IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 April 2024, accepted 17 June 2024, date of publication 26 June 2024, date of current version 3 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3419130

RESEARCH ARTICLE

AC.Rank$_A$: Rule Ranking Method via Aggregation of Objective Measures for Associative Classifiers

MAICON DALL'AGNOL AND VERONICA OLIVEIRA DE CARVALHO
Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista (Unesp), Rio Claro, São Paulo 13506-900, Brazil
Corresponding author: Veronica Oliveira De Carvalho (veronica.carvalho@unesp.br)

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior–Brasil (CAPES).

ABSTRACT Among the inherently interpretable learning algorithms are associative classifiers, which are induced in steps. Regarding the ranking step, it is carried out using objective measures in order to sort the rules. Generally, the CSC method is used based on the two standard measures of association rules (support and confidence). However, several measures are available in the literature, leading to a secondary problem, as there is no measure that is suitable for all explorations. In this context, new proposals have emerged, one of which aims to aggregate a set of measures in order to use them simultaneously. The idea is to reduce the need to choose a single measure, also considering different aspects (semantics) for ranking the rules. Works in this context have been proposed. However, they present problems in relation to the performance and/or interpretability of the generated models. In them it is possible to observe an inverse relationship between performance and interpretability, i.e., when model performance is high, interpretability is low (and vice versa). Therefore, this work presents a rule ranking method via aggregation of objective measures, named *AC.Rank*$_A$, to be incorporated into associative classifiers induction flows, aiming to obtain models that present a better balance between performance and interpretability. The method was evaluated by comparing several induction flows when ranking takes place via CSC (baseline) and via *AC.Rank*$_A$. The results demonstrate that *AC.Rank*$_A$ can maintain the performance of the models, but with better interpretability.

INDEX TERMS Associative classifiers, aggregation, interpretability, objective measures, performance, rule ranking.

I. INTRODUCTION

One of the most common machine learning tasks is classification, which aims to assign a category, named class, to unlabeled instances. Among the techniques that stand out in this task is associative classifier (AC). Associative classifiers are part of a larger family of algorithms, named rule-based, which use rules to represent the extracted knowledge. ACs make use of association rules (ARs) to generate a classifier. This is done through a special type of AR known as a classification association rule (CAR). A CAR is a rule of type $A \Rightarrow C$ in which the antecedent (A) leads to a certain consequent (C). A contains a set of <attribute = value> pairs and C contains a class of a given problem. Thus, an AC is a model composed of rules of this type in order to classify new instances. The use of these rules is advantageous, as they are easily interpretable by experts who can analyze them and make decisions with computational aid. Due to its inherent interpretability, several areas have used it, such as medical [1], [2], [3], security [4], [5] and software [6], [7].

According to [8] recent studies demonstrate that AC has the following advantages over traditional interpretable classification approaches: (i) accuracy, the algorithms in this family are capable of building efficient and accurate models since the training phase is based on ARs, allowing the discovery of all possible relationships between <attribute = value> pairs; (ii) usability, unlike the decision tree family, AC algorithms do not require redesigning the entire model when the rule set is updated and adjusted; (iii) readability, the final model of an AC is composed of a simple set of rules that allows the end user to easily understand and interpret the results. However, some advantages such as

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

accuracy are still subject to debate. Much of this is due to the balance between interpretability and accuracy, reported in several works [9], [10], although this has been questioned, as discussed in [11]. Even with several XAI (Explainable Artificial Intelligence) methods available today, such as LIME and SHAP, there are works such as [11] that propose the use of more interpretable classifiers instead of trying to create explanation models for black box algorithms via such methods.

As in other families of algorithms, efforts have been made to further popularize AC through the implementation of packages, as seen in [8], [12], and [13], which contribute by making algorithms available for general use. The algorithm that stands out the most is CBA [14], available in all implementations surveyed, which is generally used as a baseline for comparison with new proposals/solutions. According to [15], CBA is probably the most used algorithm in the family.

