Balancing agents for mining imbalanced multiclass datasets – performance evaluation

Joanna Jedrzejowicz[1][0000−0003−4979−5476]
and Piotr Jedrzejowicz[2][0000−0001−6104−1381]

[1] Institute of Informatics, Faculty of Mathematics, Physics and Informatics, University of Gdansk, 80-308 Gdansk, Poland, joanna.jedrzejowicz@ug.edu.pl
[2] Department of Information Systems, Gdynia Maritime University, 81-225 Gdynia, Poland, p.jedrzejowicz@umg.edu.pl

Keywords: Imbalanced data · Multiclass datasets · Dataset balancing.

Abstract. The paper deals with mining imbalanced multiclass datasets. The goal of the paper is to evaluate the performance of several balancing agents implemented by the authors. Agents have been constructed from 5 state-of-the-art classifiers designed originally for mining binary imbalanced datasets. To transform binary classifiers into multiclass ones, we use the one-versus-one (OVO) approach making use of the collective decision taken by the majority voting. The paper describes our approach and provides detailed description of the respective balancing agents. Their performance is evaluated in an extensive computational experiment. The experiment involved multiclass imbalanced datasets from the Keel imbalanced datasets repository. Experiment results allowed to select best performing balancing agents using statistical tools.

1 Introduction

Multiclass imbalanced data mining is a challenging task in the field of machine learning and data mining. It refers to the scenario where a dataset contains multiple classes, but the number of instances in each class is significantly imbalanced. This can occur when one or more classes dominate the dataset, while the instances of other classes are scarce.

Imbalanced datasets can pose problems for machine learning algorithms, as they may not be able to accurately classify the minority classes due to the lack of sufficient training data. Additionally, in multiclass imbalanced data mining the mutual relationships between classes are complex and hence difficult to identify [24]. As a result, traditional machine learning algorithms tend to perform poorly on multiclass imbalanced datasets, leading to poor prediction accuracy and biased models.

To address these issues, various approaches have been proposed in the literature. As has been observed by [16], algorithms for dealing with multiclass problems can be broadly categorized into binarization approaches and ad hoc solutions. According to [7], binarization aims at decomposing the M-class problem

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

into M(M-1)/2 binary subproblems (OVO one-versus-one) or M binary subproblems (OVA one-versus-all). There has been a significant amount of research in the field of multiclass imbalanced data mining over the past few decades. Some of the key techniques that have been proposed in the literature include:

1. Undersampling: This approach involves reducing the number of instances in the majority classes, such that the resulting dataset is more balanced (see for example [20], [2]).
2. Oversampling: This approach involves increasing the number of instances in the minority classes, such that the resulting dataset is more balanced (see for example [1], [23], [18], [16]).
3. Cost-sensitive learning: This approach involves modifying the loss function of a machine learning algorithm such that it takes into account the relative importance or cost of misclassifying different classes (see for example [17], [27]).
4. Ensemble methods: Methods such as bagging, boosting, and stacking have been shown to be effective in handling imbalanced datasets (see for example [10], [26], [9], [6], [22]).
5. Algorithm level methods: Dedicated methods adapted to multiclass imbalance (see for example [12], [19], [5]).

The main advantage of undersampling is that it is simple and fast to implement, but it can also lead to the loss of important information from the majority classes. The main advantage of oversampling is that it can help improve the performance of machine learning algorithms on minority classes, but it can also lead to overfitting if the synthetic samples are not generated carefully. Another potential drawback, as pointed out in [16], is that classic oversampling algorithms consider only information from the minority class neglecting information from the majority classes. Cost-sensitive learning suffers often from the lack of information on the relative importance of misclassifying different classes. Ensemble methods can combine the predictions of multiple classifiers to improve the overall performance. Their performance relies on diversity between classifiers involved which is not always easy to achieve.

In this paper we evaluate the performance of 5 balancing agents. Balancing numbers of the majority and minority classes examples is one of a key approaches in constructing classifiers able to deal with imbalanced datasets. The idea is to preprocess training dataset to maximize the performance of a classifier used to classify data with unknown class labels. Balancing can be based on oversampling, undersampling or both. Balancing numbers of the majority and minority classes examples is one of a key approaches in constructing classifiers able to deal with imbalanced datasets. The idea is to preprocess training dataset to maximize the performance of a classifier used to classify data with unknown class labels.

