# Multi-criteria analysis involving Pareto-optimal misclassification tradeoffs on imbalanced datasets

Marcos M. Raimundo*†, Fernando J. Von Zuben†
*School of Applied Mathematics – Fundação Getúlio Vargas, Rio de Janeiro, RJ, Brazil
†School of Electrical and Computer Engineering – University of Campinas (UNICAMP), Campinas, SP, Brazil
E-mail: marcosmrai@gmail.com, vonzuben@dca.fee.unicamp.br

***Abstract*—On binary classification, the goal of minimizing the false positive and false negative rates creates a conflict, being impossible to optimize both simultaneously. This challenge is even more significant on imbalanced classification datasets since an incorrect choice of the relative relevance of each objective on the optimization can lead to ignoring, or poorly learning the minority class. The proposal of this work takes into account the existing conflict among the learning losses of the classes, and use a deterministic multi-objective optimization method, called MONISE, to create a set of solutions with diverse misclassification tradeoffs among the classes. Since accuracy is no longer a proper criterion for imbalanced datasets, we had to resort to multiple criteria to report the performance: each classifier, proposed or competitors, was selected and reported using the same metrics. We used $F_1$, kappa and g-mean for general evaluation of performance and $F_\beta$s ($F_{1/16}$, $F_{1/4}$, $F_4$ and $F_{16}$) to emulate a shifting decision maker preference from precision to recall; all comparisons were made using a Friedman test with Finner posthoc test. However, when we take into account multiple metrics without any prior knowledge, it may become impossible to pinpoint the best method, since the evaluation criteria may also be in conflict. Again, to solve this, we resorted to a Friedman test with a non-dominated ranking. With this multi-criteria analyses, we conclude that explicitly considering multiple objectives on the optimization can guide to promising results.**

## I. Introduction

The presence of imbalanced datasets is a common problem in classification tasks characterized by a high distinction in the number of samples associated with each class. This can be a natural issue in scenarios which are inherently imbalanced such as fraud detection, medical diagnosis, network intrusion detection, detection of oil spills, and manufacturing issue detection [1]. The imbalance of a classification set can deteriorate the performance of a non-specialized classifier. In those scenarios, it is necessary to create methodologies to handle this problem. Supported by the taxonomy explored in [2], we aim at discussing some of those methodologies.

**Cost-sensitive** approaches consist of weighting the cost of misclassification for each class and using these costs to guide the learning process. The most naïve approach tries to reduce the effect of imbalance by weighting the classes inversely proportional to the frequency of the samples on each class [3]. It can be accomplished by creating adjustable weight factors on each loss term [4], [5], by changing the boosting weight update to differently calculate the majority and minority class samples [6], [7], or by adding class-specific terms in the kernel calculation to become cost-sensitive [8].

**Sampling-level** approaches consist of cleverly under-sampling the majority class samples and/or oversampling the minority class samples. The main work on the oversampling vein, called SMOTE, creates new samples by a convex combination of minority samples with their neighbors of the same class [9]. Each minority sample creates the same number of synthetic samples [9]; it can be proportional to the ratio of majority samples in the neighborhood [10]; or the generation can be constrained to the samples in the borderline with majority class samples (at least 50% of the neighbors) [11]. The under-sampling usually removes some majority class samples from the training set. A grid search can select the percentage of random under(and over)-sampling [9] or using a wrapper algorithm to select the amount of under-re-sampling and SMOTE over-sampling by firstly finding a valid undersampling followed by a performance improvement SMOTE oversampling [12].

**Algorithmic** approaches changes the inner workings of existing algorithms to reduce the bias towards the majority class. There are several approaches in this vein: the use of Hellinger distance to adapt decision trees [13]; the use of dynamic ensemble selection to locally choose less biased classifiers [14], [15]; and the use of KNN selection of samples to construct sample-specific classifiers [16].

**Boosting** approaches adapt each step of AdaBoost to correct the bias of the majority class. It can be done by modifying the weight updating to be cost-sensitive [17], by applying SMOTE to over-sampling the minority class on each step [7], by under-sampling the majority class on each step [18], and by over-sampling the minority class with SMOTE on each step [19]. Preference is generally given to samples with more neighbours in the majority class and hard-to-learn samples [20], considered by the authors as being more frequent in the minority class. Other **ensemble** approaches are founded on creating each learning machine by sub-sampling only the majority class [21], by also removing the correctly classified samples from the majority class [21], and by bootstrap under-sampling followed by SMOTE over-sampling, thus creating balanced datasets. Other ensemble methods create a set of components using random class proportions [22], and they extend this approach for a multi-class scenario as well [23].

The use of **multi-objective optimization** can surely be considered a comprehensive approach in imbalanced classification, mainly because it can be used to (1) properly adjust the

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

number of target samples and the number of neighbors simultaneously optimizing accuracy and kappa [24], (2) minimize the number of selected samples and maximize AUC while selecting features [25], and (3) maximize the accuracy and the geometric mean of accuracies for each class [26]. However, these works usually use evaluation metrics as the conflicting objectives, instead of investigating the conflicting aspects of machine learning methods. An example of this negligence can be seen in a multinomial regression (further explored and explained in the text) expressed in Equation (1), where the loss of all classes are considered to exhibit a fixed degree of relative relevance:

$$\min_{\boldsymbol{\theta}} \quad \sum_{k=1}^{K} \left[ \sum_{i=1}^{N} -y_i^k \ln \left( \frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] \equiv \sum_{k=1}^{K} l_k(\boldsymbol{\theta}), \qquad (1)$$

where $\frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}$ is the probability that the model assigns class $k$ to sample $i$, $y_i^k = 1$ if sample $i$ is originally assigned to class $k$ and $y_i^k = 0$ otherwise, thus $-y_i^k \ln \left( \frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right)$ indicates the learning error of sample $i$ associated with class $k$.

In this case, it is possible to suppose that the class learning losses, expressed by $l_k(\boldsymbol{\theta})$, $k \in \{1, \ldots, K\}$, might be in conflict. It happens because correctly classifing more samples from one class may induce an increase in the misclassification rate of other classes. Equation (1) implicitly takes a "flat" a priori preference among samples, and the optimization method will search the model with the lowest uniformly aggregated loss. As a consequence: (*i*) this uniformly aggregated loss will indirectly improve the classification accuracy of the majority classes because they have more samples in cases of data imbalance; and (*ii*) it may also not fulfill the expectation of the user in some sensible scenarios such as in medical cases, where wrongly classifying a case as a disease may not be as crucial as sending a sick patient home. But the multi-objective concepts can further explore more intrinsic conflicting aspects of machine learning methods, such as taking into account the conflict between the error in each class and the regularization of the learning machines [27], [28].

