Performance Analysis of Binarization Strategies for Multi-class Imbalanced Data Classification

Michał Żak(✉) and Michał Woźniak

Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
{michal.zak,michal.wozniak}@pwr.edu.pl

Abstract. Multi-class imbalanced classification tasks are characterized by the skewed distribution of examples among the classes and, usually, strong overlapping between class regions in the feature space. Furthermore, frequently the goal of the final system is to obtain very high precision for each of the concepts. All of these factors contribute to the complexity of the task and increase the difficulty of building a quality data model by learning algorithms. One of the ways of addressing these challenges are so-called binarization strategies, which allow for decomposition of the multi-class problem into several binary tasks with lower complexity. Because of the different decomposition schemes used by each of those methods, some of them are considered to be better suited for handling imbalanced data than the others. In this study, we focus on the well-known binary approaches, namely One-Vs-All, One-Vs-One, and Error-Correcting Output Codes, and their effectiveness in multi-class imbalanced data classification, with respect to the base classifiers and various aggregation schemes for each of the strategies. We compare the performance of these approaches and try to boost the performance of seemingly weaker methods by sampling algorithms. The detailed comparative experimental study of the considered methods, supported by the statistical analysis, is presented. The results show the differences among various binarization strategies. We show how one can mitigate those differences using simple oversampling methods.

Keywords: Multi-class classification · Imbalanced data · Binarization strategies

1 Introduction

The goal of the supervised learning is to build a data model capable of mapping inputs x to outputs y with a good generalization ability, given a labeled set of input-output pairs $\mathcal{D} = (x_i, y_i)_{i=1}^{N}$, $\mathcal{D}$ being the training set and N being the number of training examples. Usually, each of the training inputs x_i is a d-dimensional vector of numbers and nominal values, the so-called features that

V. V. Krzhizhanovskaya et al. (Eds.): ICCS 2020, LNCS 12140, pp. 141–155, 2020.
https://doi.org/10.1007/978-3-030-50423-6_11

characterize a given example, but x_i might as well be a complex structured object like an image, a time series or an email message. Similarly, the type of the output variable can in principle be anything, but in most cases it is of a continuous type $y_i \in \mathbb{R}$ or a nominal type $y_i \in \mathbb{C}$, where, considering an m class problem, $\mathbb{C} = \{c_1, ..., c_m\}$. In the former case, it is a regression problem, while in the latter, it is a classification problem [10,22]. Classification problems are very common in a real-world scenario and machine learning is widely used to solve these types of problems in areas such as fraud detection [6,24], image recognition [17,26], cancer treatment [3] or classification of DNA microarrays [19].

In many cases, classification tasks involve more than two classes forming so-called multi-class problems. This characteristic often imposes some difficulties on the machine learning algorithm, as some of the solutions were designed strictly for binary-class problems and may not be applicable to those kinds of scenarios. What is more, problems, where multiple classes are present, are often characterized by greater complexity than binary tasks, as the decision boundaries between classes tend to overlap, which might lead to building a poor quality model by a given classifier. Usually, it is simply easier to build a model to distinguish only between two classes than to consider a multi-class problem. One approach to overcome those challenges is to use binarization strategies that reduce the task to multiple binary classification subproblems - in theory, with lower complexity - that can be solved separately by dedicated models, the so-called base learners [2,11,13,14]. The most commonly used binarization strategies are One-Vs-All (OVA) [25], One-Vs-One (OVO) [12,16] and Error-Correcting Output Codes (ECOC) [9], which is a general framework for the binary decomposition of multi-class problems.

In this paper, we focus on the performance of the aforementioned binarization strategies in the context of multi-class imbalanced problems. We aim to determine whether there are statistically significant differences among the performances of those methods, provided the most suitable aggregation scheme for a given problem. If so - whether or not one can nullify those differences by improving the quality of base learners within each binarization method with sampling algorithms. The main contributions of this work are:

- an exhaustive experimental study on the classification of multi-class imbalanced data with the use of OVA, OVO and ECOC binarization strategies.
- a comparative study of the aforementioned approaches with regard to a number of base classifier and aggregation schemes for each of the them.
- a study on the performance of the binarization strategies with the sampling algorithms used to boost the quality of their base classifiers.