Most AC algorithms perform model induction in three or four steps, namely: [A] Extraction, [B] Ranking and/or Pruning and [C] Prediction. Regarding ranking (step [B]), it is, in general, carried out through the use of objective measures (OMs), used, among other purposes, to rank the rules by their degree of importance.[1] An objective measure (OM) evaluates the statistical strength, or characteristics, of a given pattern based solely on data [16]. The best known OMs are support ($P(AC)$) and confidence ($P(C|A)$), used in the extraction of ARs. In CBA the CSC (Confidence, Support, Cardinality) method is used to order the rules as follows: a rule r_i precedes a rule r_j, in an ordered list, if the confidence of r_i is greater than r_j; in case of a tie, if the support of r_i is greater than r_j; in case of a tie, r_i was generated before r_j (cardinality).

Although support and confidence are the most traditional OMs, more than 60 measures are found in the literature, as seen in [17] and [18]. This large number of existing OMs has generated a secondary problem, since there is no OM that is suitable for all explorations [19]. Therefore, new proposals have emerged aiming to modify the ACs ranking step, either through: (i) new measures and/or measures that already exist in other contexts, but applied at this step in a unique way; (ii) the merge (aggregation) of existing measures in order to use them simultaneously (see Section III-A). Strategy (ii), adopted in this work, has the advantage of reducing the need to choose a single measure, also considering different aspects (semantics) for sorting the rules.

Works related to strategy (ii) have been developed, such as [20], [21], and [22], which have shown promising results. However, they present problems in relation to the performance and/or interpretability of the generated models. In the works of [22] and [23] it is possible to verify an inverse relationship between performance and interpretability, i.e., when model performance is high, interpretability is low (and vice versa). In this work performance is estimated via F1-Macro and interpretability via model size (see Section II-D).

[1]Ranking, ordering and sorting are used as synonyms in this work.

Considering the above, this work presents a method for ranking rules via aggregation of OMs, named $AC.Rank_A$, which can be incorporated into ACs induction flows aiming to induce models that present a better balance between performance and interpretability. It is considered, in this work, that these criteria are analyzed in relation to some baseline. For $AC.Rank_A$ to work, it must be instantiated with a set of OMs and an aggregation method. The sets of OMs explored were obtained or generated from works in the literature. The aggregation methods were extracted from both ranking aggregation and multi-criteria decision analysis (MCDA) approaches. The proposed method was evaluated in several ACs induction flows regarding performance and interpretability. The analysis was carried out by comparing different induction flows when ranking takes place via CSC (baseline) and via $AC.Rank_A$. The results obtained demonstrate that $AC.Rank_A$ can maintain the performance of the models, but with better interpretability. It is worth mentioning that this work is an extension of the proposal presented in [23], with the following differences:

- in [23] the motivation for the work was to propose a rule ranking method via aggregation of OMs that would improve the execution time and interpretability of the proposals previously found in the literature, namely: [20] and [21]. In [20] the authors present a solution based on Pareto and [21] based on a ensemble of classifiers. Both works have an execution time higher than CBA (baseline). Furthermore, the solution proposed by [21] does not, in fact, generate an interpretable model, as it is composed of a set of models. On the other hand, in this work, the motivation is to maintain the balance between performance and interpretability, as in [22];
- in [23] only a small part of the configurations explored here via $AC.Rank_A$ are evaluated, i.e., a small number of experiments are carried out, making it difficult to obtain generalizable conclusions. In this work, the proposal initially presented in [23] was extended to a rule ranking method via aggregation of OMs, which was evaluated in several induction flows, in different configurations (diverse sets of OMs combined with different aggregation methods). Considering the total number of experiments carried out here, 185,760, in different experimental flows, it is possible to verify the contribution of $AC.Rank_A$ to the area;
- in [23] only ranking aggregation methods are considered; here, in addition to these, MCDA methods are also explored, due to their recent use in ARs contexts, such as in [24], [25], and [26];
- in [23] only two sets of OMs are explored; here, the analysis was extended to five sets;
- in [23] the analyzes are carried out by comparing the proposal with the works of [20] and [21]. However, they were not designed to simultaneously optimize performance and interpretability. On the other hand, the work of [22] is aimed at optimizing both aspects.

Therefore, an analysis in relation to this work is presented here, unlike the previous work.