All of the discussed agents have roots in binary imbalanced data classification methods and all have been implemented by us in the form of a software agents using the OVO approach to make them suitable for solving multiclass problems. The goal of the paper is to evaluate their performance. The proposed software

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

agents are goal driven, reactive, autonomous, and display collaborative behaviors in the following sense:

- They try to balance majority and minority classes examples to improve the classification performance.
- They adapt to various imbalance ratio in the training datasets.
- They are independent of the classification algorithm used.
- They reach a final decision through comparing classification decisions of base classifiers and selecting the final outcome basing on the majority vote paradigm.

The list of the original algorithms implemented as agents for balancing multiclass imbalanced datasets follows:

- Adaptive synthetic sampling approach for imbalanced learning ADASYN-M [11].
- Local distribution-based adaptive minority oversampling LAMO-M [25].
- Combined synthetic oversampling and undersampling technique CSMOUTE-M [14].
- Feature-weighted oversampling approach FWSMOTE-M [21].
- Dominance-based oversampling approach DOMIN-M [13].

ADASYN has been selected as one of the most often applied oversampling techniques. The remaining algorithms have been selected as they are known to outperform many "classic" balancing techniques. Besides, they are based on relatively newly published concepts, and all use information from not only the minority class but also from the majority classes.

The rest of the paper is constructed as follows: Section 2 contains a description of the discussed agents. Section 3 contains agent performance evaluation based on computational experiment results. The final section contains conclusions and ideas for future research.

2 Balancing agents

Assume that $D \subset X \times Y$ is a multiclass training dataset with samples $(\mathbf{x}, y)$ where $\mathbf{x}$ is an instance (datarow) and y is the class identity label associated with it.

The algorithm applied to learn the best classifier for D uses one-vs-one (OVO) method. For each pair of classes from Y, the dataset D is filtered, resulting in a subset with data from two classes. If it is imbalanced, an agent modifies it performing oversampling and/or undersampling. The modified set is used to generate the best possible classifier for the two classes. Finally, all the generated classifiers are merged to perform majority vote on data from the testing set. The OVO approach makes use of collective decision taken through a voting procedure. Algorithm 1 shows the pseudo-code for the proposed approach.

As mentioned before, the agents used to balance datasets are: ADASYN-M, LAMO-M, FWSMOTE-M, DOMIN-M, CSMOUTE-M. They are briefly described.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

Algorithm 1: Schema of the approach

Input: Multiclass dataset $D = Train \cup Test$, threshold α, agent $\mathcal{G}$
Output: values of performance metrics for D

```
   /* learning                                                    */
 1 geneList ← ∅;
 2 foreach pair of classes c1, c2 ∈ Y do
   |  /* filtering training dataset to classes c1, c2            */
 3 |  T(c1, c2) ← {(x, y) ∈ Train : y = c1 ∨ y = c2};
 4 |  if imbalance ratio of T(c1, c2) is above α then
 5 |  |  use agent G to transform T(c1, c2)
 6 |  end
 7 |  apply GEP to T(c1, c2) to obtain gene g;
 8 |  merge g to geneList;
 9 end
   /* testing                                                     */
10 foreach (x, y) ∈ Test do
11 |  foreach g ∈ geneList do
12 |  |  apply g to x, compare with y and store the result;
13 |  end
14 end
15 return performance metrics as defined in 3.2
```

The first four agents perform undersampling of the majority subset using Algorithm 2. The idea is to find the centroid of minority data and keep in the majority subset only those closest to the centroid.

Oversampling is specific for each agent type. SMOTE [4], which is an oversampling method, extending minority set via interpolation, adds elements of type $x + \lambda \cdot (z - x)$ for any minority example x, z its K-neighbor and random $\lambda \in (0, 1)$(K is a parameter). ADASYN-M agent uses adaptive sampling approach introduced in [11] to extend the minority subset, using the weighted distribution of minority examples by generating with SMOTE new minority instances whose number is proportional to the proportion of K-neighbors which are in the majority subclass.

In case of LAMO-M agent, using the method introduced in [25] (Local distribution-based Adaptive Minority Oversampling), differently than in ADASYN, not all data from the minority subset are used in generation of new synthetic data. Two steps are performed:

- using two parameters k_1, k_2 defining respectively the number of neighbors for minority and majority instances, the distribution of instances is inspected and sampling seeds identified: first instances from majority subset which appear in k_1 neighborhood of minority instances are identified and sampling seeds are those minority instances which are in k_2 neighborhood of any from the first set,
- synthetic minority instances are generated from sampling seeds using interpolation.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

Algorithm 2: Undersampling with minority class centroid

Input: data from majority class $majC$, data from minority class $minC$, parameter s - size of reduced majority class.
Output: reduced majority class $redMaj \subset majC$ of size s.