The rest of this paper is organized as follows. In Sect. 2, an overview of binarization strategies used in the experiments is given. In Sect. 3 the experimental framework set-up is presented, including the classification and sampling algorithms, performance measures and datasets used in the study. The empirical analysis of obtained results has been carried out in Sect. 4. In Sect. 5 we make our concluding remarks.

2 Decomposition Strategies for Multi-classification

The underlying idea behind binarization strategies is to undertake the multi-class problems using binary classifiers with divide and conquer strategy [13]. A transformation like this is often performed with the expectation that the resulting binary subproblems will have lower complexity than the original multi-class problem. One of the drawbacks of such approach is the necessity to combine the individual responses of the base learners into the final output of the decision system. What is more, building a dedicated model for each of the binary subproblems significantly increases the cost of building a decision system in comparison to undertaking the same problem with a single classifier. However, the magnitude of this problem varies greatly depending on the chosen binarization strategy as well as the number of classes under consideration and the size of the training set itself. In this study, we focus on the most common binarization strategies: OVA, OVO, and ECOC.

2.1 One-Vs-All Strategy

OVA binarization strategy divides an m-class problem into m binary problems. In this strategy, m binary classifiers are trained, each responsible for distinguishing instances of a given class from the others. During the validation phase, the test pattern is presented to each of the binary models and the model that gives a positive output indicates the output class of the decision system. This approach can potentially result in ambiguously labeled regions of the input space. Usually, some tie-breaking techniques are required [13,22].

While relatively simple, OVA binarization strategy is often preferred to more complex methods, provided that the best available binary classifiers are used as the base learners [25]. However, in this strategy, the whole training set is used to train each of the base learners. It dramatically increases the cost of building a decision system with respect to the single multi-class classifier. Another issue is that each of the binary subproblems is likely to suffer from the aforementioned class imbalance problem [13,22].

2.2 One-Vs-One Strategy

OVA binarization strategy divides an m-class problem into $\frac{m \times (m-1)}{2}$ binary problems. In this strategy, each binary classifier is responsible for distinguishing instances of different pair of classes (c_i, c_j). The training set for each of the binary classifiers consists only of instances of the two classes forming a given pair, while the instances of the remaining classes are discarded. During the validation phase, the test pattern is presented to each of the binary models. The output of a model given by $r_{ij} \in [0, 1]$ is the confidence of the binary classifier discriminating classes i and j in favour of the former class. If the classifier does not provide it, the confidence for the latter class is computed by $r_{ji} = 1 - r_{ij}$ [12,13,22,29]. The class with the higher confidence value is considered as the output class of

the decision system. Similarly to OVA strategy - this approach can also result in ambiguities [22].

Although the number of base learners in this strategy is of m^2 order, the growth in the number of learning tasks is compensated by the learning set reduction for each of the individual problems, as demonstrated in [12]. Also, one has to keep in mind that in this method, each of the base classifiers is trained using only the instances of two classes, deeming their output not significant for the instances of all the remaining classes. Usually, the assumption is that the base learner will make a correct prediction within its domain of expertise [13].

2.3 Error-Correcting Output Codes Strategy

ECOC binarization strategy is a general framework for the binary decomposition of multi-class problems. In this strategy, each class is assigned a unique binary string of length n, called *code word*. Next, n binary classifiers are trained, one for each bit in the string. During the training phase on an example from class i, the desired output of a given classifier is specified by the corresponding bit in the code word for this class. This process can be visualized by a $m \times n$ binary code matrix. As an example, Table 1 shows a 15-bit error-correcting output code for a five-class problem, constructed using exhaustive technique [9]. During the validation phase, the test pattern is presented to each of the binary models. Then the binary code word is formed from their responses. The class which code word was the nearest to the code word formed from the base learners' responses, according to the Hamming distance, indicates the output class of the decision system.

Table 1. A 15-bit error-correcting output code for a five class problem.

