Multicriteria Classifier Ensemble Learning for Imbalanced Data

WERONIKA WĘGIER, MICHAŁ KOZIARSKI, AND MICHAŁ WOŹNIAK

1Department of Systems and Computer Networks, Wrocław University of Science and Technology, 50-370 Wrocław, Poland
2Department of Electronics, AGH University of Science and Technology, 30-059 Kraków, Poland

Corresponding author: Weronika Węgier (weronika.wegier@pwr.edu.pl)

This work was supported by the Polish National Science Centre under the grant No. 2019/35/B/ST6/04442. The experimental research was supported in part by PLGrid Infrastructure.

ABSTRACT One of the vital problems with the imbalanced data classifier training is the definition of an optimization criterion. Typically, since the exact cost of misclassification of the individual classes is unknown, combined metrics and loss functions that roughly balance the cost for each class are used. However, this approach can lead to a loss of information, since different trade-offs between class misclassification rates can produce similar combined metric values. To address this issue, this paper discusses a multi-criteria ensemble training method for the imbalanced data. The proposed method jointly optimizes precision and recall, and provides the end-user with a set of Pareto optimal solutions, from which the final one can be chosen according to the user’s preference. The proposed approach was evaluated on a number of benchmark datasets and compared with the single-criterion approach (where the selected criterion was one of the chosen metrics). The results of the experiments confirmed the usefulness of the obtained method, which on the one hand guarantees good quality, i.e., not worse than the one obtained with the use of single-criterion optimization, and on the other hand, offers the user the opportunity to choose the solution that best meets their expectations regarding the trade-off between errors on the minority and the majority class.

INDEX TERMS Classifier ensemble, imbalanced data, multi-objective optimization, pattern classification.

I. INTRODUCTION

Data imbalance, that is the disproportion among the number of observations coming from different classes, is one of the open challenges in contemporary machine learning [1]. Most of the traditional pattern recognition algorithms are susceptible to presence of uneven class distribution, which then may lead to strong bias towards the majority class. At the same time, in various problems such as diagnosis of rare illness [2], [3], computer network attacks [4], [5] or bank frauds [6], [7], it is often the minority classes that are of particular interest and should be recognised correctly.

Intuitively, in the imbalanced data classification domain, we recognize that the cost of incorrect classification of each class might vary. It might be in particular different than what typical metrics and loss functions, such as the classification accuracy, would indicate [8]. This is because traditional metrics and losses penalize misclassification of each observation equally, resulting in an inverse relationship between the number of observations from a given class and their overall impact on the performance. To solve this problem, we should aim to reduce misclassification of minority class samples, but with caution, as not to attain the opposite - high bias against the majority class [9]. In an ideal setting, the class-specific misclassification cost would be known a priori, enabling us to formulate an appropriate loss function that considers the specified cost directly. In practice, however, this cost can be challenging to estimate [1], [9]. Some attempts have been made to obtain the loss function without a specialized problem related knowledge [10], [11], yet their results do not meet expectations. For that reason, we tend to choose optimization criteria and performance metrics that balance the misclassification cost of each class. In particular, we use aggregated metrics such as AUC, G-mean and balanced accuracy that try to penalize misclassification of observations depending on their class so that each class has a roughly equal...
impact on the performance [9], [12]. However, there are some problems with this approach. First of all, aggregated metrics suffer from the loss of information, as different values of their bases (e.g., true positive and true negative rate) may produce the same value of the combined metric [13]. Secondly, most of the aggregated measures are still biased toward majority class [13] or rely strongly on the classes’ ratio [14]. Furthermore, when a model is being optimized according to some aggregated metric, only general characteristics are improving and there is no way to choose specific aspects of the method that one would want to enhance. A partial solution to this problem would be employing a parametric metric, such as $F_\beta$-measure, where one can control the importance of each of the bases. Nevertheless, it is still required to set a relation between simple metrics, which is very heavily conditioned by the problem [14].

The aforementioned issues are why we would like to propose a method independent of metric or set metrics’ combination choice, which would give a variety of solutions to select from after the learning process. Our method employs multi-objective optimization (MOO) that aims to improve all the given objectives as simple metrics, which as a result enables flexibility not possible in single-objective optimization (SOO) using a set combination of criteria. It also appears that they can achieve better results according to the same objectives than SOO algorithms [15]. In this paper, we examine the possibility of utilizing MOO algorithm to build an ensemble of classifiers directly optimizing both precision and recall. Because very often objectives are contradictory or cannot be improved simultaneously based on the nature of the problem, MOO algorithms return a set of solutions for which no objective can be improved without sacrificing at least one other objective - the so-called Pareto front. This, in turn, allows the user to examine the obtained solutions and choose the one that suits their need the most.

