An appendix to the paper "Improving Online Bagging for Complex Imbalanced Data Streams"

Bartosz Przybył and Jerzy Stefanowski

Institute of Computing Science, Poznań University of Technology, Poznań, Poland
bprzybyl98@gmail.com, jerzy.stefanowski@cs.put.poznan.pl

Abstract. This paper significantly extends our earlier paper "Improving Online Bagging for Complex Imbalanced Data Streams" [10] concerning two issues. Firstly, we explain in more detail all versions of online bagging ensembles for imbalanced data streams and also present more precisely pseudocodes of our new Neighbourhood Online Bagging ensembles. Secondly, we provide some additional results of experiments, which could not be introduced to the main paper due to its page limits. These experiments with synthetic complex imbalanced data streams have demonstrated the advantage of our proposals over earlier variants of online bagging resampling ensembles.

Keywords: data streams · class imbalance · data complexity · online bagging · neighbourhood analysis · types of minority examples

1 Introduction

This report is an appendix to the main paper on "Improving Online Bagging for Complex Imbalanced Data Streams" accepted and published at ECMLPKDD 2024 workshop [10]. While the main paper contains motivations and related work and introduces proposed modifications to the online ensemble for imbalanced data streams, this appendix contains additional information that could not be included due to a limited number of pages.

In particular, we explain in more detail all versions of online bagging ensembles for imbalanced data streams and also present more precisely pseudocodes of our proposals - Neighbourhood Oversampling Online Bagging, Neighbourhood Undersampling Online Bagging, and Hybrid Neighbourhood Online Bagging.

Furthermore, we provide some additional information on the generators of synthetic imbalanced data streams which are used to carry out experiments with these online classifiers, where the impact of different class imbalanced data factors and the local drifts is studied. It also contains some additional results of experiments, which could not be introduced to the main paper due to its page limits.

2 Neighbourhood Online Bagging

In this section, we would like to present the general idea and pseudocodes of our approaches (*Neighbourhood Oversampling Online Bagging (NOOB)*, *Neighbour-*

hood Undersampling Online Bagging (NUOB) and *Hybrid Neighbourhood Online Bagging (HNOB)*) along with the algorithms on which they are based - *Online Bagging (OB)*, *Oversampling Online Bagging (OOB)* and *Undersampling Online Bagging (UOB)*. In this work, we will present their pseudocodes and description in more detail than it is done in the main text.

Starting with the *Online Bagging* algorithm [9], which, unlike Breiman's static bagging ensemble, handles online environments by processing each example only once upon arrival. It uses the Poisson distribution parameter λ to determine how many times the current example is sent to each component classifier (typically $\lambda = 1$ for balanced classes). Each classifier is updated incrementally with these examples. The general pseudocode of the Online Bagging is presented in Algorithm 1.

Algorithm 1 Online Bagging (OB) [9]

Input: S: stream of examples
n: number of classifiers in ensemble
Output: $\mathcal{E}$: an ensemble of classifiers

```
1: E ← n incremental classifiers
2: for all examples x ∈ S do
3:     for all classifiers C_i ∈ E do          ▷ C_i - single classifier of ensemble
4:         set l ~ Poisson(1)
5:         update C_i using x, l times
6:     end for
7: end for
```

Oversampling-based Online Bagging (OOB) and *Undersampling-based Online Bagging* (UOB) were proposed in [12,13] to adjust the presence of class examples in the Poisson distribution based on the current imbalance ratio in the stream.

The number of examples from each class, sent to incrementally learn component classifiers (see the parameter l in Algorithm 1), is continuously updated to calculate λ as a function of these ratios. For OOB, λ is the ratio of the largest class size to the size of the current class, increasing minority class examples ($\lambda > 1$) while keeping the majority ones unchanged ($\lambda = 1$). For UOB, λ is the ratio of the smallest class size to the size of the current class, reducing the number of majority examples ($\lambda < 1$).

In summary, these re-sampling methods adjust how examples are sent to update classifiers in online bagging, based on the current class imbalance ratio in the stream. The general pseudocode of the Oversampling Online Bagging is presented in Algorithm 2 and Undersampling Online Bagging is presented in Algorithm 3.

Some other extensions of the online bagging classifiers have been also considered in [11,6].