Given that we further explore this intrinsic aspect of imbalanced classification problems by proposing a framework that is robust enough to satisfy multiple criteria established by decision makers, and flexible enough to also be capable of producing diverse and accurate ensemble components. The proposed framework is composed of three steps: (1) modeling the problem taking into account the conflict among the losses of the classes; (2) using multi-objective optimization to create a set of efficient classifiers with distinct preferences among the classes; (3) applying an a posteriori criterion, designed by the decision maker, to choose among the set of candidate classifiers. However, classification problems – especially with imbalanced datasets where the accuracy is strongly misleading – have no single metric to evaluate the performance of the classifiers. Because of this, we also proposed a non-dominated ranked Friedman test to properly evaluate the classifiers taking into account the non-dominance level among criteria, instead of looking at the performance of an individual criterion.

## II. Proposed method

### A. Multi-objective optimization

There are circumstances in which we want to choose, in the objective space, between two possible solutions $\mathbf{y}^i \in \mathbb{R}^m$ and $\mathbf{y}^j \in \mathbb{R}^m$ and there exist multiple objectives to be minimized ($m \geq 2$). The major challenge in this problem is posed when there exists one objective $k$ where $\mathbf{y}_k^i < \mathbf{y}_k^j$ and other objective $l$ where $\mathbf{y}_l^i > \mathbf{y}_l^j$, being impossible to establish an orderly relation between $\mathbf{y}^i$ and $\mathbf{y}^j$. This situation is called **non-dominance**. However, there are cases in which $\mathbf{y}_k^i \leq \mathbf{y}_k^j, \forall k \in \{1, \ldots, m\}$ and $\exists k : \mathbf{y}_k^i < \mathbf{y}_k^j$, making $\mathbf{y}^i$ always better than $\mathbf{y}^j$. This situation configures what is called **dominance**. In this case, it is possible to exclude $\mathbf{y}^j$ from the set of candidate solutions presented to the decision maker, thus remaining solely the non-dominated candidate solutions.

Given that, a multi-objective optimization is defined base on multiple criteria that we want to optimize. The equivalent to optimal solutions, in this case, are called efficient solutions, or Pareto-optimal solutions. They consist of a set of solutions not dominated by any other solution, and they do not dominate each other. Each one of those solutions represents a tradeoff between the objectives and might satisfy decision makers with distinct preferences. Figure 1 shows two Pareto-fronts with a tradeoff between the multinomial loss for class 1, $l_1(\boldsymbol{\theta})$, and for class 2, $l_2(\boldsymbol{\theta})$.

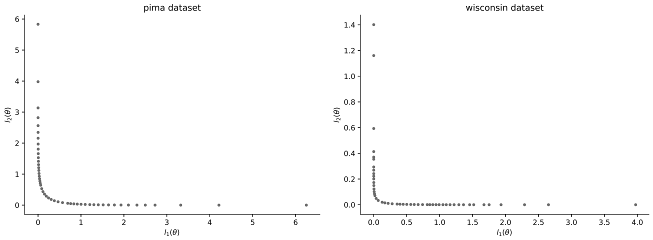

Fig. 1. Pareto front representations of the multinomial logistic error both associated with two classes for Pima and Wisconsin datasets (Available at archive.ics.uci.edu/ml)

A good parallel with the example of Figure 1 is posed by the conflict between false positive and false negative rates [29]. A well-distributed sample of the whole front can be useful to select the most appropriate classifier for different situations which the decision makers might face with, and as we can see further in the paper this also depends on the metric adopted by the decision maker.

### B. Weighted sum method and Many-objective NISE

One way to deal with the high dimensionality and orderly relation issues in multi-objective optimization is by taking the convex combination of the objective functions, called weighted sum method (Definition 1). It consists of summing all objectives with a non-negative weight for every objective.

**Definition 1.** *The definition of the weighted sum method is as follows:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{w}^\top \mathbf{f}(\mathbf{x}) \\ \textit{subject to} \quad & \mathbf{x} \in \Omega, \Omega \subset \mathbb{R}^n, \\ & \mathbf{f}(\mathbf{x}) : \Omega \to \Psi, \Psi \subset \mathbb{R}^m \end{aligned} \tag{2}$$

*where* $\mathbf{w}_i \geq 0,\ \forall i \in \{1, 2, \ldots, m\}$ *and* $\mathbf{w}^\top \mathbf{1} = 1$.

Solving the optimization problem of Definition 1 leads to a single solution that is guaranteed to be efficient [30]. However, the approach proposed in this work involves finding a set of solutions well-distributed along the whole Pareto-front. Taking an arbitrary sampling of the weights can overlook the full potential of the multi-objective perspective [31], possibly creating a poor representation of the solutions. Given that the chosen machine learning model (presented in the next section) is convex, MONISE (Many-Objective NISE) is a high-quality optimizer for convex problems with more than two objective functions [32]. This method is capable of sequentially finding the $R$ most representative efficient solutions thoughout the Pareto front, that will be later selected according to the a posteriori preference of the decision maker. Alternatively, we can try to filter and aggregate those efficient solutions in an ensemble.

### C. Class-conflicting multinomial logistic regression

Consider a classification problem with $N$ samples, in which $\mathbf{x}_i \in \mathbb{R}^d : i \in \{1, \ldots, N\}$ are the input variables and $\mathbf{y}_i^k \in \{0, 1\} : i \in \{1, \ldots, N\}, k \in \{1, \ldots, K\}$ are the output variables, each indicating the assignment of the $i$-th sample to the $k$-th class. Since we are considering multiclass single-label problems, only one class might be set to 1 for each sample ($\sum_{k=1}^K \mathbf{y}_i^k = 1, \forall i \in \{1, \ldots, N\}$). Also, it is important to define the number of samples for each class $n_k = \sum_{i=1}^N \mathbf{y}_i^k$.

