Machine Learning (2022) 111:2561–2599
https://doi.org/10.1007/s10994-022-06168-x

ROSE: robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams

Alberto Cano[1] · **Bartosz Krawczyk**[1]

Received: 11 June 2021 / Revised: 10 March 2022 / Accepted: 14 March 2022 /
Published online: 20 April 2022

Abstract

Data streams are potentially unbounded sequences of instances arriving over time to a classifier. Designing algorithms that are capable of dealing with massive, rapidly arriving information is one of the most dynamically developing areas of machine learning. Such learners must be able to deal with a phenomenon known as concept drift, where the data stream may be subject to various changes in its characteristics over time. Furthermore, distributions of classes may evolve over time, leading to a highly difficult non-stationary class imbalance. In this work we introduce Robust Online Self-Adjusting Ensemble (ROSE), a novel online ensemble classifier capable of dealing with all of the mentioned challenges. The main features of ROSE are: (1) online training of base classifiers on variable size random subsets of features; (2) online detection of concept drift and creation of a background ensemble for faster adaptation to changes; (3) sliding window per class to create skew-insensitive classifiers regardless of the current imbalance ratio; and (4) self-adjusting bagging to enhance the exposure of difficult instances from minority classes. The interplay among these features leads to an improved performance in various data stream mining benchmarks. An extensive experimental study comparing with 30 ensemble classifiers shows that ROSE is a robust and well-rounded classifier for drifting imbalanced data streams, especially under the presence of noise and class imbalance drift, while maintaining competitive time complexity and memory consumption. Results are supported by a thorough non-parametric statistical analysis.

Keywords Data streams · Concept drift · Online learning · Continual learning · Imbalanced data

Editor: Indre Zliobaite.

✉ Alberto Cano
acano@vcu.edu

Bartosz Krawczyk
bkrawczyk@vcu.edu

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

1 Introduction

Modern data is characterized by two crucial factors: volume (massive size) and velocity (ever-growing speed and changing nature of data). The combination of these two factors gave rise to the notion of data streams (Bahri et al., 2021; Bifet et al., 2019; Gomes et al., 2019b). Streaming scenarios pose unique challenges to machine learning algorithms, as we are not only concerned about their predictive power, but also about their computational complexity, response latency, and capability of adapting to and incorporating new data. Additionally, data streams evolve over time and their characteristics and definitions are subject to change. This is known as concept drift, which forces classifiers to constantly update and adapt to the current state of data (Gama et al., 2014; Lu et al., 2019a). Furthermore, challenges present in static classification can emerge in streaming environments. Class imbalance is one of the most relevant challenges (Branco et al., 2016). When combined with concept drift, no longer only the disproportions among classes pose a learning difficulty, class roles and imbalance ratio may change dynamically. This renders the majority of traditional algorithms dedicated to countering imbalanced distributions inadequate for data streams (Fernández et al., 2018). All of those mentioned challenges have led to intensive research on algorithms capable of thriving in such difficult environments. Ensembles emerged as the most powerful solutions (Krawczyk et al., 2017).

In this paper we introduce Robust Online Self-Adjusting Ensemble (ROSE), a novel online ensemble architecture dedicated to mining imbalanced and drifting data streams. It incorporates four primary features that allow ROSE to handle any type of data stream and concept drift and offer robustness to variable class imbalance over time. ROSE employs adaptive self-tuning, adjusting its parameters and ensemble line-up dynamically on the go for best performance, without the need for human supervision or ad-hoc solutions. The main contributions of this paper are:

- *Novel online ensemble architecture on dynamic feature subspaces* ROSE is an online self-adjusting ensemble for exploring variable-size feature subspaces to adapt to concept drift and dynamic class imbalance ratios in non-stationary data streams.
- *Background ensemble for concept drift adaptation* ROSE monitors the base classifiers for detecting concept drift within each of the feature subspaces. If a drift warning is emitted, the algorithm learns a new ensemble on the background on a new set of feature subspaces. The performance of base classifiers in the current and background ensemble are compared, selecting the top performing ones to replace the ensemble. This allows for adding new classifiers to the ensemble that are specialized on the current concept and discarding outdated models adapting to changes in the feature space.
- *Automatic handling of class imbalance* ROSE holds a sliding window buffer per class to keep a representation of the most recent instances on which to build new background base learners. This counters class imbalance, as the buffer enforces an undersampling of majority classes.
- *Enhancing the exposure to minority class instances* In order to further make ROSE skew-insensitive, we propose a self-adjusting λ for bagging to reflect the evolving distribution of the data classes and enforce the Hoeffding bound to improve the classification performance on minority classes.
- *Extensive and reproducible experimental framework* The performance of ROSE is examined based on a comprehensive experimental study and comparison with 30 state-of-the-art ensembles. We present seven different sets of experiments on imbalanced

streams, artificial stream generators, noisy streams, and real-world data streams. This makes the present study one of the most thorough and reproducible experimental analysis of ensemble performance with concept drift and class imbalance.

The rest of the paper is organized as follows. Section 2 presents an overview of data streams and related works in ensemble learning. Section 3 discusses the challenges and approaches for imbalanced data streams. Section 4 presents the proposed ROSE algorithm and its features. Section 5 presents a thorough experimental study on a large set of data streams, including imbalanced streams with concept drift, varying imbalance ratio, and noise, as well as an ablation study. Experimental results are also validated through non-parametric statistical analysis. Finally, Sect. 6 summarizes the concluding remarks and discusses future lines of work.

2 Learning from data streams

Preliminaries We define a data stream as a sequence $< S_1, S_2, \dots, S_n, \dots >$, in which each element S_j is a collection of instances (batch scenario) or a single instance (online scenario). Each instance is independent and randomly generated using a stationary probability distribution D_j. In this paper, we consider the supervised online learning scenario that allows us to define each element as $S_j \sim p_j(x^1, \dots, x^d, y) = p_j(\mathbf{x}, y)$, where $p_j(\mathbf{x}, y)$ is a joint distribution of j-th instance, defined by d-dimensional feature space and belonging to class y. Each instance in the stream is independent and randomly drawn from a stationary probability distribution $\Psi_j(\mathbf{x}, y)$.

Concept drift Whenever a new instance (or batch of instances) arrives, we refer to the progression of the data stream. If $\mathrm{S}_j \rightarrow \mathrm{S}_{j+1}$ (where $\mathrm{D}_j = \mathrm{D}_{j+1}$) is true, then we deal with a stationary data stream and no changes occur. However, real-life problems are very frequently subject to concept drift, where the characteristics and definitions of a stream change. Drifts can be of various characteristics and understanding what type of change is currently affecting the stream helps to better adapt to it (Lu et al., 2019a). Concept drift taxonomy analyzes two factors: (1) influence on the decision boundaries; and (2) speed of change. The former divides concept drift into virtual and real. Virtual concept drift affects only the distribution of feature values within each class, but does not affect posterior probabilities. Real concept drift affects the decision boundaries of a classifier, increasing the error of the underlying classifier. This type of drift enforces an adaptation of a classifier in order to maintain high predictive power. When looking at the speed of changes, one may distinguish three types of concept drift. Sudden drift takes place instantaneously, switching to a new distribution at a given point. Gradual drift interleaves instances from old and new concepts. Incremental concept drift can be seen as a transition between two states with multiple intermediate concepts between them. Additionally, we distinguish recurring concept drift, where previously seen concepts may reemerge.

There are two potential ways of addressing concept drift: explicit and implicit (Lu et al., 2019a). Explicit drift detection is based on the assumption that we are capable of recognizing when drift is taking place. This is achieved by combining classifiers with an external tool, called drift detectors (de Barros & de Carvalho Santos, 2018). Such detectors are capable of continuous stream monitoring and raising an alarm when it is highly probable that stream is subject to a drift. Various factors are taken into an account, such as classifier's error, statistical distribution of data, similarity metrics, etc. When drift is detected, the classifier is replaced with a new one trained on the most recent instances. The main drawback of drift detectors lies in their requirements for labeled instances (semi-supervised and unsupervised detectors also exist, although they are less accurate) and in the cost paid for false alarms (unnecessary replacement of a competent classifier). Implicit drift detection methods assume that the classifier is capable of self-adjusting to new instances coming from the stream while forgetting the old information (Liu et al., 2016). This way, new information is constantly incorporated into the learner, which should allow for adapting to evolving concepts (Kozal et al., 2021). Drawbacks of implicit methods lie in their parametrization - establishing proper learning and forgetting rates, as well as the size of a sliding window.

Ensemble learning for data streams Ensemble learning has proven itself to be one of the most effective solutions for data streams (Ghomeshi et al., 2019; Krawczyk et al., 2017). It maintains all of the advantages of this approach for static scenarios, such as improved predictive power, increased robustness and stability. Additionally, ensembles can naturally manage concept drift by incorporating new base learners trained on most recent data and discarding outdated ones (Cano & Krawczyk, 2020). New concepts offer a natural way of maintaining diversity among ensemble members, allowing them to continuously be mutually complementary (Gomes et al., 2019a). When looking at the possible approaches to ensemble learning for data streams, three main paths exist (Krawczyk et al., 2017): (1) dynamic combiners; (2) dynamic ensemble setup; and (3) dynamic ensemble updating. Combiners assume that we focus on adapting the combination rule (e.g., weights in voting) to promote classifiers that are best adapted to the current state of the stream. Dynamic ensemble setup assumes that the pool of classifiers should be constantly updated with new ones and pruned to remove its weakest members. Dynamic ensemble updating assumes that classifiers in the ensemble should not be discarded, but continuously updated with new instances, while maintaining their diversity. ROSE proposed in this paper is a hybrid approach that combines the advantages of adaptive online update of base classifiers with dynamic ensemble setup with online pruning, while managing per-class balanced instance buffers.

Continual learning and data stream mining Continual learning is one of the recently emerged paradigms in deep learning that focuses on building models that can accumulate new knowledge without forgetting the previously learned one (Parisi et al., 2019). While the majority of the works in this domain focus purely on deep neural networks and image-based benchmarks, we should note that the general idea of continual learning is not reserved only to them. There exist interesting similarities between continual learning and data stream mining, as both focus on incorporating new information into the model (Krawczyk, 2021). Data stream mining puts emphasis on adaptation to changes (i.e., handling concept drift), while continual learning puts emphasis on retaining knowledge (i.e., avoiding catastrophic forgetting). Recent works point to the potential of combining these two domains, offering learning systems capable of being robust to both catastrophic forgetting and concept drift affecting previously learned knowledge (Cano & Krawczyk, 2019; Korycki & Krawczyk, 2021a). Furthermore, the setting of data stream mining is identical to task-free (Aljundi et al., 2019) or task-agnostic (He et al., 2019) continual learning, where classes arrive mixed with each other and are not separated into pre-defined tasks. In this paper we discuss that data stream mining tools can be beneficial to continual learning scenarios and we show that having a per class buffer allows it to retain knowledge and is parallel to experience replay approaches used to avoid catastrophic forgetting (Buzzega et al., 2020).

3 Imbalanced data streams

Challenges in imbalanced data stream mining Skewed class distributions are a common problem in data stream mining (Aminian et al., 2020; Gao et al., 2008; Wu et al., 2014). When combined with concept drift novel learning difficulties arise. Imbalance ratio is no longer static and will change with the progress of the stream (Brzeziński & Stefanowski, 2017). Classes may switch their roles over time, with minority transitioning to be majority and vice versa. This is known as imbalance ratio drift and poses a significant challenge to the majority of the existing algorithms that need to have a pre–defined minority class in order to effectively balance distributions (Korycki & Krawczyk, 2021b). This drift can be independent from or connected with concept drift, where class definitions will change over time (Wang & Minku, 2020). Therefore, one must not only monitor each class for changes in its properties, but also for changes in its frequency. New classes may appear and old ones disappear, leading to oscillations between binary and multi-class imbalanced (Krawczyk, 2016). In most real-life scenarios, streams are not predefined as balanced or imbalanced, they may be imbalanced only temporarily (Wang et al., 2018). Examples of dynamic class imbalance include evolving user interests over time (where new topics emerge and old ones dynamically change their relevance) (Wang et al., 2014), social media analysis (where new events may take place and existing events may appear with fluctuating frequency) (Liu et al., 2020), or medical data streams (where patient records continually evolve over time and we observe changing ratios of admission reasons) (Al-Shammari et al., 2019).

Data-level approaches for imbalanced data streams While resampling approaches are very popular for standard imbalanced problems, they cannot be trivially adapted to streaming setting. Here, we need to keep track of which class to dynamically resample, to avoid enhancing class imbalance instead of countering it. Modifications of SMOTE algorithm for drifting data streams are popular (Bernardo et al., 2020b), with the most recent versions working with any number of classes and under limited supervision (Korycki & Krawczyk, 2020). Other popular methods include Incremental Oversampling for Data Streams (IOSDS) (Anupama & Jena, 2019) that focus on replicating instances that are not identified as noisy or overlapping; and undersampling via Selection-Based Resampling (SRE) (Ren et al., 2019) that iteratively removes the safe instances from majority class without introducing reverse bias towards the minority class. Some studies report the usefulness of combining multiple resampling approaches together in order to obtain a more diverse representation of the minority class (Bobowska et al., 2019). Drawbacks of existing data-level approaches lie in their high memory requirements (for oversampling) or possibility of removing instances from older concepts that are still relevant (for undersampling).

Algorithm-level approaches for imbalanced data streams As an alternative to resampling incoming data, one may modify the streaming classifier itself to make it skew-insensitive. This can be done either via cost-sensitive adaptation or by modifying the underlying learning mechanisms (Loezer et al., 2020). The cost-sensitive method has been applied successfully to streaming decision trees, where their leaves have been replaced with perceptrons that use threshold adjustment of their decision outputs (Krawczyk & Skryjomski, 2017). Their cost matrix is updated using the current imbalance ratio and the local difficulty factors of incoming instances. 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. Among algorithm-level modifications, the most popular one is the combination of Hoeffding Decision Trees with Hellinger splitting criteria to make them robust to imbalanced distributions (Lyon et al., 2014).

Another approach uses online one-class Support Vector Machines to track minority classes (Klikowski & Wozniak, 2020). Nearest neighbor classifiers have been used efficiently for imbalanced data streams, by modifying their sliding-window approaches with a reactive memory mechanism (Abolfazli & Ntoutsi, 2020; Roseberry et al., 2019, 2021). Drawbacks of algorithm-level solutions lie in their lack of flexibility (as they can be used only with a specific type of classifier) and in their reliance on either external drift detectors (that are either biased towards majority class or sensitive to false alarms) or implicit online adaptation (that may be delayed with respect to drift occurrence).

Ensemble learning for imbalanced data streams Combining multiple classifiers offers a very powerful way of tackling imbalanced data streams, as combining base classifiers with different skew-insensitive solutions allows for increased robustness and diversity that allows additionally to effectively handle concept drifts (Brzeziński & Stefanowski, 2018; Du et al., 2021; Grzyb et al., 2021; Krawczyk et al., 2017). The most popular approach is to combine either under- or oversampling with Online Bagging (Wang etal., 2015). Similar approaches can be applied to either Adaptive Random Forest (Ferreira et al., 2019), Online Boosting (Wang & Pineau, 2016), Random Subspaces (Klikowski & Wozniak, 2019), Dynamic Weighted Majority (Lu et al., 2017), Kappa Updated Ensemble (Cano & Krawczyk, 2020) or any ensemble that can incrementally update its base learners (Li et al., 2020). Robustness of ensembles to class imbalance can also be increased by using dedicated combination schemes or adaptive chunk-based learning (Lu et al., 2019b). Alternatively, one may see preprocessing approaches as a way of ensuring diversity among base classifiers (Korycki & Krawczyk, 2021c). This allows for anticipating the direction of concept drift and choosing the most suitable learner by dynamic classifier (or ensemble) selection (Zyblewski et al. 2021). Finally, abstaining mechanisms can be introduced into ensembles to temporarily remove most uncertain classifiers from contributing to the collective decision–making process (Korycki et al., 2019). The drawback of existing ensemble solutions lies in their specialization to imbalanced streams—they do not perform well when handling balanced streams. As in real-world applications imbalance may be a temporal characteristic of the analyzed stream, their practical applicability is severely limited.