Algorithm 2 Oversampling Online Bagging (OOB) [12,13]

Input: S: stream of examples
n: number of classifiers in ensemble
Output: $\mathcal{E}$: an ensemble of classifiers

$\mathcal{E} \leftarrow n$ incremental classifiers
for all examples $x \in S$ **do**
  if $x \in$ *minority class* **then**
    $\lambda \leftarrow (N_{maj}/N_{min})$ ▷ N_{maj} - number of examples from majority class
  else ▷ N_{min} - number of examples from minority class
    $\lambda \leftarrow 1$
  end if
  for all classifiers $C_i \in \mathcal{E}$ **do** ▷ C_i - single classifier of ensemble
    set $l \sim Poisson(\lambda)$
    update C_i using x, l times
  end for
end for

Algorithm 3 Undersampling Online Bagging (UOB) [12,13]

Input: S: stream of examples
n: number of classifiers in ensemble
Output: $\mathcal{E}$: an ensemble of classifiers

$\mathcal{E} \leftarrow n$ incremental classifiers
for all examples $x \in S$ **do**
  if $x \in$ *majority class* **then**
    $\lambda \leftarrow (N_{min}/N_{maj})$ ▷ N_{min} - number of examples from minority class
  else ▷ N_{maj} - number of examples from majority class
    $\lambda \leftarrow 1$
  end if
  for all classifiers $C_i \in \mathcal{E}$ **do** ▷ C_i - single classifier of ensemble
    set $l \sim Poisson(\lambda)$
    update C_i using x, l times
  end for
end for

In our current proposal, we have decided to modify the λ coefficient by incorporating the difficulty of the incoming example. Previously, to estimate example difficulty in static data, the local analysis of class labels among its k nearest neighbors was used [2][1]. Such an idea of the local analysis of the difficulty of the examples was originally introduced in [8]. Building on this, we define the *unsafeness level* of a minority class example x as

$$L^2_{min} = \frac{(N'_{maj})^{\Psi}}{k}, \tag{1}$$

where N'_{maj} is the number of examples belonging to the majority class among k nearest neighbours of x which are calculated on a sliding window in the stream, k is the number of nearest neighbours taken for the analysis, Ψ is a parameter responsible for additional amplification of the impact of unsafe examples (i.e. value $\Psi > 1$ amplifies the role of rare cases and outliers). In the *Neighbourhood Oversampling Online Bagging* (NOOB) this coefficient is aggregated with the class sizes coefficient to increase the number of unsafe minority examples send to update the component classifiers, which is defined as follows:

$$\lambda = (N_{maj}/N_{min}) \cdot (L^2_{min} + 1), \tag{2}$$

where N_{maj} and N_{min} represent the number of majority and minority class examples in the sliding window, respectively.

In (NOOB) it increases the Poisson distribution estimate of the number of minority examples while for the incoming majority examples $\lambda = 1$ (as in the standard online bagging). The general pseudocode of the Neighbourhood Oversampling Online Bagging is presented in Algorithm 4.

In a similar way, we propose *the Neighbourhood Undersampling Online Bagging* (NUOB), where for the incoming majority examples we define a *safeness level* as

$$L^2_{maj} = \frac{N_{maj}}{k}, \tag{3}$$

and as its consequence a new coefficient for the majority example

$$\lambda = (N_{min}/N_{maj}) \cdot (L^2_{maj})^{\Psi}, \tag{4}$$

which reduces the chance of using unsafe majority examples to update component classifiers. By setting $\lambda = 1$ for incoming minority examples, it enables undersampling that removes rare cases, outliers, and some borderline majority examples. The general pseudocode of Neighbourhood Undersampling Online Bagging is presented in Algorithm 5.

Finally, our preliminary experiments have shown that sometimes NUOB outperforms NOOB, and vice versa, leading us to propose a hybrid version.

In *Hybrid Neighbourhood Online Bagging* (HNOB), both NUOB and NOOB ensembles are trained in parallel on each incoming example. Their performance

[1] Although the first attempts were considered in the slightly earlier work [3]

Algorithm 4 Neighbourhood Oversampling Online Bagging (NOOB)

Input: S: stream of examples
n: number of classifiers in ensemble
W: window of examples
k: number of nearest neighbours
Ψ: additional coefficient for calculating safe level
Output: $\mathcal{E}$: an ensemble of classifiers

```
for all examples x ∈ S do
    calculate safe level of incoming example L²_min = (N'_maj)^Ψ / k
    if x ∈ minority class then
        λ ← (N_maj/N_min) · (L²_min + 1)
    else
        λ ← 1
    end if
    for all classifiers C_i ∈ E do                ▷ C_i - single classifier of ensemble
        set l ∼ Poisson(λ)
        update C_i using x, l times
    end for
    W ← W ∪ {x}
    if necessary remove outdated examples from W
end for
```

Algorithm 5 Neighbourhood Undersampling Online Bagging (NUOB)