The main contributions of this work are as follows:

1) formulation of the learning from imbalanced data as a multi-objective optimization task,
2) proposition of employing multi-objective optimization to train a weighted voting rule for classifier ensemble,
3) proposition of using an ensemble of classifiers model for classification of imbalanced data, where the choice of a particular ensemble can be done manually according to user’s preference or automatically according to predefined selection rule,
4) assessment of the quality of the developed model in comparison with the standard single-objective approach.

The remainder of this paper is organized as follows. Section II provides an overview of appropriate methods for handling data imbalance, with a particular focus on utilization of multi-objective optimization in ensemble learning. In Section III we present the approach proposed in this paper. Section IV contains the description of the conducted experimental analysis. Lastly, in Section V we give our final remarks together with possible future works.

II. RELATED WORKS

This section gives an overview of the important research areas related to this work: imbalanced data classification, classifier ensembles and multi-objective optimization.

A. IMBALANCED DATA CLASSIFICATION

1) METRICS

One crucial issue in the imbalanced data classification is the choice metrics. As mentioned, we generally do not have information about the loss function, hence the evaluation of classification algorithms is based on metrics defined for the binary classification problem, which try to show the algorithm’s behavior for each class. Of course, the evaluation problem for multiclass problems is far more complicated, and the metrics used are generally based on micro- or macro-averaging. Firstly, let us define a confusion matrix (Table 1).

| Predicted label | True label |
|-----------------|------------|
| Positive (1)    | TP (1) | False Positive (FP) |
| Negative (0)    | False Negative (FN) | TN (0) |

Then the following metrics may be formulated [1]:

\[
\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)
\]

\[
\text{recall} = \frac{TP}{TP + FN} \quad (2)
\]

\[
\text{specificity} = \frac{TN}{TN + FP} \quad (3)
\]

\[
\text{precision} = \frac{TP}{TP + FP} \quad (4)
\]

\[
\text{BAC} = \frac{\text{recall} + \text{specificity}}{2} \quad (5)
\]

\[
\text{G-mean} = \sqrt{\text{recall} \times \text{specificity}} \quad (6)
\]

or the currently more commonly used definition

\[
\text{G-mean} = \sqrt{\text{recall} \times \text{precision}} \quad (7)
\]

Empirical evidence proves that accuracy is strongly biased to favor the majority class and might produce misleading conclusions. This fact motivated a search for new balanced measures obtaining a trade-off between positive and negative class performances [16]. Examples of such metrics are the arithmetic (eq.5), geometric (eq.6 or eq.7) or harmonic means (eq.9) between the two components: recall and precision (or specificity).

Empirical evidence proves that accuracy is strongly biased to favor the majority class and might produce misleading conclusions. This fact motivated a search for new balanced measures obtaining a trade-off between positive and negative class performances [16]. Examples of such metrics are the arithmetic (eq.5), geometric (eq.6 or eq.7) or harmonic means (eq.9) between the two components: recall and precision (or specificity).

It is also worth mentioning the attempt to define the so-called parametric metrics, which are able to take, while calculating the value of a metric, the information given by the user, indicating the importance for him of the values of particular simple metrics. One also may find Index of Balanced Accuracy [17]

\[
IBA_\alpha = (1 + \alpha (TPR - TNR)) \times TPR \times TNR \quad (8)
\]

TABLE 1. Confusion matrix for binary problem.
or $F_{\beta\text{score}}$ [18]:

$$F_{\beta\text{score}} = \frac{(\beta^2 + 1) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}} \quad (9)$$

As we mentioned above, the parameters $\alpha$, $\beta$ express a kind of trade-off between both components and should be set properly. Hand [14] criticized the common use of $F_1\text{score}$ metric, i.e. for the parameter $\beta = 1$, as the user should set this parameter to express how much more important the recall value is to him than the precision value. Unfortunately for many practical problems, access to end-user preferences in this area is unknown. The interesting study of metrics behaviors for the chosen classification task could also be found in [13].

The ROC (Receiver Operating Characteristic) curve [19] has also been widely used in this case. Nevertheless, recent studies have shown that $AUC$ (Area under the roc curve) is a fundamentally incoherent measure (see for example [20], where the new alternative, the $H$ measure has been proposed).

Let us also observe that aggregate metrics such as $G\text{-mean}$ are ambiguous because the same metric value can be taken for different precision and recall values, as shown in Figure 1. Thus, one might suspect that machine learning methods that use such an aggregate metric as a criterion will be biased toward certain values of simple metrics, without providing the information that there are equally good solutions (in terms of a given metric) for other values of precision and recall.