1 calculate centroid CN of $minC$
2 define distances of CN to majority instances
3 $DIST \leftarrow \{dist(x, CN) : x \in majC\}$
4 sort $DIST$ in ascending order $SORT = \{d_1, \ldots, d_{|majC|}\}$
5 keep in reduced majority class instances whose distances are in the initial s segment of $DIST$
6 $redMaj \leftarrow \{x \in majC : dist(x, CN) \leq d_s\}$
7 **return** $redMaj$

FWSMOTE-M uses the algorithm introduced in [21] which applies a method based on SMOTE where the importance of attributes is weighted by Fisher score making use of difference of attribute means in each of two classes; the weights are used when calculating distances in interpolation.

In case of DOMIN-M which is based on our method suggested in [13] the relation of domination among instances is introduced and using the genetic algorithm (GA) in subsequent iteration steps the minority subset is oversampled with non-dominated members of GA population. The relation of domination uses two criteria. Assume majority objects $majC$, minority objects $minC$ fixed. The first criterion makes use of an approach suggested in [15] for oversampling strategies based on calculating real-valued potential of each instance. The potential is defined by a radial basis function based on a set of majority objects $majC$, minority objects $minC$ and parameter γ representing the spread of the function. For an instance x the potential is defined as:

$$\phi(x, majC, minC, \gamma) = \sum_{y \in majC} \exp^{-(\frac{dist(x,y)}{\gamma})^2} - \sum_{y \in minC} \exp^{-(\frac{dist(x,y)}{\gamma})^2}$$

For any two instances x, y we write:

$$x \prec_1 y \iff \phi(x, majC, minC, \gamma) < \phi(y, majC, minC, \gamma)$$

The second criterion makes use of an average distance of an instance to 25% of nearest neighbors from the majority instances. For a fixed instance x and fixed majority dataset $majC$, let $\{x_1, \ldots, x_n\}$ stand for the 25% of nearest neighbors from $majC$. Define:

$$distMajority(x, majC) = \sum_{i=1}^{n} dist(x, x_i)/n$$

$$x \prec_2 y \iff distMajority(x, majC) < distMajority(y, majC)$$

Finally, x dominates y iff

$$x \prec y \iff x \prec_1 y \ \& \ x \prec_2 y$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

The genetic algorithm starts with random population of instances and fitness defined as level of domination. After each iteration members with lowest fitness are merged into the oversampled minority set. Details are in [13].

Agent CSMOUTE is balancing using the algorithm introduced in [14]. For oversampling SMOTE is used, and undersampling is performed in the following steps: for randomly selected majority instance x and its random k-neighbor z, both are deleted from majority set and the new interpolated instance $x+\lambda\cdot(z-x)$ is introduced; this procedure is repeated to reach the proper size of the majority subset.

3 Computational experiment

3.1 Experiment plan

Experiment involved multiple class imbalanced datasets from the Keel Dataset Repository as shown in Table 1.

Table 1. Datasets used in the reported experiment. source [3]

Dataset	#Attr.	#Inst.	#Clas.	IR	Dataset	#Attr.	#Inst.	#Clas.	IR
Balance	4	625	3	5.88	New Thyroid	5	215	3	4.84
Contraceptive	9	1473	3	1.89	Pageblocks	10	548	5	164
Dermatology	34	336	6	5.55	Penbased	15	1100	10	1.95
Ecoli	7	336	8	71.50	Shuttle	9	2175	5	853
Glass	9	214	6	8.44	Thyroid	21	720	3	36.94
Hayes-Roth	4	132	3	1.70	Wine	13	178	3	1.50
Lymphography	18	148	4	40.50	Yeast	8	1484	10	23.15

Each of the discussed balancing agents has been used in the experiment to produce synthetic minority examples followed by applying the binary GEP classifier under the OVO scheme to obtain the confusion matrix from which values of the performance measures have been calculated using formula from 3.2, that is geometric mean (Gmean), mean of recall values (M.Rec.) shown as (1), accuracy (Acc.), index kappa (Kappa) shown as (2) and area under the roc curve (MAUC) shown as (3). To obtain the average values we used 5-CV scheme repeated 6 times.

Gene Expression Programming (GEP) technique, introduced by [8] combines the idea of genetic algorithms and genetic programming and makes use of a population of genes. Each gene is a linear structure divided into two parts. The first part, the head, contains functions and terminals while the second part, the tail, contains only terminals. For this study terminals are of type $(oper; attr; const)$, where the value of $const$ is in the range of attribute $attr$ and $oper$ is a relational operator from $\{<, \leq, >, \geq, =, \neq\}$. Functions are from the set $\{AND, OR, NOT, XOR, NOR\}$. For a fixed instance x from the dataset, the

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

value $g(x)$ of a gene g is boolean and thus a gene can be treated as a binary classifier.