Finally, it is mentioned that this work is the result of our master's thesis [27]. Besides, it is worth noting that other works, which also aim to improve the interpretability of models, have been proposed in the literature, such as [28], [29], and [30]. However, the proposals are focused on other steps of the induction process. The step that this work aims to address is ranking.

The paper is structured as follows: Section II presents the concepts that support this work; Section III the works related to each of the induction steps of an AC, describing some methods used in the evaluation of the method proposed here; Sections IV and V present, respectively, the proposed method, $AC.Rank_A$, as well as its evaluation; Section VI presents conclusions and future work.

II. THEORETICAL FOUNDATION

This section briefly presents the concepts that underlie this work, namely: Association Rules (Section II-A), Objective Measures (Section II-B), Associative Classifiers (Section II-C) and Evaluation Criteria (Section II-D).

A. ASSOCIATION RULES (ARs)

An association rule (AR) expresses a relation between items that occur in a given dataset. The relations are of type $A \Rightarrow C$, where A represents the antecedent, C the consequent and $A \cap C = \varnothing$. A rule occurs with a support *sup* and a confidence *conf*. Support, defined as $P(AC)$, indicates the frequency of the pattern while confidence, defined as $P(C|A)$, the probability C occurs given that A occurred. A and C are itemsets, a subset of a set of items I that appear in the dataset. An item, in this paper, is a pair <attribute = value>, since we are dealing with relational tables. There are many algorithms that can be used to extract a set of ARs, with Apriori [31] being the most traditional.[2]

B. OBJECTIVE MEASURES (OMs)

Objective measures (OMs) are present in ARs and, consequently, in ACs. An OM evaluates the statistical strength, or characteristics, of a given pattern, based solely on the data [16]. OMs can be used (i) during rule extraction, in order to prune the search space, and/or (ii) to order, i.e., rank the rules by their degree of importance, in order to find relevant patterns. In general, the higher the value of a measure in a given rule, the better ranked the rule will be.

In order to evaluate the importance of a rule through an OM, it is necessary to know the rule's contingency table. Table 1 presents the structure of a contingency table and its possible values for an abstract rule $A \Rightarrow C$. A represents the antecedent, C the consequent, $\overline{A}$ the negation of the antecedent, $\overline{C}$ the negation of the consequent, $n(X)$ the frequency of X and N the number of transactions. Thus, $n(XY)$ represents the frequency with which X and Y happen simultaneously in the N transactions of the dataset. OMs are defined as a function of these absolute frequencies. However, the most usual notation is by means of probabilities (relative frequency), obtained by dividing each element of the table by N. Rule support, for example, is defined as $P(AC) = \frac{n(AC)}{N}$ and confidence as $P(C|A) = \frac{P(AC)}{P(A)}$, where $P(A) = \frac{n(A)}{N}$. The mathematical definition of all the measures mentioned and used in this work can be found in Appendix B.

TABLE 1. Contingency table of an abstract rule $A \Rightarrow C$.

	C	$\overline{C}$	
A	n(AC)	n($A\overline{C}$)	n(A)
$\overline{A}$	n($\overline{A}C$)	n($\overline{A}\overline{C}$)	n($\overline{A}$)
	n(C)	n($\overline{C}$)	N

Although support and confidence are the most traditional OMs, more than 60 measures are found in the literature, as seen in [17] and [18], aiming to overcome limitations of some of the previously existing measures, like confidence itself (it does not detect positive or negative correlations, nor the absence of correlation [32]). This large number of existing OMs has generated a secondary problem, since there is no OM that is suitable for all explorations [19].

Choosing one or more measures to explore a set of rules becomes important given the benefits that are obtained by opting for one or another OM. However, choosing which one to use is a difficult problem [19], [33], as there are significant differences in relation to objectives (semantics), among other aspects, that distinguish them. To this end, different ways were proposed to make this choice. Selection proposals, according to [34], are divided into two categories, namely: ranking and clustering. The aim of ranking-based approaches is to list the OMs in order of importance so that the user selects the first n. The aim of clustering-based approaches is to cluster OMs into groups so that the user can select a representative measure from each group. In this way, both aim to reduce the search space. Both ranking and clustering can be carried out based on the similarity of OMs in relation to the properties they present or in relation to their behavior with regard to the ordering generated by them in a given set of rules. The "Clustering" solution based on "Behavior" is used in this work to select the sets of OMs to be aggregated, as will be seen in Section IV-A.