4 ROSE: robust online self-adjusting ensemble

This section presents the ROSE features and algorithm, a robust and well-rounded ensemble classifier that is flexible to various imbalanced data stream mining scenarios. ROSE aims at improving the effectiveness and latency in the response to fast concept drift and varying class imbalance. We will use the notation of an ensemble $\mathcal{E}$ of k γ base classifiers such that $\mathcal{E} = \{\gamma_1, \gamma_2, \dots \gamma_k\}$ are built on the data stream S.

4.1 ROSE features

The main features are: (1) online training of base classifiers on variable size random subsets of features; (2) online detection of concept drift and creation of a background ensemble for faster adaptation to changes; (3) sliding window per class to create skew-insensitive classifiers regardless of the current imbalance ratio; and (4) self-adjusting bagging to enhance the exposure of difficult instances from minority classes.

Variable size random feature subspaces ROSE builds each base classifier γ_j on a random r-dimensional feature subspace φ_j, where $1 \leq r \leq f$ from the original f-dimensional space in the data stream S. The r dimensionality and the φ_j subset of features are both randomly generated for each base learner. It allows to generate diverse feature subspaces of variable size. This is a significant difference when compared to Adaptive Random Forest (Gomes et al., 2017) which selects a static subspace dimensionality for all the base classifiers. Diverse feature subspaces of random size have demonstrated to improve the performance of the ensemble in KUE (Cano & Krawczyk, 2020). However, while KUE follows a uniform probability distribution to pick the subspace size in the range [1,f] (leading to a wide range of sizes), ROSE follows a normal distribution for subspace sizes as in Eq. 1:

$$r = \mu \times f + \frac{(1-\mu) \times f \times \mathcal{N}(0,1)}{2} \tag{1}$$

where μ is 0.7 by default and ranged [0,1] (leading to a more centered subspace size close to the mean), giving the end-user a better control on the feature subspace sizes centered around the desired mean. This allows to maintain a higher diversity of the ensemble and make base classifiers locally specialized in varying regions of the decision space. Using feature subsets offers two additional advantages—reduced effects of noise and allows for a faster adaptation to local concept drifts that affect only certain features. These advantages of this diverse ensemble architecture were demonstrated in KUE (Cano & Krawczyk, 2020).

Detection of concept drift and background ensemble ROSE monitors the base classifiers for detecting concept drift on the respective feature subspaces. Since they exploit different feature subspaces, drift may occur on one or several of the subspaces. Some features may become relevant while others may lose discriminatory power in the classification over time. If a drift warning is emitted by any of the drift detectors (we use the ADWIN drift detector), ROSE starts training another ensemble in the background. Building ensembles in the background is a successful strategy due to the different capabilities that their base classifiers have in adapting to concept drift (Minku & Yao, 2011) as the new ensemble will not be influenced by old concepts which no longer present in the current state of the data stream. ROSE combines this with the different feature subspaces used by the background ensemble, leading to enhanced diversity of individual classifiers and better adaptation to concept drift.

The background ensemble is initialized using a sliding window per class with the most recent instances, providing a solid foundation to learn the most recent decision boundaries. Newly trained base classifiers do not carry any previous history, so when old concepts become irrelevant they will offer better adaptation than their older counterparts. Additionally, new base classifiers are trained using different feature subsets than the ones already in the pool, hence offering ROSE the option to explore new areas of the decision space that may become relevant after a drift. The background ensemble continues learning instance by instance after the first drift warning was emitted, adapting to the new data distribution. After a certain number of instances, which by default is the total window size of 1000 instances, the performance of the current ensemble and the background ensemble can be compared. The novelty compared to other approaches such as (Brzeziński & Stefanowski, 2014a) is the replacement of multiple base classifiers at once. The k base classifiers of the current ensemble and the k base classifiers of the background ensemble compete to select the k best performing classifiers that will replace and become the new ensemble. The weakest worst performing classifiers are discarded. The selection of the best classifiers is driven by the maximization of the product of their accuracy and Kappa metrics.

Kappa is commonly used in imbalanced classification (Brzeziński et al., 2018, 2019). It evaluates the competence of a classifier by measuring 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 ranges from −100 (total disagreement) through 0 (default probabilistic classification) to 100 (total agreement). Kappa penalizes all-positive or all-negative predictions. Moreover, Kappa provides better insight than other metrics in detecting changes in the distribution of the classes in multi-class imbalanced data. However, Kappa may be too drastic in penalizing misclassifications on difficult data. Therefore, we propose the product of accuracy and Kappa to drive the selection and weighting of the classifiers.

This strategy allows for a two-way adaptation to drift: (1) existing base classifiers are updated in an online manner; (2) a new background ensemble is trained on the most recent data per class and on new subset of features. We combine the online incremental learning with the dynamic ensemble setup approach, allowing the addition of new classifiers to the ensemble and removal of the least accurate ones.

Sliding window per class Similar approaches in the literature employ one buffer of 1000 instances as a sliding window to train the base classifiers. However, since data classes are imbalanced, the sliding window will also be skewed. Class distributions may change over time and we need to be prepared to handle evolving and dynamic imbalance ratios. Our original contribution is to propose to employ one sliding window buffer per class to keep a representation of the most recent instances per class. Therefore, we create independent representations for any number of classes that can hold instances from various stages of the stream. ROSE uses this buffer of most recent instances per class to initialize a new ensemble upon drift warning. Since we employ one buffer per class, to keep a fair comparison with similar approaches, the sum of the buffers is limited to the same 1000 instances. Therefore, we define a maximum buffer size per class of 1000/number of classes. This strategy allows ROSE to perform an undersampling of majority classes, retaining only a fixed number of the most recent instances from them. This approach does not add any additional computational complexity, contrary to other methods (Wu et al., 2014). Whenever a new background ensemble is initialized, the sliding window per class provides a balanced class distribution to warm up the new base classifiers. This allows for alleviating the bias towards majority classes and handling evolving imbalance ratios. Furthermore, (Gao et al., 2008) strategies are designed for balancing chunk-based ensembles, while our sliding window strategy is designed for online training of ensembles. ROSE effectively scales up to any number of classes, while other approaches were designed for two-class problems and their chunk rebalancing strategies may suffer when handling more classes inside chunks of the same size.

Self-adjusting λ for bagging ROSE employs online bagging to weight and resample with replacement instances in the subspace using the $Poisson(\lambda)$ distribution. Online bagging improves the performance of data stream ensembles and it is employed in OzaBag (Oza, 2005), Leveraging Bagging (Bifet et al., 2010b), Adaptive Random Forest (Gomes et al., 2017), and KUE (Cano & Krawczyk, 2020). However, existing approaches use a fixed value for λ, typically 1 or 4. Consequently, the weighting and resampling will follow a static distribution for all of the instances, regardless of the imbalance ratio of the classes. Moreover, λ will be constant through all of the stream regardless of whether the stream is stable or recently experienced an imbalance ratio drift. On the other hand, ROSE uses a self-adjusting λ that dynamically changes over time to adapt to varying imbalance ratios, reflecting the increasing difficulty in classifying minority class instances. The initial value of λ is set as $\lambda_{min} = 4$ when the distribution of the classes is not yet known.

Ensembles based on the idea of online bagging use the $Poisson(\lambda)$ distribution to control how many times a given instance will be shown to each base learner. Standard online bagging uses $\lambda = 1$ to mimic static bagging, while algorithms like Leveraging Bagging (Bifet et al., 2010b) or Adaptive Random Forest (Gomes et al., 2017) use $\lambda = 4$ for a more aggressive exploitation of instances. ROSE proposes a dynamic self-adjusting λ value. We keep a histogram of the data class distribution in the window of most recent instances. The value of λ will be dynamically adjusted based on the most recent imbalance ratios between the instance's class and the majority class. We propose to calculate the self-adjusting λ as in Eq. 2:

$$\lambda = \lambda_{min} + \log_{10}(\#\text{majority Class}/\#\text{instance Class}) \times \lambda_{min} \tag{2}$$

where $\lambda_{min} = 4$. This self-adjusting parametrization benefits both balanced and imbalanced distributions. Under balanced data the logarithmic function makes $\lambda = 4$, similar to Leveraging Bagging or Adaptive Random Forest. On the other hand, if the imbalance ratio is 10:1 then $\lambda = 8$, or if the imbalance ratio is 100:1 then $\lambda = 12$. The logarithmic function provides a more reasonable and smoother scaling of the λ value as the imbalance ratio increases. This strategy allows ROSE to enhance the importance of the minority class instances and use them more aggressively to train a balanced classifier. Increased exposure to minority instances will also result in faster creation of new split in decision tree-based classifiers that use Hoeffding's bound, adapting faster to concept drift. Self-adaptive λ for class imbalance was also discussed in (Wang etal., 2015), but the approach proposed there was based on checking conditional clauses and switching between various formulas for lambda calculation. ROSE simplifies this by proposing a single formula for λ calculation, which leads to better classification performance.

4.2 ROSE algorithm

The algorithm to build the ROSE classifier comprises three main stages: (1) the ensemble initialization on a diverse set of random feature subspaces, (2) the ensemble model update per-instance adapting to class imbalance, and (3) the learning of a background ensemble and replacement of base learners to adapt to concept drift and varying properties of the stream. Algorithm 1 presents the pseudo-code of ROSE.

Algorithm 1: ROSE algorithm.

```
   Input: $\mathcal{S}$: data stream, $k$: number of classifiers, $\mu$: subspace size mean, $\lambda_{min}$: min $\lambda$ for bagging
   Symbols: $\mathcal{E}$: ensemble of $k$ $\gamma$ classifiers,
            $\mathcal{E}'$: background ensemble of $k$ $\gamma'$ classifiers,
            $f$: number of features,
            $w$: sliding window per class,
            $\varphi$: subspace of features for each of the $k$ classifiers,
            $\alpha$: accuracy for each of the $k$ classifiers,
            $\kappa$: Kappa for each of the $k$ classifiers
 1 for $\mathcal{S}_i \in \{\mathcal{S}_1, \dots, \mathcal{S}_n\}$ do
 2     $w_{class(\mathcal{S}_i)} \leftarrow$ slidingWindow($w_{class(\mathcal{S}_i)} \cup \mathcal{S}_i$)
 3     if $\mathcal{S}_1$ then                                    ▷ Ensemble initialization
 4         for $j \in \{1, \dots, k\}$ do
 5             $r \leftarrow$ random subspace size $\mu \times f + \frac{(1-\mu) \times f \times \mathcal{N}(0,1)}{2}$
 6             $\varphi_j \leftarrow$ $r$-dimensional random subspace of features
 7             $\gamma_j \leftarrow$ new base learner on $\varphi_j(\mathcal{S}_1)$
 8         end
 9     else                                                       ▷ Ensemble update
10         $\lambda \leftarrow \lambda_{min} + (1 - classRatio(class(\mathcal{S}_i), w) \cdot \lambda_{min}$   ▷ Self-adjusting $\lambda$
11         for $j \in \{1, \dots, k\}$ do
12             $\varphi_j \leftarrow$ instance $\mathcal{S}_i$ is weighted according to $Poisson(\lambda)$ on the respective subspace
13             $\alpha_j \leftarrow$ prequential accuracy of $\gamma_j$ after predicting $\varphi_j(\mathcal{S}_i)$
14             $\kappa_j \leftarrow$ prequential Kappa of $\gamma_j$ after predicting $\varphi_j(\mathcal{S}_i)$
15             $\gamma_j \leftarrow$ incremental train of $\gamma_j$ with $\varphi_j(\mathcal{S}_i)$
16         end
17         if ADWIN warning in $\gamma \in \mathcal{E}$ then
18             if $\mathcal{E}' = \emptyset$ then                     ▷ Background ensemble initialization
19                 for $j \in \{1, \dots, k\}$ do
20                     $r \leftarrow$ random subspace size $\mu \times f + \frac{(1-\mu) \times f \times \mathcal{N}(0,1)}{2}$
21                     $\varphi'_j \leftarrow$ $r$-dimensional random subspace of features
22                     $\gamma'_j \leftarrow$ new base learner on $\varphi'_j$
23                     for $\mathcal{S}_l \in \{\mathcal{S}_1, \dots, \mathcal{S}_{|w|}\}$ do
24                         $\varphi'_j \leftarrow$ instance $\mathcal{S}_l$ is weighted according to $Poisson(\lambda_{min})$ on the subspace
25                         $\gamma'_j \leftarrow$ incremental train of $\gamma'_j$ on $\varphi'_j(\mathcal{S}_l)$
26                     end
27                 end
28             else                                               ▷ Background ensemble update
29                 for $j \in \{1, \dots, k\}$ do
30                     $\varphi'_j \leftarrow$ instance $\mathcal{S}_i$ is weighted according to $Poisson(\lambda)$ on the respective subspace
31                     $\alpha'_j \leftarrow$ prequential accuracy of $\gamma'_j$ after predicting $\varphi'_j(\mathcal{S}_i)$
32                     $\kappa'_j \leftarrow$ prequential Kappa of $\gamma'_j$ after predicting $\varphi'_j(\mathcal{S}_i)$
33                     $\gamma'_j \leftarrow$ incremental train of $\gamma'_j$ with $\varphi'_j(\mathcal{S}_i)$
34                 end
35             end
36         end
37         if $(i -$ ADWIN warning timestamp $= |w|)$ then         ▷ Replace base classifiers
38             $\mathcal{E} \leftarrow Select(\mathcal{E} \cup \mathcal{E}', \alpha \cdot \kappa, \alpha' \cdot \kappa', k)$
39             $\mathcal{E}' \leftarrow \emptyset$
40         end
41     end
42 end
```

Ensemble initialization and diversity The main idea of the initialization phase (lines 3–8 in Algorithm 1) is to generate diverse base classifiers γ exploring variable r-dimensional random feature subspaces φ. Random subspaces of varied size sample the input feature space adding diversity of the classifiers.

Ensemble update The ensemble update phase (lines 10-16 in Algorithm 1) involves the incremental learning of the base classifiers. The self-adjusting λ for bagging (line

10) adjusts the λ value according to the class of the current instance S_i and the most recent distribution of the data classes in the sliding window per class w. Next, the prequential accuracy and Kappa metrics are calculated after classifying the instance S_i (lines 13–14). Finally, the base classifiers are updated with the instance S_i and its weight according to $Poisson(\lambda)$ (line 15).

Ensemble replacement Lines 17–40 in Algorithm 1 detail the creation and training of the background ensemble, and the replacement of base classifiers. The ensemble polls the current base classifiers for concept drift or warning detection using ADWIN on the respective feature subspaces. If a warning is detected in any of them (line 17), the algorithm starts learning an ensemble in the background on new sets of feature subspaces to early adapt to drifts. The background ensemble is initialized using the sliding window per class containing the most recent instances (lines 19–26), where instances on the sliding window are presented to the base classifiers in the order they were originally received. In the following set of instances, the background ensemble is updated on a purely online manner (lines 29–34). After a certain number of instances equal to the sliding window size of 1000 instances (line 37), the performance of the current and background base classifiers are compared to identify the best performing classifiers on their respective feature subspaces. The top performing base classifiers are selected to replace the ensemble (line 38). This strategy allows to incorporate the multiple classifiers dynamically and discard under-performing models based on outdated concepts.