2) ALGORITHMS

Three main groups of methods for dealing with imbalanced data can be distinguished: (i) data-level, (ii) algorithm-level, and (iii) hybrid methods, with the last one being a mixture of the previous two. Data-level approaches aim to balance the data artificially so that it may be used by standard classification algorithms [1]. Usually, two main techniques used are under- and oversampling. Algorithm-level approaches focus on changing the behavior of the classification algorithm itself to reduce its bias towards the majority class [21]. A problem with this approach is that it may shift preference to a minority class, which is also unacceptable. Cost-sensitive learning, where misclassification of every class has a different cost, can omit this problem. However, as we mentioned above obtaining such costs is not a straightforward task [22]. If a loss function of the prediction task is known, the overall risk could be used for the best outcome. But the loss function is a problem depended and most often costly or even impossible to obtain. Some techniques were developed, such as utility learning [11], which aims to utilize both benefits of properly classifying a class and its relevance, but it still requires specific knowledge about a problem.

Classifier ensembles are widely used as the hybrid approaches to class imbalance tasks [21], [23]. The most common models to employ or modify [24] are bagging [25] and boosting [26] methods. One of the examples of the application of bagging algorithm is Roughly Balanced Bagging [27], in which the number of samples from minority class is the same for each bag (and equals the size of minority class). The number of samples from majority class varies according to the negative binomial distribution. RUSBoost [28], and SMOTEBoost [29] are modifications of AdaBoost that use, respectively, random undersampling and SMOTE algorithm to adjust data disproportion. Another way to balance classes within the boosting algorithm is evolutionary undersampling [30]. Apart from bagging and boosting, different ensemble methods were proposed. Sun et al. [31] proposed the approach in which the majority class samples are divided into several bins employing clustering. After that, each subset of the majority class is merged with whole minority class data, creating separate balanced problems.

B. MULTI-OBJECTIVE OPTIMIZATION

Multi-objective optimization is gaining attention in the pattern recognition domain, including ensemble learning. Although the diversity of the ensemble is a crucial factor affecting its performance, and the reason why ensembles perform better than individual classifiers [32], empirical experiments show that high diversity does not directly improve the quality of the model [33]. As a result, diversity cannot be used as a sole optimization criterion. However, together with a quality metric such as classification accuracy it can be used as an objective in MOO algorithm to enhance ensemble performance. Liang et al. [34] proposed classifier ensemble utilizing mean Q-statistic measure together with general classification error as the objectives of Multimodal Multi-objective algorithm (mmoo) optimizing ensemble composition. A similar approach can be found in the work of Fletcher et al. [15]. The authors proposed a non-specialized ensemble classifier that utilized double-fault measure as a diversity assessment and popular NSGA-II as an optimization algorithm. Oliveira et al. [35] employed MOO algorithm in generating a pool of base classifiers, where both numbers of features in the training set and the classification error of individual classifier were minimized. Then during classifier ensemble selection, they use as the criteria both diversity and accuracy. Another ensemble model was proposed by Onan et al. [36], that utilized MOO algorithm in assigning

![FIGURE 1. Example of G-mean for a particular value (0.7) in the recall-precision space.](image-url)
weights to the base classifiers in a pool. Different objectives of the optimization algorithm were evaluated and overall precision and recall proved to be the best choice. In [37] Gu et al. specified different phases of an ensemble model generation that MOO might be utilized. They also highlighted the necessity of proper second objective choice (accuracy being the first) and the challenge of selecting a solution from the Pareto front.

It is worth noting here that the use of MOO methods requires interaction with the end-user. It is crucial that the end-user determine the importance of individual criteria or select a solution from a pool of non-dominated solutions. On the one hand, MOO methods provide a useful tool for proposing and selecting a solution from a limited pool of solutions of similar quality concerning a group of criteria; on the other hand, they may, especially in the case of lack of knowledge about the importance of individual criteria, cause difficulties in selecting a solution. Thus, the lack of such interaction makes it practically impossible to apply the mentioned methods effectively or forces to rely on the Pareto front for selecting a solution based on single criteria.

One of the possible ways for solving the latter could be an application of the PROMETHEE method [38]. PROMETHEE was designed to evaluate a set of solutions in the event of a multicriteria problem based on principles provided by the user (decision maker). There are two types of information required for the method to operate appropriately: (i) information between the criteria; (ii) information within each criterion. The first one comes in as a list of weights assigned to each objective and needs cooperation with the end-user. A presumption that the weights are known beforehand may be a stringent assumption, especially for inexperienced users. The second one must be provided as a preference function, describing how a change of the value of the criterion affects the utility of the solution. A pair of objective and specified preference function is called generalized criterion. Six different types of general criterion have been proposed, including

- usual criterion where solutions with a greater value of the evaluation criterion are preferred (regardless of how big the deviation is),
- v-shaped criterion, where until a certain value of the objective difference, one solution is only partially preferred over the other (proportionally to the deviation).

Finally, the PROMETHEE method conducts pair-wise comparisons between each solution and generates a graph of outranking flows. The best solution can be determined using this structure. It was successfully employed in multiple works [39], [40] to select one of the solutions from a Pareto front.

