Expert Systems With Applications 147 (2020) 113232

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Ensemble learning by means of a multi-objective optimization design approach for dealing with imbalanced data sets

Victor Henrique Alves Ribeiro*, Gilberto Reynoso-Meza

Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifícia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, PR, Brazil

ARTICLE INFO

Article history:
Received 30 March 2019
Revised 25 December 2019
Accepted 21 January 2020
Available online 22 January 2020

Keywords:
Ensemble learning
Multi-objective optimization
Imbalanced data sets

ABSTRACT

Ensemble learning methods have already shown to be powerful techniques for creating classifiers. However, when dealing with real-world engineering problems, class imbalance is usually found. In such scenario, canonical machine learning algorithms may not present desirable solutions, and techniques for overcoming this problem must be used. In addition to using learning algorithms that alleviate the imbalance between classes, multi-objective optimization design (MOOD) approaches can be used to improve the prediction performance of ensembles of classifiers. This paper proposes a study of different MOOD approaches for ensemble learning. First, a taxonomy on multi-objective ensemble learning (MOEL) is proposed. In it, four types of existing approaches are defined: multi-objective ensemble member generation, multi-objective ensemble member selection, multi-objective ensemble member combination, and multi-objective ensemble member selection and combination. Additionally, new approaches can be derived by combining the previous ones, such as multi-objective ensemble member generation and selection, multi-objective ensemble member generation and combination and multi-objective ensemble member generation, selection and combination. With the given taxonomy, two experiments are conducted for comparing (1) the performance of the MOEL techniques for generating and aggregating base models on several imbalanced benchmark problems and (2) the performance of MOEL techniques against other machine learning techniques in a real-world imbalanced drinking-water quality anomaly detection problem. Finally, results indicate that MOOD is able to improve the predictive performance of existing ensemble learning techniques.

1. Introduction

Ensemble learning methods are powerful techniques that combine the outputs of multiple models for creating a final prediction, and it is well known that such techniques outperform the performance of monolithic models (Sagi & Rokach, 2018). Thus, researchers employed ensemble models to solve many different problems, such as computer security, fraud detection, recommender systems, medicine and remote sensing (Woźniak, Graña, & Corchado, 2014). Despite this, such models are usually trained by optimizing a single metric, such as the global error or loss.

According to Krawczyk (2016), canonical learning algorithms consider the number of instances in each class to be roughly similar. However, when dealing with imbalanced data sets, the optimization of a single metric may not indicate the true performance of a predictive model. For instance, in a scenario where 99% of the data set is composed of one class and only 1% of the other, guessing all instances as the majority class produces an accuracy of 99%. In addition to this, it is common that the minority class is the most important in such scenarios. Therefore, it is necessary to overcome this problem with the development of solutions for dealing specifically with class imbalance (Douzas & Bacao, 2018; Lemaître, Nogueira, & Aridas, 2017; Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2010).

The application of multi-objective optimization (MOO) can improve the predictive performance on a high imbalance scenario, since multiple objectives are defined and optimized, such as class-specific errors. MOO has already shown to be a useful tool for designing ensemble models. A research by Ribeiro and Reynoso-Meza (2019a) reviews several applications, such as the generation of base learners (Peimankar, Weddell, Jalal, & Lapthorn, 2017; Rosales-Perez, Garcia, Gonzalez, Coello, & Herrera, 2017; Rosales-Pérez, Gonzalez, Coello, Escalante, & Reyes-Garcia, 2014; Smith & Jin, 2014; Tan, Lim, & Cheah, 2014), the selection of models from

* Corresponding author.
E-mail addresses: victor.henrique@pucpr.edu.br (V.H. Alves Ribeiro), g.reynosomeza@pucpr.br (G. Reynoso-Meza).

https://doi.org/10.1016/j.eswa.2020.113232
0957-4174/

a pool (Peimankar et al., 2017; 2018) and the weighting of the base models (Zhao et al., 2018). MOO is used in complex scenarios, where multiple conflicting objectives must be optimized. Even though there exist different evaluation metrics for imbalanced learning, they are also often conflicting, such as F-measure and G-mean. Therefore, techniques where a proper trade-off between such conflicting objectives can be obtained should be used.

However, it is important to understand where MOO can be applied inside the ensemble learning methodology to achieve the best results. With this in mind, this paper presents an extensive study on the application of multi-objective optimization design (MOOD) procedures for the design of ensemble learning models, where optimization is used for generating, selecting and combining pools of base learners. Moreover, producing robust models that deal with complex problems is of great importance. To this end, experiments are conducted on imbalanced data sets designated for binary classification problems to answer the following question:

- How can ensemble learning methodologies take benefit from MOOD approaches?

Results on the proposed experiments indicate that the application of MOOD procedures for selecting and combining base models improves the predictive performance of ensembles. Results also show that MOOD produces proper pools of classifiers when feature and instance selection is performed on data sets with smaller numbers of features (less than 50) and instances (less than 10,000). As a conclusion, the following contributions are highlighted:

- Proposal of multi-objective ensemble learning (MOEL), a taxonomy for the application of MOO in the different steps of the ensemble learning methodology.
- Benchmark experiments using different techniques for pool generation and aggregation methods indicate where MOO can be applied for improving the performance of ensemble models.
- MOEL is used for solving a real-world industrial challenge regarding drinking-water quality anomaly detection, achieving better results than the competition's winner.

The remainder of this paper is organized as follows: Section 2 details the problems related to imbalanced data sets; Section 3 indicates ensemble learning methodologies for dealing with imbalance in data sets, along with the proposal of applying MOOD for improving predictive performance in such scenarios; Section 4 brings two experiments for MOOD in ensemble learning. In the first case study, tests are performed on 100 imbalanced data sets. In the second study, a real-world drinking-water quality anomaly detection problem is approached with the presented techniques. Section 5 discusses the results from the previous experiments. Finally, the work is concluded with some final remarks and future research.

2. The problem of imbalanced data sets

When dealing with imbalanced problems, canonical machine learning algorithms tend to be biased towards the majority group. This happens because such algorithms consider the number of objects in each group to be roughly similar. However, the minority class is usually the most important when dealing with skewed distributions, and intelligent systems must be designed to overcome such bias (Krawczyk, 2016). The importance of such problem has been confirmed by the high number of research and books on the subject, such as Fernández et al. (2018).

Many studies deal with imbalanced learning (Singh & Purohit, 2015). To tackle imbalanced data problems, three approaches are identified by Krawczyk (2016): data-level, algorithm-level, and hybrid methods. When using data-level methods, the training data set is balanced by using techniques to remove instances from the majority class (undersampling) or create new instances in the minority class (oversampling). It is stated that such tasks are usually performed randomly. However, more advanced techniques must be used to keep important instances or add meaningful new data. Algorithm-level methods focus on modifying existing learning algorithms to remove the bias on majority groups. This is usually done by introducing a cost-sensitive approach, where each class has a different penalty for computing the learner's loss function. Another approach is related to one-class learning, where only a target group is selected. Hybrid methods, as the name suggests, combine the two previously mentioned methods.

In addition to the previous techniques, Haixiang et al. (2017) presents some important insights, such as hybrid sampling and feature selection. Hybrid sampling is a combination of oversampling and undersampling. Feature selection, despite being used few times in literature, can help models improve predictive performance by removing irrelevant features, reducing the risk of discarding important instances that could be considered noisy.

According to Krawczyk (2016), binary classification is considered the most developed branch when dealing with imbalanced data sets. Learning issues in classification tasks are not caused only by the imbalance ratio, but also by the existence of difficult instances (mainly in the minority group), such as overlapping distributions and outliers. An analysis of the minority class instances' neighborhood is suggested for understanding learning issues.

Krawczyk (2016) indicates several future research directions that can be followed to help improve the classification of imbalanced data sets. For instance, ensemble learning techniques are currently one of the most popular approaches for dealing with imbalanced problems. However, there are still some drawbacks. It is desirable to understand how diversity measures can be used when dealing with imbalance, since the minority class is usually intact in current techniques. It is also important to understand the relation between imbalance ratio and ensemble size. With that in mind, ensemble pruning techniques dedicated for imbalanced problems should be developed. Also, the combination step in ensemble learning could exploit the individual qualities of its base learners.

To solve the imbalanced learning problem, this manuscript focus on the application of a famous ensemble learning technique, random forest (RF), an imbalanced learning specific technique, random undersampling boosting (RUSBoost), and the application of MOOD in the different steps of ensemble learning. The following section presents such approaches.

3. Ensemble methodologies for imbalanced data sets

To deal with imbalanced data sets, ensemble learning models have been applied with success in many different applications (Haixiang et al., 2017; He, Zhang, & Zhang, 2018). Re-sampling ensemble techniques are some of the most used in such cases. To compare the application of a MOO approach for designing ensemble models, the following techniques are presented: RF, a classic ensemble learning model, but not designed for imbalanced problems; RUSBoost, a boosting algorithm specifically designed for dealing with imbalanced data sets; and MOEL, a collection of different approaches for designing ensemble models using MOO.

3.1. Random forests

RF has been presented by Breiman (2001), being a classical ensemble learning algorithm. It is a noise robust technique, but has not been designed to deal with problems with imbalanced data sets. Despite this, such method is used in this work to analyze the effects of MOO when dealing with such type of problems.

The RF algorithm builds a set of decision trees, each one using a different random subset of the original instances of the problem. This is similar to the bootstrap aggregating (bagging) methodology (Breiman, 1996). However, the RF also uses random sets of features at each node for growing the decision trees. Such methodology enables the ensemble to have better accuracy and noise robustness.

3.2. RUSBoost

The RUSBoost is an algorithm specifically designed for dealing with imbalanced data sets (Seiffert et al., 2010). It is an adaptation of the adaptive boosting (AdaBoost) algorithm (Freund & Schapire, 1997), where random under-sampling is used for balancing the training set of the base classifiers. According to Haixiang et al. (2017), this can be considered a technique that performs preprocessing for improving predictive performance on imbalanced problems.