Given that most methods for imbalanced classification act as a meta learner, we select the regularized multinomial logistic regression as the base classifier. This choice is convenient to keep the convexity of the optimization, a necessary condition to the proper behavior of the proposed algorithm. The traditional formulation is depicted in Formulation (3):

$$\min_{\boldsymbol{\theta}} \quad w_1 \sum_{k=1}^{K} \frac{1}{u_k} \sum_{i=1}^{N} - \left[ y_i^k \ln \left( \frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^K e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_2 ||\boldsymbol{\theta}||_2^2, \tag{3}$$

where $u_k$ helps in correcting the optimization to take into account the minority class. By setting $u_k$ to be the number of samples of class $k$ ($u_k = n_k$), we end up with a "flat" preference among the classes. On the other hand, setting to one will guide to a "flat" preference among the samples ($u_k = 1$).

To explore the potential of the multi-objective framework in the imbalanced scenarios, instead of using the simple regularized multinomial model, we are going to adopt the multinomial model that considers the loss of every class $\left( -\frac{1}{n_k} \sum_{i=1}^N y_i^k \ln \left( \frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^K e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right)$ as conflicting objectives. This alternative, called here **class-conflicting multinomial logistic regression** is presented in Formulation (4):

$$\min_{\boldsymbol{\theta}} \quad \sum_{k=1}^{K} w_k \left[ -\frac{1}{n_k} \sum_{i=1}^{N} y_i^k \ln \left( \frac{e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}}{\sum_{j=1}^K e^{\boldsymbol{\theta}_k^\top \phi(\mathbf{x}_i)}} \right) \right] + w_{K+1} ||\boldsymbol{\theta}||_2^2, \tag{4}$$

In Formulation (4), the weight $w_k$ may be interpreted as the misclassification weight of the $k$-th class, and it also characterizes the proposal as a cost-sensitive approach where the weights are Pareto-optimally sampled instead of arbitrarily adjusted.

### D. Model selection and ensemble aggregation

Knowing that those generative procedures produce a set of candidate models, and that the multi-objective optimization methods are capable of generating a diverse set of learning machines [33], it is meaningful to create procedures to aggregate these models, and we employ three distinct procedures to construct high-performance classifiers: (1) making the prediction using the best model (kept with no additional tag identifier), consisting in a simple model selection; (2) selecting the 10 best models (an elitist approach) and aggregating their outputs by summing up the output probabilities (identified as **Elt**), a procedure called distribution summation [34]; and (3) combining all output probabilities using another multinomial logistic regression with flat preference among the classes (identified as **Stk**), a procedure called stacking [34]. Considering, as an example, the **MO&AllAdHoc** generation strategy explained in the next section and taking all three aggregation proposals, we end up with three classifiers: **MO&AllAdHoc**, that selects the best model coming from all the three ways to generate models, **EltMO&AllAdHoc** that sums the distribution of the 10 best models, and **StkMO&AllAdHoc** that uses another model trained using the output of those models as the feature vector.

## III. Experimental setup

The **proposed** framework consists in exploring different combinations of models generated using multi-objective optimization. Then, starting from this set of efficient learning models, we can select the best model (using a posteriori preferences) or combine them using an ensemble. Formulation (4) is used to generate 150 efficient models using MONISE, that we call here as **MO**. However, since it could not find the models with a flat preference among the samples, as well as the models with flat preference among the classes, we created two approaches using Formulation (3): **StandardMO** for the 50 models generated with $u_k = 1$, and **CSMO** for the 50 models generated with $u_k = n_k$. Given that, **MO&AllAdHoc** uses the classifiers from all three generation proposals (**MO**, **StandardMO**, and **CSMO**) that take into account all ways of generating classifiers by manipulating the class weights, and consists in the primary formulation of this proposal. Also, **AllAdHoc** aggregates **StandartMO** and **CSMO** to clarify the impact of the models coming from a closed form selection of the weights among the classes (Formulation (3)) and coming from letting multi-objective optimization to find the most representative models (Formulation (4)).

The **comparison** was made using the following algorithms[1]: 1. cost-sensitive methods: Standard – keep the same importance for each sample $u_k = 1$ (StandardManual), ad-hoc cost sensitive – keep the same importance for each class $u_k = n_k$ (CSManual); 2. over-sampling methods: SMOTE, ADASYN, random oversampling (RndOverSamp); 3. under-sampling methods: ENN, Tomek-Links (TL), random undersampling (RndOverSamp); 4. over-sampling followed by under-sampling methods: SMOTEENN, SMOTETL; and 5. ensemble methods: SMOTEBoost, RAMOboost, Easy Ensemble. Given that those methods act as meta-learners, we use a standard regularized multinomial regression as their base classifier, the number of neighbors is kept as $k = 5$, the under-sampling methods are targeted to achieve $\underline{s} + \alpha(s_k - \underline{s})$ samples and the over-sampling methods are targeted to achieve $s_k + \alpha(\overline{s} - s_k)$ and $\alpha$ is varied in the set: $\{0.01, 0.1, 0.2, 0.5, 1\}$. Notice that these settings also work on multi-class problems. Also, after sampling, the training procedure evaluates models taking constant steps on a logarithmic scale ($\lambda \in \{2^{-\frac{P}{2}}$,-$2^{-\frac{P}{2}+1}, \ldots, 2^{\frac{P}{2}-1}, 2^{\frac{P}{2}}\} \cup \{0\}$) [35]. We use cross-validation to select: $\alpha$ from the sampling methods (except TomekLinks and ENN), $\lambda$ from the cost sensitive and sampling methods, and the number of candidate models in the ensemble approaches.

Those methods are **evaluated** using KEEL dataset[2]: it has 22 datasets with imbalance level lower than 9; 78 datasets with imbalance level higher than 9; and 15 datasets with more than two classes. The testing procedures adopted are the 5-fold splitting procedure, and 25% of the remaining dataset separated to perform validation. For all methodologies (proposed and competitors), we use the same evaluation metric to tune the method in the validation set and to evaluate them in the test set. This information is crucial because of the multiplicity of evaluation metrics. The reported performance for a method in a given dataset consists in the mean obtained along the five folds.

Given that, we created 3 experiments with distinct objectives:

- **Experiment 1**: In this experiment, we want to compare the multi-objective methods with methods designed to imbalanced classification datasets. Therefore we compared the best model from multi-objective methods (**MO**, **MO&AllAdHoc**, **CSMO**, **AllAdHoc** and **StardardMO**) with all imbalanced methods for the tree sets of KEEL.
- **Experiment 2**: In this experiment, we want to simulate the capability of the methods to accomplish preferences of the decision maker. This simulation was done using $F_\beta$ measures ($F_{16}$, $F_4$, $F_{1/4}$ and $F_{1/16}$), that is capable of tuning the preferences towards precision ($\beta < 1$) or recall ($\beta > 1$). We compared the same methods coming from the previous experiment.
- **Experiment 3**: In this experiment, we want to compare only the best multi-objective generation in the other experiments, followed by elite selection **EltMO&AllAdHoc** (the best 10 methods given a measure) or staking aggregation **StkMO&AllAdHoc** (creates another model with the outputs) and the ensemble methods designed to imbalanced classification (**SMOTEBoost**, **RAMOBoost**, **EasyEnsemble**). We also kept the ensemble generation followed by a simple model selection **MO&AllAdHoc**.

