Preprocessed Dynamic Classifier Ensemble Selection for Highly Imbalanced Drifted Data Streams

Paweł Zyblewski[0000−0002−4224−6709]1, Robert Sabourin[2],
Michał Woźniak[0000−0003−0146−4205]1

[1] *Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland*
[2] *École de Technologie Supérieure, Montreal, Canada*

Abstract

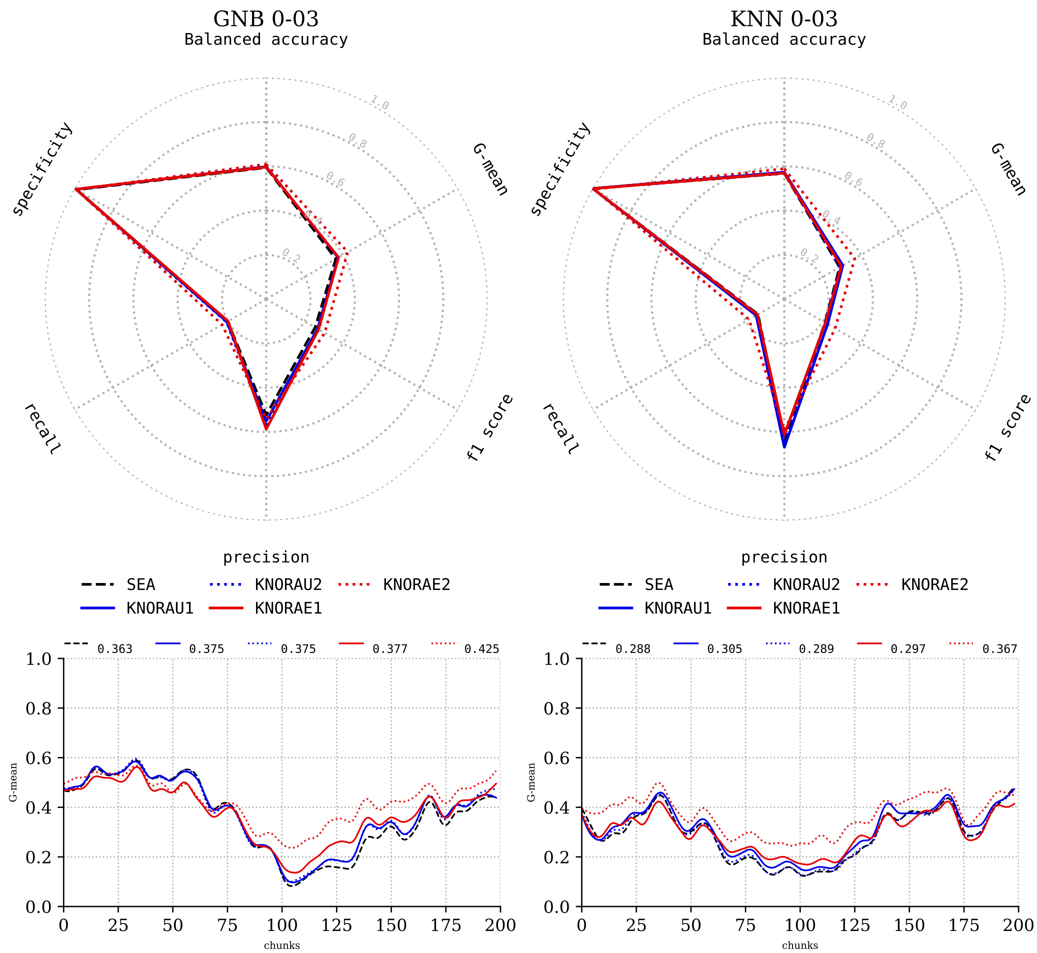
This work aims to connect two rarely combined research directions, i.e., non-stationary data stream classification and data analysis with skewed class distributions. We propose a novel framework employing stratified bagging for training base classifiers to integrate data preprocessing and dynamic ensemble selection methods for imbalanced data stream classification. The proposed approach has been evaluated based on computer experiments carried out on 135 artificially generated data streams with various imbalance ratios, label noise levels, and types of concept drift as well as on two selected real streams. Four preprocessing techniques and two dynamic selection methods, used on both bagging classifiers and base estimators levels, were considered. Experimentation results showed that, for highly imbalanced data streams, dynamic ensemble selection coupled with data preprocessing could outperform online and chunk-based *state-of-art* methods.

