

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

Gabriel Aguiar · Bartosz Krawczyk · Alberto Cano\*

**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 on how to evaluate these algorithms. This work presents a taxonomy of algorithms for imbalanced data streams and proposes a standardized, exhaustive, and informative experimental testbed 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 the largest experimental study conducted so far 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 testbed is fully reproducible and easy to extend with new methods. This way we propose first standardized approach to conducting experiments in imbalanced data streams that can be used by other researchers to create 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

## 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 traditional 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) 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.

---

G. Aguiar  
 Department of Computer Science, Virginia Commonwealth University, 401 W. Main St. ERB2334, Richmond, VA 23284  
 E-mail: aguiargj@vcu.edu

B. Krawczyk  
 Department of Computer Science, Virginia Commonwealth University, 401 W. Main St. ERB2316, Richmond, VA 23284  
 E-mail: bkrawczyk@vcu.edu

\* A. Cano (Corresponding author)  
 Department of Computer Science, Virginia Commonwealth University, 401 W. Main St. ERB2314, Richmond, VA 23284  
 E-mail: acano@vcu.edu

Another vital challenge in data stream mining lies in the need for a learning algorithm to display robustness to class imbalance (Krawczyk, 2016; Fernández et al., 2018). 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 and Krawczyk, 2020; Bernardo et al., 2020b; Bernardo and Della Valle, 2021a), algorithm adaptation mechanism (Loezer et al., 2020; Lu et al., 2020), or ensembles (Zyblewski et al., 2021; Cano and Krawczyk, 2022). 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 propose a taxonomy of imbalanced data streams algorithms, create a standardized, exhaustive, and informative experimental setup, and conduct a comparison of state-of-the-art classifiers.

**Motivation.** While there are a plethora of algorithms designed for drifting and imbalanced streams in the literature, there is a lack of standardized and agreed-upon procedures on how to evaluate these algorithms. Existing studies are either limited to a single family of algorithms or do not provide insights into what aspects of imbalanced data streams must be considered and translated into benchmark problems. An experimental analysis of oversampling in drifting streams was presented by Bernardo and Della Valle (2022), focusing on C-SMOTE and how it can benefit selected classifiers. In Brzeziński et al. (2021) authors examined the interactions between the global class imbalance ratio, local data difficulty factors, and concept drift, using five classifiers. While these works are an important first step, there is a need for a unified and holistic study of learning from imbalanced data streams that could be used as a template for researchers to evaluate their newly proposed algorithms.

**Overview and contributions.** In this paper, we propose the first complete and holistic framework for benchmarking and evaluating classifiers for imbalanced data streams. We summarize existing works and propose an elastic taxonomy of algorithms 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 provide a multi-level taxonomy that summarizes recent progress in learning from imbalanced data streams and provides a survey of the most important contributions.
- **Holistic and reproducible evaluation framework.** We propose the first complete and holistic framework for evaluating classifiers for 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 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. This creates the largest experimental study conducted in the data stream mining domain.
- **Recommendations and open challenges.** Based on the exhaustive experimental study, we formulate recommendations for end-users that will allow to understand strong and weak sides 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.

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

## 2 Data streams

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

### 2.1 Data stream characteristics

The main characteristics of data streams can be summarized as follows (Gama, 2010; 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  $\langle s_1, s_2, s_3, \dots, s_\infty \rangle$ . We consider a supervised scenario  $s_i = (X, y)$ , where  $X = [x_1, x_2, \dots, x_f]$  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 possible approaches to tackle related challenges.

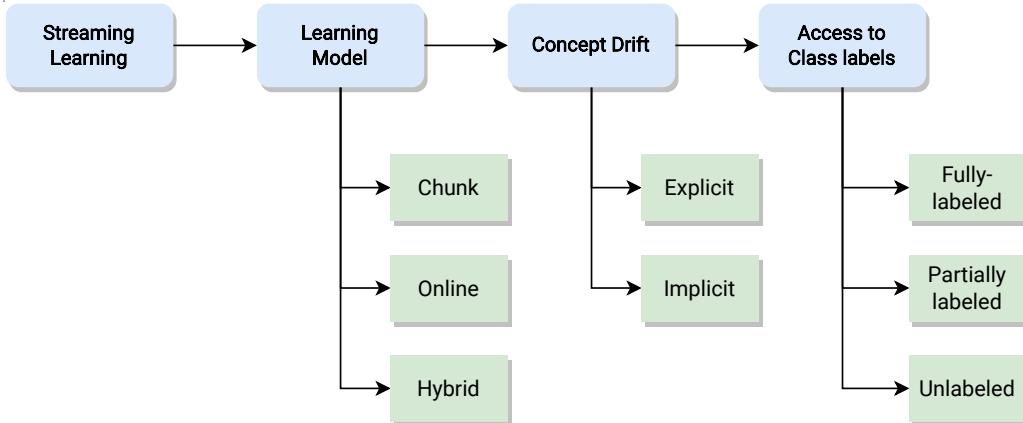


Fig. 1: Streaming learning taxonomy.

## 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 maintain 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):

- **Explicit.** Here drift adaptation is managed by an external tool, called drift detector (Barros and 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 and 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 streams

Class imbalance is one of the most vital problems in contemporary machine learning (Fernández et al., 2018). 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. Many real-world applications suffer from this problem, including fault diagnosis (Zhu and Song, 2010), anomaly detection (Khreich et al., 2010), medical diagnosis (Mazurowski et al., 2008), among others.

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 and 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 ( $P(y | x)$ ) will change over time (Wang and 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., 2018). Most algorithms dedicated to imbalanced data streams do not perform as well on balanced problems as their canonical

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

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 distributions for continual deep learning (Chrysakis and Moens, 2020; Kim et al., 2020; Arya and Hanumat Sastry, 2022; Priya and Uthra, 2021).

Figure 2 presents a taxonomy of characteristics of data streams algorithms. The specific details are discussed in the following subsections.

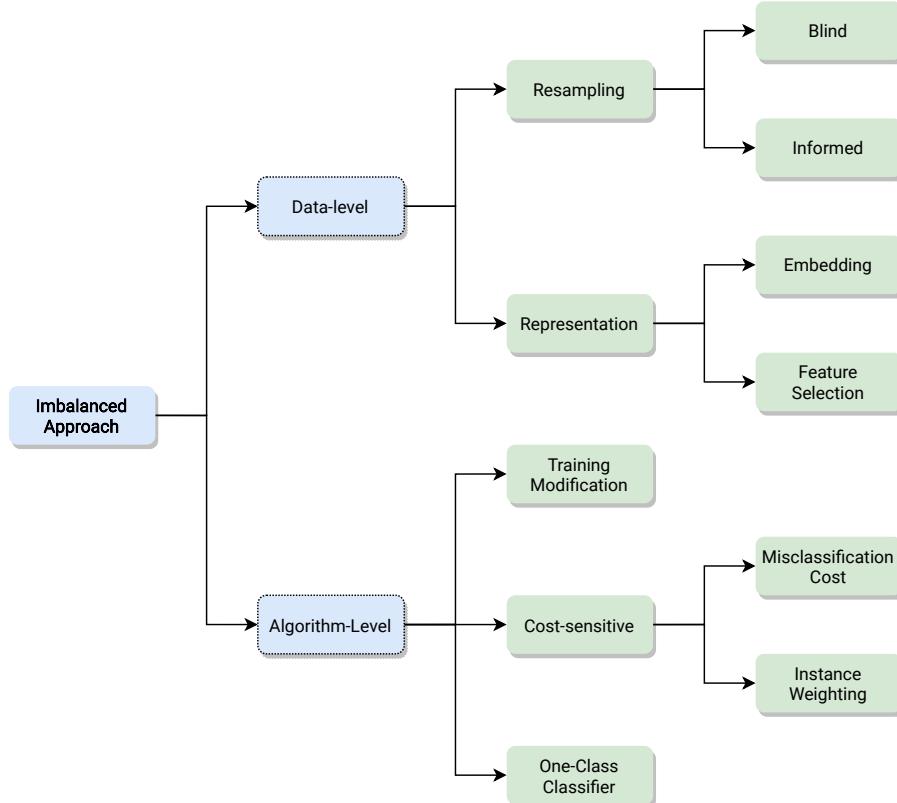


Fig. 2: Taxonomy of approaches for imbalanced data streams.

### 3.1 Data-level approaches

While resampling techniques are very popular for static imbalanced problems (Fernández et al., 2018; 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., 2018).

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., 2018). Those versions focus on keeping track of changes in the stream by employing either adaptive windows (Korycki and Krawczyk, 2020) or data sketches (Bernardo and Della Valle, 2021a,b). 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 and 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 and 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 and Krawczyk, 2020; Sadeghi and Viktor, 2021) or by applying binarization schemes and pairwise resampling (Mohammed et al., 2020a). Active learning techniques such as 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.

### 3.2 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). 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 and Hammer, 2020; Roseberry et al., 2019; Abolfazli and Ntoutsi, 2020). Genetic programming has been successfully used for induction of robust classifiers from the stream (Jedrzejowicz and Jedrzejowicz, 2020), as well as increasing skew-insensitive rule interpretability and recovery speed from concept drift (Cano and Krawczyk, 2019).

Cost-sensitive methods have been applied to streaming decision trees. Krawczyk and Skryjomska (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., 2021).

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 and Woźniak, 2015). One can use adaptive online one-class Support Vector Machines to track minority classes and their changes over time (Klikowski and 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 modelling (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 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 and Stefanowski, 2018; Grzyb et al., 2021; Du et al., 2021).

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

- **Backbone.** This defines how base classifiers are trained. For homogeneous ensembles the most popular streaming algorithms are based on either bagging or boosting. It is worthwhile noting that there exist alternative approaches, such as using projections or heterogeneous ensembles.
- **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 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 and Pineau, 2016). Similar strategies can be applied to Adaptive Random Forest (Gomes et al., 2017), Online Boosting (Klikowski and Woźniak, 2019; Gomes et al., 2019), Dynamic Weighted Majority (Lu et al.,

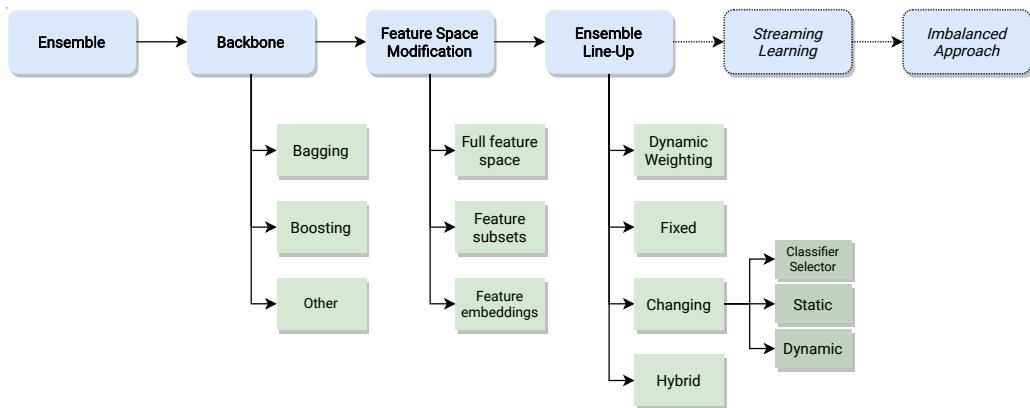


Fig. 3: Taxonomy of ensemble definition for imbalanced data streams.

2017), Dynamic Feature Selection (Wu et al., 2014), Adaptive Random Forest with resampling (Ferreira et al., 2019), Kappa Updated Ensemble (Cano and Krawczyk, 2020), Robust Online Self-Adjusting Ensemble (Cano and Krawczyk, 2022) or any ensemble that can incrementally update its base learners (Ancy and 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 and 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., 2021) 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., 2020) 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., 2021) 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., 2020) 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.

## 5 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 vs 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. The experimental study in Section 6 is divided into the following experiments whereas Section 7 discusses the lessons learned and recommendations.

#### Index of experiments

- 6.1. Binary class experiments
  - 6.1.1 Static imbalance ratio
  - 6.1.2 Dynamic imbalance ratio
  - 6.1.3 Instance-level difficulties
  - 6.1.4 Concept drift and static imbalance ratio
  - 6.1.5 Concept drift and dynamic imbalance ratio
  - 6.1.6 Real-world binary class imbalanced datasets
- 6.2. Multi-class experiments
  - 6.2.1 Static imbalance ratio
  - 6.2.2 Dynamic imbalance ratio
  - 6.2.3 Concept drift and static imbalance ratio
  - 6.2.4 Concept drift and dynamic imbalance ratio
  - 6.2.5 Impact of the number of classes
  - 6.2.6 Real-world multi-class imbalanced datasets
  - 6.2.7 Semi-synthetic multi-class imbalanced datasets
- 6.3. Overall comparison

### 5.1 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 1 with their characteristics according to the proposed taxonomy. Specific properties of the ensemble models are presented in Table 2. 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<sup>1</sup>. All results, interactive plots and tables are available on the website<sup>2</sup>. Algorithms were run on a cluster with 192 Intel Xeon cores, 6 TB RAM, and Centos 7. All algorithms use the parameter settings recommended by their authors. Detailed information about the specific parameters configuration is available on the GitHub repository.

### 5.2 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 (Brzeziński et al., 2021) were used. Those generators are presented in Table 3, 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).

### 5.3 Performance evaluation

In our experiments, 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). The metrics for imbalanced data streams include Kappa, G-Mean, and the Area Under the Curve (AUC). They are calculated over a sliding window of 500 instances.

<sup>1</sup> Source code and experiments are available at <https://github.com/canoalberto/imbalanced-streams>

<sup>2</sup> Interactive plots and tables for all experiments are available at <https://people.vcu.edu/~acano/imbalanced-streams>

Table 1: Data stream algorithms and their taxonomy.

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

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

Table 2: Ensemble algorithms and their taxonomy.

Ensemble	Backbone	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
ORUB	Boosting	Feature subsets	Fixed
OUOB	Bagging	Feature subsets	Fixed

Table 3: 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)	✓ Tree
SEA	3	2	✓ Function
Sine	3	2	✓ Function
Text	100	2	✗

Kappa is used to evaluate classifiers in imbalanced settings (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.

$$\text{Kappa} = \frac{n \sum_{i=1}^c x_{ii} - \sum_{i=1}^c x_{i\cdot}x_{\cdot i}}{n^2 - \sum_{i=1}^c x_{i\cdot}x_{\cdot i}} \cdot 100 \quad (1)$$

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  $x_{\cdot i}$ ,  $x_{i\cdot}$  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.

To tackle a balance between the performance of classifiers on the majority and minority classes, many researchers consider sensitivity and specificity (Brzeziński and 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. 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.

$$\text{Sensitivity} = \text{Recall} = \frac{TP}{TP + FN} \quad (2) \qquad \qquad \text{Specificity} = \frac{TN}{TN + FP} \quad (3)$$

$$G - \text{Mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \quad (4)$$

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 and Minku, 2020) for multi-class problems defining the Prequential Multi-Class (PMAUC) as Eq. 5.

$$\text{PMAUC} = \frac{1}{C(C-1)} \cdot \sum_{i \neq j} A(i | j) \quad (5)$$

where  $A(i | j)$  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 and Minku, 2020).

Both AUC and G-Mean are blind regarding the level of the class imbalance, while Kappa takes into account the class distribution. Therefore, in cases of extreme imbalance ratios, the Kappa metric can be very dissimilar but more informative than 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. Therefore, it is important to evaluate the algorithms using both metrics in order 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. Results for all metrics (Accuracy, Kappa, G-Mean, AUC, PMAUC, WMAUC, and EWMAUC) are available on the website.

## 6 Results

This section presents the experimental results from the set of benchmarks proposed to answer the research questions. Section 6.1 shows the experiments on binary class imbalanced streams. Section 6.2 shows the experiments on multi-class imbalanced streams. Finally, Section 6.3 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<sup>3</sup>.

### 6.1 Binary class experiments

The first set of experiments focuses on binary class problems with a positive minority class and a majority negative 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.

#### 6.1.1 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 3 with static imbalance ratios 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 4 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 maintains good performance on majority class but neglects the minority class. 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.

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

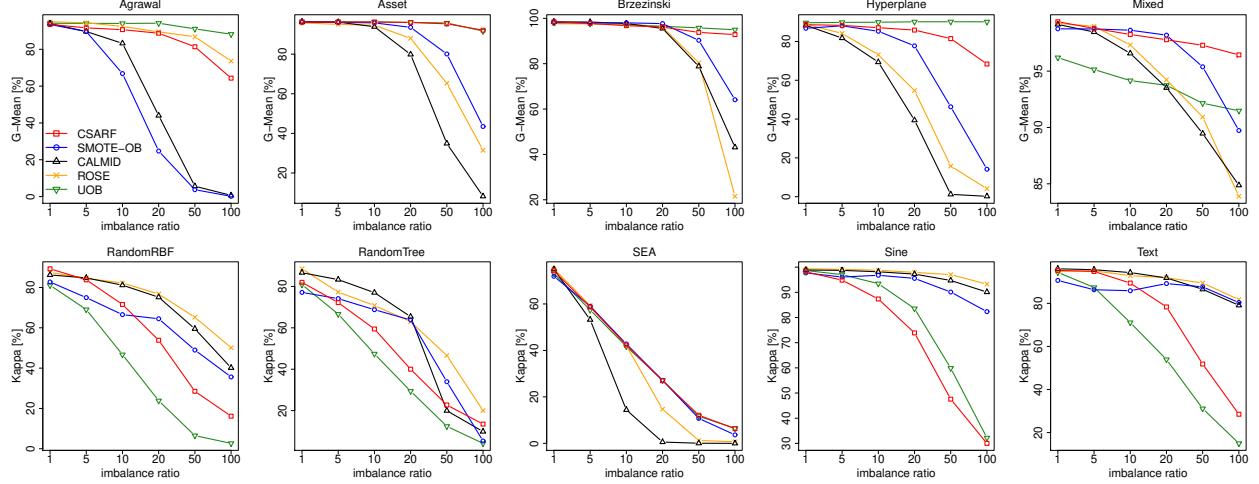


Fig. 4: Robustness to different levels of static class imbalance ratio (G-Mean and Kappa).