We **compared** the classification algorithms for a set of datasets using a Friedman test [36], with $p = 0.01$ as a threshold to indicate the statistical difference, and using Finner posthoc test [37] with the same threshold. This is done for each evaluation metric and for each experiment.

With the problem of using accuracy for imbalanced learning, the decision of which metric should be used to evaluate the quality of a predictor in imbalanced learning becomes challenging. Because of this, we use three distinct metrics: kappa, g-mean (it consists of the geometric mean of the recall for every class) and $F_1$, where the multi-class versions of g-mean consists in a geometric mean of the recall of each label, and the $F_1$ measure is the average of $F_1$ for each label. Notice that all these metrics are capable of handling imbalanced datasets [24], and $F_\beta$ measure is capable of simulating change in preferences.

However, when comparing the algorithms for multiple metrics, it is interesting to find a procedure to rank these algorithms. To meet this goal, we also compared the classification algorithms for a set of datasets using a Friedman test for non-dominance (with Finner posthoc test) by using non-dominated sorting to rank the algorithms (further explained in the Appendix).

## IV. Results

Tables I to V present information for each evaluated method (indicated in the **Method** column): the average rank (in the **Rank** column); the number of methods better than the evaluated method (in the #< column); and the number of methods worse than the evaluated method (in the #> column). If the Friedman test is rejected, both columns #< and #> will be marked with a dash (–). This orderly relation (better and worse) is accounted only if there is statistical significance according to the Finner *posthoc* test. The last group of columns (non-dom) shows the Friedman test for non-dominance.

**Experiment 1** is presented in Tables I, II and III for all competitors with a single choice coming from the different multi-objective model generations for the datasets with imbalance level lower than 9, imbalance level higher than 9 and multiclass classification, respectively; the comparison is made using kappa, $F_1$ and g-mean score. **Experiment 2** is presented in Table IV and compares the same methods but for $F_\beta$ measures ($F_{16}$, $F_4$, $F_{1/4}$ and $F_{1/16}$). Finally, **Experiment 3** is presented in Table V and compares ensemble methods designed

[1] Available at contrib.scikit-learn.org/imbalanced-learn and github.com/dialnd/imbalanced-algorithms

[2] Available at sci2s.ugr.es/keel/imbalanced.php

TABLE I
FRIEDMAN RANK (AVERAGE) CONSIDERING G-MEAN, KAPPA AND $F_1$ METRICS FOR THE DATASETS WITH IMBALANCE LEVEL LOWER THAN 9.

| | kappa | | | $F_1$ | | | g-mean | | | non-dom | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> |
| MO&AllAdHoc | 8.06 | – | – | 8.65 | – | – | 6.27 | 0 | 4 | 5.11 | 0 | 6 |
| EasyEnsemble | 7.77 | – | – | 8.06 | – | – | 4.72 | 0 | 7 | 5.84 | 0 | 4 |
| AllAdHoc | 9.49 | – | – | 8.99 | – | – | 7.34 | 0 | 3 | 6.54 | 0 | 2 |
| CSMO | 8.40 | – | – | 8.95 | – | – | 6.72 | 0 | 4 | 6.84 | 0 | 2 |
| RAMOBoost | 8.49 | – | – | 8.47 | – | – | 8.20 | 0 | 3 | 8.43 | 0 | 1 |
| SMOTEBoost | 9.40 | – | – | 9.38 | – | – | 9.29 | 0 | 1 | 8.54 | 0 | 1 |
| MO | 10.40 | – | – | 10.27 | – | – | 9.68 | 1 | 1 | 8.90 | 0 | 1 |
| SMOTETomek | 9.09 | – | – | 9.06 | – | – | 10.22 | 1 | 0 | 9.27 | 0 | 1 |
| SMOTEENN | 9.93 | – | – | 10.36 | – | – | 9.81 | 1 | 0 | 9.54 | 0 | 1 |
| RndOverSamp | 9.13 | – | – | 8.95 | – | – | 8.79 | 0 | 2 | 9.93 | 0 | 0 |
| RndUnderSamp | 9.02 | – | – | 9.06 | – | – | 9.61 | 0 | 1 | 10.18 | 0 | 0 |
| CSManual | 8.90 | – | – | 8.88 | – | – | 7.38 | 0 | 3 | 10.24 | 0 | 0 |
| SMOTE | 8.15 | – | – | 7.88 | – | – | 8.95 | 0 | 2 | 10.45 | 1 | 0 |
| ENN | 10.06 | – | – | 10.29 | – | – | 11.84 | 3 | 0 | 10.99 | 1 | 0 |
| ADASYN | 10.38 | – | – | 9.90 | – | – | 9.63 | 0 | 1 | 11.15 | 2 | 0 |
| StandardMO | 11.43 | – | – | 11.31 | – | – | 14.11 | 8 | 0 | 11.31 | 2 | 0 |
| TomekLinks | 11.36 | – | – | 11.15 | – | – | 13.70 | 6 | 0 | 12.61 | 4 | 0 |
| StandardManual | 11.43 | – | – | 11.27 | – | – | 14.65 | 12 | 0 | 15.04 | 9 | 0 |

TABLE II
FRIEDMAN RANK (AVERAGE) CONSIDERING G-MEAN, KAPPA AND $F_1$ METRICS FOR THE DATASETS WITH IMBALANCE LEVEL HIGHER THAN 9.