Key words: Dynamic ensemble selection, Imbalanced data, Data stream, Data preprocessing, Concept drift

1. Experimental evaluation

1.1. Experiment 1 – Dynamic selection level

In Figure 1 we see that in the case of KNN (Figure 1b) KNORAE1 slightly deteriorates aggregate metrics and performs better than SEA only when the concept drift occurs. The best results are achieved by KNORAE2, which



(a) GNB (b) KNN

Figure 1: Experiment 1 results for *Gaussian Naïve Bayes* and k-Nearest Neighbors classifiers.

improves aggregate metrics at the expense of *precision*, and is by far the best of the tested methods when it comes to dealing with concept drift.

Based on the results obtained, the following methods of *Dynamic Ensemble Selection* were selected for further experiments:

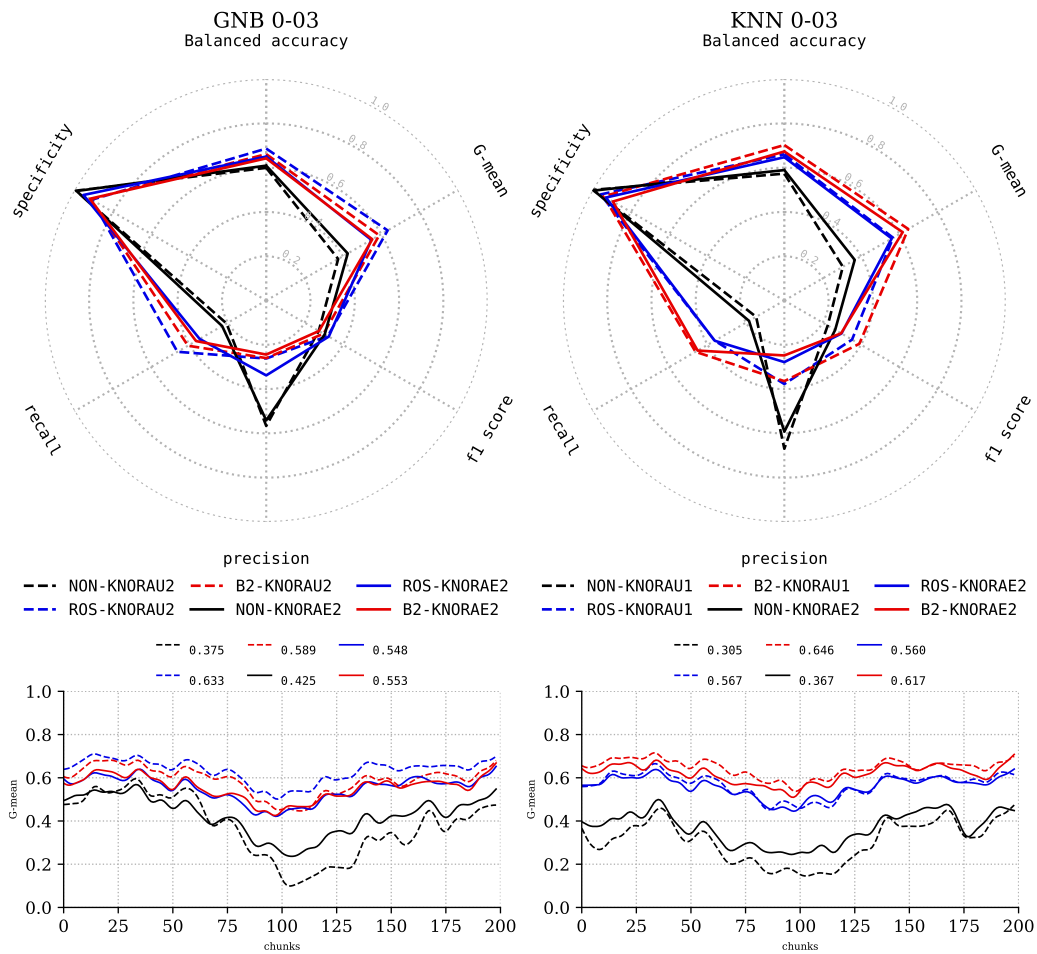
- GNB - KNORAU and KNORAE on the base classifiers level (KNORAU2, KNORAE2).
- KNN - KNORAU on the bagging classifiers level (KNORAU1) and KNORAE

