Applied Soft Computing 68 (2018) 677–692

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Online ensemble learning with abstaining classifiers for drifting and noisy data streams

Bartosz Krawczyk*, Alberto Cano

Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA

ARTICLE INFO

Article history:
Received 28 February 2017
Received in revised form 2 December 2017
Accepted 5 December 2017
Available online 9 December 2017

Keywords:
Machine learning
Data stream mining
Concept drift
Ensemble learning
Abstaining classifier
Diversity

ABSTRACT

Mining data streams is among most vital contemporary topics in machine learning. Such scenario requires adaptive algorithms that are able to process constantly arriving instances, adapt to potential changes in data, use limited computational resources, as well as be robust to any atypical events that may appear. Ensemble learning has proven itself to be an effective solution, as combining learners leads to an improved predictive power, more flexible drift handling, as well as ease of being implemented in high-performance computing environments. In this paper, we propose an enhancement of popular online ensembles by augmenting them with abstaining option. Instead of relying on a traditional voting, classifiers are allowed to abstain from contributing to the final decision. Their confidence level is being monitored for each incoming instance and only learners that exceed certain threshold are selected. We introduce a dynamic and self-adapting threshold that is able to adapt to changes in the data stream, by monitoring outputs of the ensemble and allowing to exploit underlying diversity in order to efficiently anticipate drifts. Additionally, we show that forcing uncertain classifiers to abstain from making a prediction is especially useful for noisy data streams. Our proposal is a lightweight enhancement that can be applied to any online ensemble method, improving its robustness to drifts and noise. Thorough experimental analysis validated through statistical tests proves the usefulness of the proposed approach.

1. Introduction

In the context of the big data era, information systems produce a continuous flow of massive collections of data surpassing storage and computation capabilities of traditional knowledge extraction methods. Big data is characterized by its properties which include volume, velocity, variety, veracity, variability, visualization, and value. In recent years, researchers have mainly focused on the scalability of data mining algorithms to address the ever increasing data volume [1]. However, one cannot ignore the importance of the remaining ones, especially velocity and variability. Velocity is critical in real-time decision systems, where new instances are continuously evaluated and fast decision must be outputted under time constraints. Variability refers to the non-stationary properties of data, which may shift over time, leading to a phenomenon known as concept drift. The velocity, variability, and complexity of data through time calls for development of effective and efficient algorithms that can dynamically adapt to changes [2], and provide fast decision and model update [3,4].

Ensemble learning is a popular approach for adapting to the dynamic nature of data streams [5]. An ensemble is a set of individual classifiers whose predictions are combined, leading to better accuracy than those from the individual classifiers. Ensembles optimize the coverage by generating complementary and diverse classifiers. Classifier ensembles are attractive for data streams because they facilitate adaptation to changes in data characteristics, e.g. by adding new components trained on recent data and removing components representing outdated concepts. However, many times concept drifts are recurrent and by simply removing outdated classifiers we would be forgetting useful knowledge for future recurring drifts. On the contrary, an approach based on abstaining classifiers [6–8] would allow both to refrain predictions when confidence regarding new instances is being lost during drift, and explore the diversity of the ensemble by allowing only a selective subset of learners to partake in the decision making. Moreover, noise in data is a recurrent difficulty in data streams. Noise can be seen as temporal or spatial fluctuations that may mislead drift detectors. Noise may fool the drift detector and make it believe a concept drift occurred, leading to an update of the model based on noisy data input, thus deteriorating accuracy.

In this paper, we introduce a lightweight and flexible abstaining extension for online ensembles that allows to exclude some of the

* Corresponding author.
E-mail addresses: bkrawczyk@vcu.edu (B. Krawczyk), acano@vcu.edu (A. Cano).

https://doi.org/10.1016/j.asoc.2017.12.008
1568-4946/

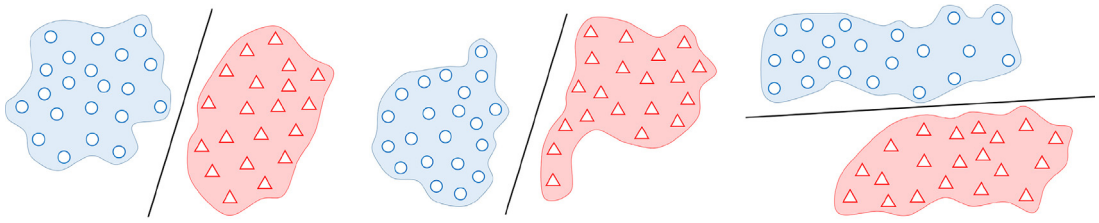


Fig. 1. Two types of concept drift according to their influence on learned classification boundaries.

classifiers from the voting process. Our method utilizes a constant monitoring of the certainty of base classifiers for each incoming instance. If a classifier displays a maximum certainty below a specified threshold, then it abstains from making a prediction (i.e., it is excluded from the voting process). We propose an adaptive scheme for threshold calculation that follows changes in the stream, thus allowing for enhancing or diminishing the role of abstaining according to the current situation. By allowing classifiers to refrain from making a decision, we allow to dynamically change the ensemble set-up (as different classifiers may partake in the classification of each instance), allowing to exploit their local competencies. Additionally, influence of classifiers that are poorly recovering from the change that took place is being diminished, thus reducing the ensemble error. It also allows to exploit the diversity in ensembles, which is a key factor in efficient learning from streams [9]. Here, we are able cover various subsets of decision space (due to diversity), but at the same time use only most accurate ensemble members (as diversity does not necessarily lead to each base learner being competent [10]).[1] Furthermore, the abstaining modification should improve the robustness of online ensembles to noise, without a need for complex and costly filters or data preprocessing solutions. Random noise is most likely to influence the certainty of the classifier regarding given instance, as it may shift the given object with the respect to learned decision boundary. Our approach is designed to select those classifiers for the voting step that are least likely to be influenced by the noise distribution.

The main contributions of this paper are as follow:

- A novel dynamic lightweight methodology for online ensembles by extending them with abstaining classifiers, that can be applied to any online ensemble model with minimal restrictions on the type of base classifier used.
- Efficient selection of most competent classifiers for each instance that allows to exploit the underlying diversity of base learners and reduce the error during drift recovery.
- Abstaining leading to increased robustness to noisy data streams without a requirement for costly preprocessing.
- An extensive experimental analysis on a large number of data stream benchmarks comparing the performance of popular online ensembles with Adaptive Hoeffding Tree, Naïve Bayes, and Multi-layer Perceptron as base classifiers, considering the canonical, static and dynamic abstaining versions.
- A study on the relationships between ensemble size vs. accuracy, and noise vs. accuracy.

The manuscript is organized as follows. Section 2 presents the background in data stream mining. Section 3 describes the proposed online abstaining classifiers methodology. Section 4 presents the experimental study. Finally, Section 5 summarizes the concluding remarks.

2. Data stream mining

A popular view on data stream is to consider it as a ordered sequence of instances that arrive over time and may be potentially of unbounded size [11]. Such settings differ from a canonical static scenario and thus are connected with specific constraints to be put onto learning algorithm designed for such environments. We assume that instances arrive one by one (online processing) or in form of data blocks (chunk processing). They arrive rapidly within given time intervals – most works assume they are identical, yet in many real-life scenarios these intervals may vary. Due to the potentially infinite size of the stream and characteristics of contemporary computing systems one cannot fit the entire stream in the memory and each instance should be processed once and then discarded. Additionally, as instances arrive continuously, their processing time must be as small as possible, in order to provide real-time responsiveness and avoid data queues. Finally, characteristic of streams may change over time due to various conditions, which is known as concept drift [12].

Data stream is the sequence of states $\mathbf{S} = \{S_1, S_2, \ldots, S_n\}$. As a state we define the subsequence generated according to a given distribution, where state S_i is generated by a distribution D_i. By a stationary data stream we will consider a sequence of instances characterized by a transition $S_j \rightarrow S_{j+1}$, where $D_j = D_{j+1}$. However, concept drift presence will lead to changes in distributions and definitions of learned concepts over time. Presence of drift can affect the underlying properties of classes that were used to train the current classifier, thus reducing its relevance with the progress of changes. At some point the drop in accuracy may be so significant that one cannot anymore consider classifier as a competent one. Thus, developing methods to tackle concept drift presence is of vital importance for data stream mining [2,13].

Concept drift can be considered with regard to its influence on learned classification boundaries. We distinguish here two types of drift – virtual and real. Former type of concept drift does not impact the decision boundaries (posterior probabilities), but affects only the conditional probability density functions. Therefore, it should not directly influence the classifier being used, still should be detected. Latter type of concept drift has effect on decision boundaries (or posterior probabilities) and potentially may have impact on unconditional probability density functions. This type of changes may significantly influence performance of a classifier. Fig. 1 depicts both of those types of drift.

Another view on types of concept drifts is based on the severity and speed of changes. Sudden concept drift is characterized by S_j being rapidly replaced by S_{j+1}, where $D_j \neq D_{j+1}$. Gradual concept drift can be considered as a transition phase where examples in S_{j+1} are generated by a mixture of D_j and D_{j+1} with their varying proportions. Incremental concept drift has a much slower ratio of changes, where the difference between D_j and D_{j+1} is not so significant, usu-

[1] As competent we mean a classifier that is updated with the current state of the stream, being able to accurately recognize new instances. Competence deteriorates when stream evolves and classifier is not able to adapt properly, losing generalization capabilities.

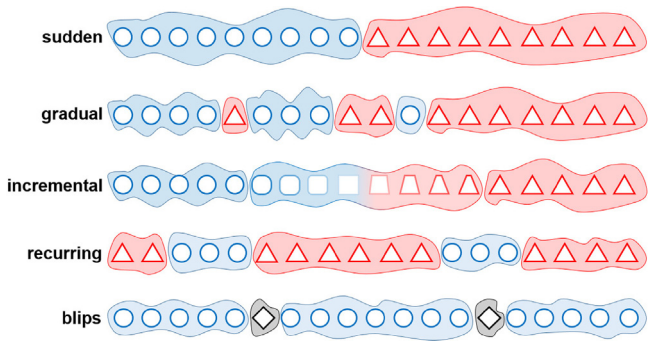


Fig. 2. Four types of concept drift according to severity and speed of changes, and noisy blips.

ally not statistically significant. In recurring concept drift a state from kth previous iteration may reappear $D_{j+1} = D_{j-k}$. It may happen once or periodically. Blips, or outliers, should be ignored due to their random nature and lack of any meaningful information being carried [14]. Fig. 2 depicts those five types of drift. Please note that in most real-life scenarios we do not have a clearly defined type of change, thus leading to so-called mixed concept drift, which may exhibit hybrid characteristics of previous types.

Noise is another difficulty embedded in the nature of data streams. It can be seen as temporal fluctuations in the incoming concept that do not translate into actual changes, are harmful to the classification system and thus must be filtered out. There exist a number of solutions for handling noisy data streams proposed in the literature. However, their application is limited, as one must expect for the noise to appear. However, in real-life scenarios noise may appear unexpectedly or periodically. Therefore, improving general robustness to noise is important for stream classifiers.

In order to tackle the presence of concept drift one may use three general solutions:

- Rebuild the classifier from scratch every time new instances arrive with the stream;
- Monitor the progress of changes in stream characteristics and update the model only when the severity of drift reaches a certain level;
- Use adaptive learning algorithm that can adapt to new instances and forget old ones in order to naturally follow drifts in the stream.