| | kappa | | | $F_1$ | | | g-mean | | | non-dom | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> |
| MO&AllAdHoc | 8.12 | 0 | 3 | 7.01 | 0 | 7 | 7.10 | 0 | 10 | 5.18 | 0 | 13 |
| AllAdHoc | 8.46 | 0 | 3 | 9.45 | 0 | 1 | 6.84 | 0 | 10 | 6.29 | 0 | 12 |
| CSMO | 9.99 | 1 | 1 | 10.90 | 3 | 1 | 5.09 | 0 | 13 | 6.69 | 0 | 11 |
| MO | 8.63 | 0 | 3 | 8.96 | 0 | 1 | 7.77 | 1 | 6 | 6.79 | 0 | 11 |
| SMOTEBoost | 7.32 | 0 | 6 | 7.22 | 0 | 6 | 7.88 | 1 | 6 | 7.23 | 0 | 10 |
| RAMOBoost | 7.81 | 0 | 3 | 7.83 | 0 | 5 | 7.94 | 1 | 5 | 8.05 | 1 | 6 |
| SMOTEENN | 8.72 | 0 | 3 | 8.41 | 0 | 3 | 9.85 | 5 | 4 | 8.75 | 2 | 3 |
| SMOTETomek | 8.83 | 0 | 3 | 8.56 | 0 | 2 | 10.11 | 5 | 4 | 9.55 | 4 | 2 |
| EasyEnsemble | 14.32 | 16 | 0 | 15.16 | 17 | 0 | 6.98 | 0 | 10 | 9.85 | 5 | 2 |
| CSManual | 10.31 | 1 | 1 | 10.83 | 3 | 1 | 5.94 | 0 | 10 | 9.87 | 5 | 2 |
| RndUnderSamp | 8.44 | 0 | 3 | 8.34 | 0 | 3 | 10.25 | 7 | 3 | 10.13 | 5 | 2 |
| RndOverSamp | 8.99 | 0 | 3 | 8.83 | 0 | 1 | 9.82 | 5 | 4 | 10.39 | 5 | 1 |
| SMOTE | 8.73 | 0 | 3 | 8.60 | 0 | 2 | 9.78 | 5 | 4 | 10.89 | 6 | 1 |
| StandardMO | 9.88 | 0 | 1 | 9.91 | 2 | 1 | 13.53 | 14 | 0 | 10.93 | 6 | 1 |
| ADASYN | 8.72 | 0 | 3 | 8.63 | 0 | 2 | 10.49 | 8 | 3 | 10.97 | 6 | 1 |
| ENN | 10.06 | 1 | 1 | 9.77 | 1 | 1 | 12.51 | 12 | 1 | 11.94 | 7 | 1 |
| TomekLinks | 11.67 | 11 | 1 | 11.12 | 5 | 1 | 13.74 | 14 | 0 | 12.56 | 11 | 0 |
| StandardManual | 11.91 | 11 | 0 | 11.37 | 8 | 1 | 15.25 | 15 | 0 | 14.83 | 16 | 0 |

TABLE III
FRIEDMAN RANK (AVERAGE) CONSIDERING G-MEAN, KAPPA AND $F_1$ METRICS FOR THE DATASETS WITH MULTIPLE CLASSES.

| | kappa | | | $F_1$ | | | g-mean | | | non-dom | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> |
| MO&AllAdHoc | 8.36 | 0 | 0 | 6.79 | 0 | 2 | 7.79 | – | – | 5.66 | 0 | 3 |
| AllAdHoc | 6.93 | 0 | 0 | 6.73 | 0 | 2 | 7.53 | – | – | 5.76 | 0 | 3 |
| CSMO | 8.23 | 0 | 0 | 6.46 | 0 | 2 | 7.13 | – | – | 6.36 | 0 | 2 |
| RndOverSamp | 7.43 | 0 | 0 | 7.40 | 0 | 2 | 8.16 | – | – | 7.26 | 0 | 0 |
| SMOTETomek | 8.49 | 0 | 0 | 9.43 | 0 | 0 | 7.06 | – | – | 8.23 | 0 | 0 |
| StandardMO | 10.13 | 0 | 0 | 7.26 | 0 | 2 | 11.40 | – | – | 8.66 | 0 | 0 |
| SMOTEENN | 9.53 | 0 | 0 | 9.80 | 0 | 0 | 8.90 | – | – | 8.99 | 0 | 0 |
| RndUnderSamp | 7.86 | 0 | 0 | 8.66 | 0 | 0 | 8.63 | – | – | 9.19 | 0 | 0 |
| ADASYN | 7.63 | 0 | 0 | 8.73 | 0 | 0 | 9.40 | – | – | 9.33 | 0 | 0 |
| RAMOBoost | 8.66 | 0 | 0 | 8.86 | 0 | 0 | 10.16 | – | – | 9.49 | 0 | 0 |
| SMOTE | 7.43 | 0 | 0 | 8.80 | 0 | 0 | 8.90 | – | – | 9.80 | 0 | 0 |
| CSManual | 8.46 | 0 | 0 | 9.09 | 0 | 0 | 8.16 | – | – | 10.06 | 0 | 0 |
| EasyEnsemble | 12.59 | 0 | 0 | 12.46 | 0 | 0 | 8.96 | – | – | 10.19 | 0 | 0 |
| SMOTEBoost | 9.06 | 0 | 0 | 9.59 | 0 | 0 | 9.59 | – | – | 10.26 | 0 | 0 |
| TomekLinks | 9.99 | 0 | 0 | 10.26 | 0 | 0 | 12.49 | – | – | 11.03 | 0 | 0 |
| MO | 14.56 | 0 | 0 | 14.73 | 5 | 0 | 12.59 | – | – | 12.90 | 2 | 0 |
| ENN | 14.16 | 0 | 0 | 14.43 | 5 | 0 | 11.23 | – | – | 13.83 | 3 | 0 |
| StandardManual | 11.40 | 0 | 0 | 11.43 | 0 | 0 | 12.83 | – | – | 13.90 | 3 | 0 |

TABLE IV

FRIEDMAN RANK (AVERAGE) CONSIDERING $F_\beta$ METRICS ($F_{16}$, $F_4$, $F_{1/4}$ AND $F_{1/16}$) FOR THE DATASETS WITH IMBALANCE LEVEL HIGHER THAN 9.

