Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 3477–3486

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Framework for imbalanced data classification

Mikołaj Błaszczyk[a,*], Joanna Jędrzejowicz[a]

[a]*Institute of Informatics, Faculty of Mathematics, Physics and Informatics, University of Gdańsk, 80-308 Gdańsk, Poland*

Abstract

Classifying imbalanced data remains a challenging task. The paper presents a framework for imbalanced datasets classification which makes use of different methods of oversampling and methods of dynamical selection of classifiers. The framework allows to perform extensive experiments to determine best possible configuration for the examined dataset in terms of geometric mean metric (g-mean). The results on benchmark datasets are presented.

Keywords: imbalanced data; oversampling; dynamic selection of classifiers

1. Introduction

Building classifiers for imbalanced datasets is a difficult and demanding task. In case of binary classification, considered in this paper, data from one class called majority class outnumber instances from the other, that is minority class. Traditional classification algorithms do not perform well on imbalanced datasets. Classic algorithms often fail to represent data distribution when confronted with high imbalance ratio (between the number of the majority instances and the number of minority class instances). Even if the classification performance is close to 99 percent accuracy for the majority class, it can be quite misleading if the minority class is 1 percent of the whole dataset. Minority class examples can be treated as noise by the classifier, and vice versa, noise wrongly identified as minority class. What is more, imbalanced datasets appear in real-world applications, for example fraudulent credit card transaction detection, software defect prediction, bioinformatics and biomedical decision making, and many other. Generally, the minority class instances is of greater interest and higher cost of misclassifying.

Classifying imbalanced data has been extensively studied. Already in the review [4] over five hundred papers were collected and examined from the perspective of applied methods and achieved results on the benchmark datasets. Since then the number of papers, specialized workshops, and conferences has grown tremendously. And still there is motivation to look for more effective and accurate methods for classifying imbalanced data.

* Corresponding author.
E-mail address: m.blaszczyk.312@studms.ug.edu.pl

10.1016/j.procs.2021.09.121

In the literature two types of approaches are applied: data-level solutions and algorithm-level ones. The data-level approaches aim to reduce the imbalance ratio by either undersampling the data in the majority class [14], oversampling the minority instances [16], [5], or combining both methods [3]. In the algorithm-driven approach no change to the input data is applied. In this case algorithm classifier modifications and ensemble methods are applied. Since combinations of sampling with ensemble classifiers proved to be very promising, as follows from the study [14], in this paper both approaches are used in a framework allowing for extensive experiments with undersampling/oversampling and dynamic selection of classifiers and groups of classifiers. The framework includes preprocessing which makes use of sampling methods, classification and finally evaluation. The main contribution of this paper is the development of the framework which allowed to conduct extensive number of experiments with varying methods of sampling and varying choice of dynamically chosen classifiers. The paper reports only some results due to the limits of the paper length. The second contribution is a new method of selecting classifiers, denoted as KNORA-Pass in what follows.

The paper is organized as follows. Section 2 contains brief overview of recent results in the field. Section 3 presents the general idea of methods used in the framework. Section 4 describes benchmark datasets and accuracy measures used in the computational experiments and the results of experiments, and finally Section 5 contains conclusions.

2. Related work

The classification of imbalanced data is a complex task and standard classifiers, based on the assumption of even distribution of instances among classes, can not cope with it. Therefore most of specialized methods use some form of sampling. Both, undersampling and oversampling change the distribution by eliminating the majority class instances, or increasing the minority class. In such case there is danger of deleting some potentially useful data, or in case of oversampling, increase the possibility of over-fitting. For example SMOTE [3] is a well known method that creates new samples belonging to the minority class by generating artificial data, so-called synthetic samples. This is achieved by linearly interpolating a random minority instance and one of its neighbors from the minority class. The method is popular and different aspects were studied: choice of sampling rates, different methods of creating new data, making use of all samples in the process etc. Another method, ADASYN [5], generates new examples depending on the weights of minority instances which determine how many times the instance may be used for synthetic examples generation. The weight is defined as the proportion of the majority class samples in the neighborhood of a synthesized instance.