Weighted voting to classify new instances ROSE combines its base classifiers using weighted voting, where weights are calculated based on the product of the accuracy and Kappa of each individual classifier, similar to the selection of the best performing classifiers in the ensemble replacement. The combination of the two metrics is preferred to the individual metrics for two main reasons: (1) not to introduce an excessive bias by having a metric too sensitive to skew class distributions (accuracy), and (2) Kappa may produce extremes while accuracy provides better continuity, which is preferred to multiply classifier weights.

Time and memory complexity analysis The primary ensemble comprises k base classifiers. The base classifier for ROSE is HoeffdingTree (Hulten et al., 2001), also known as VFDT, which builds a decision tree with a constant time and constant memory per instance. Thus, the ensemble initialization on the first instance $\mathcal{S}_1$ has a time complexity of $\mathcal{O}(k)$. The ensemble model update and incremental learning on a subsequent instance $\mathcal{S}_i$ has a time complexity of $\mathcal{O}(k \cdot \lambda)$ to update the k existing classifiers according to the current λ. Moreover, if the algorithm trains the background ensemble of another k classifiers, it adds a time complexity of $\mathcal{O}(k \cdot \lambda)$ but only when a drift warning is detected. Consequently, the worst-case time complexity of ROSE is $\mathcal{O}(2 \cdot k \cdot \lambda \cdot |\mathcal{S}|)$.

The memory complexity of the base classifier HoeffdingTree is $\mathcal{O}(f \cdot v \cdot l \cdot c)$ where f is the number of features, v is the maximum number of values per feature, l is the number of leaves in the tree, and c is the number of classes (Hulten et al., 2001). However, ROSE performs r-dimensional random subspace projections for each of the k classifiers, where $r \leq f$, then effectively reducing the memory complexity of HoeffdingTree to $\mathcal{O}(r \cdot v \cdot l \cdot c)$. ROSE also needs to store a sliding window per class w of the most recent instances. Therefore, the worst-case memory complexity of ROSE comprising k classifiers in the primary ensemble plus the k classifiers in the background ensemble is $\mathcal{O}((2 \cdot k \cdot r \cdot v \cdot l \cdot c) + (|w| \cdot f))$. The reduction of the feature subspaces makes ROSE competitive in time and memory complexity compared to its counterparts.

4.3 Comparison between ROSE and the Kappa updated ensemble

Our previous work introduced KUE (Cano & Krawczyk, 2020), which is also driven by the Kappa metric. Therefore, it is necessary to clearly describe the major differences between KUE and ROSE, as they are both driven by the same metric for ensemble lineup management. While KUE is a chunk-based general-purpose ensemble for drifting data streams (and also happens to do well for imbalanced data), ROSE is an online ensemble specifically designed for imbalanced data streams with dynamic imbalance ratio and concept drift, offering a number of features designed specifically to tackle these challenges. We want to highlight that all of underlying ROSE features are not simple extensions of our previous work, but are novel contributions that lead to the excellent robustness to non-stationary, imbalanced, and difficult data. The detailed comparison between the two is provided in Table 1 and the differences of the experimental studies are in Table 2.

5 Experimental study

The experimental study was designed to answer the following research questions (RQ):

- *RQ1* Can ROSE outperform state-of-the-art ensemble methods under static imbalance ratios?
- *RQ2* Can ROSE outperform state-of-the-art ensemble methods under drifting imbalance ratios?
- *RQ3* Can ROSE offer better learning capabilities under instance-level difficulties?
- *RQ4* Does ROSE exhibit improved robustness to drifting noise on imbalanced streams?
- *RQ5* Does ROSE maintain its performance when handling real-world data streams?
- *RQ6* How does each of ROSE features improve the competence of the ensemble?

Experimental setup

Algorithms Table 3 enumerates the ensemble classifiers used in the experiments. Ensembles are categorized based on their general-purpose versus class-imbalance design. All ensembles are evaluated with the same parameter settings of 10 base classifiers using HoeffdingTree as the base learner. Algorithms employing a sliding window use a buffer size of 1000 instances. No individual hyperparameter optimization was conducted for any algorithm as we believe algorithms should exhibit a robust performance off the shelf. Results reported for all algorithms/benchmarks are for a single run.

The source code for ROSE and the experimental setups for the seven experiments are publicly available on GitHub.[1] All algorithms are implemented in MOA (Bifet et al., 2010a), where their source code is publicly available, and run on an Intel Xeon CPU E5-2690v4 with 384 GB memory and CentOS 8.

Experiments 1 to 5 show the detailed results for the nine most representative ensembles (ROSE, KUE, ARF, LB, SRP, OOB, UOB, OUOB, and CSMOTE). Experiment 6 shows the aggregated results for all 31 ensembles on all benchmarks. Experiment 7 presents an ablation study of ROSE's features.

[1] Source code and experimental setup available at https://github.com/canoalberto/ROSE.

Table 1 Algorithmic differences between KUE and ROSE

	KUE	ROSE
Purpose	Data streams with concept drift	Imbalanced streams with concept drift
Training model	Chunk-based (blocks of 1000 instances)	Online (instance by instance)
Bagging	Fixed λ	Self-adjusting λ based on imbalance ratio
Instances window	One window	One window per class
Subspaces of features	Uniform distribution [1,f]	Normal distribution $\mu \times f + \frac{(1-\mu)\times f \times \mathcal{N}(0,1)}{2}$
Background ensemble	No	Yes
Base classifier replacement	One base classifier per chunk	Multiple base classifiers at any time
Base classifier selection	Kappa-only driven	Kappa and accuracy driven
Drift detector	No (concept drift is handled through dynamic classifier selection)	Yes, simultaneously using ADWIN to detect concept drift on each of the feature subspaces and through dynamic classifier selection

Performance evaluation Algorithms are compared using their prequential Kappa and AUC (Brzeziński & Stefanowski, 2017) metrics and their rank. The rank is calculated using the Friedman's test rank (Demšar, 2006). Let r_i^j be the rank of the j-th of k algorithms on the i-th of N datasets. The algorithm's rank is calculated as $R_j = \frac{1}{N}\sum_i r_i^j$.

5.1 Experiment 1: analyzing robustness to static class imbalance

Goal of the experiment This experiment was designed to address **RQ1** and evaluate the robustness of the classifiers to static class imbalance (general-purpose vs. imbalance-specific ensembles, respectively) without enforced concept drift. It is desired that any classifier designed for skewed data will display a high robustness to different levels of imbalance, i.e., output stable predictive performance regardless of the disproportion among classes. To evaluate this, we prepared six data stream benchmarks {Agrawal, AssetNegotiation, RandomRBF, SEA, Sine, Hyperplane} with static imbalance ratios of {5, 10, 20, 50, 100}. This allows us not only to gain insight into how each given classifier behaves under specific class distributions but also how it performs with increasing class imbalance. Figure 1 illustrates the performance of the selected general-purpose and imbalance-specific ensembles, respectively, with the increasing static imbalance ratio. Tables 4 and 5 present the average Kappa and AUC for each of the evaluated imbalance ratios averaged over the six data stream benchmarks, and the overall rank of the algorithms according to Friedman. Best results in the tables are presented in bold font.

Comparison with class-imbalance ensembles ROSE was compared with OOB, UOB, OUOB, and CSMOTE. We can see that UOB displays the worst robustness to increasing imbalance ratio, showing significant drops in performance when IR becomes higher than 20 (with the exception of AssetNegotiation, SEA, and Sine datasets). This can be explained by the fact that with increased IR, there are increasing less minority instances in each batch. As UOB uses undersampling, it tries to reduce the size of the majority class. This

Table 2 Experimental study differences between KUE and ROSE

	KUE	ROSE
Ensemble algorithms compared	15 general-purpose	21 general-purpose9 imbalanced-specific
(Traditional) standard datasets	Yes (13 datasets)	No
Class-balanced generators without concept drift	Yes (20 generators)	No
Class-balanced generators with concept drift	Yes (25 generators)	No
Imbalanced datasets	Yes (7 datasets)	Yes (24 datasets)
Imbalanced generators with static imbalance ratio	Yes (20 generators)	Yes (36 generators)
Imbalanced generators with dynamic imbalance ratio	Partial (6 generators)	Yes (12 generators)
Instance-level difficulties in imbalanced data	No	Yes (39 datasets)
Drifting noise and imbalance ratio	No	Yes (12 datasets)

leads to having a smaller training set that hinders the online learning capabilities of UOB. This is especially crucial when high imbalance ratio is combined with concept drift, since the small sample size will reduce the chances of quick recovery from changes. Even when instances from new concept will arrive their numbers will be continually reduced, leading to much more prolonged adaptation process. OOB, the counterpart of UOB, displays much better robustness and stability to varying imbalance ratio. However, especially for the hyperplane dataset, we can see a significant drop in performance when handling higher imbalance ratios. The hyperplane generator uses sudden drifts internally, which allows us to understand the reason behind such a drop. Oversampling of instances in case of high imbalance ratio, leads to an oversaturation of the classifier with instances from old concepts. This may significantly reduce the forgetting capability of any underlying classifier, thus leading to lower reactivity to concept drift. CSMOTE and OUOB display very good robustness to increasing imbalance ratio. Sadly, their predictive power is lowest from all methods, proving that robustness on its own is not enough. The proposed ROSE combines robustness with best predictive performance, showing that ROSE is capable of handling even high class imbalance. This is especially desired in various real-world continual and streaming problems, where we do not know how the imbalance ratio may change over time and we need a classifier that can offer stable performance regardless of characteristics of incoming data. ROSE is capable of outperforming all of the methods on most of the datasets. On a few of them (notably AssetNegotiation and SEA) ROSE returns comparable performance to reference methods, but never is outperformed by them.

Comparison with general-purpose ensembles ROSE was compared with KUE, ARF, LB, and SRP. Surprisingly, general-purpose ensembles display similar robustness to increasing imbalance ratio as skew-insensitive approaches discussed previously. This can be explained by the diversity of base learners employed in those ensembles. Using mutually complimentary learners leads to a reduction in bias towards the majority class and better management of even higher class disproportion. This shows that the idea of classifier diversity, strongly explored in ROSE, is a key factor in designing effective ensemble learners for imbalanced data streams. However, SRP has problems with stability on SEA, Hyperplane, and Sine datasets, where increasing imbalance ratio leads to higher variance in its results. ROSE outperforms all of the reference ensemble methods in terms of stability and predictive power on all imbalance ratios.

Table 3 Algorithms employed in the experimental evaluation

	References	Algorithm
General-purpose ensembles	Cano and Krawczyk (2020)	KUE: Kappa Updated Ensemble.
	Wang et al. (2003)	AWE: Accuracy Weighted Ensemble.
	Brzeziński and Stefanowski (2011)	AUE1: Accuracy Updated Ensemble 1.
	Brzeziński and Stefanowski (2014b)	AUE2: Accuracy Updated Ensemble 2.
	Kolter and Maloof (2007)	DWM: Dynamic Weighted Majority.
	Gomes and Enembreck (2014)	SAE2: Social Adaptive Ensemble 2.
	Jaber et al. (2013)	DACC: Dynamic Adaptation to Concept Changes.
	Jaber et al. (2013)	ADACC: Anticipative and Dynamic Adaptation to Concept Changes.
	Gomes et al. (2017)	ARF: Adaptive Random Forest.
	de Carvalho Santos et al. (2014)	ADOB: Adaptable Diversity-based Online Boosting.
	de Barros et al. (2016)	BOLE: Boosting-like Online Learning Ensemble.
	Bonab and Can (2018)	GOOWE: Geometrically Optimum and Online-Weighted Ensemble.
	Van Rijn et al. (2015)	HEB: Heterogeneous Ensemble BLAST.
	Bifet et al. (2010b)	LB: Leveraging Bagging Adwin.
	Pelossof et al. (2009)	OCB: Online Coordinate Boosting.
	Oza (2005)	OBA: Online Bagging.
	Bifet et al. (2009)	OBAD: Online Bagging with ADWIN.
	Bifet et al. (2009)	OBASHT: Online Bagging Adaptive-Size Hoeffding Tree.
	Oza (2005)	OBO: Online Boosting.
	Oza (2005)	OBOA: Online Boosting with ADWIN.
	Gomes et al. (2019a)	SRP: Streaming Random Patches.
Class imbalance ensembles	This paper	ROSE: Robust Online Self-Adjusting Ensemble.
	Wang et al. (2016)	OOB: Oversampling Online Bagging.
	Wang et al. (2016)	UOB: Undersampling Online Bagging.
	Wang and Pineau (2016)	OSMOTE: Online Continuous Synthetic Minority Oversampling Bagging.
	Wang and Pineau (2016)	OUOB: Online Undersampling and Oversampling Bagging.
	Bernardo et al.,(2020b)	CSMOTE: Continuous Synthetic Minority Oversampling.
	Wang and Pineau (2016)	OADA: Online AdaBoost.
	Wang and Pineau (2016)	OADAC2: Online AdaC2.
	Wang and Pineau (2016)	ORUS: Online Random Undersampling Boosting.
	Bernardo et al. (2020a)	IRL: Incremental Rebalancing Learning.

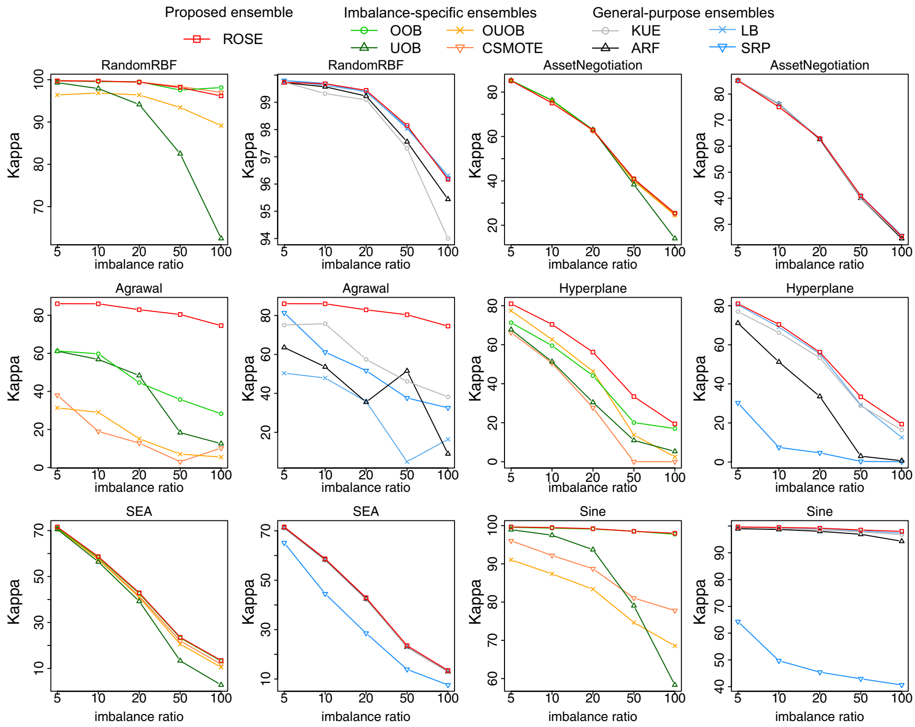


Fig. 1 Robustness to class imbalance ratios (Kappa). The first group of algorithms includes imbalanced-specific ensembles (ROSE vs. OOB, UOB, OUOB, CSMOE). The second group of algorithms includes general-purpose ensembles (ROSE vs. KUE, ARF, LB, SRP)

Table 4 Kappa averages over the six stream benchmarks on static class imbalance ratios

Imbalance ratio	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
5	**79.80**	73.87	73.73	73.01	63.71	75.17	72.10	67.12	65.17
10	**72.46**	68.07	64.49	65.84	50.00	68.56	63.87	59.56	56.54
20	**64.65**	59.12	54.04	57.15	42.58	57.67	53.34	49.36	47.91
50	**54.25**	47.65	44.86	42.18	33.55	45.33	34.80	35.64	35.26
100	**47.07**	40.52	33.87	37.40	29.02	40.00	22.28	28.68	31.84
Average	**63.65**	57.85	54.20	55.12	43.77	57.35	49.28	48.07	47.34
Rank	**2.00**	4.84	5.46	3.86	5.81	3.17	6.40	7.14	6.31

5.2 Experiment 2: analyzing robustness to drifting class imbalance

Goal of the experiment This experiment was designed to address **RQ2** and evaluate the robustness of classifiers to a scenario with drifting imbalance ratio. Concept drift may also affect the class distributions, changing the learning difficulty over time. While many existing methods are designed to cope well with the static imbalance ratio present during the training phase, they lack effective mechanisms that allow for skew-insensitive adaptation to time-varying disproportions between classes. To evaluate this, we prepared six data stream benchmarks {Agrawal, AssetNegotiation, RandomRBF, SEA, Sine, Hyperplane} with drifting imbalance ratio representing first increasing and then decreasing imbalance ratio {5, 10, 20, 100, 20, 10, 5}. This allows us to analyze not only how each analyzed classifier is able to cope with class imbalance, but also how adaptive it is to the dynamic imbalance ratio occurrences. Figure 2 illustrates the prequential Kappa over time for the selected general-purpose and imbalance-specific ensembles. Tables 6 and 7 present the average Kappa and AUC for each of the drift types on the six generators, and the rank of the algorithms.