on the base classifiers level (KNORAE2).

1.2. *Experiment 2 – Pairing* DES *with preprocessing techniques*

The following are the results of combining selected methods of *Dynamic Ensemble Selection* with preprocessing techniques. The experiment was divided into parts related to *oversampling* and *undersampling*.

Oversampling



(a) GNB (b) KNN

Figure 2: Experiment 2.1 results for *Gaussian Naïve Bayes* and *k*-Nearest Neighbors classifiers.

Table 1: *G-mean*-based performance metrics regarding sudden drift for Experiment 2.1.

Performance metric	NON-U	ROS-U	B2-U	NON-E	ROS-E	B2-E
				GNB		
performance loss	0.833	0.654	0.526	0.789	0.626	0.524
restoration time	0.026	0.012	0.010	0.023	0.010	0.009
				KNN		
performance loss	1.000	0.476	0.466	1.000	0.664	0.462
restoration time	0.021	0.014	0.013	0.010	0.012	0.009

Based on Figure 2, we can conclude that in the case where KNN is the base classifier (Figure 2b), DES methods achieve the best results when combined with *Borderline2*-SMOTE. An increase in aggregated metrics and *recall* can be observed at the expense of *precision* and a small decrease in *specificity*. The use of *Random Oversampling* also leads to a significant improvement over pure *Dynamic Ensemble Selection*, but in this case, the classification algorithm is not as good at detecting the minority class as when coupled with *Borderline2*-SMOTE.

