Dynamic Classifier Selection for Data with Skewed Class Distribution Using Imbalance Ratio and Euclidean Distance

Paweł Zyblewski(✉) and Michał Woźniak

Department of Systems and Computer Networks, Faculty of Electronics,
Wrocław University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{pawel.zyblewski,michal.wozniak}@pwr.edu.pl

Abstract. Imbalanced data analysis remains one of the critical challenges in machine learning. This work aims to adapt the concept of *Dynamic Classifier Selection* (DCS) to the pattern classification task with the skewed class distribution. Two methods, using the similarity (distance) to the reference instances and class imbalance ratio to select the most confident classifier for a given observation, have been proposed. Both approaches come in two modes, one based on the k-Nearest Oracles (KNORA) and the other also considering those cases where the classifier makes a mistake. The proposed methods were evaluated based on computer experiments carried out on 41 datasets with a high imbalance ratio. The obtained results and statistical analysis confirm the usefulness of the proposed solutions.

Keywords: Classifier ensemble · Dynamic Classifier Selection · Imbalanced data

1 Introduction

Traditional machine learning algorithms assume that the number of instances belonging to problem classes is relatively similar. However, it is worth noting that in many real problems the size of one class (*majority class*) may significantly exceed the size of the second one (*minority class*). This makes the algorithms biased towards the majority class, although the correct recognition of less common class is often more important. This research trend is known as learning from imbalanced data [8] and it is still widely discussed in scientific works.

There are three main approaches to dealing with the imbalanced data classification:

- *Data-level methods* focusing on modifying the training set in such a way that it becomes suitable for classic learning algorithms (e.g., *oversampling* and *undersampling*).

V. V. Krzhizhanovskaya et al. (Eds.): ICCS 2020, LNCS 12140, pp. 59–73, 2020.
https://doi.org/10.1007/978-3-030-50423-6_5

- *Algorithm-level methods* that modify existing classification algorithms to offset their bias towards the majority class.
- *Hybrid methods* combining the strengths of the previously mentioned approaches.

Many works on imbalanced data classification employ classifier ensembles [16]. One of the more promising directions is the *Dynamic Ensemble Selection* (DES) [5]. Dynamic selection (DS) methods select a single classifier or an ensemble (from an available classifier pool) to predict the decision for each unknown query. This is based on the assumption that each of the base classifiers is an expert in a different region of the feature space. The classification of each unknown sample by DES involves three steps:

- Definition of the region of competence; that is, how to define the local region surrounding the unknown sample, in which the competence level of the base models is estimated. This local region of competence is found in the dynamic selection dataset (DSEL), which is usually part of the training set.
- Defining the selection criterion later used to assess the competence of the base classifiers in the local region of competence (e.g., accuracy or diversity).
- Determination of the selection mechanism deciding whether we choose a single classifier or an ensemble.

Previous work related to the imbalanced data classification using classifier ensembles and DES involves various approaches. Ksieniewicz in [9] proposed an *Undersampled Majority Class Ensemble* (UMCE) employing different combination methods and pruning, based on a k-fold division of the majority class to divide an imbalanced problem into many balanced ones. Chen et al. [4] presented the *Dynamic Ensemble Selection Decision-making* (DESD) algorithm to select the most appropriate classifiers using a weighting mechanism to highlight the base models that are better suited for recognizing the minority class. Zyblewski et al. in [17] proposed the *Minority Driven Ensemble* (MDE) for highly imbalanced data streams classification and Roy et al. in [14] combined preprocessing with dynamic ensemble selection to classify both binary and multiclass stationary imbalanced datasets.

The main contributions of this work are as follows:

- The proposition of the new dynamic selection methods adapted for the classification of highly imbalanced data.
- Experimental evaluation of the proposed algorithms based on a high number of diverse benchmark datasets and a detailed comparison with the *state-of-art* approaches.

2 Dynamic Ensemble Selection Based on Imbalance Ratio and Euclidean Distance

This paper proposes two algorithms for dynamic classifier selection for the imbalanced data classification problem. These are respectively the Dynamic Ensemble

Selection using Euclidean distance (DESE) and the Dynamic Ensemble Selection using Imbalance Ratio and Euclidean distance (DESIRE).

The generation of the classifier pool is based on the *Bagging* approach [2], and more specifically on the *Stratified Bagging*, in which the samples are drawn with replacement from the minority and majority class separately in such a way that each bootstrap maintains the original training set class proportion. This is necessary due to the high imbalance, which in the case of standard bagging can lead to the generation of training sets containing only the majority class.

Both proposed methods are derived in part from algorithms based on local oracles, and more specifically on KNORA-U [7], which gives base classifiers weights based on the number of correctly classified instances in the local region of competence and then combines them by weighted majority voting. The computational cost in this type of method is mainly related to the size of the classifier pool and the DSEL size, as the k-nearest neighbors technique is used to define local competence regions, which can be costly for large datasets. Instead of hard voting, DESE and DESIRE are based on the probabilities returned by the base models and they calculate weights for each classifier for both the minority and majority classes separately.

Proposed methods come in two variants: *Correct* (denoted as C), where weights are modified only in the case of correct classification, and *All* (denoted as A), where, in addition to correct decisions, weights are also affected by incorrect ones. The exact way of weights calculation is presented in Algorithm 1.