Comparison with class-imbalance ensembles We can see that while all methods can handle drifting imbalance ratios, their main differences lie in how strongly imbalance drift affects them and how quickly they can recover from the imbalance drift. It is interesting to see that both over- and undersampling based ensemble methods perform significantly worse in the case of drifting imbalance ratios. While OOB (on Kappa) and UOB (on AUC) are the best performing method from all class-imbalance ensembles (despite their lack of explicit drift handling mechanisms), their hybrid OUOB counterpart often falls short to most of the methods. This can be explained by its inability to effectively switch between different resampling approaches that leads to slower recovery from changes in imbalance ratio and drifts. In all of six benchmarks CSMOTE (with ARF) shows the biggest drops in performance among all methods when imbalance ratio increases. This can be explained by the inability of the online *k*-nearest neighbor-based oversampling method to properly model the majority class with the increasing class disproportions. This forces CSMOTE to introduce artificial instances to wrong classes, not being able to adapt quickly enough to sudden changes. Therefore, we can conclude that SMOTE-based solutions are not suitable to handle drifting imbalance ratios in data streams, especially when imbalance ratio is increasing over time. ROSE offers superior performance to all four of class-imbalance ensembles, showing both smaller drops in performance when imbalance ratio drift occurs, but also displaying quicker recovery rates after the drift, leading to faster adaptations to new concepts with different class proportions. This shows that ROSE offers great capabilities of adaptation to drifting and imbalanced data streams, far outperforming state-of-the-art skew-insensitive solutions.

Comparison with general-purpose ensembles We can see that general-purpose ensemble approaches cannot cope with the imbalance drift and require significant time to recover from changes in class ratios and thus offer lower recovery rates than ROSE. Even if their performance on a fully learned concept is satisfactory, they require more instances than ROSE to achieve this performance and capture the properties of concept with new imbalance ratio. It is interesting to notice that in case of the Kappa metric, KUE and LB ensemble classifiers work better than skew-insensitive solutions discussed earlier. This shows that ensemble approaches can effectively utilize their diversity to offer faster adaptation to sudden changes. Skew-insensitive solutions (especially CSMOTE) do not emphasize diversity during their base classifier update, thus leading to slower adaptation to drifts. ROSE combines the advantages of both approaches, combining fast adaptation via promoted diversity of base classifiers with skew-insensitive mechanisms offering robustness to static and drifting imbalance ratios.

5.3 Experiment 3: analyzing robustness to instance-level difficulties

Goal of the experiment This experiment was designed to address **RQ3** and evaluate the robustness of the data stream classifiers to instance-level difficulties (Brzeziński et al.,

Table 5 AUC averages over the six stream benchmarks on static class imbalance ratios

Imbalance ratio	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
5	**88.33**	85.10	84.38	84.20	78.90	85.93	85.87	81.06	80.55
10	**83.46**	81.06	78.60	79.50	71.10	81.28	82.16	75.93	75.19
20	**78.88**	75.67	73.21	74.71	67.26	75.46	77.93	70.59	70.56
50	**73.84**	70.29	69.37	68.38	63.31	69.80	72.65	64.40	65.11
100	**70.85**	67.43	65.09	66.48	61.63	67.59	70.25	61.62	63.61
Average	**79.07**	75.91	74.13	74.65	68.44	76.01	77.77	70.72	71.01
Rank	**2.04**	5.33	6.04	4.40	6.31	2.97	3.57	7.53	6.80

2021). We used two imbalance generators[2] to create scenarios with the presence of borderline or rare instances, as well as with both types at once, while experiencing a split of the cluster. Difficult instances were created for the minority class to present a significantly more challenging scenario. We evaluated their influence on classifiers on their own and combined with medium IR equal to 10 and high IR equal to 100. Borderline instances are challenging to classifiers as they lie in the uncertainty area of the decision space and strongly impact the induction of the classification border. Rare instances are overlapping with the majority class, leading to small sample and sparse subconcepts created within the minority class. So far only few works discussed the idea of analyzing the instance-level difficulty in the context of data streams and concept drift, while this issue is of crucial importance in imbalanced data domain. Figure 3 and Tables 8 and 9 show the performance of the ensemble methods on data streams with various ratios of instance-level difficulties injected into the stream.

Comparison with class-imbalance ensembles All four reference methods were designed to learn from imbalanced data stream, but only by considering the global imbalance ratio. One can see that none of the state-of-the-art skew-insensitive classifiers display any additional robustness to the increasing number of both borderline and rare instances. Out of the two types, rare instances pose much more difficulty to all methods. UOB and OOB cannot effectively handle the borderline instances, as their sampling methods only increase the overlap on the boundary, leading to a decreased certainty between the classes. This is especially visible in the case of OOB, as it may enhance the presence of borderline instances that are overlapping with the majority class, leading effectively to higher error on both classes. In case of rare instances, UOB, OOB, and OUOB cannot efficiently clean their neighborhoods or oversample them in a meaningful manner, leading to significant drops in predictive performance with an increased ratio of difficult instances. Interestingly, CSMOTE displays much better performance than random resampling approaches, which is particularly visible on the rare instances. This can be attributed to the fact that rare instances create the small sample size problem, not offering enough information for classifiers to efficiently capture their properties. CSMOTE indirectly increases the density of instances in their neighborhood, leading to their increased importance during the classifier training. ROSE can handle borderline and rare instances more effectively than those four classifiers, due to its capabilities of increased exposure to difficult instances. The proposed self-adjusting λ allows for displaying borderline and rare instances multiple times to base classifiers, increasing their adaptation to local data characteristics. This shows that

[2] Imbalance generators available at https://github.com/dabrze/imbalanced-stream-generator.

Table 6 Kappa averages over the six stream benchmarks on drifting class imbalance ratios

IR drift	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
Sudden	**62.02**	53.45	50.93	57.28	40.46	52.47	44.32	48.38	47.60
Gradual	**57.87**	53.03	47.23	51.62	38.38	51.22	43.71	45.13	44.82
Average	**59.95**	53.24	49.08	54.45	39.42	51.85	44.02	46.75	46.21
Rank	**2.23**	4.57	5.64	3.18	7.59	3.64	6.25	5.86	6.05

Bold values indicates best results

Table 7 AUC averages over the six stream benchmarks on drifting class imbalance ratios

IR drift	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
Sudden	**78.38**	73.63	72.45	75.36	66.78	74.59	75.83	71.20	71.07
Gradual	**76.64**	73.68	70.78	72.90	65.94	74.28	75.74	69.86	69.78
Average	**77.51**	73.65	71.62	74.13	66.36	74.43	75.78	70.53	70.43
Rank	**2.36**	5.09	6.09	3.64	8.05	3.55	3.45	6.18	6.59

Bold values indicates best results

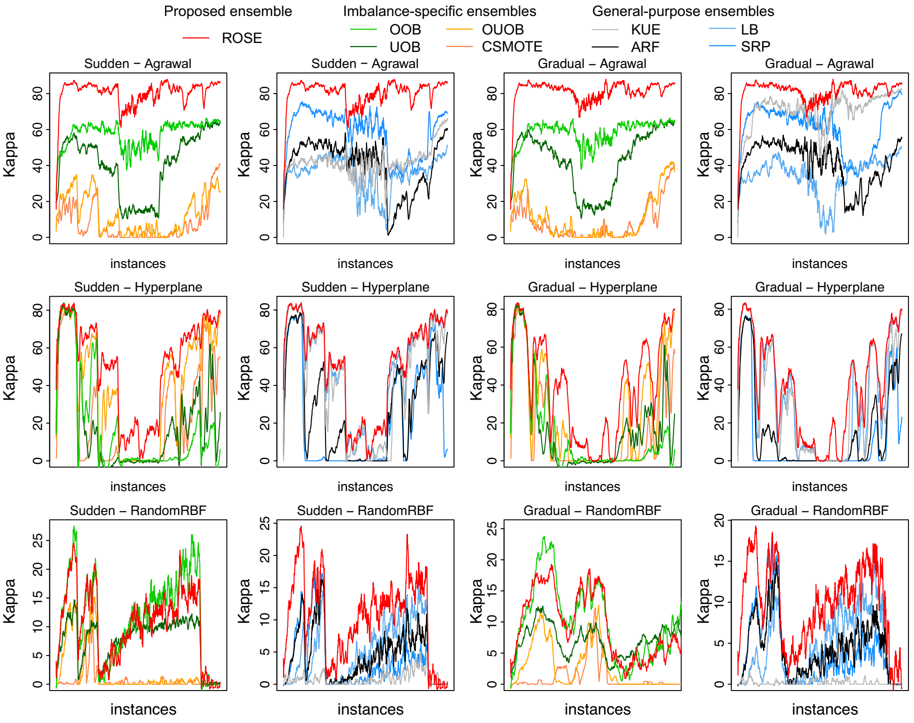


Fig. 2 Prequential Kappa on drifting class imbalance ratios. The first group of algorithms includes imbalanced-specific ensembles (ROSE vs. OOB, UOB, OUOB, CSMOE). The second group of algorithms includes general-purpose ensembles (ROSE vs. KUE, ARF, LB, SRP)

ROSE displays high robustness not only to class imbalance, but also data irregularities and instance-level difficulties.

Comparison with general-purpose ensembles Results show that when only instance-level difficulties are present the general-purpose ensembles display performance and robustness similar to their skew-insensitive counterparts. This is a very interesting observation, as it shows that instance-level difficulties pose completely different challenges to learning systems than imbalanced data. And while they can be a part of the imbalanced problem, they can pose as a difficult problem on their own. Surprisingly, KUE which in other experiments was one of the best performing methods, here is among the most affected by increasing ratios of difficult instances. This shows that while KUE displays great performance on standard and cleaned data streams, it cannot be effectively applied to data streams with irregularities. Here the second most robust method, after the proposed ROSE, is SRP. By training classifiers on random feature subspaces, SRP alters the instance-level characteristics (by altering distances between instances), which may lead to better capturing of rare objects. This observation applies to ROSE, as our approach uses a similar mechanism. ARF also displays good robustness to instance-level difficulties. As ARF trains its base learners on both subsets of instances and features, some of the learners in its pool are going to be more affected by difficult instances than the others. However ARF, unlike ROSE, does not offer any mechanisms for increasing the exposure to difficult instances—which in the end results in it having worse predictive performance on difficult data streams than the proposed ROSE.

5.4 Experiment 4: analyzing robustness to drifting noise on imbalanced streams

Goal of the experiment This experiment was designed to address **RQ4** and evaluate the robustness of classifiers to a scenario with the presence of drifting noise affecting the features. To make this scenario more realistic and at the same time challenging, we combined it with the dynamic imbalance ratio examined in Experiment 2. This way, both noise and IR drift over time. Feature distribution affects the definition of class boundaries, thus leading to a more challenging skewed learning scenario with higher degree of overlap between classes. Features affected by noise should be discarded by a classifier (Krawczyk & Cano, 2018), as they may display a highly negative impact on the learning from minority classes. We used the six generators from Experiment 2 with drifting imbalance ratio, further injecting noise into a varying ratio of features $\{10\%, 20\%, 30\%, 40\%\}$. Figure 4 shows the plots depicting robustness to increasing noise ratio and Fig. 5 depicts the comparison between ROSE and best performing ensemble methods under drifting imbalance ratio and 20% of features being subject to noise. This is further accompanied by Tables 10 and 11 showing average Kappa and AUC metrics under varying noise levels over the six data stream benchmarks.