C. MOO ENSEMBLES FOR IMBALANCE DATA PROBLEM

As most of the presented ensemble models employ accuracy as one of their optimization objectives, they are not appropriate for highly imbalanced problems. For disproportional data, more often there is less focus on the ensemble diversity, and more on the quality of the individual models. Ribeiro et al. [39] tested different approaches for using MOO in the ensemble creation: in member generation, member selection, member combination and in a mix of member selection and combination. False positive ratio and false negative ratio were common objectives amongst the approaches, with additional criteria depending on a task (e.g., the number of selected classifiers in member selection). Another possible objectives were presented by Felicioni et al. [41], who chose to employ Relative Cross-Entropy and the Area Under the Precision-Recall Curve. Fernandez et al. [42] compared ensembles created based on solutions from Pareto front, where subsets of original dataset were optimised in regard to their size and learning capabilities (according to auc measure), with single classifier based on solution with the highest quality metric value. They also investigated the impact of different counter-measures to imbalance, including sampling weighting and smote oversampling algorithm. Bhowam et al. [43] also employ MOO method in base classifier generation and selection, however, instead of popular genetic algorithms genetic programming was used. In this case classes’ accuracies with incorporated pairwise failure crediting as a diversity measure were the objectives of optimization. Soda [44] proposed model which is a mixture of standard classifier and classifier trained with a consideration of sample disproportion (e.g., with oversampling). The decision when which classifier’s prediction will be valid (based on a threshold) is optimised according to accuracy and G-mean measures. Let us note that the proposer choice of combination rules that are used to obtain the final decision from classifier ensembles is crucial. A plethora of decision optimization techniques for classifier ensembles are available. However, a conceptually simple and effective method is weighted voting, which is based on the linear combination of labels returned by the base classifiers and it is preferred for highly imbalanced datasets [45], [46].

Let us shortly present how it works [47]. Let \( \Pi \) denotes a pool of \( N \) base classifiers \( \Pi = \{\Psi_1, \Psi_2, \Psi_k, \ldots, \Psi_N\} \) used by the classifier ensemble \( \hat{\Psi} \). The classifier ensemble \( \hat{\Psi} \) classifies about the instance \( x \) on the basis of the following rule

\[
\hat{\Psi}(x) = \arg \max_{i \in M} \sum_{k=1}^{N} \Psi_k(x) = i \ w_k
\]

where \( w_k \) is the weight that is assigned to the \( k \)th individual classifier, \( M \) a set of possible labels, and \([ \cdot ]\) stands for Iverson’s bracket.

III. PROPOSED METHOD

This section describes a classifier ensemble learning algorithm dedicated to imbalanced data classification. As mentioned earlier, it is generally unknown how significant the misclassification cost is to the user for each class for such problems. The use of learning algorithms in which the fitness
A pool of classifiers, which become ensemble members, is generated based on the provided set of models. Models may vary based on distinctive classification methods or different parameters - in this work, the first option was chosen. Diversity is further assured by training each of the created classifiers with a different subset of training data - for that, stratified bagging is used. Two parameters should be provided: the number of bags, or the number of classifiers trained from a single model, and the size of the subset sampled from whole training data. The pseudocode of the process is presented in Algorithm 1.

Algorithm 1 Generating a Pool of Diverse Classifiers

\[ \text{Input: } LS - \text{learning set} \]
\[ TM - \text{set of learning methods} \]
\[ b - \text{number of bags per model} \]
\[ s - \text{size of each bag} \]

\[ \text{Output: } \Pi - \text{pool of trained classifiers} \]
\[ cr - \text{class ratio of } LS \]
\[ N - \text{size of } TM \]

\[ \text{for } i \leftarrow 1 \text{ to } N \text{ do} \]
\[ \text{for } j \leftarrow 1 \text{ to } b \text{ do} \]
\[ TS \leftarrow \text{set of } s \text{ observations sampled with replacement from } LS \text{ that preserve } cr \]
\[ \Psi_i^j \leftarrow \text{train classifier using } TM_i \text{ and } TS \]
\[ \Pi \leftarrow \Pi \cup \{\Psi_i^j\} \]
\[ \text{end for} \]
\[ \text{end for} \]

A. POOL OF CLASSIFIER GENERATION

A flow chart of the model is presented in Figure 2.

FIGURE 2. A flow chart of the model.

function will be an aggregation of precision and recall criteria, indicating, e.g., G-mean or AUC as a criterion in the case of single-objective optimization indicates a single solution to the problem. Moreover, as shown in Fig.1 the aggregated criteria are not unambiguous and may prefer models due to the selected criterion. Hence, we propose the use of MOO. It should allow obtaining a group of non-dominated solutions, from which the users can choose the best one in the context of his/her own applications.