Obviously, the first solution is connected to a prohibitive computational cost and thus only two remaining ones are used in data stream mining field. Most important approaches for adapting learning systems to concept drift include:

- **Concept drift detectors**: they can be seen as external algorithms that are combined with given classifier. Their purpose is to monitor specific properties of data stream, such as standard deviation [15], predictive error [16], or instance distribution [17]. Any change to these characteristics is assumed to be caused by the drift presence. Thus, by measuring the level of changes, detectors are able to report the incoming shift.
- **Sliding windows**: these techniques keep a buffer with most recent instances that are being considered as representative for the current state of the stream [18]. They are used for the training / updating purposes and are discarded once newer instances arrive with the stream. It allows to track data stream by storing its most recent state [19].
- **Online learners**: update the model instance by instance, thus accommodating changes in stream as soon as they occur. Such learners must fulfill a set of requirements [20]: each object must be processed only once in the course of training, computational complexity of handling each instance must be as small as possible, and its accuracy should not be lower than that of a classifier trained on batch data.
- **Ensemble learners**: using a combination of several classifiers is a very popular approach for data stream mining (see a thorough survey by Krawczyk et al. [5]). Due to their compound structure [21] they can easily accommodate changes in the stream, offering gains in both flexibility and predictive power [22]. Two main approaches here assume a changing line-up of the ensemble [23] or updating base classifiers [24]. In the former solution a new classifier is being trained on recently arrived data (usually collected in a form of chunk) and added to the ensemble. Pruning is used to control the number of base classifiers and remove irrelevant or oldest models. A weighting scheme allows to assign highest importance to newest ensemble components, although more sophisticated solutions allow to increase weights of classifiers that are recently best-performing. Here one can use static classifier, as the dynamic line-up keeps a track of stream progress. Latter solutions assume that a fixed-size ensemble is kept, but update each component when new data become available.

Proper evaluation of classifiers for data stream mining is much more complex than in static scenarios, as one must take into account various characteristics. It is worth to notice that a good algorithm cannot excel at one of them, while under-performing on others. Instead, it should aim to strike a balance among all of these criteria:

- **Predictive power**: a popular criterion measured in all learning systems. However, due to the dynamic nature of streams the relevance of error fades with time, making the usage of prequential metrics necessary [25].
- **Memory consumption**: necessary to evaluate due to imposed hardware limitations for processing potentially infinite streams.
- **Update time**: reports how much time is required by a classifier to accommodate for new instances and model or decision boundaries.
- **Decision time**: another time-related measure used. It informs us how much time classifier requires to make a prediction for each new instance (or their batch).

3. Online ensembles of abstaining classifiers

In this section we will describe in detail the proposed dynamic abstaining modification for online ensemble learning from drifting data streams.

3.1. Dynamic abstaining mechanism for exploring diversity

Diversity is a key factor influencing the performance of ensemble learning methods for mining non-stationary data streams [5,9]. In stationary scenarios diversity allows ensembles to combine mutually complementary classifiers to cover the decision space more extensively. In non-stationary cases diversity may be translated to capability of handling concept drifts. By having a pool of varying base learners, one may be able to anticipate the incoming drift, as at least one of the classifiers will have a decision boundary that can be quickly adapted to the new concept. However, diversity does not translate directly into accuracy. While having a pool of diverse learners, they do not necessarily have to make an accurate combined model. Furthermore, even if one of the classifiers is better suited for managing the new state of the stream, the aggregated decision making may diminish its influence on the final class being predicted. Therefore, while having diversity is beneficial to

the ensemble, one needs a tool to smartly manage it in order to efficiently exploit the capabilities it offers.

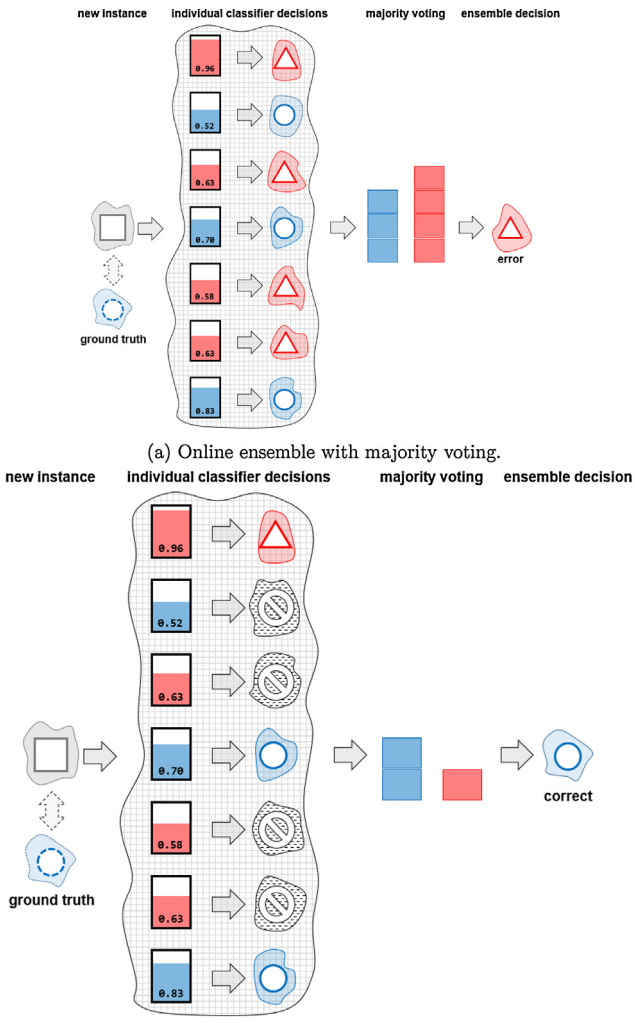
We propose to use abstaining for excluding uncertain classifiers from the class label prediction. It means that each base classifier in the ensemble may choose either to output a label prediction or abstain from making a decision. In a discussed online setting this would translate to selecting an ever-changing subset of classifiers for each incoming instance, based on their predispositions for a certain object. We may relate such an approach to works on dynamic ensemble selection in static environments [26]. Such methods require a separate validation set to measure competencies of base learners and are computationally costly [27], which is prohibitive for their usage in mining high-speed data streams in online mode. Therefore, we need to investigate different directions.

Abstaining classifiers have been mainly investigated in the context of rule-based classifiers [8]. Here, a rule that did not cover the classified instance was abstaining instead of using any generalization approach. However, we want our approach to be more flexible and work with a wider range of base classifiers. We propose to base abstaining on the certainty displayed by each base classifier regarding the new instance to be classified.

Let the data stream $\mathbf{DS} = \{(x_1, j_1), (x_2, j_2), \ldots, (x_k, j_k), \ldots\}$ be a potentially infinite set of instances, where x_k stands for feature vector ($x_k \in \mathcal{X}$) describing the kth object and j_k its label $j_k \in \mathcal{M}$. We assume that we have at our disposal a pool of L individual online classifiers that form an ensemble $\hat{\Psi} = \{\Psi_1, \Psi_2, \ldots, \Psi_L\}$, where each base model is able to give continuous output in a form of support functions $F_l(x, j) \in [0;1]$ for object x belonging to jth class. Such support function may be used to measure the certainty of each base learner regarding its label prediction. While most of online ensembles use voting with discrete decisions, many online classifiers are able to additionally return their support functions. We propose to utilize a hybrid architecture. The final decision regarding predicted label is made using majority voting, but abstaining is based on comparing the certainty of each classifier from the pool with a given threshold. If classifier fails to satisfy the threshold, then it abstains from making a decision. Thus for each instance, we will select only those classifiers for voting that satisfy the current threshold restrictions.

Using static threshold value will lead to poor performance on drifting data streams. Therefore, we propose to use a dynamic abstaining threshold. We modify its value based on the correctness of ensemble decision. If the ensemble of selected classifiers was able to predict correctly the label, then it means that we have selected competent classifiers and we may lower the threshold in order to probe for additional similarly competent learners. On the other hand, an incorrect decision may indicate a drift occurrence, as majority of classifiers were not able to properly classify the instance. In such a case the threshold needs to be increased in hope that less competent classifiers will be excluded and we will use the ones most suitable for the current state of the stream.

Such an approach is able to exploit the diversity in the ensemble. When drift occurs, one or few classifiers may adapt much faster to the change than remaining ones. Majority voting used in online ensembles will not allow them to be properly utilized and the overall recovery rate of the entire ensemble will be much slower than its most competent components. By employing adaptive abstaining, we will be able to consider those few classifiers by using only them for predicting new labels until remaining classifiers will update themselves sufficiently to properly classify new concepts. The proposed abstaining will lead to selective control over diversity, allowing to either use larger number of base models when they all can contribute useful information, or reverting to a smaller subset of classifiers that can better anticipate the direction of the drift.



(a) Online ensemble with majority voting.

(b) Online ensemble with majority voting and abstaining threshold currently set to 0.65.

Fig. 3. Toy example of the differences between standard online ensemble and proposed abstaining online ensemble. The true class label of the incoming instance is blue one.

A possible drawback of the proposed method lies in the fact that high support may not directly translate to a competent classifier – in fact it may also point to classifiers that are confident of a wrong decision. However, by using an ensemble that maintains its diversity during the stream processing we aim at reducing cases where majority of highly confident classifiers in the pool would be actually incompetent. Additionally, our algorithm in its current state assumes the continuous access to true class labels. It can be easily extended with active learning paradigm to work with partially labeled data streams [28].

A illustrative toy example is depicted in Fig. 3. It presents two ensembles – canonical one and abstaining. Both use majority voting to make a decision regarding new instance. Numbers inside classifiers show their support functions for selected class, while voting uses only discrete outputted information. In the first case, wrong class is selected due to classifiers with low certainty offering a majority of votes. In the second case, an abstaining threshold equal to 0.65 is introduced, excluding four most uncertain classifiers. One may note that even if instead of voting one would use classifier combination operators on support functions (maximum or average), the scenario would be exactly the same and without abstaining a wrong class would be predicted. Classifier combiners based only on support functions may be too restrictive (such as

max operator) or too sensitive to small changes in support functions (such as *avg* operator). One may also use support functions to weight votes, but many works point to the fact that such a weighted voting does not improve the results in a statistically significant manner. The proposed hybrid approach allows to select classifiers for voting based on their support functions, combining advantages of both solutions.

The discussed situation may easily be translated into a drifting scenario, where classifiers are bound to lose certainty and competence when drifts occur. By using an adaptive abstaining threshold, we will allow to modify the outcomes of voting to promote diverse and accurate base learners. Such a dynamic abstaining will select classifiers that display lowest loss of certainty during the drift occurrence, promoting learners that can quickly recover after a drift or that were anticipating change presence most accurately. A general framework for the proposed dynamic abstaining approach is presented in Algorithm 1.

Algorithm 1. Proposed general framework for dynamic abstaining online ensembles.

input: ensemble $\widehat{\Psi}$, abstaining threshold $\theta \in [0,1]$, adjustment factor $s \in [0,1]$
$\theta \leftarrow$ initialize threshold
L $\leftarrow$ size of the ensemble
while *end of stream = FALSE* **do**
  obtain new instance x from the stream
  for $l \leftarrow 1$; $l \leq L$; $l++$ **do**
    obtain classifier support $F_{\Psi_l}(x)$ for each class
    if $\max_{j \in \mathcal{M}} F_{\Psi_l}(x,j) < \theta$ **then**
      Ψ_l abstains from the decision
    else
      Ψ_l participates in voting
  $z \leftarrow$ result of non-abstaining classifiers voting
  obtain label y of object x
  if $z == y$ **then**
    $\theta \leftarrow \theta - s \quad (\text{if} \quad \theta > 0)$
  else
    $\theta \leftarrow \theta + s \quad (\text{if} \quad \theta < 1)$

The proposed modification is lightweight, as it only requires to keep and update a single additional variable and conduct L simple comparisons during testing phase for each instance. Training phase with new instance is not affected. The abstaining modification should not impose any significant computational costs on the learning procedure.