In the reported experiment, for all considered balancing agents and datasets, the GEP classifier has been used with the following parameter value settings: population size – 100; number of iterations – 200; probabilities of mutation, RIS transposition, IS transposition, 1-point and 2-point recombination – 0.5, 0.2, 0.2, 0.2, 0.2, respectively. For selection the roulette wheel method has been used. Parameter values for oversampling agent algorithms have been set as in original papers describing implementation for the binary classification task.

3.2 Performance measures

To define classifier performance measures used in the experiments, assume that the dataset contains data from k classes. The elements of confusion matrix $C = \{c_{ij} : i, j \leq k\}$, where c_{ij} describes the number of instances that were predicted as class i but belonged to class j, allow to define for each class $m \leq k$:

- $TP_m = c_{mm}$ - the number of true positives (examples of class m which were classified correctly),
- $FP_m = \sum_{i=1,i\neq m}^{k} c_{mi}$ - the number of false positives (examples that were wrongly assigned to class m),
- $TN_m = \sum_{i=1,i\neq m}^{k} \sum_{j=1,j\neq m}^{k} c_{ij}$ - the number of true negative predictions regarding class m,
- $FN_m = \sum_{i=1,i\neq m}^{k} c_{im}$ - the number of false negatives for class m.

The Precision and Recall for class m are defined as:

$$Precision_m = \frac{TP_m}{TP_m + FP_m},\ Recall_m = \frac{TP_m}{TP_m + FN_m}$$

and used for the measure $Gmean$ and average accuracy which is the arithmetic mean of recall values of all the classes $Mean - recall$:

$$Gmean = (\prod_{i=1}^{k} Recall_i)^{\frac{1}{k}},\ M.Rec = \frac{1}{k}\sum_{i=1}^{k} Recall_i \tag{1}$$

As noted in [22], for multiclass classification $Kappa$ measure is less sensitive to class distribution than $Accuracy$:

$$Kappa = \frac{n\sum_{i=1}^{k} TP_i - ABC}{n^2 - ABC},\ Accuracy = \frac{1}{n}\sum_{i=1}^{k} TP_i \tag{2}$$

where n is the size of the dataset and $ABC = \sum_{i=1}^{k}(TP_i + FP_i)(TP_i + FN_i)$. The average AUC (area under the ROC curve) is defined as:

$$MAUC = \frac{1}{k(k-1)}\sum_{i\neq j}^{k} AUC(i,j) \tag{3}$$

where $AUC(i,j)$ is the area under the curve for the pair of classes i and j.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

3.3 Experiment results

In Table 2 computational experiment results averaged over 30 runs produced using 5 cross-validation scheme for 6 times, and further on, averaged over 14 considered datasets, are shown. In Figures 1 – 5 the above results are shown in the form of Box & Whiskers plots.

Table 2. Average values of performance measures obtained in the experiment.

Performance measure	DominM	CsmouteM	FWSmoteM	AdasynM	LamoM
Accuracy	**0.559**	0.362	0.548	0.522	0.355
Kappa Index	**0.393**	0.257	0.345	0.362	0.190
Balanced recall	**0.607**	0.520	0.555	0.588	0.431
MAUC	0.345	0.315	0.329	**0.352**	0.258
Gmean	**0.584**	0.419	0.483	0.548	0.410