C. ASSOCIATIVE CLASSIFIER (AC)

In the AC literature, some algorithms have become traditional due to their uses and concepts. Among those widely cited are CBA [14], CMAR [35] and CPAR [36]. A review of ACs can be found at [37]. CBA is, in general, used as a baseline in most related works, and is also the first algorithm proposed in this category. According to [15], CBA is probably the most used algorithm in the family and is therefore presented here. Algorithm 1 presents the necessary steps for inducing a model via CBA, i.e., **Extraction**, **Ranking**, **Pruning** and **Prediction**,

[2]The implementation of this and other algorithms is available at http://www.philippe-fournier-viger.com/spmf/.

which are described below. For reasons of simplicity and ease of understanding, some CBA steps were omitted from Algorithm 1.

Algorithm 1 CBA Algorithm. Adapted from [14].

Require: Dataset D
Ensure: Classifier C
1: $R \leftarrow CARs(D)$ ▷ **Extraction** of CARs
2: $R \leftarrow Sort(R)$ ▷ **Ranking**
3: **for all** $r \in R$ **do** ▷ **Pruning**
4: $temp \leftarrow \varnothing$
5: **for all** $d \in D$ **do**
6: **if** d satisfies the conditions of r **then**
7: store $d.id$ in $temp$
8: mark r if it correctly classifies d
9: **end if**
10: **end for**
11: **if** r is marked **then**
12: insert r at the end of C
13: delete all cases with ids in $temp$ of D
14: select a default class for the current classifier C
15: **end if**
16: **end for**
17: **return** C

1) RANKING

From the extracted CARs (line 1), stored in R, order them from largest to smallest according to the relationship $\succ$ (line 2). Given two rules, r_i and r_j, $r_i \succ r_j$, that is, r_i has higher precedence than r_j if: the confidence of r_i is greater than r_j; in case of a tie, the support of r_i is greater than r_j; in case of a tie, r_i was generated before r_j (cardinality). This ranking method is known by the acronym **CSC** (Confidence, Support, Cardinality), which will be referred to as **[CSC]** (see Section V).

2) PRUNING

For each rule $r \in R$, ordered by precedence, the training set D is traversed in order to find the transactions $d \in D$ that are *covered* by r, that is, that satisfy the conditions imposed by r (line 6). If r correctly classifies any of the transactions *covered* by it, the algorithm marks r (line 8), so that r becomes a potential candidate for the classifier (lines 11-15). Subsequently, all cases *covered* by r are deleted from D (line 13) and a default (majority) class (considering the remainder of D) is associated with the current classifier (line 14). Finally, when there are no more rules r or transactions in D, the rule selection stopping criterion is reached (line 16). This pruning method is known by the acronym **M1**, which will be referred to as **[M1]** (see Section V). In the end, a classifier C, given by $C = \{r_1, r_2, r_3, \ldots, r_n,$ default class$\}$, is returned, where $r_i \in R$ and $r_i \succ r_j$.

3) PREDICTION

From the obtained model, for a new instance to be classified, the class associated with the first rule that matches the characteristics of the object is used. This prediction method will be referred to as **[O]** (**O**rdinal prediction) (see Section V).

D. EVALUATION CRITERIA

The classification task aims to assign a category, named class, to unlabeled instances. Thus, once the models are obtained, they need to be evaluated according to a performance metric on data not yet seen. To this end, evaluation measures are used, such as Precision, Recall and F-measure (micro and/or macro and/or weighted macro versions). Furthermore, in order to guarantee a good estimate of the evaluation measure, as well as to enable the configuration of hyperparameters, different validation strategies can be used, such as holdout, cross-validation and stratified cross-validation. Therefore, in relation to performance, this work evaluates $AC.Rank_A$, in ACs flows, via F1-Macro estimated via 10-fold stratified cross-validation. Due to the fact that these concepts are basic in machine learning literature, they will not be detailed here, which can be consulted at [32].