For each instance, the proposed algorithms perform the following steps:

- In step 2, the k-nearest neighbors of a given instance are found in DSEL, which form the local region of competence LRC.
- In step 4, each classifier Ψ_j from the pool classifies all samples belonging to LRC.
- In steps 5–13, the classifier weights are modified separately for the minority and majority class, starting from the value of 0. The *All* variant uses all four conditions, while the *Correct* variant is based only on the conditions in lines 6 and 8. In the case of DESE, the modifications are based on the Euclidean distance between the classified sample and its neighbor from the local competence region, and in the case of DESIRE, the Euclidean distance is additionally scaled by a percentage of the minority or majority class in such a way that more emphasis is placed on the minority class.

Finally, the weights obtained from DESE or DESIRE are normalized to the $[0, 1]$ range and multiplied by the ensemble support matrix. The combination is carried out according to the maximum rule [6], which chooses the classifier that is most confident of itself. The choice of this combination rule was dictated by a small number of instances in the datasets, which significantly reduces the risk of base classifiers overfitting.

Algorithm 1: Pseudocode of the proposed weight calculation methods assuming that the minority class is positive and the majority class is negative.

```
Input:
    $E$, classifier pool,
    $D_t$, test dataset,
    $DSEL$, Dynamic Selection Dataset,
    $k$, number of nearest neighbors,
    $min, maj$, respectively the percentage of minority and majority classes in
    the training set,
    $W \leftarrow \varnothing$, empty weights array of shape (n_classifiers, n_samples, 2).
Output:
    $W$, weights array of shape (n_classifiers, n_samples, 2).
for each sample $x_i$ in $D_t$ do
    $LRC \leftarrow$ the $k$ nearest neighbors of $x_i$ in $DSEL$
    for each Classifier $\Psi_j$ in $E$ do
        $Predict \leftarrow predict(LRC, \Psi_j)$
        for each $neighbor$ in $len(LRC)$ do
            if $Predict[neighbor]$ = True negative then
                $W[j,i,0] += \begin{cases} ED[x_i, neighbor] & \text{for DESE} \\ ED[x_i, neighbor] * min & \text{for DESIRE} \end{cases}$
            else if $Predict[neighbor]$ = True positive then                      Correct
                $W[j,i,1] += \begin{cases} ED[x_i, neighbor] & \text{for DESE} \\ ED[x_i, neighbor] * maj & \text{for DESIRE} \end{cases}$
            else if $Predict[neighbor]$ = False negative then                     All
                $W[j,i,1] -= \begin{cases} ED[x_i, neighbor] & \text{for DESE} \\ ED[x_i, neighbor] * min & \text{for DESIRE} \end{cases}$
            else if $Predict[neighbor]$ = False positive then
                $W[j,i,0] -= \begin{cases} ED[x_i, neighbor] & \text{for DESE} \\ ED[x_i, neighbor] * maj & \text{for DESIRE} \end{cases}$
        end for
    end for
end for
```

3 Experimental Evaluation

This section presents the details of the experimental study, the datasets used and the results that the proposed approaches have achieved compared to the *state-of-art* methods.

3.1 Experimental Set-Up

The main goal of the following experiments was to compare the performance of proposed dynamic selection methods, designed specifically for the task of imbalanced data classification, with the *state-of-art* ensemble methods paired with preprocessing. The evaluation in each of the experiments is based on 5 metrics commonly used to assess the quality of classification for imbalanced problems. These are *F1 score* [15], *precision* and *recall* [13], *G-mean* [11] and *balanced* *accuracy score* (BAC) [3] according to the *stream-learn* [10] implementation. All experiments have been implanted in *Python* and can be repeated using the code on *Github*[1].

As the base models three popular classifiers, according to the *scikit-learn* [12] implementation, were selected, i.e. *Gaussian Naive Bayes* (GNB), *Classification and Regression Trees* (CART) and *k-Nearest Neighbors* classifier (KNN). The fixed size of the classifier pool has been determined successively as 5, 15, 30 and 50 base models. The evaluation was carried out using 10 times repeated 5-fold cross-validation. Due to the small number of instances in the datasets, DSEL is defined as the entire training set.

The experiments were carried out on 41 datasets from the KEEL repository [1], which contain binary problems created through various combinations of class merging. All datasets have a high imbalance ratio of at least 9. Problems characteristics are presented in Table 1.

Table 1. Datasets characteristics.

Dataset	Instances	Features	IR	Dataset	Instances	Features	IR
ecoli-0-1_vs_2-3-5	244	7	9	glass2	214	9	12
ecoli-0-1_vs_5	240	6	11	glass4	214	9	15
ecoli-0-1-3-7_vs_2-6	281	7	39	glass5	214	9	23
ecoli-0-1-4-6_vs_5	280	6	13	led7digit-0-2-4-5-6-7-8-9_vs_1	443	7	11
ecoli-0-1-4-7_vs_2-3-5-6	336	7	11	page-blocks-1-3_vs_4	472	10	16
ecoli-0-1-4-7_vs_5-6	332	6	12	shuttle-c0-vs-c4	1829	9	14
ecoli-0-2-3-4_vs_5	202	7	9	shuttle-c2-vs-c4	129	9	20
ecoli-0-2-6-7_vs_3-5	224	7	9	vowel0	988	13	10
ecoli-0-3-4_vs_5	200	7	9	yeast-0-2-5-6_vs_3-7-8-9	1004	8	9
ecoli-0-3-4-6_vs_5	205	7	9	yeast-0-2-5-7-9_vs_3-6-8	1004	8	9
ecoli-0-3-4-7_vs_5-6	257	7	9	yeast-0-3-5-9_vs_7-8	506	8	9
ecoli-0-4-6_vs_5	203	6	9	yeast-0-5-6-7-9_vs_4	528	8	9
ecoli-0-6-7_vs_3-5	222	7	9	yeast-1_vs_7	459	7	14
ecoli-0-6-7_vs_5	220	6	10	yeast-1-2-8-9_vs_7	947	8	31
ecoli4	336	7	16	yeast-1-4-5-8_vs_7	693	8	22
glass-0-1-4-6_vs_2	205	9	11	yeast-2_vs_4	514	8	9
glass-0-1-5_vs_2	172	9	9	yeast-2_vs_8	482	8	23
glass-0-1-6_vs_2	192	9	10	yeast4	1484	8	28
glass-0-1-6_vs_5	184	9	19	yeast5	1484	8	33
glass-0-4_vs_5	92	9	9	yeast6	1484	8	41
glass-0-6_vs_5	108	9	11				