Table 4: 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	94.55	94.54	93.84	<b>95.09</b>	94.73	94.89	94.60	93.30	94.16	93.84
	<b>94.12</b>	86.68	89.10	92.10	91.99	93.02	93.63	93.44	93.91	93.06
	<b>93.50</b>	73.12	76.99	80.32	83.81	89.88	88.72	90.52	92.92	92.14
	<b>92.27</b>	57.57	67.47	70.14	69.60	78.75	79.08	83.66	90.14	91.21
	88.93	39.47	50.87	48.04	48.72	65.76	47.87	68.15	74.08	<b>89.23</b>
	83.23	29.86	37.64	36.35	36.89	47.76	13.40	45.29	60.74	<b>86.61</b>
Kappa	89.15	89.13	88.25	<b>90.24</b>	89.53	89.84	89.25	86.75	88.38	87.74
	82.16	79.26	81.56	<b>84.79</b>	84.41	84.52	84.28	80.82	83.68	78.58
	70.96	65.81	70.11	74.63	75.63	<b>78.26</b>	73.12	72.65	77.28	65.40
	55.30	52.63	60.54	63.86	62.11	66.14	57.24	62.55	<b>69.70</b>	49.32
	31.90	36.09	47.50	44.77	44.12	<b>56.91</b>	28.02	47.63	<b>56.91</b>	26.81
	17.96	28.09	35.56	34.46	34.36	42.49	5.31	35.31	<b>46.78</b>	13.66
Avg. G-Mean	<b>91.10</b>	63.54	69.32	70.34	70.95	78.34	69.55	79.06	84.33	91.01
Avg. Kappa	57.91	58.50	63.92	65.46	65.03	69.69	56.20	64.28	<b>70.46</b>	53.59
Rank G-Mean	<b>3.03</b>	8.13	8.06	5.64	6.38	5.02	5.88	5.28	3.51	4.07
Rank Kappa	6.38	6.52	6.40	4.08	4.80	4.03	6.20	6.08	<b>3.72</b>	6.78

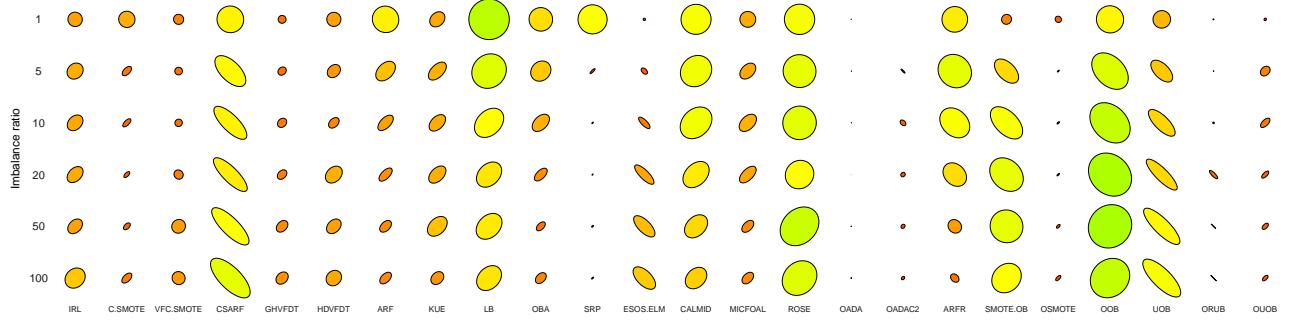


Fig. 5: Comparison of all 24 algorithms for different levels of static class imbalance ratio. The axes of the ellipse represent G-Mean and Kappa metrics. The bigger the axes the better rank of the algorithm on the metrics. The more rounded the ellipse the more agreement between the metrics. The color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

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. CSARF is not capable of performing exceedingly well on minority class and its good performance according to G-mean is boosted by the majority class. 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 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. 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 base classifiers in the ensemble pool.

#### 6.1.2 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 6,7,8,9 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 imbalance ratio. To increase readability, the line plots were smoothed using a moving average of 20 data points. Table 5 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.

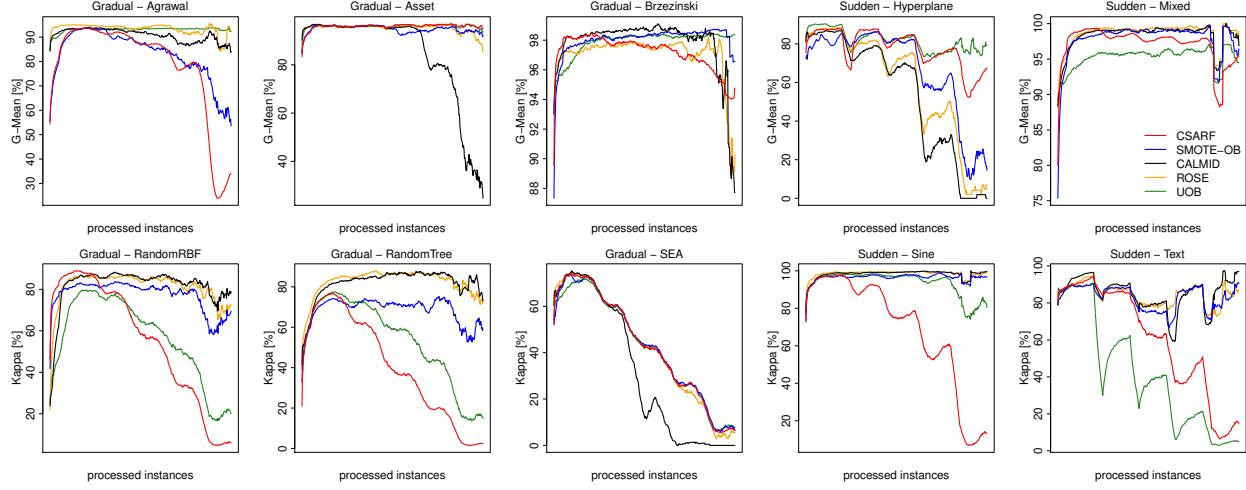


Fig. 6: G-Mean and Kappa on increasing class imbalance ratio with gradual and sudden drift.

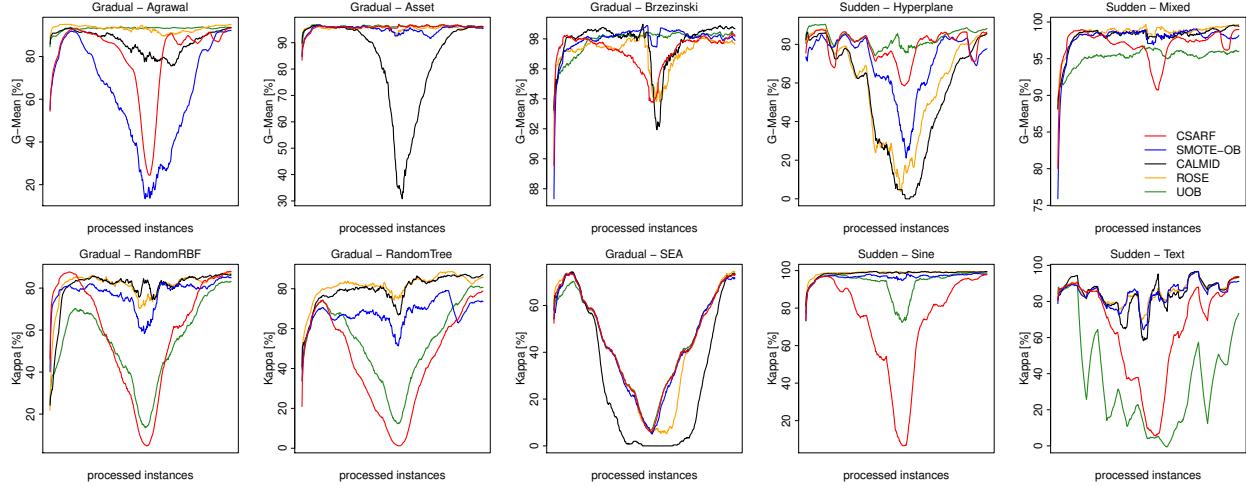


Fig. 7: G-Mean and Kappa on increasing decreasing class imbalance ratio with gradual and sudden drift.

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.

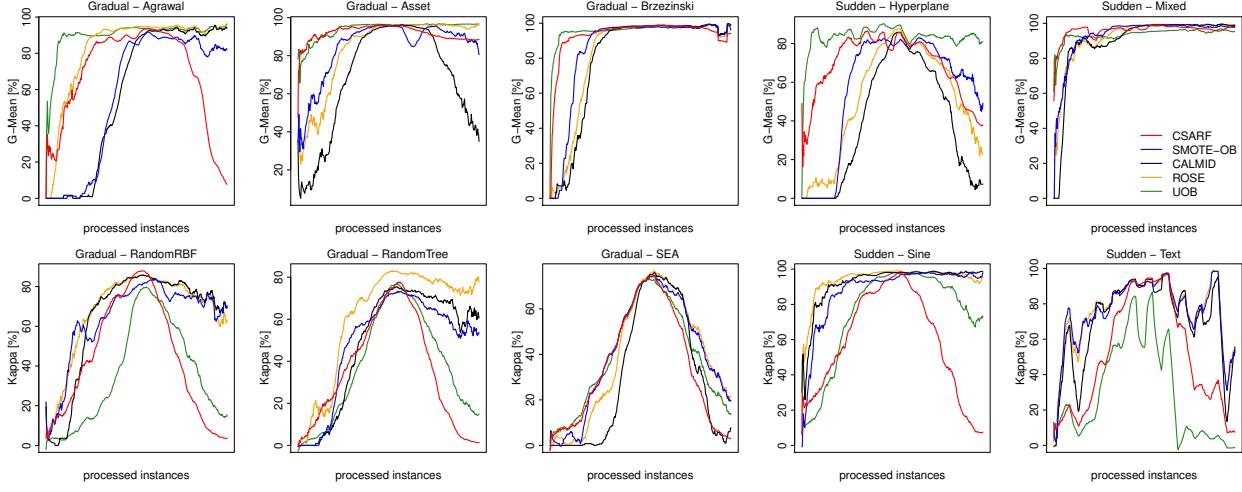


Fig. 8: G-Mean and Kappa on flipping class imbalance ratio with gradual and sudden drift.

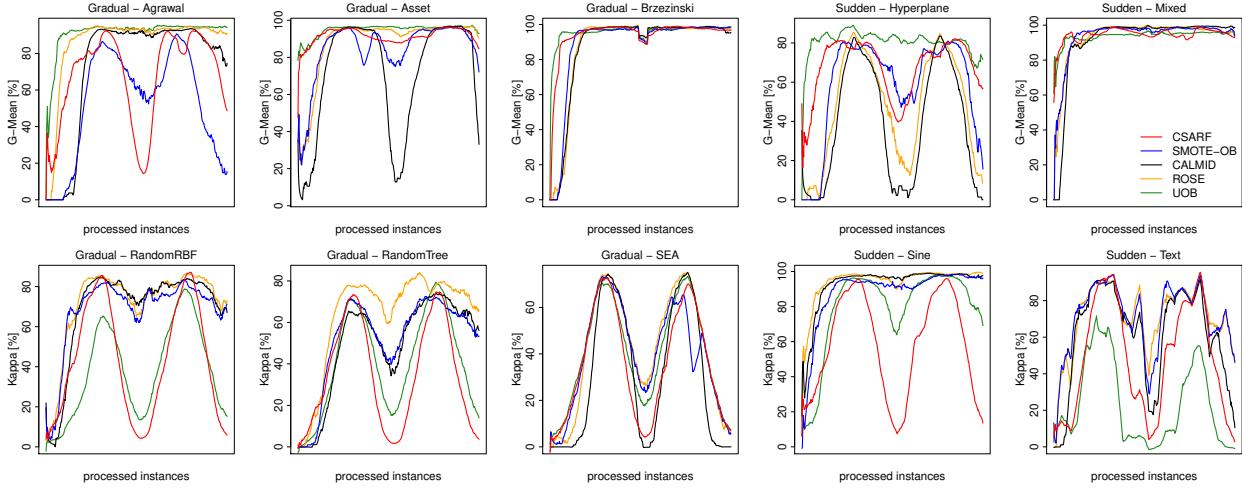


Fig. 9: G-Mean and Kappa on flipping and reflipping class imbalance ratio with gradual and sudden drift.

*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.

Table 5: 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	89.25	80.97	82.97	83.32	82.84	88.31	85.80	88.39	<b>90.94</b>
	Gradual	Inc. then dec.	89.57	76.24	81.30	82.00	81.51	87.79	83.24	85.65	89.00
	Flipping	84.58	64.27	70.57	69.58	67.65	78.25	65.74	75.91	79.90	<b>90.10</b>
	Reflipping	84.30	64.68	74.71	72.24	70.34	80.54	65.07	75.36	82.21	<b>87.70</b>
G-Mean	Sudden	Increasing	89.21	81.37	83.55	82.61	82.28	88.37	84.33	85.71	90.77
	Sudden	Inc. then dec.	89.92	77.14	82.13	83.05	82.66	88.24	83.43	86.64	89.30
	Flipping	84.73	65.44	72.44	71.78	69.29	79.99	66.05	77.53	80.41	<b>87.67</b>
	Reflipping	84.67	66.27	74.97	74.72	71.38	81.62	66.07	77.32	82.20	<b>87.37</b>
Kappa	Gradual	Increasing	54.14	71.56	71.87	73.07	73.77	<b>74.36</b>	73.83	71.50	68.49
	Gradual	Inc. then dec.	59.23	67.53	71.48	73.11	73.02	<b>75.65</b>	71.66	69.95	69.98
	Flipping	45.14	56.93	61.33	61.45	59.71	<b>66.68</b>	50.74	61.68	61.57	51.62
	Reflipping	43.50	56.82	61.89	62.58	61.45	<b>67.85</b>	52.93	60.74	60.80	52.39
Kappa	Sudden	Increasing	54.57	72.63	73.20	73.77	73.94	<b>75.01</b>	73.87	69.98	68.33
	Sudden	Inc. then dec.	59.97	68.84	72.42	74.67	74.55	<b>76.37</b>	72.51	71.49	69.89
	Flipping	45.15	58.56	63.74	63.86	61.82	<b>69.12</b>	51.46	63.29	62.09	50.59
	Reflipping	43.76	58.43	62.27	64.81	62.98	<b>69.09</b>	54.02	62.64	60.72	51.43
Avg. G-Mean		87.03	72.05	77.83	77.41	75.99	84.14	74.96	81.56	85.59	<b>89.01</b>
Avg. Kappa		50.68	63.91	67.27	68.42	67.65	<b>71.77</b>	62.63	66.41	65.23	55.79
Rank G-Mean		3.81	8.60	7.26	5.81	6.79	3.94	6.73	5.09	3.88	<b>3.09</b>
Rank Kappa		8.50	6.26	5.32	4.53	4.80	<b>2.66</b>	6.33	5.00	4.54	7.06

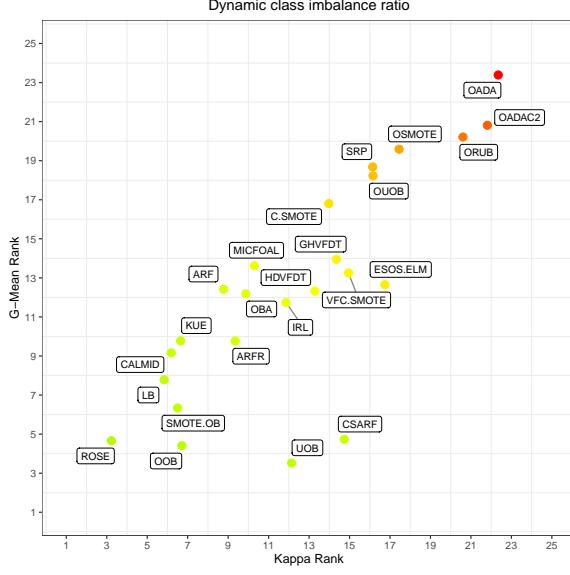


Fig. 10: Comparison of all 24 algorithms for dynamic class imbalance ratio. Color gradient represents the product of both metrics.

### 6.1.3 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%} and in the combined scenario they represent {0%, 20%, 40%}. 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 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.

