal of the experiment. This experiment was designed to address **RQ5** and to evaluate the performance of the classifiers on 18 multi-class real-world imbalanced and drifting data streams. The previous experiments focused on analyzing how the classifiers can deal with multiple learning difficulties in multi-class data streams. This allowed us to examine their behavior in very specific and controlled scenarios. Furthermore, with data stream generators we have full control over the created data, but we cannot generate specific scenarios that are present in real-world scenarios, because they are characterized by merging various learning difficulties at varying frequency and intensity. The real-world data streams

Table 19 PMAUC and Kappa on multi-class imbalanced datasets

Dataset	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC										
activity	71.85	72.06	58.74	71.33	71.42	61.05	60.31	68.04	71.98	**73.01**
connect-4	77.71	78.10	70.72	76.26	**80.01**	72.79	74.17	77.32	76.60	76.06
cov-pok-elec	6.91	6.91	7.12	6.91	6.91	**9.16**	6.93	7.27	6.91	7.28
covtype	29.92	29.82	28.11	29.13	29.75	**31.04**	29.97	28.84	29.80	27.00
crimes	48.10	48.37	47.07	47.69	**48.61**	47.29	47.56	46.18	48.47	44.60
fars	**45.66**	35.95	44.59	36.31	37.15	42.39	41.07	38.99	41.10	45.45
gas	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
hypothyroid	**0.14**	**0.14**	0.13	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	0.13
kddcup	**98.12**	96.77	96.58	95.91	97.47	95.75	96.92	94.28	97.06	97.39
kr-vs-k	6.41	6.40	6.66	6.47	6.38	**9.90**	6.41	7.07	6.41	6.79
lymph	24.29	23.50	26.66	20.97	27.13	**31.11**	24.01	22.22	27.34	28.13
olympic	98.64	98.36	92.39	97.38	**98.74**	95.80	97.82	97.75	98.23	97.53
poker	98.28	98.21	98.32	95.06	**99.12**	94.87	97.58	94.30	98.15	97.95
sensor	23.64	23.62	24.42	24.81	25.87	23.31	23.68	24.57	23.63	**26.42**
shuttle	39.36	39.39	32.10	**41.88**	41.37	39.57	40.28	36.67	38.69	36.45
tags	**78.27**	76.70	71.59	74.49	73.54	64.39	67.68	74.28	75.75	66.15
thyroid	34.04	34.04	33.97	34.04	**35.25**	34.03	34.03	33.99	34.04	33.64
zoo	4.40	4.39	5.90	5.48	4.67	**6.53**	6.14	4.82	4.39	6.29
Kappa										
activity	59.43	59.62	29.23	51.77	57.96	36.30	34.36	43.21	**61.37**	53.05
connect-4	36.88	32.23	23.75	33.63	33.53	34.79	30.81	**41.70**	35.53	37.42
cov-pok-elec	0.19	0.32	1.38	0.34	0.17	**34.38**	0.32	4.95	0.33	3.99
covtype	47.91	52.00	38.73	43.89	52.46	**78.25**	60.98	41.61	52.25	34.60
crimes	50.40	66.25	62.21	65.73	68.38	66.51	62.09	47.53	**70.31**	50.91
fars	11.53	10.59	8.24	10.26	7.61	13.98	13.80	14.78	0.84	**21.59**
gas	0.01	0.02	0.02	0.05	0.05	0.13	0.03	**0.14**	0.05	0.03
hypothyroid	1.42	1.24	−0.04	**1.45**	1.34	1.29	1.14	1.43	1.21	−0.49
kddcup	70.18	76.88	77.14	69.06	**82.22**	71.68	80.04	55.16	76.80	80.75
kr-vs-k	0.46	0.67	3.28	0.87	0.20	**51.27**	0.25	9.36	0.60	4.41
lymph	−4.15	1.63	22.86	0.33	34.71	**66.38**	17.65	0.96	13.83	18.56
olympic	73.92	87.08	75.49	**88.07**	87.81	76.72	72.91	86.93	78.47	83.83
poker	88.86	90.81	90.77	84.98	**93.65**	85.41	89.35	68.97	90.59	89.63
sensor	0.14	0.36	1.37	2.37	**4.40**	1.18	0.01	2.01	0.37	4.35
shuttle	72.19	71.84	30.42	68.24	**82.98**	54.16	79.39	47.16	64.71	39.02
tags	26.20	32.25	24.80	34.93	8.80	29.74	30.71	**37.29**	35.57	26.97
thyroid	3.08	3.21	2.05	**3.47**	0.00	3.38	3.21	2.84	3.20	0.37
zoo	1.79	1.94	24.51	18.20	7.27	**50.95**	38.25	10.03	1.89	28.44
Avg. PMAUC	**43.65**	42.93	41.39	42.46	43.53	42.17	41.93	42.04	43.26	42.79
Avg. Kappa	30.02	32.72	28.68	32.09	34.64	**42.03**	34.18	28.67	32.66	32.08
Rank PMAUC	**4.06**	5.22	6.89	6.11	4.17	5.56	6.11	6.17	5.11	5.61
Rank Kappa	7.00	5.39	6.83	5.06	4.94	**4.11**	6.00	5.17	5.22	5.28

Bold font highlights the best result

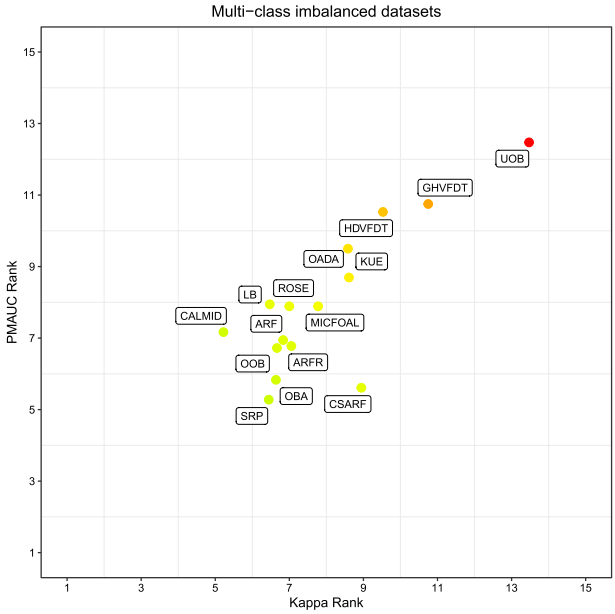

Fig. 40 Comparison of all 15 algorithms for multi-class imbalanced datasets. Color gradient represents the product of both metrics (Color figure online)

employed in the experiment are popular benchmarks for data streams classifiers, and their specifications are presented in Table 18. The PMAUC and Kappa for the five selected classifiers are presented in Fig. 39. Table 19 provides the average PMAUC and Kappa for the selected top 10 classifiers for each dataset. Figure 40 illustrates the overall performance of all classifiers for all real-world datasets.

Characteristics of real-world data streams. By analyzing the performance of classifiers in real-world datasets it is worth to bring up the difference between artificial streams and real-world imbalance data streams. In real-world datasets data was collected in order to model a specific phenomenon observations and does not hold clear probabilistic mechanisms such as stream generators. Also, in a multi-class real world scenario, relations between features and classes are not so clearly defined as it is on artificial generators. This benchmark allows us to gain insights about the classifiers examining them under real unique and challenging conditions.

Discussion

Impact of class imbalance approach. Similar to the previously analyzed binary case, real-world datasets bring a combination of various challenges in addition to the multi-class nature of analyzed streams. However, contrary to our observations from binary experiments, we cannot determine for any of the evaluated classifiers to be better than its peers. Also, it is possible to notice that on average PMAUC was very similar for all classifiers, while Kappa values tend to highlight more differences among algorithms. This shows that Kappa is an effective metric for multi-class imbalanced data streams, allowing us to gain more insight into how each of the algorithms is performing.

Analyzing the resampling-based approaches, we can see UOB returned unsatisfactory results, confirming our observations regarding its inability to cope with multiple classes. OOB returned much better predictive power, however only for datasets with relatively small number of classes. This confirms our previous observations that blind resampling methods are not suitable for problem characterized by a high number of classes to be

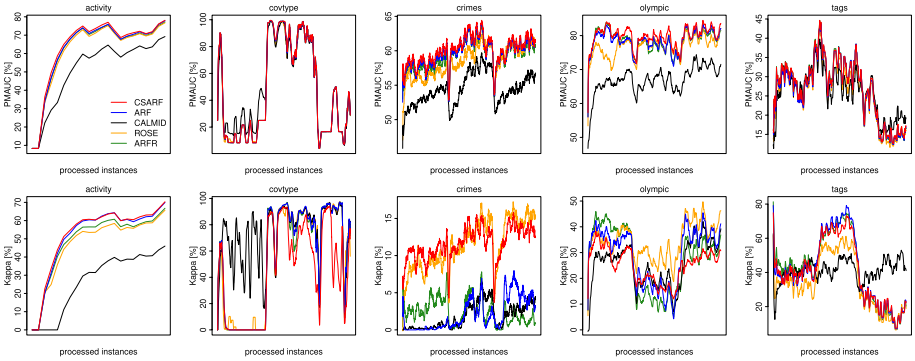

Fig. 41 PMAUC and Kappa on semi-synthetic multi-class imbalanced datasets

learned from. Interestingly, ARFR returned much better results, but on a similar level than standard ARF. This shows that major reason behind the success of ARFR lies not in the chosen informative resampling, but in a good selection of the ensemble architecture.

For algorithm-level approaches CSARF remained among the best-performing classifiers, displaying excellent PMAUC metric, but falling behind when it comes to Kappa evaluation. ROSE, CALMID and MICFOAL presented highly satisfactory results. It is worth to note that ROSE did not perform as well as it in previous scenarios, which can be explained by lack of specific learning difficulties in analyzed real world data streams (as ROSE excels in very difficult problems). CALMID and MICFOAL demonstrated better performance than on artificial domains, showing that their mechanisms lead to good performance over real-world problems.