An important issue in applying sampling is finding the best possible imbalance ratio [13]. In [15] genetic algorithms were applied to determine the proportion of undersampled and oversampled instances. This approach was lately continued and expanded in [8], where the special strategy was applied for the stopping the genetic algorithm when the performance of the classifier deteriorates.

Several methods for coping with imbalanced data use clustering algorithms as a preprocessing stage in order to undersample the majority class by choosing prototypes from clusters to balance with the minority class, see [11], [14].

Algorithms specially tailored for imbalanced data have been suggested in several papers. The SplitBal algorithm [12] combined ensemble learning with dividing the majority class instances into bins matching, in size, the minority class and generating a different classifier for each respective balanced set. In [6] Gene Expression Programming classifier was adapted to imbalanced environment with the application of incremental learning paradigm. Rare- class Nearest Neighbor algorithm from [17] which is a modification of kNN algorithm introduces dynamic local query neighborhood to ensure adequate presence of data from the minority class.

3. Proposed approach

As mentioned earlier our method performs sampling first, then applies basic classifiers and finally performs dynamic selection of classifiers. In this last step the knowledge on how a given classifier is performing on the neighbours of a given instance, is used. Therefore each dataset used in the experiments is partitioned into: training set, used for basic classifiers, validation set used in the process of selecting members of the ensemble, and finally testing set to evaluate the ensemble. This guarantees that no data-leakage appears.

Table 1. Datasets used in experiments.

	Yeast5	Pageblocks0	Mammography
IR (imbalance ratio)	32.73	8.79	42.01
Instances	1483	5472	11182
Attributes	7	10	5

3.1. Dynamic selection of classifiers

There are different ways to select classifiers to estimate given testing sample. Three main ones described in [9] are: static ensemble selection, dynamic classifier selection (DCS) and dynamic ensemble selection (DES). The difference between static and dynamic methods is that static selection chooses one ensemble of classifiers for all testing samples meanwhile dynamic selection methods choose different ensembles of classifiers for different testing samples. The further difference between DCS and DES is that DCS methods choose different single classifier for different testing sample meanwhile DES choose whole ensemble of classifiers.

In [9] the authors introduced a version of dynamic ensemble selection of classifiers using K-nearest-oracles (KNORA) DES method. Instead of choosing one ensemble of classifiers for all test samples we can create ensembles dynamically by selecting different ensembles for different test samples. For any test data point KNORA finds its K-nearest neighbors in the validation set, which will be called examples in what follows. Using the knowledge about the correct class of examples it checks which classifier classified correctly. As shown in Fig. 1 for a test instance it finds its K closest examples in the validation set and based on classifier selection criteria decides which classifier should be used and added to the ensemble for classifying given sample in the test dataset. Authors proposed 4 selection criteria out of which we use 2:

- KNORA-Eliminate - for a given number of K closest examples in validation dataset of test sample, it checks if any of provided classifiers correctly estimates all of these K examples. In case it does, given classifier is added to the ensemble of classifiers. After assembling the ensemble, all of the chosen classifiers submit their votes on tested sample. In case there was no classifier that could predict every example, we decrease the number of K closest examples, by one until at least one of the classifiers correctly predicts all of K examples,
- KNORA-Union - for a given number of K closest examples in validation dataset of test sample it checks how many times given classifier correctly estimates an example in K closest examples. After that every classifier is added to ensemble with given number of votes that equals the number of correct predictions on K examples in validation dataset.

Based on these rules we added our own selection criteria:

- KNORA-Pass - which for a given number of K closest examples on validation dataset of test sample checks if any of provided classifiers correctly passes the prediction test with ”pass” value. For example we set the pass value to 70% and check if any of provided classifiers correctly estimates 70% of K closest examples on validation set. In the case where the classifier passes the ”test” it is later added to the ensemble of classifiers. The idea is to eliminate weakest classifiers and add these which can predict most samples.

4. Computational experiments and datasets

We report the experiments conducted on 2 datasets from KEEL repository [1] and 1 dataset from MachineLearning-Mastery repository [2]. Each with different number of instances, different number of attributes and different imbalance ratio. These datasets are summarized in Table 1.