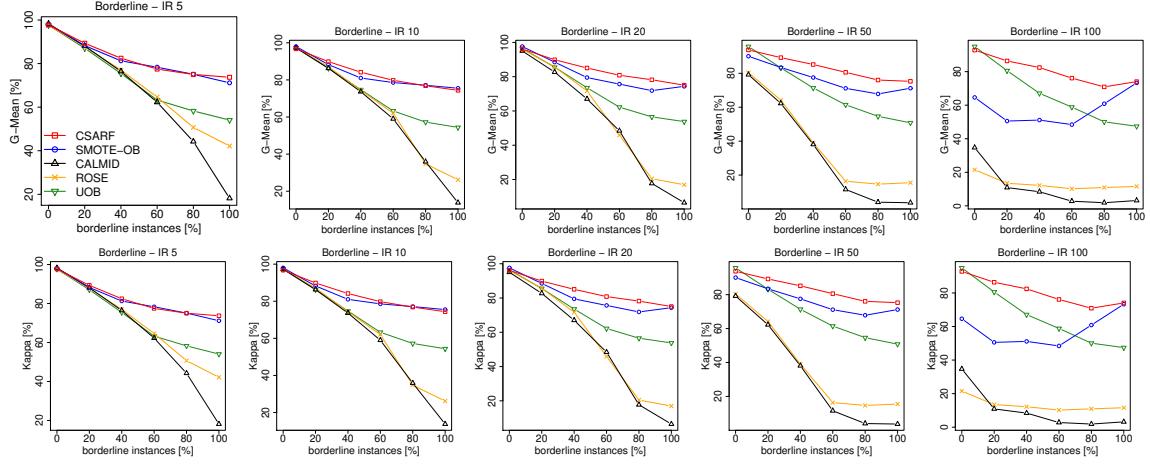


Fig. 11: Robustness to borderline instances for static imbalance ratio (G-Mean and Kappa).

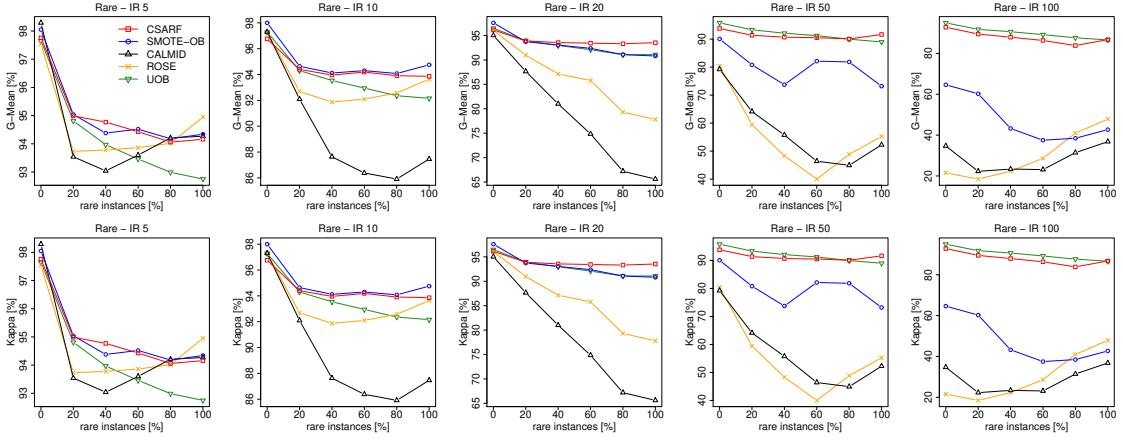


Fig. 12: Robustness to rare instances for static imbalance ratio (G-Mean and Kappa).

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 6 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 Figures 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, Figure 21 illustrates the performance of the classifiers with increasing imbalance ratio. Table 7 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, Figure 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 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

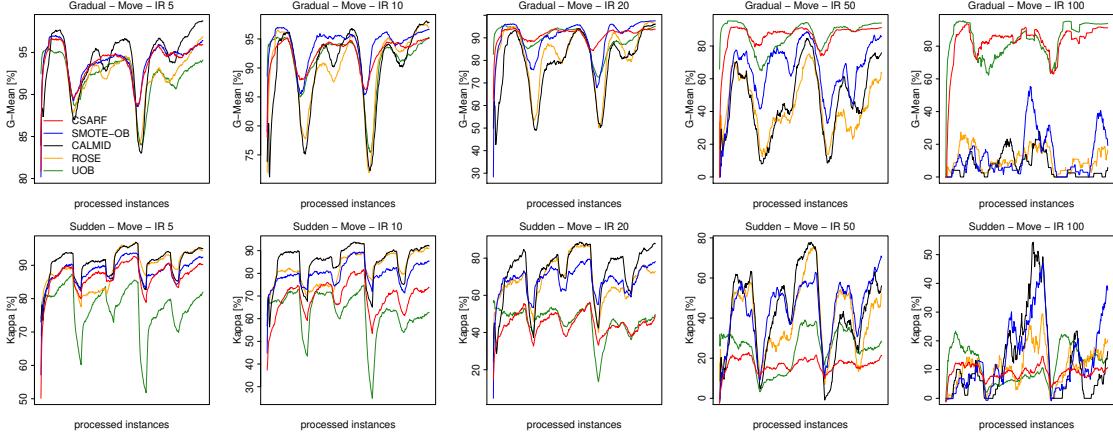


Fig. 13: G-Mean and Kappa on moving minority clusters for static imbalance ratio.

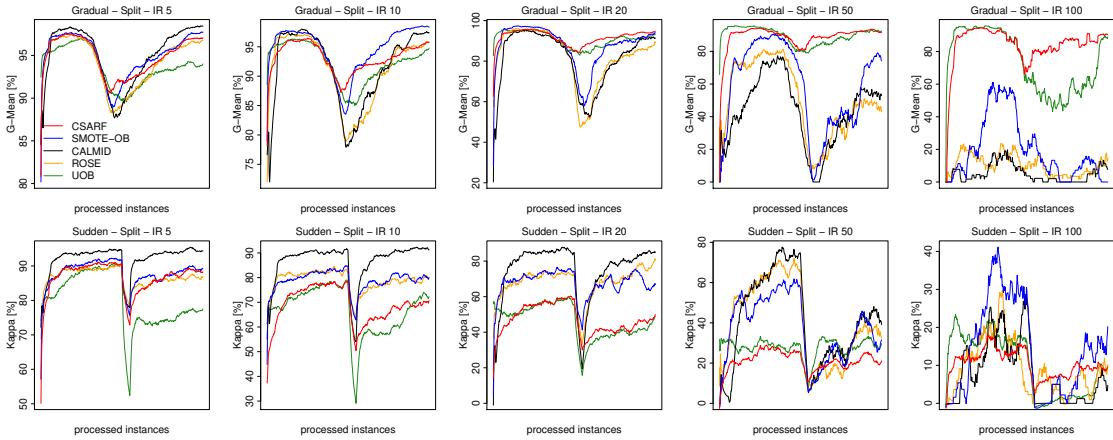


Fig. 14: G-Mean and Kappa on splitting minority clusters for static imbalance ratio.

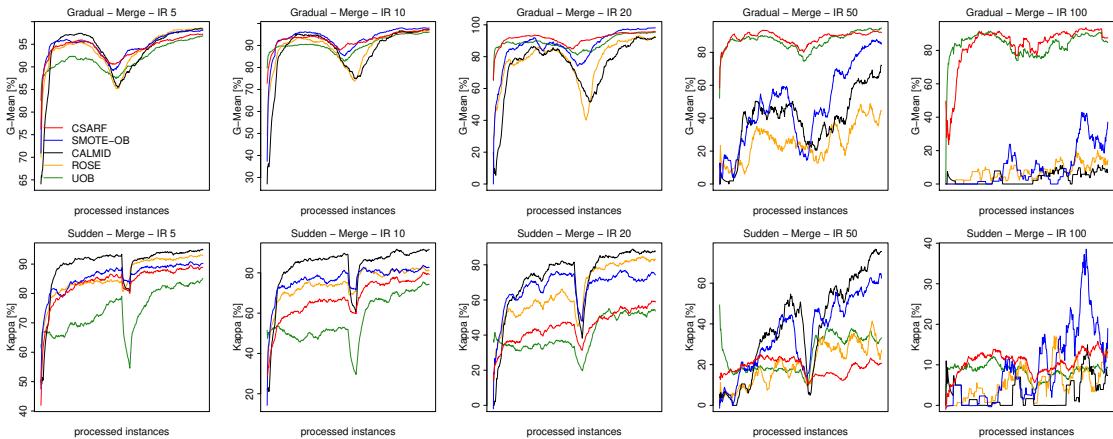


Fig. 15: G-Mean and Kappa on merging minority clusters for static imbalance ratio.

Table 6: 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	Borderline	95.03	93.03	92.95	93.97	94.49	94.64	<b>95.29</b>	95.08	95.04	94.27
	Rare	<b>82.63</b>	61.15	59.89	61.72	64.57	69.97	74.17	82.03	67.87	72.63
	Moving	95.44	94.78	92.27	95.05	95.71	94.91	<b>96.32</b>	95.70	93.62	92.58
	Splitting	95.96	95.11	93.64	95.44	96.34	95.46	<b>96.89</b>	96.20	95.39	93.96
	Merging	95.40	94.07	91.45	94.59	95.42	95.23	<b>96.39</b>	95.87	94.49	92.55
IR 100	Borderline	87.86	16.24	17.88	23.91	28.61	29.96	14.16	47.80	63.97	<b>90.14</b>
	Rare	<b>80.46</b>	2.26	3.68	7.34	10.30	13.32	9.96	58.15	38.69	66.50
	Moving	<b>84.88</b>	0.06	4.64	3.83	6.91	14.61	10.33	18.37	52.10	84.29
	Splitting	<b>86.38</b>	0.84	4.78	2.92	9.35	13.15	11.78	24.11	59.56	73.36
	Merging	<b>85.88</b>	0.73	4.86	3.53	3.36	8.60	6.35	11.81	52.89	84.39
Kappa	Borderline	80.22	84.17	82.82	84.13	<b>84.22</b>	80.23	81.58	79.87	80.79	77.32
	Rare	<b>70.17</b>	51.33	50.58	51.74	53.52	54.70	61.78	58.85	52.78	47.05
	Moving	82.62	<b>88.99</b>	84.56	88.35	88.78	83.25	86.96	83.76	80.03	75.92
	Splitting	84.40	<b>90.45</b>	86.20	89.13	90.12	84.83	88.75	85.80	84.05	79.61
	Merging	81.95	<b>88.97</b>	83.49	88.11	88.51	85.10	86.73	84.66	81.17	73.76
IR 100	Borderline	13.24	13.02	14.36	19.80	23.71	23.36	2.44	26.14	<b>37.55</b>	11.92
	Rare	13.58	1.70	2.76	5.59	8.34	8.94	2.96	<b>44.04</b>	25.01	4.97
	Moving	9.20	0.05	3.41	2.84	5.25	9.40	1.87	10.79	<b>31.16</b>	11.11
	Splitting	10.52	0.66	3.53	2.13	7.22	8.26	1.96	12.54	<b>37.17</b>	9.84
	Merging	10.85	0.55	3.64	2.63	2.44	5.52	1.33	8.43	<b>35.01</b>	8.62
All IRs	Avg. G-Mean	<b>88.40</b>	52.58	53.59	57.98	62.58	64.77	65.13	77.25	75.77	83.12
All IRs	Avg. Kappa	50.02	46.04	45.40	49.95	53.36	51.06	47.24	<b>56.77</b>	54.71	41.26
All IRs	Rank G-Mean	<b>2.07</b>	8.09	9.00	7.03	5.67	5.97	4.13	3.57	4.58	4.90
All IRs	Rank Kappa	5.32	5.88	7.18	4.50	<b>3.03</b>	5.34	6.00	4.39	4.82	8.54

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 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, resulting in poorer performance on the majority class. So, while it is capable of handling minority difficulties and clusters by cost penalty-lead training, it ignores the majority instances 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.

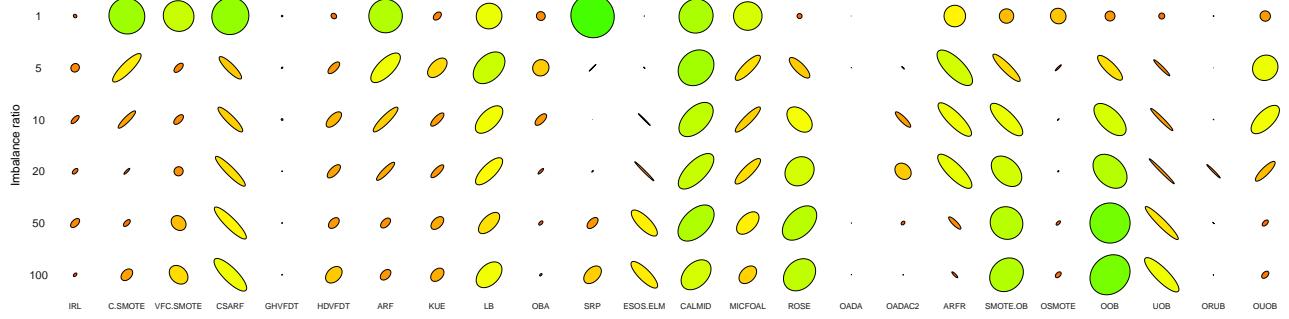


Fig. 16: Comparison of all 24 algorithms for borderline instances on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics.

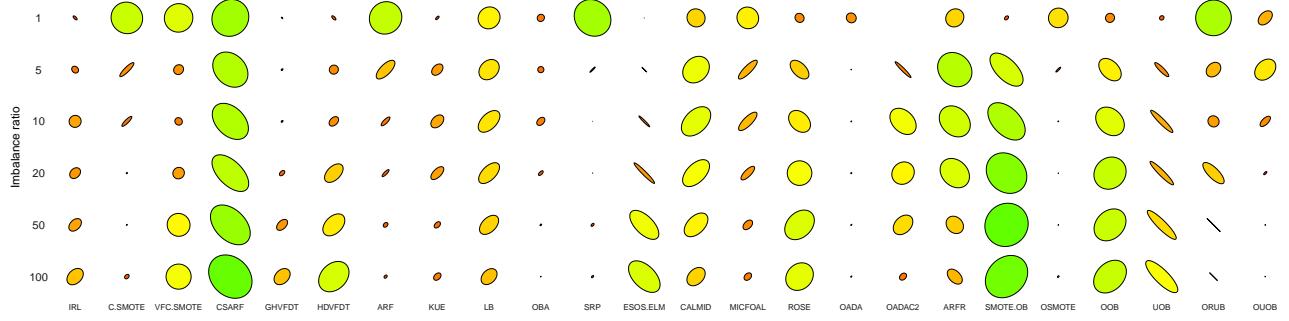


Fig. 17: Comparison of all 24 algorithms for rare instances on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics.

*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

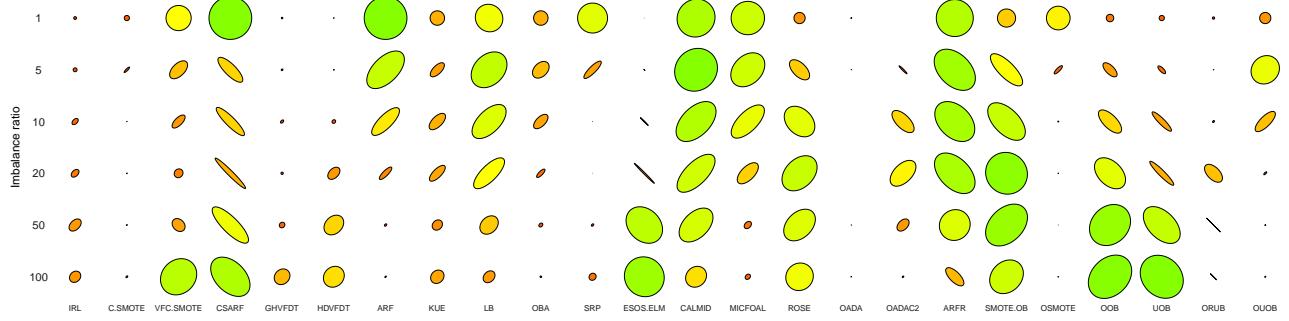


Fig. 18: Comparison of all 24 algorithms for moving minority clusters on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics.

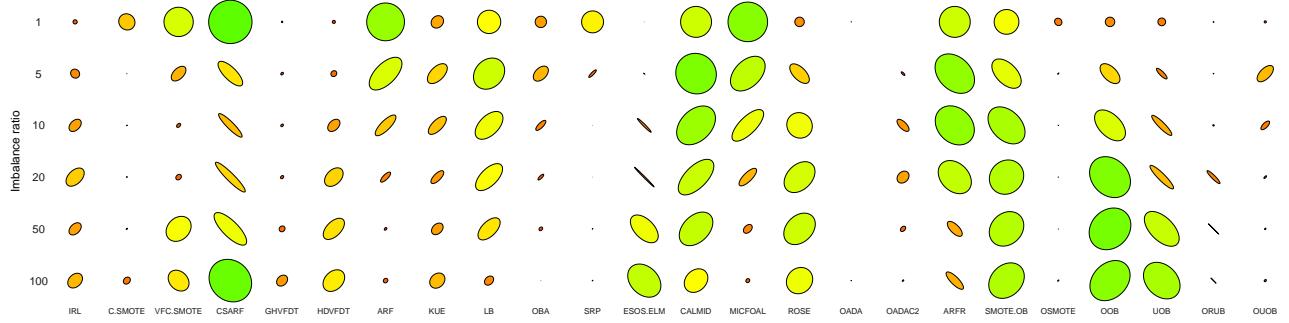


Fig. 19: Comparison of all 24 algorithms for splitting minority clusters on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics.