Input: S: stream of examples
n: number of classifiers in ensemble
W: window of examples
k: number of nearest neighbours
Ψ: additional coefficient for calculating safe level
Output: $\mathcal{E}$: an ensemble of classifiers

```
for all examples x ∈ S do
    calculate safe level of incoming example L²_maj = N_maj / k
    if x ∈ majority class then
        λ ← (N_min/N_maj) · (L²_maj)^Ψ
    else
        λ ← 1
    end if
    for all classifiers C_i ∈ E do                ▷ C_i - single classifier of ensemble
        set l ∼ Poisson(λ)
        update C_i using x, l times
    end for
    W ← W ∪ {x}
    if necessary remove outdated examples from W
end for
```

is continuously evaluated using a metric suitable for imbalanced data (in our case, the *G–mean* of both classes). The ensemble with the better evaluation measure in the last assessment is then selected to make the class prediction for the current example in the stream. The pseudocode of Hybrid Neighbourhood Online Bagging approach is presented in Algorithm 6.

Algorithm 6 Hybrid Neighbourhood Online Bagging (HNOB)

Input: S: stream of examples
$NOOB$: Neighbourhood Oversampling Online Bagging classifier
$NUOB$: Neighbourhood Undersampling Online Bagging classifier

for all examples $x \in S$ **do** ▷ g_o - current value of *G–mean* for $NOOB$
  if $g_o \geq g_u$ **then** ▷ g_u - current value of *G–mean* for $NUOB$
    Make prediction for x using $NOOB$
  else
    Make prediction for x using $NUOB$
  end if
  Update *G–mean* values g_o for $NOOB$ and g_u for $NUOB$
  Train $NOOB$ and $NUOB$ using x
end for

3 Additional Experimental Results

3.1 Experimental Setup

In this subsection, we give more details on the experimental setup. Let us recall that the aim of our experiments is to investigate the influence of the discussed earlier data difficulty factors and drifts on the predictive performance of selected online classifiers. As these experiments refer to the earlier study with binary imbalanced streams, we follow and extend its experimental setup [4]. We decided to consider the binary classes only, although it could be also generalized for multiple classes as it was done in [7].

Our experiments cover the following difficulty factors and drifts:

- *Imbalanced ratios*
  - *Static imbalance* - the percentage of minority class examples present in the stream from the beginning (from 1% to the fully balanced class ratio)
  - *Imbalance drift* - the percentage of minority class examples that will appear in the data stream after concept drift (we considered similar class global ratios)
- *Types of minority examples*

  - *Borderline* - the percentage of minority class examples labeled as borderline that will appear in the stream after concept drift is observed
  - *Rare* - the percentage of minority class examples labeled as rare that will appear in the stream after concept drift is observed

  We considered both the static amounts of such examples from the beginning of the stream as well as the drifts from 0% percentage to the given one (the following percentages are considered: 20%, 40%, 60%, 80%, and the most extreme 100%).
- *Changes in class composition*
  - *Class split drift* - split of each minority concepts into smaller sub-clusters
  - *Class merge drift* - merge of existing minority sub-clusters into larger concepts
  - *Class move drift* - move of existing minority sub-clusters in the attribute space

  In this case of class composition - we considered 3, 5, and 7 possible sub-concepts.

We carried out the experiments in a controlled framework based on synthetic generated data, where each data factor can be modeled and parametrized according to different planned scenarios.

We used the same generator as used in the previous experiments [4], where minority classes are generated in elliptical spheres and the majority class instances uniformly surround them[2]. Please note that quite similar analysis of synthetic datasets is also considered in a quite recent paper [1].

This section is divided into several subsections, where each subsection addresses a different number of difficulty factors that can occur in the data stream. In these subsections, analysis of the results of existing algorithms (*basic online bagging, undersampling online bagging, oversampling online bagging*) is performed alongside the new approaches (*neighbourhood oversampling online bagging, neighbourhood undersampling online bagging*). In the last subsection, the focus is mainly put on the hybrid approach (*hybrid neighbourhood online bagging*) with the presentation of its results on concept drifting data streams.

We have considered two basic evaluation measures for prediction of the ensemble classifiers: Recall of both classes and their aggregation into G-mean: for justification of choosing them, see e.g. [5,12].

3.2 Data streams with single factors

Starting from studies on the impact of the global imbalanced ratio the following observation has been made - it is well coped by nearly all specialized online bagging, and the new neighbourhood online bagging does not strongly outperform earlier re-sampling online bagging ensembles - see Figure 1. However, it is not a surprise as it was expected - see earlier studies [4]. Indeed the new *Neighbourhood Oversampling Online Bagging, Neighbourhood Undersampling Online*

[2] See https://github.com/dabrze/imbalanced-stream-generator for its description and code.