3.3. Multi-objective optimization design in ensemble learning

Much work has been performed for using MOO in machine learning (Al-Sahaf et al., 2019; Alexandropoulos, Aridas, Kotsiantis, & Vrahatis, 2019). The application of MOO techniques for generating ensembles has been presented by Gu, Cheng, and Jin (2015) as multi-objective ensemble generation (MOEG). However, such taxonomy is used for the generation of base learners by means of MOO. A review on recent applications is presented in Ribeiro and Reynoso-Meza (2019a), where it is indicated that MOO can be applied in three different steps of the ensemble learning methodology. Such steps are: member, or pool, generation; member selection, or pruning; and member combination, or weighting.

With such, this paper proposes a new taxonomy for the field of study, namely MOEL. Also, based on the step of the ensemble learning methodology where MOO is applied, this work proposes the following taxonomy for the different types of MOEL:

- Multi-objective ensemble member generation (MOEMG);
- Multi-objective ensemble member selection (MOEMS);
- Multi-objective ensemble member combination (MOEMC).

where additional hybrid models can be formed by combining the previous types:

- Multi-objective ensemble member selection and combination (MOEMSC);
- Multi-objective ensemble member generation and selection (MOEMGS);
- Multi-objective ensemble member generation and combination (MOEMGC);
- Multi-objective ensemble member generation, selection and combination (MOEMGSC).

The listed approaches are detailed in the following section. In all cases, the MOO step is performed according to Section 3.3.8. Finally, to select a preferred non-dominated ensemble from the selection and combination tasks, a multi-criteria decision making (MCDM) step is performed according to Section 3.3.9.

For all of the following approaches, false negative ratio (FNR) and false positive ratio (FPR) are selected as the classification metrics to be minimized in the multi-objective problems (MOPs). FPR and FNR are computed according to Eqs. 1 and 2, where false positives (FP) is the number of negative instances incorrectly classified as positive, false negatives (FN) is the number of positive instances incorrectly classified as negative, true negatives (TN) is the number of correctly classifier negative instances, and true positives (TP) is the number of correctly classifier positive instances. Such objectives can be viewed as conflicting class-specific errors. Thus, by minimizing FNR and FPR, a more complete Pareto front is obtained through MOO.

$$FPR = FP/(FP + TN) \tag{1}$$

$$FNR = FN/(FN + TP) \tag{2}$$

3.3.1. Multi-objective ensemble member generation

The first type is also the most commonly found in literature, being used for time series forecasting (Bui, Dinh et al., 2018; Smith & Jin, 2014), remaining useful life estimation (Zhang, Lim, Qin, & Tan, 2016), imbalanced data classification (Fernández, García, del Jesus, & Herrera, 2008), image classification (Albukhanajer, Jin, & Briffa, 2017), and fault diagnosis (Ma & Chu, 2019). In it, a set of non-dominated models is generated through MOO, and the models are combined to build an ensemble. The main goal is to build strong and diverse base models to form an ensemble.

For this task, many different objectives have been defined in the MOP formulation, such as predictive performance metrics, diversity and complexity metrics. Also, different decision variables can be used, such as the selection of features and instances, or hyper-parameters tuning. In this work, classification scores and complexity measures are minimized for creating a pool of classifiers, which is done through feature and instance selection. Thus, the MOP for the MOEMG is defined as follows:

$$\min_{\boldsymbol{g}} \boldsymbol{J}(\boldsymbol{g}) = [J_{fpr}(\boldsymbol{g}), J_{fnr}(\boldsymbol{g}), J_f(\boldsymbol{g}), J_i(\boldsymbol{g})] \tag{3}$$

subject to

$$\boldsymbol{g} = [\boldsymbol{f}, \boldsymbol{i}] \tag{4}$$

$$f_a \in \{0, 1\}, a \in \left[1, \ldots, n_f\right] \tag{5}$$

$$i_b \in \{0, 1\}, b \in [1, \ldots, n_i] \tag{6}$$

where the objectives are

- $J_{fpr}(\boldsymbol{g})$: The FPR, which indicates the ratio of real negative instances that are incorrectly classified as positive;
- $J_{fnr}(\boldsymbol{g})$: The FNR, which indicates the ratio of real positive instances that are incorrectly classified as negative;
- $J_f(\boldsymbol{g})$: Number of selected features, which can be considered as a complexity measure;
- $J_i(\boldsymbol{g})$: Number of selected instances, which can also be considered as a complexity measure.

while the decision variables are

- f_a: Binary selection of each of the n_f features from the data set;
- i_b: Binary selection of each of the n_i instances from the data set.

An overview of the MOEMG method is illustrated in Fig. 1. In the MOP, a model's predictive performance is evaluated on a validation data set. The predictive model is optimized by a MOO algorithm through feature, instance or hyper-parameter selection using a learning algorithm and a training data set. Different from the following methods, where MCDM is used to select a preferable single predictive model, the final solution is composed of the full set of non-dominated base models. If no additional method is used for combining or selecting the ensemble's base models, simple majority voting (Sagi & Rokach, 2018) is used.

3.3.2. Multi-objective ensemble member selection

In MOEMS, the selection of the base models from a pool of classifiers is performed using MOO. In this task, the goal is to build a strong ensemble by pruning base models that could interfere in

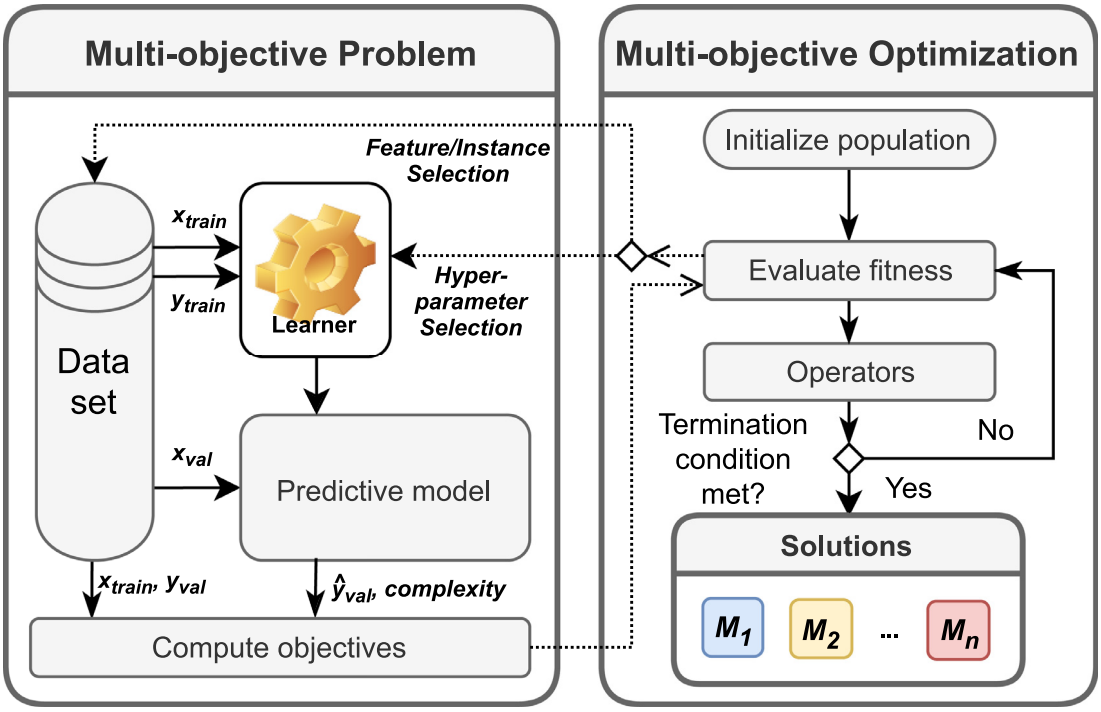

Fig. 1. The Multi-objective ensemble member generation (MOEMG) approach.

the predictive performance. Also, the decision variable is usually a binary vector for selecting or not the base models, as designed for solving problems of sentiment analysis (Onan, Korukoğlu, & Bulut, 2017) and power transformers' dissolved gas contents prediction (Peimankar, Weddell, Jalal, & Lapthorn, 2018). In this work, classification scores and the number of models are minimized for building such ensembles, being the MOP defined as follows:

$$\min_{\boldsymbol{m}} \boldsymbol{J}(\boldsymbol{m}) = [J_{fpr}(\boldsymbol{m}), J_{fnr}(\boldsymbol{m}), J_m(\boldsymbol{m})] \tag{7}$$

subject to

$$m_i \in \{0, 1\}, i \in [1, \ldots, n_m] \tag{8}$$

where the new objective and decision variables are

- $J_m(\boldsymbol{m})$: Ratio of selected base models (r_m), computed according to Eq. 9;
- m_i: Binary selection of each of the n_m base models from the pool.

$$r_m = \sum_{i=1}^{n_m} m_i / n_m \tag{9}$$

An overview of the MOEMS method is illustrated in Fig. 2. In the MOP, a pool of base learners is used to predict outputs from a validation data set, and the pruning of the pool is performed to optimize the prediction score by a MOO algorithm. After the optimization, the MCDM step performs the ranking and selection of a preferable set of models.