Class	Code word														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

In contrast to OVA and OVO strategies, ECOC method does not have a predefined number of binary models that will be used to solve a given multi-class problem. This number is determined purely by the algorithm one chooses to generate the ECOC code matrix. A measure of the quality of error-correcting code is the minimum Hamming distance between any pair of code words. If the minimum Hamming distance is l, then the code can correct at least $\frac{l-1}{2}$ single-bit errors.

2.4 Aggregation Schemes for Binarization Techniques

For the binarization techniques mentioned above, an aggregation method is necessary to combine the responses of an ensemble of base learners. In the case of ECOC binarization strategy, this aggregation method is embedded in it. An exhaustive comparison study has been carried out in [13], including various aggregation methods for both OVA and OVO binarization strategies. For our experimental study, the implementations of the following methods for OVA and OVO decomposition schemes have been used:

- OVA
  1. *Maximum Confidence Strategy*;
  2. *Dynamically Ordered One-Vs-All*.
- OVO
  1. *Voting Strategy*;
  2. *Weighted Voting Strategy*;
  3. *Learning Valued Preference for Classification*;
  4. *Decision Directed Acyclic Graph*

For ECOC strategy, the exhaustive codes were used to generate the code matrix if the number of classes m in the problem under consideration satisfied $3 \leq m \leq 7$. In other cases, the random codes were used as implemented in [23].

3 Experimental Framework

In this section, the set-up of the experimental framework used for the study is presented. The classification and sampling algorithms used to carry out the experiments are described in Sect. 3.1. Next, the performance measure used to evaluate the built models is presented in Sect. 3.2. Section 3.3 covers the statistical tests used to compare the obtained results. Finally, Sect. 3.4 describes the benchmark datasets used in the experiments.

3.1 Classification Used for the Study

One of the goals of the empirical study was to ensure the diversity of the classifiers used as base learners for binarization strategies. A brief description of the used algorithms is given in the remainder of this section.

- *Naïve Bayes* [22] is a simple model that assumes the features are conditionally independent given the class label. In practice, even if Naïve Bayes assumption is not true, it often performs fairly well.
- *k-Nearest Neighbors (k-NN)* [22] is a non-parametric classifier that simply uses chosen distance metric to find k points in the training set that are nearest to the test input x, and returns the most common class among those points as the estimate.

- *Classification and Regression Tree (CART)* [22] models are defined by recursively partitioning the input space, and defining a local model in each resulting region of input space.
- *Support Vector Machines (SVM)* [27] maps the original input space into a high-dimensional feature space via so-called kernel trick. In the new feature space, the optimal separating hyperplane with maximal margin is determined in order to minimize an upper bound of the expected risk instead of the empirical risk.
- *Logistic Regression* [22] is the generalization of the linear regression to the (binary) classification, so called Binomial Logistic Regression. Further generalization to Multi-Class Logistic Regression is often achieved via OVA approach.

During the building phase, for each of aforementioned base classifiers an exhaustive search over specified hyperparameter values was performed in attempt to build the best possible data model for a given problem - the values of hyperparameters used in the experiments are shown in Table 2. Furthermore, various sampling methods were used to boost the performance of base learners, namely SMOTE [7], Borderline SMOTE [15], SMOTEENN [4] and SMOTETomek [5]. All of the experiments were conducted using the Python programming language and libraries from the SciPy ecosystem (statistical tests and data manipulation) as well as scikit-learn (classifier implementations and feature engineering) and imbalanced-learn (sampling algorithms implementations) [18,23,28].

Table 2. Hyperparameter specification for the base learners used in the experiments.

Algorithm	Hyperparameters
Naive Bayes	—
k-Nearest Neighbors	$k \in \{1, 3, 5\}$
	Distance metric = Minkowski metric
CART	Split criterion $\in$ {Gini Impurity, Information Gain}
	Maximum depth = $(3, 11)$
	Minimum leaf samples $\in \{1, 3, 5\}$
SVM	Kernel type $\in$ {RBF, Linear}
	Regularization parameter $\in$ {0.001, 0.01, 0.1, 1}
	Kernel coefficient $\in$ {0.0001, 0.001, 0.01, 0.1, 1}
Logistic Regression	Regularization parameter $\in$ {0.001, 0.01, 0.1, 1}
	Penalty $\in \{l1, l2\}$