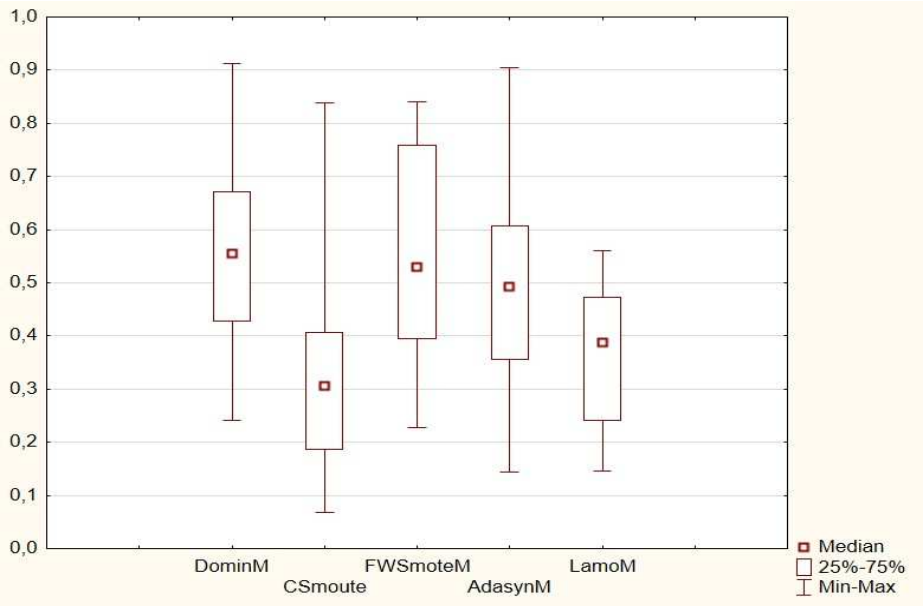

Fig. 1. Box & Whisker plot of average accuracies obtained in the reported experiment.

To evaluate the results shown in Table 2 and Figures 1–5 we have performed the Friedman ANOVA by ranks test for each of the considered performance measures. The null hypothesis for the procedure is that the different agents produced statistically similar results i.e. produced samples drawn from the same population, or specifically, populations with identical medians. As it is shown

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

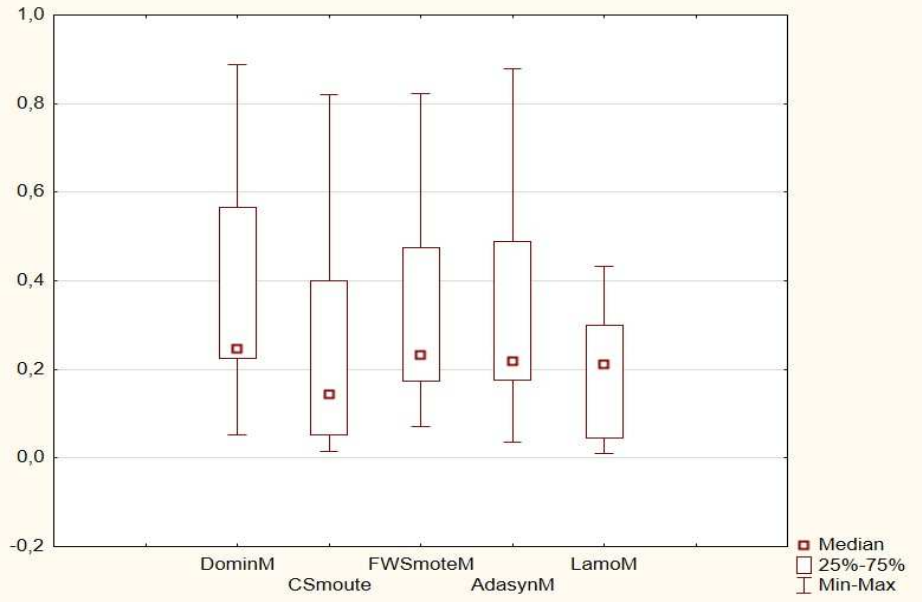

Fig. 2. Box & Whisker plot of average kappa indexes obtained in the reported experiment.

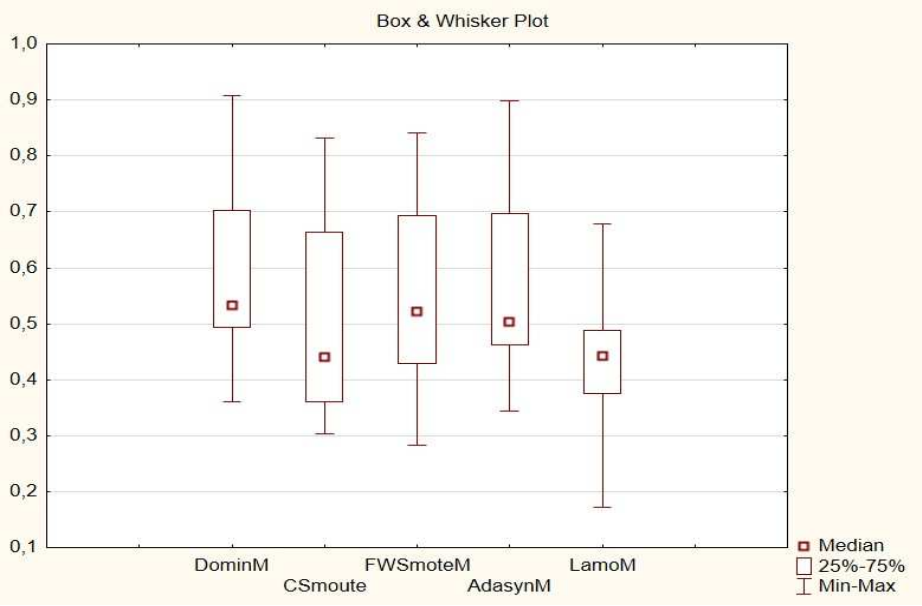

Fig. 3. Box & Whisker plot of average balanced recall values obtained in the reported experiment.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