3.3.3. Multi-objective ensemble member combination

In MOEMC, the weighting of the base models in a weighted majority voting scheme is performed using MOO. MOEMC has been employed for sentiment analysis (Onan, Korukoğlu, & Bulut, 2016) and wind speed forecasting (Liu, Duan, Li, & Lu, 2018; Qu et al., 2017). In this approach, the goal is to build a strong ensemble by adjusting the weights of the base models to improve the predictive performance of the whole ensemble. This work minimizes the classification scores and the total sum of weights to build stronger ensembles, being the MOP defined as follows:

$$\min_{\boldsymbol{w}} \boldsymbol{J}(\boldsymbol{w}) = [J_{fpr}(\boldsymbol{w}), J_{fnr}(\boldsymbol{w}), J_w(\boldsymbol{w})] \tag{10}$$

subject to:

$$0 \leq w_i \leq 1, i \in [1, \ldots, n_m] \tag{11}$$

where the new objective and decision variables are

- $J_w(\boldsymbol{w})$: Mean of the applied weights ($\overline{w}$), computed according to Eq. 12;
- w_i: Voting weight applied to each of the n_m base models from the pool.

$$\overline{w} = \sum_{i=1}^{n_m} w_i / n_m \tag{12}$$

An overview of the MOEMC method is illustrated in Fig. 3. In the MOP, a pool of base learners is used to predict outputs from a validation data set, and the weights are adjusted to optimize the prediction score by a MOO algorithm. After the optimization, the MCDM step performs the ranking and selection of a preferable set of weights.

3.3.4. Multi-objective ensemble member selection and combination

The MOEMSC task is a combination of MOEMS and MOEMC, where both the selection and weighting of the base models are optimized simultaneously to create ensembles. To this end, the decision variables from both MOEMS and MOEMC tasks are used, while the prediction scores and sum of weights from the selected models are minimized. Thus, the MOP formulation for such task can be performed as follows:

$$\min_{\boldsymbol{mw}} \boldsymbol{J}(\boldsymbol{mw}) = [J_{fpr}(\boldsymbol{mw}), J_{fnr}(\boldsymbol{mw}), J_{mw}(\boldsymbol{mw})] \tag{13}$$

subject to

$$\boldsymbol{mw} = [\boldsymbol{m}, \boldsymbol{w}] \tag{14}$$

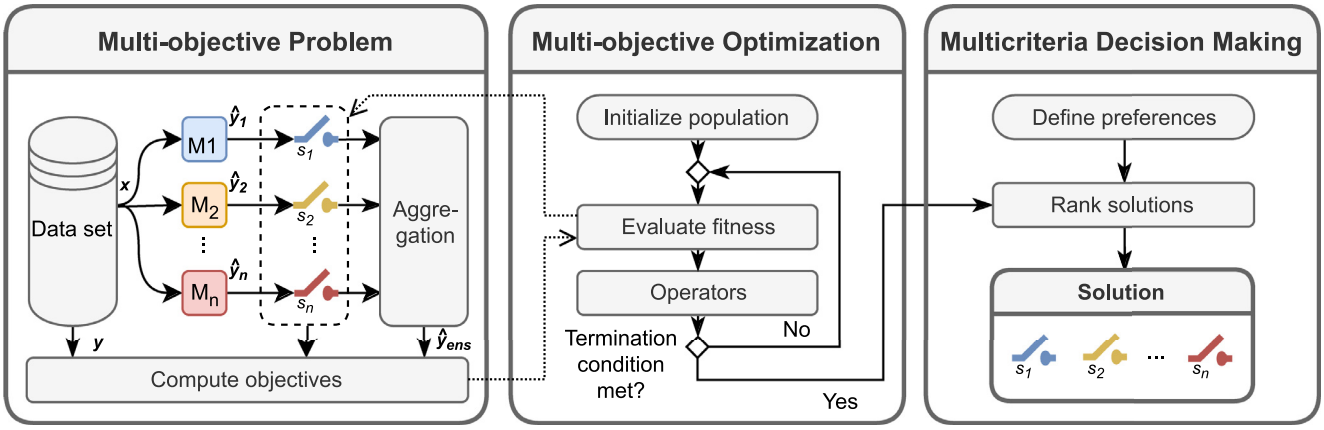

Fig. 2. The Multi-objective ensemble member selection (MOEMS) approach.

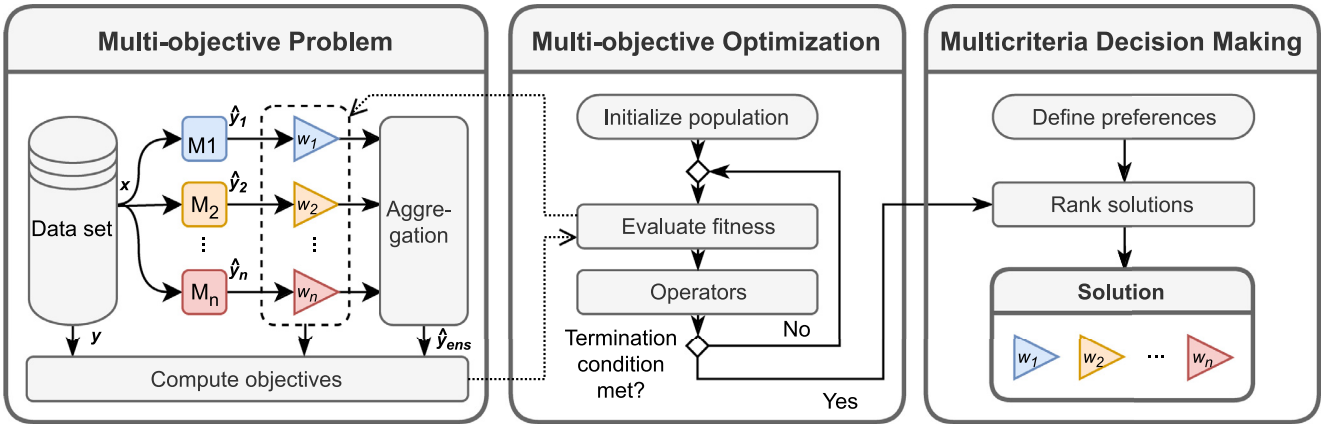

Fig. 3. The Multi-objective ensemble member combination (MOEMC) approach.

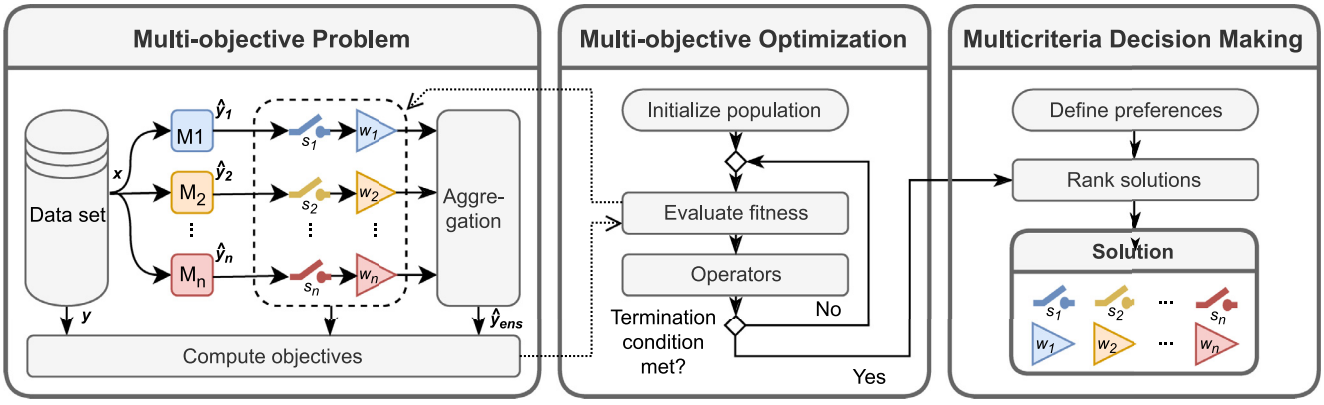

Fig. 4. The Multi-objective ensemble member selection and combination (MOEMSC) approach.

where the new objective ($J_{mw}(\boldsymbol{mw})$) is the mean of used weights ($\overline{mw}$), computed according to Eq. 15. The decision variables ($\boldsymbol{mw}$) are a concatenation of the selection of each model (m_i) and their weights (w_i), presented in the previous approaches (MOEMS and MOEMC).

$$\overline{mw} = \sum_{i=1}^{n_m} m_i \cdot w_i / n_m \tag{15}$$

An overview of the MOEMSC method is illustrated in Fig. 4. In the MOP, a pool of base learners is used to predict outputs from a validation data set, and the pruning and weighting of the pool is performed to optimize the prediction score by a MOO algorithm. After the optimization, the MCDM step performs the ranking and selection of a preferable set of weights and selected models.

3.3.5. Multi-objective ensemble member generation and selection

The combination of MOEMG and MOEMS has been presented in Ribeiro and Reynoso-Meza (2018a, 2018b, 2019a) for detecting events in water quality. Such procedure has also been employed for fault detection (Peimankar et al., 2017) and wind power forecasting (Hao & Tian, 2019). This combination is designated as MOEMGS, and is composed of two MOPs. Both problems are optimized by means of MOO, and a MCDM step is performed for selecting a final solution.

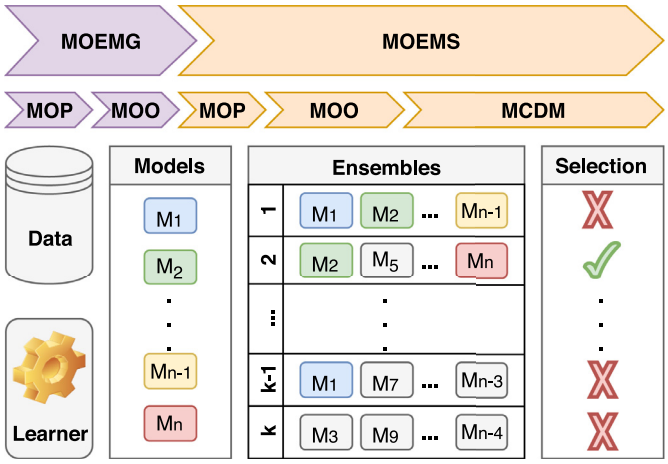

Fig. 5. The Multi-objective ensemble member generation and selection (MOEMGS) approach.