Comparison with class-imbalance ensembles We can see that the reference streaming classifiers, while able to work with class imbalance as the only learning difficulty, they do not possess any mechanisms for coping with the presence of noise in the stream. Neither over- or undersampling based solutions can remove noisy or redundant features, leading to noise significantly impairing their learning. UOB and CSMOTE are the ones that are impacted most negatively by noise. While CSMOTE performance can be easily explained (SMOTE uses Euclidean distance to generate artificial instances, thus noise decreases the quality of oversampling), the poor performance of UOB comes as a surprising observation. Random undersampling does not use any feature-based information, thus the noise

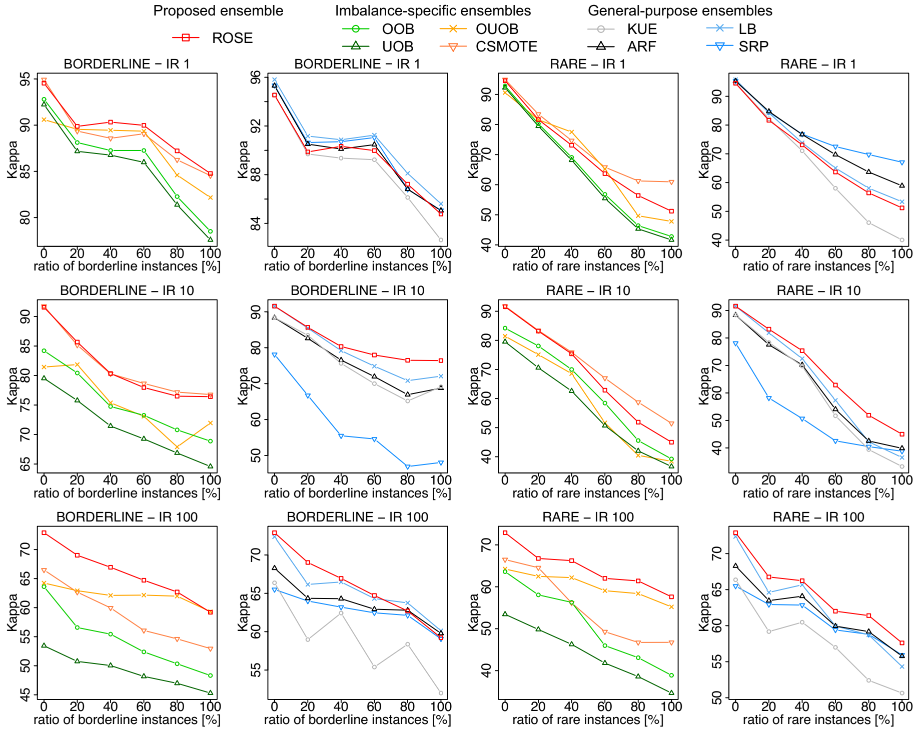


Fig. 3 Robustness to borderline and rare instances under different class imbalance (Kappa). The first group of algorithms includes imbalanced-specific ensembles (ROSE vs. OOB, UOB, OUOB, CSMOE). The second group of algorithms includes general-purpose ensembles (ROSE vs. KUE, ARF, LB, SRP)

Table 8 Kappa over the stream benchmarks on instance-level difficulties

Instance-level difficulty	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
Borderline – IR 1	89.45	88.60	89.71	**90.47**	89.94	86.04	85.18	87.62	88.79
Borderline – IR 10	81.41	75.29	75.86	78.99	58.33	75.39	71.24	75.27	**81.64**
Borderline – IR 100	**65.92**	58.92	63.74	65.53	62.73	54.45	49.11	62.12	58.80
Rare – IR 1	70.10	65.32	74.84	71.72	**77.68**	64.69	63.71	68.73	73.52
Rare – IR 10	68.31	60.14	62.09	63.66	51.48	62.58	57.00	59.27	**71.39**
Rare – IR 100	**64.49**	57.68	61.79	62.61	60.91	50.99	44.09	60.26	54.99
Borderline + Rare – IR 1	72.90	71.51	75.31	73.97	**76.27**	70.12	68.93	74.85	73.13
Borderline + Rare – IR 10	70.24	61.33	61.94	65.44	48.13	65.46	60.29	62.10	**70.37**
Borderline + Rare – IR 100	**60.99**	55.31	60.39	60.93	59.83	49.43	45.38	59.17	54.81
Average	**71.47**	65.71	69.53	70.23	65.02	63.95	60.16	67.58	69.70
Rank	**2.72**	6.31	3.50	2.82	4.76	6.85	8.44	5.51	4.10

Bold values indicates best results

must negatively impact the bagging procedure itself. UOB does not have any explicit drift handling mechanism, which may lead to performance degradation over time under non-stationary noise. At the same time, ROSE shows robustness to varying levels of noise,

Table 9 AUC over the stream benchmarks on instance-level difficulties

Instance-level difficulty	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
Borderline – IR 1	94.77	94.34	94.90	**95.27**	95.01	93.07	92.64	93.86	94.44
Borderline – IR 10	92.55	86.31	85.77	88.09	77.23	92.36	91.47	85.46	**94.29**
Borderline – IR 100	85.39	83.69	81.86	83.24	81.07	88.39	90.37	80.96	**93.35**
Rare – IR 1	85.11	82.70	87.47	85.90	**88.89**	82.39	81.90	84.41	86.80
Rare – IR 10	82.93	78.84	79.24	80.38	74.44	81.92	80.92	77.79	**85.35**
Rare – IR 100	82.84	81.52	80.69	81.45	80.14	84.09	84.43	79.90	**87.74**
Borderline + Rare – IR 1	86.52	85.82	87.72	87.04	**88.20**	85.12	84.53	87.50	86.62
Borderline + Rare – IR 10	84.13	78.91	78.57	80.43	72.93	84.22	83.31	78.63	**86.32**
Borderline + Rare – IR 100	81.71	79.75	79.99	80.45	79.54	82.81	85.59	79.34	**88.15**
Average	86.24	83.44	84.04	84.65	81.94	85.96	85.95	83.07	**89.30**
Rank	3.56	6.54	5.22	4.44	6.01	4.68	5.28	7.01	**2.26**

Bold values indicates best results

significantly outperforming reference solutions. It is very important to notice that the difference between ROSE and reference classifiers in terms of their predictive power (measured both as prequential Kappa and prequential AUC) is much more significant in Experiment 2 (which used the same data streams but without noise). This shows that reference skew-insensitive classifiers are strongly impacted by noise, while ROSE suffers significantly lower drops in performance regardless of the noise level present. By analyzing Fig. 5 we can see that the combination of feature noise and drifting imbalance ratio becomes even more challenging for reference classifiers. We observe that with the increasing imbalance ratio the negative impact of noise strengthens. This can be attributed to the impact of noise on both majority and minority classes. Their distributions become shifted, leading to increasing overlapping and more difficult borderline instances. With increasing imbalance ratio, we have less and less safe minority instances, thus negatively impacting the adaptation of classifiers to drifts. ROSE is capable of removing noisy features and effectively utilizing feature subspaces to train noise-insensitive classifiers that improve its adaptation to concept drift and incorporation of new, useful knowledge into ROSE ensemble.

Comparison with general-purpose ensembles While standard ensembles do not offer high robustness to class imbalance, they are capable of handling feature noise as well as skew-insensitive streaming classifiers. In some cases (e.g., Agrawal or Hyperplane generators) we observe that general-purpose ensembles display higher robustness to noise than their skew-insensitive counterparts. This can be explained by some specific mechanisms embedded in the ensembles that allow to handle implicitly some noise. KUE uses a combination of feature subspaces (like ROSE) that allow to filter out noisy features from being included in newly trained classifiers. Additionally, KUE uses an abstaining mechanism that removes the most uncertain classifiers from the voting procedure. If a classifier is highly affected by noisy features, its certainty will become closer to a random classifier. Abstaining mechanism will temporarily switch off such a classifier, leading to better a response to noisy data streams. ARF also uses feature subspaces, but in each decision tree node, leading to a reduced probability of noisy features becoming the backbone of its base classifiers. ROSE displays the highest robustness to any level of noise due to its capability of using feature subspaces combined with the use of a background ensemble to explore new random subspaces without noise.

Table 10 Kappa averages over the six stream benchmarks with drifting noise

Noise level (%)	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
0	**59.95**	53.24	49.08	54.45	39.42	51.85	44.02	46.75	46.21
10	**52.93**	47.17	42.01	45.90	33.69	47.55	34.32	41.15	39.49
20	**50.18**	43.52	38.91	43.91	30.11	43.95	31.00	39.65	37.83
30	**43.09**	36.34	30.97	35.93	21.94	36.34	24.13	33.53	31.85
40	**36.54**	29.80	24.98	29.83	19.55	31.33	22.08	27.71	25.26
Average	**48.54**	42.01	37.19	42.00	28.94	42.20	31.11	37.76	36.13
Rank	**1.70**	4.43	6.02	3.75	7.61	3.57	6.43	5.40	6.09

Bold values indicates best results

5.5 Experiment 5: real-world datasets

Goal of the experiment This experiment was designed to address **RQ5** and evaluate the predictive power of ROSE on 24 real-world imbalanced and drifting data streams. The four previous experiments focused on analyzing the robustness of ROSE to various learning difficulties present in imbalanced data streams, allowing us to gain deeper insight into why ROSE is a highly effective and well-rounded classifier. We used data stream generators to have a full control over the created data and to simulate specific challenging scenarios. 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 strength or frequency. Their imbalance ratio changes over time, while concept drift may oscillate among different types with varying speed. Therefore, evaluating ROSE against reference methods on real-world data streams is a crucial step towards proving effectiveness of our classifier. The real-world data streams employed in the study are popular benchmarks for streaming classifiers. This will allow readers to position the effectiveness of the methods among other studies on data streams, even those not focusing on class imbalance. Figure 6 shows the plot depicting the prequential Kappa over time and Table 12 presents the prequential metrics (accuracy, Kappa, and AUC) averaged across all 24 datasets and the ranks.

Unique nature of real-world imbalanced data streams It is important to highlight a crucial difference between artificial and real-world imbalanced data streams. All generators are probabilistic and base the generation of instances on prior probability taken from current parametric imbalance ratio. With the change of imbalance ratio, the underlying probability of generating instance from minority and majority classes also change. Still, their appearance in the stream is dictated strictly by these priors, leading to bounded time windows within which minority and majority instances appear. This does not hold for real-world imbalanced data streams, as they were collected following some specific phenomenon observations and are not bounded with such clear probabilistic mechanisms. Therefore, there are no uniform characteristics to be observed over extended periods of time and the arrival of class-specific instances is dictated by how the observations were collected. This poses unique challenges to imbalanced data stream mining, such as latency with which instances from a specific class arrive, or extended periods of time when instances from only a single class appear. Such a formulation of data streams is much more challenging for existing streaming classifiers, as it makes blind adaptation to every new instance

Table 11 AUC averages over the six stream benchmarks with drifting noise

Noise level (%)	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
0	**77.51**	73.65	71.62	74.13	66.36	74.43	75.78	70.53	70.43
10	**75.01**	70.85	68.30	70.25	63.63	73.45	73.08	68.38	67.30
20	**73.41**	69.09	67.05	69.36	62.05	71.72	71.82	67.66	66.58
30	**70.04**	65.46	63.09	65.31	58.55	68.71	69.36	64.99	63.63
40	**66.81**	62.39	60.22	62.37	57.53	66.65	66.29	62.04	60.49
Average	**72.56**	68.29	66.06	68.28	61.63	70.99	71.27	66.72	65.69
Rank	**2.39**	5.16	6.56	4.47	8.07	2.98	2.91	5.71	6.75

Bold values indicates best results

insufficient. Instead, it forces guided adaptation when useful knowledge is retained to avoid the forgetting of specific classes. This makes real-world imbalanced data streams akin to continual / lifelong learning, where robustness to catastrophic forgetting becomes a key issue. Such benchmarks allow us to gain additional insights into ROSE and reference classifiers, allowing to evaluate them under these unique and challenging conditions.

Comparison with class-imbalance ensembles It is very interesting to see that skew-insensitive methods deliver inferior results to ROSE on most of the datasets with respect to Kappa. This shows us that these reference methods cannot handle compound real-world problems that are characterized by mixed drifts and varying class imbalance ratios. The especially low performance of OOB and UOB can be attributed to their online nature. They adapt their resampling strategy to the newly arriving instance, not being able to retain any memory of the previously seen concepts. When subject to a high latency of instances from a certain class (i.e., one of classes not appearing for a certain period) they will become highly skewed and cannot effectively recover from such an extreme imbalance ratio. OUOB and CSMOTE display much better performance, showing that their more complex mechanisms (hybrid resampling for OUOB and guided oversampling for CSMOTE) are able to capture more compound characteristics of the real-world streams. ROSE is capable of outperforming all four reference methods, which we contribute to storing buffers for each class independently, making ROSE robust to catastrophic forgetting in such latency scenarios.

Comparison with general-purpose ensembles When analyzing prequential accuracy, we can see that SRP is the best performing method. However, when analyzing skew-insensitive metrics such as Kappa we can see that ROSE outperforms every single ensemble method. This shows how using accuracy as a metric may lead to false conclusions and how existing ensemble methods can be biased towards the majority class. This is especially visible in case of ARF and LB that achieve great prequential accuracy on all benchmarks, but a significantly lower Kappa. ROSE offers flexibility to various challenges present in real-world data, delivering stable performance. It is important to note that ROSE always achieves high ranks, while reference ensembles are characterized by a high variance in their performance, making them impractical for deployment in new, unknown domains.

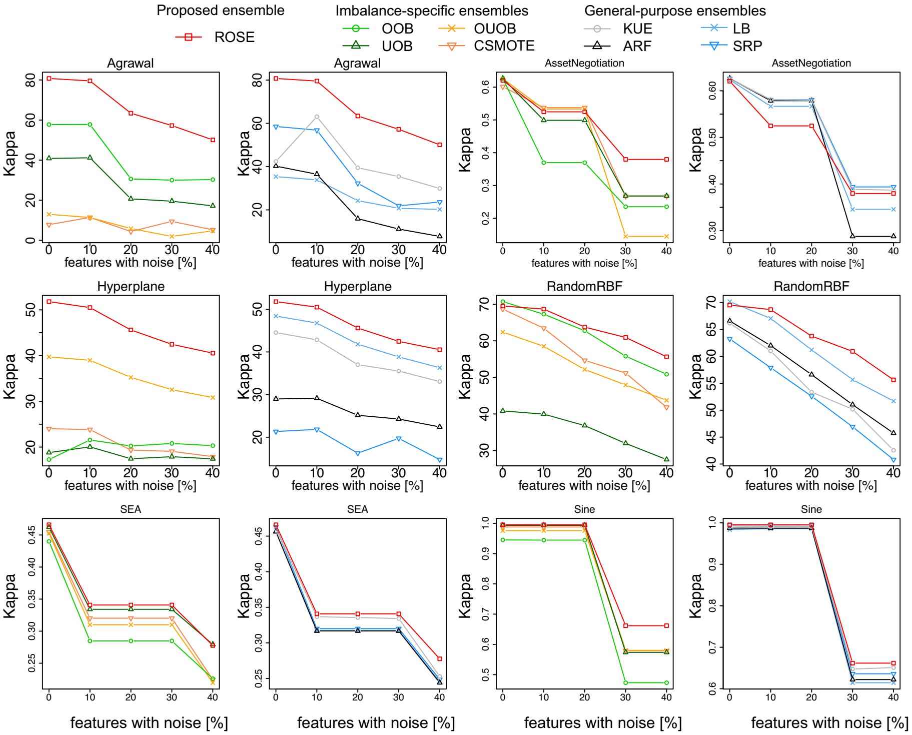


Fig. 4 Robustness to different levels of noise with sudden drift (Kappa). The first group of algorithms includes imbalanced-specific ensembles (ROSE vs. OOB, UOB, OUOB, CSMOE). The second group of algorithms includes general-purpose ensembles (ROSE vs. KUE, ARF, LB, SRP)

5.6 Experiment 6: overall comparison and statistical analysis

Goal of the experiment The previous experiments presented a detailed evaluation of ROSE against selected reference methods on imbalanced and drifting data streams, as well as on real-world benchmarks. Due to the readability of the results, we compared ROSE with four top performing skew-insensitive ensembles and four top-performing general-purpose ensembles. However, each experiment was actually run using all 30 ensembles listed in Table 3, resulting in the biggest study of learning from imbalanced data streams conducted so far. In this section, we present the summary of results for all of 30 ensembles, including non-parametric and Bayesian statistical analyses. Tables 13 and 14 present the results for all classifiers according to prequential Kappa and AUC. Results are divided into five major groups (following the previous five experiments) and averaged over all benchmarks belonging to a given group. The meta rank represents the rank of the ranks across all of the benchmarks. Table 15 shows the averages and ranks of the runtime (seconds per 10,000 instances) and memory consumption (RAM-hours) of all algorithms.