However, another important criterion to be used to evaluate models induced via rule-based algorithms is interpretability. According to [38], interpretability has become increasingly important in the context of predictive tasks, although there is still no consensus on this notion. In general, as noted in [11], [38], and [39], it is possible to obtain interpretable prediction models (i) via non-interpretable algorithms, such as deep neural networks, on which XAI methods are applied, or (ii) via inherently interpretable algorithms, such as those based on rules and trees. Still according to [38], although inherently interpretable algorithms seem easy to understand, there is no exact mathematical definition for the concept of interpretability. Therefore, each work chooses to evaluate this criterion using a different measure.

Given the above, in this work we chose to measure interpretability through the number of rules contained in the model, as in other recent works in the context of AC [22], [29], [30], [40], [41], [42]. In all these works, the smaller the number of rules, the better the induced model, i.e., the more interpretable it is. Finally, it is worth mentioning that it is also estimated via stratified 10-fold cross-validation. This criterion was called model size ($\mathcal{L}$).

III. RELATED WORKS

As mentioned, most traditional ACs algorithms perform model induction in three or four steps, namely: [A] Extraction, [B] Ranking and/or Pruning and [C] Prediction, as can be seen in Fig. 1. They directly influence the quality of the generated model. In view of this, efforts have been made to improve each of these steps. It is noted, in the respective figure, that each step of the process can be seen as an independent method, which together are capable of generating a model. CBA, like CMAR and CPAR, can be seen in this way. Therefore, with regard to the rule ranking method via aggregation of OMs proposed here, $AC.Rank_A$, it can be seen in Fig. 1 that the highlighted step is the one in which the $AC.Rank_A$ must be used.

Given the above, this section aims to present works related to the steps presented in Fig. 1, i.e., different methods per step,

FIGURE 1. Induction steps of an AC.

with some of them being used later in the evaluation of the method proposed here (Section V). Finally, in Section III-D a work well correlated to this one is presented, from which some methods were used during the evaluation of $AC.Rank_A$. It is worth mentioning that the extraction step is not covered below, since in this work this step was the only one in which different methods were not evaluated. According to [43] this step is often associated with two widely used algorithms, Apriori and FP-Growth, although other strategies are possible together with other algorithms, as in [44]. Therefore, in this work, only the Apriori algorithm was used.

A. RANKING STEP

The ranking step, the focus of this work, is of great importance, as it directly affects the performance of the classifier, as it impacts the subsequent steps (pruning and prediction) (see Fig. 1). Therefore, several works seek to improve this step, aiming to organize the rules that favor the performance of the generated model and/or interpretability through a compact set of rules.

Ranking is often performed using confidence, followed by support and cardinality (CSC (see Section II-C)). However, other proposals have emerged over the years, either through: (i) new measures and/or measures that already exist in other contexts, but applied at this step in a unique way; (ii) the merge (aggregation) of existing measures in order to use them simultaneously. In general, the solutions are based on works developed for the context of ARs, which are then adapted to the context of ACs.

Considering the above, in relation to item (ii), works were developed in the context of ARs, such as that of [45] via Integral Choquet, and those of [46] and [47] via Pareto dominance. Recently, other ways of aggregating OMs, using solutions from the area of multi-criteria decision analysis (MCDA), have been used. References [24] and [25] employ similar ideas regarding aggregation through the use of ELECTRE I [48]. On the other hand, [26] employ ELECTRE II [48] aiming at ranking rules via the simultaneous use of support, confidence and lift. It is worth mentioning that, both in solutions that use Pareto and in those that use MCDA, the motivation for aggregating OMs comes from the fact that each measure provides a ranking according to its semantics and, therefore, considering them all simultaneously, we obtain multiple views that, together, contribute to a more comprehensive ranking, not being restricted to the biased view of one or two OMs.