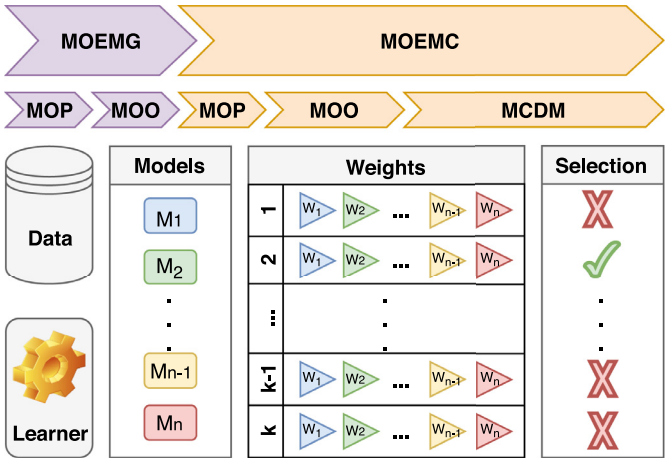

Fig. 6. The Multi-objective ensemble member generation and combination (MOEMGC) approach.

An overview of the methodology is illustrated in Fig. 5. In the MOEMG step, a MOP is designed with a training data and a learning algorithm, and the MOO results in a set of non-dominated base models. In the MOEMS step, the MOP uses such base models and prunes them in the MOO stage, resulting in a set of non-dominated ensembles. A MCDM step performs the selection of a final preferred ensemble.

3.3.6. Multi-objective ensemble member generation and combination

To the best of the authors' knowledge, the combination of MOEMG and MOEMC has still not been performed in literature. Such combination is designated as MOEMGC, and is composed of two MOPs. Both problems are optimized by means of MOO, and a MCDM step is performed for selecting a final solution.

An overview of the methodology is illustrated in Fig. 6. In the MOEMG step, a MOP is designed with a training data and a learning algorithm, and the MOO results in a set of non-dominated base models. In the MOEMC step, the MOP selects the weights of each model in the MOO stage, resulting in a set of non-dominated ensembles. A MCDM step performs the selection of a final preferred ensemble.

3.3.7. Multi-objective ensemble member generation, selection and combination

To the best of the authors' knowledge, the combination of MOEMG and MOEMSC has still not been performed in literature.

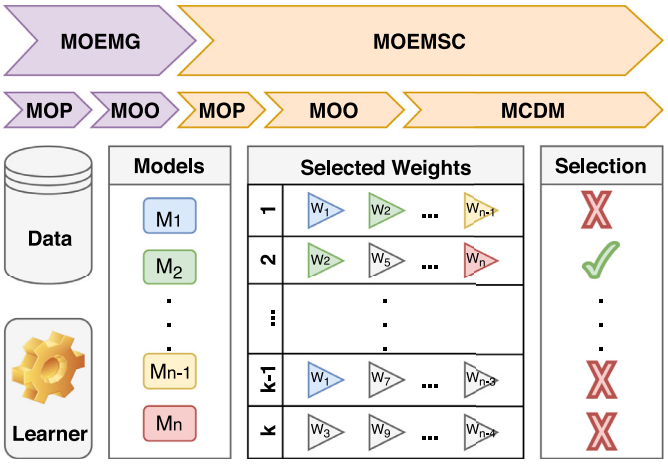

Fig. 7. The Multi-objective ensemble member generation, selection and combination (MOEMGSC) approach.

Such combination is designated as MOEMGSC due to the fact that all the steps of the ensemble learning methodology are performed using MOO. It is composed of two MOPs, where both problems are optimized by means of MOO. Subsequently, a MCDM step is performed for selecting a final solution.

An overview of the methodology is illustrated in Fig. 7. In the MOEMG step, a MOP is designed with a training data and a learning algorithm, and the MOO results in a set of non-dominated base models. In the MOEMSC step, the MOP selects the weights of each model and prunes them in the MOO stage, resulting in a set of non-dominated ensembles. A MCDM step performs the selection of a final preferred ensemble.

3.3.8. Multi-objective optimization

All of the MOPs are optimized with the multi-objective differential evolution with spherical pruning (spMODE-II) algorithm (Reynoso-Meza, Blasco, Sanchis, & Martínez, 2014a), which, as indicated in Ribeiro and Reynoso-Meza (2019a), presents desirable convergence, diversity and pertinence characteristics. Such algorithm has shown good results when dealing with problems composed of three and more objectives, since it is embedded with a preference handling mechanism (Reynoso-Meza, Sanchis, Blasco, & García-Nieto, 2014b; Ribeiro & Reynoso-Meza, 2018b; 2019a). The MOO algorithm is configured with a crossover rate of 0.2, a scaling factor of 0.5, a population of 20 individuals, and 100 generations, resulting in a total of 2,000 function evaluations.

3.3.9. Multi-criteria decision making

To select a preferred ensemble from the set of non-dominated solutions, MCDM is used. The preference ranking organization method for enriched evaluation (PROMETHEE) (Behzadian, Kazemzadeh, Albadvi, & Aghdasi, 2010; Brans, Vincke, & Mareschal, 1986) is selected for such task. Such technique follows a step-wise procedure.

1. Pairwise differences between each solution are computed for all objectives.
2. A preference function, based on the significance and insignificance levels, is applied.
3. The overall preference index is computed by performing a weighted sum of the objectives for each pairwise comparison.
4. Positive and negative outranking flows are calculated for each solution, based on the overall pairwise comparisons.
5. The positive and negative outranking flows are subtracted to compute a final outranking flow, used for ranking the solutions.

Table 1
Matrix with (in)significant differences. Significant (S) and Insignificant (I) differences for each design objectives are defined.

I/S differences matrix		
Objective	I	S
$J_{fpr}(\boldsymbol{x})$ (-)	0.01	0.05
$J_{fnr}(\boldsymbol{x})$ (-)	0.01	0.05
$J_{complexity}(\boldsymbol{x})$ (-)	0.01	0.05

Table 1 presents the (in)significant differences defined for the experiments in this work, where $J_{complexity}(\boldsymbol{x})$ varies according to the selected MOEL approach: $J_m(\boldsymbol{x})$ for MOEMS, $J_w(\boldsymbol{x})$ for MOEMC, and $J_{mw}(\boldsymbol{x})$ for MOEMSC. The significant and insignificant values of 0.05 and 0.01 indicate differences of 5% and 1% of the total objective values, respectively.

4. Experiments and results

This section proposes two experiments where MOEL is employed. In the first one, 100 data sets from the knowledge extraction based on evolutionary learning (KEEL) data set repository (Fernández et al., 2008) are used for performing statistical comparison of different ensemble learning techniques. The second experiment is composed of a real-world drinking-water quality anomaly detection problem, originated from an industrial challenge (Rehbach, Chandrasekaran, Rebolledo, Moritz, & Bartz-Beielstein, 2018a).

4.1. Experiment on imbalanced benchmark data sets

The first experiment is intended for performing statistical comparison of the presented MOEL approaches with existing ensemble methods. To this end, 100 imbalanced binary classification data sets from the KEEL repository are used (Fernández et al., 2008). Table 2 details the characteristics of the used data sets with the number of features (n_f), number of instances (n_i) and imbalance ratio (IR). The latter is computed according to Eq. 16. As detailed in Table 2, the number of features ranges from 3 to 41, the number of features ranges from 92 to 5472, and the imbalance ratio ranges from 1.8 to 129.4.

$$IR = \frac{\text{number of majority class instances}}{\text{number of minority class instances}} \tag{16}$$

While MOEMG, RF, and RUSBoost are used for pool generation, MOEMS, MOEMC, MOEMSC, and the original RF and RUSBoost combination methods are used for pool aggregation. For comparison purposes with the MOEMG, which uses 2000 function evaluations, the same number of base classifiers could be used by RF and RUSBoost. However, only 200 base classifiers are used. This number follows a recent research that compares pool generation methods (Ribeiro & Reynoso-Meza, 2019b). It has been shown that a lower number of base classifiers does not affect its performance, when being compared to the MOOD approach. Nevertheless, it is actually less computationally expensive to do so.

For the experiments with each of the 100 data sets, 5-fold cross validation is used. Additionally, the training folds are split in half for training and optimizing the models where MOO is employed. The results for the experiment are analyzed with FNR and FPR, since these are the two conflicting classification metrics used for optimization, and the F-measure and G-means (Japkowicz, 2013), specific metrics for assessing imbalanced learning problems. The first one is the harmonic mean between precision and recall, while the other is the geometric mean between specificity and recall or precision and recall. The F-measure and G-means are computed as follows:

$$\text{F-measure} = 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall}) \tag{17}$$

$$\text{G-mean}_1 = \sqrt{\text{Specificity} \cdot \text{Recall}} \tag{18}$$

$$\text{G-mean}_2 = \sqrt{\text{Precision} \cdot \text{Recall}} \tag{19}$$

where

$$\text{Precision} = TP/(TP + FP) \tag{20}$$

Table 2
Number of features (n_f), number of instances (n_i), and imbalance ratio (IR) of each imbalanced binary classification data set collected from the knowledge extraction based on evolutionary learning (KEEL) data set repository (Fernández et al., 2008).