3.2 Performance Measures

Model evaluation is a crucial part of an experimental study, even more so when dealing with imbalanced problems. In the presence of imbalance, evaluation metrics that focus on overall performance, such as overall accuracy, have a tendency to ignore minority classes because as a group they do not contribute much to the general performance of the system. To our knowledge, at the moment there is no consensus as to which metric should be used in imbalance data scenarios, although several solutions have been suggested [20,21]. Our goal was to pick a robust metric that ensures reliable evaluation of the decision system in the presence of strong class imbalance and at the same time is capable of handling multi-classification problems. Geometric Mean Score (G-Mean) is proven metric that meets both of these conditions - it focuses only on recall of each class and aggregates it multiplicatively across each class:

$$G - Mean = (\prod_{i=1}^{m} r_i)^{1/m}, \tag{1}$$

where r_i represents recall for $i - th$ class and m represents number of classes.

3.3 Statistical Tests

The non-parametric tests were used to provide statistical support for the analysis of the results, as suggested in [8]. Specifically, the Wilcoxon Signed-Ranks Test was applied as a non-parametric statistical procedure for pairwise comparisons. Furthermore, the Friedman Test was used to check for statistically significant differences between all of the binarization strategies, while the Nemenyi Test was used for posthoc comparisons and to obtain and visualize critical differences between models. The fixed significance level $\alpha = 0.05$ was used for all comparisons.

3.4 Datasets

The benchmark datasets used to conduct the research were obtained from the KEEL dataset repository [1]. The set of benchmark datasets was specially selected to ensure the robustness of the study and includes data with a varying numbers of instances, number and type of class attributes and the imbalance ratio of classes. The characteristics of the datasets used in the experiments are shown in Table 3 - for each dataset, it includes the number of instances

(#Inst.), the number of attributes (#Atts.), the number of real, integer and nominal attributes (respectively #Real., #Int., and #Nom.), the number of classes (#Cl.) and the distribution of classes (#Dc.). All numerical features were normalized, and categorical attributes were encoded using the so-called *one-hot encoding*.

4 Experimental Study

In this section, the results of the experimental study are presented. Table 4 shows the results for the best variant of each binarization strategy for the benchmark datasets without internal sampling. As we can see, in this case the OVO strategy outperformed the other two methods. Friedman Test returned $p - Value = p = 0.008$, pointing to a statistically significant difference between the results of those methods. However, Nemenyi Test revealed only the statistically significant difference between OVO and ECOC methods. Results obtained for each binarization strategy and critical differences for posthoc tests are visualized respectively in Fig. 1 and Fig. 2.

Table 3. Summary description of the datasets.

Dataset	#Inst.	#Atts.	#Real.	#Int.	#Nom.	#Cl.	#Dc.
Automobile	159	25	15	0	10	6	48/46/29/20/13/3
Balance	625	4	4	0	0	3	288/288/49
Car	1728	6	0	0	6	4	1210/384/69/65
Cleveland	297	13	13	0	0	5	160/54/35/35/13
Contraceptive	1473	9	6	0	3	3	629/511/333
Dermatology	358	34	0	34	0	6	111/71/60/48/48/20
Ecoli	336	7	7	0	0	8	143/77/52/35/20/5/2/2
Flare	1066	11	0	0	11	6	331/239/211/147/95/43
Glass	214	9	9	0	0	6	76/70/29/17/13/9
Hayes_roth	160	4	0	4	0	3	65/64/31
Led7digit	500	7	7	0	0	10	57/57/53/52/52/51/49/47/45/37
Lymphography	148	18	3	0	15	4	81/61/4/2
New_thyroid	215	5	4	1	0	3	150/35/30
Pageblocks	548	10	10	0	0	5	492/33/12/8/3
Thyroid	720	21	6	0	15	3	666/37/17
Vehicle	846	18	0	18	0	4	218/217/212/199
Wine	178	13	13	0	0	3	71/59/48
Winequality_red	1599	11	11	0	0	6	681/638/199/53/18/10
Yeast	1484	8	8	0	0	10	463/429/244/163/51/44/35/30/20/5
Zoo	101	16	0	0	16	7	41/20/13/10/8/5/4

Table 4. G-mean results for tested binarization strategies without sampling.