On the other hand, considering aggregation, but in the context of AC, some work has been carried out. Reference [20] explore the aggregation of OMs via Pareto, when applied to rank the rules, based on the work of [46]. In this case, the proposed method is evaluated in the CBA ranking step. Reference [21] perform the aggregation through a ensemble of classifiers, in which each model of the ensemble is induced using a different measure. Finally, [22] explore the aggregation of OMs in the ranking step via a multi-objective optimization model, described in greater detail in Section III-D. As mentioned before, although these works are promising, they present problems in relation to the performance and/or interpretability of the generated models. Of the three works, the one by [21] does not, in fact, generate an interpretable model, since it is composed of a set of models (ensemble). Thus, in itself, it already presents interpretability problems. The work of [20], as presented and discussed in [23], presents an inverse relationship between performance and interpretability, i.e., when model performance is low, interpretability is high (and vice versa). However, the solution was not designed to optimize both criteria simultaneously. The solution presented in [22] is aimed at optimizing both aspects. However, even in this case, it is possible to observe the inverse relationship between performance and interpretability.

B. PRUNING STEP

Pruning, like ranking, can be omitted depending on the model's induction flow. However, when it is present, its impact is large, being responsible, for example, for reducing the size of the rule set and removing redundant and/or low-quality rules. Although the pruning process is often carried out after ranking, so as not to directly affect the ranking itself, the performance of the induced model depends on the "agreement" between the steps when both occur. Works that present solutions involving the pruning step in the context presented here are found in [28], [29], [49], [50], and [51].

As mentioned, each step of the induction process can be seen as an independent method (see Fig. 1). Thus, to evaluate the method proposed here, some pruning methods from the literature were selected to compose different algorithmic flows, together with the ranking method under analysis, in order to evaluate its performance. Therefore, the methods described below are used, in addition to **[M1]** (CBA), in the experiments in Section V.

[DN] — **D**y**N**amic Pruning of [40]. The pruning method presented by [40] was one of those selected and adapted for this work. In the aforementioned work, an AC is induced through iterative pruning, which works in conjunction with a ranking step, as described below:

(i) Ranking: the rules of the set R are ranked according to an OM;
(ii) Selection: the first rule r_i from set R is selected;
(iii) Instance Deletion: all instances (transactions) covered by r_i are removed from the dataset;
(iv) Rule Deletion: rule r_i is removed from the rule set R;
(v) Update of OMs: the remaining rules from set R have their OMs values updated based on the remaining instances.

The process is repeated until all rules have been processed or there are no more instances to cover. The process is expensive, as it involves recalculating all OMs used at each iteration.

However, each iteration ensures that the best rule is selected, given the current state of the dataset.

[CV] — **C**o**V**erage Pruning of [22]. The pruning method presented by [22] is based on the coverage with which the rules, already ordered, match the transactions. This method is described in Section III-D.

C. PREDICTION STEP

Prediction is the step in which the class for unseen instances is determined. In this step, two procedures are generally used: one through the precedence of the rules, directly impacted by ranking and/or pruning, named **Rule List**, and the other through multiple rules, named **Rule Set** [52]. In the first, a single rule is used to determine the class, the one with the best precedence that matches the instance, while in the second, a set of rules is used to define it. As an example, while CBA uses the **Rule List** procedure, CMAR and CPAR the **Rule Set**. Works that present solutions involving this step are found in [40] and [53].

To evaluate the method proposed here, as in the pruning step, some prediction methods from the literature were selected to compose different algorithmic flows, together with the ranking method under analysis, in order to evaluate its performance. Therefore, the methods described below are used, as well as **[O]** (CBA), in the experiments in Section V.

[R] — **R**ank-based Prediction of [54]. In the aforementioned work, the prediction takes place via **Rule Set**, but it also takes into account the order of the rules, characteristic of **Rule List**. The proposed method aims to (i) solve the problem of considering only the first rule, resulting in the discarding of other potentially useful rules, as well as (ii) avoid using complex mathematical equations to compute the weight of the rules, making the interpretability of the process difficult. To this end, the authors propose the following process: the class of a given instance is given by the largest sum, obtained between two scores, in one of the possible classes. The first score is obtained by adding the ranks of the rules that match the instance to be predicted in a given class. The second by the number of rules of a given class that match the instance.

[V] — **V**oting prediction. This method is a simplified version of the **Rule Set** methods. Most methods use some weighting between the rules to perform the prediction, for example, in CPAR, which is made from the average of the accuracies of the top-k rules that cover the instance to be classified in each of the classes. In this work a simplified version is used, so that the class is selected through a simple majority voting procedure among the rules that cover the instance.