Impact of ensemble architecture. While this experiment follows all our previous observations, we should focus on a comparison between general-purpose and skew-insensitive ensembles. Similarly to experiment with high number of classes, we can observe very good performance of general-purpose ensembles on real-world imbalanced benchmarks. CSARF displayed the best results in real-world datasets regarding PMAUC but the worst regarding Kappa. This shows that in the analyzed benchmarks adaptation to change and ability to better separate classes in lower-dimensional subspaces can return at least as good performance as dedicated mechanisms for tackling class imbalance.

7.2.7 Semi-synthetic multi-class imbalanced datasets

Goal of the experiment. This experiment was designed to address more in-depth **RQ5** and to evaluate the robustness of the classifiers to semi-synthetic data streams (Korycki & Krawczyk, 2020). We used all 9 multi-class semi-synthetic data streams proposed in.[4] These benchmarks simulate critical class ratio changes and concept drifts. This allows us to analyze how the classifiers are able to cope with dynamic changes and concept drifts with real-world data streams, how they are able to adapt to those changes. Figure 41 illustrates the performance of five selected algorithms in the semi-synthetic data streams. Table 20 presents the average PMAUC and Kappa for the top 10 classifiers for

[4] https://github.com/mlrep/imb-drift-20.

Table 20 PMAUC and Kappa on semi-synthetic multi-class imbalanced datasets

Dataset	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC										
activity	71.68	71.47	55.74	70.73	**71.78**	63.13	62.28	70.63	70.69	69.48
connect-4	77.56	77.95	66.96	76.00	**79.92**	72.19	72.84	77.22	76.25	72.89
covtype	46.79	46.65	45.64	46.14	46.99	**48.93**	46.86	46.53	46.51	48.70
crimes	60.08	59.70	55.82	55.83	**61.68**	54.37	56.79	58.54	59.25	54.96
gas	44.62	**45.47**	33.15	42.60	44.97	42.31	43.06	43.73	44.16	36.55
olympic	**80.05**	78.74	71.48	76.39	75.54	67.43	70.23	76.98	78.62	65.77
poker	28.53	27.69	28.85	27.65	28.51	29.55	28.90	26.74	26.13	**29.64**
sensor	**43.81**	43.68	42.03	43.64	41.72	43.38	43.63	43.11	43.69	43.50
tags	26.91	26.54	21.56	27.74	27.55	25.98	**28.29**	25.79	26.70	23.58
Kappa										
activity	64.64	63.53	19.13	55.72	**67.82**	41.22	45.33	57.74	60.99	45.24
connect-4	37.68	31.92	19.68	33.64	33.49	32.97	29.85	**41.45**	30.75	31.56
covtype	44.59	54.99	49.18	52.09	59.29	**73.40**	61.28	55.20	54.09	56.45
crimes	11.61	2.13	4.20	0.59	**14.33**	1.96	5.16	11.49	2.48	4.48
gas	68.84	**71.70**	24.94	53.97	70.92	49.34	60.89	59.49	58.22	29.68
olympic	25.24	29.52	22.16	28.15	10.04	26.33	29.49	**34.48**	27.28	14.25
poker	15.00	18.81	33.40	24.83	24.88	**48.52**	46.49	33.99	13.31	39.19
sensor	52.27	56.73	45.84	55.17	55.87	59.55	**60.62**	50.13	56.88	56.46
tags	39.63	40.73	9.18	38.09	45.27	41.28	**55.60**	33.52	40.63	11.03
Avg. PMAUC	**53.34**	53.10	46.80	51.86	53.18	49.70	50.32	52.14	52.44	49.45
Avg. Kappa	39.94	41.12	25.30	38.03	42.44	41.62	**43.86**	41.94	38.29	32.04
Rank PMAUC	**2.89**	3.67	8.67	5.78	3.67	6.78	5.56	6.22	5.11	6.67
Rank Kappa	5.44	4.67	8.67	6.56	3.89	4.89	**3.67**	4.56	6.11	6.56

Bold font highlights the best result

each of the evaluated streams and the overall rank of the algorithms. Figure 42 provides a comparison of all algorithms.

Discussion

Impact of class imbalance approach. Semi-synthetic benchmarks allowed us to use real-world data to create much more challenging scenarios with rapidly evolving imbalance ratios. Thus, we preserved the desirable characteristics of real-world problems (such as mixed types of drift) but enhanced them with much more challenging problem from the imbalance standpoint. When analyzing the results, we can see that all classifiers formed two clusters when looking at their predictive performance.

For resampling-based methods, we can see that UOB and OOB returned opposite performance, despite them sharing similar core. Here, we can see the superiority of oversampling, which confirms observations found in (Korycki & Krawczyk, 2020), where authors of these semi-synthetic benchmarks postulated that smart oversampling is the best solution. ARFR again returned very similar performance to standard ARFR, highlighting that its predictive power can mainly be attributed to its robust core design.

For algorithm-level methods CSARF achieved the best-performing classifier regarding PMAUC, while surprisingly displaying good results on Kappa. ROSE, CALMID and MICFOAL displayed great performance, showing that their hybrid mechanisms

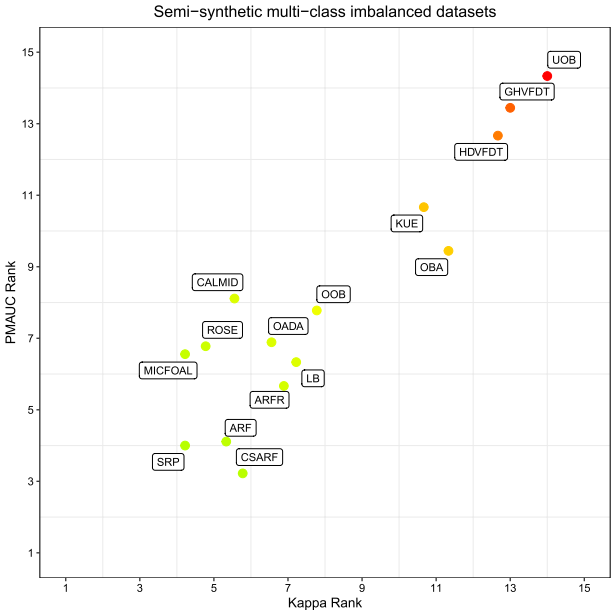

Fig. 42 Comparison of all 15 algorithms for semi-synthetic multi-class imbalanced datasets. Color gradient represents the product of both metrics (Color figure online)

are capable of efficient handling of rapid changes in imbalance ratios within real-world datasets.

Impact of ensemble architecture. By adding sudden and extreme changes in real-world benchmarks datasets, we could see an increase in the gap between best and worst performing methods. Similarly, to the previous experiments we can observe an excellent performance returned by SRP, showing a significant potential in using low-dimensional representations for imbalanced data streams, direction so far only explored in (Korycki & Krawczyk, 2021b).

7.3 Overall comparison

Goal of the experiment. The previous experiments discussed how different individual underlying data properties affected the performance of the classifiers. The goal of this experiment is to perform a joint comparison of the algorithms, identify performance trends and divergences, that will allow us to make recommendations to end-users. Moreover, we analyze the computational and memory complexity of the algorithms to address **RQ6**. The goal of any algorithm for data streams is to simultaneously maximize the classification metrics while minimizing the runtime and memory consumption (Krempl et al., 2014). However, these are often conflicting objectives and highly accurate methods often require long runtimes, which is not acceptable for real-time high-speed data streams. Table 21 shows the runtime and memory consumption of the 24 algorithms both for binary and multi-class imbalanced streams. Figures 43 and 44 present a pairwise joint comparison of the algorithm's ranks on G-Mean, PMAUC, Kappa, runtime and memory consumption across all experiments. Figure 45 shows a circular stacked barplot with the ranks for the four metrics. The bigger the stack the better aggregated performance. The circular barplot displays the algorithms sorted clockwise based on the stack size.

Table 21 Comparison of runtime (seconds per 1,000 instances) and memory consumption (RAM-Hours)

Algorithm	Binary class experiments				Multi-class experiments			
	Runtime (seconds)	Memory (RAM-Hours)	Runtime (Rank)	Memory (Rank)	Runtime (seconds)	Memory (RAM-Hours)	Runtime (Rank)	Memory (Rank)
IRL	0.15	3.93E-05	5.34	8.13	–	–	–	–
C-SMOTE	18.01	2.39E-02	18.98	20.75	–	–	–	–
VFC-SMOTE	2.51	2.52E-03	14.48	18.97	–	–	–	–
CSARF	3.97	1.12E-02	14.56	18.83	3.35	5.94E-06	12.53	13.70
GHVFDT	**0.01**	**1.16E-08**	**1.14**	**1.78**	0.09	**9.92E-11**	2.00	**1.35**
HDVFDT	**0.01**	2.85E-08	2.14	3.21	**0.08**	2.14E-10	**1.70**	1.75
ARF	3.65	8.42E-03	14.43	18.65	3.14	1.31E-06	11.93	12.78
KUE	0.11	4.93E-06	7.34	7.79	0.28	1.64E-08	6.30	5.13
LB	0.13	7.30E-06	8.17	9.16	0.33	1.21E-08	7.53	7.45
OBA	0.05	2.60E-07	5.03	5.91	0.25	2.25E-08	5.65	6.95
SRP	3.51	6.27E-03	15.03	18.54	5.34	8.56E-06	13.98	14.55
ESOS-ELM	2.61	1.11E-06	15.29	9.41	–	–	–	–
CALMID	0.09	2.80E-06	7.13	8.21	0.25	2.25E-08	5.65	6.95
MICFOAL	1.91	3.37E-03	12.72	16.98	1.95	3.72E-06	10.23	11.20
ROSE	0.13	6.45E-06	8.70	10.03	0.49	2.82E-08	8.90	8.70
OADA	24.48	2.10E-02	20.45	15.23	97.24	1.54E-04	15.00	10.08
OADAC2	31.61	2.11E-02	21.57	15.81	–	–	–	–
ARFR	1.25	8.73E-04	11.86	16.71	2.94	1.13E-06	11.35	12.38
SMOTE-OB	46.71	2.78E-01	21.10	21.08	–	–	–	–
OSMOTE	113.38	3.66E-01	23.29	15.70	–	–	–	–
OOB	0.07	1.94E-06	6.07	6.69	0.25	6.05E-08	5.28	4.90
UOB	0.03	3.55E-07	3.99	4.70	0.08	1.08E-09	2.33	3.25
ORUB	14.18	4.20E-03	18.57	12.32	–	–	–	–
OUOB	52.58	3.18E-01	22.60	15.41	–	–	–	–

Bold font highlights the best result

Discussion

Classification metrics. All the above experiments showcased the importance of using not only more than a single metric for evaluating classifiers for imbalanced data streams, but also the importance of using diverse and complimentary metrics. G-mean and PMAUC are strongly correlated with each other and follow the same trends, thus making using both redundant. However, by adding Kappa metric we gained an additional insight into specific characteristics of evaluated classifiers, thus allowing us to better understand which of the classifiers favor only minority classes and which return balanced performance over all analyzed classes.