Dataset	OVA		OVO		ECOC	
	G-mean	Rank	G-mean	Rank	G-mean	Rank
Automobile	0.51 ± 0.17	3	0.57 ± 0.17	2	0.58 ± 0.17	1
Balance	0.91 ± 0.02	3	0.94 ± 0.02	2	0.95 ± 0.02	1
Car	0.92 ± 0.02	2	0.94 ± 0.02	1	0.81 ± 0.05	3
Cleveland	0.20 ± 0.06	1	0.17 ± 0.08	2	0.14 ± 0.06	3
Contraceptive	0.53 ± 0.02	1	0.50 ± 0.02	2	0.49 ± 0.01	3
Dermatology	0.97 ± 0.01	1.5	0.96 ± 0.01	3	0.97 ± 0.01	1.5
Ecoli	0.25 ± 0.01	2	0.25 ± 0.01	2	0.25 ± 0.01	2
Flare	0.47 ± 0.04	1	0.46 ± 0.08	2	0.41 ± 0.08	3
Glass	0.51 ± 0.15	2	0.55 ± 0.10	1	0.44 ± 0.11	3
Hayes_roth	0.83 ± 0.02	1.5	0.83 ± 0.03	1.5	0.74 ± 0.08	3
Led7digit	0.72 ± 0.02	2	0.75 ± 0.01	1	0.71 ± 0.02	3
Lymphography	0.67 ± 0.14	1	0.57 ± 0.26	2	0.38 ± 0.23	3
New_thyroid	0.94 ± 0.02	1.5	0.94 ± 0.02	1.5	0.90 ± 0.05	3
Pageblocks	0.50 ± 0.23	3	0.57 ± 0.21	1	0.54 ± 0.25	2
Thyroid	0.88 ± 0.07	3	0.90 ± 0.07	2	0.92 ± 0.05	1
Vehicle	0.80 ± 0.02	2	0.81 ± 0.03	1	0.77 ± 0.03	3
Wine	0.99 ± 0.01	1	0.98 ± 0.01	2	0.97 ± 0.01	3
Winequality_red	0.18 ± 0.06	1.5	0.18 ± 0.06	1.5	0.10 ± 0.03	3
Yeast	0.40 ± 0.04	1.5	0.40 ± 0.04	1.5	0.37 ± 0.05	3
Zoo	0.84 ± 0.13	1.5	0.84 ± 0.12	1.5	0.79 ± 0.19	3
Avg. rank	—	1.8	—	**1.675**	—	2.525

Table 5 shows results for binarization strategies after enhancing the performance of base learners with sampling methods. Although the results are visibly better than they were obtained using pure binarization schemes, the hierarchy seems to be preserved with OVO outperforming the other two techniques, which is confirmed by the Friedman Test returning $p - Value = p = 0.006$ pointing to statistically significant difference and Nemenyi Test revealing only statistically significant difference between OVO and ECOC strategies. Those results seem to be consistent with the study carried out in [11], which points out that OVO app-

roach confronts a lower subset of instances and, therefore, is less likely to obtain a highly imbalanced training sets during binarization. Results obtained for each binarization strategy with the usage of internal sampling algorithms and critical differences for posthoc tests are visualized respectively in Fig. 3 and Fig. 4.

Wilcoxon Signed-Ranks Test was performed to determine whether or not there is a statistically significant difference between each strategy pure variant and variant enhanced with sampling algorithms. As shown in Table 6, in every case, the usage of sampling algorithms to internally enhance base models significantly improved the overall performance of the binarization strategy.

Table 5. G-mean results for tested binarization strategies with sampling.