The main goal of the proposed method is to classify imbalanced data, though in principle it may also be used on balanced datasets. Three phases of training can be distinguished in the proposed approach. First, we utilize bagging to create a pool of base classifiers based on several different classification algorithms. Next, MOO algorithm is used to produce a Pareto optimal set of classifier ensembles, jointly optimizing precision and recall of the resulting model. Each ensemble constructed in this step is encoded as a vector of weights assigned to individual ensemble members. Lastly, since MOO methods return not one, but a set of solutions, we have to choose which weights will be used in the final ensemble. The choice could be made by the user manually, or we may employ a predefined criterion based on the PROMETHEE rule, or the best ensemble could be chosen in the context of the single metric presented in (eq. 1-9). After that, proposed model is ready to classify incoming unlabeled data, employing previously calculated weights to aggregate members decision during weighted voting (eq. 10). The flow chart of the model is presented in Figure 2.

Let us describe each step of the proposed algorithm.

B. WEIGHT OPTIMIZATION ALGORITHM

The distinctive part of proposed method is its weights optimization algorithm. Good weight assignment is important to mute weaker ensemble members and amplify the strong ones in a process of weighted majority voting as presented in eq. 10. To better embrace problem of imbalanced data we decided to use MOO algorithm in the process of weight optimization. NSGA-II [48] is a genetic algorithm that sorts all individuals based on their dominance. In our method individuals are represented by a list of real values

\[ Ind = [w_1, w_2, \ldots, w_N] \text{ where } w_i \in [0, 1], \]

where \( w_i \) stands for the weight of \( i \)-th base classifier, while \( N \) is the size of the classifier ensemble.

We propose two fitness functions \( F_1 \) and \( F_2 \).

\[ \begin{align*}
\text{maximize } & F_1 (w_1, w_2, \ldots, w_N) = \text{recall} \\
\text{maximize } & F_2 (w_1, w_2, \ldots, w_N) = \text{precision}
\end{align*} \] (11)

The algorithm seeks to maximize both precision (eq. 4) and recall (2). Because these goals are opposing each other, focusing on only one of them may lead to the situation when all or none of the samples are recognised as positive class.

C. SOLUTION CHOICE

The algorithm 1 returns a set of non-dominated individuals representing a combination rule of classifier ensemble (eq.10). Still, finally, the single set of weights that defines the classifier ensemble must be chosen. Thus, selecting the
final solution must be determined. As we mentioned earlier, the solution selection from the Pareto front should be made with the cooperation of the end-user. He should either independently select a solution from the pool of non-dominated solutions or should define the importance of each criterion by defining weights for precision and recall to employ, e.g., PROMETEE method.

IV. EXPERIMENTAL STUDY

This section presents the results of conducted experiments dedicated to testing the quality of the designed model and comparing its variants with different solution choice rules with SOO methods.

A. OBJECTIVES

The experiment aimed answer the following research questions:

RQ1 Is it possible for the MOO-based model to outperform the quality of the models optimized in regards to its individual objectives?

RQ2 Will the MOO-based model have a better quality than the ones created by single optimization of aggregated metrics?

RQ3 Is there the best approach of selecting one model from the Pareto front solutions?

The consecutive segments describe utilized benchmarked datasets, the configuration of experimental studies, as well as analysis and discussion of obtained results.

B. SETUP

The experiment scenario is presented in Fig. 3. Let us explain its details.

1) CHOICE OF BENCHMARK DATASETS

We run the experiments on 26 different imbalanced datasets from KEEL [49], UCI [50] and Kaggle repositories. The datasets can be split into two categories: (1) small datasets (up to 1484 samples) and (2) big datasets (up to 318k samples). Chosen datasets also differ in the numbers of features (ranging from 3 to 187) and the Imbalance Ratio (IR), varying from 53.6% to 0.2%. Some of the datasets did not represent a binary problem. In these cases, we binarized the dataset, i.e., one class was selected as a minority class and the rest were labeled as one majority one. The description of all datasets is presented in Table 2.

2) IMPLEMENTATION AND REPRODUCIBILITY

Complete source code, sufficient to repeat the experiments, was made available at. Together with the code, we provided the complete results of the conducted experimental analysis as well as additional results using data not presented in this paper.

The proposed algorithm, as well as the experiments described in this work, were implemented in the Python programming language. We also used scikit-learn module [51] included code of used classifiers, while the implementation of optimization algorithms based on from pymoo module [52].

3) CHOICE OF BASE CLASSIFIERS

All ensembles were based of the same set of different classifiers from scikit-learn module [51] were as follows:

- AdaBoost - Decision Tree Classifier as base estimator with 50 iterations,
- Random Forest - with 100 estimators and Gini impurity as split criterion,
- Naive Bayes,
- k-nearest neighbors classifier with $k = 5$,
- multi-layer perceptron (MLP) with one hidden layer of 100 neurons, rectified linear unit function for activation and Adam solver for weights optimisation,
- decision tree (CART) - with Gini impurity as split criterion and no max depth set.

Although AdaBoost and Random Forest are examples of ensemble approach, in our research they are considered as a single estimators.