| | $F_{16}$ | | | $F_4$ | | | $F_{1/4}$ | | | $F_{1/16}$ | | | non-dom | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> |
| MO&AllAdHoc | 6.77 | 0 | 10 | 7.37 | 0 | 8 | 6.81 | 0 | 7 | 6.64 | 0 | 9 | 4.78 | 0 | 16 |
| AllAdHoc | 7.47 | 0 | 7 | 7.49 | 0 | 8 | 7.92 | 0 | 3 | 7.18 | 0 | 6 | 6.01 | 0 | 13 |
| CSMO | 5.92 | 0 | 10 | 6.61 | 0 | 10 | 11.28 | 5 | 1 | 11.29 | 6 | 1 | 7.38 | 1 | 9 |
| MO | 7.80 | 0 | 5 | 7.88 | 0 | 5 | 9.14 | 0 | 2 | 8.98 | 0 | 2 | 7.42 | 1 | 9 |
| SMOTEBoost | 7.58 | 0 | 6 | 6.53 | 0 | 10 | 9.20 | 0 | 2 | 9.69 | 1 | 1 | 7.49 | 1 | 9 |
| RAMOBoost | 7.64 | 0 | 6 | 6.94 | 0 | 9 | 9.58 | 1 | 1 | 9.97 | 2 | 1 | 8.81 | 2 | 3 |
| SMOTEENN | 9.67 | 3 | 4 | 8.90 | 0 | 4 | 9.16 | 0 | 2 | 8.94 | 0 | 2 | 9.05 | 2 | 3 |
| SMOTETomek | 9.71 | 3 | 4 | 9.72 | 4 | 4 | 8.71 | 0 | 2 | 9.30 | 1 | 2 | 9.55 | 2 | 3 |
| StandardMO | 13.01 | 14 | 0 | 12.35 | 11 | 0 | 7.85 | 0 | 3 | 8.12 | 0 | 3 | 9.80 | 2 | 1 |
| RndUnderSamp | 9.94 | 4 | 4 | 9.92 | 6 | 4 | 8.70 | 0 | 2 | 8.51 | 0 | 3 | 10.09 | 5 | 1 |
| CSManual | 6.08 | 0 | 10 | 6.68 | 0 | 10 | 12.15 | 11 | 1 | 11.88 | 10 | 1 | 10.18 | 5 | 1 |
| SMOTE | 9.41 | 3 | 4 | 9.27 | 3 | 4 | 8.99 | 0 | 2 | 8.78 | 0 | 2 | 10.46 | 5 | 1 |
| EasyEnsemble | 8.11 | 0 | 4 | 10.69 | 7 | 2 | 15.25 | 17 | 0 | 14.90 | 17 | 0 | 10.61 | 5 | 1 |
| RndOverSamp | 10.08 | 6 | 4 | 9.98 | 6 | 3 | 8.37 | 0 | 3 | 8.56 | 0 | 3 | 10.63 | 5 | 1 |
| ADASYN | 10.45 | 7 | 3 | 10.12 | 6 | 2 | 8.51 | 0 | 3 | 8.63 | 0 | 3 | 10.70 | 5 | 1 |
| ENN | 12.70 | 13 | 0 | 12.39 | 12 | 0 | 9.77 | 1 | 1 | 9.87 | 2 | 1 | 12.05 | 8 | 0 |
| TomekLinks | 14.15 | 14 | 0 | 13.84 | 14 | 0 | 9.58 | 1 | 1 | 9.61 | 1 | 1 | 12.17 | 8 | 0 |
| StandardManual | 14.42 | 14 | 0 | 14.22 | 14 | 0 | 9.94 | 1 | 1 | 10.06 | 2 | 1 | 13.72 | 15 | 0 |

TABLE V

FRIEDMAN RANK (AVERAGE) CONSIDERING G-MEAN, KAPPA AND $F_1$ METRICS FOR THE DATASETS WITH IMBALANCE LEVEL HIGHER THAN 9.

| | kappa | | | $F_1$ | | | g-mean | | | non-dom | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> | Rank | #< | #> |
| StkMO&AllAdHoc | 3.06 | 0 | 1 | 3.12 | 0 | 1 | 2.92 | – | – | 2.42 | 0 | 4 |
| EltMO&AllAdHoc | 3.10 | 0 | 1 | 3.14 | 0 | 1 | 3.35 | – | – | 3.07 | 0 | 2 |
| MO&AllAdHoc | 3.28 | 0 | 1 | 3.02 | 0 | 1 | 3.51 | – | – | 3.44 | 1 | 1 |
| SMOTEBoost | 3.02 | 0 | 1 | 3.04 | 0 | 1 | 3.79 | – | – | 3.63 | 1 | 0 |
| RAMOBoost | 3.24 | 0 | 1 | 3.20 | 0 | 1 | 3.92 | – | – | 4.10 | 2 | 0 |
| EasyEnsemble | 5.27 | 5 | 0 | 5.44 | 5 | 0 | 3.48 | – | – | 4.32 | 3 | 0 |

to imbalanced classification (**SMOTEBoost**, **RAMOBoost**, **EasyEnsemble**), considering the multi-objective generations with ensemble aggregation methods (**EltMO&AllAdHoc** and **StkMO&AllAdHoc**).

## V. DISCUSSION

In **Experiment 1** (Tables I, II and III), we see that **MO&AllAdHoc** has an overall better performance in all scenarios regardless of the imbalance level and number of classes, mainly considering the non-dominance Friedman test but never statistically worse than any other method in other metrics. Most of this behaviour can be explained by aggregating the classifiers from the ad-hoc cost-sensitive models (**CSMO**) and the models that take into account the conflict among all classes (**MO**). These models with a diversity of misclassification weights for each class are capable of creating high-quality performance classifiers, with no need for resampling, oversampling or other approaches. The use of misclassification weights, specially when sampled by multi-objective approaches, can deal with the raw data without excluding or artificially creating samples. This attribute might be considered the driving force of such performance.

In **Experiment 2** (Table IV), the importance of using Pareto-optimal misclassification weights is even more clear; **MO&AllAdHoc** is statistically better (in the non-dom Friedman test) than all other approaches except for **AllAdHoc**. **MO&AllAdHoc** had a great performance on all $F_\beta$ metrics, standing out as a good model no matter the scenario. This increase in performance, when compared with Experiment 1, along with the decrease in performance of **CSMO** and the non-significant improvement of **AllAdHoc**, highlights the relevance of multi-objective optimization. Exploring the conflict on the misclassification among the classes has a main role in this performance.

In Experiments 1 and 2 we can see that the best contender algorithms are EasyEnsemble for low imbalance, and SMOTEBoost for high imbalance. It can be explained by the fact that EasyEnsemble keeps the samples from minority classes and resample the majority classes to have the same number of samples – it performs well when there are enough samples from the minority class (with low imbalance level) but can be misleading with fewer samples. On the other hand, SMOTEBoost was capable of improving the performance with boosting methods but keeping the complexity and performance by over-sampling the minority class. Given that, we designed Experiment 3 to compare only the ensemble methods for imbalanced classification and confirm the quality of our proposal.