Data set	n_f	n_i	IR	Data set	n_f	n_i	IR	Data set	n_f	n_i	IR	Data set	n_f	n_i	IR
1	9	214	1.8	26	7	202	9.1	51	7	459	14.3	76	11	168	32.6
2	7	220	1.9	27	9	172	9.1	52	9	214	15.5	77	8	1484	32.7
3	9	683	1.9	28	8	506	9.1	53	7	336	15.8	78	6	2935	35.2
4	8	768	1.9	29	8	1004	9.1	54	10	472	15.9	79	11	656	35.4
5	4	150	2.0	30	8	1004	9.1	55	8	731	16.4	80	7	281	39.1
6	9	214	2.1	31	6	203	9.2	56	34	358	16.9	81	8	2338	39.3
7	8	1484	2.5	32	7	244	9.2	57	16	101	19.2	82	8	581	40.5
8	3	306	2.8	33	7	224	9.2	58	9	184	19.4	83	8	1484	41.4
9	18	846	2.9	34	9	92	9.2	59	9	129	20.5	84	11	900	44.0
10	18	846	2.9	35	7	205	9.3	60	9	230	22.0	85	11	855	46.5
11	18	846	3.0	36	7	257	9.3	61	8	693	22.1	86	41	1061	49.5
12	9	214	3.2	37	8	528	9.4	62	9	214	22.8	87	8	1622	49.7
13	18	846	3.3	38	13	988	10.0	63	8	482	23.1	88	6	1460	53.1
14	7	336	3.4	39	6	220	10.0	64	18	148	23.7	89	11	1482	58.3
15	5	215	5.1	40	9	192	10.3	65	11	1066	23.8	90	10	1485	58.4
16	5	215	5.1	41	7	336	10.6	66	6	1728	24.0	91	9	3316	66.7
17	7	336	5.5	42	7	443	11.0	67	6	1728	25.6	92	11	691	68.1
18	19	2308	6.0	43	6	240	11.0	68	6	2901	26.6	93	8	1916	72.7
19	9	214	6.4	44	9	108	11.0	69	6	2244	27.8	94	41	2233	73.4
20	8	1484	8.1	45	9	205	11.1	70	8	1484	28.1	95	41	1610	75.7
21	7	336	8.6	46	9	214	11.6	71	11	1599	29.2	96	6	2193	80.2
22	10	5472	8.8	47	6	332	12.3	72	10	244	29.5	97	10	2075	82.0
23	7	200	9.0	48	13	173	12.3	73	41	1642	30.0	98	10	1477	85.9
24	8	514	9.1	49	6	280	13.0	74	8	947	30.6	99	41	2225	100.1
25	7	222	9.1	50	9	1829	13.9	75	8	502	32.5	100	8	4174	129.4

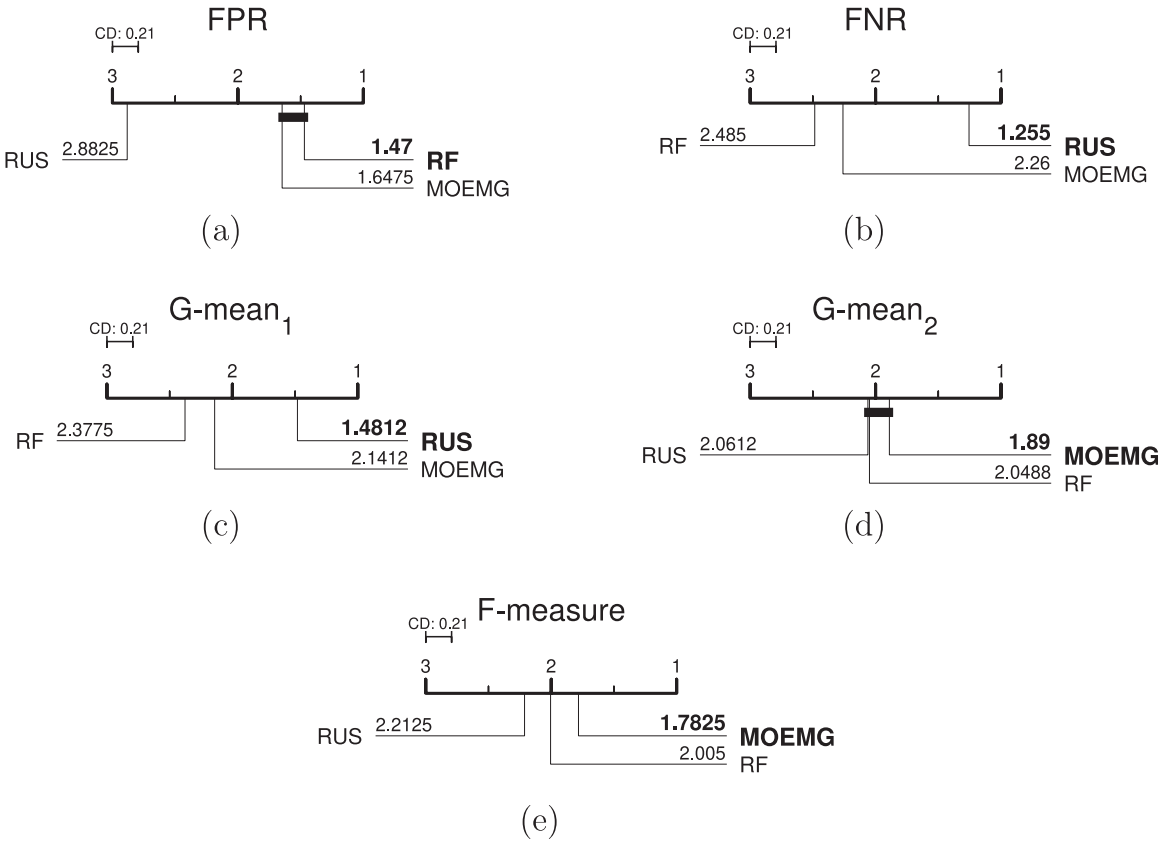

Fig. 8. The pool generation methods' critical differences (CD) plots for the five metrics: (a) false positive ratio (FPR), (b) false negative ratio (FNR), (c) G-mean$_1$, (d) G-mean$_2$, and (e) F-measure.

$$\text{Recall} = TP/(TP+FN) \tag{21}$$

$$\text{Specificity} = TN/(TN+FP) \tag{22}$$

Finally, the critical differences (CD) plot (Demšar, 2006) is used for comparing the results of the pool generation and aggregation methods. Such plot is designed after executing the non-parametrical Friedman's test (Pereira, Afonso, & Medeiros, 2015), which tests the null hypothesis that no differences are found among multiple methods in dependent samples, and the Nemenyi's post-hoc test (Pereira et al., 2015), which is employed for finding the differences among the tested methods.

The results from this experiment can be found in Section 4.1.1, 4.1.2, and 4.1.3. Section 4.1.1 and 4.1.2 use five CD plots to visualize the mean ranks achieved by the tested pool generation and pool aggregation methods for each metric (FPR, FNR, G-mean$_1$, G-mean$_2$, and F-measure), respectively. Section 4.1.3 details the number of wins for the five metrics by each tested combination of pool generation and aggregation methods. Also, a correlation plot between the data sets' characteristics and the metrics attained by each combination is presented. In the results below, RUSBoost is simplified as RUS for visualization purposes.

4.1.1. Pool generation results

In Fig. 8, all the plots for the pool generation methods show a CD value of 0.21, which represents the minimal statistically significant difference of mean ranks. This value is reached by using a significance level of 0.01. Also, scales show values between 1 and 3, representing the number of compared methods (RUSBoost, RF, and MOEMG). Below the scale, the methods are shown along with their mean ranks. Moreover, a thick line connects the methods with no statistically significant differences.

For the FPR, RF accomplishes the best mean rank (1.47) along with MOEMG (1.6475), followed by RUSBoost (2.8825). Differently, RUSBoost attains the best mean rank (1.255) for FNR, followed by MOEMG (2.26) and RF (2.485). The same order is found for the G-mean$_1$, where RUSBoost produces a mean rank of 1.4812, followed by MOEMG (2.1412) and RF (2.3775). A different situation occurs for the G-mean$_2$, where MOEMG (1.89) is followed by RF (2.0488) and RUSBoost (2.0612). Despite the order for the previous metric, there is a line connecting all methods, indicating no statistically significant difference between them. Finally, MOEMG obtains the best mean rank (1.7825) for the F-measure, followed by RF (2.005) and RUSBoost (2.2125).

When comparing the pool generation methods through Fig. 8, the trade-off between FPR and FNR is clear, where models that were better in one metric are worst in the other. Despite this, MOEMG is able to keep a balance between both metrics. Additionally, the CD plots show a correlation between G-mean$_1$ and FPR, since the latter is composed of specificity ($1 - FNR$) instead of precision. In this metric, MOEMG also maintains a median rank. However, when G-mean$_2$ and F-measure are analyzed, MOEMG achieves the best mean ranks. Specifically for the F-measure, the employment of MOO in the pool generation step produces a significant improvement in the results.

4.1.2. Pool aggregation results

For the pool aggregation methods in Fig. 9, a 0.01 significance level introduces the CD value of 0.33. Also, the scale shows values from 1 to 4, indicating all compared methods: MOEMS, MOEMC, MOEMSC, and the original aggregation method for the pool generation algorithms (indicated as "Original"). The same previous five metrics are analyzed.