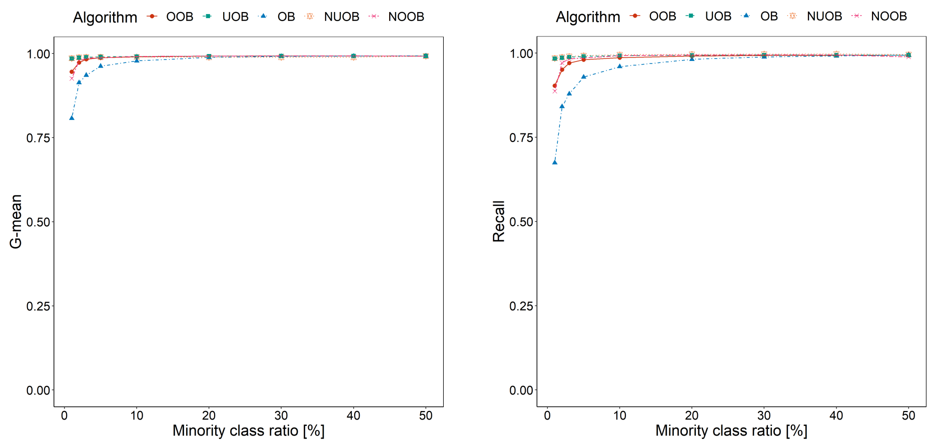

Fig. 1. Plots showing aggregated *G-mean* measures (the left-hand figure) and *Recall* (the right-hand figure) of algorithms reacting to different percentage of *static imbalance* drift

Bagging work with the modification of λ parameter in a quite similar way to the version already used inside *Undersampling Online Bagging* and *Oversampling Online Bagging*.

	OB	UOB	OOB	NUOB	NOOB
start	0.9959	0.9959	0.9959	0.9959	0.9940
pre-dirft	0.9971	0.9966	0.9971	0.9967	0.9966
post-drift	0.9698	0.9653	0.9678	0.9701	0.9636
end	0.9914	0.9902	0.9898	0.9914	0.9900

	OB	UOB	OOB	NUOB	NOOB
start	1.0000	1.0000	1.0000	1.0000	0.9941
pre-dirft	0.9976	0.9975	0.9976	0.9987	0.9959
post-drift	0.9634	0.9565	0.9619	0.9707	0.9482
end	0.9930	0.9919	0.9914	0.9946	0.9877

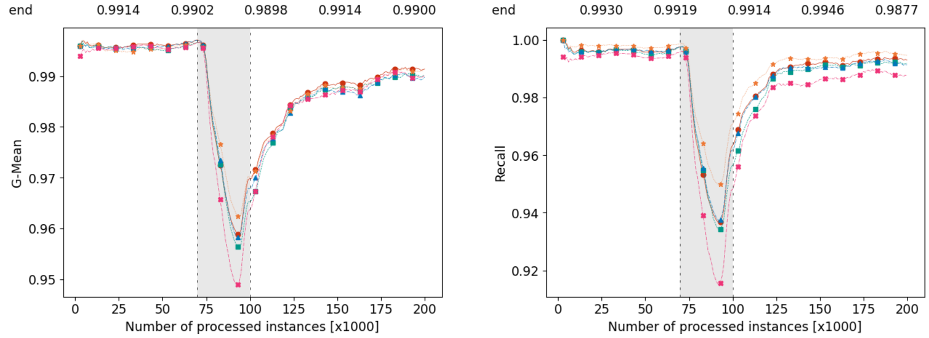

Fig. 2. Plots showing *G-mean* measure (the left-hand figure) and *Recall* measure (the right-hand figure) of bagging variants reacting to *minority class split into 5 sub-concepts* drift

For class decomposition, proposed *Neighbourhood Oversampling Online Bagging, Neighbourhood Undersampling Online Bagging* have performed slightly bet-

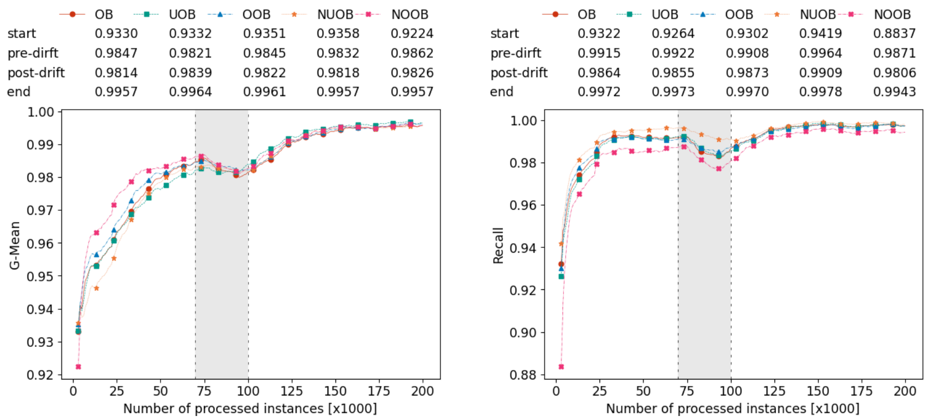

Fig. 3. Plots showing *G-mean* measure (the left-hand figure) and *Recall* measure (the right-hand figure) of bagging variants reacting to *minority class merge into 5 sub-concepts* drift

ter than *Undersampling Online Bagging, Oversampling Online Bagging*, see Figure 2. In particular, it concerns *Neighbourhood Oversampling Online Bagging (NOOB).*