To evaluate the statistical significance of the results over multiple datasets, Figs. 7 and 8 present the visualization of Bonferroni-Dunn tests (multiple algorithm comparison) for both metrics using a p-value of 0.01. The algorithms are sorted according to their rank. The critical distance (CD) interval indicates the difference of ranks between algorithms to be

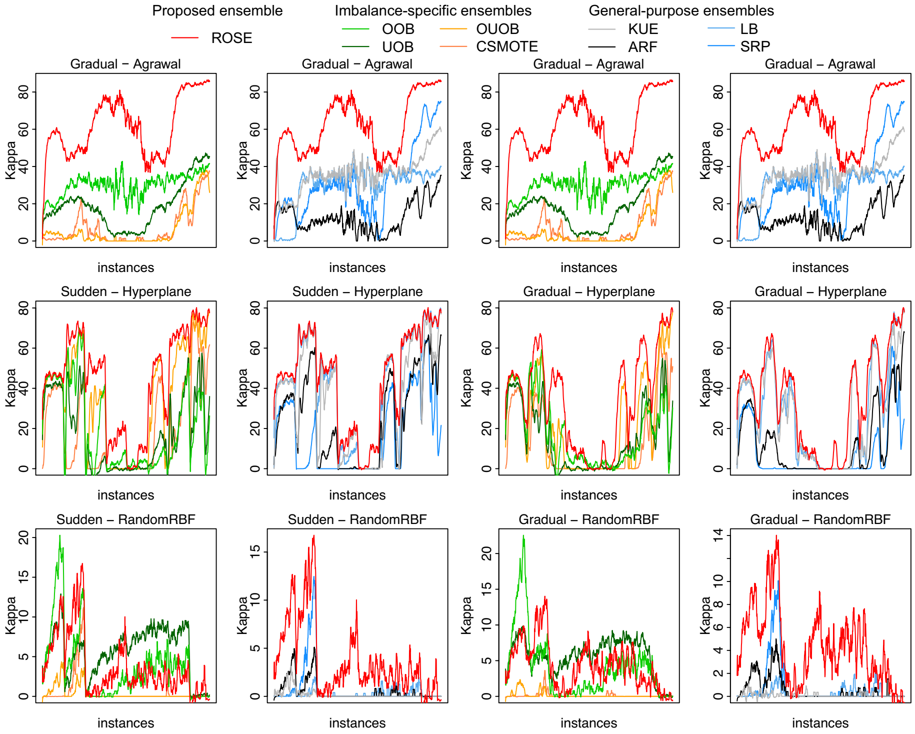


Fig. 5 Prequential Kappa on drifting noise (noise on 20% of the features). The first group of algorithms includes imbalanced-specific ensembles (ROSE vs. OOB, UOB, OUOB, CSMOE). The second group of algorithms includes general-purpose ensembles (ROSE vs. KUE, ARF, LB, SRP)

considered statistically different. Furthermore, Fig. 9 depicts the visualizations of Bayesian rank test (pairwise algorithm comparison) between ROSE and best performing skew-insensitive method (OOB) and best performing general-purpose ensemble method (LB). This test returns probabilities that one model will outperform the other based on measured performance. The top region indicates practical equivalence, while the lower right portion denotes better performance for ROSE and the remaining side for the opposing algorithm.

Comparison with reference classifiers We observe that ROSE achieves the best performance and ranks regardless of the benchmark (from five major groups) and outperforms

Table 12 Performance on 24 real-world datasets

Dataset	ROSE	KUE	ARF	LB	SRP	OOB	UOB	OUOB	CSMOTE
Avg. Accuracy	86.81	78.83	85.89	84.85	**88.69**	79.60	51.24	83.30	81.44
Avg. Kappa	**62.48**	39.73	58.94	57.92	61.85	50.14	23.35	59.08	57.84
Avg. AUC	**88.96**	84.71	86.33	86.05	87.42	87.74	86.69	86.55	88.52
Rank Accuracy	3.42	6.10	3.81	4.13	**1.75**	6.58	8.63	5.56	5.02
Rank Kappa	**2.67**	6.75	4.69	4.79	3.08	5.58	7.88	5.13	4.44
Rank AUC	**2.79**	6.81	5.58	5.33	3.71	4.52	6.69	5.65	3.92

Bold values indicates best results

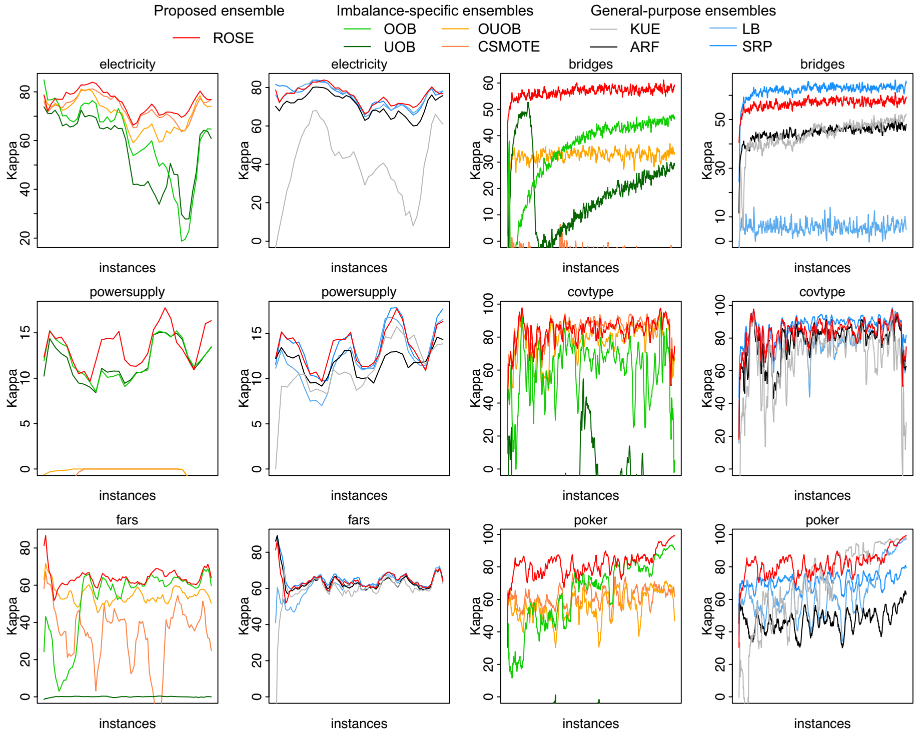


Fig. 6 Prequential Kappa on real-world datasets. The first group of algorithms includes imbalanced-specific ensembles (ROSE vs. OOB, UOB, OUOB, CSMOE). The second group of algorithms includes general-purpose ensembles (ROSE vs. KUE, ARF, LB, SRP)

in a statistically significant manner all of 30 methods. It is important to note that ROSE delivers a very stable performance and high ranks over all benchmarks. This cannot be said about any of the other methods that are subject to high variation depending on the benchmark. This is further augmented by the observation of behaviors on Kappa and AUC. While ROSE always achieves best rank on both metrics, the second-best performing method for Kappa is LB, while for AUC is OOB. At the same time LB for AUC is ranked as the fourth classifier. This showcases that ROSE is a well-rounded and flexible classifier, capable of dealing with various learning challenges present in imbalanced and drifting data streams. This allows ROSE to be efficiently deployed on a data stream with no prior knowledge of its characteristics, imbalance ratio, or presence of noise. Due to its self-adjusting nature ROSE can tackle any emerging and unknown difficulties, while remaining robust to skewed distributions. ROSE exhibits a runtime similar to LB and ARF while improving the Kappa and AUC. ROSE requires additional memory compared to KUE to train and store the background ensemble. While DACC is the fastest and has the lowest memory consumption its Kappa and AUC ranks are among the worst. On the other hand, OSMOTE and OUOB show the slowest runtime and the largest demand of memory resources.

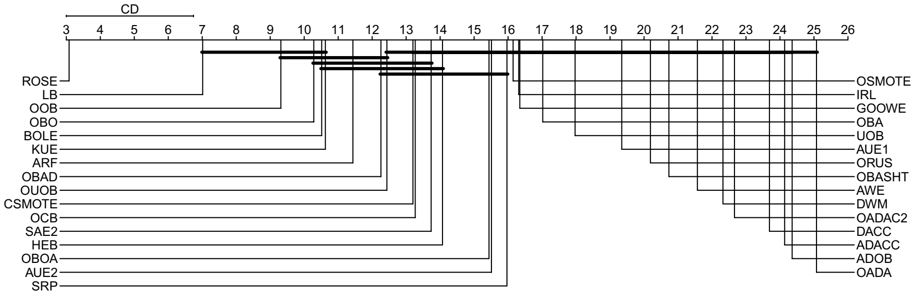


Fig. 7 Bonferroni-Dunn statistical analysis on Kappa

5.7 Experiment 7: ablation study

Goal of the experiment Previous experiments allowed us to establish the effectiveness and robustness of ROSE when facing diverse benchmarks within learning from imbalanced data streams. In this final experiment, we aim at performing an ablation study to gain deeper insights into why ROSE is such an effective classifier and which of its features help to improve the accuracy and robustness to drift and class imbalance. ROSE consists of four main features. We performed an ablation study by switching off each of these features individually and seeing how they influence the performance of our ensemble. Therefore, the static lambda version uses a fixed $\lambda = 4$, the one window version uses a single sliding window of 1000 instances regardless the number of classes, the no background ensemble skips the training of an ensemble on the background, the all features version uses all input features for learning on all base classifiers, and the none version uses none of these features. Moreover, we also compare ROSE by testing three other alternatives, replacing one classifier at a time, selecting uniform subspace distributions, and using the (Wang etal., 2015) λ rule. Tables 16 and 17 present the averaged results of ROSE without its features over the previous five experimental studies along with the three alternatives. Figure 10 shows the prequential performance of ROSE and its impaired versions over time for selected representative data stream benchmarks.

Self-adjusting λ for bagging Usage of adaptive λ in online bagging has a significant impact on ROSE performance. This is especially visible for benchmarks with drifting imbalance ratio, instance-level difficulties, and noise. The λ parameter explicitly controls the Poisson distribution for online bagging, and thus implicitly moderates the exposure of instances to base classifiers of our ensemble. By presenting more difficult instances to classifiers several times, we focus their adaptation on such challenging cases. This is crucial for better adaptation to minority classes, as borderline/rare instances should be better modeled by the classifier, while safe instances do not require such an exposure. Existing methods use a fixed value of λ, while ROSE proposes a self-adjusting modification. This is a crucial reason behind ROSE adaptation to drifting imbalance and instance-level difficulties, as the impact of most challenging instances is boosted during adaptation. At the same time, this increases the robustness of ROSE to noise, as potentially noisy instances are less exposed to ROSE and thus do not deteriorate the adaptation process.

Sliding window per class Storing individual sliding windows for each class seems to have the lowest impact on ROSE from all four features. We can see that it offers small improvements for drifting imbalance, instance-level difficulties, and noisy streams, but the

Table 13 Comparison of ranks using all algorithms (Kappa)—mean (standard deviation)

Algorithm	Static IR	Drifting IR	Instance-level	Noise	Datasets	Average rank	Meta rank
ROSE	**2.57** (3.41)	**3.77** (3.59)	**3.92** (3.00)	**2.13** (2.37)	**5.33** (3.72)	**3.55** (3.22)	**3.08** (3.19)
KUE	7.57 (4.07)	8.61 (4.16)	14.08 (7.14)	8.71 (6.28)	18.31 (9.06)	11.46 (6.14)	10.62 (7.27)
AWE	23.16 (4.46)	21.70 (5.05)	23.10 (3.97)	19.90 (5.70)	22.75 (7.51)	22.12 (5.34)	21.57 (5.61)
AUE1	20.04 (6.88)	19.02 (6.64)	16.64 (6.65)	19.48 (7.06)	22.52 (6.57)	19.54 (6.76)	19.34 (7.04)
AUE2	16.04 (7.55)	15.50 (7.33)	13.36 (7.23)	14.88 (7.80)	20.56 (7.20)	16.07 (7.42)	15.51 (7.80)
DWM	23.71 (4.55)	22.77 (6.02)	22.62 (5.40)	22.84 (5.27)	17.52 (7.03)	21.89 (5.65)	22.32 (5.78)
SAE2	16.66 (5.41)	14.14 (5.71)	14.82 (4.80)	10.38 (4.67)	19.67 (3.78)	15.13 (4.87)	13.74 (5.84)
DACC	26.60 (3.28)	26.41 (4.16)	21.69 (7.56)	24.31 (5.54)	17.94 (8.17)	23.39 (5.74)	23.69 (6.48)
ADACC	27.60 (2.98)	26.73 (3.92)	21.21 (7.32)	25.07 (5.46)	18.06 (8.19)	23.73 (5.57)	24.14 (6.52)
ARF	9.87 (6.49)	11.95 (5.78)	4.85 (2.39)	14.84 (5.93)	11.44 (7.75)	10.59 (5.67)	11.43 (6.85)
ADOB	18.81 (8.91)	22.73 (8.16)	29.21 (3.01)	25.33 (7.78)	22.50 (9.57)	23.72 (7.49)	24.36 (8.30)
BOLE	13.86 (6.61)	9.80 (6.85)	15.82 (6.63)	7.49 (4.62)	8.77 (6.56)	11.15 (6.25)	10.51 (6.78)
GOOWE	18.19 (6.08)	16.70 (8.23)	13.59 (5.91)	15.57 (8.02)	20.63 (7.36)	16.94 (7.12)	16.34 (7.60)
HEB	11.89 (5.04)	12.86 (6.08)	15.26 (5.02)	15.11 (6.16)	12.63 (7.79)	13.55 (6.02)	14.07 (6.16)
LB	5.83 (3.99)	5.68(3.52)	3.95 (3.40)	8.07 (5.77)	11.06 (6.48)	6.92 (4.63)	7.01 (5.44)
OCB	11.79 (4.01)	11.73 (4.11)	19.26 (5.53)	9.25 (4.28)	21.83 (9.71)	14.77 (5.53)	13.27 (7.18)
OBA	11.47 (5.54)	14.61 (8.63)	20.97 (6.81)	18.04 (7.02)	17.15 (4.31)	16.45 (6.46)	17.02 (7.34)
OBAD	9.90 (5.19)	10.52 (5.48)	10.38 (5.96)	13.53 (6.89)	15.65 (5.87)	12.00 (5.88)	12.26 (6.51)
OBASHT	20.09 (3.44)	21.16 (5.27)	18.33 (4.43)	23.01 (3.79)	16.81 (5.70)	19.88 (4.53)	20.73 (4.85)
OBO	8.10 (4.22)	9.32 (6.83)	11.79 (7.39)	9.97 (6.95)	13.02 (6.01)	10.44 (6.28)	10.28 (6.71)
OBOA	16.60 (7.02)	15.18 (6.10)	15.72 (6.93)	15.41 (7.01)	13.60 (6.93)	15.30 (6.80)	15.44 (6.94)
SRP	14.29 (11.3)	19.18 (9.46)	10.18 (9.40)	20.80 (9.46)	7.17 (6.61)	14.32 (9.26)	15.97 (10.8)
OOB	5.13 (4.55)	8.41 (8.85)	16.23 (8.36)	6.98 (6.67)	13.56 (5.55)	10.06 (6.80)	9.31 (7.98)
UOB	14.74 (7.49)	15.36 (7.72)	21.15 (7.56)	16.96 (8.72)	23.60 (7.02)	18.36 (7.70)	17.97 (8.53)
OSMOTE	17.63 (5.84)	15.55 (7.17)	13.23 (7.89)	18.19 (5.86)	11.77 (8.08)	15.27 (6.97)	16.15 (7.13)
OUOB	16.13 (5.81)	13.23 (5.44)	9.28 (3.83)	11.87 (7.14)	13.50 (7.64)	12.80 (5.97)	12.43 (6.66)
CSMOTE	14.09 (8.44)	14.73 (6.65)	9.92 (9.38)	14.80 (6.61)	9.98 (7.31)	12.70 (7.68)	13.20 (7.90)
OADA	27.99 (2.30)	27.98 (3.52)	23.74 (6.88)	25.84 (7.01)	17.54 (9.80)	24.62 (5.90)	25.07 (7.24)
OADAC2	25.57 (3.99)	23.66 (5.08)	23.41 (7.43)	20.75 (8.42)	23.29 (8.79)	23.34 (6.74)	22.66 (7.61)
ORUS	27.03 (3.92)	23.00 (4.50)	16.41 (10.6)	20.05 (7.78)	14.29 (11.9)	20.16 (7.76)	20.19 (9.14)
IRL	13.07 (4.70)	14.00 (7.41)	21.87 (5.66)	16.47 (6.53)	13.54 (7.67)	15.79 (6.39)	16.31 (7.03)