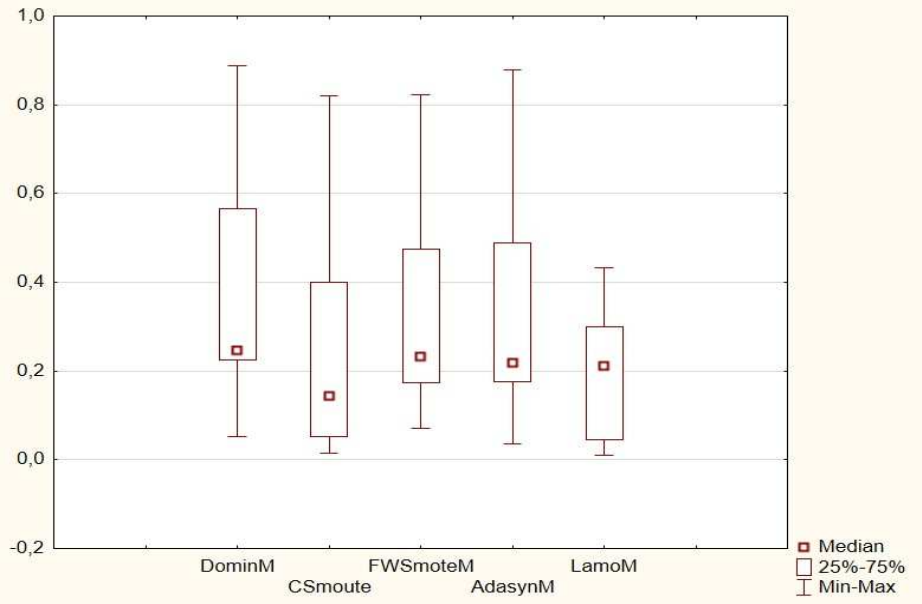

Fig. 4. Box & Whisker plot of average Kappa indexes obtained in the reported experiment.

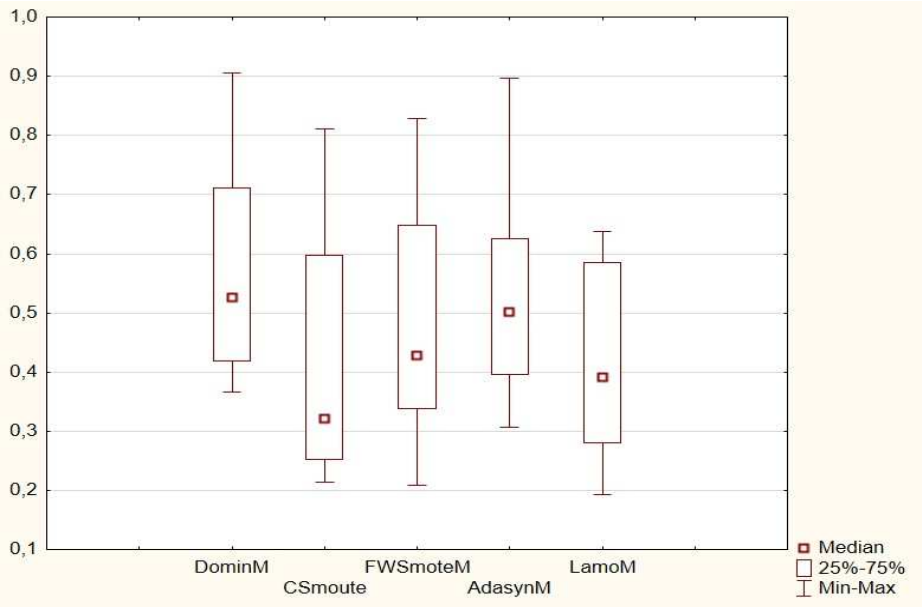

Fig. 5. Box & Whisker plot of average Gmean values obtained in the reported experiment.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

in Table 3 summarizing the above test results, the null hypothesis for results measured using each of the considered performance measure, should be rejected at the significance level of 0.05. The Kendall concordance coefficient calculated for results produced using each of performance measures shows a fair agreement in the rankings of the variables among cases. The above findings tell us that there are statistically significant differences in the performance of the considered agents.