Subsections 3.2 and 3.3 present the results of experiments comparing the presented methods, DESE in experiment 1 and DESIRE in experiment 2, with *state-of-art* ensemble algorithms used for the imbalanced data classification.

Both proposed and reference methods occur in versions with preprocessing (in the form of *random oversampling*) and without it, the use of oversampling is denoted by the letter O found before the acronym of the method. As a reference method, a single classifier, as well as stratified bagging and dynamic selection in the form of the KNORA-U algorithm were selected.

[1] https://github.com/w4k2/iccs20-desire.

The radar diagrams show the average global ranks achieved by each of the tested algorithms in terms of each of the 5 evaluation metrics, while the tables show the results of the Wilcoxon signed-rank ($p = 0.05$) statistical test for a pool size of 5 base classifiers. The numbers under the average rank of each method indicate the algorithms which are statistically significantly worse than the one in question. The complete results for each of the 41 datasets and the full statistical analysis can be found on the *Github*[2].

3.2 Experiment 1 – Euclidean Distance-Based Approach

In Fig. 1 we can see how the average ranks for DESE and reference methods changed in terms of different metrics depending on the ensemble size. We can see that the proposed methods (especially ODESE-C) for 5 base models achieve higher rankings in terms of each metric with an exception of *recall*. While the single classifier and bagging are preferring *recall*, ODESE-C and DESE-C prefer *precision*. As the number of base classifiers increases, BAC and *G-mean*-based rankings deteriorate to KNORA-U level, while the *F1 score* remains high due to high *precision*.

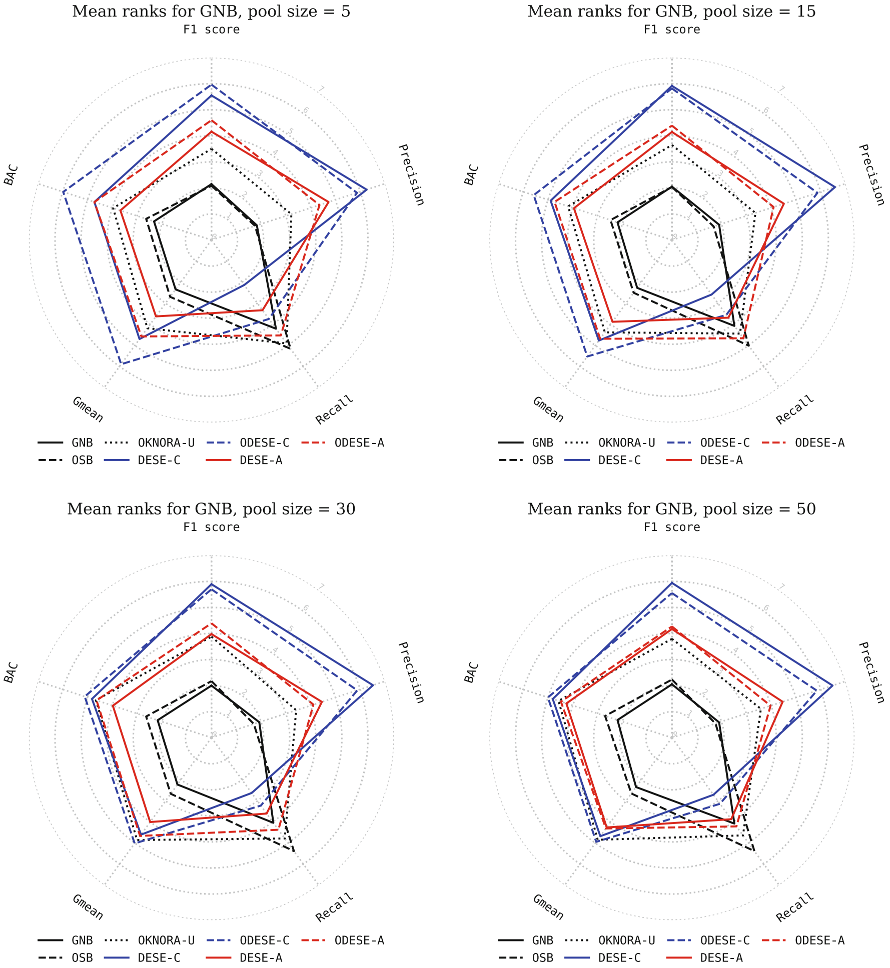
Table 2 presents the results of the statistical analysis, which shows that the ODESE-C method performs statistically significantly better than all reference methods in terms of each metric except for *recall*.

When the base classifier is CART, as seen in Fig. 2, for the smallest pool, DESE-C (both without and with oversampling) achieves higher ranks than the reference methods in terms of each of the five metrics. Along with the increase in the number of classifiers, we can observe that while OKNORA-U and OSB stand out in terms of *precision*, ODESE-C performs better in terms of other metrics, and ODESE-A, despite the low *F1 score* and *precision*, achieves the highest average ranks in terms of BAC, *G-mean* and *recall*. Table 3 confirms that for the five base classifiers, ODESE-C is statistically significantly better than all reference methods, while ODESE-A performs statistically significantly better than ODESE-C in terms of *recall*, *G-mean* and BAC.