[P] — **P**robabilistic Prediction of [22]. The prediction method presented by [22] is based on probabilities computed based on the k best rules that cover the instance to be predicted. This method is described in Section III-D.

D. MoMAC

As mentioned, the work of [22] is the most related to the work presented here, as it was designed to simultaneously optimize performance and interpretability. For this reason, it is described here. The others, such as [20] and [21], also mentioned in Section III-A, although related, were not proposed to optimize both aspects.

In [22], although the authors present a complete algorithmic flow, the focus of the work is precisely on the ranking step through the simultaneous use of multiple values, which provide the basis for calculating several existing OMs, namely: $P(A)$, $P(\overline{A})$, $P(B)$, $P(\overline{B})$, $P(AB)$, $P(\overline{A}B)$, $P(A\overline{B})$, $P(\overline{AB})$, $P(A)P(B)$, $P(A|B)$, $P(A|\overline{B})$, $P(\overline{A}|B)$, $P(\overline{A}|\overline{B})$, $P(B|A)$ e $P(\overline{B}|A)$. Thus, ranking can be seen as an aggregation of values, from which a value is generated so that the rules can be ordered in a way that improves the performance and interpretability of the induced models. The algorithm, named MoMAC, is designed to optimize these two criteria. Its pruning and prediction methods were selected and implemented to compose different algorithmic flows, as previously mentioned, to evaluate the ranking method proposed here. Considering the flow in Fig. 1, below is the description of each of the MoMAC steps. It is worth mentioning that the extraction step, as in this work, is not explored by them, with only the mention of a modification in the Apriori algorithm for a more efficient extraction of the CARs.

1) RANKING

The ranking method they proposed can be seen as presented in Fig. 2 (Section IV, page 88868), i.e., a method where it is necessary to specify a set of OMs and an aggregation method. The set of MOs is, in fact, as already mentioned, composed of the 15 probabilities presented above. The aggregation method is a multilayer neural network composed of 2 hidden layers, with 8 and 4 neurons, respectively, and sigmoid activation functions. The network receives the 15 probabilities as input and generates a single value as output, which represents their aggregate value. Based on this learned value, the ranking of the rules is carried out. The proposal was modeled as a multi-objective optimization problem, since two criteria must be optimized: performance and interpretability. The network weights are optimized using a genetic algorithm named NSGA-II [55], enabling the selection of the best individuals considering the model's total error and/or its size. The final result is given by a set of individuals considered optimal, leaving it up to the user to select the model considered ideal.

2) PRUNING

The pruning method is based on the coverage with which the rules, already ordered, match the transactions. From a set of ordered rules, iterations are performed, unstacking them one by one until there are no more rules left or until the coverage of all transactions is greater than the parameter k (default: $k = 3$). Thus, each transaction must be covered by at least k rules. For each rule, transactions with coverage less than k are checked and those that the rule covers are increased by one. If the rule covers at least one transaction then it is added to the model. Uncovered or partially covered transactions ($k < 3$) are used

to determine the default class; otherwise, all transactions are used and the majority class is considered.

3) PREDICTION

In order to predict the class of a given instance x_i, the prediction method they proposed finds all the first k rules that match it. Let R_k be the set of rules found. For each possible class $l \in L$, the probability of the instance x_i belonging to the class is calculated using (1). In equation, z_l, which represents the instance's membership in class l, is divided by the instance's membership in relation to all other classes. z_l is computed via (2), which is divided into two parts: the first, in which the aggregated value ($I(.)$), learned in the ranking step, associated with each rule r_j is added, as long as r_j belongs to the same class as the current class l; the second, in which $\frac{1-I(r_j)}{L-1}$ is added to each rule r_j that does not belong to the current class l. In the end, the class with the highest probability is the one chosen to be assigned to instance x_i.

$$P(y_i = l|x_i) = \frac{z_l}{\sum_{j=1}^{L} z_j} \tag{1}$$

$$z_l = \sum_{\substack{j \in R_k \\ C_j = l}} I(r_j) + \sum_{\substack{j \in R_k \\ C_j \neq l}} \frac{1 - I(r_j)}{L - 1} \tag{2}$$