Bold values indicates best results

gains are overshadowed by remaining features. However, for real-world data streams we can see a much higher improvement of having a sliding window per class. This is due to the nature of these benchmarks discussed in detail in Experiment 5. Artificially generated data streams follow a probabilistic distribution and thus there always will be instances from each class present in the stream (although with varying ratios). Real-world data are not bounded by such mechanisms and thus some classes may periodically disappear from the stream. This situation is identical to catastrophic forgetting in continual learning of deep architectures, where accommodation of new information leads to discarding of the previously seen one. Buffers per class in ROSE offer robustness to catastrophic forgetting, as even during periods of high latency our ensemble will have access to instances from all

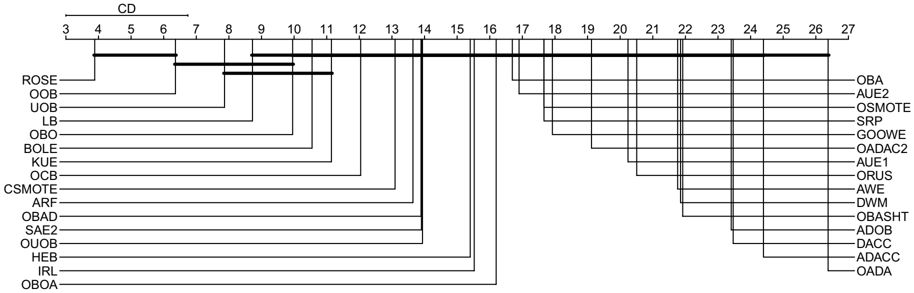
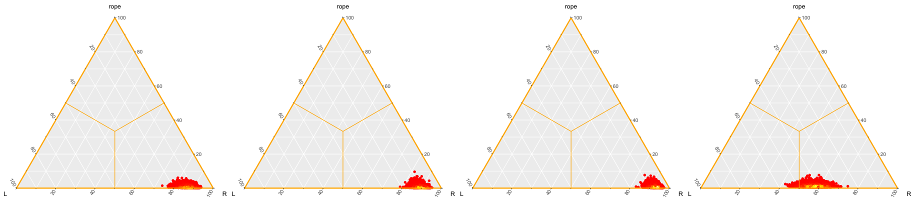


Fig. 8 Bonferroni-Dunn statistical analysis on AUC



Kappa - ROSE vs LB

Kappa - ROSE vs OOB

AUC - ROSE vs LB

AUC - ROSE vs OOB

Fig. 9 Bayesian test: ROSE versus LB and OOB on Kappa and AUC

the classes. This makes our individual sliding windows indispensable for any real-world scenario and offers a powerful backbone for adapting ROSE to class-incremental learning problems in the future.

Background ensemble Background ensemble offers a quick and safe way for ROSE to completely restart its architecture in case of a sudden changes or a strong noise presence in the stream. While ROSE adapts its base classifiers in an online manner, in some scenarios it may be more beneficial to replace most, if not all, base classifiers with new ones trained on the most recent concept (as adaptation may be too slow to properly recover from drift.) This is a significant step further from existing adaptive ensemble architectures that train only a single background classifier and use it to replace the worst performing member of the ensemble. This limits their adaptation capabilities to sudden drifts or extreme changes in imbalance ratios, as only one classifier can be replaced at a time. The background ensemble significantly improves the robustness to noise, as if the most of the base classifiers use noisy features, then one-by-one replacement will not be enough. As each member of background ensemble is trained on a new feature subset, we limit the chances of using the same noisy features in both old and new ensembles.

Random feature subspaces The combination of feature and instance subspaces support the diversity among the base classifiers in ROSE. This factor leads to one of the most significant gains in classification accuracy for ROSE, showing the importance of using diversified subspace representations for training base learners for drifting and imbalanced data streams. When subspaces are combined with self-adjusting λ, ROSE effectively gains a mechanism to control its own diversity. As it is known for data stream ensembles, high diversity is helpful when recovering from concept drift, while low / moderate diversity allows better exploitation of the stable concept. Furthermore, such

Table 14 Comparison of ranks using all algorithms (AUC)—mean (standard deviation)

Algorithm	Static IR	Drifting IR	Instance-level	Noise	Datasets	Average rank	Meta rank
ROSE	**2.49** (3.11)	**4.41** (3.83)	**4.82** (2.38)	**3.21** (2.44)	**6.40** (5.53)	**4.26** (3.46)	**3.88** (3.43)
KUE	8.59 (4.08)	10.11 (4.43)	11.23 (5.57)	10.41 (5.76)	18.35 (7.60)	11.74 (5.49)	11.14 (6.23)
AWE	23.67 (3.97)	20.73 (7.41)	25.15 (2.47)	19.38 (6.72)	23.17 (7.55)	22.42 (5.62)	21.76 (6.39)
AUE1	20.84 (6.32)	19.02 (6.43)	19.00 (6.28)	19.95 (6.93)	23.56 (5.86)	20.48 (6.36)	20.24 (6.68)
AUE2	17.09 (6.98)	16.25 (7.09)	16.33 (6.86)	15.91 (7.33)	21.75 (6.58)	17.47 (6.97)	16.90 (7.30)
DWM	23.93 (4.10)	21.14 (8.92)	25.31 (3.17)	21.52 (7.45)	15.08 (6.46)	21.39 (6.02)	21.85 (7.05)
SAE2	16.54 (5.13)	13.64 (4.76)	14.56 (4.57)	11.27 (4.55)	18.90 (5.47)	14.98 (4.90)	13.90 (5.48)
DACC	26.86 (2.77)	24.95 (6.55)	23.79 (7.00)	23.57 (5.78)	16.25 (8.44)	23.08 (6.11)	23.47 (6.75)
ADACC	27.74 (2.77)	27.02 (3.66)	23.18 (6.71)	25.25 (5.21)	15.94 (8.46)	23.83 (5.36)	24.39 (6.51)
ARF	11.11 (6.39)	13.61 (6.30)	8.23 (3.53)	16.81 (6.54)	14.56 (8.19)	12.87 (6.19)	13.64 (7.08)
ADOB	16.99 (10.6)	21.93 (8.24)	30.44 (1.88)	24.13 (8.46)	20.04 (10.3)	22.71 (7.90)	23.41 (9.35)
BOLE	12.91 (5.58)	9.66 (5.71)	16.95 (6.20)	7.20 (4.01)	9.81 (6.29)	11.31 (5.56)	10.55 (6.39)
GOOWE	18.80 (5.81)	17.75 (8.02)	17.00 (6.01)	17.19 (7.94)	20.96 (6.62)	18.34 (6.88)	17.92 (7.25)
HEB	13.30 (5.59)	13.50 (6.19)	17.23 (2.95)	16.11 (6.19)	14.63 (7.78)	14.95 (5.74)	15.40 (6.02)
LB	6.71 (3.91)	6.68(3.33)	6.41 (3.45)	9.85 (5.43)	13.17 (7.23)	8.56 (4.67)	8.72 (5.42)
OCB	10.24 (4.29)	11.00 (4.58)	16.15 (6.64)	8.50 (3.67)	21.92 (9.84)	13.56 (5.80)	12.04 (7.13)
OBA	12.61 (5.49)	15.27 (8.35)	15.21 (5.67)	19.28 (6.78)	16.85 (5.23)	15.85 (6.30)	16.69 (6.88)
OBAD	11.07 (5.16)	11.80 (5.93)	13.56 (5.16)	15.09 (6.39)	16.06 (5.52)	13.52 (5.63)	13.89 (6.08)
OBASHT	20.79 (2.97)	21.73 (5.15)	20.87 (4.17)	24.28 (3.35)	16.77 (6.18)	20.89 (4.36)	21.92 (4.75)
OBO	8.63 (4.02)	10.59 (6.76)	9.51 (5.69)	10.11 (6.03)	11.44 (5.56)	10.06 (5.61)	9.95 (5.76)
OBOA	16.69 (6.42)	15.50 (5.69)	17.69 (6.67)	16.15 (6.57)	13.85 (7.49)	15.98 (6.57)	16.20 (6.68)
SRP	15.13 (11.1)	20.59 (8.47)	12.82 (8.94)	22.41 (8.32)	9.13 (6.87)	16.02 (8.73)	17.66 (10.1)
OOB	4.37 (3.95)	7.77 (8.73)	7.62 (5.74)	4.91 (6.07)	11.23 (7.30)	7.18 (6.36)	6.36 (6.59)
UOB	5.71 (5.44)	7.23 (6.98)	9.62 (8.01)	5.18 (6.57)	18.60 (9.51)	9.27 (7.30)	7.87 (8.28)
OSMOTE	18.39 (5.90)	17.20 (7.44)	15.49 (8.25)	19.47 (5.82)	13.92 (7.26)	16.89 (6.94)	17.66 (6.99)
OUOB	16.97 (5.49)	14.36 (5.25)	12.41 (4.37)	12.82 (6.80)	15.71 (8.41)	14.45 (6.07)	13.94 (6.50)
CSMOTE	14.86 (7.94)	16.39 (6.10)	5.36 (3.64)	16.76 (5.46)	9.92 (6.70)	12.65 (5.97)	13.10 (7.82)
OADA	28.27 (2.07)	28.70 (1.89)	25.87 (6.95)	27.27 (5.24)	19.04 (9.86)	25.83 (5.20)	26.38 (6.39)
OADAC2	25.01 (3.92)	19.59 (7.78)	22.26 (10.6)	14.89 (8.93)	20.52 (10.1)	20.45 (8.28)	19.12 (9.56)
ORUS	25.94 (5.17)	23.52 (4.04)	18.51 (11.1)	20.01 (7.72)	14.90 (11.3)	20.58 (7.86)	20.51 (8.93)
IRL	13.74 (4.42)	14.34 (7.47)	15.41 (6.03)	17.11 (6.85)	13.58 (8.19)	14.84 (6.59)	15.52 (6.77)

Bold values indicates best results

subspaces limit the chances of using noisy features / instances to adapt the classifiers, leading to better robustness when learning from imbalanced data streams.

Alternative mechanisms Finally, we analyze the benefits of the ROSE features over the state-of-the-art existing mechanisms in the literature. This will allow us to prove that not only ROSE as whole offers superior performance to reference ensembles, but also that every mechanism introduced by ROSE is individually justified. Existing ensemble methods usually replace the single worst classifier (Brzeziński & Stefanowski, 2014a), while ROSE may replace several of them simultaneously. By offering the flexibility of replacing multiple classifiers at once, ROSE offers improved adaptability to changing and difficult data. In case of sudden drifts or evolving data complexity levels more than a single classifier can become outdated and simply replacing them one by one will lead to slower recovery rates after the change. This is most pronounced in the case of drifting IR and instance-level difficulties. When building feature subspaces for base classifiers reference approaches usually use a uniform probability to select the number of used features (Cano & Krawczyk, 2020), while ROSE replaces this with a normal distribution. We can see that this leads to most significant gain when handling instance-level difficulties and real-world datasets. This can be explained by the fact that better-defined subspaces lead to improved separation among instances, thus leading to reduction in classification difficulties and lower susceptibility to presence of noisy features. Finally, we have compared ROSE self-adaptive λ from Eq. 2 to the approach proposed by (Wang etal., 2015). We can see that our λ calculation method outperforms the reference one, especially when dealing with noisy and real-world datasets. This shows that ROSE λ adaptation process is less prone to temporal disturbances caused by noise, as well as can better handle diverse combinations of concept drift and imbalance ratio changes present in real-world problems.

6 Conclusions and future work

Summary In this paper, we have introduced Robust Online Self-Adjusting Ensemble (ROSE) for mining drifting and imbalanced data streams. The novelty of ROSE lies in an original ensemble architecture design with multiple features designed for a high level of interplay with each other. ROSE uses base classifiers trained on subsets of both instances and features, which allows for handling both complex and noisy data streams. ROSE offers a hybrid architecture that maintains a fixed-size pool of classifiers updated in an online manner, but is capable of automatic training of new classifiers. Drift detectors associated with each base classifier in the pool (and hence with a feature subset that it represents) control the training of a background ensemble. A Kappa-based classifier selection is used to determine if the newly trained learner should be added to the ensemble. ROSE is capable of handling both standard and imbalanced data streams without any need for switching between these modes. This is achieved by using balanced buffers per class that store instances to train new classifiers; and thanks to adaptive λ parameter that forces increased exposure of minority instances to all classifiers.

Main research findings The extensive experimental study on a total of five diverse benchmarks proved that ROSE not only is capable of significantly outperforming 30 state-of-the-art skew-insensitive and general-purpose ensembles, but additionally can handle a variety of difficult data stream mining scenarios (such as skewed classes, evolving class imbalance ratio, instance-level difficulties, noisy features, or binary and multi-class problems) without the need of end-user tuning or supervision. ROSE has a runtime and memory consumption comparable to reference methods such as Kappa Updated Ensemble, Leveraging Bag, and Adaptive Random Forest.

Lessons learned Experimental comparison: In order to gain a deeper insight into the performance of any algorithm for imbalanced data streams, it must be be evaluated using a diverse set of scenarios, including static and dynamic imbalance ratios paired with instance-level difficulties, noise, and concept drift, as well as real-world datasets to have a holistic comparison. Only such a thorough experimental study allows to formulate specific recommendations for applicability areas. We can conclude that ROSE is a well-rounded ensemble capable of displaying robustness under diverse difficulties present in imbalanced data streams. Handling skewed distributions: ROSE shows that robustness to

Table 15 Comparison of averages and ranks of the runtime (seconds per 10,000 instances) and memory consumption (RAM-hours)—mean (standard deviation)