The number of bags for each classifier was set to 3, which resulted in a pool of 18 models.

4) OPTIMIZATION ALGORITHMS

A standard genetic algorithm (GA) was used for single objective optimization, with a tournament selection, two-point crossover and a Gaussian mutation. For both NSGA-II and GA the number of iterations was set to 500, population consisted of 500 individuals and probability of mutation was 50%.
As the objectives for SOO we chose: precision, recall, balanced accuracy (bac), F1-score, geometric mean (G-mean) and area under roc curve (auc).

5) SOLUTION CHOICE RULES
Because the experiments are run on the benchmark datasets, we do not have access to expert knowledge about the mutual importance of precision and recall. Thus, it is impossible to select the best solution manually without this information. We propose to choose the best solution returned by NSGA II using one of the following rules:

- based on each of the objectives - where either recall or precision has the best value;
- balanced solution - with the smallest difference precision and recall;
- based on PROMETHEE rule with usual criterion and a slight advantage of either recall or precision. The advantage of either criterion was assured by setting the weights (0.6 for favorable objective, 0.4 for the other).

6) DATA PARTITIONING
Inside the model data was divided into training and validation set (used during composition optimization process) with a ratio of 70:30%. The experiments were conducted using 5 × 2 cross-validation and presented results are the average value of all the metrics from individual folds.

7) MODEL COMPARISONS
We divided experiments into two parts. In the first part we compare MOO and SOO optimisation algorithms’ results presented in a form of Pareto fronts. For the second segment of overall classification performance analysis we included two comparative methods of AdaBoost model built on decision trees and Bagging Ensemble consisting of 18 members. Both methods are popular examples of widely utilised ensemble models, without a particular usage of data preprocessing algorithms, as they were not an object of this research and could disrupt the comparison.

8) RESULT ANALYSIS
To analyse the classification performance of the chosen algorithms the Friedman test and Nemenyi post hoc test at a significance level of 0.05 was chosen [16].

C. RESULTS
1) PARETO FRONT ANALYSIS
Figures 4 and 5 present examples of Pareto fronts obtained from the MOO algorithm during one of the conducted iterations (folds) of the experiment together with values of each objective for solutions of single-objective optimization.

We may observe that for big datasets Pareto front is well defined with a lot of diverse solutions. Most SOO solutions lay directly on or at a small distance to the front. The only exceptions are solutions optimized concerning both objectives, that are precision and recall, which sometimes lay far away from the rest. However, such models would not be acceptable in realistic scenarios because they are biased to predict too much or too few samples as a minority class (as the other objective is very low). In the case of smaller datasets, the Pareto front is typically very constrained (in the worst case, the algorithm returned either only one solution or a few solutions but with the same objective values). Moreover, the rest of the single-objective solutions are distributed irregularly, though, with few exceptions, in a similar area. Even though the Pareto front is relatively tight, obtained solutions are considerably balanced in terms of objectives and, in some cases, are even better than the ones given by SOO. Still, this suggests that to achieve a satisfactory performance, particularly a well-defined, wide front, the method requires a reasonably large dataset. Finally, it is worth noting that the single-objective solutions typically have a different position on the Pareto front, indicating that the optimal precision/recall trade-off depends on the specific performance metric choice. This, in turn, indicates the usefulness of multi-objective optimization, which can simultaneously produce multiple solutions optimized to the particular metrics.
2) CLASSIFICATION PERFORMANCE ANALYSIS

The detailed results for the chosen metrics, i.e., precision, recall, balanced accuracy (bac), F1-score, geometric mean (G-mean), and area under roc curve (auc) are presented in the article’s repository. Fig. 6 presents the radar plots to compare the average ranks of chosen metrics for all tested methods, while Fig. 7 includes results of the Friedman NxN test and critical difference returned by Nemenyi post hoc test (significance level: 0.05).

Analyzing the classification quality, it should be noted that for almost all aggregate metrics, the best classification quality is achieved by methods based on MOO and applying a selection solution rule based on recall, i.e., either selecting the best solution relative to recall (MOO recall) or preferring in the PROMETHEE recall rule (MOO promethee recall). A method that indicates the best consideration with similar recall and precision values (MOO balanced) achieves similar results. Interestingly, the methods based on optimizing one criterion related to aggregated metrics obtained worse results than the mentioned methods based on MOO. However, it should be noted that the difference is not statistically significant. The situation is different when analyzing methods from the point of view of simple metrics, such as precision.
and recall. Here, the SOO approach returns the best solutions with a criterion related to a given metric by far. However, as mentioned in the previous section, where Pareto fronts were analyzed, the obtained solutions are usually unacceptable due to the very low value of the second simple metric. For the mentioned simple metrics, the selected solutions returned by MOO are not statistically significantly worse than the mentioned SOO methods. They return much better balanced (from the precision and recall point of view) solutions. When comparing proposed ensemble with established methods of AdaBoost and Bagging it should be noted, that the latter seem to be generally inferior for almost all selected metric. Only for a precision and F1-measure, which is highly dependent on the preceding, ensemble using standard bagging obtains result better than most of the MOO based model. Nevertheless it must be highlighted, that for all the metrics there can be always chosen a variant of proposed method which has higher average rank, though the difference is statistically significant only in the case of recall values. Regarding AdaBoost, both figure 6 and statistical tests results show that most of the MOO based methods are of the better quality, where for precision, recall and F1-measure the difference is statistically significant.