For each of the three datasets, six basic classifiers were trained (Tables 2-3) with the parameters as follows: Decision Tree, KNN Classifier (K-nearest neighbours) with numbers of neighbours = 5, SVC (Support Vector Machine), Naive

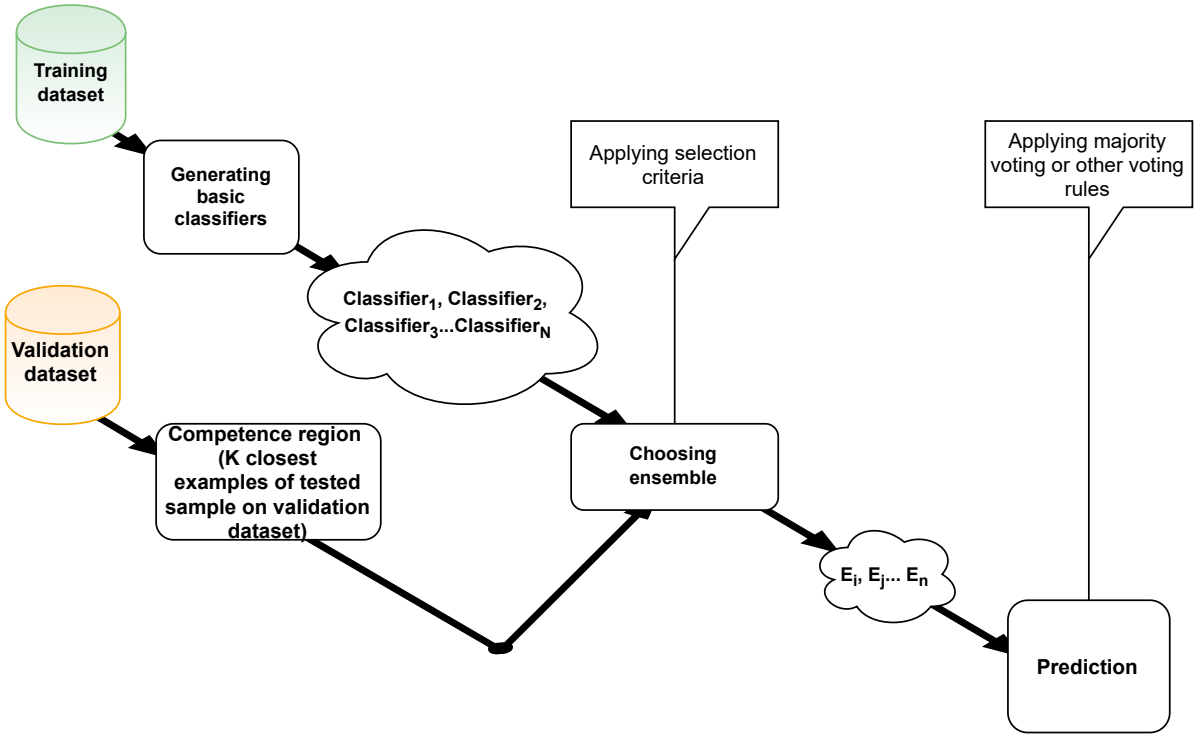

Fig. 1. Process of prediction for KNORA DES methods

Bayes, Random Forest with number of trees in forest = 100 and number of features considered when looking for best split equal to $n = \sqrt{numberOfFeatures}$, Multilayer Perceptron with constant $learningrate = 0.001$ and maximum number of iterations equal 1000. Other parameters were set as default in sklearn library.

4.1. Best accuracy metric

To show most realistic accuracy score, the g-mean value was used to display performance of every basic classifier. G-mean is a geometric mean of sensitivity and specificity:

$$g - mean = \sqrt{sensitivity \times specificity} = \sqrt{\frac{TP}{P} \times \frac{TN}{N}}$$

where $P = TP + FN$, $N = FP + TN$ and TP, TN, FP, FN are, respectively, true positives, true negatives, false positives and false negative values from the confusion matrix.