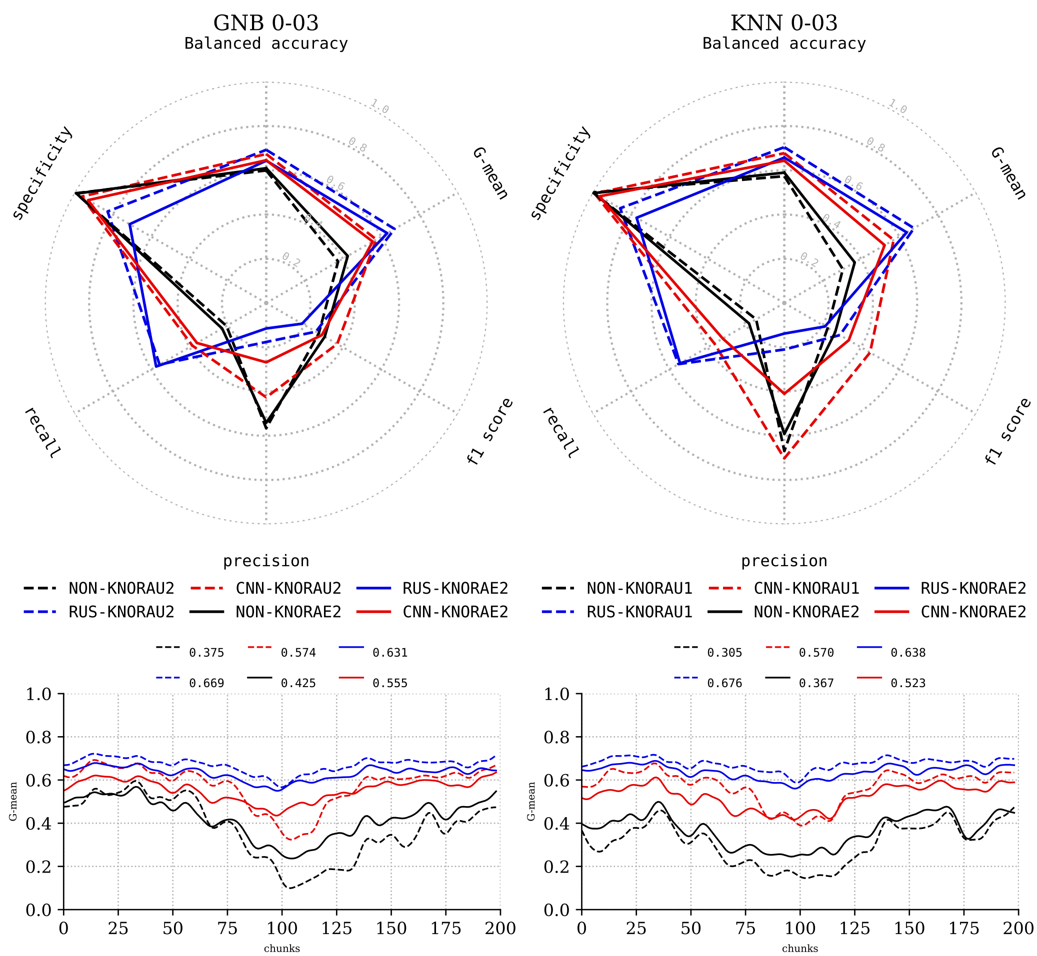
Table 1 contains *performance loss* and *restoration time* values in terms of *G-mean* averaged over all runs, referring to *sudden* concept drift. In the case of GNB and KNN, generally methods paired with Borderline2-SMOTE achieve the smallest *performance loss* and *restoration time* values. This may be due to the generation of artificial minority samples near the decision boundary.

It should be noted that better performance in terms of *performance loss* and *restoration time* does not necessarily mean better classification performance.

Undersampling

Table 2: *G-mean*-based performance metrics regarding sudden drift for Experiment 2.2.

Performance metric	NON-U	RUS-U	CNN-U	NON-E	RUS-E	CNN-E
			GNB			
performance loss	0.833	0.549	0.607	0.789	0.421	0.726
restoration time	0.026	0.017	0.021	0.023	0.010	0.010
			KNN			
performance loss	1.000	0.535	0.710	1.000	0.508	0.546
restoration time	0.021	0.020	0.014	0.010	0.008	0.010



(a) GNB (b) HT

Figure 3: Experiment 2.2 results for *Gaussian Naïve Bayes* and k-Nearest Neighbors classifiers.

For KNN (Figure 3b), the use of both *undersampling* methods leads to a significant improvement in the results achieved in terms of aggregated metrics and *recall*. *Random Undersampling* performs slightly better in terms of *G-mean* and *recall*, but it is worse than CNN when it comes to *F1 score* and *specificity*, it also achieves much worse *precision* (especially in combination with KNORAE2). On the presented run, we can see that the improvement of the results takes place not only during the concept drift but along the entire length of the processed data streams.

In Table 2 we can see *performance loss* and *restoration time* values for undersampling methods. In the case of GNB and KNN, RUS achieves the best values of these metrics. In the case of SVM classifier, CNN allows DES techniques achieving the lowest performance loss and restoration time, but simultaneously, it leads to the worst classification performance.

Acknowledgements

This work was supported by the Polish National Science Centre under the grant No. 2017/27/B/ST6/01325.