Table 2. Statistical tests on mean ranks for GNB with pool size = 5.

	GNB **(1)**	OSB **(2)**	OKNORA-U **(3)**	DESE-C **(4)**	ODESE-C **(5)**	DESE-A **(6)**	ODESE-A **(7)**
F1 score	2.146	2.085	3.500	5.549	5.963	4.159	4.598
	–	–	1,2	1,2,3,6,7	1,2,3,6,7	1,2,3	1,2,3
precision	1.829	1.756	3.220	6.256	5.866	4.720	4.354
	–	–	1,2	*all*	1,2,3,6,7	1,2,3	1,2,3
recall	4.207	5.159	4.902	2.134	3.744	3.329	4.524
	4	4,5,6	4,5,6	–	4	4	4,5,6
G-mean	2.341	2.695	4.183	4.695	5.890	3.622	4.573
	–	–	1,2	1,2,6	*all*	1	1,2,6
BAC	2.317	2.634	3.963	4.720	5.976	3.671	4.720
	–	–	1,2	1,2,6	*all*	1,2	1,2,6

[2] https://github.com/w4k2/iccs20-desire/tree/master/article_results.



Mean ranks for GNB, pool size = 5

Mean ranks for GNB, pool size = 15

Mean ranks for GNB, pool size = 30

Mean ranks for GNB, pool size = 50

Fig. 1. Mean ranks for GNB classifier.

Table 3. Statistical tests on mean ranks for CART with pool size = 5.

	CART **(1)**	**OSB** **(2)**	**OKNORA-U** **(3)**	**DESE-C** **(4)**	**ODESE-C** **(5)**	**DESE-A** **(6)**	**ODESE-A** **(7)**
F1 score	2.683 –	2.841 –	2.988 –	5.329 1,2,3,6,7	5.561 1,2,3,6,7	4.256 1,2,3	4.341 1,2,3
precision	2.634 –	3.976 1	4.195 1,6,7	5.695 *all*	5.134 1,2,3,6,7	3.195 –	3.171 –
recall	3.293 2,3	2.622 –	2.695 –	3.890 2,3	4.463 1,2,3,4	5.366 1,2,3,4,5	5.671 1,2,3,4,5
G-mean	3.098 –	2.671 –	2.817 –	4.061 2,3	4.634 1,2,3,4	5.232 1,2,3,4	5.488 1,2,3,4,5
BAC	3.098 –	2.585 –	2.732 –	4.280 1,2,3	4.829 1,2,3,4	5.085 1,2,3,4	5.390 1,2,3,4,5

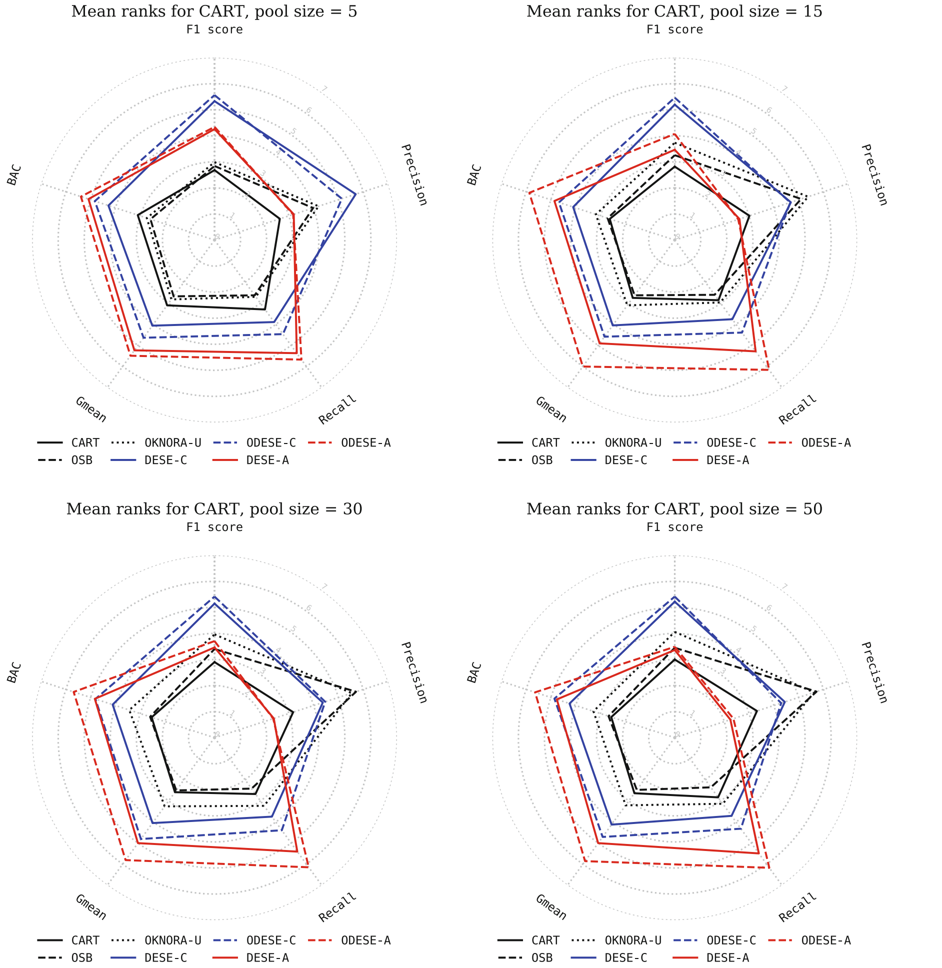


Fig. 2. Mean ranks for CART classifier.