As we are looking at imbalanced sets of data we cannot evaluate the accuracy as the number of correct predictions divided by the number of all instances. That method only works if the number of occurrences of minority class is equal to that of majority class. That is not the case we are testing. As we can see in Table 1 the imbalance ratio ranges from 8.79 to 42.01 so the datasets are highly imbalanced.

As stated in M. Kubat's work [10]: *"this score maximizes accuracy on each of the two classes while keeping these accuracies balanced."* Owing to that solution, even if we get high accuracy on majority class and low accuracy on minority class - the g-mean score will result in low value.

Tables 2, 3, 4 show accuracy of every basic classifier trained on respective datasets with different balancing technique used. Three highest g-mean scores are denoted in bold. To balance data two synthetic methods were used - SMOTE and ADASYN. To check their pertinence we also conducted two other experiments - one with Random oversampling (ROS) and another one without any balancing technique.

Table 2. Comparison of g-mean for **base** classifiers for yeast5 dataset. Parameters described in 4.

Yeast5 Dataset	Decision Tree	KNN	SVC	Bayes	Random Forest	Multilayer Perceptron
No balancing	**0.8586**	**0.8186**	0.0	**0.7975**	0.7504	0.0
SMOTE	0.9072	**0.9587**	**0.9573**	0.8296	0.9081	**0.9631**
ADASYN	0.8949	**0.9704**	**0.9566**	0.8068	0.9194	**0.9621**
ROS	0.8828	**0.9492**	**0.9584**	0.7600	0.8850	**0.9653**

Table 3. Comparison of g-mean for **base** classifiers for pageblocks0 dataset. Parameters described in 4.

Pageblocks0 Dataset	Decision Tree	KNN	SVC	Bayes	Random Forest	Multilayer Perceptron
No balancing	**0.8450**	0.7802	0.3724	0.6116	**0.8598**	**0.8160**
SMOTE	**0.9067**	0.8811	0.5162	0.6345	**0.9177**	**0.9023**
ADASYN	**0.8933**	0.8891	0.4956	0.6674	**0.9302**	**0.9195**
ROS	0.8774	**0.8834**	0.4686	0.6255	**0.8997**	**0.8937**

Table 4. Comparison of g-mean for **base** classifiers for mammography dataset. Parameters described in 4.

Mammography Dataset	Decision Tree	KNN	SVC	Bayes	Random Forest	Multilayer Perceptron
No balancing	**0.7158**	**0.7051**	0.6491	**0.8333**	0.7007	0.6004
SMOTE	0.8222	**0.8896**	**0.9072**	0.8494	0.8711	**0.8945**
ADASYN	0.8141	**0.8884**	**0.8932**	0.8154	0.8588	**0.8997**
ROS	0.7485	**0.8605**	**0.9019**	0.8592	0.8056	**0.9018**

To provide more reliable results we used Stratified K-Folds cross validator. We computed g-mean value of all basic classifiers in each of the 5 folds and after that summarized with the mean value of all folds.

There are cases when g-mean score can achieve 0.0 value. It can only occur in 2 cases: either the created classifier could not correctly predict any of majority class sample or any of minority class sample. For example in SVC method used on yeast5 dataset without any balancing technique the created classifier could not correctly predict any of minority class sample provided in testing dataset.

4.2. Preparing DES methods

Dynamic Ensemble Selection (DES) methods need pool of classifiers to work as intended. That is why we created three different pools for each balancing method used on each dataset. They are as follows:

- a pool with all six basic classifiers (named later as ”all pool”),
- a pool containing only 3 basic classifiers with highest g-mean score (named later as ”top 3 pool”),
- a pool containing 5 classifiers - 3 duplication of classifier with the highest g-mean score and 2 with next highest accuracy (named later as ”top 3 duplicate pool”).

The next thing to do is to provide DES methods with the region of competence argument. The region of competence in our experiments is assumed as the number of 15 closest examples checked in validation set for the given classified sample. It means that when DES method checks provided sample it then finds 15 closest examples of that sample in validation set and checks accuracy of every classifier in the classifiers pool on this small data set. To show how the size of the competence region relates to g-mean, we show in Fig. 2 the results for all of DES algorithms using ”all pool”, computed for ”yeast5” dataset with competence region starting from 3 closest examples to 21. Horizontal axis contains the number of examples and vertical axis - the value of g-mean.