D. LESSONS LEARNED
The answer to the first research question is that the MOO based model can outperform SOO in regards to its objective, but it occurs rarely. However, for almost all datasets for these models, the highest value of one metric corresponded with the lowest value of other metrics, which signalizes bias of said models and would be unacceptable in real-life problems. Similar situation occurred in a case of standard bagging ensemble, where even though its rank according to precision value was considerable, but for recall it was worse than most of the MOO based models. It is worth mentioning that MOO ensembles which were composed based on the highest value of recall and precision from the Pareto front had similar values of respective metrics with the lesser trade-off of opposed ones. SOO models based on aggregated metrics’ optimization seem to not have such a problem as the ones optimizing either precision and recall - their results for both of these metrics are considerably more balanced. The same can
be stated about AdaBoost model. Nevertheless, MOO based ensembles have better or similar values of both precision and recall to ensembles based on SOO optimization of chosen aggregated metrics. As shown in figure 6, MOO methods also have higher average ranks in all respective metrics, which answers the second research question. Nonetheless, it is worth emphasizing that the differences between methods are not always statistically significant (Fig. 7) and may be problem-dependent.

As for the third question, we considered five different methods of choosing solutions for the Pareto front in our experiments. None emerged as the best for all of the chosen metrics. The ones selected according to the best value of either precision and recall turned out to have the highest values of the corresponding metric. However they suffered from the same problem of having significant (however smaller than SOO) trade-off among metrics. For the balanced methods, the values of objectives were slightly lower. However, the trade-off was the smallest and it was by average better than aggregated metrics’ method (figure 6). PROMETHEE methods returned the same or very similar results as selecting solutions based on the best objective values. It was most probably caused by chosen relation criterion - however, for a specific problem, one can set a precise preference function to obtain a more significant result. Nonetheless, especially for numerous data sets, there were remarkably more solutions to choose from and many were close to the aggregated metrics’ models (as shown in figures 4 and 5). Therefore, a potential users can select one according to their preferences.

V. CONCLUSION

The paper focused on the application of multiobjective optimization (MOO) methods to the problem of imbalanced data classification. Classifier ensemble was used as a classification model, for which the problem of weight training for weighted voting of a heterogeneous classifier ensemble was treated as an optimization problem. Computer experiments on a large number of different datasets have shown that classifier ensembles trained using a MOO can return better, more balanced models than methods based on single objective optimization methods (SOO). SOO methods tend to choose models that try to optimize a single criterion but do so at the expense of the other criteria. Additionally, by using the methods of classifier training with the use of MOO, we can present to the end user a pool of models from which the best solution can be selected in the context of the user’s needs.

Possible direction of further research include the problem of choosing other optimization methods or using MOO to select promising directions of searching for a solution that can later be solved as SOO task with constrains. Additionally, the authors intend to use MOO to optimize the hyperparameters of previously developed preprocessing methods such as CCR [53].
REFERENCES

[1] B. Krawczyk, “Learning from imbalanced data: Open challenges and future directions,” *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016.

[2] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, “Learning from imbalanced data in surveillance of nosocomial infection,” *Artif. Intell. Med.*, vol. 37, no. 1, pp. 7–18, May 2006.

[3] M. Khaliﬁta, S. Chakraborty, and M. Popescu, “Predicting disease risks from highly imbalanced data using random forest,” *BM C Med. Inform. Decis. Making*, vol. 11, no. 1, pp. 1–13, Dec. 2011.

[4] M. Azizian, A. Jumabek, and A. Ali, “Combating imbalance in network intrusion datasets,” in *Proc. GrC*, 2006, pp. 732–737.

[5] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, “Effecti ve detection of sophis- ticated online banking fraud on extremely imbalanced data,” *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2012.

[6] K. Fu, D. Cheng, Y. Tu, and L. Zhang, “Credit card fraud detection using convolutional neural networks,” in *Proc. Int. Conf. Neural Inf. Process.*. Cham, Switzerland: Springer, 2016, pp. 483–490.

[7] M. A. Maloof, “Learning when data sets are imbalanced and when costs are unequal and unknown,” in *Proc. Workshop Learn. Imbalanced Data Sets*, vol. 2, 2003, pp. 1–2.

[8] Y. Sun, A. K. Wong, and M. S. Kamel, “Classification of imbalanced data: A review,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, Jun. 2009.