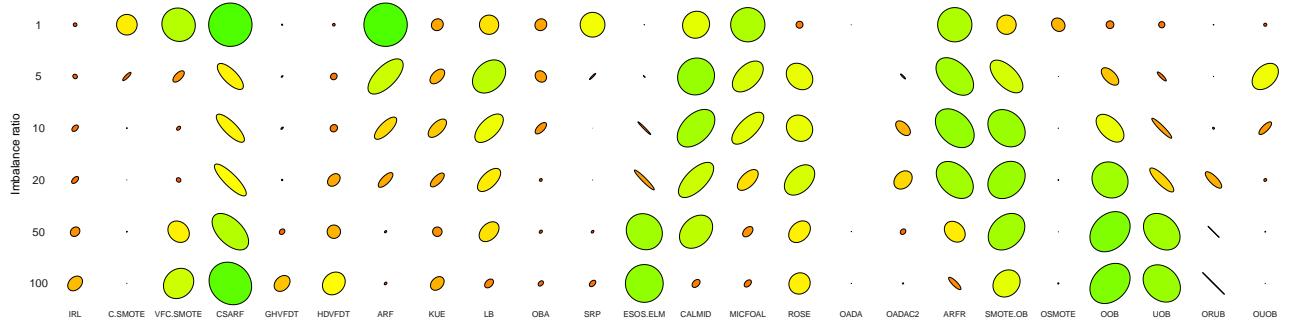


Fig. 20: Comparison of all 24 algorithms for merging minority clusters on static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics.

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

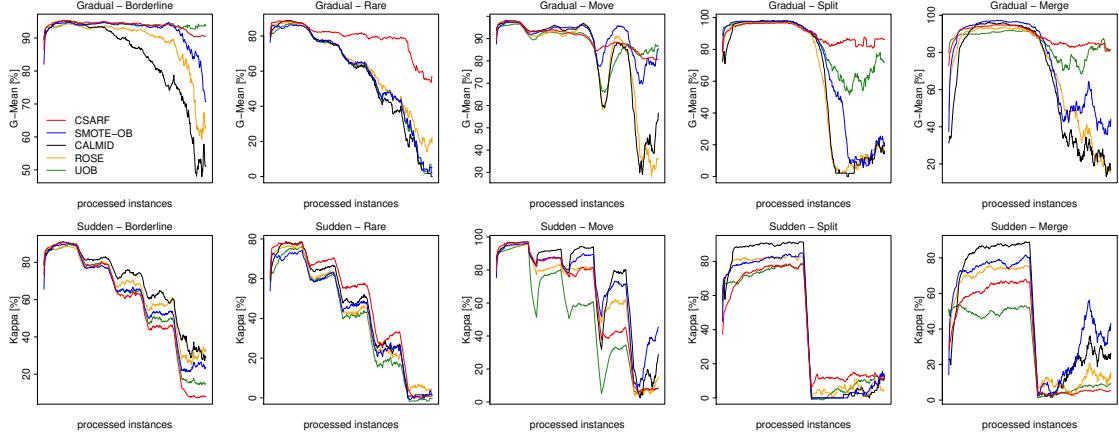


Fig. 21: G-Mean and Kappa on increasing borderline, rare, moving, splitting, and merging minority clusters and increasing imbalance ratio.

Table 7: 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	93.75	81.91	79.81	82.48	82.28	88.04	89.26	91.94	93.61	<b>94.04</b>
	Rare	<b>77.88</b>	55.38	54.95	55.98	55.34	60.43	55.99	56.80	56.56	56.46
	Moving	<b>92.13</b>	67.18	68.70	70.51	80.28	79.64	79.45	88.06	89.23	87.04
	Splitting	<b>90.12</b>	50.85	55.53	51.90	56.12	56.78	56.67	59.75	79.94	76.70
	Merging	<b>87.87</b>	51.69	49.17	60.78	61.81	63.24	58.79	69.90	80.73	82.20
Kappa	Borderline	57.48	68.50	67.37	67.51	<b>68.88</b>	62.91	65.32	61.89	61.76	59.39
	Rare	<b>47.56</b>	44.41	43.17	44.17	44.85	42.60	43.35	41.73	41.91	39.38
	Moving	56.32	60.79	59.86	63.82	<b>71.36</b>	65.08	69.75	69.32	63.37	51.44
	Splitting	42.38	46.36	47.71	46.61	49.06	45.05	46.12	45.65	<b>51.08</b>	42.30
	Merging	33.93	44.35	41.27	50.97	51.62	43.12	43.71	<b>51.94</b>	49.37	29.49
Avg. G-Mean		<b>87.76</b>	62.31	62.64	65.07	67.17	70.00	68.73	73.52	79.43	79.02
Avg. Kappa		47.82	53.12	52.21	54.52	<b>56.92</b>	51.48	53.54	53.67	53.03	45.31
Rank G-Mean		<b>1.25</b>	9.25	9.00	7.50	7.33	5.33	6.00	3.75	2.92	2.67
Rank Kappa		8.00	4.21	5.58	3.92	<b>1.63</b>	6.67	4.92	5.33	5.25	9.50

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 (from when sub-concepts existed) that are not able to generalize well over the current state of the stream.

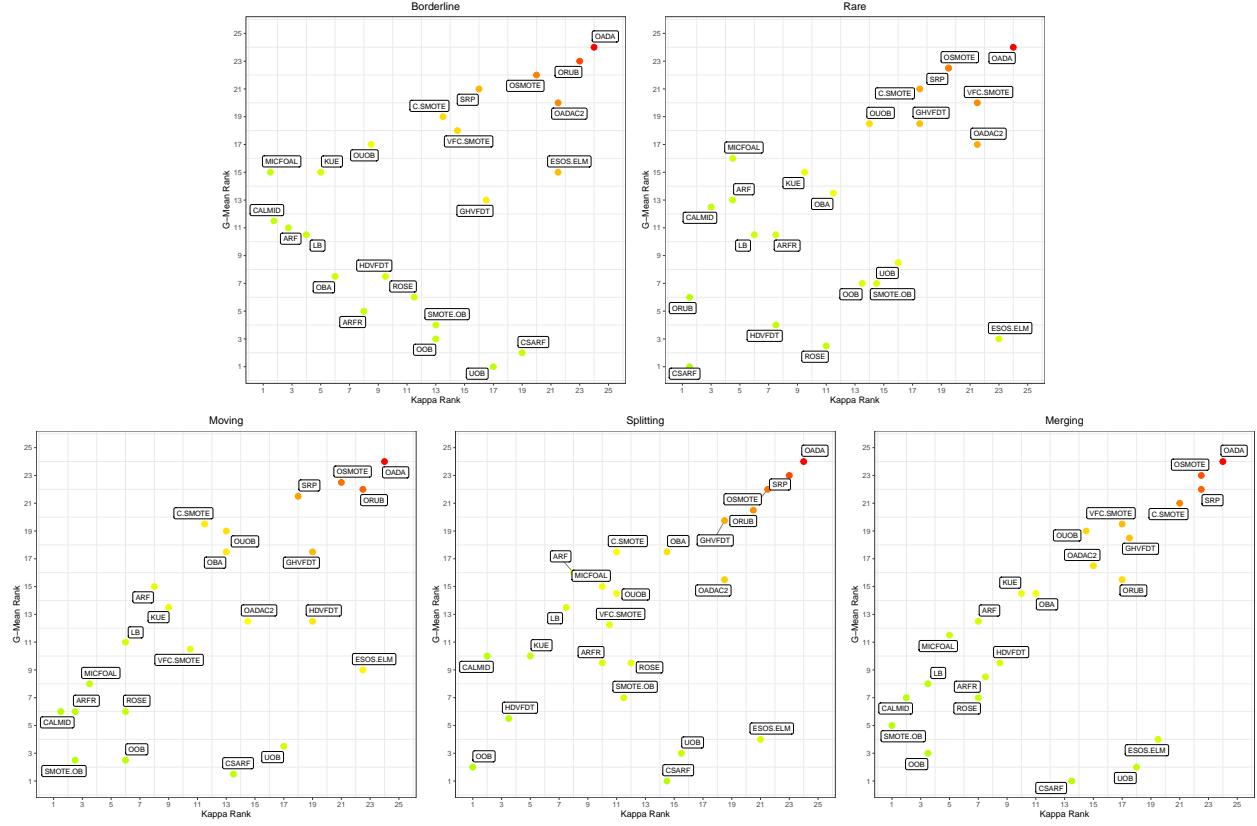


Fig. 22: Comparison of all 24 algorithms on borderline, rare, moving, splitting, merging minority clusters and increasing imbalance ratio. Color gradient represents the product of both metrics.

#### 6.1.4 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. 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 6.1.1 with two types of concept drift: gradual and sudden. They were combined with the static imbalance ratio examined in experiment 6.1.1. 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 8 presents the G-Mean and Kappa for the top 10 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

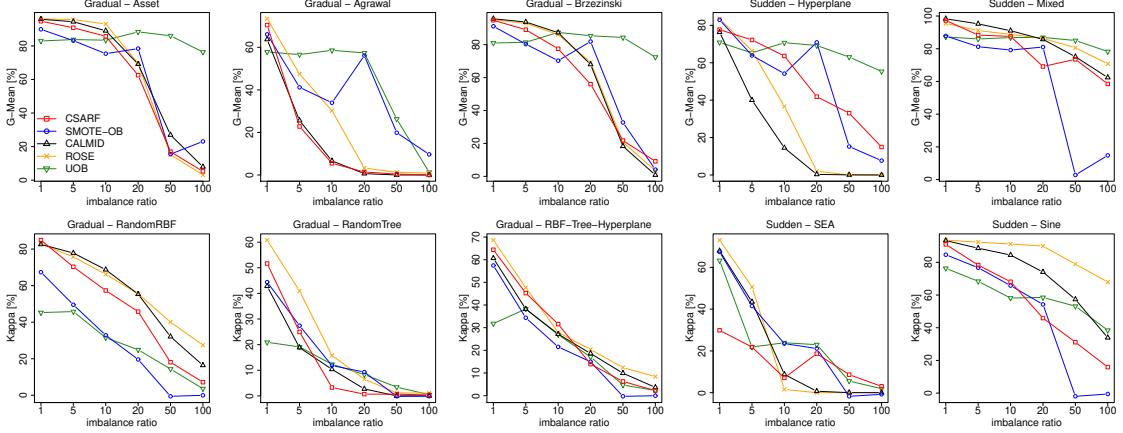


Fig. 23: Robustness to concept drift with static class imbalance ratio (G-Mean and Kappa).

Table 8: 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	87.82	87.83	85.41	88.09	86.26	<b>88.80</b>	87.71	83.74	78.88	78.14
	5	<b>86.60</b>	67.54	71.99	75.38	76.81	84.42	83.91	84.59	76.65	76.40
	10	<b>84.68</b>	46.25	52.40	54.30	61.52	76.16	75.55	80.35	71.21	74.91
	20	<b>80.95</b>	31.36	39.03	38.66	41.70	59.45	57.46	70.20	60.06	73.14
	50	<b>74.27</b>	15.35	20.83	20.74	23.41	37.09	28.66	42.83	36.66	69.27
	100	<b>64.80</b>	9.95	11.58	13.27	12.38	22.00	9.37	21.53	24.50	63.70
Kappa	1	75.74	75.77	71.67	76.32	72.63	<b>77.72</b>	75.58	69.74	58.04	56.68
	5	66.59	56.50	58.37	64.07	63.85	<b>70.46</b>	68.08	62.25	52.79	45.33
	10	53.21	39.05	42.63	45.81	50.00	<b>60.54</b>	54.37	53.21	45.66	33.72
	20	37.07	26.26	31.20	32.58	34.40	<b>45.81</b>	33.88	42.94	36.59	22.31
	50	19.38	12.63	17.11	17.47	18.95	<b>28.32</b>	13.41	26.48	22.17	10.16
	100	10.48	8.67	10.09	11.82	10.69	<b>17.23</b>	2.80	15.04	15.69	4.42
Avg. G-Mean		<b>79.85</b>	43.05	46.87	48.41	50.35	61.32	57.11	63.87	57.99	72.59
Avg. Kappa		43.74	36.48	38.51	41.35	41.75	<b>50.01</b>	41.35	44.94	38.49	28.77
Rank G-Mean		<b>1.87</b>	7.98	7.63	6.57	6.54	4.22	4.93	4.26	6.00	5.03
Rank Kappa		3.92	7.08	6.75	5.41	5.43	<b>3.05</b>	4.88	4.61	5.94	7.93

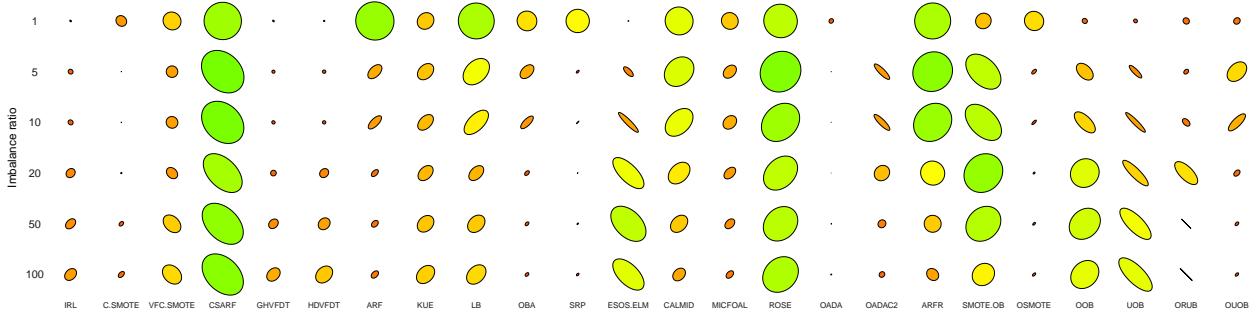


Fig. 24: Comparison of all 24 algorithms for concept drift with static class imbalance ratio. Axes of the ellipse represent G-Mean and Kappa metrics. Color gradient represents the product of both metrics.

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 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 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.

#### 6.1.5 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 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 6.1.4 with gradual and sudden concept drift, and combined them with the dynamic increasing imbalance ratio proposed in experiment 6.1.2. Figure 25 illustrates the performance of the selected classifiers with dynamic increasing imbalance ratio under the presence of concept drift. Table 9 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.

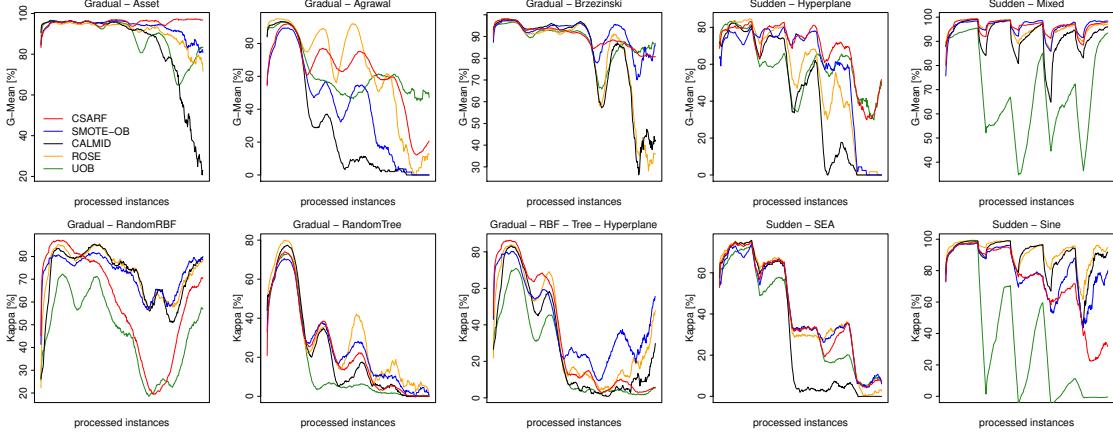


Fig. 25: G-Mean and Kappa on concept drift with increasing imbalance ratio.