Two best performing classifiers across all of experiments were ROSE and CSARF. ROSE returned single best performance regarding Kappa metric and one of the best for the other metrics. This allows us to conclude that ROSE is a well-rounded classifier that demonstrates robustness to various learning difficulties embedded in imbalanced and drifting data streams, both binary and multi-class. CSARF returned excellent results in

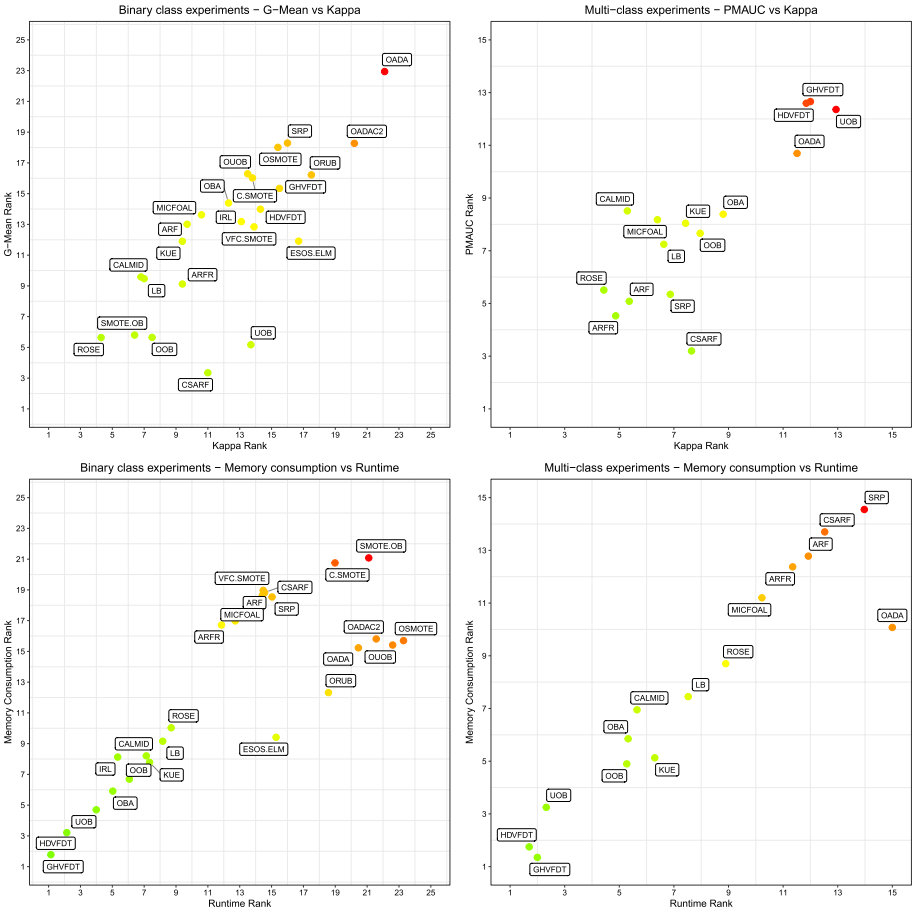

Fig. 43 Overall comparison of algorithms' ranks for G-Mean/PMAUC versus Kappa and Memory Consumption versus Runtime on binary and multi-class imbalanced benchmarks. Color gradient represents the product of each pair of metrics (Color figure online)

both types of experiments for G-Mean (for binary tasks) and PMAUC (for multi-class tasks) metrics. However, its rank dropped significantly under Kappa evaluation, showing that CSARF is driven by its performance on minority classes, not balanced performance on all of them. Furthermore, CSARF becomes unsuitable for scenarios with very high number of classes.

Other highly ranked classifiers included SMOTE-OB and OOB for binary scenarios and ARFR for multi-class ones. SMOTE-OB was the only classifier based on SMOTE that ranked among top performers, showing that SMOTE-based resampling for drifting streams needs to be further developed to achieve success, especially under instance-level difficulties. OOB did not get good results on multi-class scenarios, with close to average performance. Since SMOTE-OB does not support multi-class problems we could not evaluate it in this scenario. For multi-class imbalanced data streams, ARFR returned excellent results. This can be explained by the ability of its architecture to deal with multi-class scenarios and adapt to changes on multiple classes. This combined with a informed resampling

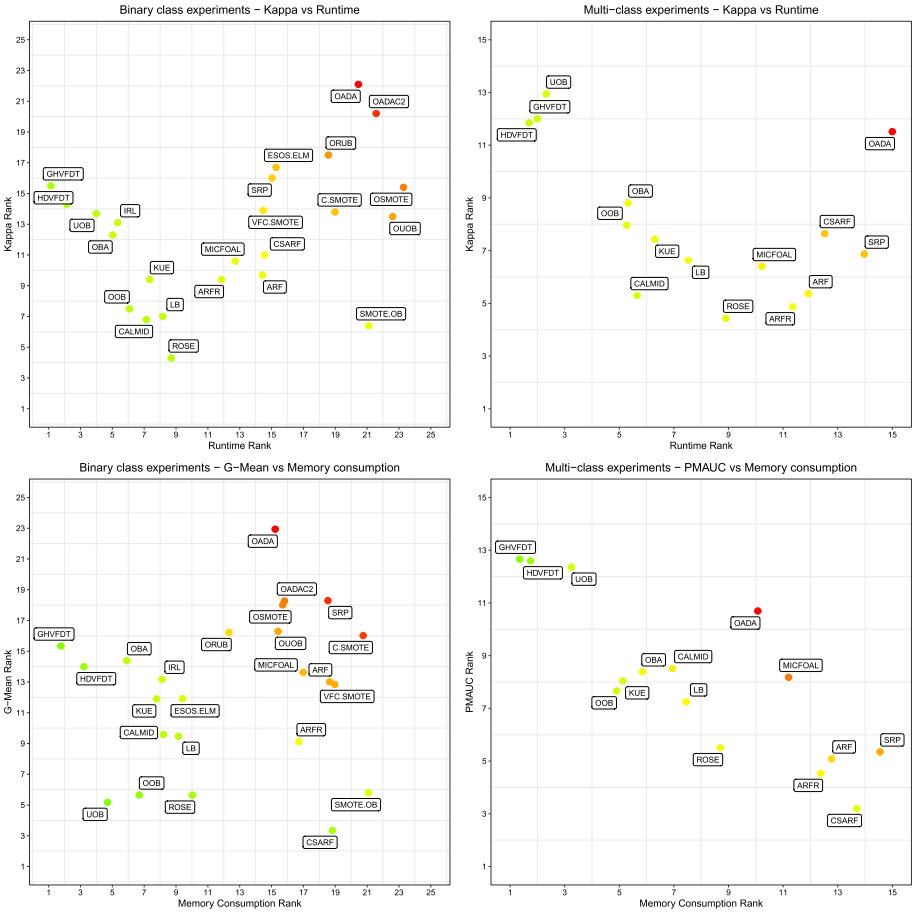

Fig. 44 Overall comparison of algorithms' ranks for G-Mean/PMAUC/Kappa versus Runtime/Memory Consumption on binary and multi-class imbalanced benchmarks. Color gradient represents the product of each pair of metrics (Color figure online)

approach lead to an effective classifier capable of handling multiple skewed classes in the stream.

OADA can be pointed out as the worst classifiers regarding classification metrics for both settings. This gives us insights about limitations of boosting-based ensembles for imbalanced data streams, where various learning difficulties destabilize the Boosting procedure and lead to low predictive power.

Computational and memory complexity. When evaluating a classifier for data stream mining, we have to take into account how much resources are needed to run it. In the streaming setting we often deal with a situation where memory or computational power is limited, thus we may not choose the best classifier, but the one that fits our scenario. HDVFDT and GHVFDT are characterized by a very small memory usage, and fast runtime. This happens because they are tree-based classifiers which are naturally lightweight, with simple structure and low-cost prediction mechanisms. UOB can also be seen as a relatively low-cost classifier, which is justified by its nature of removing samples from the data

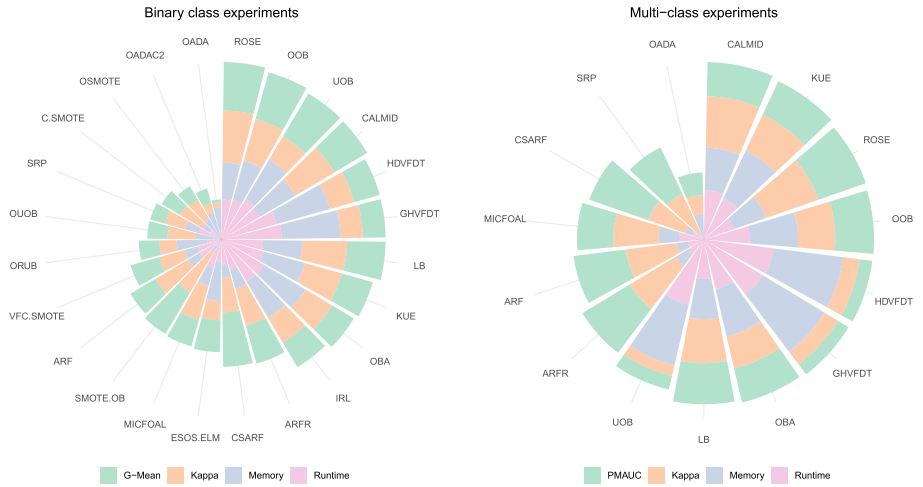

Fig. 45 Overall comparison of stacked algorithms' ranks for G-Mean/PMAUC, Kappa, runtime, and memory on binary and multi-class imbalanced benchmarks. Algorithms are sorted clockwise by the stacked ranks best to worst. Equal weight for the four metrics

streams in order to balance it. Therefore, we reduce the size of each batch and obtain more compact base classifiers.