3.2. Flexible areas of applicability

The proposed abstaining approach can be applied to almost any existing online ensemble learning algorithm. Therefore, it should be seen as a lightweight enhancement that will allow to improve the performance of underlying ensemble classifier, rather than classifier itself. We give the following suggestions regarding the base method to which our augmentation can be applied:

- It should be an online ensemble that ensures diversity among their members and allows to monitor the performance of its base classifiers. Most streaming algorithms based on Bagging, Boosting or Random Subspaces are suitable.
- Current version of the abstaining assumes equal importance of all base classifiers and usage of majority voting. However, it can be relatively simply applied to methods using weighted classifier combination, with a requirement for a proper weight normalization once some of base classifiers have been excluded from the label prediction for a given instance.
- Ensemble must use base classifiers that are able to work in an online mode and can return their confidence levels for each new instance (e.g., as support functions).
- We strongly suggest to use online ensembles with drift detector to update the pool of base models with new, competent ones after the change has been identified and remove the outdated ones. Pruning offers two advantages. Firstly, it will allow to discard incompetent classifiers that would require too much time to adapt to new state of the stream. They could display high confidence, yet low competence and thus negatively impact the abstaining module. Secondly, adding classifiers trained only on recent instances will positively impact the diversity of the ensemble that our abstaining module aims to exploit.

3.3. Tackling noisy data streams

In previous subsections we have discussed drifting data streams and how abstaining may improve ensemble adaptation to non-stationary conditions. However, it is interesting to analyze the potential of abstaining for alleviating the influence of noise on the accuracy of online learning methods.

In many real-life problems noise is bound to appear. It may happen due to corrupted data source, transmission errors, or malicious activities influencing received data. It has been widely discussed in stationary data mining, as noisy training instances have huge impact on the generalization capabilities of the learned model [29]. This problem becomes even more challenging in learning from non-stationary data, as not only incoming instances may be corrupted with varying and evolving level of noise, but additionally one must be able to distinguish between noise and actual concept drift. It is very likely that isolated noisy samples may stimulate drift detector, thus leading to premature rebuilding of the classification model. There is a number of solutions proposed for mining noisy data streams, usually relying on filtering of the incoming data

[30], training dedicated classifiers [31], or ensembles [32,33]. These methods usually impose additional computational cost that may be often prohibitive when mining high-speed data streams. Additionally, in real-life scenarios we often cannot predict the appearance of noise. Additionally, noise is not always prevalent in the stream, but it is more likely to appear periodically. Paying the cost of using a dedicated method when it is not always necessary may be not well motivated. On the other hand, it is difficult to decide when to switch from a standard classifier to one devoted specifically to noisy streams. It seems worthwhile to investigate solutions that do not influence the standard stream mining process, while offering increased robustness to noise with limited or no additional cost.

The proposed dynamic abstaining ensemble modification will enhance the robustness of underlying learning method to noise in the stream. If an instance is influenced by noise, it will shift its position with the respect to a decision boundary. Learning and classification difficulties can originate from such information corruption. The closer the noisy instance gets to the decision boundary, the lower certainty of given classifier. An ensemble solution may benefit from its diversity, as due to using varying decision boundaries, decision of base classifiers may be differently influenced by the noisy sample. Some of them will lose confidence (especially if noisy instance will shift to the opposite side of the decision boundary), whereas others may still properly recognize it due having different, yet complementary class separation boundaries. Therefore, by abstaining most uncertain classifiers, we will enhance the role of those few least likely to be influenced by noise. Nevertheless, we acknowledge the possibility of noise so strongly shifting the instance that a classifier would display high certainty, despite classifying it to a wrong class. In such a case learner will still be allowed to participate in voting, while not being competent. Yet, it is not likely that all of the confident classifiers will be at the same time similarly incompetent (once again due to the underlying diversity). Abstaining shows some similarities to a work by Zhu et al. [34], where they proposed a dynamic classifier selection for noisy data streams. Their approach required significant additional computation and access to validation set that may be impossible to access in dynamic data streams. Our method is lightweight and does not require access to additional instances.

We acknowledge that while our proposal may improve robustness of the classification phase to noisy instances, it does not influence the robustness of the training phase. Noisy instances may still impact the classifier update step. However, our aim was to not explicitly address the noise, but to show that the abstaining solution may offer improved noise robustness at no cost. We plan to investigate excluding noisy samples from the training phase using active learning in our future works.

4. Experimental study

In this section we present the experimental study in order to evaluate the effectiveness of online ensemble learning with abstaining classifiers. Experiments were designed to answer the following research questions:

- Does the abstaining modification of online ensemble learning leads to improvements in accuracy during mining drifting data streams?
- Does the dynamic abstaining threshold that adapts to the changes in data offers significant improvement over a static one?
- Is the efficiency of abstaining related to the choice of a base classifier?
- How does the efficiency of abstaining relate to the ensemble size?
- Is the abstaining modification truly lightweight, not imposing a serious increase in computational complexity of the ensemble methods?
- Can abstaining improve the robustness of online ensembles in noisy data streams without any additional data preprocessing?

Table 1
Data stream benchmarks characteristics.

Dataset	Instances	Features	Classes	Drift
RBF_grad	1,000,000	20	4	Gradual
RBF_blip	1,000,000	10	6	Blips
Hyp_slow	1,000,000	10	4	Incremental
Hyp_fast	1,000,000	10	4	Incremental
SEA_*s udden*	1,000,000	3	2	Sudden
LED_fast	1,000,000	7	10	Mixed
LED_nodr	1,000,000	7	10	No drift
RTree	1,000,000	10	6	Sudden recurring
Waveform	1,000,000	40	3	Mixed
CovType	581,012	54	7	Virtual
Electricity	45,312	8	2	Unknown
Poker	1,000,000	10	10	Unknown

4.1. Data stream benchmarks

We used 12 data streams to evaluate the performance of the abstaining modification of online ensembles. We have selected a diverse set of benchmarks with varying characteristics, including real datasets and stream generators with different properties concerning nature, speed and number of concept drifts, which are shown in Table 1.

For experiments with noisy data streams, we took all of 12 datasets and injected a random feature noise into them. The noise level ranged between 5% and 50%, allowing us to evaluate the robustness of examined methods varying degree of feature corruption, creating 120 new data stream benchmarks for the second experiment.

4.2. Set-up

During the experiments, we have used two online ensemble learning methods for evaluating the efficiency of the proposed abstaining extension:

- Online Bagging (OB) [35] is a modification of popular ensemble learning approach suitable to streaming scenarios. Here we assume that each incoming instance from the stream may be replicated zero, one, or many times in order to update each ensemble member. Therefore, each base classifier is given k copies of the new instance, where k varies for each of them. The value of k is selected on the basis of Poisson distribution, where $k \sim Poisson(1)$. We apply an extended version of Online Bagging that uses ADWIN drift detector [16] for replacing weakest classifier with a new one after drift happens.
- Leveraging Bagging (LB) [36] is a modification of Online Bagging that aims at increasing the role of randomization in input for base classifiers. Leveraging Bagging increases resampling from $Poisson(1)$ to $Poisson(\lambda)$ (where λ is a user-defined parameter). There is a possibility of using error-correcting output codes for classification, but for the abstaining extension we use canonical majority voting option. Leveraging Bagging also relies on ADWIN for updating ensemble set-up in case of drift.

As base learners we utilize three popular online classifiers: Adaptive Hoeffding Tree [37], Naïve Bayes and Multi-layer Perceptron. Their parameters are given in Table 2. For examined ensemble methods the main parameter is the number of base learners, which we will examine further in details.

Table 2
Used online classifiers and their parameters.

Acronym	Name	Parameters
AHT	Adaptive Hoeffding Tree	Paths: 10 splitConfidence: 0.01 Leaves: Naïve Bayes
NB	Naïve Bayes	–
MLP	Multi-layer Perceptron	Hidden nodes:10 Learning: online backpropagation Iterations: 300 Learning rate: 0.01 Momentum: 0.01
LB	Leveraging Bagging	$\lambda = 2$

We use the following experimental framework:

- Classification methods were evaluated with the use of four different metrics: prequential accuracy, memory consumption, update time and classification time.
- We have used an online learning scenario with test-then-train solution. It means that each incoming instance is first used to evaluate the performance of tested ensembles and then to update them. Each experiment was repeated 10 times and we report averaged results over these runs.
- The proposed dynamic abstaining modification used an initial threshold $\theta = 0.65$ and adjustment factor $s = 0.01$. These parameters may be easily adjusted to the specific user's needs and the nature of analyzed stream. If rapid changes are to be expected, then adjustment factor should be increased to allow for faster adaptation. If changes are expected to be of slower nature, then lower values of adjustment factor will lead to a more stable performance. We propose a single value for all types of examined streams in our experimental study. Our initial experiments on θ value initialization show that as long as the selected value is not an extreme one (close to 0 or 1) the ensemble stabilizes very quickly regardless of the chosen parameter.
- To assess the significance of the results, we conducted a rigorous statistical analysis [38]. We used Shaffer post-hoc analysis for multiple comparison over multiple datasets. For all of statistical tests the significance level was set to $\alpha = 0.05$.
- To gain a better understanding about what is the actual factor behind robustness to noise, we propose to use the Equalized Loss of Accuracy (ELA) [39]. ELA helps us to check if the performance on noisy data is related to actual robustness or just to differences in initial predictive accuracies. It is computed as $ELA_{x\%} = (100 - Acc_{x\%})/Acc_{0\%}$, where $Acc_{x\%}$ is the test accuracy with an overlapping level $x\%$ and $Acc_{0\%}$ is the test accuracy in the original dataset D. Therefore, the lower the value of ELA, the more robust is given classifier to noise. At the same time, ELA takes into an account the fact that a classifier with a low base accuracy

Table 3
Average prequential accuracies and computational complexities for canonical and abstaining online ensembles with Adaptive Hoeffding Tree as a base classifier. Best obtained results are in bold.

Dataset	Online Bagging			Leveraging Bagging		
	NoAbst	StAbst	DyAbst	NoAbst	StAbst	DyAbst
RBF_grad	91.47	87.43	**93.18**	93.68	89.85	**95.09**
RBF_blip	92.38	88.16	**93.70**	94.16	90.12	**95.03**
Hyp_slow	**89.93**	85.12	89.12	**85.48**	79.49	84.98
Hyp_fast	88.96	86.84	**91.38**	87.52	85.38	**90.07**
SEA_sudden	88.07	82.13	**91.66**	87.24	80.98	**90.32**
LED_fast	67.62	63.05	**71.16**	66.74	59.18	**70.30**
LED_nodr	51.23	**51.47**	**51.47**	50.64	**50.93**	**50.93**
RTree	**43.30**	40.35	42.00	**39.79**	36.81	38.14
Waveform	81.84	80.38	**83.97**	82.32	81.06	**84.04**
CovType	80.40	79.11	**81.39**	81.04	80.03	**81.82**
Electricity	77.31	75.18	**80.48**	77.06	74.92	**78.11**
Poker	61.18	**62.03**	**62.03**	82.62	**82.81**	**82.81**
Avg. RAM-Hours	0.003	0.003	0.004	0.02	0.02	0.03
Avg. Train time (s)	4.76	4.76	4.93	12.03	12.03	12.76
Avg. Test time (s)	0.38	0.38	0.40	0.44	0.44	0.46

Table 4
Average prequential accuracies and computational complexities for canonical and abstaining online ensembles with Naïve Bayes as a base classifier. Best obtained results are in bold.