Table 9: 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	<b>85.87</b>	55.63	61.44	62.58	63.46	73.78	60.97	75.75	72.64
	Gradual	<b>77.98</b>	45.68	53.66	47.72	53.52	65.12	50.47	70.41	65.88
Kappa	Sudden	55.02	48.85	51.04	54.98	55.05	<b>62.84</b>	52.95	57.46	48.44
	Gradual	41.63	36.38	40.74	38.89	42.09	<b>48.97</b>	39.66	45.73	39.03
Avg. G-Mean	<b>81.92</b>	50.66	57.55	55.15	58.49	69.45	55.72	73.08	69.26	70.66
Avg. Kappa	48.33	42.61	45.89	46.93	48.57	<b>55.90</b>	46.31	51.59	43.74	31.58
Rank G-Mean	<b>1.15</b>	9.25	7.20	7.60	6.45	4.10	7.45	3.20	4.45	4.15
Rank Kappa	4.70	8.00	6.30	5.80	4.80	<b>1.95</b>	5.60	2.95	5.70	9.20

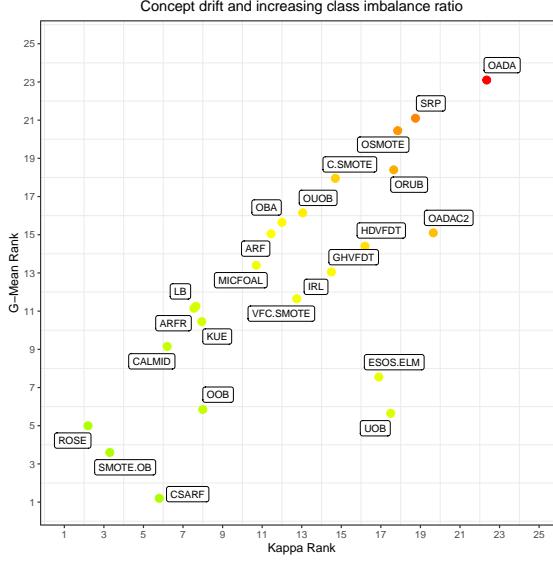


Fig. 26: Comparison of all 24 algorithms for concept drift with increasing class imbalance ratio. Color gradient represents the product of both metrics.

*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 can 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.

#### 6.1.6 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 for imbalanced data streams classifiers, and their specifications are presented at Table 10. Figure 27 illustrates the performance of the five selected classifiers in the real-world datasets. Table 11 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 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.

Table 10: Real-world binary datasets specifications.

Dataset	Instances	Features
adult	45,222	14
amazon	8,000	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

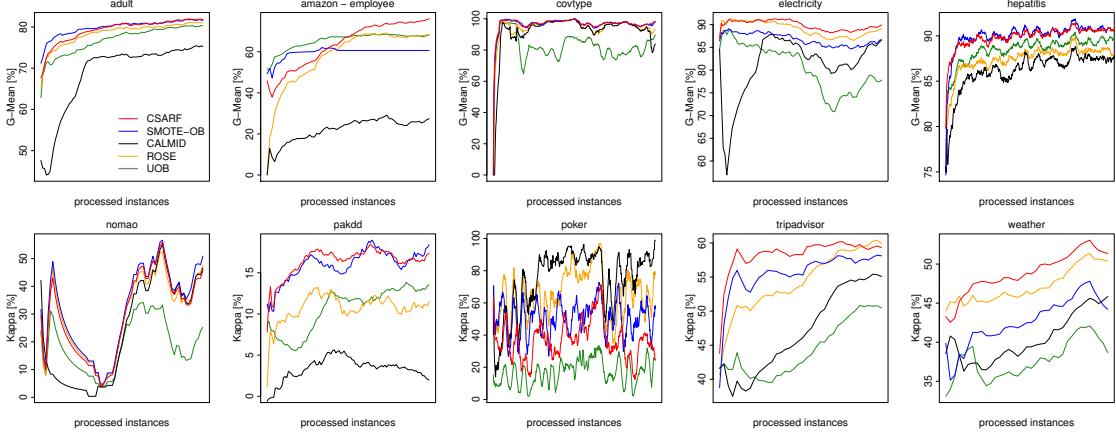


Fig. 27: G-Mean and Kappa on binary class imbalanced datasets.

Table 11: G-Mean and Kappa on binary class imbalanced datasets.

Dataset	CSARF	ARF	KUE	LB	CALMID	ROSE	ARFR	SMOTE-OB	OOB	UOB
adult	80.16	72.59	69.70	73.13	71.43	79.31	<b>80.50</b>	80.46	79.30	77.90
amazon	23.06	6.75	0.00	5.43	4.28	20.99	18.20	<b>24.09</b>	10.26	18.52
amazon-emp	<b>67.89</b>	51.74	33.65	24.49	37.13	62.89	50.62	59.58	56.84	66.96
census	<b>48.57</b>	27.66	21.71	28.92	31.11	39.61	41.83	44.23	35.19	44.86
coil2000	57.34	4.74	3.97	5.07	0.85	16.30	19.87	51.33	20.79	<b>60.92</b>
covtype	<b>97.23</b>	94.23	89.55	94.09	93.42	95.36	93.46	97.03	94.07	82.87
creditcard	32.25	27.36	18.55	25.84	26.64	29.19	27.41	28.37	28.90	<b>32.35</b>
electricity	<b>89.88</b>	89.38	70.68	88.65	82.96	88.97	89.43	86.68	81.65	79.81
gmsc	<b>76.59</b>	31.55	31.46	29.39	43.23	58.01	66.63	71.92	70.57	72.55
hepatitis	<b>90.51</b>	85.53	85.56	86.28	87.54	88.37	90.39	90.18	89.65	89.36
internet-ads	<b>13.90</b>	13.72	0.00	12.98	13.17	13.36	0.00	<b>13.90</b>	13.36	13.54
kddcup	7.71	7.25	4.26	7.41	7.64	7.67	7.32	7.44	7.80	<b>7.84</b>
nomao	43.27	37.76	28.14	34.95	34.39	39.75	35.23	41.60	40.38	<b>43.94</b>
pakdd	<b>59.28</b>	1.79	10.30	4.86	18.07	38.19	57.14	54.88	50.99	56.21
poker	<b>79.34</b>	45.02	55.52	56.02	79.27	74.14	50.43	62.26	73.70	72.47
spam	<b>79.60</b>	78.31	56.68	76.94	55.41	77.33	77.54	78.83	71.98	70.68
tripadvisor	<b>80.69</b>	74.30	68.33	73.02	71.64	77.63	79.31	80.40	75.50	72.54
twitter	<b>64.74</b>	18.24	18.77	18.20	19.60	47.29	56.81	64.37	59.15	63.39
weather	<b>76.55</b>	68.15	60.64	67.87	68.93	74.44	74.31	73.53	69.41	67.78
<hr/>										
adult	53.33	53.75	52.22	54.37	51.60	<b>55.41</b>	53.35	53.76	52.44	48.27
amazon	1.64	0.55	0.00	0.35	-0.07	<b>2.58</b>	0.71	1.93	1.38	1.86
amazon-emp	29.76	31.71	13.91	7.30	13.45	<b>33.30</b>	17.93	15.64	21.83	12.03
census	20.64	19.63	13.40	20.26	21.02	26.19	23.49	<b>26.84</b>	23.53	23.14
coil2000	12.94	1.24	0.65	1.41	-0.10	5.00	2.88	<b>13.32</b>	5.95	6.89
covtype	85.33	<b>91.15</b>	82.65	90.09	88.60	89.95	83.77	89.57	83.50	45.78
creditcard	17.49	26.42	18.04	24.96	25.87	<b>28.19</b>	22.14	27.45	27.65	7.92
electricity	<b>80.00</b>	79.97	50.55	78.46	68.94	78.36	79.65	73.93	65.75	62.19
gmsc	29.17	16.67	16.68	15.40	25.51	34.63	32.02	35.48	<b>35.92</b>	26.06
hepatitis	72.86	76.63	75.56	76.55	<b>77.36</b>	74.58	74.72	74.28	74.02	69.67
internet-ads	<b>13.53</b>	13.33	0.00	12.48	12.70	12.91	0.00	<b>13.53</b>	12.91	13.12
kddcup	7.03	7.14	3.58	7.30	<b>7.43</b>	7.36	7.03	7.17	7.35	6.34
nomao	32.96	32.95	17.23	29.93	28.57	33.01	28.72	<b>35.41</b>	27.10	20.68
pakdd	<b>16.85</b>	0.24	0.81	0.56	3.51	11.13	15.88	16.63	14.05	11.25
poker	36.81	37.67	46.45	49.51	<b>75.72</b>	65.72	30.39	50.73	53.94	17.27
spam	74.05	74.93	48.51	71.96	50.35	72.38	71.13	<b>75.10</b>	61.04	58.47
tripadvisor	<b>58.82</b>	58.25	49.80	54.88	50.33	56.70	58.29	56.79	49.92	46.23
twitter	<b>26.06</b>	5.56	3.43	5.29	5.78	19.92	23.54	24.40	23.22	22.14
weather	<b>49.77</b>	46.89	37.26	44.54	42.33	48.27	45.54	43.12	40.33	37.30
Avg. G-Mean	<b>61.50</b>	44.00	38.29	42.82	44.56	54.15	53.50	58.48	54.18	57.60
Avg. Kappa	37.84	35.51	27.93	33.98	34.15	<b>39.77</b>	35.33	38.69	35.89	28.24
Rank G-Mean	<b>1.45</b>	7.00	9.34	7.74	7.47	4.39	4.97	3.29	4.92	4.42
Rank Kappa	4.13	4.95	8.66	6.11	6.37	3.29	5.39	<b>3.24</b>	5.29	7.58

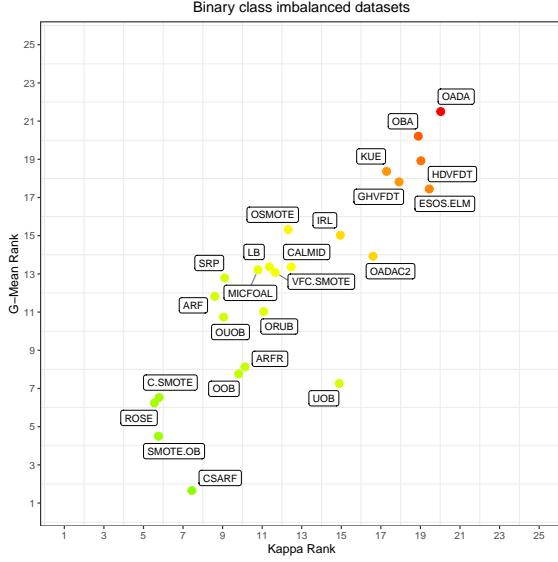


Fig. 28: Comparison of all 24 algorithms for binary class imbalanced datasets. Color gradient represents the product of both metrics.

## 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 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.

## 6.2 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 and 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 1.

### 6.2.1 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 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 12 summarizes the performance of the top 10 classifiers for each generator and their average ranking regarding each metric. For overall comparison, Figure 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, therefore losing performance on the majority classes. 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

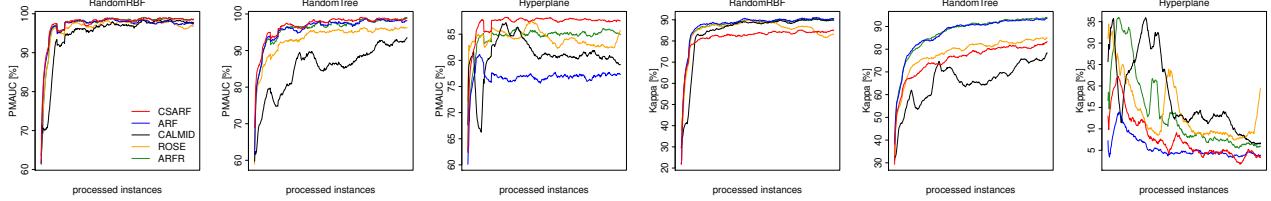


Fig. 29: PMAUC and Kappa on multi-class static imbalance ratio.

Table 12: PMAUC and Kappa on multi-class static imbalance ratio.

	Generator	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC	Hyperplane	<b>87.39</b>	77.00	81.94	76.97	76.93	81.07	75.29	84.26	84.80	87.24
	RandomRBF	<b>97.91</b>	97.40	94.11	96.54	97.39	95.53	96.66	96.77	97.71	94.84
	RandomTree	<b>97.21</b>	96.49	93.87	83.92	95.26	86.07	91.13	93.83	96.45	92.11
Kappa	Hyperplane	7.43	5.47	<b>20.03</b>	6.98	2.96	16.98	6.10	13.46	13.44	19.25
	RandomRBF	82.59	<b>88.55</b>	79.57	86.08	87.67	85.85	87.87	85.76	87.99	78.85
	RandomTree	76.55	88.18	78.74	55.55	69.66	66.83	80.80	79.25	<b>88.29</b>	68.48
Avg. PMAUC		<b>94.17</b>	90.30	89.97	85.81	89.86	87.56	87.69	91.62	92.99	91.40
Avg. Kappa		55.52	60.74	59.44	49.54	53.43	56.55	58.26	59.49	<b>63.24</b>	55.53
Rank PMAUC		<b>1.00</b>	4.00	6.67	8.33	5.67	7.67	8.00	5.00	2.67	6.00
Rank Kappa		6.67	4.00	5.00	7.33	7.00	6.00	4.67	5.00	<b>2.67</b>	6.67

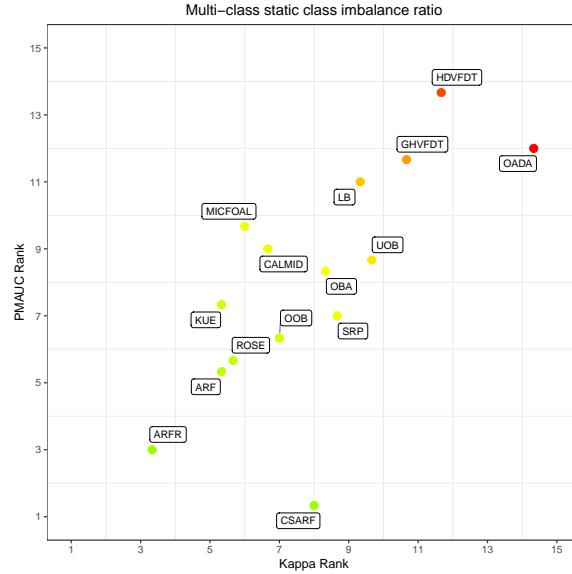


Fig. 30: Comparison of all 15 algorithms for multi-class static class imbalance ratio. Color gradient represents the product of both metrics.

a general-purpose ensemble, therefore it was expected not to display robustness on pair with dedicated skew-insensitive solutions. KUE and SRP could satisfactorily 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.

### 6.2.2 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 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 13 presents the performance for the top 10 classifiers, and their average ranking. To summarize, Figure 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 classifier, 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 (Figure 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.

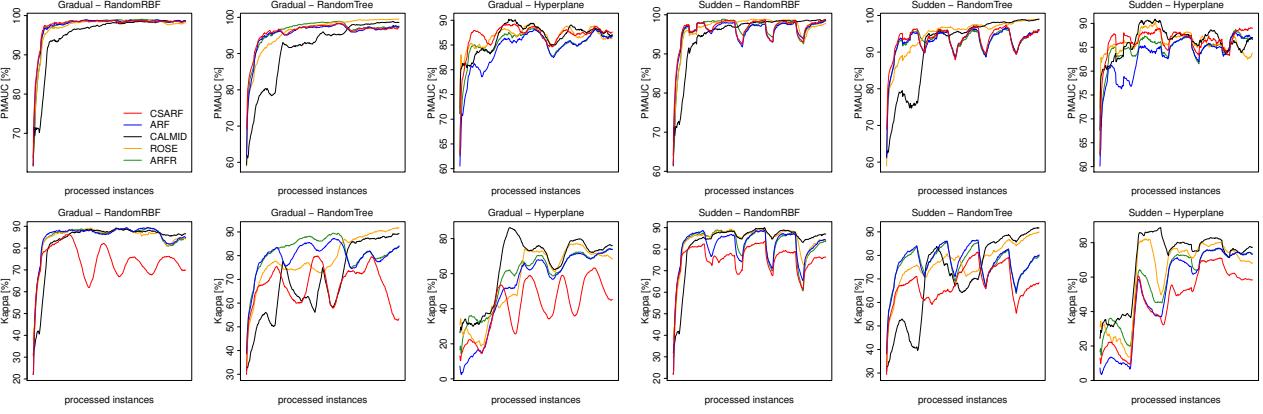


Fig. 31: PMAUC and Kappa on multi-class shifting imbalance ratio.

Table 13: PMAUC and Kappa on multi-class shifting imbalance ratio.