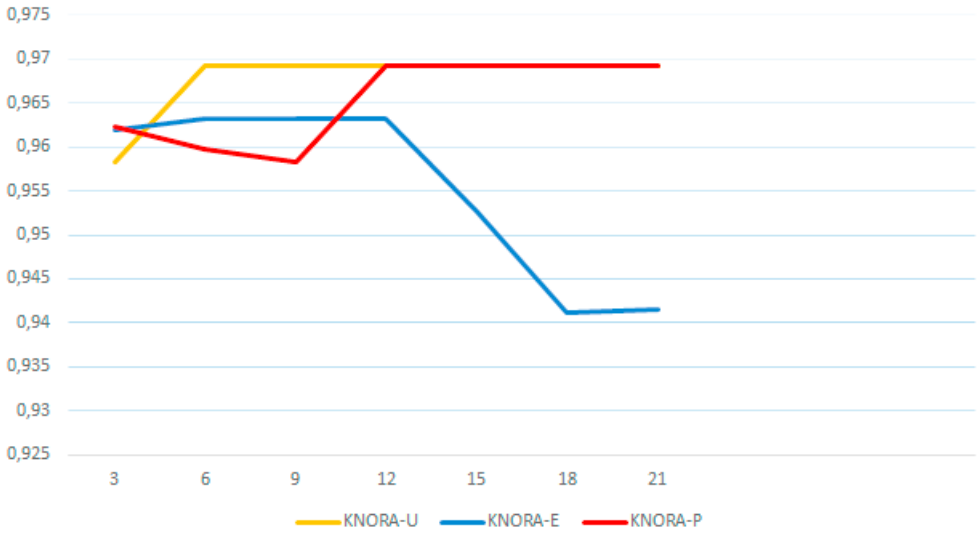

Fig. 2. Change of g-mean (Y axis) for different regions of competence (X axis) for yeast5 dataset.

4.3. Checking accuracy of DES methods

As described in Section 3 we prepared three different Dynamic Ensemble Selection algorithms. All of them were based on K-nearest oracle with elimination, union and pass rules and were conducted as shown in Fig. 3. As we can see in Fig. 4, 5 we can achieve same or better accuracy when using dynamic selection methods in best case scenarios. In most cases ADASYN technique was generally speaking better balancing method for both ”pageblocks0” and ”yeast5” datasets.