Table 4. Statistical tests on mean ranks for KNN with pool size = 5.

	KNN (1)	OSB (2)	OKNORA-U (3)	DESE-C (4)	ODESE-C (5)	DESE-A (6)	ODESE-A (7)
F1 score	3.585 –	4.305 3	3.476 –	4.549 1,6	4.390 –	3.744 –	3.951 –
precision	5.317 3,5,6,7	3.963 3,7	3.049 –	4.976 2,3,5,6,7	3.659 –	3.878 7	3.159 –
recall	1.427 –	5.232 1,4,6	5.366 1,4,6	2.463 1	4.939 1,4,6	3.305 1,4	5.268 1,4,6
G-mean	1.537 –	5.061 1,4,6	4.866 1,4,6	2.720 1	5.110 1,4,6	3.427 1,4	5.280 1,4,6
BAC	1.659 –	5.012 1,4,6	4.841 1,4,6	2.780 1	5.024 1,4,6	3.415 1,4	5.268 1,4,6

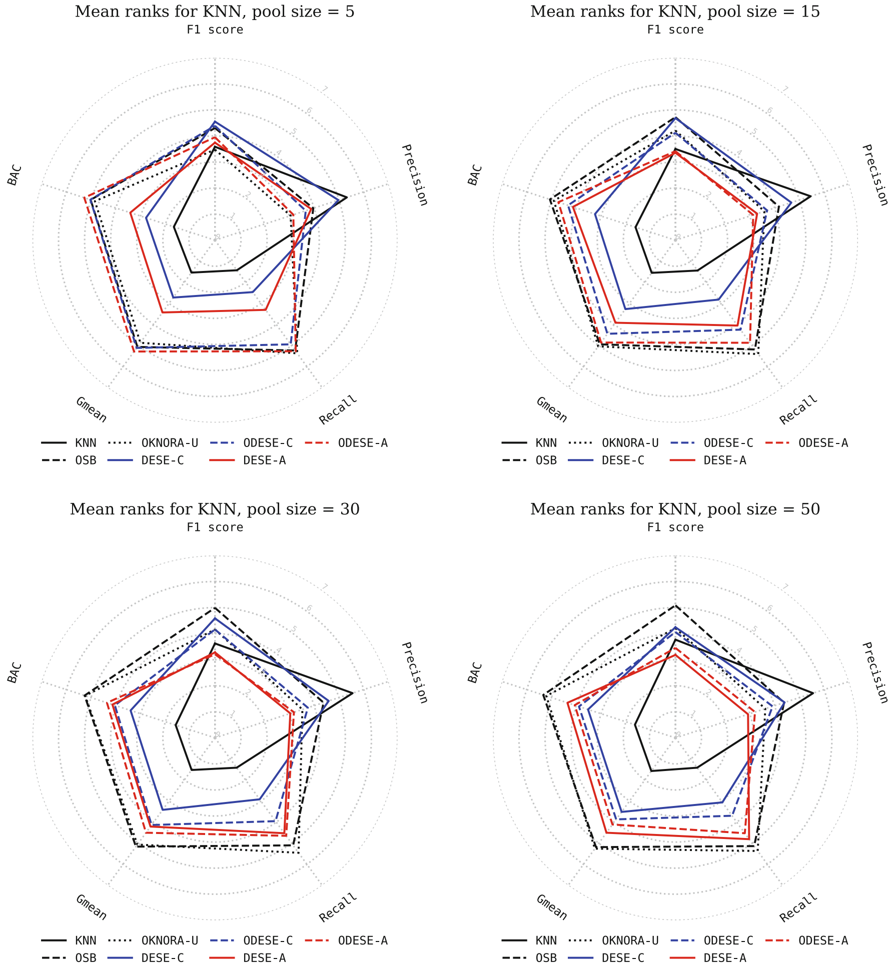


Fig. 3. Mean ranks for KNN classifier.

In Fig. 3 and Table 4 we can see that the proposed methods using oversampling do not differ statistically from the reference methods, except for a single classifier, which is characterized by high *precision* but at the same time achieves the worst mean ranks based on the remaining metrics. Together with the increase in the base classifier number, KNORA-U and OSB achieve higher average ranks than ODESE-C and ODESE-A.

3.3 Experiment 2 – Scaled Euclidean Distance-Based Approach

The results below show the average ranks for the proposed DESIRE method, which calculates weights based on Euclidean distances scaled by the percentages of the minority and majority classes in the training set.

In the case of GNB as the base model (Fig. 4), the ODESIRE-C method achieves the best results compared to reference methods in terms of mean ranks based on *F1 score*, *precision*, *G-mean* and BAC. When the ensemble size increases, the proposed method is equal to OKNORA-U in terms of BAC and *G-mean* but retains the advantage in terms of *F1 score* and *precision*. Also, the more base classifiers the smaller the differences between DESIRE using preprocessing and the version without it. Table 5 presents the results of the statistical analysis, which shows that ODESIRE-C is statistically better than all reference methods when the number of base classifiers is low.