In **Experiment 3** (Table V), we can see that using multiple models generated with multi-objective optimization can also improve the classification performance, achieving a performance even better than **MO&AllAdHoc** and having a performance statistically better than all baseline ensembles. It shows that the framework also creates models with enough diversity to improve the performance.

Overall, it is possible to acknowledge that multi-objective optimization is quite effective in many ways. First of all, the simple manually adhoc weighted models (**StandardManual** and **CSManual**) are always defeated by their multi-objective trained counterparts (**StandardMO** and **CSMO**) and it can be generally explained by the better exploration of models against the grid-search. Moreover, despite the class-conflicting models (**MO**) not always generating the best performed classifiers, some of those models are successfully aggregated with more traditional ones to become the best strategy among the evaluated classifiers. It is explained by the high flexibility of those models (better seen in the multiple classes of Table III, which has an even poorer performance) thus not being capable of automatically finding the class weights employed in the traditional models. However, these traditional models are not always the best fit, and having these more flexible options clearly confers robustness to the performance.

## VI. Conclusion

This paper demonstrates that imbalanced classification tasks can be successfully solved when properly exploring two well-established tradeoffs: (1) Tradeoff established by the performance on each class, characterized by the inherent conflict among the losses produced by minority and majority classes on regularized multinomial logistic regression; (2) Tradeoff established by the performance on each criterion, given that there is an absence of a clear order when multiple performance criteria are considered. Both tradeoffs are explicitly modeled here under the perspective of multi-objective optimization.

The conflict among the class losses is conveniently modeled by a weighted sum formulation, including a regularization term, so that multiple efficient and diverse learning models are acquired when using a many-objective solver called MONISE. Given that MONISE is capable of automatically spreading the solutions all over the Pareto front, the diversity here is much more effective than the one possibly obtained by grid search. Those multiple efficient solutions produced by MONISE may be filtered to provide the best individual learning model (e.g. by the a posteriori preference of the decision maker), or may compose an ensemble, capable of exploring the existing diversity of candidate solutions. The higher the level of data imbalance, the more intense the gain in performance of our proposal. Accordingly, we clearly demonstrate that defining a priori ad-hoc weights for each class loss is not effective, when taken in isolation, but it tends to contribute in an ensemble formed by a diverse set of learning models characterized by efficient tradeoffs.

When evaluating the learning methods, given that multiple performance criteria (multiple metrics) are involved in imbalanced classification tasks, a new proposal is conceived to rank the candidate solutions, resorting to the non-dominance relation produced by the performance criteria when adopting the Friedman test. Therefore, we put down the rank of a learning method if there is another learning method with better performance for all metrics, thus providing a reliable ranking of the methods even when there is no clear preference among the metrics.

## Acknowledgment

We thank the São Paulo Research Foundation (FAPESP), Grants #14/13533-0 (FAPESP/CAPES agreement), and the National Council for Scientific and Technological Development (CNPq), Grant #307228/2018-5, for the financial support.

## Appendix

### A. Friedman test

The so-called Friedman test [36] is used to evaluate if there is a difference between $K$ treatments (here, classifiers) considering a set of $D$ trials (here, datasets) by assigning a rank for the classifiers, from the best (1, second best would be 2) to the worst ($K$) method. Mathematically, let's consider $p_{i,j} \in \mathbb{R}$ the performance of method $j \in \{1, \ldots, M\}$ in the dataset $i \in \{1, \ldots, D\}$. Given that, $|\mathcal{P}^{>}_{i,j}| = |x_{i,k} : x_{i,k} > x_{i,j} \; \forall k \in \{1, \ldots, K\}|$ is the number of methods better than $j$ for dataset $i$ and $|\mathcal{P}^{\equiv}_{i,j}| = |x_{i,k} : x_{i,k} = x_{i,j} \; \forall k \in \{1, \ldots, K\}|$ is the number of methods equivalent to $j$ for dataset $i$. Now, we can find the rank $r_{i,j}$ in Equation 5:

$$r_{i,j} = |\mathcal{P}^{>}_{i,j}| + \frac{|\mathcal{P}^{=}_{i,j}|^2 + |\mathcal{P}^{=}_{i,j}|}{4}, \tag{5}$$

the average Friedman rank in Equation 6:

$$\overline{r}_{\cdot j} = \frac{\sum_{i=1}^{D} \overline{r}_{i,j}}{D}, \tag{6}$$

and the $Q$ statistics in Equation 7:

$$Q = \frac{12D}{K(K+1)} \sum_{j=1}^{K} \left( \overline{r}_{\cdot j} - \frac{K+1}{2} \right)^2 \tag{7}$$

The test consists of evaluating how close the $Q$ statistics is to a $\chi^2$ distribution since the premise confirms whether the rank is, in fact, random.

### B. Friedman test with non-dominated sorting

On the other hand, when there is more than one evaluation metric, the ordering relation becomes more complex. To solve that, we resort to the dominance concepts, presented previously, and apply a non-dominated sorting [38]. This ordering method creates a set of Pareto fronts $\mathcal{P}^1, \ldots \mathcal{P}^F$, where the first front considers all non-dominated solutions, and the $f$-th front considers all solutions only dominated by the previous fronts. Mathematically, $x_i \in \mathcal{P}^f$ iff there exists $x_j \in \mathcal{P}^{f-1}$ that dominates $x_i$ and there is no $x_j \in \mathcal{P}^f \cup \mathcal{P}^F$ that dominates $x_i$.

Given that, let's consider $p_{i,j} \in \mathbb{R}^M$ the performance vector of method $j$ in the dataset $i$. We can redefine the number of methods better than $j$ for dataset $i$ as $|\mathcal{P}^{>}_{i,j}| = |x_{i,k} : \forall x_{i,k} \in \mathcal{P}^1 \cup \ldots \cup \mathcal{P}^{k-1} \wedge x_{i,j} \in \mathcal{P}^k|$, and the number of methods equivalent to $j$ for dataset $i$ as $|\mathcal{P}^{\equiv}_{i,j}| = |\mathcal{P}^k : x_{i,j} \in \mathcal{P}^k|$.

With those definitions, it is possible to use Equations 5, 6, 7 to calculate the Q-statistcs and test the hypothesis. However, we can anticipate the problem of too many ties in the non-dominated sorting so that the statistics might be hurt [39], thus we use the correction presented in Equation 8.

$$C = 1 - \sum_{f=1}^{F} \frac{|\mathcal{P}^f|^3 - |\mathcal{P}^f|}{N(K^3 - K)} \tag{8}$$

and compare $Q_c = Q/C$ to a $\chi^2$ distribution to test the hypothesis.