Dataset	Online Bagging			Leveraging Bagging		
	NoAbst	StAbst	DyAbst	NoAbst	StAbst	DyAbst
RBF_grad	88.58	84.54	**90.62**	90.46	85.15	**92.20**
RBF_blip	88.74	85.19	**90.92**	90.85	84.97	**92.36**
Hyp_slow	**89.17**	84.92	88.82	**85.02**	80.00	84.03
Hyp_fast	88.39	86.22	**91.02**	86.71	84.61	**89.74**
SEA_sudden	85.82	79.12	**89.09**	84.66	78.76	**87.75**
LED_fast	67.14	62.82	**70.88**	67.48	62.97	**71.02**
LED_nodr	54.66	**55.01**	**55.01**	53.72	**53.96**	**53.96**
RTree	**39.75**	36.69	38.52	**37.22**	33.91	36.47
Waveform	80.67	79.25	**83.01**	81.17	79.89	**83.27**
CovType	**78.62**	77.41	78.25	**79.91**	77.85	79.52
Electricity	73.82	70.38	**77.00**	74.81	71.42	**77.96**
Poker	60.97	**61.13**	**61.13**	81.98	**82.06**	**82.06**
Avg. RAM-Hours	0.001	0.001	0.001	0.01	0.01	0.01
Avg. Train time (s)	2.13	2.13	2.39	7.02	7.02	7.48
Avg. Test time (s)	0.22	0.22	0.23	0.27	0.27	0.28

Table 5
Average prequential accuracies and computational complexities for canonical and abstaining online ensembles with Multi-layer Perceptron as a base classifier. Best obtained results are in bold.

Dataset	Online Bagging			Leveraging Bagging		
	NoAbst	StAbst	DyAbst	NoAbst	StAbst	DyAbst
RBF_grad	90.28	86.10	**91.97**	92.44	88.74	**94.21**
RBF_blip	90.37	85.89	**92.01**	92.63	88.82	**94.25**
Hyp_slow	**88.97**	83.06	88.26	**86.40**	81.34	85.92
Hyp_fast	88.62	86.13	**91.31**	86.99	84.50	**91.40**
SEA_sudden	87.72	81.16	**90.98**	87.31	80.48	**90.49**
LED_fast	67.93	63.11	**71.52**	68.11	63.47	**71.96**
LED_nodr	50.95	**60.57**	**60.57**	49.89	**50.16**	**50.16**
RTree	**40.03**	36.88	39.01	**38.31**	35.12	37.28
Waveform	82.03	81.36	**84.43**	82.14	81.37	**84.58**
CovType	80.54	79.42	**80.93**	81.39	80.58	**81.91**
Electricity	76.15	74.11	**79.36**	75.93	73.79	**79.17**
Poker	63.17	**63.49**	**63.49**	84.68	**85.00**	**85.00**
Avg. RAM-Hours	0.009	0.009	0.0010	0.10	0.10	0.11
Avg. Train time (s)	6.99	6.99	7.86	20.04	20.04	21.01
Avg. Test time (s)	0.43	0.43	0.062	0.59	0.59	0.62

Table 6
Shaffer's test for comparison between different ensemble approaches. Symbol '>' stands for situation in which dynamic abstaining approach is superior.

AHT		NB		MLP	
Hypothesis	p-value	Hypothesis	p-value	Hypothesis	p-value
OB-DyAbst vs. OB-NoAbst	>(0.027)	OB-DyAbst vs. OB-NoAbst	>(0.031)	OB-DyAbst vs. OB-NoAbst	>(0.025)
OB-DyAbst vs. OB-StAbst	>(0.000)	OB-DyAbst vs. OB-StAbst	>(0.000)	OB-StAbst vs. OB-StAbst	>(0.000)
LB-DyAbst vs. LB-NoAbst	>(0.022)	LB-DyAbst vs. LB-NoAbst	>(0.028)	LB-DyAbst vs. LB-NoAbst	>(0.018)
LB-DyAbst vs. LB-StAbst	>(0.000)	LB-DyAbst vs. LB-StAbst	>(0.000)	LB-StAbst vs. LB-StAbst	>(0.000)

$Acc_{0\%}$ that is not deteriorated at higher noise levels is still not a better choice than a better classifier suffering from a low loss of accuracy when the noise level is increased.

4.3. Experiment 1: Learning from drifting data streams

This experiment aims at analyzing the influence of the proposed dynamic abstaining modification on the performance of ensemble learning methods with the respect to used committee approach and base learner. Apart from the canonical and dynamic abstaining versions, we present the results for a static abstaining, where the threshold is fixed for the entire data stream. Additionally, we examine the correlation between the ensemble size and the usage of abstaining mode.

Averaged prequential accuracies, memory consumption, as well as update and test times of examined methods are given in Table 3 for ensembles of AHTs, in Table 4 for ensembles of NBs, and in Table 5 for ensembles of MLPs. Memory usage, as well as update and testing times calculated over 1000 instances processed in an online mode. We presented the best results coming from ensembles of size evaluated in separate experiments, for each data stream ranging from 5 to 50 base classifiers. The relationships between the size and accuracy are depicted in Fig. 4 for ensembles using AHTs. Dependencies for remaining base classifiers were identical. For the sake of clarity, Fig. 4 depicts only canonical and dynamically abstaining ensembles, as static ensembles are bound to underperform on drifting data streams and thus we do not need to focus on them. Shaffer post-hoc test results are given in Table 6.

Firstly, we will discuss the comparison among canonical, static abstaining and dynamic abstaining methods. One can easily see that static ensembles return inferior accuracy on all of drifting data streams. It can be explained by a lack of adaptiveness to changes in the data. A preset threshold cannot capture the dynamic nature of incoming instances and thus is not sufficient for non-stationary scenarios. In cases of severe concept drift it may be too high and force to abstain most of classifiers, despite their adaptation to the new state of the stream. This takes place frequently right after the drift, when ensemble members try to recover and some base classifiers suffer lower loss of competence than others. Despite it, a fixed threshold will not differentiate between stable and drifting stages of the stream, thus excluding classifiers in moment when loss of certainty does not necessarily mean complete loss of competence. Here the static abstaining can be seen as being too strict. On the other hand, one may imagine a situation in which we deal with moment when stream stabilizes. Only most competent classifiers should be promoted and allowed to participate in voting. On the other hand, static threshold may be too small and permit too many classifiers to partake in the final decision making. Here static abstaining can be seen as being too liberal and not flexible enough to capture dynamics of drifting streams. Nevertheless, it is interesting to notice the performance of static abstaining on datasets with no drift (LED_nodr and Poker). Here, it is able to offer a small improvement over canonical online ensemble learning and return identical performance as its dynamic counterpart. We can explain it by the fact that there is no need to adapt the threshold, as there are no drastic changes in the stream. Therefore, ensembles are able to benefit from excluding less competent classifiers, while at the same time not being affected by a drastic change of concepts. However, as existence of static data streams is very unlikely in real-world scenarios, we may conclude that, to no surprise, static abstaining approach should be avoided in online ensemble learning.

Secondly, we will compare canonical online ensembles with the dynamic abstaining modification. One can see that the proposed dynamic abstaining leads to significant improvements in obtained accuracies on most of used benchmarks, especially those affected by drift. For ensembles using AHT and MLP classifiers, the canonical approach was better only on two datasets (Hyp_slow and RTree), while for ones using NB classifier on three datasets (Hyp_slow, RTree and CovType). The proposed modification, despite its simplicity, is able to improve the performance of online ensemble learning algorithms in non-stationary scenarios, regardless of ensemble model or base classifier used. Continuous monitoring

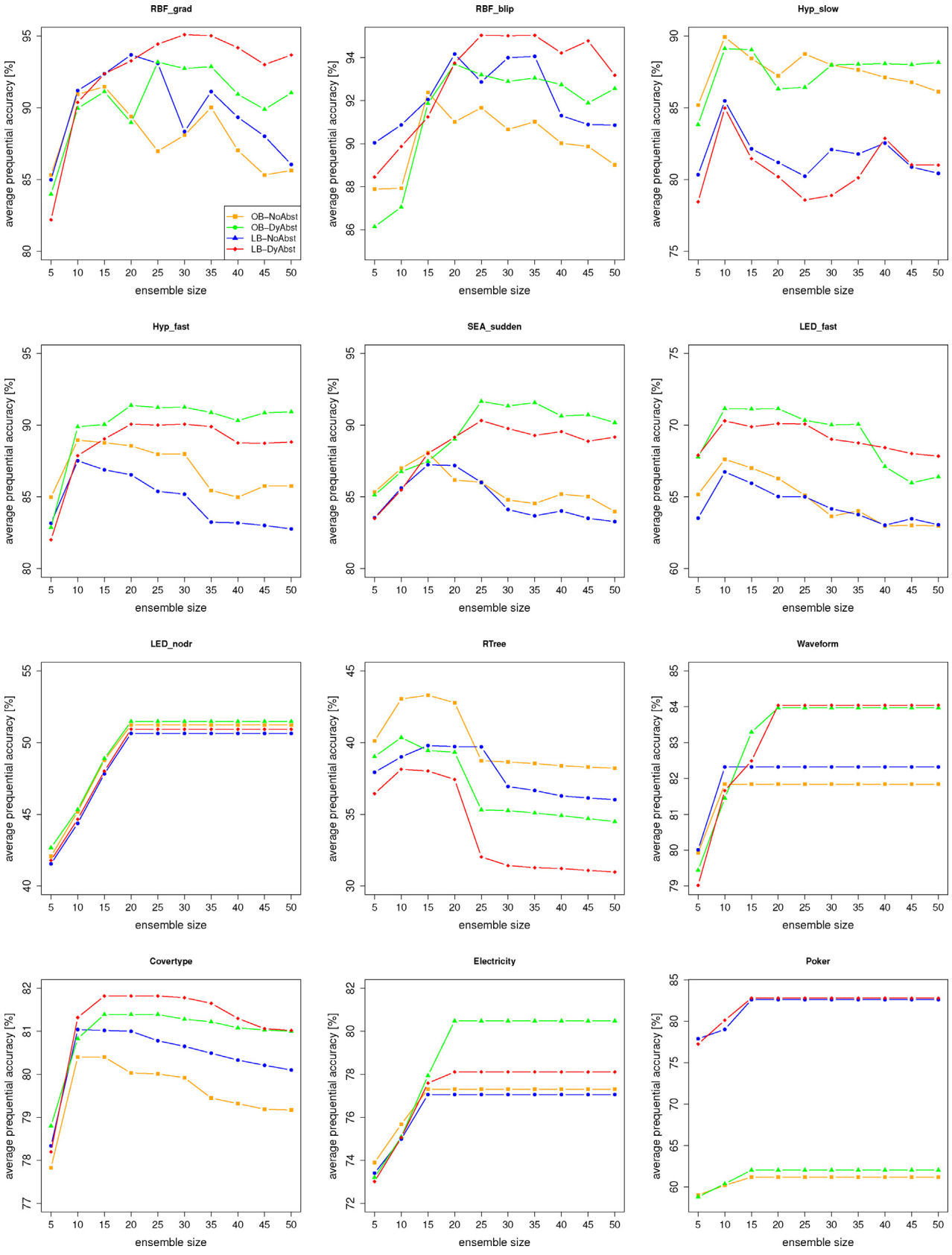


Fig. 4. Relationship between ensemble size and accuracy (Adaptive Hoeffding Tree).