Figure 5 shows that for a small classifier pool, ODESIRE-C achieves higher ranks than reference methods in terms of each evaluation metric, and as the classifier number increases, it loses significantly in *precision* compared to OSB and OKNORA-U. ODESIRE-A has a high *recall*, which unfortunately is reflected by the lowest *precision* and *F1 score*. In Table 6 we see that for 5 base classifiers, DSIRE-C both with and without preprocessing is statistically significantly better than reference methods in terms of all metrics except one, *G-mean* in the case DESIRE-C and *recall* for ODESIRE-C.

When the base classifier is KNN (Fig. 6), as in the case of DESE, ODESIRE-C is not statistically worse than OSB and OKNORA-U (Table 7) and as the number of classifiers in the pool increases, the average global ranks significantly deteriorate compared to reference methods.

Table 5. Statistical tests on mean ranks for GNB with pool size = 5.

	GNB (1)	OSB (2)	OKNORA-U (3)	DESIRE-C (4)	ODESIRE-C (5)	DESIRE-A (6)	ODESIRE-A (7)
F1 score	2.341 –	2.280 –	4.159 1,2	5.634 1,2,3,6,7	6.098 1,2,3,6,7	3.878 1,2	3.610 1,2
precision	2.244 –	2.098 –	3.902 1,2	6.341 1,2,3,6,7	6.098 1,2,3,6,7	3.976 1,2,7	3.341 1,2
recall	4.037 4	4.890 4,5	4.427 4,5	1.939 –	3.305 4	4.183 4,5	5.220 1,3,4,5,6
G-mean	2.341 –	2.793 –	4.622 1,2,6	4.829 1,2,6,7	5.976 *all*	3.610 1	3.829 1,2
BAC	2.341 –	2.634 –	4.427 1,2,6	4.829 1,2,6	6.061 *all*	3.610 1,2	4.098 1,2

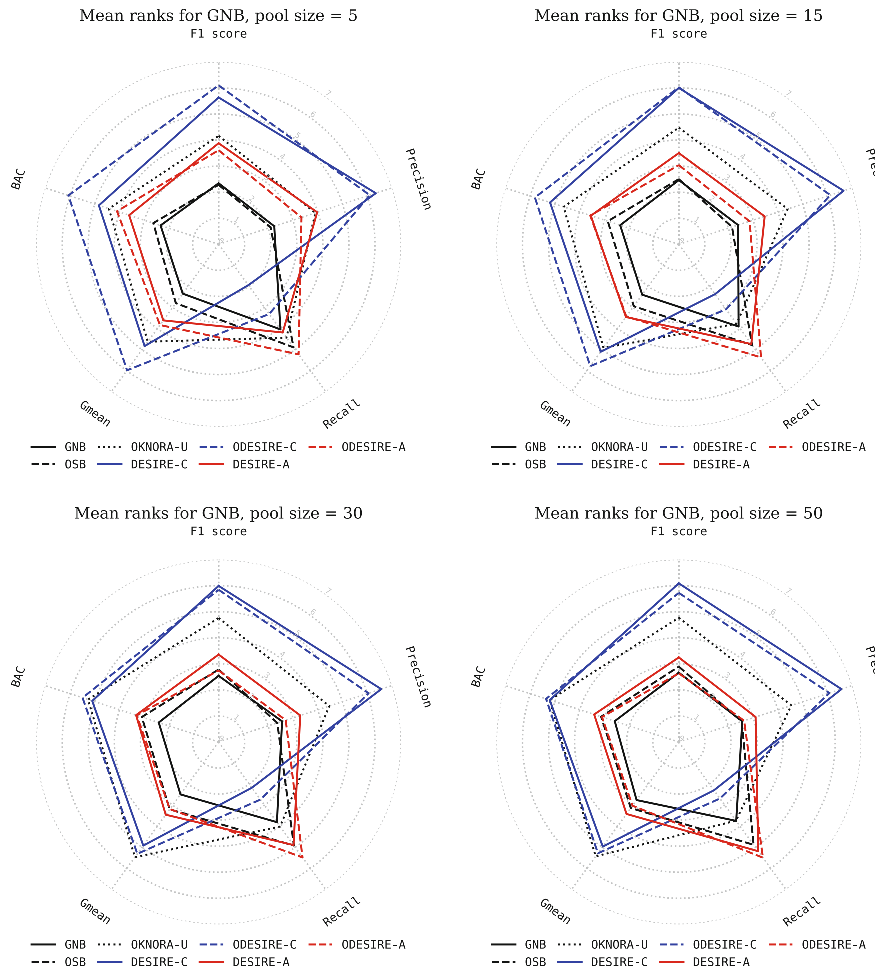


Fig. 4. Mean ranks for GNB classifier.

Table 6. Statistical tests on mean ranks for CART with pool size = 5.

	CART **(1)**	**OSB** **(2)**	**OKNORA-U** **(3)**	**DESIRE-C** **(4)**	**ODESIRE-C** **(5)**	**DESIRE-A** **(6)**	**ODESIRE-A** **(7)**
F1 score	3.415 6	3.768 6	3.915 6,7	5.622 1,2,3,6,7	5.768 1,2,3,6,7	2.524 –	2.988 –
precision	3.683 6,7	4.659 1,6,7	4.878 1,6,7	5.793 *all*	5.256 1,6,7	1.793 –	1.939 –
recall	3.146 2,3	2.488 –	2.561 –	3.793 2,3	4.110 1,2,3	5.817 1,2,3,4,5	6.085 1,2,3,4,5
G-mean	3.049 –	2.598 –	2.744 –	4.280 1,2,3	4.817 1,2,3,4	5.183 1,2,3,4	5.329 1,2,3,4
BAC	3.073 –	2.537 –	2.683 –	4.744 1,2,3	5.110 1,2,3	4.695 1,2,3	5.159 1,2,3