Table 3. Summary of the Friedman ANOVA test results

Measure	Chi Sqr.	p-value	Conc. C.
Accuracy	32.58700	0.0000	0.58193
Kappa	36.17204	0.0000	0.64593
M. Rec.	34.42140	0.0000	0.54324
MAUC	31.63880	0.0000	0.52731
Gmean	28.97491	0.0001	0.51741

To gain better knowledge of the performance of the considered balancing agents we have carried out a series of pairwise comparisons using the Wilcoxon matched pairs tests. The null hypothesis in such case states that results produced by two different agents are drawn from samples with the same distribution. Test results are summarized in Table 4.

Table 4. Wilcoxon matched pair test results

Measure	Compared agents	T	Z	p-value
Accuracy	**DOMIN-M vs. FWSMOTE-M**	**38.00000**	**0.91026**	**0.36269**
Accuracy	DOMIN-M vs. ADASYN-M	5.00000	2.98188	0.00287
Accuracy	DOMIN-M vs. CSMOUTE-M	0.00000	3.29577	0.00098
Accuracy	DOMIN-M vs. LAMO-M	2.00000	3.17021	0.00152
Kappa	DOMIN-M vs. ADASYN-M	5.00000	2.98188	0.00287
Kappa	DOMIN-M vs. FWSMOTE-M	12.00000	2.54245	0.01101
Kappa	DOMIN-M vs. CSMOUTE-M	0.00000	3.29577	0.00098
Kappa	DOMIN-M vs. LAMO-M	0.00000	3.17980	0.00147
M.Rec	DOMIN-M vs. FWSMOTE-M	10.00000	2.83981	0.00451
M.Rec	DOMIN-M vs. ADASYN-M	6.00000	3.06699	0.00216
M.Rec	DOMIN-M vs. CSMOUTE-M	0.00000	3.40777	0.00066
M.Rec	DOMIN-M vs. LAMO-M	0.00000	3.29577	0.00098
MAUC	**DOMIN-M vs. ADASYN-M**	**40.00000**	**1.13592**	**0.25599**
MAUC	DOMIN-M vs. FWSMOTE-M	3.00000	3.10744	0.00189
MAUC	DOMIN-M vs. CSMOUTE-M	5.00000	2.98188	0.00287
MAUC	DOMIN-M vs. LAMO-M	0.00000	3.29577	0.00098
Gmean	**DOMIN-M vs. ADASYN-M**	**18.00000**	**1.92186**	**0.05462**
Gmean	DOMIN-M vs. FWSMOTE-M	8.00000	2.79355	0.00521
Gmean	DOMIN-M vs. CSMOUTE-M	0.00000	3.29577	0.00098
Gmean	DOMIN-M vs. LAMO-M	6.00000	2.91911	0.00351

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

Data from Table 4 allow drawing the following observations valid at the significance level of 0.05:

- For Accuracy measure, DOMIN-M and FWSMOTE perform statistically equally well.
- For the Accuracy measure, DOMIN-M outperforms statistically ADASYN-M, CSMOUTE-M, and LAMO-M.
- For the Kappa index measure, DOMIN-M outperforms statistically all the remaining agents.
- For the Mean-Recall measure, DOMIN-M outperforms statistically all the remaining agents.
- For the MAUC measure, DOMIN-M and ADASYN-M perform statistically equally well.
- For the MAUC measure, DOMIN-M outperforms statistically FWSMOTE-M, CSMOUTE-M, and LAMO-M.
- For the Gmean measure, DOMIN-M and ADASYN-M perform statistically equally well.
- For the Gmean measure, DOMIN-M outperforms statistically FWSMOTE-M, CSMOUTE-M, and LAMO-M.

4 Conclusions

The paper contributes by proposing a set of balancing agents able to deal with mining multiclass imbalanced datasets. The proposed agents are based on several state-of-the-art binary classifiers and use OVO (One-versus-One) strategy to deal with the multiclass problems. Agents can be used as stand-alone classifiers or serve as components (base classifiers) in ensembles of classifiers. The goal of the paper was to evaluate the performance of the considered agents when mining multiclass imbalanced datasets. The computational experiment has shown that among considered approaches the DOMIN-M assures the best performance no matter which performance measure was used. DOMIN-M agent performance is closely followed by that of the ADASYN-M and FWSMOTE-M agents. The above three agents should be considered as a promising option when looking for a tool for mining multiclass imbalanced datasets.

Future research will concentrate on the experimental study of the proposed agent's performance in different ensemble architectures including boosting, bagging, and stacking. Another promising direction of research would concentrate on undersampling part of balancing strategies. Using more advanced techniques like PCA or metaheuristic based techniques could be advantageous.