Dataset	OVA*sampling*		OVO*sampling*		ECOC*sampling*	
	G-mean	Rank	G-mean	Rank	G-mean	Rank
Automobile	0.56 ± 0.22	3	0.61 ± 0.19	1	0.60 ± 0.20	2
Balance	0.88 ± 0.08	3	0.61 ± 0.19	1	0.92 ± 0.02	2
Car	0.91 ± 0.02	2	0.93 ± 0.02	1	0.82 ± 0.07	3
Cleveland	0.24 ± 0.06	2	0.25 ± 0.05	1	0.18 ± 0.06	3
Contraceptive	0.53 ± 0.02	1	0.52 ± 0.03	2	0.48 ± 0.02	3
Dermatology	0.96 ± 0.01	2.5	0.96 ± 0.01	2.5	0.97 ± 0.02	1
Ecoli	0.26 ± 0.01	2	0.26 ± 0.01	2	0.26 ± 0.01	2
Flare	0.56 ± 0.03	2	0.57 ± 0.03	1	0.52 ± 0.03	3
Glass	0.65 ± 0.07	1	0.62 ± 0.05	3	0.64 ± 0.06	2
Hayes_roth	0.84 ± 0.04	1	0.83 ± 0.04	2	0.68 ± 0.07	3
Led7digit	0.72 ± 0.02	2.5	0.74 ± 0.01	1	0.72 ± 0.02	2.5
Lymphography	0.66 ± 0.20	1	0.58 ± 0.26	2	0.55 ± 0.33	3
New_thyroid	0.94 ± 0.02	1.5	0.94 ± 0.04	1.5	0.92 ± 0.06	3
Pageblocks	0.57 ± 0.26	3	0.63 ± 0.20	1	0.59 ± 0.16	2
Thyroid	0.90 ± 0.06	3	0.92 ± 0.06	2	0.95 ± 0.05	1
Vehicle	0.81 ± 0.02	1.5	0.81 ± 0.02	1.5	0.80 ± 0.02	3
Wine	0.98 ± 0.01	1.5	0.98 ± 0.01	1.5	0.97 ± 0.01	3
Winequality_red	0.36 ± 0.08	1	0.33 ± 0.08	2	0.14 ± 0.05	3
Yeast	0.50 ± 0.03	2	0.51 ± 0.03	1	0.41 ± 0.05	3
Zoo	0.85 ± 0.13	1	0.84 ± 0.12	2	0.80 ± 0.16	3
Avg. rank	—	1.875	—	**1.6**	—	2.525

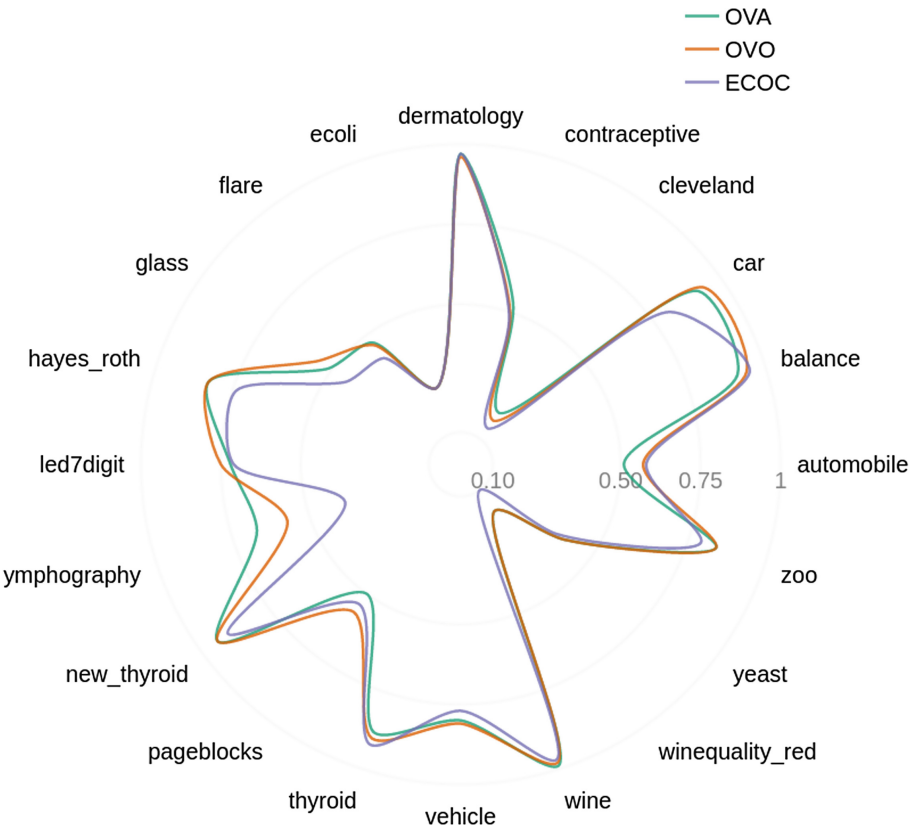

Fig. 1. G-mean results for tested binarization strategies without sampling.