MOEMS achieves the lowest mean rank (2.2817) for the FPR metric, followed in order by the MOEMSC (2.455), Original (2.595)

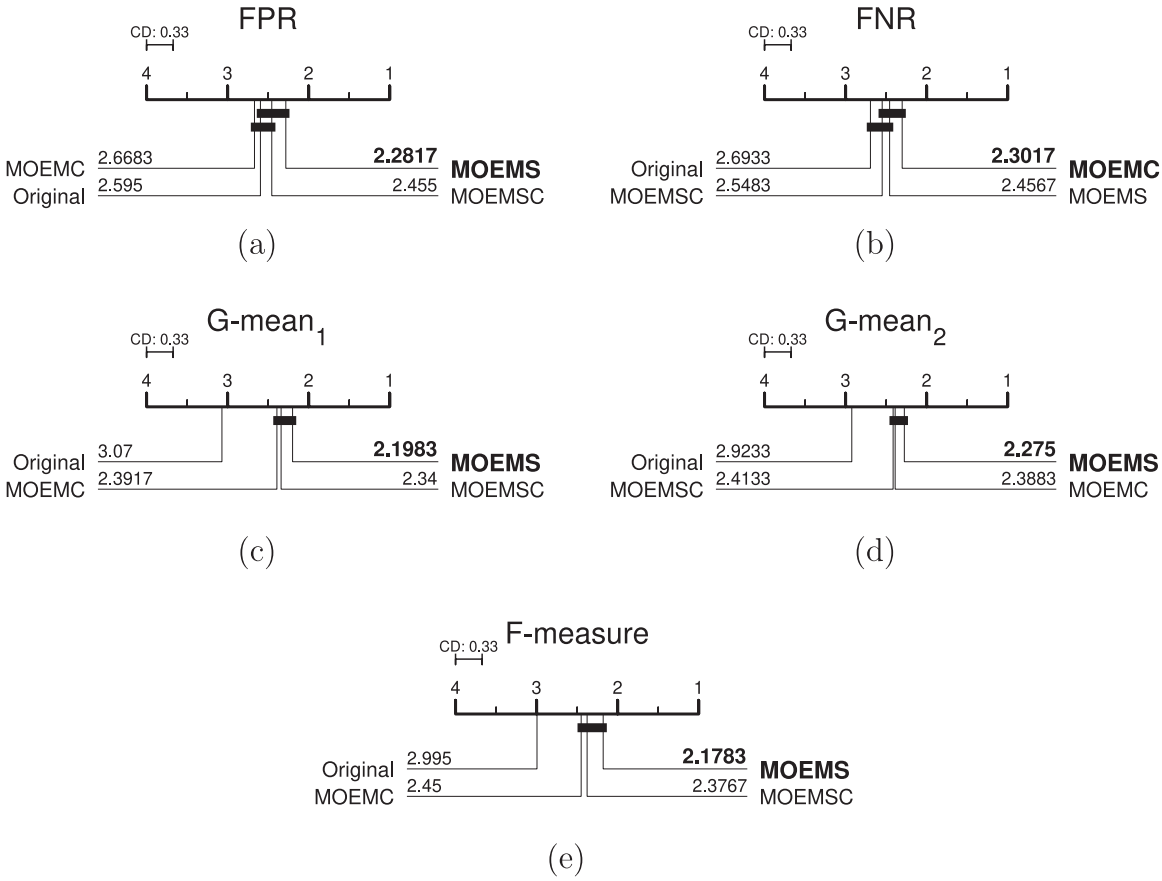

Fig. 9. The pool aggregation methods' critical differences (CD) plots for the five metrics: (a) false positive ratio (FPR), (b) false negative ratio (FNR), (c) G-mean$_1$, (d) G-mean$_2$, and (e) F-measure.

and MOEMC (2.6683). Despite this, MOEMSC, MOEMS, and Original show no statistical significant difference. The same scenario occurs for MOEMSC, Original, and MOEMC. For the FNR, the lowest mean rank is attained by MOEMC (2.3017), followed by MOEMS (2.4567), MOEMSC (2.5483) and Original (2.6933). No statistical significant differences are found between MOEMC, MOEMS, and MOEMSC neither MOEMS, MOEMSC, and Original.

For the G-mean$_1$, no statistically significant difference is found between MOEMS (2.1983), MOEMSC (2.34) and MOEMC (2.3917). However, Original produces the worst mean rank (3.07). The same occurs for G-mean$_2$, where Original accomplishes a mean rank of 2.9233. For such metric, MOEMS, MOEMC, and MOEMSC reaches statistically insignificant different mean ranks of 2.275, 2.3883, and 2.4133, respectively. Finally, for the F-measure, the best mean ranks are attained by MOEMS (2.1783), MOEMSC (2.3767), and MOEMC (2.45). The Original aggregation method accomplishes the worst mean rank (2.995) for such metric.

For pool aggregation methods, Fig. 9 shows the advantages of employing MOOD procedures. In all five metrics, MOEL methods are always among the two best solutions. FPR is the only metric were the Original combination method does not produce the worst mean rank. Also, in both FPR and FNR, there is no significant difference in the mean ranks when using or not MOO. However, this is not the case for G-mean$_1$, G-mean$_2$ and F-measure, where MOEL techniques always improve the results in a statistically significant manner. This will be further discussed in Section 5.

4.1.3. Models results

For this analysis, all the tested models (pool generation plus aggregation methods) are used, resulting in twelve models. Table 3 presents the number of wins obtained by each model for each of the five metrics. Additionally, the model with most number of wins for a specific metric is presented in bold and underline.

MOEMGS (MOEMG + MOEMS) attains the most number of wins for three metrics (FPR (26), G-mean$_2$ (29), and F-measure (34)), while RF with MOEMS reaches the most number of wins for FNR (30) and G-mean$_1$ (18). Additionally, other strong models for FPR are RF with MOEMSC (16 wins) and RF (15 wins). RF with MOEMC accomplishes 20 wins for the FNR. MOEMGS produces 15 wins for the G-mean$_1$, while RF with MOEMS achieves 13 wins for G-mean$_2$ and 11 for F-measure. When all possible combinations of pool generation and aggregation methods are tested on the benchmark problems, MOEMS accomplishes the most number of wins

Table 3
Number of wins achieved by each model for false positive ratio (FPR), false positive ratio (FPR), G-mean$_1$, G-mean$_2$, and F-measure. Values in bold and underline indicate the model with most number of wins for the specific metric.

Model	FPR	FNR	G-mean$_1$	G-mean$_2$	F-measure
MOEMGS	**26**	11	15	**29**	**34**
MOEMGC	10	2	4	11	11
MOEMGSC	7	0	1	0	2
MOEMG	3	4	8	1	4
RF+MOEMS	8	**30**	**18**	13	11
RF+MOEMC	10	20	14	11	6
RF+MOEMSC	16	2	8	5	5
RF	15	6	5	5	6
RUS+MOEMS	2	6	7	7	7
RUS+MOEMC	2	9	8	8	7
RUS+MOEMSC	0	5	6	3	2
RUS	1	5	6	7	5

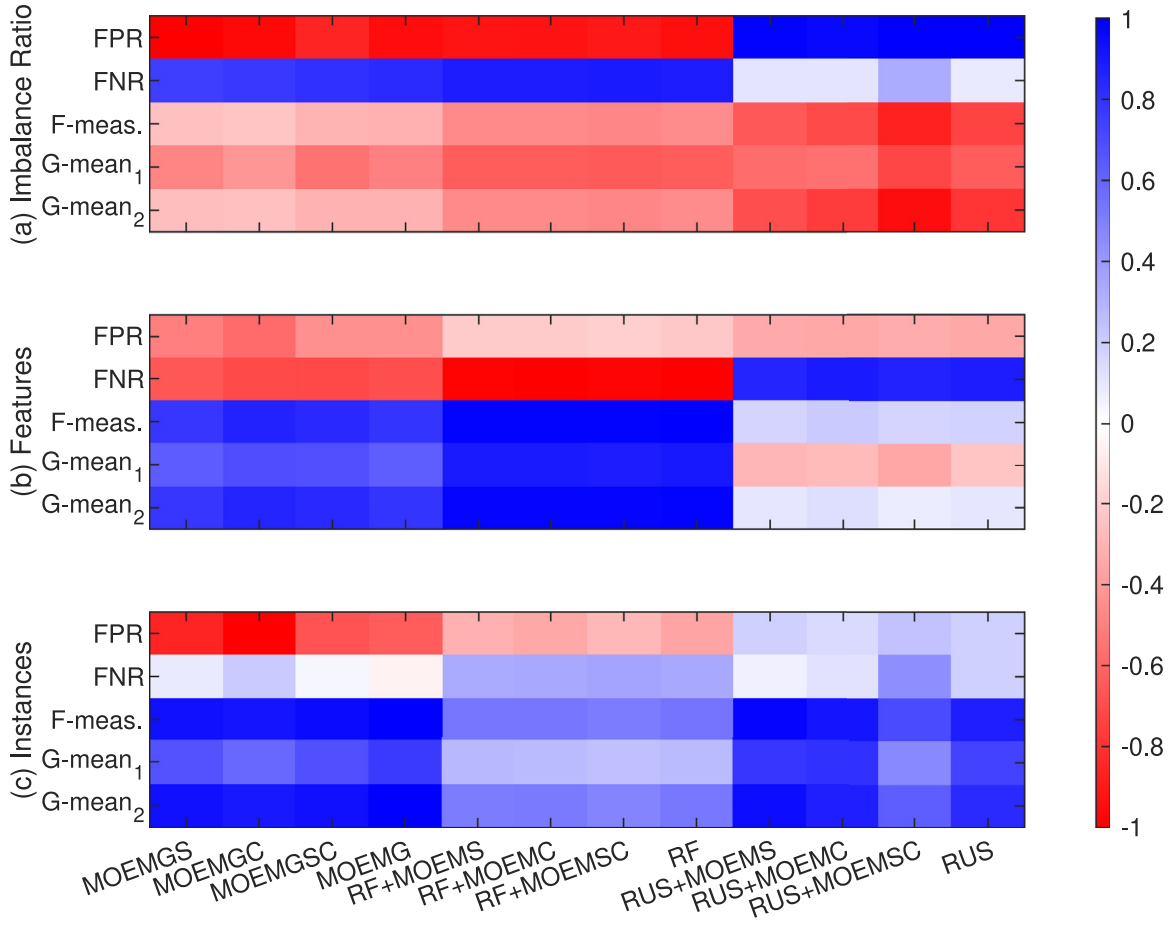

Fig. 10. Correlations, for each of the twelve models, between the five metrics used for evaluation and three data set characteristics: (a) imbalance ratio, (b) number of features, and (c) number of instances.

when combined with MOEMG (for FPR, G-mean$_2$, and F-measure) and RF (for FNR and G-mean$_1$).