The merge and move class decompositions are not so demanding for specialized online bagging ensembles, see Figure 3. However, the split of one minority class in several sub-clusters (sub-concepts) has shown to be more difficult, and there the role of new classifiers is greater than from the merging and moving case.

Definitely, the most interesting and visible results have been obtained for the static presence or drifts of unsafe types of minority class examples - see Figures for rare examples in the main paper [10]. It is also possible to analyze it more deeply by the averaged results of G-mean presented in Table 1.

Summarizing these results both NOOB and NUOB are useful to improve the recovery of classifiers after the drift from safe to unsafe proportions of the minority class. Their improvements are more visible in the case of rare examples.

This is confirmed by the other analysis presented in the main paper, i.e. the global comparison of different ensembles over multiple categories of data streams with the non-parametric Friedman test - according to it NUOB is usually the best choice.

3.3 Data streams with pairs of factors

In the case of a pair of given factors and drifts, the usefulness of the newly proposed neighbourhood bagging ensembles is even more visible than for single factors. Now, the combination of strong imbalance or class splits with unsafe types of minority examples is better solved by NUOB and NOOB than by UOB or OOB. It can be observed in a representative case - see Figure 4.

Table 1. The impact of borderline and rare type of minority examples in static, balanced streams on *G-mean* values

Configuration	N	OB	UOB	OOB	NUOB	NOOB
Safe stream	0%	0.992	0.992	0.993	0.992	0.991
Borderline[N]	20%	0.978	0.978	0.978	0.979	0.969
	40%	0.974	0.974	0.974	0.976	0.965
	60%	0.972	0.971	0.972	0.973	0.963
	80%	0.971	0.970	0.971	0.972	0.961
	100%	0.969	0.968	0.969	0.970	0.959
Rare[N]	20%	0.935	0.935	0.935	0.935	0.934
	40%	0.865	0.865	0.865	0.865	0.866
	60%	0.779	0.779	0.779	0.779	0.800
	80%	0.684	0.680	0.690	0.677	0.752
	100%	0.677	0.668	0.689	0.661	0.692

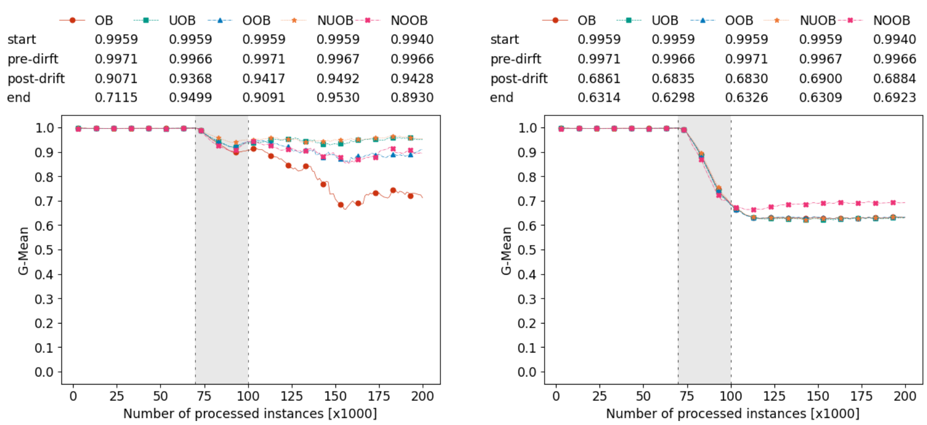

Fig. 4. Plots showing *G-mean* measure of bagging variants reacting to two kind of drift *80% borderline minority examples and imbalanced ratio changing from 50% to 1%* (the left-hand figure) and *60% rare minority examples and minority class split into 5 sub-concepts* (the right-hand figure)

Furthermore, we prepared a special chart presenting differences between considered ensemble classifiers over the most difficult combinations of pairs of data factors and drifts (see Figure 5). One can notice that NUOB and NOOB are better than OOB and UOB concerning G-mean, and the difference is even more visible for the Recall of the minority class. As expected (due to the tuned value of Ψ parameter) they are particularly efficient with handling different pairs with drifts to rare examples inside the minority class. NUOB is usually better for the rare examples' pairs, while NOOB wins for borderline examples' pairs.