of ensemble performance and adapting the abstaining threshold as the stream progresses plays a major role here, by allowing to select only most competent classifiers for the voting process, thus reducing the chance of a number of weak classifiers outvoting the few better adapted ones. Such behavior is especially useful during the presence of drift, when we are able to take advantage of

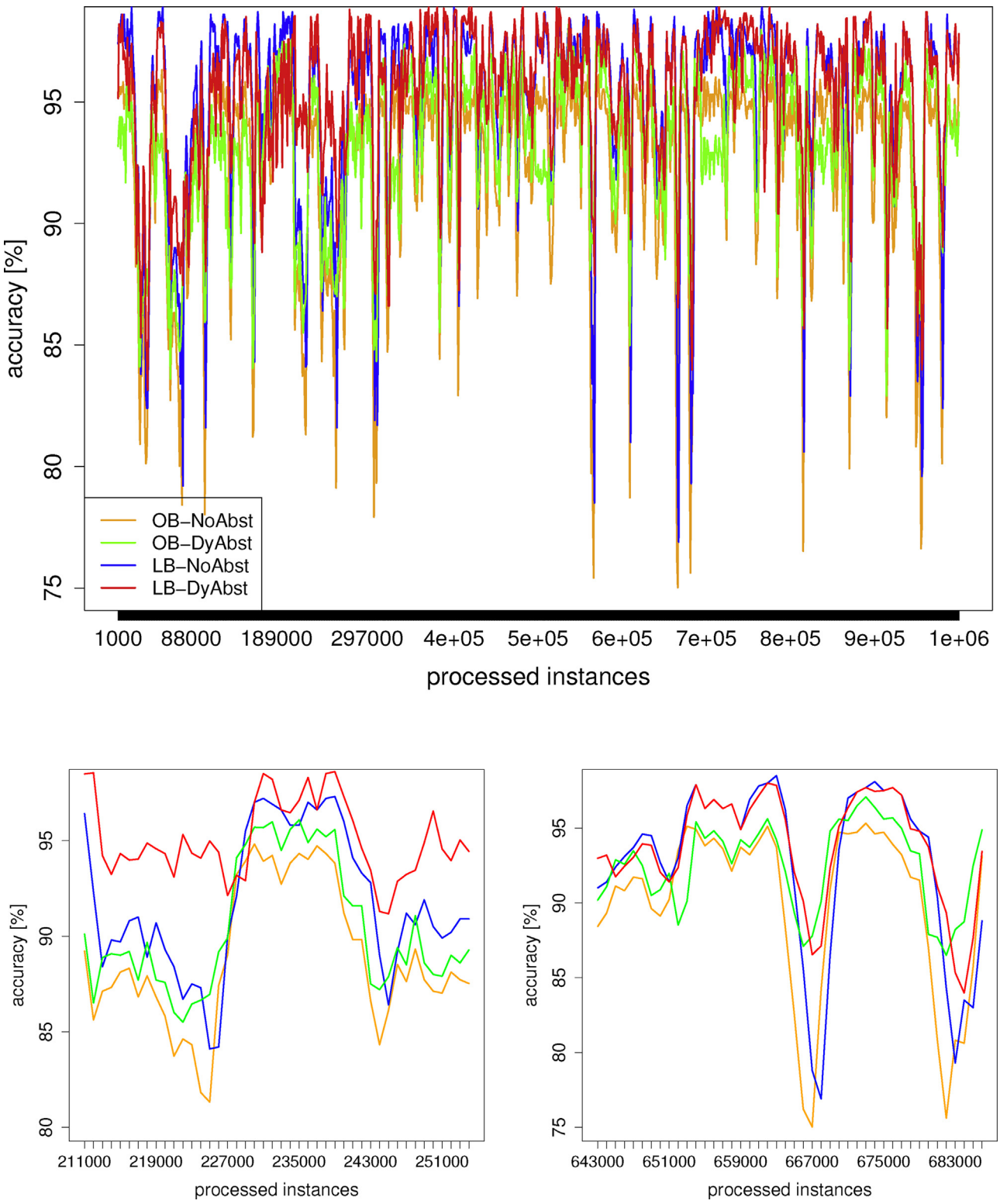


Fig. 5. (*Top*) Prequential accuracies of evaluated ensembles with Adaptive Hoeffding Tree as a base classifier averaged over windows of 1000 instances for RBF_grad data stream. (*Bottom*) Selected subsets of instances showcasing the drift recovery capabilities of examined methods.

a diverse set of base learners that can anticipate potential directions in which data may evolve. Selecting for voting only the ones that show faster recovery is a crucial factor in reducing the error when facing instances from emerging concept. Additionally, one must underline the interplay between the abstaining module and concept drift detector utilized by examined ensembles. When the weakest base classifier is replaced by the new one, the new model is expected to be the most competent one. It has been trained on only the most recent instances, while remaining models combine information extracted from recent and previous instances, thus mixing concepts. Here, such a single classifier is likely to be outvoted by other models, as they may adapt to novel concept much slower. By

using abstaining, we ensure that uncertain classifiers are excluded, thus increasing the chances of the new learner to guide the decision making process.

One must also analyze the situations in which abstaining returned unsatisfactory performance. For Hyp_slow data stream we may assume that the drop in accuracy is related to the slow nature of changes. The concept drift introduced here is incremental in nature, but the speed of change is low. A situation may happen in which local mistakes of ensemble lead to uneven ratio of threshold changes in comparison to actual drift. Classifiers that could still contribute to the decision making process were abstaining, weakening the collective predictive power. In case of RTree stream we deal with a highly randomized dataset. Due to the sudden and random nature of changes a situation is bound to happen, when a classifier trained on previous concept display a high certainty on the new instance arriving after a rapid drift (e.g., two classes suddenly exchanged their positions). It will lead to selecting classifiers that seem like competent ones, but in fact did not yet managed to accommodate new distribution of instances. Additionally, according to the way most of online ensembles are updated, concept drift detector can replace only one classifier at a time. Even with abstaining modification they are vulnerable to situations in which all of base classifiers suddenly suffer a significant drop in actual competence. Finally, we may assume that a combination of abstaining with NB classifier cannot properly capture properties of CovType stream.

Having discussed the general properties of dynamic abstaining ensembles, we will analyze the correlation between the ensemble size and achieved performance. Fig. 4 depicts the accuracies for ensembles having from 5 to 50 base learners. One can see two tendencies. For static streams or ones with small drifts both Online Bagging versions display behavior similar to the static Bagging. After growing to a certain size, adding new ensemble members do not contribute to the accuracy. On the other hand, larger pool of classifiers does not impact negatively the performance. This follows observations in many studies on the behavior of static Bagging. Yet, for drifting data streams we observe a different behavior. Here, ensembles improve their performance up to a certain ensemble size and then adding more base learners lead to a significant drop of accuracy. It can be explained by the negative impact of wrongly managed diversity. While more diverse pool of learners should allow for better anticipation of potential drifts, having many weak models may actually destabilize voting procedure during drifts. What is interesting, ensembles augmented with dynamic abstaining achieve better performance when using larger number of base

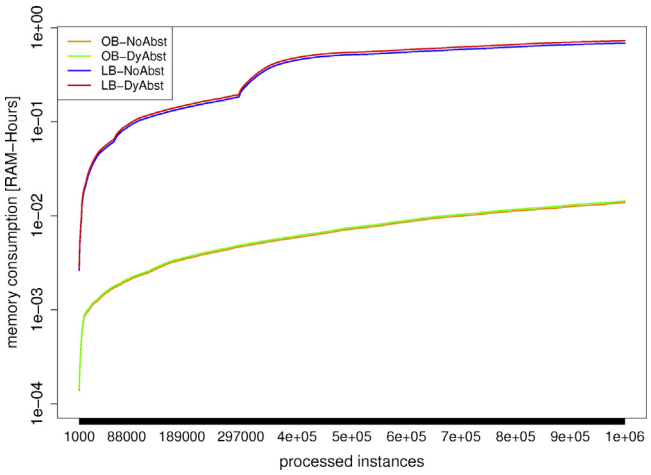


Fig. 6. Memory usage (log scale) of evaluated ensembles with Adaptive Hoeffding Tree as a base classifier averaged over windows of 1000 instances for RBF_grad data stream. Abstaining imposes almost no additional time complexity (between 3% and 6% more compared to the original version).

learners and display less significant drops in accuracy when ensemble size grows beyond the optimal point for given dataset. It further proves that abstaining allows to better manage diversity within ensemble, allowing it to be used for tackling concept drifts.

Fig. 5 shows the progress of prequential accuracy on RBF_grad data stream for ensembles using AHT as base classifier. The results were averaged over 1000 instances. By visual inspection one may easily see moments where concept drift occurred. Both canonical implementations of OB and LB display a significant error for a number of instances after the drift, showing that despite using drift detector they require some time to recover their performance. On the other hand, their abstaining versions are able to adapt much faster after the drift occurrence, which can be easily observed on the plot. They display much smaller sudden drops in accuracy and are able to recover faster. Abstaining increase the role of newly added (after drift) base classifier that is able to more efficiently contribute to the final decision. In many cases abstaining ensembles are also able to display improved performance on relatively static parts of the stream. We must notice few cases when they display higher error than canonical methods, most likely due to incorrectly estimated abstaining threshold that excludes too many classifiers. Nevertheless, it is important to note that abstaining versions always behave better than their canonical counterparts

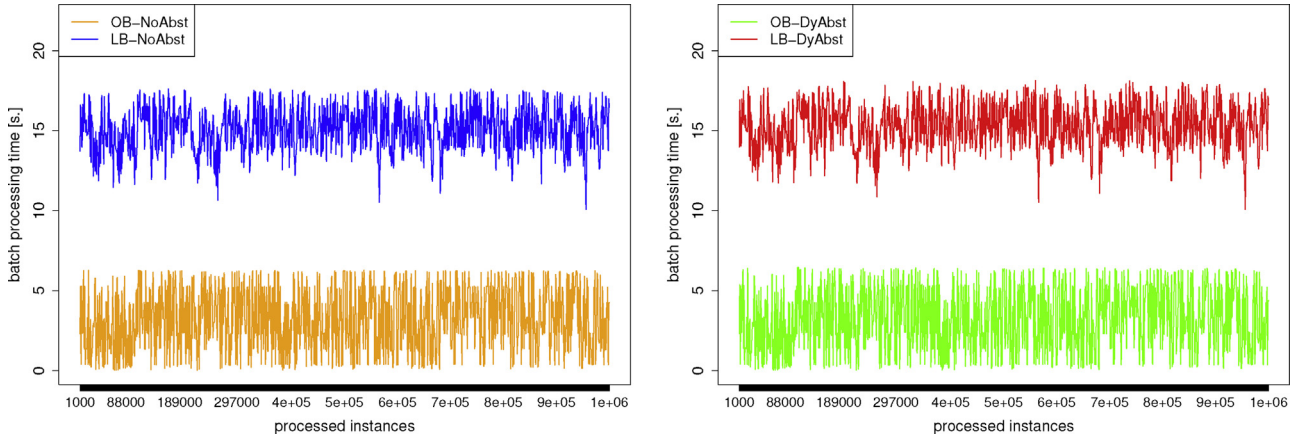


Fig. 7. Ensemble update time (test + train time) with Adaptive Hoeffding Tree as a base classifier averaged over windows of 1000 instances for RBF_grad data stream: *(left)* ensembles without abstaining, *(right)* ensembles with abstaining. Abstaining imposes almost no additional time complexity (between 0% and 3% more compared to the original version).