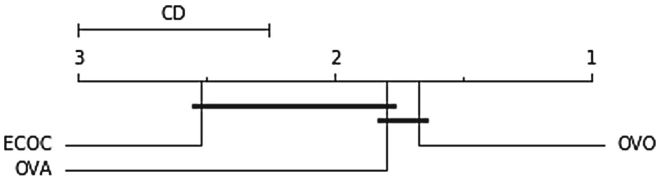

Fig. 2. Critical differences for Nemenyi Test for tested binarization strategies without sampling.

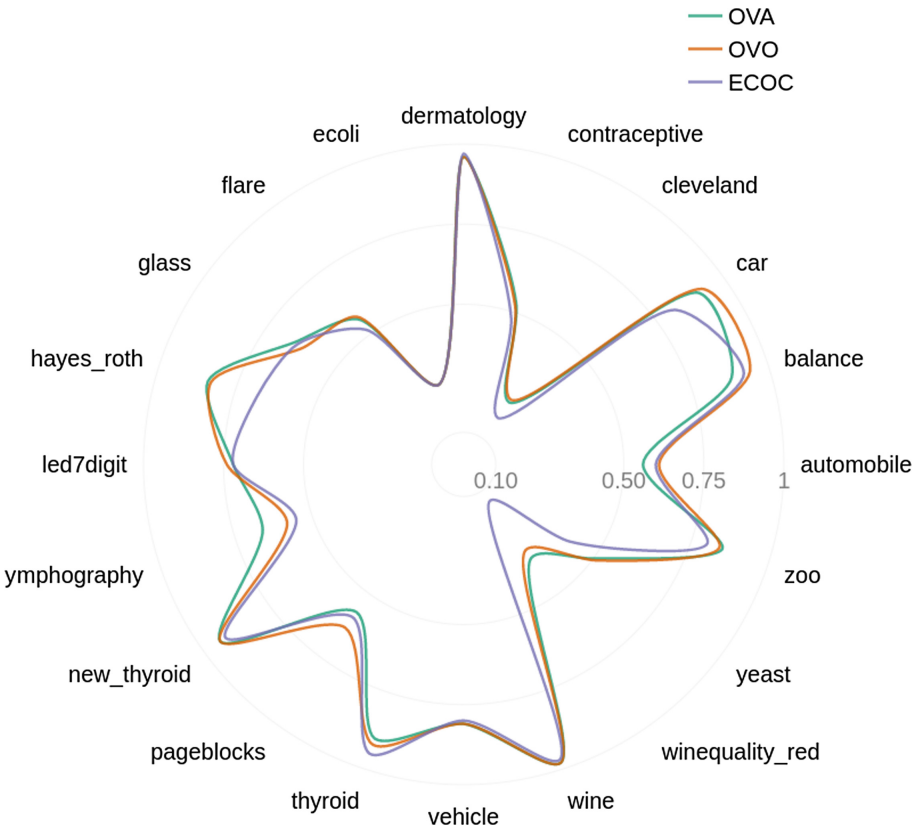

Fig. 3. G-mean results for tested binarization strategies with sampling.

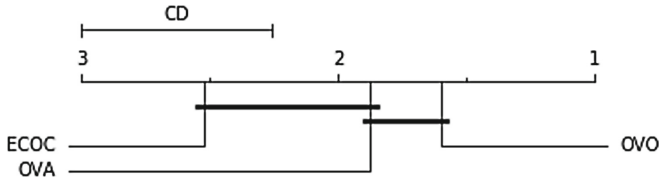

Fig. 4. Critical differences for Nemenyi Test for tested binarization strategies with sampling.