When analyzing the classifiers that require the highest computational resources, we can see that they are dominated by oversampling-based approaches. This comes as an obvious observation, as oversampling increases the size of the already big data stream by generating a high number of artificial instances. Additionally, the increase in computational cost lies in the oversampling method itself. All SMOTE-based approaches rely on nearest neighbor computation to generate artificial instances, which leads to significant increases in their complexity. Out of approaches relying on blind oversampling (and thus free of nearest neighbor computations), OUOB and OSMOTE consumed the highest amount of resources. This can be explained by it employing resampling mechanisms combined with dynamic switching between them and drift detectors. Out of the classifiers that do not rely on resampling, OADA was the most computational heavy one. Although its memory consumption is similar to other ensemble approaches, the runtime was bigger that its peers. This is another motivation against using current boosting-based algorithms for imbalanced data streams.

Relationship between predictive power and computational and memory complexity. Is there a trade-off?. We can see that both analyzed criteria are often in a direct opposition to each other, the most lightweight classifiers are also among the worst performing ones. Therefore, how one can strike a balance between predictive power and computational complexity? How to select the best trade-off for imbalanced data streams? To select the most suitable classifier for a given data stream we cannot always get the best-performing regarding classification metrics, due to resources restrictions. For example, SMOTE-OB got excellent results regarding classification metrics, but often required more than 256GB of RAM per run, a prohibitive number for many real-world scenarios. On the other hand, we cannot also choose the lightweight classifier if it does not present good predictive power for the problem (e.g. single tree-based classifiers are non-competitive to most of ensembles for imbalanced data streams).

Analyzing all our experiments, we aim to select such classifiers that balance both sides. We can clearly see that OOB, UOB, ROSE and CALMID presented the best trade-off between their predictive performance and computational complexity for binary and multi-class experiments. ROSE presented the best overall performance when using equal weights for the predictive performance and complexity metrics. While second, the oversampling method in OOB demanded more memory and runtime when building the classifier. UOB undersamples the majority class which reduces the runtime complexity of the classifier learning. However, UOB has shown that undersampling in multi-class imbalanced data did not perform as in the binary scenario. ROSE and CALMID rely on highly efficient hybrid architectures and do not employ any costly mechanisms such as oversampling or adaptive cost-sensitive matrix. When focusing on the predictive metrics only, ROSE, SMOTE-OB, and CSARF perform the best on binary class while ARFR, ROSE and CSARF perform the best on multi-class.

8 Lessons learned

In order to address **RQ7** and summarize the knowledge we extracted through the extensive experimental evaluation, this section presents the lessons learned and recommendations for future researchers.

Design of the experimental study. To gain insights into the performance of classifiers and fairly evaluate them for imbalanced data streams, a properly designed experimental testbed is crucial. The experimental evaluation must be done in a holistic and comprehensive manner that will assess the robustness of the classifiers to the most important challenges embedded in imbalanced data streams. These must include: (i) static and dynamic imbalance ratios with switching class roles; (ii) instance-level difficulties; (iii) various types and speeds of concept drift; (iv) binary and multi-class scenarios; (v) increasing number of classes; and (vi) real-world datasets. Only such a comprehensive evaluation will allow for comparing new classifiers to existing state-of-the-art. For the sake of reproducible research, this paper offers a ready to use testbed available on GitHub that allows for easy and reproducible evaluation of new classifiers designed for imbalanced data streams.

Class imbalance approach. Our experiments showed that among the top performing methods we had two approaches based on training modifications (ROSE and CALMID), two approaches based on resampling (ARFR and SMOTE-OB), and one cost-sensitive method (CSARF). This is a very interesting outcome, as it shows that any of existing approaches to class imbalance can achieve excellent robustness and thus confirms the no-free-lunch theory - there is no single best way of tackling class imbalance in drifting data streams. Each of these solutions has their merits and works best in slightly different settings. In the next section we will formulate recommendations on what algorithms should be used in which scenarios. For future research it is important to understand what characteristics of each successful algorithm led to its superior performance, as those characteristics should be preserved and further developed when designing new classifiers.

Desirable properties of data-level solutions. When analyzing the resampling-based algorithms, we can see the dominance of oversampling approaches, both in their blind and informative versions. Blind oversampling has much lower computational cost and good reactivity to concept drift. However, it fails in multi-class scenarios, especially with high number of classes. Informative oversampling based on SMOTE, when combined with ensembles, offer a very high predictive power, being able to handle instance-level

difficulties and adapt to various types of non-stationary stream characteristics. This came at the price of extremely high computational complexity (mainly due to the distance calculations), as well as being currently designed only for binary problems.

Desirable properties of algorithm-level solutions. When analyzing the algorithm-level solutions, we can see that two main dominant approaches were based either on modifying training method or using cost-sensitive classification. ROSE stands as a primary example of effective training modification, as it offers top performance over a plethora of analyzed scenarios and the best robustness to various learning difficulties. This can be contributed to combination of diversity assurance for base classifiers (on both instance and feature levels), effective classifier replacement scheme (where pruning can replace multiple classifiers at once), and not relying on any resampling scheme (instead using class-specific buffers that allow for handling high number of classes). Those modifications allowed ROSE to strike a balance between predictive power (across all metrics) and its computational complexity. Cost-sensitive solution realized within CSARF showed that the combination of efficient design with cost matrix leads to a highly competitive classifier that offers great adaptation to concept drift and do not rely on any resampling. However, current limitations of cost-sensitive approaches include bias towards G-mean/PMAUC (while underperforming on Kappa) and inability to effectively handle higher number of classes.

Ensemble architectures. All experiments pointed out to the dominance of bagging-based and hybrid ensemble architectures (please note that most successful hybrid architectures were also rooted in bagging). Both static and dynamic ensemble setups worked well with bagging initialization, showing that this leads to creation of diverse base learners that can perform well under concept drift and various learning challenges. Furthermore, ensembles that added a feature space diversification on top of bagging, such as ROSE or ARFR were among the top performers. This shows that the feature space manipulation is a highly promising direction. Boosting proved to be the least efficient, not being able to cope with high imbalance ratios or data-level difficulties.

Adaptation to concept drift versus robustness to class imbalance. We can see that the most challenging scenarios where when dynamic class imbalance was combined with concept drift. Here we could observe that the classifiers either focused on drift adaptation, or handling bias towards majority classes. Interestingly, classifiers with very good adaptation mechanisms tend to perform slightly better in these complex scenarios than their counterparts that focus mainly on robustness to imbalance.

Data-level difficulties. Instance-level characteristics can be very disruptive to existing algorithms for imbalanced data streams. They should be analyzed not only as individual instances, but also as subconcepts within minority class that can evolve over time (e.g. merge or split). We can see that resampling-based solutions tend to perform well under these difficulties, mirroring observations for static data. However, none of the algorithms could explicitly use the instance-level characteristics to their advantage, as suggested by (Krawczyk & Skryjomski, 2017).

Handling high number of classes. When analyzing the robustness of classifiers to very high number of classes, we observed that SRP, a general-purpose ensemble with no skew-insensitive mechanisms, returns one of the best performances. This, combined with very good performance of ROSE, allows us to conclude that for multi-class imbalanced problems with very high number of classes using lower dimensional representations may lead to simplification of learning tasks. Using feature subspaces may lead to more diverse capturing of relationships among classes. This confirms observations made by (Korycki & Krawczyk, 2021b) that discussed the merit of low-dimensional embeddings for extremely imbalanced and difficult data streams.

Classifier evaluation. To evaluate a classifier in imbalanced data streams, we require the use of multiple diverse and complimentary metrics. In our testbed we argue for the use of Kappa and G-Mean/PMAUC. These metrics assess different and complementary perspectives, thus if only one is provided the evaluation of a classifier is biased towards measuring how it performs on minority class under highly imbalance ratio (Kappa) or on how it balances majority and minority class performance (PMAUC/G-mean). We showed how under high imbalance ratios, Kappa significantly penalizes the false positives whereas G-Mean tolerates a larger proportion of false positives.

Computational and memory complexity. One must take into an account the trade-off between predictive power and computational complexity. Algorithms requiring lowest resource consumption are among the weakest ones (such as skew-insensitive versions of Adaptive Very Fast Decision Trees). On the other hand, some of the best performing classifiers are characterized by almost prohibitive computational complexity (e.g. SMOTE-OB). ROSE, CALMID and OOB presented the best trade-off between computational resources consumption and predictive power.

9 Recommendations

After analyzing all the scenarios and evaluating different approaches to class imbalance, we could summarize some recommendations to help future researchers when designing their own algorithms to tackle imbalanced data streams and other learning difficulties:

Choose the best off-the-shelf algorithms. If you are looking for efficient classifiers for solving your real-world imbalanced data streams, or you are looking for effective reference methods for your experiments, it is important to be aware of the most efficient off-the shelf solutions. Based on our exhaustive experimental study, we can recommend ROSE, CSARF, OOB, ARFR, and CALMID as the ready to use and effective classifiers. We especially recommend using ROSE due to its balanced performance, great trade-off between predictive power and computational cost, excellent robustness in all analyzed scenarios, as well as ease of use due to its autonomously self-adaptive parameters.