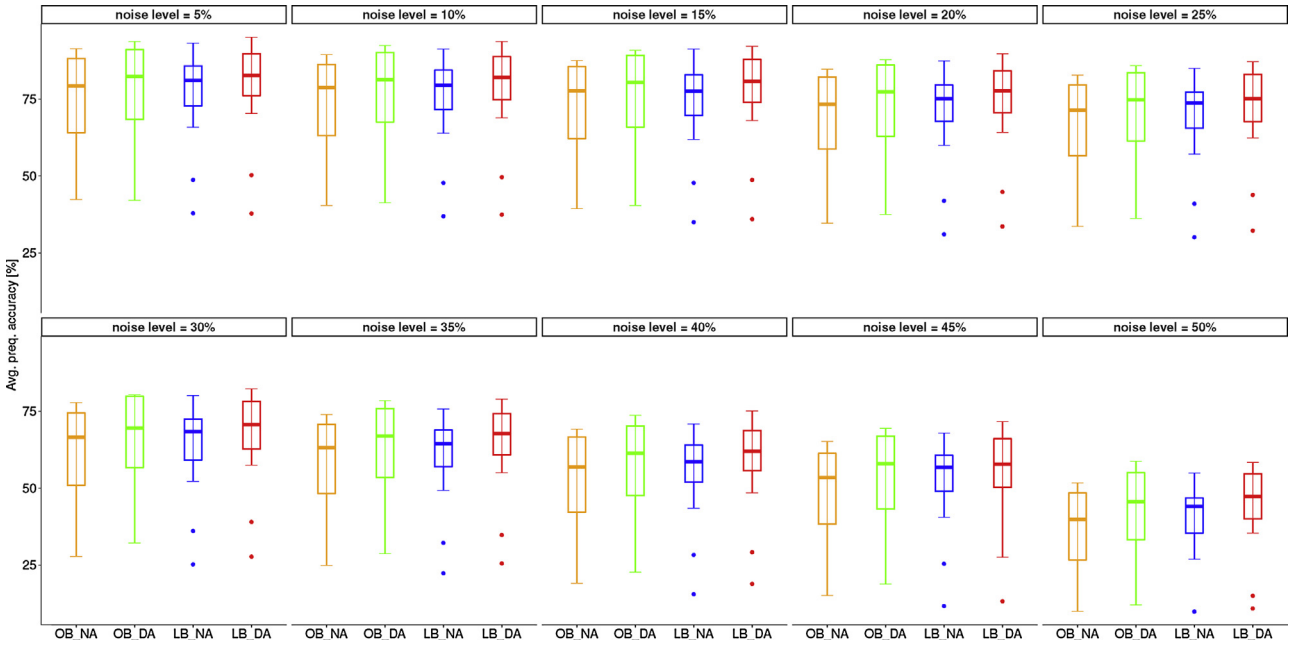


Fig. 8. Averaged prequential accuracy over all data streams with respect to varying level of noise for canonical and abstaining online ensembles with Adaptive Hoeffding Tree as a base classifier.

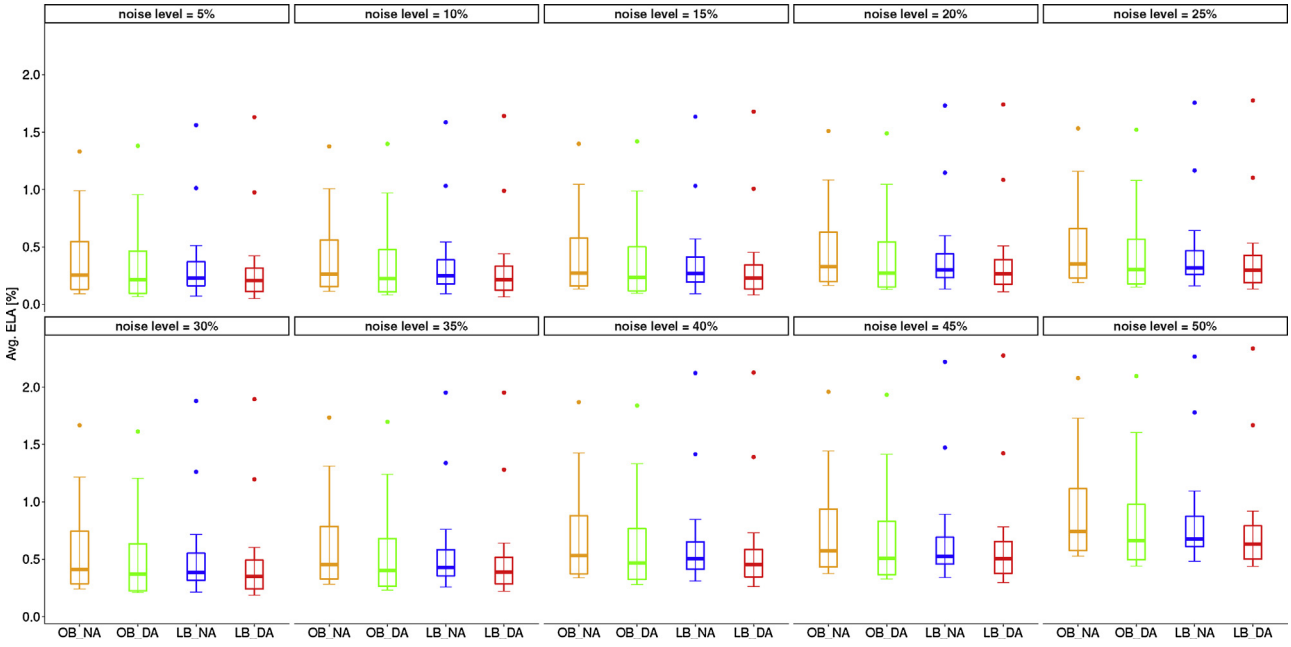


Fig. 9. Averaged ELA over all data streams with respect to varying level of noise for canonical and abstaining online ensembles with Adaptive Hoeffding Tree as a base classifier.

during drift periods, and is a highly desirable property for non-stationary environments.

By analyzing memory consumption and update times (see Table 3 and Figs. 6 and 7), one can see that the proposed abstaining modification imposes negligible additional computational costs upon the ensemble learning procedure. In most cases, we observe only small increase in training times (on average 2–4%) and used memory (3–6%). The classification time is always exactly the same, as we do not modify the voting procedure itself, only select classifiers that partake in it. However, one may also assume that the increase is partially due abstaining ensembles utilizing slightly larger pools of base classifiers, which will obviously affect the computational complexity. Still, conducted experiments prove that our modification is truly lightweight and does not have a negative impact on the computational requirements.

4.4. Experiment 2: Learning from drifting data streams under the presence of noise

The aim of the second experiment was to evaluate if the proposed dynamic abstaining modification can increase the robustness of online ensemble learning methods to noise present in data streams. The 12 datasets described in Table 3 were injected with randomized feature noise [40] ranging between 5% and 50%. It

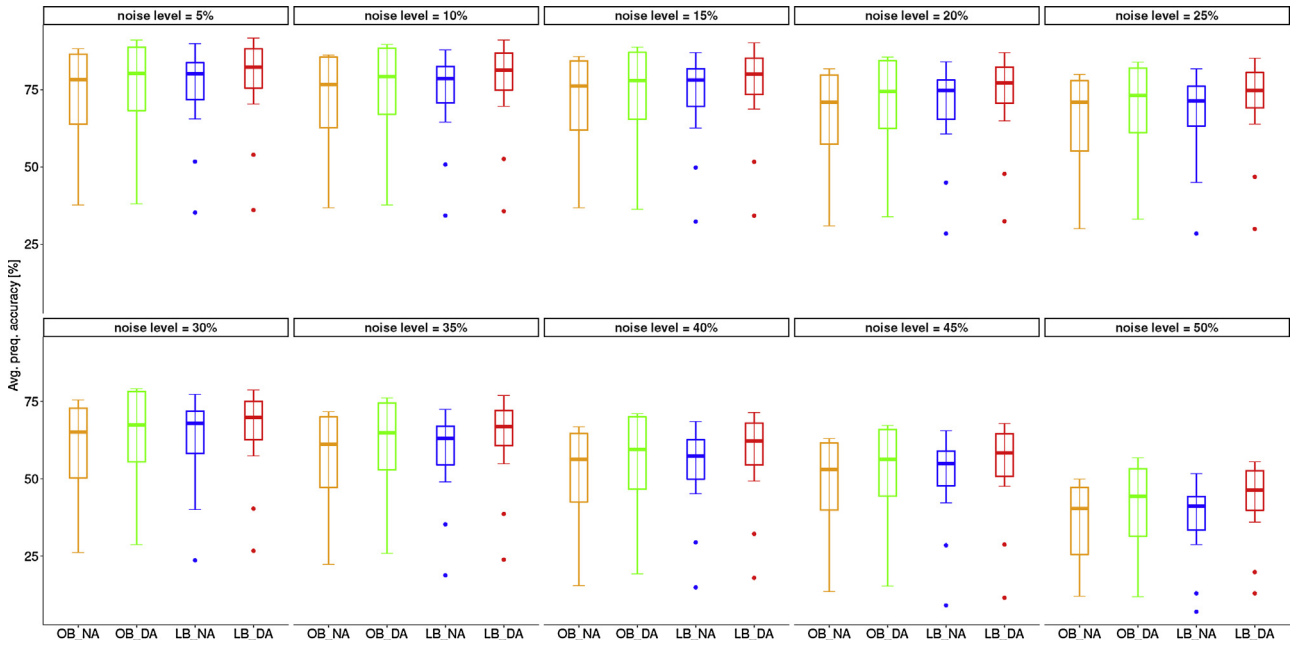


Fig. 10. Averaged prequential accuracy over all data streams with respect to varying level of noise for canonical and abstaining online ensembles with Naïve Bayes as a base classifier.

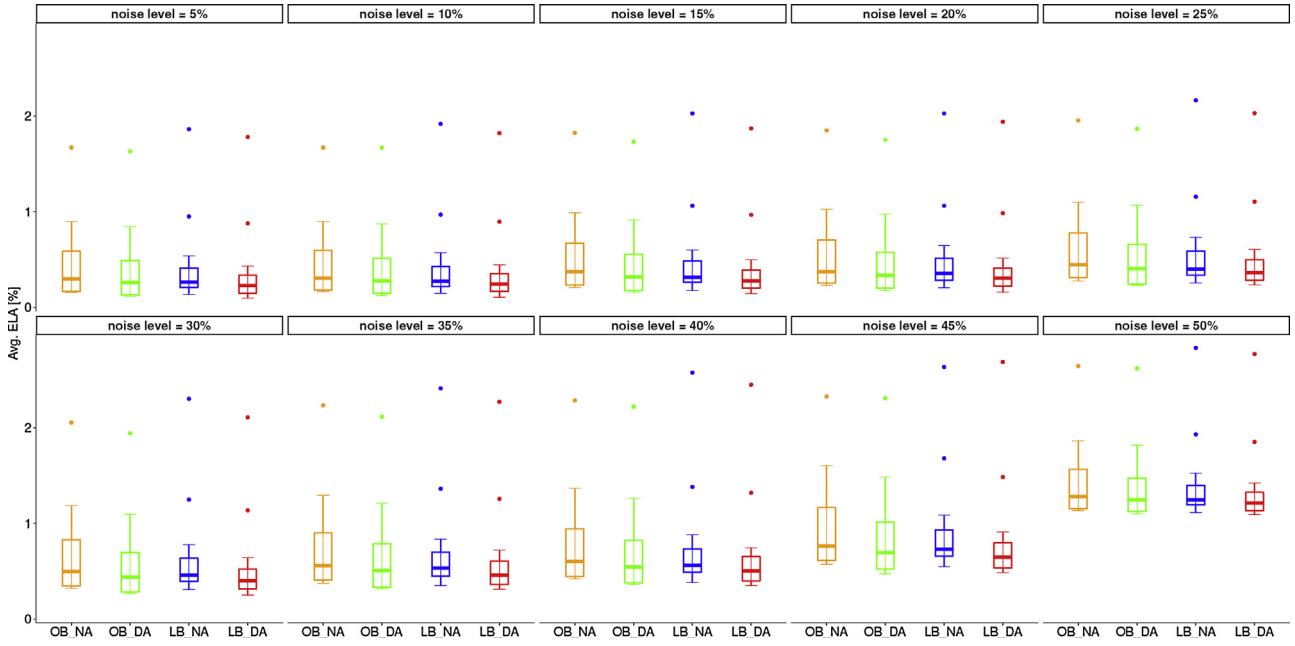


Fig. 11. Averaged ELA over all data streams with respect to varying level of noise for canonical and abstaining online ensembles with Naïve Bayes as a base classifier.

means that *x*% of attributes in the dataset are corrupted. To corrupt an attribute, approximately *x*% of the examples in the dataset are chosen and values of their selected features are replaced with random values from the original distribution. A uniform distribution is used either for numerical or nominal attributes. We repeat the random noise injection every 1000 instances in order to dynamically change features that are corrupted. If we would modify the same features for the entire data stream, then classifiers may actually learn an underlying noise distribution. We created 120 noisy data stream benchmarks that were used in the following experiments, leading to a thorough study of the influence of noise on online ensemble learning.