When MOEMG, RF, and RUSBoost are compared, the most number of wins are achieved by the first method, followed by RF. RUSBoost presents the worst results, but such problem is highly dependent on the data set characteristics. For all three pool generation methods, a greater number of wins is achieved when MOEMS and MOEMC are used. However, this is not the case for MOEMSC, which presents lower number of wins. Such results indicate the great advantage of using MOOD for member generation, selection, and combination. The worst results in the MOEMSC aggregation method are caused because more decision variables must be adjusted, and more function evaluations would be necessary for improving such results.

Additionally, for understanding how the data sets affect the presented methods, Fig. 10 plots the correlation between three data set characteristics (imbalance ratio, number of features, and number of instances) and the five previously presented evaluation metrics. For this task, the Pearson's correlation coefficient is used. Such coefficient is computed according to Eq. 23, where $\boldsymbol{a}$ is the data sets' desired characteristic and $\boldsymbol{b}$ the desired evaluation metric, $\bar{a}$ and $\bar{b}$ are their means, and $D = 100$ is the number of samples (data sets).

$$r_{a,b} = \frac{\sum_{d=1}^{D}(a_d - \bar{a})(b_d - \bar{b})}{\sqrt{\sum_{d=1}^{D}(a_d - \bar{a})^2}\sqrt{\sum_{d=1}^{D}(b_d - \bar{b})^2}} \tag{23}$$

Fig. 10 shows that F-measure, G-mean$_1$, and G-mean$_2$ are negatively affected by the imbalance ratio, despite the tested model. On the one hand, for all models with MOEMG and RF, an increase in the imbalance ratio decreases the FPR while increasing the FNR. On the other hand, for the models composed of RUSBoost, a higher imbalance ratio increases the FPR while mildly increasing the FNR. Therefore, as expected, higher imbalance ratios hurt all algorithms, including the ones where MOO is employed.

For the models generated with MOEMG and RF, FPR and FNR decreases while F-measure, G-mean$_1$, and G-mean$_2$ increase with the number of features. This effect is stronger for RF than for MOEMG. That is, a higher number of features benefit such models. However, for the models generated through RUSBoost, higher number of features mildly strengthen FPR, G-mean$_2$ and F-measure, while slightly affecting G-mean$_1$ and greatly affecting FNR. This is due to the presence of feature selection mechanisms in RF and MOEMG, which is absent in RUSBoost.

Finally, for all models, a higher number of instances benefits F-measure, G-mean$_1$, and G-mean$_2$, while affecting FNR. FPR is also benefited for MOEMG and RF, which is not the case for RUSBoost. The increase in the imbalanced evaluation metrics for all models is an expected situation, which is interesting and important. Simply put, when more samples are available in the data set, more information is available for the learning algorithms to create better predictive models.

In all the plots from Fig. 10, it is shown that the pool generation step is more susceptible to variations in the data set characteristics than the pool aggregation. Also, MOEMG and RF present similar responses to the data set characteristics, which differs from RUSBoost. Analyzing both methods, this can be expected, since both RF and MOEMG (in the presented approach) deal

with stochastic approaches for the selection of both instances and features.

4.2. Experiment on a real-world drinking-water quality anomaly detection problem

According to Rehbach et al. (2018a), an essential task for water supply companies is the provision of clean and safe drinking-water. Thus, highly sensitive sensors are employed to supervise relevant environmental and water data. The available data can be monitored for performing an early recognition of quality anomalies, enabling water supply companies to counteract in time when events are found. Inspired by this task, a real-world drinking-water quality anomaly detection problem has been proposed as an industrial challenge at the 2018's Genetic and Evolutionary Computing Conference (GECCO).

The data set is composed of a training set and a test set, each composed of a time series with 139,566 instances and 9 features. Additionally, 72 features have been engineered in Ribeiro and Reynoso-Meza (2018c), 8 for each original feature, being: Two signal delays, moving average, moving standard deviation, moving minimum, moving maximum, squared values and peak frequencies. The data set presents an imbalance ratio of 67.8365. For such challenge, the F-measure is selected as the evaluation metric, where the winning solution produced a score of 0.56123041 (Rehbach, Chandrasekaran, Rebolledo, Moritz, & Bartz-Beielstein, 2018b).

The 12 previous methods (as seen in Table 3) are applied for solving the drinking-water quality anomaly detection problem. In this data set, the original training set is split in 70% and 30% for training the models and optimizing them. Later, the full test set is used for assessing the models' performances. Finally, the same 5 metrics, FPR, FNR, G-mean$_1$, G-mean$_2$, and F-measure, are analyzed. Final results are shown in the following subsection.

4.2.1. Models results

Table 4 brings the results for the proposed experiment. In such table, the scores of the 12 methods are shown for each of the 5 used metrics, and the best values for each metric are highlighted in bold and underline. Additionally, the best F-measure score achieved in the industrial challenge is presented (Rehbach et al., 2018b).

The best FPR score is produced by the combination of RUSBoost+MOEMS (0.0004), followed closely by the combination of RUSBoost+MOEMC (0.0004) and MOEMGS (0.0004). RUSBoost is the best solution, followed by its combination with MOEMSC, MOEMC, and MOEMS for FNR (0.3280, 0.3439, 0.3637, and 0.3916, respectively) and G-mean$_1$ (0.8119, 0.8053, 0.7929, and 0.7779, respectively). Finally, RUSBoost+MOEMS achieves the best scores for G-mean$_2$ (0.6163296) and F-measure (0.6292). It is important to notice that all methods with RUSBoost outperform the challenge's winning solution (Rehbach et al., 2018b).

For the proposed problem, RUSBoost achieves the best results for all metrics among the pool generated methods. Such behavior can be related to the high number of instances, which empowers the results of such pool generation methods (as viewed in Fig. 10). Such factor could also empower the MOEMG technique, but the number of decision variables becomes extremely large, which makes a higher number of function evaluations necessary. However, its computational cost would be much higher. For the pool aggregation methods, MOEMS is the technique which most improves the results on F-measure and G-mean$_2$. However, MOEMC and MOEMSC are also able to improve the results from RUSBoost. Moreover, MOEMGS outperforms the original RUSBoost algorithm. Therefore, it is shown that MOEL can improve the results of ensemble learning techniques for imbalanced data sets.

5. Discussion

Results from the experiments evidence two important information. First, MOOD approaches are able to produce strong pools of base classifiers for dealing with imbalanced data sets. Second, MOOD of pool aggregation methods are efficient tools for improving the predictive performance of ensemble models, through ensemble member selection or weighting.

In the first experiment, MOEMG proved to be the best pool generation method for F-measure (Fig. 8). Additionally, when combined with MOEMS, it attained the best results for both G-mean$_2$ and F-measure (Table 3). However, this was not the case for the second experiment, where RUSBoost achieved the best scores (Table 4). The difference in such results is mostly related to the number of instances. The high number of instances in the real-world problem (139,566) benefits RUSBoost for F-measure and G-mean$_2$, but increases the number of decision variables for the proposed MOEMG. Nevertheless, MOEMG obtains better results for smaller data sets.

When pool aggregation methods are compared in the first experiment, it is shown that MOEL (MOEMS, MOEMC and MOEMSC) significantly improves the results for all imbalanced metrics (Fig. 9). Moreover, in the second experiment, such techniques improve the G-mean$_2$ and F-measure scores for both RUSBoost and RF (Table 4). The presented results for pool aggregation corroborate the advantage of employing MOOD ap-

Table 4
Scores achieved by each model for false positive ratio (FPR), false negative ratio (FNR), G-mean$_1$, G-mean$_2$, and F-measure. Values in bold and underline indicate the models with best values for the specific metric.

Model	FPR	FNR	G-mean$_1$	G-mean$_2$	F-measure
MOEMGS	0.0004	0.4453	0.7382	0.6019	0.5473
MOEMGC	0.0007	0.5127	0.6957	0.5877	0.5490
MOEMGSC	0.0020	0.4328	0.7478	0.5544	0.5424
MOEMG	0.0008	0.4912	0.7102	0.5861	0.5445
RF+MOEMS	0.0034	0.4405	0.7418	0.4800	0.4754
RF+MOEMC	0.0042	0.4384	0.7440	0.4732	0.4664
RF+MOEMSC	0.0030	0.4431	0.7409	0.5083	0.5071
RF	0.0041	0.4444	0.7394	0.4714	0.4657
RUS+MOEMS	**0.0004**	0.3916	0.7779	**0.6296**	**0.6292**
RUS+MOEMC	0.0004	0.3637	0.7929	0.6167	0.6166
RUS+MOEMSC	0.0011	0.3439	0.8053	0.6203	0.6200
RUS	0.0008	**0.3280**	**0.8119**	0.5964	0.5963
Challenge's winner[a]	–	–	–	–	0.5612

[a] Results available in Rehbach et al. (2018b).

proaches, which empowers the performance of ensemble learning techniques.

Further knowledge is acquired from the presented experiments. Despite the existence of several metrics for imbalanced classification, it is shown that they are also conflicting. The different results from G-means and F-measure confirm that the choice of a single metrics does not guarantee the best result on others. Therefore, achieving satisfactory results on multiple objectives is highly desirable. If a single imbalanced metric is chosen for mono-objective optimization, other metrics would be affected. MOOD approaches favors the development of machine learning models when such trade-offs are found.

Further discussion can be put into types of MOEL approaches and when to use each of them. MOEMG attains the best results among the pool generation techniques when fewer instances are available for training. However, the computational cost grows as the number of samples and features increases.