[9] A. P. Bradley, “The use of the area under the ROC curve in the evalu- ation of machine learning algorithms,” *BMC Med. Inform. Decis. Making*, vol. 11, no. 1, pp. 1–13, Dec. 2011.

[10] P. Azizian, A. Jumabek, and A. Ali, “Combating imbalance in network intrusion datasets,” in *Proc. GrC*, 2006, pp. 732–737.

[11] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, “Effective detection of sophis- ticated online banking fraud on extremely imbalanced data,” *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2012.

[12] B. Krawczyk, “Learning from imbalanced data: Open challenges and future directions,” *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016.

[13] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, “Learning from imbalanced data in surveillance of nosocomial infection,” *Artif. Intell. Med.*, vol. 37, no. 1, pp. 7–18, May 2006.

[14] M. Khaliﬁta, S. Chakraborty, and M. Popescu, “Predicting disease risks from highly imbalanced data using random forest,” *BM C Med. Inform. Decis. Making*, vol. 11, no. 1, pp. 1–13, Dec. 2011.

[15] M. Azizian, A. Jumabek, and A. Ali, “Combating imbalance in network intrusion datasets,” in *Proc. GrC*, 2006, pp. 732–737.

[16] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, “Effecti ve detection of sophis- ticated online banking fraud on extremely imbalanced data,” *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2012.

[17] K. Fu, D. Cheng, Y. Tu, and L. Zhang, “Credit card fraud detection using convolutional neural networks,” in *Proc. Int. Conf. Neural Inf. Process.*. Cham, Switzerland: Springer, 2016, pp. 483–490.

[18] M. A. Maloof, “Learning when data sets are imbalanced and when costs are unequal and unknown,” in *Proc. Workshop Learn. Imbalanced Data Sets*, vol. 2, 2003, pp. 1–2.

[19] Y. Sun, A. K. Wong, and M. S. Kamel, “Classification of imbalanced data: A review,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, Jun. 2009.

[20] A. P. Bradley, “The use of the area under the ROC curve in the eval- uation of machine learning algorithms,” *BMC Med. Inform. Decis. Making*, vol. 11, no. 1, pp. 1–13, Dec. 2011.

[21] P. Azizian, A. Jumabek, and A. Ali, “Combating imbalance in network intrusion datasets,” in *Proc. GrC*, 2006, pp. 732–737.

[22] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, “Effecti ve detection of sophis- ticated online banking fraud on extremely imbalanced data,” *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2012.

[23] B. Krawczyk, “Learning from imbalanced data: Open challenges and future directions,” *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016.

[24] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, “Learning from imbalanced data in surveillance of nosocomial infection,” *Artif. Intell. Med.*, vol. 37, no. 1, pp. 7–18, May 2006.

[25] M. Khaliﬁta, S. Chakraborty, and M. Popescu, “Predicting disease risks from highly imbalanced data using random forest,” *BM C Med. Inform. Decis. Making*, vol. 11, no. 1, pp. 1–13, Dec. 2011.
[48] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.

[49] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, “KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework,” J. Multiple-Valued Log. Soft Comput., vol. 17, pp. 1–4, Jun. 2011.

[50] D. Dua and C. Graff. (2017). UCI Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and M. Blondel, “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011.

[52] J. Blank and K. Deb, “Pymoo: Multi-objective optimization in Python,” IEEE Access, vol. 8, pp. 89497–89509, 2020.

[53] M. Koziarski and M. Wozniak, “CCR: A combined cleaning and resampling algorithm for imbalanced data classification,” Int. J. Appl. Math. Comput. Sci., vol. 27, no. 4, pp. 727–736, Jan. 2018, doi: 10.1515/amcs-2017-0050.

WERONIKA WĘGIER received the M.S. degree in computer science from the Wrocław University of Science and Technology, Poland, in 2020, where she is currently pursuing the Ph.D. degree in information and communication technology with the Doctoral School. Her research interests include pattern recognition, imbalanced data classification, ensemble learning, and model optimization.

MICHAŁ KOZIARSKI received the M.Sc. degree in computer science from the Wrocław University of Science and Technology, Poland, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electronics, AGH University of Science and Technology, Poland. His research interests include computer vision, neural networks, and imbalanced data classification.

MICHAŁ WOŹNIAK (Senior Member, IEEE) is currently a Professor of computer science at the Department of Systems and Computer Networks, Wrocław University of Science and Technology, Poland. He has been involved in research projects related to the topics mentioned above and has been a consultant for several commercial projects for well-known Polish companies and public administration. He has published over 300 articles and three books. His research interests include machine learning, compound classification methods, classifier ensembles, data stream mining, and imbalanced data processing. He was awarded numerous prestigious awards for his scientific achievements as the IBM Smarter Planet Faculty Innovation Award (twice) or the IEEE Outstanding Leadership Award, and several best paper awards of the prestigious conferences.