Analyze the dynamics of imbalance ratio. In data streams where imbalance ratio is static, oversampling and training modification methods return excellent performance. Ensembles based on bagging and hybrid architecture are a good choice. When it comes to evolving imbalance ratios, we need a more sophisticated mechanism adapting to the changing imbalance ratio. Here we can see a dominance of algorithm-level solutions that offer dynamic ensemble line-up with effective pruning, such as ROSE.

Consider the presence of concept drift. Our experiments showed that many skew-insensitive classifiers suffer due to their lackluster adaptation mechanisms. On the other hand, general-purpose classifiers can display surprisingly good performance in specific cases, showing the impact of recovery from concept drift. This allows us to recommend paying close attention to embedding an efficient concept drift adaptation mechanism into your method. Regardless of how robust your skew-insensitive mechanism will be, it will not be sufficient to cope with the drifting nature of imbalanced data streams.

Check for instance-level difficulties. Instance-level difficulties in data streams pose significant difficulties to most of the classifiers (Brzeziński et al., 2021). It is crucial to analyze your stream to understand if such factors are present. We noticed that methods based on oversampling tend to handle instance-level difficulties particularly well. However, none of them can directly take an advantage of such challenging instances to

improve adaptation and robustness. Existing research suggest that incorporating such information during learning from imbalanced streams may be highly beneficial (Krawczyk & Skryjomski, 2017). Therefore, we recommend to truly understand the nature of streams you are working with and focusing on how you can leverage this information to make your classifiers more robust.

Consider the number of classes. There is a significant difference in developing methods for binary and multi-class imbalanced data streams. While some of algorithms work well regardless of the number of classes (e.g. ROSE), other are very sensitive to it and their performance deteriorates significantly with increase in the number of classes (e.g. CSARF). Multi-class data streams will require the development of dedicated resampling algorithms, just like in the static scenarios (Krawczyk et al., 2020). Existing resampling methods work well mainly in binary cases and do not translate well to a higher number of classes. Finally, most of the existing classifiers work under fixed number of classes. This should be considered when dealing with emerging and disappearing classes, as existing classifiers need to be extended with dedicated mechanisms to handle this phenomenon (Masud et al., 2009, 2010a, b).

Think outside of the box. While data-level and algorithm-level solutions are the most popular approaches to handling class imbalance, there are other promising directions to explore. Instead of focusing on another online resampling method or cost-sensitive modification, explore alternative solutions. Our experiments showed the high promise behind low-dimensional representations for imbalanced data streams, as firstly explored by (Korycki & Krawczyk, 2021b). This is just the tip of an iceberg in developing novel techniques tailored to imbalanced data streams that do not follow these two most popular directions.

Use fair and holistic evaluation. New classifiers for imbalanced data streams should always be compared with both the popular methods (e.g. OOB or UOB), as well as with the most recently published and top performing ones (as of the time of this study these will include ROSE, CSARF, or OOB). It is important to use an established experimental setup and follow the best practices in this field. This paper provides reproducible code for the entire testbed, along with all examined classifiers and datasets. This is the first standardized approach for evaluating classifiers for imbalanced data streams. We recommend for future researchers to simply plug-in their new methods into our framework to ensure fair and holistic evaluation of newly proposed methods.

Do not neglect using general-purpose ensembles as reference. Our experiments showed that general-purpose ensembles can return surprisingly good performance for non-stationary imbalanced data streams, due to their well-designed drift adaptation mechanisms. Therefore, it is important to use them as a point of reference to see if the proposed skew-insensitive mechanism actually contributes significantly to the performance of a new classifier.

Use multiple performance metrics. There are many performance metrics for evaluating imbalanced data streams including Kappa, G-Mean, PMAUC, WMAUC, EWMAUC. Section 6.3 presented the different aspects these performance metrics assess, and acknowledged the different biases in individual metrics. We recommend using multiple metrics exhibiting complementary behavior rather than picking a single metric.

Ensure reproducible research. Reproducible research is the key towards the advancement of the machine learning community. If you want your method to have an impact, always provide the source code on GitHub and use popular frameworks such as MOA (Bifet et al., 2010b), River (Montiel et al., 2020), Stream-learn (Ksieniewicz & Zyblewski, 2022), and Scikit-multiflow (Montiel et al., 2018). This will make sure that other researchers can

use your classifier, as well as that it can be easily embedded in existing frameworks, for comparison with other methods.

One size does not fit all. This survey paper presents a very large experimental evaluation of as many imbalanced data scenarios as possible in order to compare existing methods in the state of the art. It is not our intention nor realistic that every study from now on is required to always use the full set of benchmarks. Our goal is that future works can build on our recommendations to include some of the benchmarks proposed as appropriate in each work, acknowledging that not all of them are necessary nor suitable for all studies.

10 Open challenges and future directions

After formulating recommendations regarding the currently available algorithms, we will now present and discuss open challenges and future directions for learning from imbalanced data streams.

Informative and fast resampling. Our experimental study showed that current undersampling-based methods underperform for imbalanced data streams, especially when faced with multiple classes. There is a need to develop novel and informative undersampling approaches that can adapt to concept drift and allow to efficiently tackle dynamic class imbalance, while preserving the desirable low computational complexity. Current informative oversampling methods are rooted in SMOTE, offering good improvements in predictive power at the high computational cost. We should develop novel oversampling methods that do not rely on a nearest neighbor approach, thus reducing the computational complexity and alleviating SMOTE limitations (Krawczyk et al., 2020).

Proactive instead of reactive tackling of dynamic class imbalance. Existing methods focus on adaptation to both concept drift and dynamic class imbalance after the change has taken place. But is there a possibility to anticipate the change? Can we predict how the class imbalance will evolve over time and offer proactive approach? This would significantly reduce the recovery time after changes in data streams and lead to more robust classifiers.

Improving boosting-based ensembles. We have discussed how existing boosting-based ensembles perform poorly for imbalanced data streams. Yet boosting is one of the most successful ensemble architectures and deserves a second chance. We hope that the weaknesses of boosting identified in this paper will help other researchers develop more suitable classifiers based on this architecture, capable of fast adaptation to changes and overcoming small sample size in minority classes.

Handling evolving number of classes. While we investigated the impact of the number of classes on imbalanced problems, we have not touched upon dynamic changes in class numbers (Masud et al., 2009, 2010a, b). In data stream scenarios classes may emerge, disappear, and recur over time. An evolving number of classes combined with dynamic imbalance ratio creates an extremely challenging scenario that requires new and flexible models capable of detecting and incorporating new classes into their structures, as well as forgetting the outdated classes and remembering recurring classes (Masud et al., 2011, 2012; Al-Khateeb et al., 2012; Sun et al., 2016). We envision strong parallels with continual and lifelong learning approaches (Korycki & Krawczyk, 2021a).

Fairness in imbalanced data streams. Algorithmic fairness is a subject of intense research (Iosifidis et al., 2021), aiming at creating non-biased classifiers that do not rely on protected attributes. Recent works by suggest that algorithmic fairness and class imbalance

are the two sides of the same coin, as protected information is often displayed by under-represented, minority groups. Fairness in data stream mining could benefit from enhancing existing methods with skew-insensitive approaches, as both domains aim at countering bias in data.

Online skew-insensitive feature selection. We have noticed a superior performance of ensembles based on reduced feature subspaces, especially for difficult multi-class problems. While existing methods are based on randomized approaches, there is a need to develop efficient online feature selection methods insensitive to class imbalance. This will allow not only to create more compact classifier, filter irrelevant features, but also eliminate features that increase bias towards the majority class. This could further be expanded into scenarios where the feature space size evolves over time.

Beyond binary and multi-class imbalanced data streams. Most of the existing research in imbalanced data streams focuses on binary and multi-class classification. However, multiple other tasks in data streams may be subject to data imbalance. Multi-label data is inherently imbalanced and calls for dedicated methods capable of handling multi-target outputs (Alberghini et al., 2022). Regression from streams is also frequently subject to imbalance in the form of rare values, as frequencies of specific ground truths may evolve over time (Branco et al., 2017; Aminian et al., 2021). Finally, streaming times series also require dedicated resampling and skew-insensitive methods to facilitate robust predictions.

11 Conclusions

Summary. In this paper, we offered an exhaustive and informative experimental review of classification methods for imbalanced data streams. We designed a robust experimental framework, publicly available for reproducibility, to evaluate state-of-the-art classifiers in varied scenarios and understand how each aspect of imbalanced data streams affects the performance of classifiers, and provide a template for future researchers to evaluate their newly classifiers with the state of the art. With this experimental framework, we performed an experimental comparison with 24 algorithms in multiple scenarios to analyze their behavior and discuss their performance trends and divergences. The classifiers were evaluated on 515 benchmarks with different difficulties such as dynamic and static imbalance ratio, with and without concept drift, the presence of data-level difficulty factors, and real-world problems. All these settings were evaluated isolated and combined, in a binary and multi-class scenario, to gain insights and understand how they would affect the underlying learning mechanisms of data-streams classifiers. Throughout the experiments, we could demonstrate which approaches work or do not work for each scenario, such as undersampling techniques were undermined in multi-class scenarios, and dynamic ensemble methods such as ROSE could do better in many different settings, demonstrating robustness. Our proposed experimental framework allowed us to get insights into all the classifiers and how would they perform in different scenarios, therefore future researchers can follow the same standard of evaluation when proposing their classifier for imbalanced data streams, in order to achieve the most transparent and complete results possible.

Towards the future of reproducible research in data stream mining. We proposed a standardized and holistic framework for evaluating imbalanced data streams. We strongly believe that this is a crucial step towards unifying the community working in this domain, offering a flexible tool for long-time practitioners, and an easy way to get started for newcomers. Guidelines and recommendations formulated in this paper should allow more

streamlined and effective improvement of existing algorithms and development of new solutions. Only as a community working together, we can truly advance our understanding of data streams and design truly impactful, well-rounded, and thoroughly evaluated algorithms that will be used in both academia and industry.

We hope that our framework will begin to grow over time with new algorithms, problems, and benchmarks being added by the community. There are still many questions unanswered in this domain and many open challenges for the future. We look forward to discovering new knowledge together.