## References

[1] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of Imbalanced Data: A Review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[2] J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, "Diversity techniques improve the performance of the best imbalance learning ensembles," *Information Sciences*, vol. 325, pp. 98–117, 2015.

[3] J. Bradford, C. Kunz, R. Kohavi, C. Brunk, and C. Brodley, "Pruning decision trees with misclassification costs," *Machine Learning: ECML-98*, vol. 1398, pp. 131 – 136, 1998.

[4] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Machine Learning*, vol. 46, pp. 191–202, 2002.

[5] S. Datta and S. Das, "Near-Bayesian Support Vector Machines for imbalanced data classification with equal or unequal misclassification costs," *Neural Networks*, vol. 70, pp. 39–52, 2015.

[6] S. J. Stolfo, J. Zhang, and P. K. Chan, "AdaCost : Misclassification Cost-sensitive Boosting," in *International Conference on Machine Learning*, no. May, 1999.

[7] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTE-Boost : Improving Prediction," *Lecture Notes in Computer Science*, vol. 2838, pp. 107–119, 2003.

[8] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted F-measure and kernel scaling for imbalanced data learning," *Information Sciences*, vol. 257, pp. 331–341, 2014.

[9] N. V. Chawla, K. W. Bowyer, and L. O. Hall, "SMOTE : Synthetic Minority Over-sampling Technique," *Artificial Intelligence*, vol. 16, pp. 321–357, 2002.

[10] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adaptive Synthetic Sampling Approach for Imbalanced Learning," in *IEEE International Joint Conference on Neural Networks*, no. 3, 2008, pp. 1322–1328.

[11] H. Han, W.-y. Wang, and B.-h. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," in *International Conference on Intelligent Computing*, 2005, pp. 878–887.

[12] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi, "Automatically countering imbalance and its empirical relationship to cost," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 225–252, 2008.

[13] D. A. Cieslak, T. R. Hoens, N. V. Chawla, and W. P. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Mining and Knowledge Discovery*, vol. 24, no. 1, pp. 136–158, 2012.

[14] A. Roy, R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "A study on combining dynamic selection and data preprocessing for imbalance learning," *Neurocomputing*, vol. 286, pp. 179–192, 2018.

[15] M. A. Souza, G. D. Cavalcanti, R. M. Cruz, and R. Sabourin, "On evaluating the online local pool generation method for imbalance learning," in *International Joint Conference on Neural Networks*. IEEE, 2019, pp. 1–8.

[16] J. Hu, Y. Li, W. X. Yan, J. Y. Yang, H. B. Shen, and D. J. Yu, "KNN-based dynamic query-driven sample rescaling strategy for class imbalance learning," *Neurocomputing*, vol. 191, pp. 363–373, 2016.

[17] Y. Sun, A. Wong, and Y. Wang, "Parameter inference of cost-sensitive boosting algorithms," *Machine Learning and Data Mining in Pattern Recognition*, vol. 3587, no. July, pp. 21–30, 2005.

[18] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: Improving classification performance when training data is skewed," in *International Conference on Pattern Recognition*, 2008, pp. 1–4.

[19] S. Chen, H. He, and E. A. Garcia, "RAMOBoost: Ranked minority oversampling in boosting," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1624–1642, 2010.

[20] H. Guo and H. L. Viktor, "Learning from Imbalanced Data Sets with Boosting and Data Generation : The DataBoost-IM Approach," *ACM SIGKD Explorations Newsletter - Special issue on learning from imbalanced datasets*, vol. 6, no. 1, pp. 30–39, 2004.

[21] X.-y. Liu, J. Wu, and Z.-h. Zhou, "Exploratory Undersampling for Class Imbalance Learning," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 39, no. 2, pp. 539–550, 2009.

[22] J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, and L. I. Kuncheva, "Random Balance: Ensembles of variable priors classifiers for imbalanced data," *Knowledge-Based Systems*, vol. 85, pp. 96–111, 2015.

[23] J. J. Rodríguez, J.-F. Díez-Pastor, Á. Arnaiz-González, and L. I. Kuncheva, "Random Balance ensembles for multiclass imbalance learning," *Knowledge-Based Systems*, vol. 193, pp. 1–24, 2020.

[24] J. Li, S. Fong, R. K. Wong, and V. W. Chu, "Adaptive multi-objective swarm fusion for imbalanced data classification," *Information Fusion*, vol. 39, pp. 1–24, 2018.

[25] A. Fernandez, C. J. Carmona, M. J. Del Jesus, and F. Herrera, "A Pareto-based Ensemble with Feature and Instance Selection for Learning from Multi-Class Imbalanced Datasets," *International Journal of Neural Systems*, vol. 27, no. 6, pp. 1–21, 2017.

[26] P. Soda, "A multi-objective optimisation approach for class imbalance learning," *Pattern Recognition*, vol. 44, no. 8, pp. 1801–1810, 2011.

[27] A. Akan and S. Sayn, "SVM classification for imbalanced data sets using a multiobjective optimization framework," *Ann Oper Res*, vol. 216, pp. 191–203, 2014.

[28] S. Datta and S. Das, "Multiobjective Support Vector Machines: Handling Class Imbalance With Pareto Optimality," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1602–1608, 2019.

[29] P. Castillo, M. Arenas, J. Merelo, V. Rivas, and G. Romero, "Multiobjective optimization of ensembles of multilayer perceptrons for pattern classification," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4193 LNCS, pp. 453–462, 2006.

[30] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer, 1999.

[31] A. Jubril, "A nonlinear weights selection in weighted sum for convex multiobjective optimization," *Facta universitatis-series: Mathematics and Informatics*, vol. 27, no. 3, pp. 357–372, 2012.

[32] M. Raimundo, P. Ferreira, and F. Von Zuben, "An extension of the non-inferior set estimation algorithm for many objectives," *European Journal of Operational Research*, vol. 284, pp. 53–66, 2020.

[33] M. M. Raimundo and F. J. Von Zuben, "Investigating multiobjective methods in multitask classification." in *International Joint Conference on Neural Networks*, 2018, pp. 1–9.

[34] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[35] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification." *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics*, vol. 42, no. 2, pp. 513–29, 2012.

[36] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[37] H. Finner, "On a Monotonicity Problem in Step-Down Multiple Test Procedures," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 920–923, 1993.

[38] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[39] D. J. Sheskin, *Parametric and Nonparametric Statistical Procedures*. Chapman and Hall/CRC, 2000.