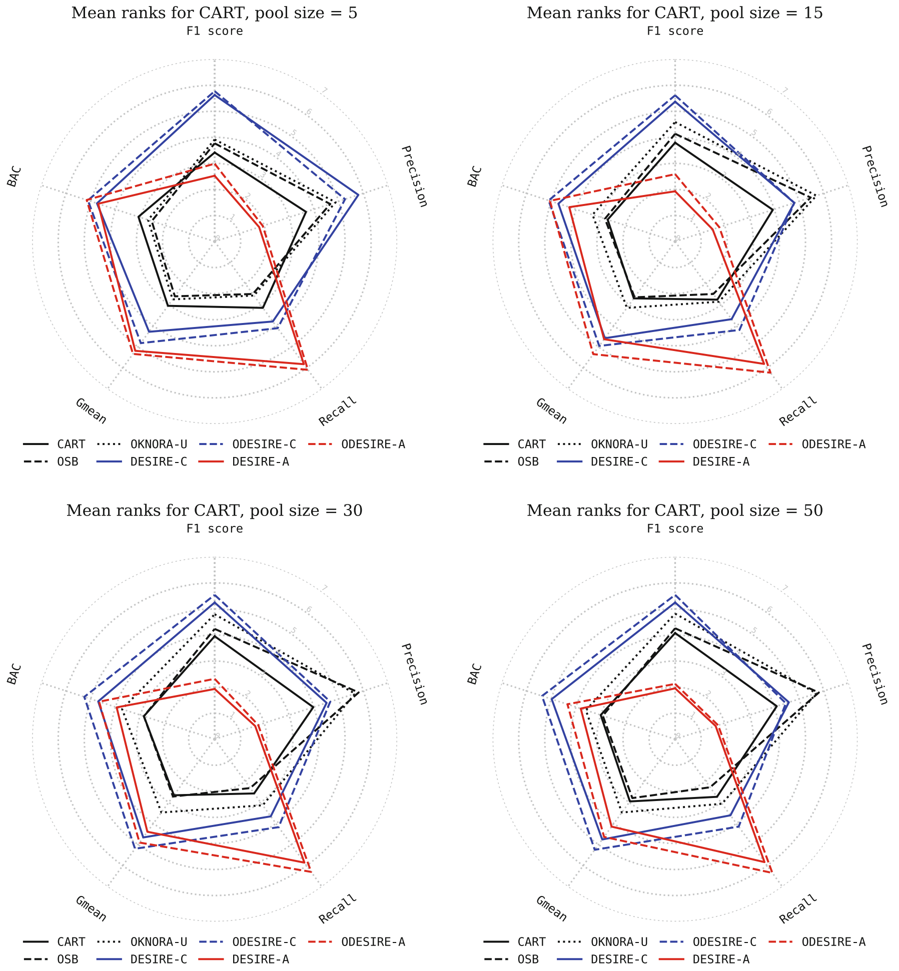
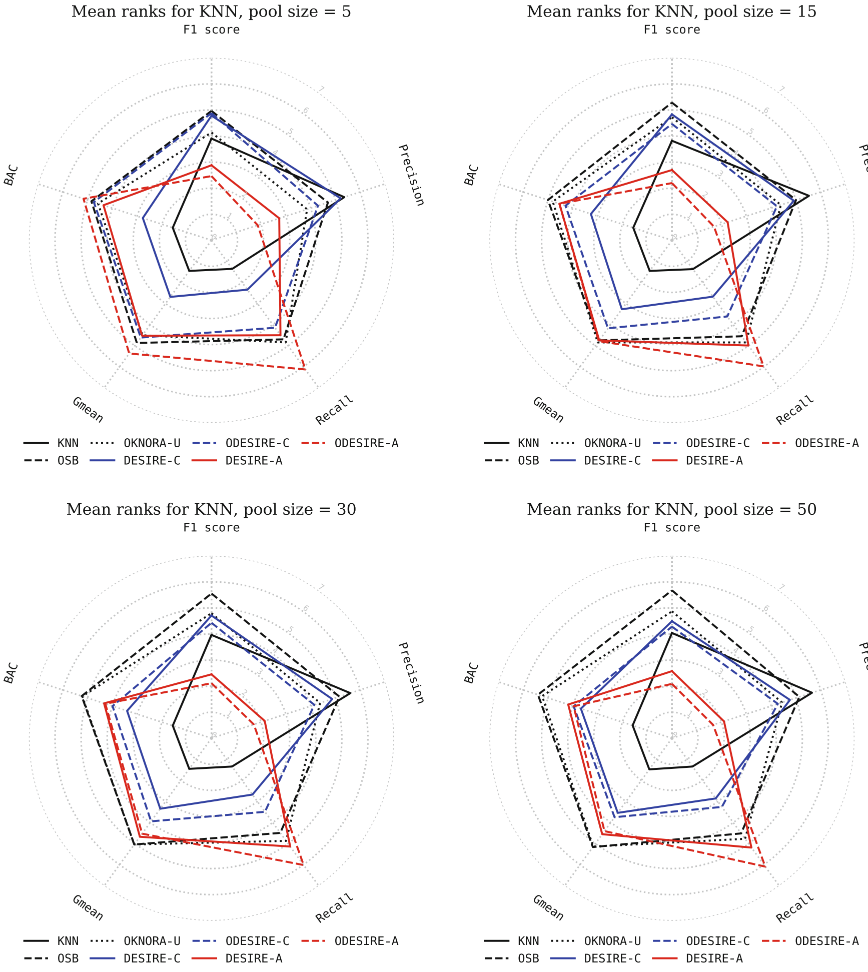


Fig. 5. Mean ranks for CART classifier.