Acknowledgements 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.

Author contributions Gabriel Aguiar contributed to the manuscript preparation. Alberto Cano contributed to the experimental evaluation and manuscript preparation. Bartosz Krawczyk contributed to the manuscript preparation. All authors read and approved the final manuscript.

Funding This research was partially supported by the 2018 VCU Presidential Research Quest Fund (Alberto Cano) and an Amazon AWS Machine Learning Research award (Alberto Cano & Bartosz Krawczyk).

Data availability Data & materials available at https://github.com/canoalberto/imbalanced-streams.

Code availability Source code is available at https://github.com/canoalberto/imbalanced-streams.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Abolfazli, A., & Ntoutsi, E. (2020). Drift-aware multi-memory model for imbalanced data streams. In *IEEE international conference on big data* (pp. 878–885).

Aguiar, G., & Cano, A. (2023). An active learning budget-based oversampling approach for partially labeled multi-class imbalanced data streams. In *38th ACM/SIGAPP symposium on applied computing* (pp. 1–8).

Al-Khateeb, T., Masud, M. M., Khan, L., Aggarwal, C., Han, J., & Thuraisingham, B. (2012). Stream classification with recurring and novel class detection using class-based ensemble. In *IEEE international conference on data mining* (pp. 31–40).

Al-Shammari, A., Zhou, R., Naseriparsaa, M., & Liu, C. (2019). An effective density-based clustering and dynamic maintenance framework for evolving medical data streams. *International Journal of Medical Informatics, 126*, 176–186.

Alberghini, G., Barbon, S., & Cano, A. (2022). Adaptive ensemble of self-adjusting nearest neighbor subspaces for multi-label drifting data streams. *Neurocomputing, 481*, 228–248.

Aminian, E., Ribeiro, R. P., & Gama, J. (2019). A study on imbalanced data streams. In *European conference on machine learning and knowledge discovery in databases* (pp. 380–389).

Aminian, E., Ribeiro, R. P., & Gama, J. (2021). Chebyshev approaches for imbalanced data streams regression models. *Data Mining and Knowledge Discovery, 35*(6), 2389–2466.

Ancy, S., & Paulraj, D. (2020). Handling imbalanced data with concept drift by applying dynamic sampling and ensemble classification model. *Computer Communications, 153*, 553–560.

Anupama, N., & Jena, S. (2019). A novel approach using incremental oversampling for data stream mining. *Evolving Systems, 10*(3), 351–362.

Arya, M., & Hanumat Sastry, G. (2022). A novel deep ensemble learning framework for classifying imbalanced data stream. In *IOT with smart systems* (pp. 607–617).

Bahri, M., Bifet, A., Gama, J., Gomes, H. M., & Maniu, S. (2021). Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11*, e1405.

Barros, R. S. M., & Santos, S. G. T. C. (2018). A large-scale comparison of concept drift detectors. *Information Sciences, 451*, 348–370.

Bernardo, A., & Della Valle, E. (2021a). SMOTE-OB: Combining SMOTE and online bagging for continuous rebalancing of evolving data streams. In *IEEE international conference on big data* (pp. 5033–5042).

Bernardo, A., & Della Valle, E. (2021b). VFC-SMOTE: Very fast continuous synthetic minority oversampling for evolving data streams. *Data Mining and Knowledge Discovery, 35*(6), 2679–2713.

Bernardo, A., Della Valle, E., & Bifet, A. (2020a). Incremental rebalancing learning on evolving data streams. In *International conference on data mining workshops* (pp. 844–850).

Bernardo, A., Gomes, H. M., Montiel, J., Pfahringer, B., Bifet, A., & Della Valle, E. (2020b). C-SMOTE: Continuous synthetic minority oversampling for evolving data streams. In *IEEE international conference on big data* (pp. 483–492).

Bernardo, A., Ziffer, G., & Valle, E. D. (2021). IEBench: Benchmarking streaming learners on imbalanced evolving data streams. In: *International conference on data mining* (pp. 331–340).

Bhowmick, K., & Narvekar, M. (2022). A semi-supervised clustering-based classification model for classifying imbalanced data streams in the presence of scarcely labelled data. *International Journal of Business Intelligence and Data Mining, 20*(2), 170–191.

Bian, S., & Wang, W. (2007). On diversity and accuracy of homogeneous and heterogeneous ensembles. *International Journal of Hybrid Intelligent Systems, 4*(2), 103–128.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavalda, R. (2009). New ensemble methods for evolving data streams. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 139–148).

Bifet, A., Holmes, G., & Pfahringer, B. (2010a). Leveraging bagging for evolving data streams. In *European conference on machine learning and knowledge discovery in databases* (pp. 135–150).

Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., & Seidl, T. (2010b). MOA: Massive online analysis, a framework for stream classification and clustering. In *Workshop on applications of pattern analysis* (pp. 44–50).

Bobowska, B., Klikowski, J., & Woźniak, M. (2019). Imbalanced data stream classification using hybrid data preprocessing. In *European conference on machine learning and knowledge discovery in databases* (pp. 402–413).

Bourdonnaye, F. D. L., & Daniel, F. (2022). Evaluating resampling methods on a real-life highly imbalanced online credit card payments dataset. CoRR arXiv:2206.13152.

Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR), 49*(2), 1–50.

Branco, P., Torgo, L., & Ribeiro, R. P. (2017). SMOGN: a pre-processing approach for imbalanced regression. In *International workshop on learning with imbalanced domains: Theory and applications* (pp. 36–50).

Brzeziński, D., & Stefanowski, J. (2017). Prequential AUC: Properties of the area under the ROC curve for data streams with concept drift. *Knowledge and Information Systems, 52*(2), 531–562.

Brzeziński, D., & Stefanowski, J. (2018). Ensemble classifiers for imbalanced and evolving data streams. In *Data mining in time series and streaming databases* (pp. 44–68). World Scientific.

Brzeziński, D., Stefanowski, J., Susmaga, R., & Szczęch, I. (2018). Visual-based analysis of classification measures and their properties for class imbalanced problems. *Information Sciences, 462*, 242–261.

Brzeziński, D., Stefanowski, J., Susmaga, R., & Szczech, I. (2019). On the dynamics of classification measures for imbalanced and streaming data. *IEEE Transactions on Neural Networks and Learning Systems, 31*(8), 2868–2878.

Brzeziński, D., Minku, L. L., Pewinski, T., Stefanowski, J., & Szumaczuk, A. (2021). The impact of data difficulty factors on classification of imbalanced and concept drifting data streams. *Knowledge and Information Systems, 63*, 1429–1469.

Cano, A., & Krawczyk, B. (2019). Evolving rule-based classifiers with genetic programming on GPUs for drifting data streams. *Pattern Recognition, 87*, 248–268.

Cano, A., & Krawczyk, B. (2020). Kappa Updated Ensemble for drifting data stream mining. *Machine Learning, 109*, 175–218.

Cano, A., & Krawczyk, B. (2022). ROSE: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Machine Learning, 111*, 2561–2599.

Chrysakis, A., & Moens, M. (2020). Online continual learning from imbalanced data. *International Conference on Machine Learning, 119*, 1952–1961.

Cieslak, D. A., & Chawla, N. V. (2008). Learning decision trees for unbalanced data. In *European conference on machine learning and knowledge discovery in databases* (pp. 241–256).

Czarnowski, I. (2021). Learning from imbalanced data streams based on over-sampling and instance selection. In *International conference on computational science* (pp. 378–391).

Czarnowski, I. (2022). Weighted Ensemble with one-class Classification and Over-sampling and Instance selection (WECOI): An approach for learning from imbalanced data streams. *Journal of Computational Science, 61*, 101614.

da Costa, V. G. T., de Leon Ferreira, A. C. P., Junior, S. B., et al. (2018). Strict Very Fast Decision Tree: A memory conservative algorithm for data stream mining. *Pattern Recognition Letters, 116*, 22–28.

Ditzler, G., Roveri, M., Alippi, C., & Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine, 10*(4), 12–25.

Du, H., Zhang, Y., Gang, K., Zhang, L., & Chen, Y. C. (2021). Online ensemble learning algorithm for imbalanced data stream. *Applied Soft Computing, 107*, 107378.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets* (Vol. 10). Springer.

Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research, 61*, 863–905.

Ferreira, L. E. B., Gomes, H. M., Bifet, A., & Oliveira, L. S. (2019). Adaptive random forests with resampling for imbalanced data streams. In *International joint conference on neural networks* (pp. 1–6).

Galar, M., Fernández, A., Barrenechea, E., & Herrera, F. (2014). Empowering difficult classes with a similarity-based aggregation in multi-class classification problems. *Information Sciences, 264*, 135–157.

Gama, J. (2010). *Knowledge discovery from data streams*. CRC Press.

Gama, J. (2012). A survey on learning from data streams: current and future trends. *Progress in Artificial Intelligence, 1*, 45–55.

Gama, J., Sebastiao, R., & Rodrigues, P. P. (2013). On evaluating stream learning algorithms. *Machine Learning, 90*(3), 317–346.

Gao, K. (2015). Online one-class SVMs with active-set optimization for data streams. In *IEEE international conference on machine learning and applications* (pp. 116–121).

García, V., Sánchez, J. S., & de Jesús Ochoa Domínguez H, Cleofas-Sánchez L,. (2015). Dissimilarity-based learning from imbalanced data with small disjuncts and noise. *Pattern Recognition and Image Analysis, Lecture Notes in Computer Science,9117*, 370–378.

Ghazikhani, A., Monsefi, R., & Yazdi, H. S. (2013). Ensemble of online neural networks for non-stationary and imbalanced data streams. *Neurocomputing, 122*, 535–544.

Ghazikhani, A., Monsefi, R., & Yazdi, H. S. (2014). Online neural network model for non-stationary and imbalanced data stream classification. *International Journal of Machine Learning and Cybernetics, 5*(1), 51–62.

Gomes, H. M., Barddal, J. P., Enembreck, F., & Bifet, A. (2017a). A survey on ensemble learning for data stream classification. *ACM Computing Surveys (CSUR), 50*(2), 1–36.