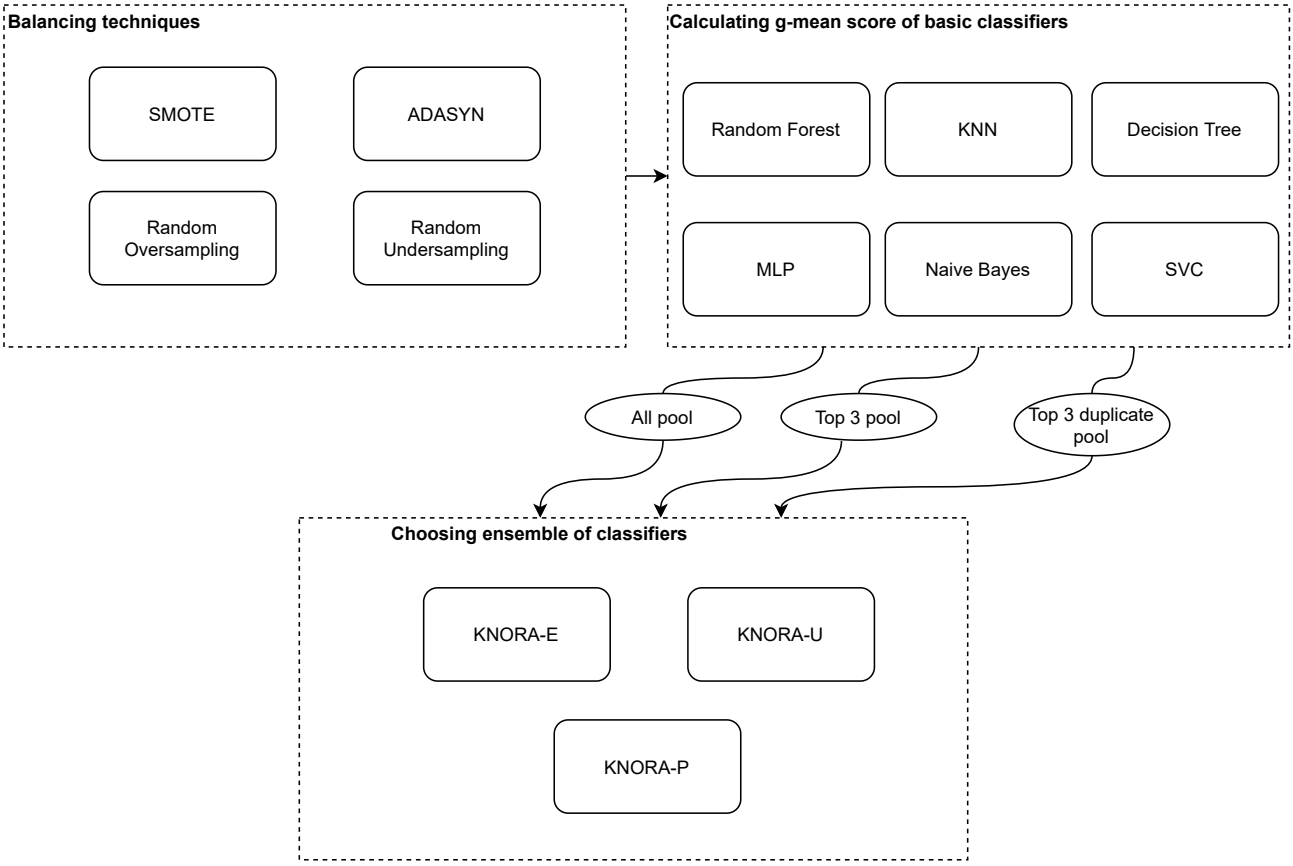

Fig. 3. Chart showing process of conducted experiments.

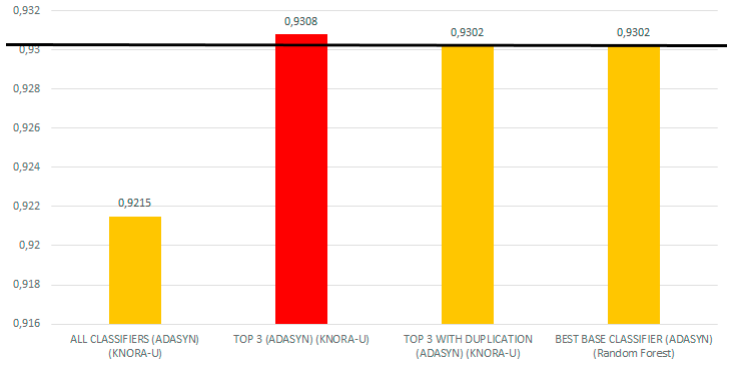

Fig. 4. Chart showing g-mean value (Y axis) for methods with the highest score for each kind of pool (X axis) for pageblocks0 dataset. Cases with best accuracy are coloured in red. A black line shows the accuracy of the best basic classifier.

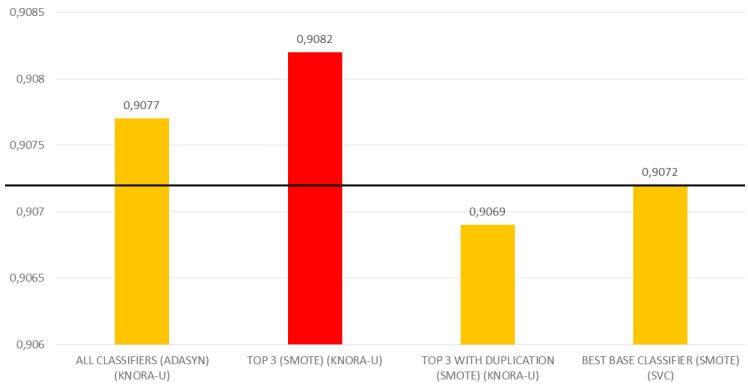

Fig. 5. Chart showing g-mean value (Y axis) for methods with the highest score for each kind of pool (X axis) for mammography dataset. Cases with best accuracy are coloured in red. A black line shows the accuracy of the best basic classifier.