MOEMS, as a binary problem, is able achieve satisfactory results faster than MOEMC and MOEMSC for pool aggregation. This is due to the smaller size of the search space. MOEMC, as a continuous problem, enables a higher number of combinations than the previous technique. Therefore, it can perform a finer ensemble adjustment. The downsize of such approach is the higher number of necessary function evaluations. Finally, as a combination of the two previous methods, MOEMSC can be used for finding better results than MOEMS with a lower computational cost than MOEMC. However, due to the number of decision variables, it should take longer than MOEMC for finding similar results. Nevertheless, the choice of the MOEL approach to be used also consists of a multi-objective problem, where the trade-off between computational cost and predictive performance must be selected by a decision maker.

6. Conclusions and future research

This work presents the problem of learning on imbalanced data sets, along with the study of MOOD approaches to build ensemble models that can better work on such type of problem. We propose to answer the question of how MOOD can benefit ensemble learning techniques. To answer such question, first, a taxonomy for MOEL is presented, followed by experiments on benchmark problems, and the application of such techniques on a real-world problem.

With the presented results, it is confirmed that the application of a MOOD approach for generating base learners through simultaneous features and instances selection improves the creation of ensembles on problems with lower number of features and instances available for training. Moreover, this study concludes that MOOD, when used for pruning and weighting the base classifiers, can improve the performance of RF and RUSBoost ensembles. Therefore, the authors conclude that the application of MOO is an efficient method to design ensemble models for solving imbalanced classification problems.

Future research shall focus on: The comparison of different MOO algorithms and parameters for returning a Pareto front approximation; the comparison of different MCDM techniques for selecting a preferred solution from the set of non-dominated ensemble models; the development/comparison of a mixed-integer MOO algorithm for better dealing with feature and instances selection problems; comparison with other ensemble learning and classifier selection/pruning techniques (Galar, Fernández, Barrenechea, Bustince, & Herrera, 2016); and the application of the MOEL techniques for building better regression and multi-class classification models. Other interesting research topics can focus on the analysis of computational complexity in ensemble learning tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Victor Henrique Alves Ribeiro: Writing - original draft. **Gilberto Reynoso-Meza:** Writing - review & editing.

Acknowledgments

This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES), the *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq), and the *Fundação Araucária* (FAPPR) - Brazil - Finance Codes: 159063/2017-0-PROSUC, 310079/2019-5-PQ2, 437105/2018-0-Univ, and PRONEX-042/2018.

References

Al-Sahaf, H., Bi, Y., Chen, Q., Lensen, A., Mei, Y., Sun, Y., … Zhang, M. (2019). A survey on evolutionary machine learning. *Journal of the Royal Society of New Zealand, 49*(2), 205–228.

Albukhanajer, W. A., Jin, Y., & Briffa, J. A. (2017). Classifier ensembles for image identification using multi-objective pareto features. *Neurocomputing, 238*, 316–327.

Alexandropoulos, S.-A. N., Aridas, C. K., Kotsiantis, S. B., & Vrahatis, M. N. (2019). Multi-objective evolutionary optimization algorithms for machine learning: A recent survey. In *Approximation and optimization* (pp. 35–55). Springer.

Behzadian, M., Kazemzadeh, R. B., Albadvi, A., & Aghdasi, M. (2010). Promethee: A comprehensive literature review on methodologies and applications. European Journal of Operational Research, *200*(1), 198–215.

Brans, J.-P., Vincke, P., & Mareschal, B. (1986). How to select and how to rank projects: The promethee method. European Journal of Operational Research, *24*(2), 228–238.

Breiman, L. (1996). Bagging predictors. Machine Learning, *24*(2), 123–140.

Breiman, L. (2001). Random forests. Machine Learning, *45*(1), 5–32.

Bui, L. T., Dinh, T. T. H., et al. (2018). A novel evolutionary multi-objective ensemble learning approach for forecasting currency exchange rates. *Data & Knowledge Engineering, 114*, 40–66.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, *7*(Jan), 1–30.

Douzas, G., & Bacao, F. (2018). Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with Applications, *91*, 464–471.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets*. Springer.

Fernández, A., García, S., del Jesus, M. J., & Herrera, F. (2008). A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems, 159*(18), 2378–2398.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, *55*(1), 119–139.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (2016). Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets. *Information Sciences, 354*, 178–196.

Gu, S., Cheng, R., & Jin, Y. (2015). Multi-objective ensemble generation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5*(5), 234–245.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications, 73*, 220–239.

Hao, Y., & Tian, C. (2019). A novel two-stage forecasting model based on error factor and ensemble method for multi-step wind power forecasting. *Applied Energy, 238*, 368–383.

He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications, 98*, 105–117.

Japkowicz, N. (2013). Assessment metrics for imbalanced learning. Imbalanced Learning: Foundations, Algorithms, and Applications, 187–206.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence, 5*(4), 221–232.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research, 18*(1), 559–563.

Liu, H., Duan, Z., Li, Y., & Lu, H. (2018). A novel ensemble model of different mother wavelets for wind speed multi-step forecasting. Applied Energy, *228*, 1783–1800.

Ma, S., & Chu, F. (2019). Ensemble deep learning-based fault diagnosis of rotor bearing systems. *Computers in Industry, 105*, 143–152.

Onan, A., Korukoğlu, S., & Bulut, H. (2016). A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications, 62*, 1–16.

Onan, A., Korukoğlu, S., & Bulut, H. (2017). A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification. *Information Processing & Management, 53*(4), 814–833.

Peimankar, A., Weddell, S. J., Jalal, T., & Lapthorn, A. C. (2017). Evolutionary multi-objective fault diagnosis of power transformers. *Swarm and Evolutionary Computation, 36*, 62–75.

Peimankar, A., Weddell, S. J., Jalal, T., & Lapthorn, A. C. (2018). Multi-objective ensemble forecasting with an application to power transformers. *Applied Soft Computing, 68*, 233–248.

Pereira, D. G., Afonso, A., & Medeiros, F. M. (2015). Overview of friedman's test and post-hoc analysis. *Communications in Statistics-Simulation and Computation, 44*(10), 2636–2653.

Qu, Z., Zhang, K., Mao, W., Wang, J., Liu, C., & Zhang, W. (2017). Research and application of ensemble forecasting based on a novel multi-objective optimization algorithm for wind-speed forecasting. *Energy Conversion and Management, 154*, 440–454.

Rehbach, F., Chandrasekaran, S., Rebolledo, M., Moritz, S., & Bartz-Beielstein, T. (2018a). GECCO Challenge 2018: Online Anomaly Detection for Drinking Water Quality. http://www.spotseven.de/gecco/gecco-challenge/gecco-challenge-2018/.

Rehbach, F., Chandrasekaran, S., Rebolledo, M., Moritz, S., & Bartz-Beielstein, T. (2018b). GECCO Challenge 2018 Results. http://www.spotseven.de/gecco/gecco-challenge/gecco-challenge-2018/gecco-challenge-2018-results/.

Reynoso-Meza, G., Blasco, X., Sanchis, J., & Martínez, M. (2014a). Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice, 28*, 58–73.

Reynoso-Meza, G., Sanchis, J., Blasco, X., & García-Nieto, S. (2014b). Physical programming for preference driven evolutionary multi-objective optimization. *Applied Soft Computing, 24*, 341–362.

Ribeiro, V. H. A., & Reynoso-Meza, G. (2018a). A multi-objective optimization design framework for ensemble generation. In *Proceedings of the genetic and evolutionary computation conference companion GECCO '18* (pp. 1882–1885). New York, NY, USA: ACM. doi:10.1145/3205651.3208219.

Ribeiro, V. H. A., & Reynoso-Meza, G. (2018b). Multi-objective support vector machines ensemble generation for water quality monitoring. In *2018 ieee congress on evolutionary computation (cec)* (pp. 1–6). doi:10.1109/CEC.2018.8477745.

Ribeiro, V. H. A., & Reynoso-Meza, G. (2018c). Online anomaly detection for drinking water quality using a multi-objective machine learning approach. In *Proceedings of the genetic and evolutionary computation conference companion* (pp. 1–2). ACM.

Ribeiro, V. H. A., & Reynoso-Meza, G. (2019a). A holistic multi-objective optimization design procedure for ensemble member generation and selection. *Applied Soft Computing, 83*.

Ribeiro, V. H. A., & Reynoso-Meza, G. (2019b). A study of pareto-based methods for ensemble pool generation and aggregation. In *2019 ieee congress on evolutionary computation (cec)* (pp. 2145–2152). IEEE.

Rosales-Perez, A., Garcia, S., Gonzalez, J. A., Coello, C. A. C., & Herrera, F. (2017). An evolutionary multi-objective model and instance selection for support vector machines with pareto-based ensembles. *IEEE Transactions on Evolutionary Computation, PP*(99). 1–1

Rosales-Pérez, A., Gonzalez, J. A., Coello, C. A. C., Escalante, H. J., & Reyes-Garcia, C. A. (2014). Multi-objective model type selection. *Neurocomputing, 146*, 83–94.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(4), e1249.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40*(1), 185–197.

Singh, A., & Purohit, A. (2015). A survey on methods for solving data imbalance problem for classification. *International Journal of Computer Applications, 127*(15), 0975–8887.

Smith, C., & Jin, Y. (2014). Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction. *Neurocomputing, 143*, 302–311.

Tan, C. J., Lim, C. P., & Cheah, Y. N. (2014). A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models. *Neurocomputing, 125*, 217–228. doi:10.1016/j.neucom.2012.12.057.

Woźniak, M., Graña, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion, 16*, 3–17.

Zhang, C., Lim, P., Qin, A., & Tan, K. C. (2016). Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE Transactions on Neural Networks and Learning Systems, *28*(10), 2306–2318.

Zhao, J., Jiao, L., Xia, S., Basto Fernandes, V., Yevseyeva, I., Zhou, Y., & T. M. Emmerich, M. (2018). Multiobjective sparse ensemble learning by means of evolutionary algorithms. *Decision Support Systems, 111*, 86–100. doi:10.1016/j.dss.2018.05.003.