Drift	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB	
PMAUC	Sudden	92.28	90.95	92.27	<b>92.45</b>	90.41	91.36	89.55	91.87	91.30	92.33
	Gradual	<b>93.82</b>	92.82	92.08	92.50	91.89	91.91	91.07	93.69	93.24	92.90
Kappa	Sudden	63.45	70.98	72.82	73.05	65.81	<b>76.49</b>	69.54	73.21	71.87	67.28
	Gradual	63.64	73.30	70.69	69.86	68.41	74.79	70.72	74.39	<b>75.09</b>	69.35
Avg. PMAUC	<b>93.05</b>	91.88	92.18	92.47	91.15	91.64	90.31	92.78	92.27	92.62	
Avg. Kappa	63.55	72.14	71.76	71.45	67.11	<b>75.64</b>	70.13	73.80	73.48	68.31	
Rank PMAUC	<b>2.63</b>	5.50	5.25	4.67	7.33	7.08	8.13	4.08	5.08	5.25	
Rank Kappa	9.25	4.33	5.08	5.33	7.92	<b>3.25</b>	5.33	3.58	3.67	7.25	

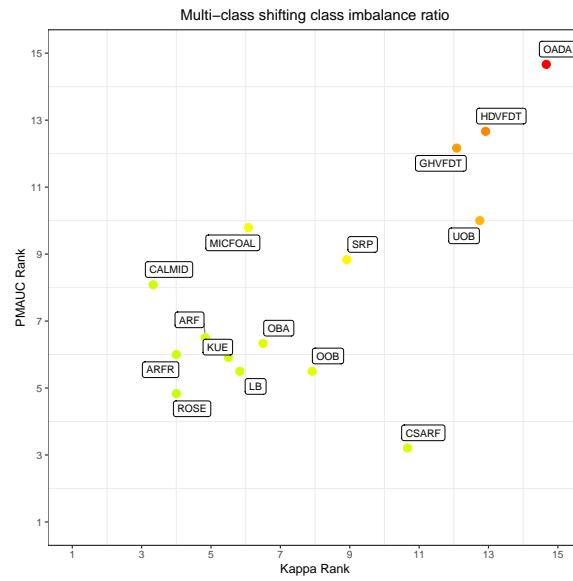


Fig. 32: Comparison of all 15 algorithms for multi-class shifting class imbalance ratio. Color gradient represents the product of both metrics.

### 6.2.3 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 6.2.1, 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 14 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

*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 exceptions to this rule are KUE, which as we observed in binary experiments cannot perform well in scenarios 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 ratio, both PMAUC and Kappa become sensitive to speed of changes.

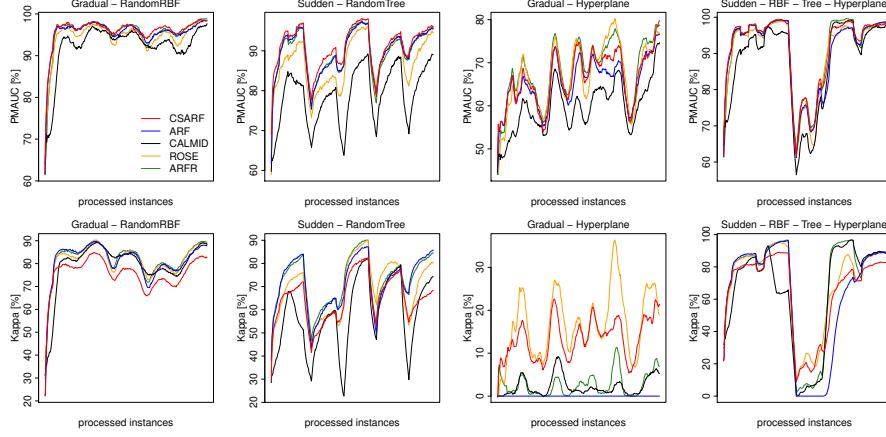


Fig. 33: PMAUC and Kappa on concept drift and multi-class static imbalance ratio.

Table 14: 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	<b>87.07</b>	86.21	77.82	82.54	84.81	80.11	81.13	85.73	86.97
	Gradual	84.02	83.39	75.27	80.86	82.40	78.20	78.59	83.11	<b>84.23</b>
Kappa	Sudden	56.29	54.47	47.60	49.85	50.04	51.83	49.66	<b>60.81</b>	58.16
	Gradual	49.02	46.38	40.20	46.35	42.39	46.71	42.88	<b>54.65</b>	50.35
Avg. PMAUC		85.55	84.80	76.55	81.70	83.61	79.16	79.86	84.42	<b>85.60</b>
Avg. Kappa		52.66	50.43	43.90	48.10	46.22	49.27	46.27	<b>57.73</b>	54.25
Rank PMAUC		<b>1.75</b>	3.13	9.00	6.13	4.38	8.25	7.50	4.00	2.00
Rank Kappa		4.38	5.63	7.88	6.13	6.88	5.13	6.63	<b>2.00</b>	2.25
										8.88
										8.13

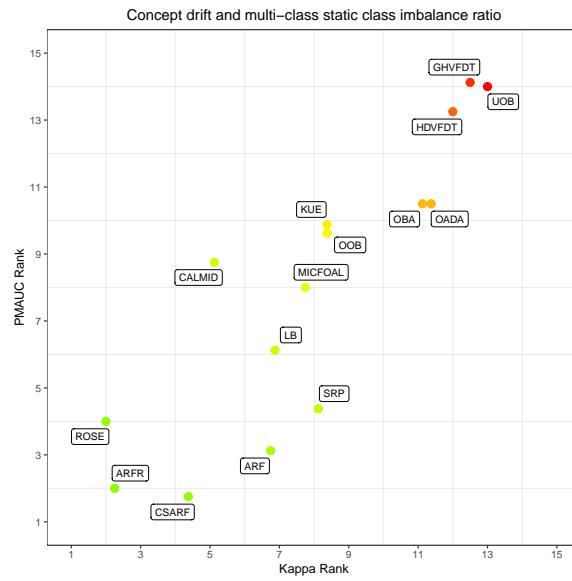


Fig. 34: Comparison of all 15 algorithms on concept drift and multi-class static class imbalance ratio. Color gradient represents the product of both metrics.

#### 6.2.4 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 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 6.2.2, 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 15 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 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 ARFR 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.

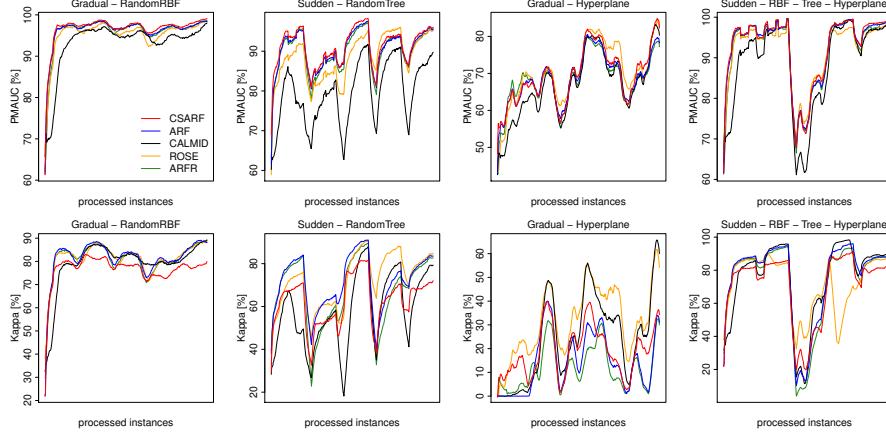


Fig. 35: PMAUC and Kappa on concept drift and multi-class shifting imbalance ratio.

Table 15: 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	80.68	80.75	76.58	77.53	81.27	76.56	77.56	<b>81.85</b>	80.54
	Gradual	80.81	81.49	79.91	79.12	<b>82.08</b>	78.23	79.31	81.99	81.38
Kappa	Sudden	46.26	51.72	46.60	47.78	49.01	47.84	48.87	<b>55.65</b>	50.12
	Gradual	43.42	49.17	46.02	46.78	48.93	45.80	47.14	<b>52.00</b>	48.19
Avg. PMAUC		80.75	81.12	78.25	78.32	81.67	77.39	78.43	<b>81.92</b>	80.96
Avg. Kappa		44.84	50.45	46.31	47.28	48.97	46.82	48.00	<b>53.82</b>	49.15
Rank PMAUC		3.65	3.58	7.09	6.21	<b>3.01</b>	7.48	7.13	3.63	3.95
Rank Kappa		7.33	3.74	6.56	5.10	4.28	5.29	5.59	<b>2.83</b>	4.85

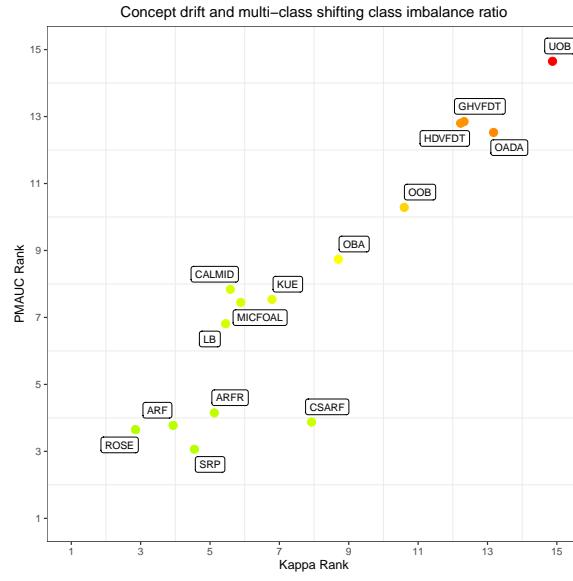


Fig. 36: Comparison of all 15 algorithms on concept drift and multi-class shifting class imbalance ratio. Color gradient represents the product of both metrics.

### 6.2.5 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 6.2.2. 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 16 summarizes the performance of the top 10 classifiers for each number of classes and their average ranking regarding each metric. For overall comparison, Figure 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.

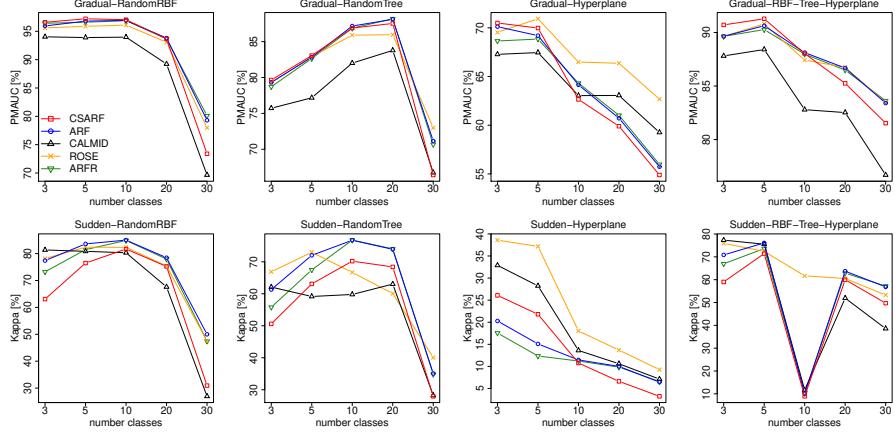


Fig. 37: Impact of the number of classes on PMAUC and Kappa under concept drift and multi-class shifting imbalance ratio.

Table 16: 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	<b>85.75</b>	84.82	80.04	83.17	84.19	81.18	81.49	84.01	84.26	78.35
	5	<b>86.66</b>	86.14	81.08	84.71	86.06	82.03	82.66	86.12	85.82	78.51
	10	81.16	81.58	77.79	80.34	81.99	78.10	78.28	<b>84.30</b>	81.60	75.40
	20	81.47	81.94	80.25	79.60	<b>82.93</b>	78.94	79.01	82.24	81.93	75.90
	30	68.68	71.10	72.08	63.80	<b>73.18</b>	66.72	70.73	72.92	71.18	66.30
Kappa	3	48.08	55.81	53.45	60.02	43.91	60.48	54.22	<b>62.44</b>	52.50	43.42
	5	55.34	58.49	53.22	58.97	54.38	58.75	54.77	<b>64.06</b>	56.03	44.21
	10	46.55	49.82	42.85	47.43	50.19	44.78	45.31	<b>55.86</b>	49.61	34.66
	20	49.60	53.95	48.20	47.74	<b>56.12</b>	46.70	48.89	51.57	53.64	36.88
	30	24.63	34.16	33.82	22.25	<b>40.24</b>	23.40	36.84	35.19	33.99	22.16
Avg. PMAUC		80.75	81.12	78.25	78.32	81.67	77.39	78.43	<b>81.92</b>	80.96	74.89
Avg. Kappa		44.84	50.45	46.31	47.28	48.97	46.82	48.00	<b>53.82</b>	49.15	36.27
Rank PMAUC		3.65	3.58	7.09	6.21	<b>3.01</b>	7.48	7.13	3.63	3.95	9.29
Rank Kappa		7.33	3.74	6.56	5.10	4.28	5.29	5.59	<b>2.83</b>	4.85	9.45

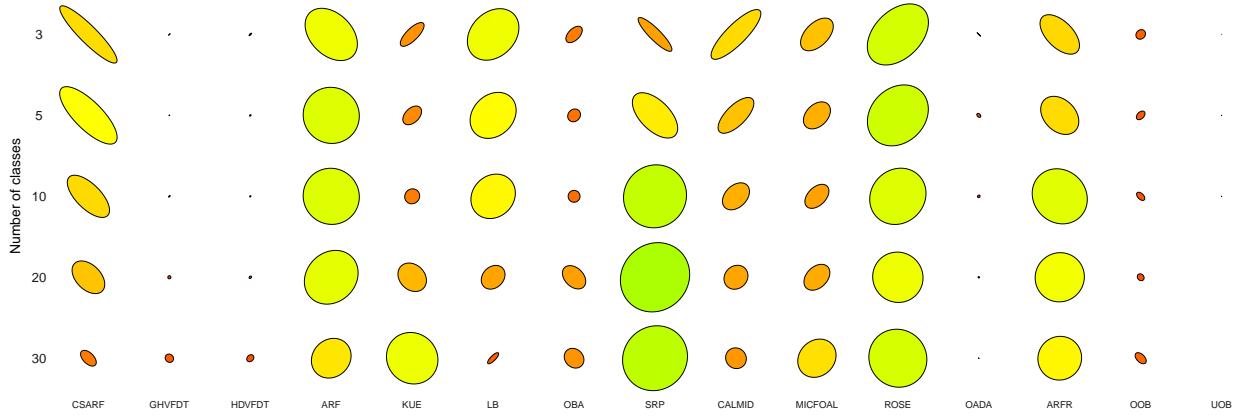


Fig. 38: Comparison of all 15 algorithms on the number of classes under concept drift and multi-class shifting imbalance ratio. Axes of the ellipse represent PMAUC and Kappa metrics. Color gradient represents the product of both metrics.

### 6.2.6 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 employed in the experiment are popular benchmarks for data streams classifiers, and their specifications are presented in Table 17. The PMAUC and Kappa for the five selected classifiers are presented in Figure 39. Table 18 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 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 backbone ensemble architecture.

Table 17: Real-world multi-class datasets specifications.

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

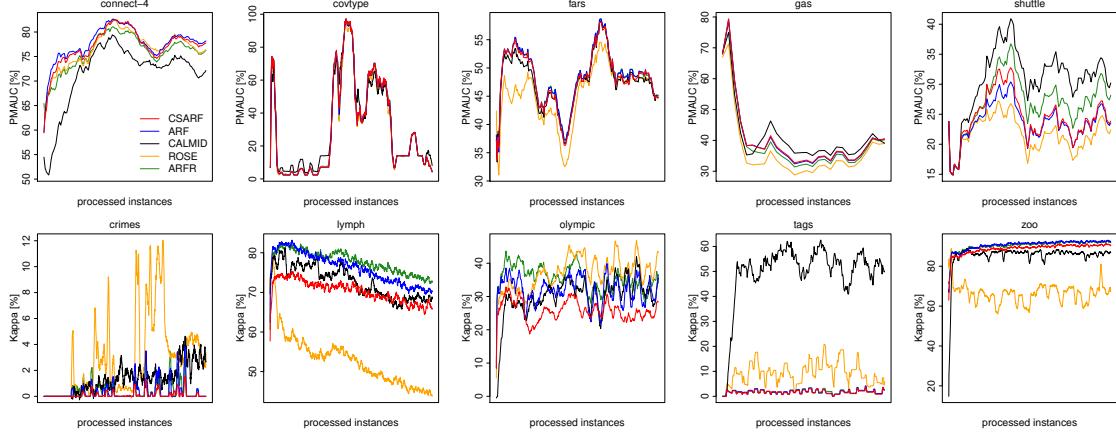


Fig. 39: PMAUC and Kappa on multi-class imbalanced datasets.

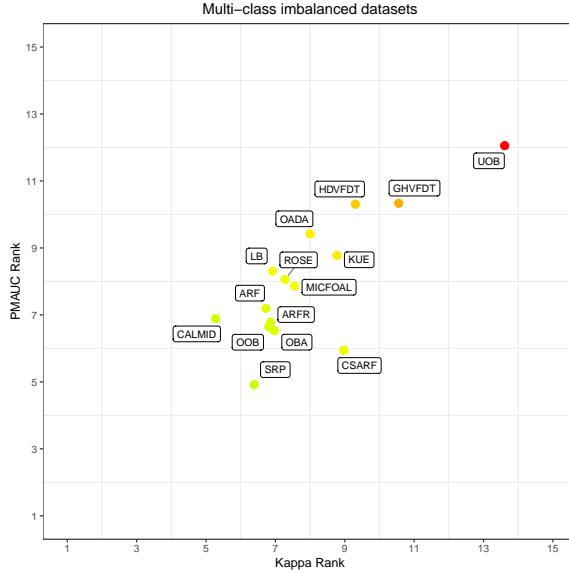


Fig. 40: Comparison of all 15 algorithms for multi-class imbalanced datasets. Color gradient represents the product of both metrics.

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. SRP displayed the best results in real-world datasets regarding PMAUC, and it was the second best 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.

Table 18: PMAUC and Kappa on multi-class imbalanced datasets.