In order to provide an insight into the influence of the noise on classifier performance, we report not only the accuracy, but also ELA metric. It indicates, if the differences in accuracies under a certain level of noise are only due to the original differences between methods, or are actually caused by one of the method being more robust.

Prequential accuracies and ELA averaged over all of datasets with respect to varying level of noise and used base classifiers are given in Figs. 8–13. Full results for these datasets are to be found in supplementary materials to this paper. Results of Shaffer's post-hoc test over all of noise levels are given in Table 7. Additionally, a visualization of the correlation between the noise level and accuracy/ELA for LED_fast data stream is depicted in Fig. 14. We present

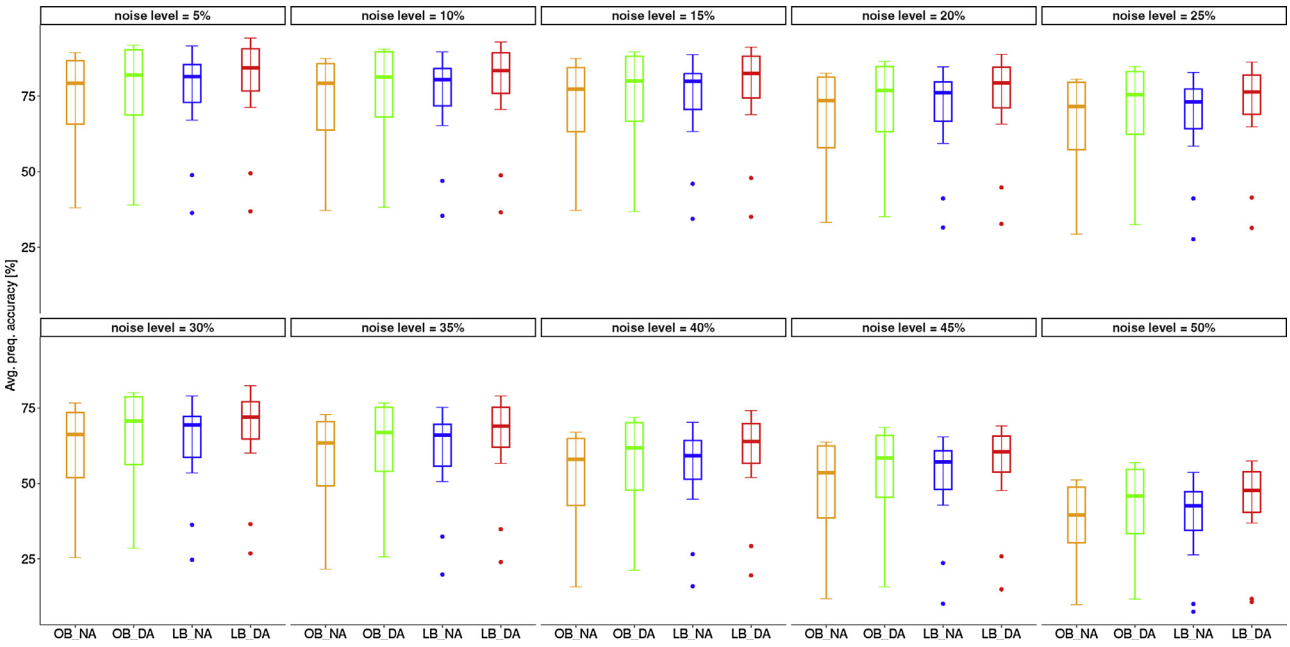


Fig. 12. Averaged prequential accuracy over all data streams with respect to varying level of noise for canonical and abstaining online ensembles with Multi-layer Perceptron as a base classifier.

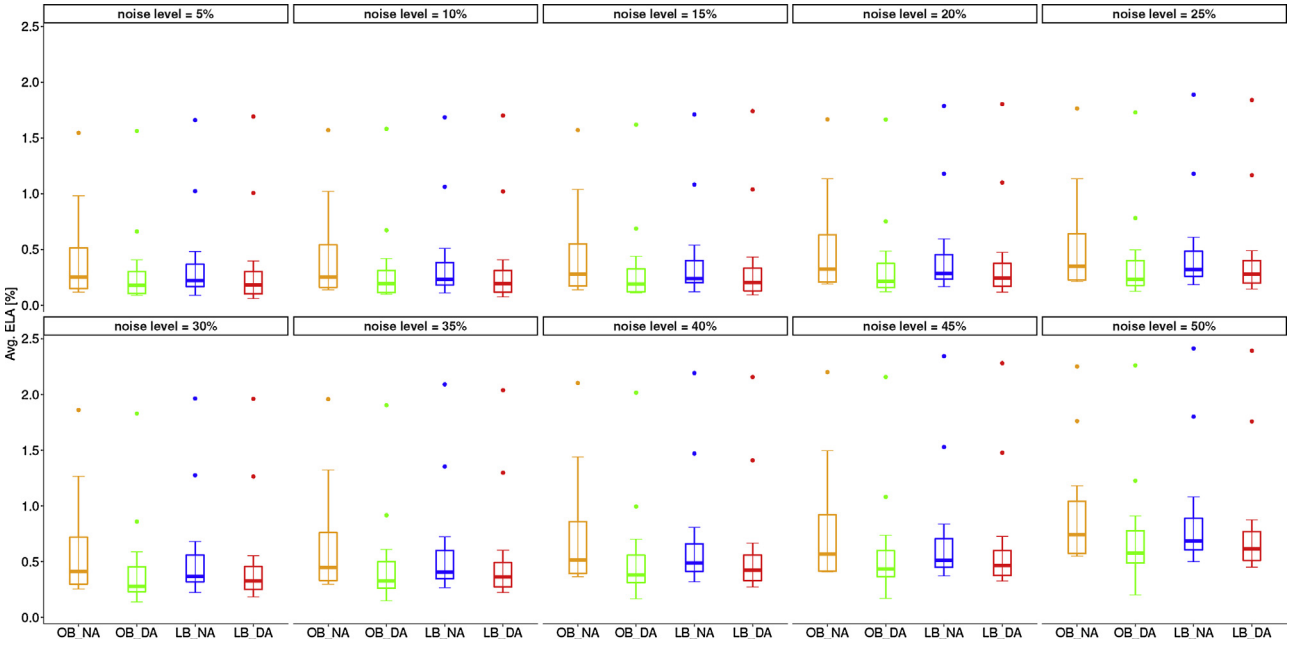
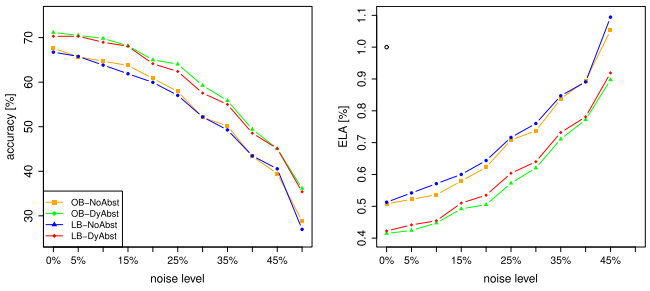


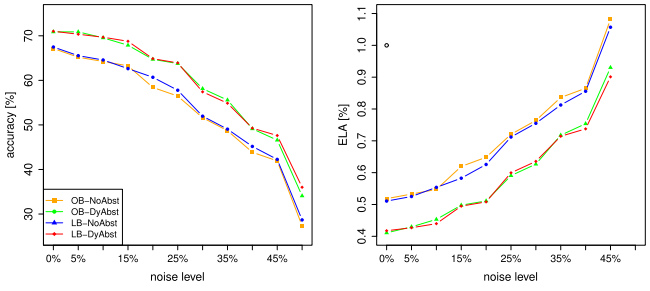
Fig. 13. Averaged ELA over all data streams with respect to varying level of noise for canonical and abstaining online ensembles with Multi-layer Perceptron as a base classifier.

Table 7
Shaffer's test for comparison between different ensemble approaches, averaged among all noise levels with respect to prequential accuracy and ELA metrics. Symbol '>' stands for situation in which dynamic abstaining approach is superior.

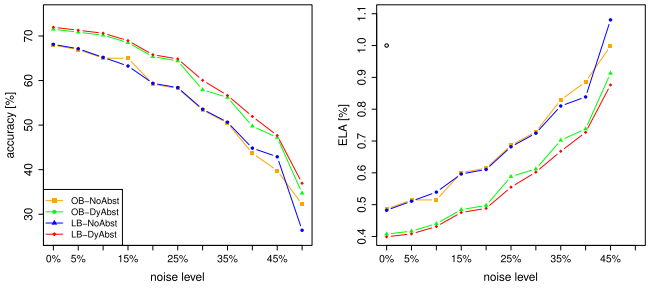
AHT		NB		MLP	
Hypothesis	*p*-value	Hypothesis	*p*-value	Hypothesis	*p*-value
Prequential accuracy					
OB-DyAbst vs. OB-NoAbst	>(0.102)	OB-DyAbst vs. OB-NoAbst	>(0.099)	OB-DyAbst vs. OB-NoAbst	>(0.107)
LB-DyAbst vs. LB-NoAbst	>(0.0108)	LB-DyAbst vs. LB-NoAbst	>(0.103)	LB-DyAbst vs. LB-NoAbst	>(0.111)
ELA					
OB-DyAbst vs. OB-NoAbst	>(0.032)	OB-DyAbst vs. OB-NoAbst	>(0.029)	OB-DyAbst vs. OB-NoAbst	>(0.033)
LB-DyAbst vs. LB-NoAbst	>(0.041)	LB-DyAbst vs. LB-NoAbst	>(0.40)	LB-DyAbst vs. LB-NoAbst	>(0.042)



(a) Accuracy and ELA for ensembles with Adaptive Hoeffding Tree.



(b) Accuracy and ELA for ensembles with Naïve Bayes.



(c) Accuracy and ELA for ensembles with Multi-layer Perceptron.