What is interesting in this case ”top 3 duplicate pool” achieved the same g-mean score as best basic classifier. Probably with proportion of 3 instances of best classifier and 2 other classifiers, our created model always decides on the class estimated by this one duplicated classifier as it holds majority in voting pool. Fig. 5 shows an interesting change. ”Top 3 duplicate pool” achieved lower g-mean score than best basic classifier. As of tested datasets we can conclude that ”top 3 duplicate pool” with proportion 3 of the same classifier to 2 of other classifiers is not a good idea to follow. The best classifier in most cases will dominate in estimation phase preventing us from increasing g-mean score accuracy or even making it lower like in Fig. 5.

Observing ”top 3 pool” we can see completely different results. Where in ”yeast5” dataset this pool achieved the worst accuracy it is contrariwise for both ”pageblocks0” and ”mammography” datasets. Here in both cases KNORA-U algorithm with ”top 3 pool” was the best method to tackle these datasets. It is worth to observe that both of these datasets contain much more instances than ”yeast5” (where in ”pageblocks0” it is nearly 4 times more instances and in ”mammography” it is nearly 8 times more instances) and that can be a reason why DES methods tackled the problem better with ”top 3 pool” on those datasets.

The last interesting pool is ”all pool”. In both ”yeast5” and ”pageblocks0” datasets it displayed worse results. The only case when it proved better than basic classifier was when used on ”mammography” dataset. As weaker classifiers affect DES methods it proves that it is better to not include every possible classifier in dynamic selection pools.

As we deducted that DES methods performed best with SMOTE and ADASYN balancing techniques and with ”top 3 pool” let us check how all of the 3 algorithms tackled those two datasets. We can see in Fig. 6 that KNORA-P

was better than basic classifier in 2 out of 4 cases, KNORA-E was better in 2 out of 4 cases and KNORA-U was better in 3 out of 4 cases.

Fig. 6. Chart showing g-mean value for DES methods balanced with SMOTE and ADASYN for both ”pageblocks0” and ”mammography” datasets.

5. Conclusions and future work

In this work we tried to show how DES algorithms tackle the problem of imbalanced datasets in cooperation with balancing techniques like SMOTE, ADASYN and ROS. We have also proven that proper matching of classifiers to DES classifiers pool have a direct impact on the prediction accuracy.

Choosing every available classifier to the DES classifiers pool is not always a good idea. It is especially visible when some of the basic classifiers achieve low g-mean score. Very important thing with DES methods is to keep large variety of basic classifiers and at the same time low number of basic classifiers with low g-mean score. Dynamic Ensemble Selection algorithms with proper preparation can achieve better results than basic classifiers. There is still much work to be done. Presented three selection criteria: KNORA-E, KNORA-U and KNORA-P are very simple so there is a possibility to create even better criteria. Likewise checking the competence region as K closest examples is the most simplistic way to check which of the classifiers in pool we should choose. We believe that the DES methods can be greatly improved by conducting more research on finding for the best validation data for tested sample.

Our proposed criteria KNORA-P was not the best DES classifier as we could see in Fig. 4 and 5. Still it achieved second best g-mean score from all the 3 DES methods in most tested cases. As it is a very straightforward way of selecting best classifiers for the ensemble, it probably can also be improved.

It is important to note that all experiments concerned binary classification. We did not check how DES methods could perform on multi classification problem. From other works [9] and [7] we can only conclude that dynamic selection approach can also help improve classification accuracy with multi classification problems like for example letter recognition.

Further work can be related to looking for other methods for choosing competence region, new approach to find best classifier from classifiers pool and last but not least conducting more tests on multi classification data sets.