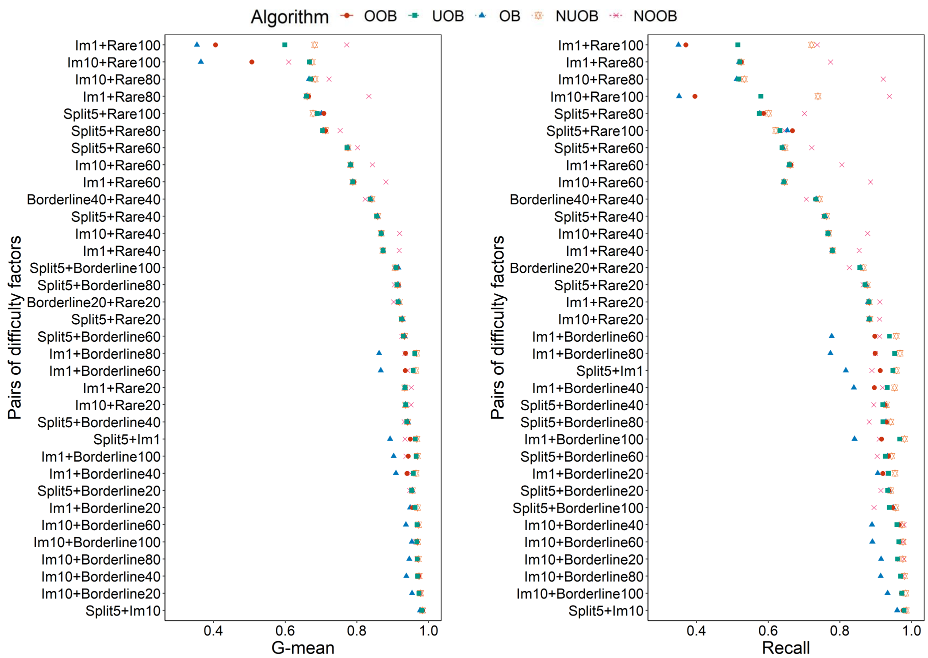

Fig. 5. Comparison of the effect of pairs of factors classifier G-mean (the left-hand figure) and Recall (the right-hand figure). Scenarios ranked according to the median performance of all classifiers on a given scenario

3.4 Complex scenarios

Let us recall that complex scenarios, where 3 or 4 data factors or drifts occur together in the stream, were the most difficult in earlier experimental studies [4]. The earlier OOB and UOB bagging ensembles were not able to sufficiently recover from such changes in the streams. Again, the newly proposed *Neighbourhood Oversampling Online Bagging* and *Neighbourhood Undersampling Online Bagging* were able to (at least partly) help in such scenarios. In the main paper, we presented representative figures showing their usefulness. Here, we refer the reader to the next specialized plot (see Figure 6), where again we ranked the most difficult complex scenarios. The reader can notice that, in particular, NUOB proposed ensemble can work with three elements (split, imbalance, and rare examples). NOOB is usually the second, but for some triples, it is also a winner.

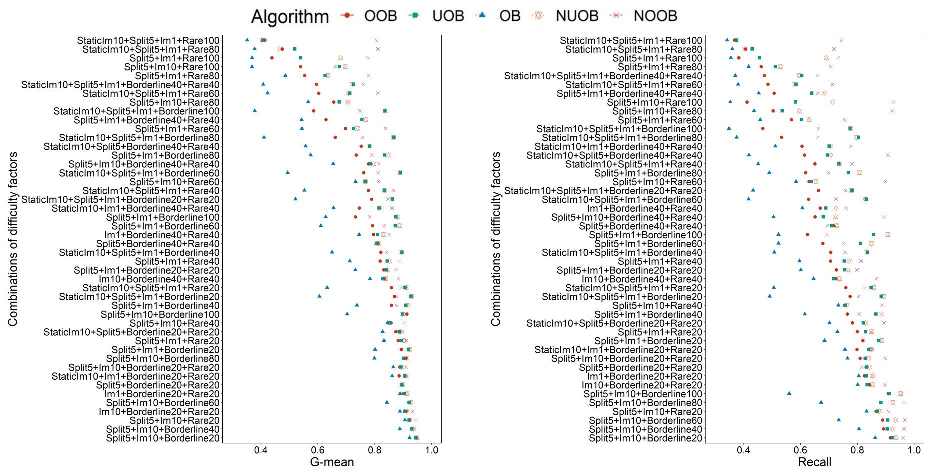

Fig. 6. Comparison of the effect of complex scenarios on G-mean (the left-hand figure) and Recall (the right-hand figure) of the analyzed classifiers. Scenarios ranked according to the median performance of all classifiers on a given scenario

We recall that in the main paper, we summarized the results of global statistical analysis – and presented average ranks in the Friedman test for multiple complex scenarios. This analysis shows that NUOB and NOOB are working the best for such multiple factors.