Algorithm	Runtime—seconds	Memory—RAM-hours	Runtime—rank	Memory—rank
ROSE	7.2 (1.9)	0.042 (0.019)	20.72 (0.60)	19.54 (1.39)
KUE	5.2 (1.0)	0.018 (0.005)	18.56 (0.78)	17.69 (0.91)
AWE	1.1 (0.0)	0.001 (0.000)	4.36 (0.53)	4.36 (0.48)
AUE1	3.8 (2.0)	0.016 (0.009)	17.72 (1.68)	15.31 (2.78)
AUE2	2.0 (0.8)	0.004 (0.002)	11.26 (2.35)	10.33 (2.56)
DWM	0.6 (0.1)	0.000 (0.000)	1.67 (0.47)	1.92 (0.57)
SAE2	1.3 (0.3)	0.002 (0.001)	6.08 (2.27)	5.59 (1.66)
DACC	**0.6** (0.0)	**0.001** (0.000)	**1.33** (0.47)	**1.21** (0.40)
ADACC	0.7 (0.0)	0.001 (0.000)	3.00 (0.00)	2.87 (0.33)
ARF	12.9 (4.4)	0.125 (0.073)	22.28 (0.45)	22.03 (0.62)
ADOB	8.6 (1.0)	0.010 (0.001)	16.21 (0.46)	21.03 (1.02)
BOLE	4.2 (0.2)	0.005 (0.000)	13.54 (0.98)	16.51 (1.24)
GOOWE	2.0 (0.2)	0.005 (0.001)	13.36 (0.83)	10.72 (1.08)
HEB	2.0 (0.1)	0.004 (0.000)	10.79 (0.85)	10.69 (1.26)
LB	6.4 (1.5)	0.026 (0.010)	19.79 (0.61)	18.95 (1.15)
OCB	1.6 (0.4)	0.002 (0.001)	7.28 (1.28)	8.49 (2.10)
OBA	2.3 (0.2)	0.004 (0.000)	11.21 (0.72)	12.82 (0.67)
OBAD	1.5 (0.2)	0.002 (0.000)	7.28 (1.28)	7.28 (1.87)
OBASHT	1.4 (0.1)	0.002 (0.000)	5.85 (0.95)	5.95 (0.88)
OBO	4.3 (0.2)	0.014 (0.001)	17.97 (0.73)	16.41 (0.71)
OBOA	21.3 (12.3)	0.313 (0.297)	22.72 (0.45)	22.72 (0.90)
SRP	313.4 (106.6)	71.995 (41.225)	28.87 (0.61)	29.21 (0.69)
OOB	2.8 (0.2)	0.006 (0.001)	14.31(0.91)	14.31 (0.76)
UOB	1.8 (0.3)	0.003 (0.000)	8.72 (0.71)	9.44 (0.96)
OSMOTE	1,069.5 (636.1)	450.841 (368.807)	30.97 (0.16)	31.00 (0.00)
OUOB	344.9 (156.2)	139.497 (123.730)	29.69 (0.65)	29.46 (0.63)
CSMOTE	227.1 (67.9)	50.522 (28.527)	28.38 (0.80)	28.26 (0.74)
OADA	92.7 (30.3)	7.976 (8.760)	25.38 (0.89)	25.44 (0.87)
OADAC2	88.6 (11.7)	5.992 (2.595)	25.38 (0.54)	25.64 (0.62)
ORUS	119.2 (32.4)	13.612 (8.926)	26.90 (0.44)	26.87 (0.52)
IRL	39.6 (16.9)	2.617 (1.930)	24.41 (0.93)	23.97 (0.53)

Bold values indicates best results

class imbalance can be achieved by exploiting learning mechanisms and per-class forgetting, outperforming existing resampling approaches. Computational and memory complexity: all features of ROSE contribute to its high predictive performance (as seen during the ablation study), while being characterized by low time and memory complexities. This allows ROSE to display resource consumption on par with algorithm like Leveraging Bagging and Adaptive Random Forest. Metrics: ROSE evaluation have shown that that Kappa and AUC metrics provide complementary information about the performance of the classifiers. While Kappa strengthens the significance of the minority class under highly imbalanced datasets, AUC offers a balanced trade-off between the majority and minority

Table 16 Contribution of each of the ROSE features in improving Kappa + alternatives

Algorithm	ROSE	Static λ	One window	No background	All features	None	Replace One	Uniform subspace	Wang λ
Static IR	**63.65**	62.10	63.27	63.37	59.07	55.31	63.51	63.63	62.72
Drifting IR	**60.36**	57.80	60.09	58.78	59.39	56.73	59.34	60.23	60.07
Instance diff	**71.47**	68.48	71.16	67.03	71.23	70.40	69.56	69.64	70.78
Noise	**48.54**	43.93	48.18	47.52	47.12	42.88	48.12	48.01	45.34
Datasets	**62.48**	59.87	61.49	60.48	57.96	55.45	61.29	61.14	60.24
Average	**61.30**	58.44	60.84	59.44	58.95	56.15	60.36	60.53	59.83
Rank	**2.99**	6.36	4.19	5.80	4.48	5.57	4.48	3.95	5.18

Bold values indicates best results

Table 17 Contribution of each of the ROSE features in improving AUC + alternatives

Algorithm	ROSE	Static λ	One window	No background	All features	None	Replace One	Uniform subspace	Wang λ
Static IR	**79.07**	78.15	78.08	79.02	76.34	74.66	78.52	78.55	77.43
Drifting IR	**77.72**	75.95	76.80	77.13	76.82	75.16	77.26	77.53	77.51
Instance diff	**86.24**	83.72	86.09	84.40	86.14	84.73	84.37	85.17	85.61
Noise	**72.56**	69.21	71.63	71.14	71.73	68.67	71.69	71.71	69.31
Datasets	**88.96**	86.25	87.31	84.02	81.51	81.10	86.94	85.41	84.05
Average	**80.91**	78.66	79.98	79.14	78.51	76.86	79.76	79.67	78.78
Rank	**3.38**	7.57	4.35	6.08	4.48	6.58	4.78	4.23	4.54

Bold values indicates best results

classes. Therefore, a complete comparison of classifiers should include both complementary metrics.

Future works Our future works will concentrate on extending the classifier generation and selection procedure in such a way that will increase the probability of features that were marked as drifting ones to be included in the newly created feature subspaces. We will further investigate connections between data stream mining and continual learning to adapt ROSE mechanisms to create robust deep learning architectures. We will study the application to multi-label classification where labels are often highly imbalanced.

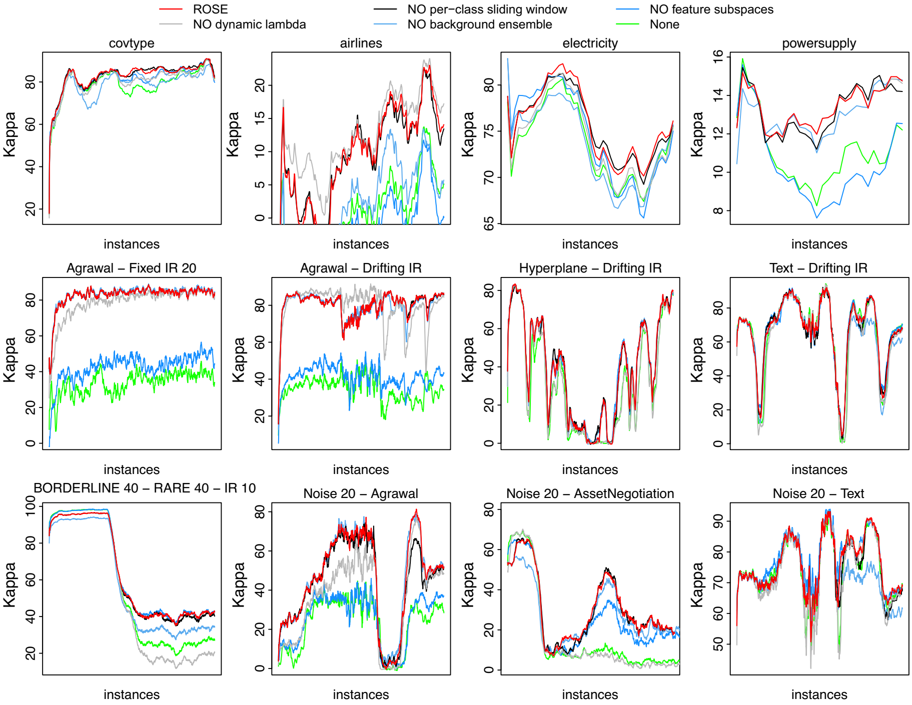


Fig. 10 Contribution of each of the ROSE features in different experiments

Acknowledgements This research was partially supported by the 2018 VCU Presidential Research Quest Fund and an Amazon AWS Machine Learning Research award. High Performance Computing resources provided by the High Performance Research Computing (HPRC) Core Facility at Virginia Commonwealth University (https://chipc.vcu.edu) were used for conducting the research reported in this work.

Author contributions AC contributed to the design of the algorithm, implementation, experimental evaluation, and manuscript preparation. BK contributed to the manuscript preparation. All authors read and approved the final manuscript.

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

Availability of data and material Data & materials available at https://github.com/canoalberto/ROSE.

Code availability Source code is available at https://github.com/canoalberto/ROSE.

Declarations

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

References

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

Al-Shammari, A., Zhou, R., Naseriparsa, 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.

Aljundi, R., Kelchtermans, K., & Tuytelaars, T. (2019). Task-free continual learning. In *IEEE conference on computer vision and pattern recognition* (pp. 11254–11263).

Aminian, E., Ribeiro, R. P., & Gama, J. (2020). A study on imbalanced data streams. In *Machine learning and knowledge discovery in databases* (pp. 380–389).

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

Bahri, M., Bifet, A., Gama, J., Gomes, H. M., & Maniu, S. (2021). Data stream analysis: Foundations, major tasks and tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11*(3), e1405.

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

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

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, 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., Kirkby, R., & Pfahringer, B. (2010). MOA: Massive online analysis. *Journal of Machine Learning Research, 11,* 1601–1604.

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

Bifet, A., Hammer, B., & Schleif, F. (2019). Recent trends in streaming data analysis, concept drift and analysis of dynamic data sets. In *European symposium on artificial neural networks*.

Bobowska, B., Klikowski, J., & Wozniak, M. (2019). Imbalanced data stream classification using hybrid data preprocessing. *Machine Learning and Knowledge Discovery in Databases, 1168,* 402–413.

Bonab, H. R., & Can, F. (2018). GOOWE: Geometrically optimum and online-weighted ensemble classifier for evolving data streams. *ACM Transactions on Knowledge Discovery from Data, 12*(2), 25.

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

Brzeziński, D., & Stefanowski, J. (2011). Accuracy updated ensemble for data streams with concept drift. In *International conference on hybrid artificial intelligence systems* (pp. 155–163).

Brzeziński, D., & Stefanowski, J. (2014). Combining block-based and online methods in learning ensembles from concept drifting data streams. *Information Sciences, 265,* 50–67.

Brzeziński, D., & Stefanowski, J. (2014). Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems, 25*(1), 81–94.

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. *Data Mining in Time Series and Streaming Databases, Machine Perception and Artificial Intelligence, 83,* 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, L. L., Pewinski, T., Stefanowski, J., & Szumaczuk, A. (2021). The impact of data difficulty factors on classification of imbalanced and concept drifting data streams. *Knowledge and Information Systems, 63*(6), 1429–1469.

Buzzega, P., Boschini, M., Porrello, A., & Calderara, S. (2020). Rethinking experience replay: A bag of tricks for continual learning. In *25th international conference on pattern recognition* (pp. 2180–2187).

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*(1), 175–218.

de Carvalho Santos, S. G. T., Júnior, P. M. G., dos Santos Silva, G. D., & de Barros, R. S. M. (2014). Speeding up recovery from concept drifts. In *European conference on machine learning and knowledge discovery in databases* (pp. 179–194).

de Barros, R. S. M., & de Carvalho Santos, S. G. T. (2018). A large-scale comparison of concept drift detectors. *Information Sciences, 451–452,* 348–370.

de Barros, R. S. M., de Carvalho Santos, S. G. T., & Júnior, P. M. G. (2016). A boosting-like online learning ensemble. In *International joint conference on neural networks* (pp. 1871–1878).

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research, 7,* 1–30.

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

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from Imbalanced Data Sets*. Springer.

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

Gama, J., $\breve{Z}$liobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys, 46*(4):44:1–44:37.

Gao, J., Ding, B., Fan, W., Han, J., & Yu, P. S. (2008). Classifying data streams with skewed class distributions and concept drifts. *IEEE Internet Computing, 12*(6), 37–49.

Ghomeshi, H., Gaber, M. M., & Kovalchuk, Y. (2019). Ensemble dynamics in non-stationary data stream classification. In *Learning from data streams in evolving environments* (pp. 123–153). Springer.

Gomes, H. M., & Enembreck, F. (2014). SAE2: Advances on the social adaptive ensemble classifier for data streams. In *ACM symposium on applied computing* (pp. 798–804).

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., et al. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning, 106*(9–10), 1469–1495.

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

Gomes, H. M., Read, J., Bifet, A., Barddal, J. P., & Gama, J. (2019). Machine learning for streaming data: State of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter, 21*(2), 6–22.

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

He, X., Sygnowski, J., Galashov, A., Rusu, A. A., Teh, Y. W., & Pascanu, R. (2019). Task agnostic continual learning via meta learning. CoRR arXiv:abs/1906.05201

Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 97–106).

Jaber, G., Cornuéjols, A., & Tarroux, P. (2013). A new on-line learning method for coping with recurring concepts: The ADACC system. In *International conference on neural information processing* (pp. 595–604).

Klikowski, J., & Wozniak, M. (2019). Multi sampling random subspace ensemble for imbalanced data stream classification. In R. Burduk, M. Kurzynski, & M. Wozniak (Eds.), *International conference on computer recognition systems* (Vol. 977, pp. 360–369).

Klikowski, J., & Wozniak, M. (2020). Employing one-class SVM classifier ensemble for imbalanced data stream classification. *International Conference on Computational Science, 12140,* 117–127.

Kolter, J. Z., & Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research, 8,* 2755–2790.

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). Concept drift detection from multi-class imbalanced data streams. In *IEEE international conference on data engineering* (pp. 1068–1079).

Korycki, L., & Krawczyk, B. (2021c). Low-dimensional representation learning from imbalanced data streams. In *Pacific-Asia conference on advances in knowledge discovery and data mining* (Vol. 12712 LNCS, 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 (big data)* (pp. 2334–2343).

Kozal, J., Guzy, F., & Wozniak, M. (2021). Employing chunk size adaptation to overcome concept drift. CoRR arXiv:abs/2110.12881

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., & Cano, A. (2018). Online ensemble learning with abstaining classifiers for drifting and noisy data streams. *Applied Soft Computing, 68,* 677–692.

Krawczyk, B., & Skryjomski, P. (2017). Cost-sensitive perceptron decision trees for imbalanced drifting data streams. *Machine Learning and Knowledge Discovery in Databases, 10535,* 512–527.

Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Wozniak, M. (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion, 37,* 132–156.

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.

Liu, C., Feng, L., & Fujimaki, R. (2016). Streaming model selection via online factorized asymptotic bayesian inference. In *IEEE international conference on data mining* (pp. 271–280).

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

Loezer, L., Enembreck, F., Barddal, J. P., & de Souza Britto Jr, A. (2020). Cost-sensitive learning for imbalanced data streams. In *Proceedings of the 35th annual ACM symposium on applied computing* (pp. 498–504).

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering, 31*(12), 2346–2363.

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

Lu, Y., Cheung, Y. M., & Tang, Y. Y. (2019). Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift. *IEEE Transactions on Neural Networks and Learning Systems, 31*(8), 2764–2778.

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

Minku, L. L., & Yao, X. (2011). DDD: A new ensemble approach for dealing with concept drift. *IEEE Transactions on Knowledge and Data Engineering, 24*(4), 619–633.

Oza, N. C. (2005) Online bagging and boosting. In *IEEE international conference on systems, man and cybernetics* (pp. 2340–2345).

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks, 113,* 54–71.

Pelossof, R., Jones, M., Vovsha, I., & Rudin, C. (2009). Online coordinate boosting. In *IEEE international conference on computer vision* (pp. 1354–1361).

Ren, S., Zhu, W., Liao, B., Li, Z., Wang, P., Li, K., et al. (2019). Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning. *Knowledge-Based System, 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).

Roseberry, M., Krawczyk, B., Djenouri, Y., & Cano, A. (2021). Self-adjusting k nearest neighbors for continual learning from multi-label drifting data streams. *Neurocomputing, 442,* 10–25.

Van Rijn, J. N., Holmes, G., Pfahringer, B., & Vanschoren, J. (2015). Having a blast: Meta-learning and heterogeneous ensembles for data streams. In *IEEE international conference on data mining* (pp. 1003–1008).

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, H., Fan, W., Yu, P. S., & Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 226–235).

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

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

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

Wang, S., Minku, L. L., & Yao, X. (2018). A systematic study of online class imbalance learning with concept drift. *IEEE Transactions on Neural Networks 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).

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* (pp. 2817–2823).

Zyblewski, P., Sabourin, R., & Wozniak, M. (2021). Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams. *Information Fusion, 66,* 138–154.

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