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., Holmes, G., & Abdessalem, T. (2017b). Adaptive random forests for evolving data stream classification. *Machine Learning, 106*(9), 1469–1495.

Gomes, H. M., Read, J., & Bifet, A. (2019). Streaming random patches for evolving data stream classification. In *IEEE international conference on data mining* (pp. 240–249).

Gomes, H. M., Grzenda, M., Mello, R., Read, J., Le Nguyen, M. H., & Bifet, A. (2022). A survey on semi-supervised learning for delayed partially labelled data streams. *ACM Computing Surveys, 55*(4), 1–42.

Grzyb, J., Klikowski, J., & Woźniak, M. (2021). Hellinger distance weighted ensemble for imbalanced data stream classification. *Journal of Computational Science, 51*, 101314.

Guo, N., Yu, Y., Song, M., Song, J., & Fu, Y. (2013). Soft-CsGDT: soft cost-sensitive Gaussian decision tree for cost-sensitive classification of data streams. *International workshop on big data, streams and heterogeneous source mining: algorithms, systems, programming models and applications* (pp. 7–14).

Han, M., Chen, Z., Li, M., Wu, H., & Zhang, X. (2022). A survey of active and passive concept drift handling methods. *Computational Intelligence, 38*(4), 1492–1535.

Han, M., Zhang, X., Chen, Z., Wu, H., & Li, M. (2023). Dynamic ensemble selection classification algorithm based on window over imbalanced drift data stream. *Knowledge and Information Systems, 65*(3), 1105–1128.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284.

He, H., & Ma, Y. (2013). *Imbalanced learning: Foundations, algorithms, and applications*. Wiley.

Iosifidis, V., Zhang, W., & Ntoutsi, E. (2021). Online fairness-aware learning with imbalanced data streams. arXiv preprint arXiv:2108.06231.

Japkowicz, N. (2013). *Assessment metrics for imbalanced learning. Imbalanced learning: Foundations, algorithms, and applications* (pp. 187–206).
Jedrzejowicz, J., & Jedrzejowicz, P. (2020). GEP-based classifier with drift detection for mining imbalanced data streams. *Procedia Computer Science, 176*, 41–49.
Jiao, B., Guo, Y., Gong, D., & Chen, Q. (2022). Dynamic ensemble selection for imbalanced data streams with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*.
Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., & Ghédira, K. (2018). Discussion and review on evolving data streams and concept drift adapting. *Evolving Systems, 9*(1), 1–23.
Kim, C. D., Jeong, J., & Kim, G. (2020). Imbalanced continual learning with partitioning reservoir sampling. In *European conference on computer vision* (vol. 12358, pp. 411–428).
Klikowski, J., & Woźniak, M. (2019). Multi sampling random subspace ensemble for imbalanced data stream classification. In *International conference on computer recognition systems* (pp. 360–369).
Klikowski, J., & Woźniak, M. (2020). Employing one-class SVM classifier ensemble for imbalanced data stream classification. In *International conference on computational science* (pp. 117–127).
Klikowski, J., & Wozniak, M. (2022). Deterministic sampling classifier with weighted bagging for drifted imbalanced data stream classification. *Applied Soft Computing*, 108855.
Komorniczak, J., Zyblewski, P., & Ksieniewicz, P. (2021). Prior probability estimation in dynamically imbalanced data streams. In *International joint conference on neural networks* (pp. 1–7).
Korycki, Ł., Cano, A., & Krawczyk, B. (2019). Active learning with abstaining classifiers for imbalanced drifting data streams. In *IEEE international conference on big data* (pp. 2334–2343).
Korycki, Ł., & Krawczyk, B. (2020). Online oversampling for sparsely labeled imbalanced and non-stationary data streams. In *International joint conference on neural networks* (pp. 1–8).
Korycki, L., & Krawczyk, B. (2021a). Class-incremental experience replay for continual learning under concept drift. In *IEEE conference on computer vision and pattern recognition workshops* (pp. 3649–3658).
Korycki, L., & Krawczyk, B. (2021b). Low-dimensional representation learning from imbalanced data streams. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 629–641).
Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence, 5*(4), 221–232.
Krawczyk, B. (2021). Tensor decision trees for continual learning from drifting data streams. *Machine Learning, 110*(11), 3015–3035.
Krawczyk, B., Galar, M., Wozniak, M., Bustince, H., & Herrera, F. (2018). Dynamic ensemble selection for multi-class classification with one-class classifiers. *Pattern Recognition, 83*, 34–51.
Krawczyk, B., Koziarski, M., & Wozniak, M. (2020). Radial-based oversampling for multiclass imbalanced data classification. *IEEE Transactions on Neural Networks and Learning Systems, 31*(8), 2818–2831.
Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion, 37*, 132–156.
Krawczyk, B., & Skryjomski, P. (2017). Cost-sensitive perceptron decision trees for imbalanced drifting data streams. In *European conference on machine learning and knowledge discovery in databases* (pp. 512–527).
Krawczyk, B., & Wozniak, M. (2015). One-class classifiers with incremental learning and forgetting for data streams with concept drift. *Soft Computing, 19*(12), 3387–3400.
Krempl, G., Žliobaitė, I., Brzeziński, D., Hüllermeier, E., Last, M., Lemaire, V., Noack, T., Shaker, A., Sievi, S., Spiliopoulou, M., et al. (2014). Open challenges for data stream mining research. *ACM SIGKDD Explorations Newsletter, 16*(1), 1–10.
Ksieniewicz, P. (2021). The prior probability in the batch classification of imbalanced data streams. *Neurocomputing, 452*, 309–316.
Ksieniewicz, P., & Zyblewski, P. (2022). Stream-learn—open-source python library for difficult data stream batch analysis. *Neurocomputing, 478*, 11–21.
Lango, M., & Stefanowski, J. (2022). What makes multi-class imbalanced problems difficult? An experimental study. *Expert Systems with Applications, 199*, 116962.
Lee, K. J. (2018). Online class imbalance learning for quality estimation in manufacturing. In *IEEE international conference on emerging technologies and factory automation* (pp. 1007–1014).
Li, Z., Huang, W., Xiong, Y., Ren, S., & Zhu, T. (2020). Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm. *Knowledge-Based Systems, 195*, 105694.
Li-wen, W., Wei, G., & Yi-cheng, Y. (2021). An online weighted sequential extreme learning machine for class imbalanced data streams. *Journal of Physics: Conference Series, 19–4*(1), 012008.
Liang, X., Song, X., Qi, K., Li, J., Liu, J., & Jian, L. (2021). Anomaly detection aided budget online classification for imbalanced data streams. *IEEE Intelligent Systems, 36*(3), 14–22.

Lipska, A., & Stefanowski, J. (2022). The influence of multiple classes on learning online classifiers from imbalanced and concept drifting data streams. arXiv preprint arXiv:2210.08359.

Liu, W., Zhang, H., Ding, Z., Liu, Q., & Zhu, C. (2021). A comprehensive active learning method for multiclass imbalanced data streams with concept drift. *Knowledge-Based Systems, 215*, 106778.

Liu, X., Fu, J., & Chen, Y. (2020). Event evolution model for cybersecurity event mining in tweet streams. *Information Sciences, 524*, 254–276.

Loezer, L., Enembreck, F., Barddal, J. P., de Souza Britto, Jr. A. (2020). Cost-sensitive learning for imbalanced data streams. In *ACM symposium on applied computing* (pp. 498–504).

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering, 31*(12), 2346–2363.

Lu, Y., Cheung, Y.m., & Tang, Y.Y. (2017). Dynamic weighted majority for incremental learning of imbalanced data streams with concept drift. In *International joint conference on artificial intelligence* (pp. 2393–2399).

Lu, Y., Cheung, Y. M., & Tang, Y. Y. (2020). Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift. *IEEE Transactions on Neural Networks and Learning Systems, 31*, 2764–2778.

Luong, A. V., Vu, T. H., Nguyen, P. M., Pham, N. V., McCall, J. A. W., Liew, A. W., & Nguyen, T. T. (2020). A homogeneous-heterogeneous ensemble of classifiers. In *Neural information processing—27th international conference, ICONIP 2020, Bangkok, Thailand, November 18–22, 2020, Proceedings, Part V, Springer, Communications in Computer and Information Science*, (vol. 1333, pp. 251–259).

Luque, A., Carrasco, A., Martín, A., & de Las, Heras A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition, 91*, 216–231.

Lyon, R. J., Knowles, J. D., Brooke, J. M., & Stappers, B. W. (2014). Hellinger distance trees for imbalanced streams. In *IEEE international conference on pattern recognition* (pp. 1969–1974).

Malialis, K., Panayiotou, C. G., & Polycarpou, M. M. (2022). Nonstationary data stream classification with online active learning and siamese neural networks. *Neurocomputing, 512*, 235–252.

Marwa, T., Ouadfel, S., & Meshoul, S. (2021). Hybrid ensemble approaches to online harassment detection in highly imbalanced data. *Expert Systems with Applications, 175*, 114751.

Masud, M. M., Al-Khateeb, T. M., Khan, L., Aggarwal, C., Gao, J., Han, J., Thuraisingham, B. (2011). Detecting recurring and novel classes in concept-drifting data streams. In *IEEE international conference on data mining* (pp. 1176–1181).

Masud, M. M., Chen, Q., Khan, L., Aggarwal, C. C., Gao, J., Han, J., Srivastava, A., & Oza, N. C. (2012). Classification and adaptive novel class detection of feature-evolving data streams. *IEEE Transactions on Knowledge and Data Engineering, 25*(7), 1484–1497.

Masud, M. M., Chen, Q., Khan, L., Aggarwal, C., Gao, J., Han, J., & Thuraisingham, B. (2010b). Addressing concept-evolution in concept-drifting data streams. In *IEEE international conference on data mining* (pp. 929–934).

Masud, M. M., Gao, J., Khan, L., Han, J., & Thuraisingham, B. (2009). Integrating novel class detection with classification for concept-drifting data streams. In *European conference on machine learning and knowledge discovery in databases* (pp. 79–94).