References

1. Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of over-sampling and boosting techniques. Soft Comput. **19**(12), 3369–3385 (2015)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

2. Agrawal, A., Viktor, H.L., Paquet, E.: Scut: Multi-class imbalanced data classification using smote and cluster-based undersampling. In: 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K). vol. 01, pp. 226–234 (2015)
3. 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.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput. **13**(3), 307–318 (2009)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
5. Díaz-Vico, D., Figueiras-Vidal, A.R., Dorronsoro, J.R.: Deep mlps for imbalanced classification. In: 2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018. pp. 1–7. IEEE (2018)
6. Fernandes, E.R.Q., de Carvalho, A.C.P.L.F., Yao, X.: Ensemble of classifiers based on multiobjective genetic sampling for imbalanced data. IEEE Trans. Knowl. Data Eng. **32**(6), 1104–1115 (2020)
7. Fernández, A., del Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. Int. J. Approx. Reason. **50**(3), 561–577 (2009)
8. Ferreira, C.: Gene expression programming: A new adaptive algorithm for solving problems. Complex Systems **13**(2) (2001)
9. Haixiang, G., Yijing, L., Yanan, L., Xiao, L., Jinling, L.: Bpso-adaboost-knn ensemble learning algorithm for multi-class imbalanced data classification. Engineering Applications of Artificial Intelligence **49**, 176–193 (2016)
10. Hastie, T.J., Rosset, S., Zhu, J., Zou, H.: Multi-class adaboost. Statistics and Its Interface **2**, 349–360 (2009)
11. He, H., Bai, Y., Garcia, E.A., Li, S.: 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 (2008)
12. Hoens, T.R., Qian, Q., Chawla, N.V., Zhou, Z.: Building decision trees for the multi-class imbalance problem. In: Tan, P., Chawla, S., Ho, C.K., Bailey, J. (eds.) Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Proceedings, Part I. Lecture Notes in Computer Science, vol. 7301, pp. 122–134. Springer (2012)
13. Jedrzejowicz, J., Jedrzejowicz, P.: Bicriteria oversampling for imbalanced data classification. In: Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 26th International Conference KES-2022. Procedia Computer Science, vol. 207C, pp. 239–248. Elsevier (2022)
14. Koziarski, M.: CSMOUTE: combined synthetic oversampling and undersampling technique for imbalanced data classification. In: International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021. pp. 1–8. IEEE (2021)
15. Koziarski, M.: Potential anchoring for imbalanced data classification. Pattern Recognit. **120**, 108114 (2021)
16. Koziarski, M., Krawczyk, B., Wozniak, M.: Radial-based oversampling for noisy imbalanced data classification. Neurocomputing **343**, 19–33 (2019)
17. Krawczyk, B.: Cost-sensitive one-vs-one ensemble for multi-class imbalanced data. In: 2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016. pp. 2447–2452. IEEE (2016)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1

18. Li, Q., Song, Y., Zhang, J., Sheng, V.S.: Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering. Expert Syst. Appl. **147**, 113152 (2020)
19. Lin, M., Tang, K., Yao, X.: Dynamic sampling approach to training neural networks for multiclass imbalance classification. IEEE Trans. Neural Networks Learn. Syst. **24**(4), 647–660 (2013)
20. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **39**(2), 539–550 (2009)
21. Maldonado, S., Vairetti, C., Fernández, A., Herrera, F.: FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification. Pattern Recognit. **124**, 108511 (2022)
22. Rodríguez, J.J., Díez-Pastor, J., Arnaiz-González, Á., Kuncheva, L.I.: Random balance ensembles for multiclass imbalance learning. Knowl. Based Syst. **193**, 105434 (2020)
23. Sáez, J.A., Krawczyk, B., Wozniak, M.: Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. Pattern Recognit. **57**, 164–178 (2016)
24. Wang, S., Yao, X.: Multiclass imbalance problems: Analysis and potential solutions. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **42**(4), 1119–1130 (2012)
25. Wang, X., Xu, J., Zeng, T., Jing, L.: Local distribution-based adaptive minority oversampling for imbalanced data classification. Neurocomputing **422**, 200–213 (2021)
26. Yijing, L., Haixiang, G., Xiao, L., Yanan, L., Jinling, L.: Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. Knowledge-Based Systems **94**, 88–104 (2016)
27. Zhang, Z.L., Luo, X.G., García, S., Herrera, F.: Cost-sensitive back-propagation neural networks with binarization techniques in addressing multi-class problems and non-competent classifiers. Applied Soft Computing **56**, 357–367 (2017)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_1