References

[1] Alcalá-Fdez, J., Sánchez, L., García, S., del Jesús, M.J., Ventura, S., i Guiu, J.M.G., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F., 2009. KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput. 13, 307–318. URL: http://dx.doi.org/10.1007/s00500-008-0323-y, doi:10.1007/s00500-008-0323-y.

[2] Brownlee, J., . Machine learning datasets used on machinelearningmastery.com. https://github.com/jbrownlee/Datasets. Accessed: 2021-03-30.

[3] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357. URL: https://doi.org/10.1613/jair.953, doi:10.1613/jair.953.

[4] Guo, H., Li, Y., Shang, J., Mingyun, G., Yuanyue, H., Bing, G., 2017. Learning from class-imbalanced data: Review of methods and applications. Expert Syst. Appl. 73, 220–239. URL: https://doi.org/10.1016/j.eswa.2016.12.035, doi:10.1016/j.eswa.2016.12.035.

[5] He, H., Bai, Y., Garcia, E.A., Li, S., 2008. Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IJCNN 2008, pp. 1322–1328.

[6] Jedrzejowicz, J., Jedrzejowicz, P., 2021. Gep-based classifier for mining imbalanced data. Expert Syst. Appl. 164, 114058. URL: http://www.sciencedirect.com/science/article/pii/S0957417420308204, doi:https://doi.org/10.1016/j.eswa.2020.114058.

[7] Jr, A., Sabourin, R., Soares de Oliveira, L., 2014. Dynamic selection of classifiers—a comprehensive review. Pattern Recognition 47, 3665–3680. doi:10.1016/j.patcog.2014.05.003.

[8] Karia, V., Zhang, W., Naeim, A., Ramezani, R., 2019. Gensample: A genetic algorithm for oversampling in imbalanced datasets. CoRR abs/1910.10806. URL: http://arxiv.org/abs/1910.10806, arXiv:1910.10806.

[9] Ko, A.H., Sabourin, R., Alceu Souza Britto, J., 2007. From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition 41, 1718–1731. URL: https://doi.org/10.1016/j.patcog.2007.10.015.

[10] Kubat, M., Matwin, S., 1997. Addressing the curse of imbalanced training sets: One-sided selection, in: Fisher, D.H. (Ed.), Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997, Morgan Kaufmann. pp. 179–186.

[11] Lin, W.C., Tsai, C.F., Hu, Y.H., Jhang, J.S., 2016. Clustering-based undersampling in class-imbalanced data. Inf. Sci. 409, 17–26.

[12] Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., Zhou, Y., 2015. A novel ensemble method for classifying imbalanced data. Pattern Recognition 48, 1623–1637. URL: http://dx.doi.org/10.1016/j.patcog.2014.11.014, doi:10.1016/j.patcog.2014.11.014.

[13] Tharwat, A., 2018. Classification assessment methods. Appl. Computing and Informatics doi:10.1016/j.aci.2018.08.003.

[14] Tsai, C., Lin, W., Hu, Y., Yao, G., 2019. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. Inf. Sci. 477, 47–54. URL: https://doi.org/10.1016/j.ins.2018.10.029, doi:10.1016/j.ins.2018.10.029.

[15] Vannucci, M., Colla, V., 2018. Genetic algorithms based resampling for the classification of unbalanced datasets. Springer International Publishing, Cham. pp. 23–32. URL: https://doi.org/10.1007/978-3-319-59424-8_3, doi:10.1007/978-3-319-59424-8_3.

[16] Wang, X., Xu, J., Zeng, T., Jing, L., 2021. Local distribution-based adaptive minority oversampling for imbalanced data classification. Neurocomputing 422, 200–213. URL: https://doi.org/10.1016/j.neucom.2020.05.030, doi:10.1016/j.neucom.2020.05.030.

[17] Zhang, X., Li, Y., Ramamohanarao, K., Wu, L., Tari, Z., Cheriet, M., 2017. KRNN: k rare-class nearest neighbour classification. Pattern Recognition 62, 33–44. URL: http://dx.doi.org/10.1016/j.patcog.2016.08.023, doi:10.1016/j.patcog.2016.08.023.