Masud, M., Gao, J., Khan, L., Han, J., & Thuraisingham, B. M. (2010a). Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering, 23*(6), 859–874.

Mirza, B., Lin, Z., & Liu, N. (2015). Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. *Neurocomputing, 149*, 316–329.

Mohammed, R. A., Wong, K. W., Shiratuddin, M. F., & Wang, X. (2020a). Classification of multi-class imbalanced data streams using a dynamic data-balancing technique. In *International conference on neural information processing* (pp. 279–290).

Mohammed, R. A., Wong, K. W., Shiratuddin, M. F., & Wang, X. (2020b). PWIDB: A framework for learning to classify imbalanced data streams with incremental data re-balancing technique. *Procedia Computer Science, 176*, 818–827.

Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H. M., Read, J., Abdessalem, T., & Bifet, A. (2020). River: Machine learning for streaming data in python. arxiv:2012.04740.

Montiel, J., Read, J., Bifet, A., & Abdessalem, T. (2018). Scikit-multiflow: A multi-output streaming framework. *The Journal of Machine Learning Research, 19*(1), 2915–2914.

Napierala, K., & Stefanowski, J. (2016). Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems, 46*(3), 563–597.

Nguyen, H. L., Woon, Y. K., & Ng, W. K. (2015). A survey on data stream clustering and classification. *Knowledge and Information Systems, 45*, 535–569.

Nguyen, V. L., Destercke, S., & Masson, M. H. (2018). Partial data querying through racing algorithms. *International Journal of Approximate Reasoning, 96*, 36–55.

Peng, H., Sun, M., & Li, P. (2022). Optimal transport for long-tailed recognition with learnable cost matrix. In *International conference on learning representations*.

Priya, S., & Uthra, R. A. (2021). Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. *Complex & Intelligent Systems* 1–17.

Rabanser, S., Günnemann, S., & Lipton, Z. C. (2019). Failing loudly: An empirical study of methods for detecting dataset shift. In *Neural information processing systems* (pp. 1394–1406).

Read, J., & Žliobaitė, I. (2023). Learning from data streams: An overview and update. *SSRN*.

Ren, S., Liao, B., Zhu, W., Li, Z., Liu, W., & Li, K. (2018). The gradual resampling ensemble for mining imbalanced data streams with concept drift. *Neurocomputing, 286*, 150–166.

Ren, S., Zhu, W., Liao, B., Li, Z., Wang, P., Li, K., Chen, M., & Li, Z. (2019). Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning. *Knowledge-Based Systems, 163*, 705–722.

Roseberry, M., Krawczyk, B., & Cano, A. (2019). Multi-label punitive kNN with self-adjusting memory for drifting data streams. *ACM Transactions on Knowledge Discovery from Data, 13*(6), 1–31.

Sadeghi, F., & Viktor, H. L. (2021). Online-MC-Queue: Learning from imbalanced multi-class streams. In *International workshop on learning with imbalanced domains: Theory and applications* (pp. 21–34).

Santos, M. S., Abreu, P. H., Japkowicz, N., Fernández. A., Soares, C., Wilk, S., & Santos, J. (2022). On the joint-effect of class imbalance and overlap: a critical review. *Artificial Intelligence Review* (pp. 1–69).

Santos, M. S., Abreu, P. H., Japkowicz, N., Fernández, A., & Santos, J. (2023). A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research. *Information Fusion, 89*, 228–253.

Shah, Z., & Dunn, A. G. (2022). Event detection on twitter by mapping unexpected changes in streaming data into a spatiotemporal lattice. *IEEE Transactions on Big Data, 8*(2), 508–522.

Stefanowski, J. (2021). 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).

Sudharsan, B., Breslin, J. G., & Ali, M. I. (2021). Imbal-OL: Online machine learning from imbalanced data streams in real-world IoT. In *IEEE international conference on big data* (pp. 4974–4978).

Sun, Y., Li, M., Li, L., Shao, H., & Sun, Y. (2021). Cost-sensitive classification for evolving data streams with concept drift and class imbalance. *Computational Intelligence and Neuroscience* (2021).

Sun, Y., Sun, Y., & Dai, H. (2020). Two-stage cost-sensitive learning for data streams with concept drift and class imbalance. *IEEE Access, 8*, 191942–191955.

Sun, Y., Tang, K., Minku, L. L., Wang, S., & Yao, X. (2016). Online ensemble learning of data streams with gradually evolved classes. *IEEE Transactions on Knowledge and Data Engineering, 28*(6), 1532–1545.

Vafaie, P., Viktor, H., & Michalowski, W. (2020). Multi-class imbalanced semi-supervised learning from streams through online ensembles. In *International conference on data mining workshops* (pp. 867–874).

Vaquet, V., & Hammer, B. (2020). Balanced SAM-kNN: Online learning with heterogeneous drift and imbalanced data. In *International conference on artificial neural networks* (pp. 850–862).

Vuttipittayamongkol, P., Elyan, E., & Petrovski, A. (2021). On the class overlap problem in imbalanced data classification. *Knowledge Based Systems, 212*, 106631.

Wang, B., & Pineau, J. (2016). Online bagging and boosting for imbalanced data streams. *IEEE Transactions on Knowledge and Data Engineering, 28*(12), 3353–3366.

Wang, L., Yan, Y., & Guo, W. (2021). Ensemble online weighted sequential extreme learning machine for class imbalanced data streams. In *International symposium on computer engineering and intelligent communications* (pp. 81–86).

Wang, S., & Minku, L. L. (2020). AUC estimation and concept drift detection for imbalanced data streams with multiple classes. In *International joint conference on neural networks* (pp. 1–8).

Wang, S., Minku, L. L., & Yao, X. (2015). Resampling-based ensemble methods for online class imbalance learning. *IEEE Transactions on Knowledge and Data Engineering, 27*(5), 1356–1368.

Wang, S., Minku, L. L., & Yao, X. (2016). Dealing with multiple classes in online class imbalance learning. In *International joint conference on artificial intelligence* (pp. 2118–2124).

Wang, S., Minku, L. L., & Yao, X. (2018). A systematic study of online class imbalance learning with concept drift. *IEEE Transactions on Neural Networks and Learning Systems, 29*(10), 4802–4821.

Wang, S., Minku, L. L., Chawla, N., & Yao, X. (2019). Learning from data streams and class imbalance.
Wang, T., Jin, X., Ding, X., & Ye, X. (2014). User interests imbalance exploration in social recommendation: A fitness adaptation. In *ACM international conference on conference on information and knowledge management* (pp. 281–290).
Wares, S., Isaacs, J., & Elyan, E. (2019). Data stream mining: methods and challenges for handling concept drift. *SN Applied Sciences, 1*, 1–19.
Wasikowski, M., & Chen, X. (2010). Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1388–1400.
Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery, 30*(4), 964–994.
Wu K, Edwards A, Fan W, Gao J, Zhang K (2014) Classifying imbalanced data streams via dynamic feature group weighting with importance sampling. In *SIAM international conference on data mining* (pp. 722–730).
Yan, Y., Yang, T., Yang, Y., & Chen, J. (2017). A framework of online learning with imbalanced streaming data. In *AAAI conference on artificial intelligence* (Vol. 31).
Yan, Z., Hongle, D., Gang, K., Lin, Z., & Chen, Y. C. (2022). Dynamic weighted selective ensemble learning algorithm for imbalanced data streams. *The Journal of Supercomputing, 78*(4), 5394–5419.
Yang, L., Jiang, H., Song, Q., & Guo, J. (2022). A survey on long-tailed visual recognition. *International Journal of Computer Vision, 130*(7), 1837–1872.
Zhang, H., Liu, W., & Liu, Q. (2022). Reinforcement online active learning ensemble for drifting imbalanced data streams. *IEEE Transactions on Knowledge and Data Engineering, 34*(8), 3971–3983.
Zhang, H., Liu, W., Wang, S., Shan, J., & Liu, Q. (2019). Resample-based ensemble framework for drifting imbalanced data streams. *IEEE Access, 7*, 65103–65115.
Zhao, Y., Chen, W., Tan, X., Huang, K., & Zhu, J. (2022). Adaptive logit adjustment loss for long-tailed visual recognition. In *AAAI conference on artificial intelligence* (pp. 3472–3480).
Zhu, R., Guo, Y., & Xue, J. H. (2020). Adjusting the imbalance ratio by the dimensionality of imbalanced data. *Pattern Recognition Letters, 133*, 217–223.
Zhu, Z., Xing, H., & Xu, Y. (2022). Easy balanced mixing for long-tailed data. *Knowledge-Based Systems, 248*, 108816.
Žliobaitė, I., Bifet, A., Pfahringer, B., & Holmes, G. (2013). Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems, 25*(1), 27–39.
Zyblewski, P., Ksieniewicz, P., & Woźniak, M. (2019). Classifier selection for highly imbalanced data streams with minority driven ensemble. In *International conference on artificial intelligence and soft computing* (pp. 626–635).
Zyblewski, P., Sabourin, R., & Woźniak, M. (2021). Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams. *Information Fusion, 66*, 138–154.
Zyblewski, P., & Woźniak, M. (2021). Dynamic ensemble selection for imbalanced data stream classification with limited label access. In *International conference on artificial intelligence and soft computing* (pp. 217–226).

Authors and Affiliations

Gabriel Aguiar[1] · Bartosz Krawczyk[2] · Alberto Cano[3]

✉ Alberto Cano
acano@vcu.edu

Gabriel Aguiar
aguiargj@vcu.edu

Bartosz Krawczyk
bkrawczyk@vcu.edu

[1] Department of Computer Science, Virginia Commonwealth University, 401 W. Main St. ERB2334, Richmond, VA 23284, USA

[2] Department of Computer Science, Virginia Commonwealth University, 401 W. Main St. ERB2316, Richmond, VA 23284, USA

[3] Department of Computer Science, Virginia Commonwealth University, 401 W. Main St. ERB2314, Richmond, VA 23284, USA