Dataset	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB	
PMAUC	activity	71.73	71.29	61.00	70.94	72.21	60.81	57.04	66.06	72.13	<b>73.03</b>
	connect-4	77.73	78.25	67.86	76.83	<b>79.99</b>	72.80	73.83	77.20	76.48	76.20
	cov-pok-elec	6.91	6.91	7.10	6.92	6.91	<b>9.21</b>	6.96	7.25	6.91	7.22
	covtype	29.94	29.77	28.02	29.11	29.74	<b>30.94</b>	30.01	28.83	29.74	27.31
	crimes	23.62	23.63	23.98	24.69	25.85	23.31	23.65	25.25	23.64	<b>26.23</b>
	fars	48.10	48.49	47.05	47.60	<b>48.62</b>	47.67	47.60	45.54	48.46	44.63
	gas	39.51	39.52	28.01	<b>41.59</b>	41.35	39.78	40.22	37.11	38.91	36.45
	hypothyroid	<b>45.55</b>	35.92	43.93	35.98	36.99	41.89	41.46	39.65	40.97	45.31
	kddcup	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>	<b>0.01</b>
	kr-vs-k	<b>0.14</b>	<b>0.14</b>	0.13	0.13	<b>0.14</b>	<b>0.14</b>	<b>0.14</b>	<b>0.14</b>	<b>0.14</b>	0.13
	lymph	<b>98.12</b>	96.72	96.47	95.67	97.50	95.89	96.96	94.09	97.16	97.32
	olympic	<b>78.18</b>	76.82	71.14	74.61	73.41	64.96	67.87	73.89	75.73	66.16
	poker	6.41	6.40	6.61	6.39	6.38	<b>10.16</b>	6.35	6.99	6.42	6.73
	sensor	34.04	34.03	33.97	34.04	<b>35.05</b>	34.04	34.03	33.99	34.03	33.64
	shuttle	24.72	23.89	26.01	20.73	26.78	<b>31.50</b>	23.72	21.88	27.64	28.44
	tags	4.39	4.39	5.90	5.35	4.66	<b>6.54</b>	6.08	4.80	4.39	6.27
	thyroid	98.70	98.42	92.58	97.38	<b>98.74</b>	95.48	97.06	98.00	98.25	97.56
	zoo	98.31	98.23	98.21	95.05	<b>99.07</b>	95.09	97.56	93.61	98.15	97.95
Kappa	activity	61.06	59.47	32.42	49.63	<b>63.91</b>	34.69	26.52	33.64	59.89	52.71
	connect-4	36.84	32.16	19.84	34.63	33.64	33.49	30.31	<b>42.05</b>	35.62	38.22
	cov-pok-elec	0.17	0.43	1.21	0.35	0.12	<b>34.07</b>	0.48	5.26	0.28	3.30
	covtype	48.10	51.89	38.26	44.83	52.84	<b>77.96</b>	61.00	41.31	52.23	36.75
	crimes	0.12	0.34	0.54	1.95	<b>4.49</b>	1.29	-0.02	2.96	0.40	4.23
	fars	48.78	68.18	63.42	64.80	67.89	67.66	62.74	37.89	<b>70.26</b>	50.88
	gas	72.50	72.89	13.58	66.80	<b>82.22</b>	56.05	79.45	46.50	63.41	38.07
	hypothyroid	11.56	10.75	8.02	10.13	7.99	13.65	14.02	15.40	0.80	<b>22.18</b>
	kddcup	0.01	0.02	0.04	0.06	0.06	0.14	0.03	<b>0.20</b>	0.05	0.03
	kr-vs-k	<b>1.44</b>	1.23	0.04	<b>1.44</b>	1.35	1.41	1.15	0.97	1.20	-0.42
	lymph	70.38	76.26	75.92	67.29	<b>82.39</b>	72.69	80.55	52.84	77.79	79.81
	olympic	25.93	32.16	24.81	35.31	8.60	30.62	31.16	<b>37.26</b>	35.51	26.86
	poker	0.41	0.52	3.38	0.45	0.26	<b>55.75</b>	0.00	8.33	0.87	4.68
	sensor	3.09	3.17	2.16	<b>3.51</b>	0.00	3.40	3.29	2.86	3.18	0.35
	shuttle	-3.44	3.58	30.92	0.22	33.97	<b>75.50</b>	22.47	0.83	15.53	17.21
	tags	1.78	1.92	23.45	16.29	7.50	<b>51.26</b>	36.06	9.54	1.93	27.60
	thyroid	76.53	<b>88.29</b>	71.51	88.03	82.82	79.38	79.11	87.66	78.18	83.99
	zoo	88.97	90.89	90.30	85.08	<b>93.56</b>	85.90	89.36	66.39	90.60	89.64
Avg. PMAUC	<b>43.67</b>	42.94	41.00	42.39	43.52	42.23	41.70	41.91	43.29	42.81	
Avg. Kappa	30.24	33.01	27.77	31.71	34.65	<b>43.05</b>	34.32	27.33	32.65	32.01	
Rank PMAUC	4.33	5.39	6.75	6.33	<b>3.92</b>	5.19	6.11	6.25	5.25	5.47	
Rank Kappa	6.92	5.28	6.94	5.33	4.81	<b>4.06</b>	5.64	5.44	5.22	5.36	

### 6.2.7 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 and Krawczyk, 2020). We used all 9 multi-class semi-synthetic data streams proposed in <sup>4</sup>. 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 19 presents the average PMAUC and Kappa for the top 10 classifiers for 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.

<sup>4</sup> <https://github.com/mlrep/imb-drift-20>

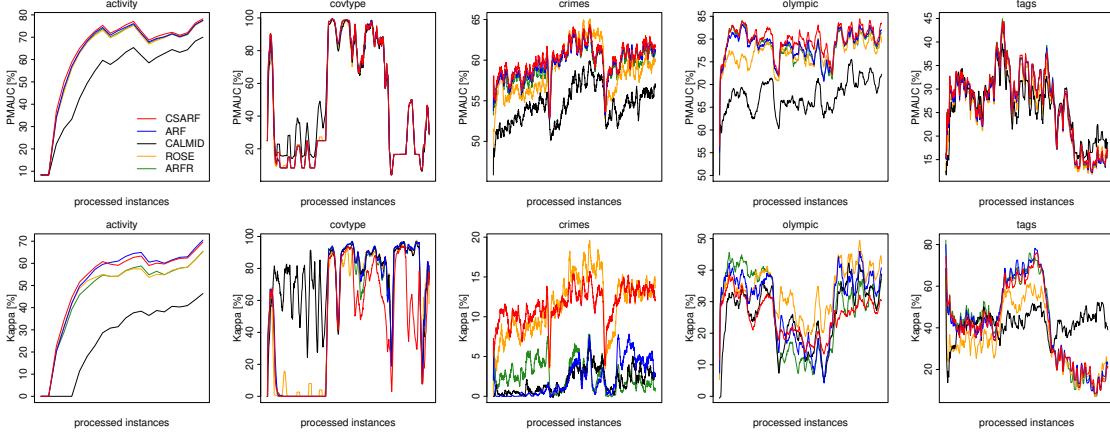


Fig. 41: PMAUC and Kappa on semi-synthetic multi-class imbalanced datasets.

Table 19: PMAUC and Kappa on semi-synthetic multi-class imbalanced datasets.

	Dataset	CSARF	ARF	KUE	LB	SRP	CALMID	MICFOAL	ROSE	ARFR	OOB
PMAUC	activity	71.88	71.27	57.07	70.79	<b>73.05</b>	64.39	61.18	70.94	71.24	68.45
	connect-4	77.58	78.05	67.25	76.24	<b>79.90</b>	72.23	72.50	76.97	76.10	72.98
	covtype	46.75	46.65	45.93	46.12	46.94	<b>49.11</b>	47.01	46.27	46.53	48.87
	crimes	60.04	59.71	56.06	55.75	<b>61.62</b>	54.46	56.75	58.71	59.29	54.91
	gas	44.98	45.02	32.27	42.59	<b>45.19</b>	40.49	42.44	43.39	44.17	36.50
	olympic	<b>80.16</b>	78.75	72.83	77.16	75.72	67.73	70.17	77.14	78.53	65.73
	poker	28.52	27.75	28.56	27.92	28.59	29.32	28.97	26.84	26.13	<b>29.43</b>
	sensor	<b>43.80</b>	43.66	42.01	43.62	41.69	43.39	43.63	43.17	43.70	43.54
	tags	26.95	26.60	21.32	27.61	27.59	26.14	<b>28.27</b>	25.69	26.81	23.56
Kappa	activity	63.26	64.08	23.79	57.88	<b>68.36</b>	42.16	45.65	59.26	59.47	49.49
	connect-4	38.10	31.84	19.86	34.45	33.63	33.02	29.31	<b>41.57</b>	30.53	32.10
	covtype	44.42	55.00	50.85	52.05	59.08	<b>75.04</b>	62.02	54.07	54.15	56.85
	crimes	11.52	2.14	4.42	0.52	<b>14.33</b>	1.89	5.14	11.41	2.53	4.58
	gas	69.09	<b>71.61</b>	24.65	54.29	70.78	49.61	59.49	59.26	58.35	30.50
	olympic	25.41	29.36	22.84	29.20	10.51	26.21	29.47	<b>34.09</b>	27.12	14.27
	poker	14.86	19.17	27.99	24.96	25.76	<b>46.97</b>	46.59	34.79	12.73	36.79
	sensor	52.36	56.56	45.97	54.76	56.00	59.86	<b>60.65</b>	50.18	56.92	56.76
	tags	39.76	40.87	8.84	37.47	45.14	42.27	<b>56.42</b>	34.07	40.92	10.29
Avg. PMAUC		<b>53.41</b>	53.05	47.03	51.98	53.37	49.70	50.10	52.12	52.50	49.33
Avg. Kappa		39.86	41.18	25.47	38.40	42.62	41.89	<b>43.86</b>	42.08	38.08	32.40
Rank PMAUC		<b>3.00</b>	3.89	8.67	5.78	3.44	6.78	5.44	6.33	5.00	6.67
Rank Kappa		5.56	4.89	8.67	6.56	3.89	4.89	<b>3.67</b>	4.78	5.89	6.22

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 and 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 the used backbone.

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 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 and Krawczyk, 2021b).

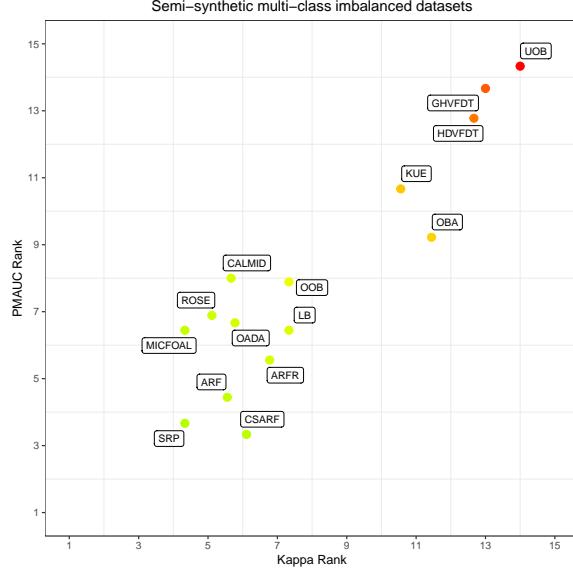


Fig. 42: Comparison of all 15 algorithms for semi-synthetic multi-class imbalanced datasets. Color gradient represents the product of both metrics.

### 6.3 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.** 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 20 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.

### 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 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.

Table 20: 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	0.17	1.77E-05	5.30	6.65	—	—	—	—
C-SMOTE	27.97	2.50E-01	16.65	18.75	—	—	—	—
VFC-SMOTE	17.56	1.24E-01	13.85	14.85	—	—	—	—
CSARF	24.68	2.04E-01	16.10	17.10	5.44	7.04E-05	12.58	12.40
GHVFDT	0.03	4.10E-07	1.50	1.90	0.06	1.72E-09	1.68	1.40
HDVFDT	<b>0.02</b>	<b>2.13E-07</b>	<b>1.50</b>	<b>1.90</b>	<b>0.06</b>	<b>1.47E-09</b>	1.90	1.65
ARF	24.55	1.99E-01	16.15	17.10	5.35	7.07E-05	12.03	11.88
KUE	0.46	8.20E-05	7.75	8.10	0.56	8.57E-07	7.83	8.18
LB	0.57	1.23E-04	9.00	9.55	0.41	2.90E-07	7.30	6.98
OBA	0.15	8.11E-06	3.85	3.75	0.28	1.21E-07	5.35	5.35
SRP	53.66	7.34E-01	19.85	20.65	8.73	1.19E-04	13.65	13.53
ESOS-ELM	5.52	1.89E-04	12.50	10.63	—	—	—	—
CALMID	0.45	8.65E-05	7.60	8.08	0.28	1.21E-07	5.35	5.35
MICFOAL	14.78	7.67E-02	13.35	14.10	4.16	5.07E-05	10.93	11.75
ROSE	0.42	8.62E-05	7.75	8.70	0.65	5.24E-07	8.55	8.25
OADA	34.22	3.25E-01	18.50	17.90	50.76	8.88E-04	15.00	14.90
OADAC2	32.37	1.41E-01	18.55	17.20	—	—	—	—
ARFR	8.64	3.01E-02	11.95	13.10	4.29	4.58E-05	10.83	10.55
SMOTE-OB	182.81	8.18E-01	23.15	18.25	—	—	—	—
OSMOTE	179.48	1.03E+00	23.75	19.65	—	—	—	—
OOB	0.42	7.01E-05	6.95	6.85	0.34	2.41E-07	5.68	6.25
UOB	0.16	9.88E-06	3.80	4.35	0.08	6.52E-09	2.43	2.95
ORUB	37.16	5.04E-01	19.25	19.90	—	—	—	—
OUOB	95.22	1.11E+00	21.40	21.00	—	—	—	—

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 backbone architecture ARF to deal with multi-class scenarios and adapt to changes on multiple classes. This combined with a informed resampling 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 unstable ensembles with 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 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 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 almost 50 times bigger than its peers. This is another motivation against using current boosting-based algorithms for imbalanced data streams.

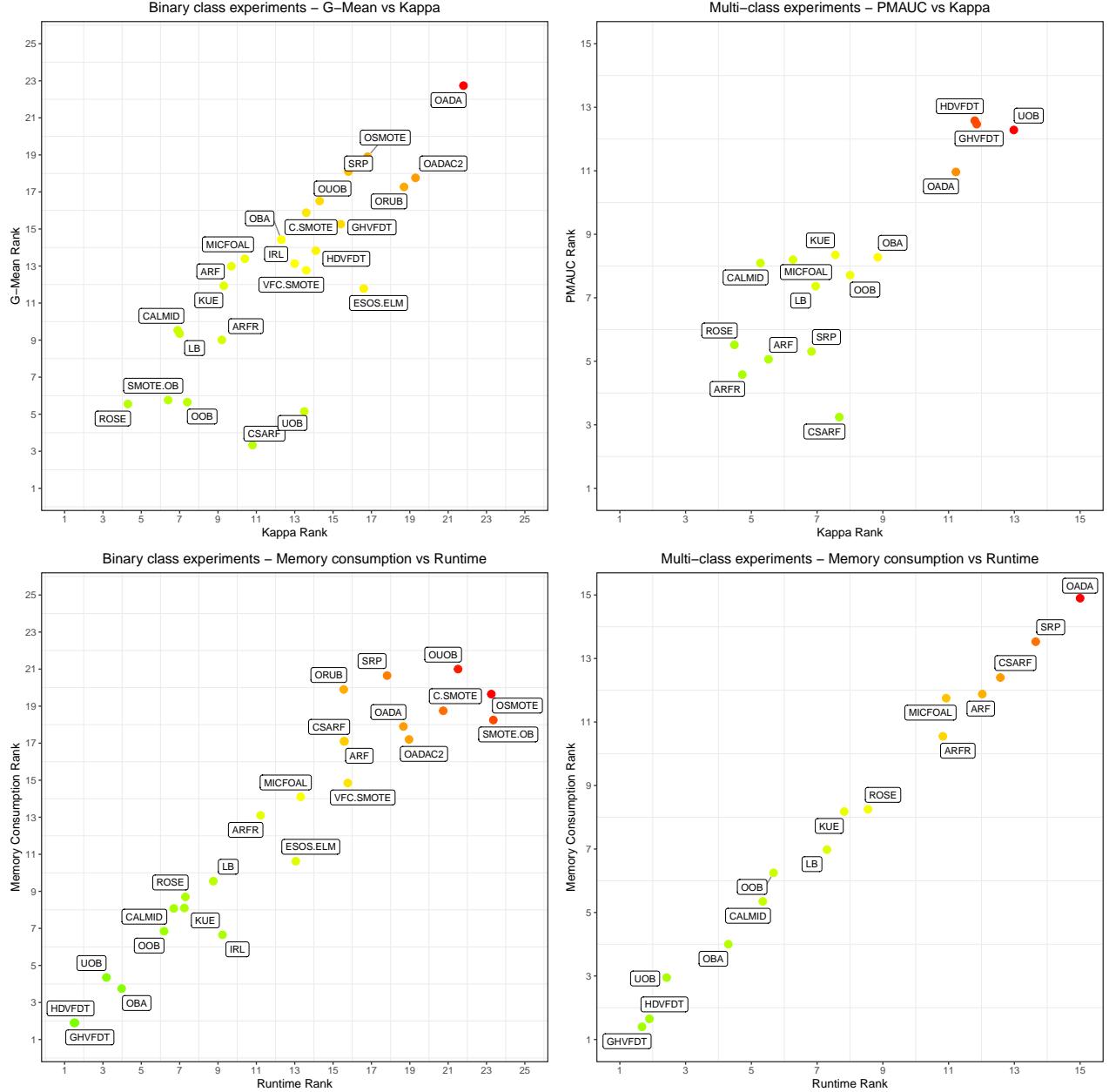


Fig. 43: Overall comparison of algorithms' ranks for G-Mean/PMAUC vs Kappa and Memory Consumption vs Runtime on binary and multi-class imbalanced benchmarks. Color gradient represents the product of each pair of metrics.