Table 7. Statistical tests on mean ranks for KNN with pool size = 5.

	KNN **(1)**	OSB **(2)**	OKNORA-U **(3)**	DESIRE-C **(4)**	ODESIRE-C **(5)**	DESIRE-A **(6)**	ODESIRE-A **(7)**
F1 score	3.902	4.963	4.134	4.780	4.878	2.878	2.463
	6,7	1,3,6,7	6,7	6,7	6,7	–	–
precision	5.354	4.695	3.854	5.207	4.293	2.732	1.866
	5,6,7	3,6,7	6,7	3,5,6,7	6,7	7	–
recall	1.354	4.695	4.841	2.341	4.146	4.500	6.122
	–	1,4	1,4	1	1,4	1,4	*all*
G-mean	1.451	4.866	4.500	2.683	4.610	4.524	5.366
	–	1,4	1,4	1	1,4	1,4	1,3,4,5,6
BAC	1.561	4.841	4.573	2.768	4.744	4.354	5.159
	–	1,4	1,4	1	1,4	1,4	1,4,6



Mean ranks for KNN, pool size = 5

Mean ranks for KNN, pool size = 15

Mean ranks for KNN, pool size = 30

Mean ranks for KNN, pool size = 50

Fig. 6. Mean ranks for KNN classifier.

3.4 Lessons Learned

The presented results confirmed that dynamic selection methods adapted specifically for the imbalanced data classification can achieve statistically better results than *state-of-art* ensemble methods coupled with preprocessing, especially when the pool of base classifiers is relatively small. This may be due to the fact that *bagging* has not yet stabilized, while the proposed method chooses the best single classifier. The *Correct* approach in which the weights of the models were changed only if the instances belonging to the local competence region were correctly classified, proved to be more balanced in terms of all 5 evaluation measures. This may indicate too high weight penalties with incorrect classification in the *All* approach. When KNN is used as the base classifier, with a small pool the proposed methods performed statistically similar to KNORA-U, and with a larger number of classifiers, achieved statistically inferior rank compared to the reference methods. This may be probably due to the support calculation method in the KNN, which is not suitable for the algorithms proposed in this work. For GNB and CART, DESE-C and DESIRE-C achieved results which are statistically better than or similar to the reference methods, often without the use of preprocessing, since it has a built-in mechanism to deal with the imbalance.

4 Conclusions

The main purpose of this work was to propose a novel solution based on dynamic classifier selection for imbalanced data classification problem. Two methods were proposed, namely DESE and DESIRE, which use the Euclidean distance and imbalance ratio in the training set to select the most appropriate model for the classification of each new sample. Research conducted on benchmark datasets and statistical analysis confirmed the usefulness of proposed methods, especially when there is a need to maintain a relatively low number of classifiers.

Future work may involve the exploration of different approaches to the base classifiers' weighting, as well as using different combination methods and the use of proposed methods for the imbalanced data stream classification.

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

References

1. Alcala-Fdez, J., et al.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Mult. Valued Log. Soft Comput. **17**, 255–287 (2010)
2. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996). https://doi.org/10.1007/BF00058655
3. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR 2010, Washington, DC, USA, pp. 3121–3124. IEEE Computer Society (2010)
4. Chen, D., Wang, X.-J., Wang, B.: A dynamic decision-making method based on ensemble methods for complex unbalanced data. In: Cheng, R., Mamoulis, N., Sun, Y., Huang, X. (eds.) WISE 2020. LNCS, vol. 11881, pp. 359–372. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34223-4_23
5. Cruz, R.M.O., Sabourin, R., Cavalcanti, G.D.C.: Dynamic classifier selection: recent advances and perspectives. Inf. Fus. **41**, 195–216 (2018)
6. Duin, R.P.W.: The combining classifier: to train or not to train? In: Object Recognition Supported by User Interaction for Service Robots, vol. 2, pp. 765–770, August 2002
7. Ko, A.H., Sabourin, R., Alceu Souza Britto, J.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recogn. **41**(5), 1718–1731 (2008)

8. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress Artif. Intell. **5**(4), 221–232 (2016). https://doi.org/10.1007/s13748-016-0094-0
9. Ksieniewicz, P.: Undersampled majority class ensemble for highly imbalanced binary classification. In: Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications. Proceedings of Machine Learning Research, Dublin, Ireland, vol. 94, pp. 82–94. PMLR, ECML-PKDD, 10 September 2018
10. Ksieniewicz, P., Zyblewski, P.: Stream-learn-open-source Python library for difficult data stream batch analysis. arXiv preprint arXiv:2001.11077 (2020)
11. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In: ICML (1997)
12. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
13. Powers, D.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol **2**, 2229–3981 (2011)
14. Roy, A., Cruz, R.M., Sabourin, R., Cavalcanti, G.D.: A study on combining dynamic selection and data preprocessing for imbalance learning. Neurocomputing **286**, 179–192 (2018)
15. Sasaki, Y.: The truth of the F-measure. Teach Tutor Mater, January 2007
16. Woźniak, M., Graña, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Inf. Fus. **16**, 3–17 (2014). Special Issue on Information Fusion in Hybrid Intelligent Fusion Systems
17. Zyblewski, P., Ksieniewicz, P., Woźniak, M.: Classifier selection for highly imbalanced data streams with *minority driven ensemble*. In: Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (eds.) ICAISC 2019. LNCS (LNAI), vol. 11508, pp. 626–635. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20912-4_57