3.5 Hybrid Neighbourhood Online Bagging

Final experimental results present a very good performance of the *Hybrid Neighbourhood Online Bagging (HNOB).*

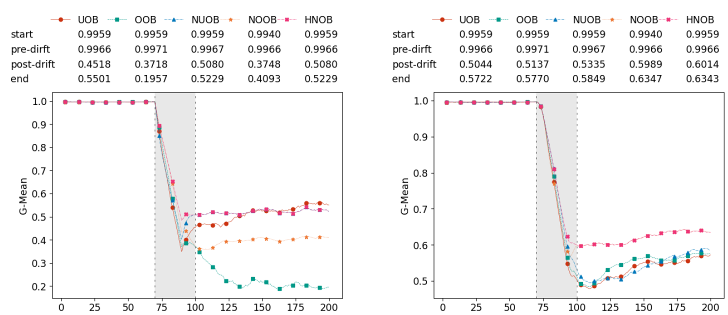

Fig. 7. Plots showing *G-mean* measure of bagging variants reacting to two kind of drift *100% rare minority examples and imbalanced ratio changing from 50% to 10%* and *80% rare minority examples and minority class split into 5 sub-concepts*

Figure 7 shows the best performance of the hybrid version – HNOB for pairs of factors, while figures showing it for the multiple, complex scenarios are given in the main paper. The supremacy of HNOB is also visible in the summary plot (see Figure 8) and inside the results of the Friedman test – presented in Table 2. It can be noticed that for all examined different pairs of factors the proposed HNOB is either the best classifier or one of the best classifiers (followed closely by the NUOB algorithm).

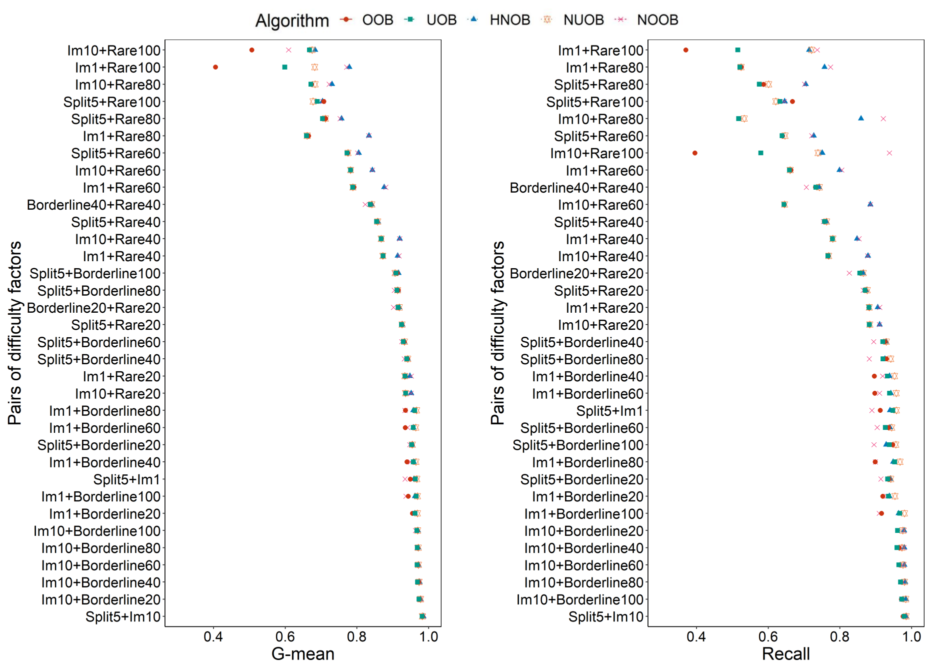

Fig. 8. Comparison of the effect of pairs of factors classifier G-mean (the left-hand figure) and Recall (the right-hand figure). Scenarios ranked according to the median performance of all classifiers on a given scenario

4 Discussion and Final Remarks

In this appendix, we extended information on our proposal of new online bagging ensembles for complex, imbalanced data streams (see Section 2). As we showed they are particularly designed to better tackle the difficulty of unsafe types of minority examples - see an extended discussion (comparing to the main paper [10]) presented in Section 2.