Fig. 14. Visualization of performance of examined ensemble methods for varying level of noise added to LED_fast data stream.

results only for canonical and dynamic abstaining ensembles, as static abstaining ensembles completely fail in noisy scenarios.

Obtained results clearly show that noisy data streams pose a significant challenge for standard online ensembles. While most of methods behave reasonably well with small levels of noise (5–10%), one may see that higher levels lead to significant drops in accuracy. Here, dynamic abstaining allows to alleviate accuracy loss, offering improved performance regardless of the level of noise. It is interesting to notice that even for datasets in which abstaining versions did not perform well (Hyp_slow and RTree), they outperform canonical approaches when noise is being introduced. One may argue that this happens due to their improved predictive power and those differences are preserved for all noise levels. One may analyze ELA results as companion to accuracy. Here, we can see that dynamic abstaining ensembles offer very significant improvement when compared to their canonical versions and such an advantage is preserved regardless of the noise level. It can be explained by the fact that when noise influences attributes, the corresponding instance is dislocated in the decision space. It holds an influence over the certainty of base classifiers, as the closer instance gets toward the decision boundary, the lower the certainty. Therefore, such classifiers will be abstaining and become excluded from the voting. As we use diversified ensembles, there is a high chance that some of base classifiers will use such a decision boundary that will display good performance despite the level of noise. By reducing the number of voting members, we reduce the probability of noisy instance affecting majority of them and leading to an incorrect decision.

Of course such a strategy is not guaranteed to work in all of cases. It is easy to come up with a counterexample, where an instance will be affected so strongly by the noise that it will shift to the opposite side of the decision boundary and will lead to selection of a classifier with a high certainty, but incorrect prediction. Such cases are bound to happen when the noise level is high. However, due to using an ensemble approach it becomes less likely that all of base classifiers will be affected in such a way.

It should be stressed that proposed method is a simple and flexible modification that can be applied to most online ensemble classifiers. Existing approaches for noisy data streams are based either on costly filters or using specific learning algorithms. Our aim was not to prove that we are better. Our aim was to show that with no direct cost and no influence on accuracy we are able to improve robustness to noise of popular online ensemble learners. In most real-life scenarios noise is unexpected and does not appear thorough the entire stream processing time. Therefore, one cannot predict when to switch to a certain noise-handling approach. Our abstaining modification can be used all the time during mining drifting data streams, and when noise will appear it will lead to an improved robustness of the underlying ensemble.

5. Conclusions and future works

In this paper we have introduced a lightweight dynamic abstaining approach that can be used to augment any online ensemble learning scheme. We utilized a certainty threshold that is used to determine which ensemble members are allowed to participate in voting. If the certainty of given base classifier is below the threshold, we allow it to abstain from making a decision and exclude it from the current class label prediction. Such a procedure is being repeated for each instance from the data stream, thus leading to an indirect dynamic ensemble selection, as for each new instance different classifiers may form a sub-ensemble. We proposed an adaptive strategy to calculate the threshold during the data stream progress. It was based on monitoring the correctness of ensemble decision and adapting the abstaining value accordingly. It allowed to track changes in the stream and promote most competent classifiers. We showed that our proposal allowed to exploit the underlying diversity in online ensembles, taking advantage of larger pool of base learners. The proposed approach imposes almost no additional cost on the original ensemble learning scheme, thus making it an attractive proposition for general-purpose mining of non-stationary data streams. Additionally, we have showed that the proposed dynamic abstaining is able to improve robustness of online ensembles to noise present in data streams.

Experimental study utilizing 12 data stream benchmarks and 120 noisy data streams proved the efficiency of dynamic abstaining modification. We examined the performance of our approach on two popular online ensemble models and three online base learners, backing-up our observations with statistical testing. Robustness to noise was evaluated using both accuracy and dedicated ELA measure to show that the good performance of our method can truly be contributed to better robustness.

Obtained results encourage us to continue our works. We plan to further investigate the methods for determining which classifiers should abstain and to further improve their ability to handle various concept drifts. At the same time, we plan to maintain the low computational complexity of our methods, thus we do not plan to follow the directions of dynamic ensemble selection methods used for data streams, as they require too much processing time

and additional instances. Finally, we envision adapting our method to mining recurrent concept drifts, by making members making abstain on new concepts, and retrieving them when one previously seen reemerges.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.asoc.2017.12.008.

References

[1] X. Wu, X. Zhu, G. Wu, W. Ding, Data mining with big data, IEEE Trans. Knowl. Data Eng. 26 (2014) 97–107.
[2] E. Lughofer, P.P. Angelov, Handling drifts and shifts in on-line data streams with evolving fuzzy systems, Appl. Soft Comput. 11 (2011) 2057–2068.
[3] M. Chen, B. Chen, A hybrid fuzzy time series model based on granular computing for stock price forecasting, Inf. Sci. 294 (2015) 227–241.
[4] E. Lughofer, E. Weigl, W. Heidl, C. Eitzinger, T. Radauer, Integrating new classes on the fly in evolving fuzzy classifier designs and their application in visual inspection, Appl. Soft Comput. 35 (2015) 558–582.
[5] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: a survey, Inf. Fusion 37 (2017) 132–156.
[6] T. Pietraszek, On the use of ROC analysis for the optimization of abstaining classifiers, Mach. Learn. 68 (2007) 137–169.
[7] T. Pietraszek, Classification of intrusion detection alerts using abstaining classifiers, Intell. Data Anal. 11 (2007) 293–316.
[8] J. Błaszczyński, J. Stefanowski, M. Zajac, Ensembles of abstaining classifiers based on rule sets, International Symposium on Methodologies for Intelligent Systems (2009) 382–391.
[9] L.L. Minku, A.P. White, X. Yao, The impact of diversity on online ensemble learning in the presence of concept drift, IEEE Trans. Knowl. Data Eng. 22 (2010) 730–742.
[10] T. Windeatt, Accuracy/diversity and ensemble MLP classifier design, IEEE Trans. Neural Netw. 17 (2006) 1194–1211.
[11] M.M. Gaber, Advances in data stream mining, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2 (2012) 79–85.
[12] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (2014), 44:1–44:37.
[13] G. Ditzler, M. Roveri, C. Alippi, R. Polikar, Learning in nonstationary environments: a survey, IEEE Comp. Int. Mag. 10 (2015) 12–25.
[14] L.I. Kuncheva, Classifier ensembles for detecting concept change in streaming data: overview and perspectives, 2nd Workshop SUEMA (ECAI 2008) (2008) 5–10.
[15] J. Gama, P. Medas, G. Castillo, P.P. Rodrigues, Learning with drift detection, in: Advances in Artificial Intelligence – SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, Proceedings, São Luis, Maranhão, Brazil, September 29–October 1, 2004, pp. 286–295.
[16] A. Bifet, R. Gavaldà, Learning from time-changing data with adaptive windowing, in: Proceedings of the Seventh SIAM International Conference on Data Mining, Minneapolis, MN, USA, April 26–28, 2007, pp. 443–448.
[17] P. Sobolewski, M. Woźniak, Concept drift detection and model selection with simulated recurrence and ensembles of statistical detectors, J. Univers. Comput. Sci. 19 (2013) 462–483.
[18] M. Woźniak, A hybrid decision tree training method using data streams, Knowl. Inf. Syst. 29 (2011) 335–347.
[19] J. Shan, J. Luo, G. Ni, Z. Wu, W. Duan, CVS: fast cardinality estimation for large-scale data streams over sliding windows, Neurocomputing 194 (2016) 107–116.
[20] P. Domingos, G. Hulten, Mining high-speed data streams, in: I. Parsa, R. Ramakrishnan, S. Stolfo (Eds.), Proceedings of the ACM Sixth International Conference on Knowledge Discovery and Data Mining, ACM Press, Boston, USA, 2000, pp. 71–80.
[21] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17.
[22] M. Woźniak, Application of combined classifiers to data stream classification, in: Computer Information Systems and Industrial Management – 12th IFIP TC8 International Conference, CISIM 2013, Proceedings, Krakow, Poland, September 25–27, 2013, pp. 13–23.
[23] Y. Sun, K. Tang, L.L. Minku, S. Wang, X. Yao, Online ensemble learning of data streams with gradually evolved classes, IEEE Trans. Knowl. Data Eng. 28 (2016) 1532–1545.
[24] J. Stefanowski, Adaptive ensembles for evolving data streams – combining block-based and online solutions, in: New Frontiers in Mining Complex Patterns – 4th International Workshop, NFMCP 2015, Held in Conjunction with ECML-PKDD 2015, Revised Selected Papers, Porto, Portugal, September 7, 2015, pp. 3–16.
[25] J. Gama, R. Sebastião, P.P. Rodrigues, On evaluating stream learning algorithms, Mach. Learn. 90 (2013) 317–346.
[26] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, T.I. Ren, META-DES: a dynamic ensemble selection framework using meta-learning, Pattern Recognit. 48 (2015) 1925–1935.
[27] P. Trajdos, M. Kurzynski, A dynamic model of classifier competence based on the local fuzzy confusion matrix and the random reference classifier, Appl. Math. Comput. Sci. 26 (2016) 175.
[28] M. Woźniak, P. Ksieniewicz, B. Cyganek, A. Kasprzak, K. Walkowiak, Active learning classification of drifted streaming data, in: International Conference on Computational Science 2016, ICCS 2016, San Diego, CA, USA, June 6–8, 2016, pp. 1724–1733.
[29] J.A. Sáez, M. Galar, J. Luengo, F. Herrera, INFFC: an iterative class noise filter based on the fusion of classifiers with noise sensitivity control, Inf. Fusion 27 (2016) 19–32.
[30] X. Zhu, P. Zhang, X. Wu, D. He, C. Zhang, Y. Shi, Cleansing noisy data streams, in: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), Pisa, Italy, December 15–19, 2008, pp. 1139–1144.
[31] S. Hashemi, Y. Yang, Flexible decision tree for data stream classification in the presence of concept change, noise and missing values, Data Min. Knowl. Discov. 19 (2009) 95–131.
[32] P. Li, X. Wu, X. Hu, Q. Liang, Y. Gao, A random decision tree ensemble for mining concept drifts from noisy data streams, Appl. Artif. Intell. 24 (2010) 680–710.
[33] P. Zhang, X. Zhu, Y. Shi, L. Guo, X. Wu, Robust ensemble learning for mining noisy data streams, Decis. Support Syst. 50 (2011) 469–479.
[34] X. Zhu, X. Wu, Y. Yang, Dynamic classifier selection for effective mining from noisy data streams, in: Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), Brighton, UK, November 1–4, 2004, pp. 305–312.
[35] N.C. Oza, S.J. Russell, Online bagging and boosting, in: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics, AISTATS 2001, Key West, FL, US, January 4–7, 2001.
[36] A. Bifet, G. Holmes, B. Pfahringer, Leveraging bagging for evolving data streams, in: Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Proceedings, Part I, Barcelona, Spain, September 20–24, 2010, pp. 135–150.
[37] A. Bifet, R. Gavaldà, Adaptive learning from evolving data streams, in: Advances in Intelligent Data Analysis VIII, 8th International Symposium on Intelligent Data Analysis, IDA 2009, Proceedings, Lyon, France, August 31–September 2, 2009, pp. 249–260.
[38] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064.
[39] J.A. Sáez, J. Luengo, F. Herrera, Evaluating the classifier behavior with noisy data considering performance and robustness: the equalized loss of accuracy measure, Neurocomputing 176 (2016) 26–35.
[40] X. Zhu, X. Wu, Class noise vs. attribute noise: a quantitative study, Artif. Intell. Rev. 22 (2004) 177–210.