Table 6. Wilcoxon Signed-Ranks Test to compare binarization strategies variants with and without internal sampling. R^+ corresponds to the sum of the ranks for pure binarization strategy and R^- for variant with sampling.

Binarization strategy	R^+	R^-	Hypothesis ($\alpha = 0.05$)	p-value
OVA	43	161	Rejected for OVA *sampling*	0.02612
OVO	18	164	Rejected for OVO *sampling*	0.00518
ECOC	33	174	Rejected for ECOC *sampling*	0.00836

5 Concluding Remarks

In this paper, we carried out an extensive comparative experimental study of One-Vs-All, One-Vs-One, and Error-Correcting Output Codes binarization strategies in the context of imbalanced multi-classification problems. We have shown that one can reliably boost the performance of all of the binarization schemes with relatively simple sampling algorithms, which was then confirmed by a thorough statistical analysis. Another conclusion from this work is that the data preprocessing methods are able to partially mitigate the quality differences among different strategies, however the statistically significant difference among obtained results persists and OVO binarization seems to be the most robust of all three - this conclusion confirms the results of previous studies carried out in this field.

Acknowledgement. This work is supported by the Polish National Science Center under the Grant no. UMO-2015/19/B/ST6/01597 as well the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Science and Technology.

References

1. Alcalá-Fdez, J., et al.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Multiple-Valued Logic Soft Comput. **17**, 255–287 (2011)
2. Allwein, E., Schapire, R., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. J. Mach. Learn. Res. **1**, 113–141 (2000)
3. Anand, A., Suganthan, P.: Multiclass cancer classification by support vector machines with class-wise optimized genes and probability estimates. J. Theor. Biol. **259**(3), 533–540 (2009)
4. Batista, G., Bazzan, B., Monard, M.: Balancing training data for automated annotation of keywords: a case study. In: WOB, pp. 10–18 (2003)
5. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. **6**(1), 20–29 (2004)

6. Chan, P., Stolfo, S.: Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 164–168 (1998)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. arXiv e-prints arXiv:1106.1813 (2011)
8. Demšar, J.: Statistical comparisons ofclassifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
9. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. Artif. Intell. Res. **2**, 263–286 (1995)
10. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2000)
11. Fernández, A., López, V., Galar, M., Jesus, M.D., Herrera, F.: Analysing the classification of imbalanced data-sets with multiple classes: binarization techniques and ad-hoc approaches. Knowl.-Based Syst. **42**, 97–110 (2013)
12. Fürnkranz, J.: Round robin classification. J. Mach. Learn. Res. **2**, 721–747 (2002)
13. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes. Pattern Recogn. **44**(8), 1761–1776 (2011)
14. Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: Empowering difficult classes with a similarity-based aggregation in multi-class classification problems. Inf. Sci. **264**, 135–157 (2014)
15. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
16. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. Ann. Stat. **26**(2), 451–471 (1998)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv e-prints arXiv:1512.03385 (2015)
18. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(17), 1–5 (2017)
19. Liu, K., Xu, C.: A genetic programming-based approach to the classification of multiclass microarray datasets. Bioinformatics **25**(3), 331–337 (2009)
20. Luque, A., Carrasco, A., Martin, A., Heras, A.: The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recogn. **91**, 216–231 (2019)
21. Mosley, L.: A balanced approach to the multi-class imbalance problem. Graduate theses and dissertations, Iowa State University (2013)
22. Murphy, K.: Machine Learning: A Probabilistic Perspective. The MIT Press, Cambridge (2012)
23. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
24. Phua, C., Lee, V., Smith, K., Gayler, R.: A comprehensive survey of data mining-based fraud detection research. arXiv e-prints arXiv:1009.6119 (2010)

25. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. **5**, 101–141 (2004)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv e-prints arXiv:1409.1556 (2014)
27. Vapnik, V.: Statistical Learning Theory. Wiley-Interscience, Hoboken (1998)
28. Virtanen, P., et al.: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
29. Zhang, Z., Krawczyk, B., Garcìa, S., Rosales-Pérez, A., Herrera, F.: Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. Knowl.-Based Syst. **106**, 251–263 (2016)