The additional results (Section 3) demonstrate that both introduced online bagging generalizations were most effective for scenarios with drifts in the types

Table 2. Average ranks in the Friedman test for different pairs of factors in streams

Data stream set	Metric	HNOB	UOB	OOB	NUOB	NOOB	CD
Imbalance + Move	G-mean	**2.17**	**2.83**	**3.75**	**3.17**	**3.08**	1.82
Imbalance + Merge		**2.08**	**2.17**	4.08	**2.75**	3.92	1.82
Imbalance + Split		**2.17**	**2.67**	4.50	**2.44**	**3.22**	1.47
Imbalance + Borderline		**2.17**	**2.92**	4.38	**2.08**	3.45	0.97
Imbalance + Rare		**1.63**	3.77	4.08	3.88	**1.65**	0.97
Split + Borderline		**2.10**	**2.90**	4.00	**2.38**	3.62	0.87
Split + Rare		**1.56**	3.64	4.14	3.56	**2.10**	0.87
Imbalance + Move	Recall	**2.50**	**2.50**	4.83	**1.58**	3.58	1.82
Imbalance + Merge		**2.42**	**2.32**	4.83	**1.67**	3.75	1.82
Imbalance + Split		**2.39**	**2.56**	4.83	**1.56**	3.67	1.47
Imbalance + Borderline		**2.20**	3.10	4.88	**1.75**	3.08	0.97
Imbalance + Rare		**1.95**	3.58	4.83	2.90	**1.75**	0.97
Split + Borderline		**2.46**	3.00	4.48	**2.04**	3.02	0.87
Split + Rare		**2.14**	3.28	4.56	**2.76**	**2.26**	0.87

of examples and also their combinations with other factors. In general Neighbourhood Undersampling Online Bagging was slightly better than Neighbourhood Overrsampling Online Bagging in many scenarios of the studied synthetic datastreams.

However, the final experimental results lead us to recommend using Hybrid Neighbourhood Online Bagging, which dynamically uses the currently superior model of a pair of parallel-trained online bagging ensembles.

References

1. Aguiar, G., Krawczyk, B., Cano, A.: A survey on learning from imbalanced data streams: taxonomy, challenges, empirical study, and reproducible experimental framework. Machine learning **113**, 1–79 (2023). https://doi.org/10.1007/s10994-023-06353-6
2. Błaszczyński, J., Stefanowski, J.: Neighbourhood sampling in bagging for imbalanced data. Neurocomputing **150**, 529–542 (10 2015). https://doi.org/10.1016/j.neucom.2014.07.064
3. Błaszczyński, J., Stefanowski, J., Idkowiak, Ł.: Extending bagging for imbalanced data. In: Proceedings of the 8th international conference on computer recognition systems CORES 2013. pp. 269–278. Springer (2013)

4. Brzeziński, D., Minku, L.L., Pewiński, T., Stefanowski, J., Szumaczuk, A.: The impact of data difficulty factors on classification of imbalanced and concept drifting data streams. Knowledge and Information Systems **63**(6), 1429–1469 (2021)
5. Brzezinski, D., Stefanowski, J., Susmaga, R., Szczęch, I.: Visual-based analysis of classification measures and their properties for class imbalanced problems. Information Sciences **462**, 242–261 (2018)
6. Du, H., Zhang, Y., Gang, K., Zhang, L., Chen, Y.C.: Online ensemble learning algorithm for imbalanced data stream. Applied Soft Computing **107**, 107378 (2021). https://doi.org/https://doi.org/10.1016/j.asoc.2021.107378
7. Lipska, A., Stefanowski, J.: The influence of multiple classes on learning from imbalanced data streams. In: Third International Workshop on Learning with Imbalanced Domains: Theory and Applications. ECMLPKDD 2022. pp. 187 – 198. PMLR (2022)
8. Napierała, K., Stefanowski, J.: Types of Minority Class Examples and Their Influence on Learning Classifiers from Imbalanced Data. Journal of Intelligent Information Systems **46**, 563–597 (2016)
9. Oza, N.C., Russell, S.J.: Experimental comparisons of online and batch versions of bagging and boosting. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001. pp. 359–364 (2001)
10. Przybył, B., Stefanowski, J.: Improving online bagging for complex imbalanced data streams. In: Proceedings of the NFMCP workshops at ECMLPKDD 2024, full version will appear in Springer CCIS series. available also as a pre-print arXiv:2410.03519 (2024)
11. Wang, B., Pineau, J.: Online bagging and boosting for imbalanced data streams. IEEE Transactions on Knowledge and Data Engineering **28**(12), 3353–3366 (2016). https://doi.org/10.1109/TKDE.2016.2609424
12. Wang, S., Minku, L., Yao, X.: Dealing with multiple classes in online class imbalance learning. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16). pp. 2118—2124 (2016)
13. Wang, S., Minku, L.L., Yao, X.: Resampling-based ensemble methods for online class imbalance learning. IEEE Trans. Knowl. Data Eng. **27**(5), 1356–1368 (2015)