*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).

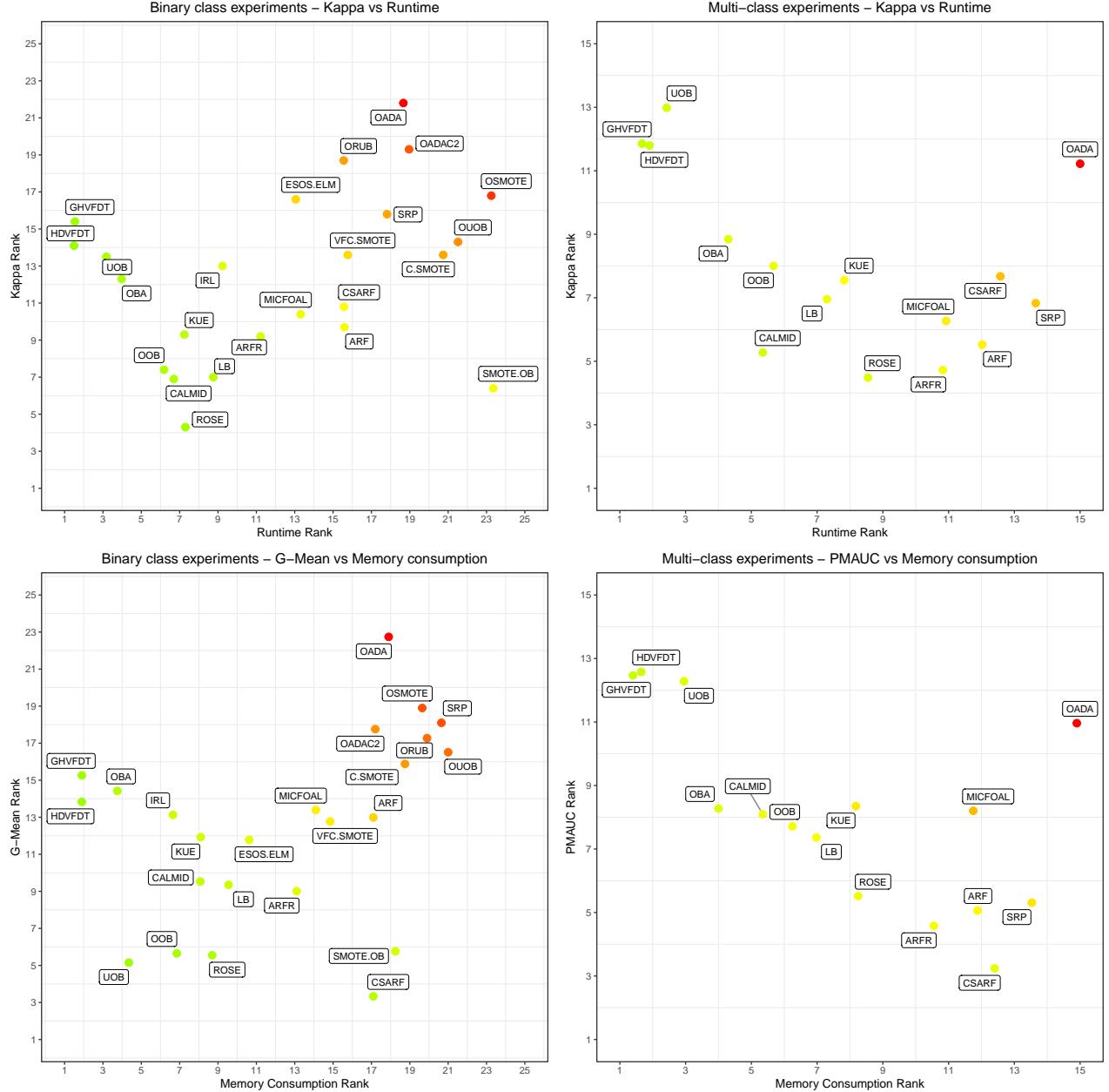


Fig. 44: Overall comparison of algorithms' ranks for G-Mean/PMAUC/Kappa vs Runtime/Memory Consumption on binary and multi-class imbalanced benchmarks. Color gradient represents the product of each pair of metrics.

Analyzing all our experiments, we aim to select such classifiers that balance both sides. We can clearly see that ROSE and CALMID presented the best trade-off between their predictive performance and computational complexity for binary and multi-class experiments. Even though they are ensemble methods, they rely on highly efficient hybrid architectures and do not employ any costly mechanisms such as oversampling or adaptive cost-sensitive matrix. OOB also very good balance between predictive power and runtime/memory consumption. This is achieved due to the combination of bagging-based architecture with online blind oversampling that is fast and does not require any costly distance computations.

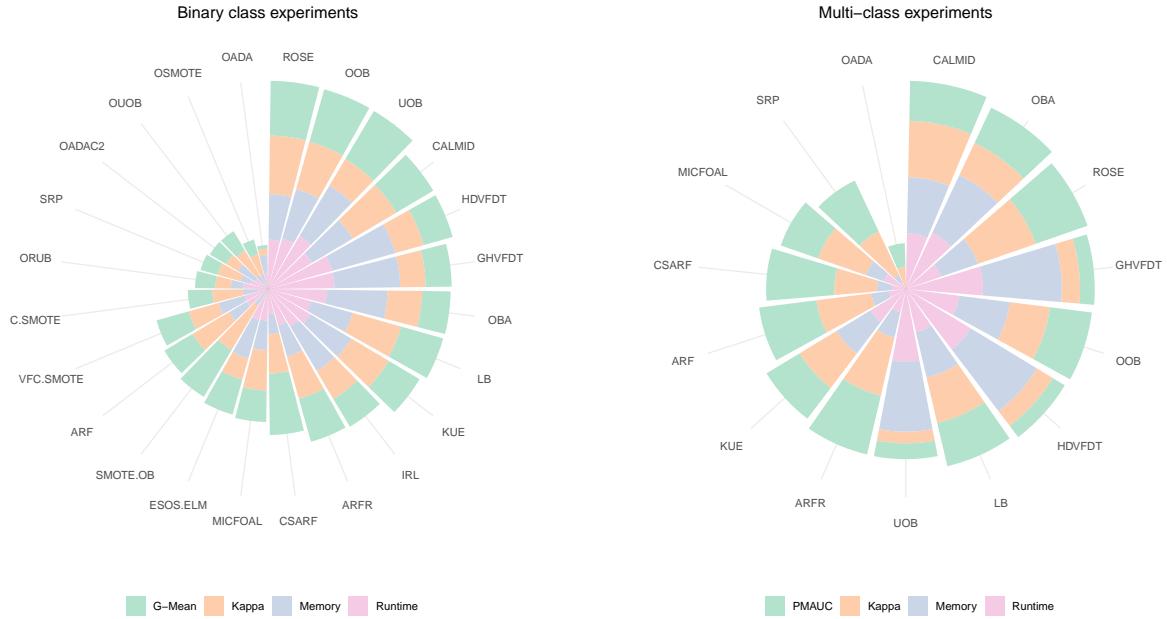


Fig. 45: Overall comparison of stacked algorithms' ranks for G-Mean/PMAUC, Kappa, runtime, and memory on binary and multi-class imbalanced benchmarks. Algorithms are sorted clockwise by the stacked ranks best to worst.

## 7 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 backbone 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 backbone, not being able to cope with high imbalance ratios or data-level difficulties.

**Adaptation to concept drift vs 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 and Skryjomska, 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 and 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 usage of Kappa and G-Mean/PMAUC. These metrics assess different 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).

**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.

## 8 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. 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 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 and Skryjomski, 2017). Therefore, we recommend to truly understand the nature of streams you are working with and focusing on how you can elevate 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), others 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.

**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 and Krawczyk, 2021b). This is just a 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 testbed 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.

**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), and Stream-learn (Ksieniewicz and Zyblewski, 2022). This will make sure that other researchers can use your classifier, as well as that it can be easily embedded in existing frameworks, such as our testbed.

## 9 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 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**. In data stream scenarios new classes may emerge over time, while the old ones may disappear. This, combined with dynamic imbalance ratio, creates an extremely challenging scenario that will require new and flexible models capable of detecting and incorporating new classes into their structures, as well as forgetting the outdated ones. We envision strong parallels with continual and lifelong learning approaches (Korycki and 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 underrepresented, 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 excellent 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. Finally, **streaming times series** also require dedicated resampling and skew-insensitive methods to facilitate robust predictions.

## 10 Conclusions

**Summary.** In this paper, we offered an exhaustive and informative review of classifiers methods for imbalanced data streams. Based on previous studies and the main characteristics of algorithms we proposed a novel taxonomy to arrange, understand and standardize each component of a classifier designed for imbalanced streams. Besides proposing a taxonomy, we designed a robust experimental setup, publicly available for reproducibility, to evaluate state-of-the-art classifiers in unique 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 setup, 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 in a plethora of 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 approach work or does not work for each scenario, such as undersampling techniques were undermined in multi-class scenarios, and dynamic ensemble methods such as ROSE could excel in many different settings, demonstrating robustness. Our proposed experimental setting 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 results possible.

**Towards the future of reproducible research in data stream mining.** We proposed a first standardized and holistic testbed for learning from 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.

## Declarations

**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).

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

**Ethics approval.** Not Applicable.

**Consent to participate.** Not Applicable.

**Consent for publication.** Not Applicable.

**Availability of data and material.** Data & materials available at <https://github.com/canoalberto/imbalanced-streams>

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

**Authors' 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.

## Acronyms

ARF . . . . .	Adaptative Random Forest
ARFR . . . . .	Adaptative Random Forest with Resampling
CALMID . . . . .	Comprehensive active learning method for multiclass imbalanced streaming data with concept drift
CSARF . . . . .	Cost Sensitive Adaptative Random Forest
C-SMOTE . . . . .	Continuous Synthetic Minority Oversampling
ESOS-ELM . . . . .	Ensemble of subset online sequential extreme learning machine
GHVFDT . . . . .	Gaussian Hellinger Very Fast Decision Tree
HDVFDT . . . . .	Hellinger Distance Very Fast Decision Tree
IRL . . . . .	Incremental Rebalancing Learning
KUE . . . . .	Kappa Updated Ensemble
LB . . . . .	Leveraging Bagging Adwin
MICFOAL . . . . .	Multiclass Imbalanced and Concept Drift Network Traffic Classification Framework Based on Online Active Learning
OADA . . . . .	Online AdaBoost
OADAC2 . . . . .	Online AdaC2
OBA . . . . .	Online Bagging with ADWIN
OOB . . . . .	Oversampling Online Bagging
ORUB . . . . .	Online RUSBoost
OSMOTE . . . . .	Online Continuous Synthetic Minority Oversampling Bagging
OUOB . . . . .	Online Undersampling and Oversampling Bagging
ROSE . . . . .	Robust Online Self-Adjusting Ensemble
SMOTE . . . . .	Synthetic Minority Oversampling Technique
SMOTE-OB . . . . .	Synthetic Minority Oversampling Technique with Online Bagging
SRP . . . . .	Streaming Random Patches
UOB . . . . .	Undersampling Online Bagging
VFC-SMOTE . . . . .	Very fast continuous synthetic minority oversampling

## 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
- 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 RP, 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 RP, 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 HM, Maniu S (2021) Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11:e1405
- Barros RSM, Santos SGTC (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 (2022) An extensive study of C-SMOTE, a Continuous Synthetic Minority Oversampling Technique for Evolving Data Streams. *Expert Systems with Applications* 196:116630
- 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 HM, 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
- 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
- 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
- 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, World Scientific, pp 44–68
- 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, Szczęch 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 LL, 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* In Press
- Chrysakis A, Moens M (2020) Online continual learning from imbalanced data. In: International Conference on Machine Learning, vol 119, pp 1952–1961
- Cieslak DA, Chawla NV (2008) Learning decision trees for unbalanced data. In: European Conference on Machine Learning and Knowledge Discovery in Databases, pp 241–256
- da Costa VGT, de Leon Ferreira ACP, Junior SB, et al. (2018) Strict Very Fast Decision Tree: a memory conservative algorithm for data stream mining. *Pattern Recognition Letters* 116:22–28
- 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 (WEKOI): An approach for learning from imbalanced data streams. *Journal of Computational Science* 61:101614
- 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 YC (2021) Online ensemble learning algorithm for imbalanced data stream. *Applied Soft Computing* 107:107378
- Fernández A, García S, Galar M, Prati RC, Krawczyk B, Herrera F (2018) Learning from imbalanced data sets, vol 10. Springer
- Fernández A, García S, Herrera F, Chawla NV (2018) SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research* 61:863–905
- Ferreira LEB, Gomes HM, Bifet A, Oliveira LS (2019) Adaptive random forests with resampling for imbalanced data streams. In: International Joint Conference on Neural Networks, pp 1–6
- Gama J (2010) Knowledge discovery from data streams. CRC Press
- Gama J, Sebastiao R, Rodrigues PP (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
- Ghazikhani A, Monsefi R, Yazdi HS (2013) Ensemble of online neural networks for non-stationary and imbalanced data streams. *Neurocomputing* 122:535–544
- Ghazikhani A, Monsefi R, Yazdi HS (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 HM, Bifet A, Read J, Barddal JP, Enembreck F, Pfahringer B, Holmes G, Abdessalem T (2017) Adaptive random forests for evolving data stream classification. *Machine Learning* 106(9):1469–1495
- Gomes HM, Read J, Bifet A (2019) Streaming random patches for evolving data stream classification. In: IEEE International Conference on Data Mining, pp 240–249
- Gomes HM, Grzenda M, Mello R, Read J, Le Nguyen MH, Bifet A (2022) A survey on semi-supervised learning for delayed partially labelled data streams. *ACM Computing Surveys* In Press
- 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. In: International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, pp 7–14
- Iosifidis V, Zhang W, Ntoutsi E (2021) Online fairness-aware learning with imbalanced data streams. arXiv preprint arXiv:210806231
- Jedrzejowicz J, Jedrzejowicz P (2020) GEP-based classifier with drift detection for mining imbalanced data streams. *Procedia Computer Science* 176:41–49
- 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
- Khreich W, Granger E, Miri A, Sabourin R (2010) Iterative Boolean combination of classifiers in the ROC space: An application to anomaly detection with HMMs. *Pattern Recognition* 43(8):2732–2752
- Kim CD, 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
- 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 L, 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
- Korycki L, 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
- 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, Skryjomska 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
- Krawczyk B, Minku LL, Gama J, Stefanowski J, Woźniak M (2017) Ensemble learning for data stream analysis: A survey. *Information Fusion* 37:132–156
- 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
- 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* p 116962
- 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* 1994(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
- 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
- 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 JP, 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 YM, Tang YY (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 YM, Tang YY (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
- Lyon RJ, Knowles JD, Brooke JM, Stappers BW (2014) Hellinger Distance Trees for Imbalanced Streams. In: IEEE International Conference on Pattern Recognition, pp 1969–1974
- Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD (2008) Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks* 21(2-3):427–436

- 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 RA, Wong KW, Shiratuddin MF, 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 RA, Wong KW, Shiratuddin MF, 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 SM, Bolmier G, Sourty R, Vaysse R, Zouitine A, Gomes HM, Read J, Abdessalem T, Bifet A (2020) River: machine learning for streaming data in python. 2012.04740
- Nguyen VL, Destercke S, Masson MH (2018) Partial data querying through racing algorithms. *International Journal of Approximate Reasoning* 96:36–55
- Priya S, Uthra RA (2021) Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. *Complex & Intelligent Systems* pp 1–17
- 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 HL (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 MS, Abreu PH, 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
- 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, 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
- 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
- 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 LL (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 LL, 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 LL, 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 LL, 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 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
- Webb GI, Hyde R, Cao H, Nguyen HL, 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 YC (2022) Dynamic weighted selective ensemble learning algorithm for imbalanced data streams. *The Journal of Supercomputing* 78(4):5394–5419
- 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

- Zhang H, Liu W, Liu Q (2020) Reinforcement online active learning ensemble for drifting imbalanced data streams. *IEEE Transactions on Knowledge and Data Engineering* In Press
- Zhu ZB, Song ZH (2010) Fault diagnosis based on imbalance modified kernel fisher discriminant analysis. *Chemical Engineering Research and Design* 88(8):936–951
- 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
- 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