IV. $AC.RANK_A$

This section presents the proposed rule ranking method via aggregation of OMs, named $AC.Rank_A$. The motivation for proposing it is based on the discussions previously presented, from which it is possible to note that:

- The ranking step presents itself as a step of great importance, as it influences the other stages and, consequently, the performance and interpretability of the model;
- In general, the CSC method (Section II-C, page 88864) is used to order the rules, based on the standard OMs of association rules, i.e., support and confidence;
- Over the years, several OMs have been proposed, as seen in [17] and [18], aiming to overcome limitations of some of the previously existing measures, such as confidence itself (Section II-B);
- This large number of existing OMs has generated a secondary problem, since there is no OM that is suitable for all explorations [19];
- In this context, new proposals have emerged aiming to modify the ACs ranking step, as described in Section III-A, either through: (i) new measures and/or measures already existing in other contexts, but applied at this step in a unique way; (ii) the merge (aggregation) of existing measures in order to use them simultaneously;
- Strategy (ii), adopted in this work, has the advantage of reducing the need to choose a single measure, also considering different aspects (semantics) for ordering the rules;
- Works in this context, such as [20], [21], and [22], have been proposed, which have shown promising results. However, they present problems in relation to the performance and/or interpretability of the generated models. In these works, through the works of [23] and [22], it is possible to verify an inverse relationship between performance and interpretability, i.e., when model performance is high, interpretability is low (and vice versa).

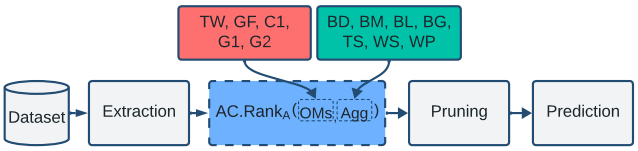


FIGURE 2. $AC.Rank_A$ method to be incorporated into ACs induction flows.

Considering the above, this work presents a method for ranking rules via aggregation of OMs that can be incorporated into ACs induction flows in order to induce models that present a better balance between performance and interpretability. The results demonstrate that $AC.Rank_A$ can maintain the performance of the models, but with better interpretability. Therefore, the ranking step, highlighted in Fig. 2, is modified to incorporate the method proposed here. For $AC.Rank_A$ to work, it must be instantiated with a set of OMs and an aggregation method (parameters *OMs* and *Agg* in the figure). Possible sets of OMs are **[TW]**, **[GF]**, **[C1]**, **[G1]** and **[G2]**, described in Section IV-A. Possible aggregation methods are **[BD]**, **[BM]**, **[BL]**, **[BG]**, described in Section IV-B2, **[TS]**, **[WS]** and **[WP]**, described in Section IV-B1.

A. OBJECTIVE MEASURES SETS

As mentioned in Section II-B, due to the large number of existing measures, choosing one or more measures to explore a set of rules becomes important given the benefits that are obtained when opting for one or another measure. However, choosing which one to use is a difficult problem [19], [33], as there are significant differences in relation to objectives (semantics), among other aspects, that distinguish them. To this end, different ways were proposed to make this choice, such as clustering, adopted in this work. The focus here is the clustering carried out based on the similarity of the OMs in relation to the ordering generated by them in a given set of rules. In this way, three works found in the literature were selected and used, [17], [56], and [57], which are briefly described below.

1) [TW] GROUP

References [17] and [18] present a behavioral study of a set of 61 OMs when used to rank ARs. To do so, the authors group the OMs that present similar behavior based on how they rank a set of rules; thus, those that order the rules in the same way are clustered into the same group. Once the clustering is obtained, a measure from each cluster can be explored, thus reducing the exploration space. Since the work of [18] presents similar results to those of [17], it was not

considered. Thus, in [17], applying the method they proposed, 21 groups of OMs are obtained. A brief description of this work, as well as how the clustering is carried out, can be found at [57].

Of the 61 OMs analyzed by the authors, only 50 were used to perform the clustering. The 11 OMs that were left out are mathematically equivalent to other measures and were, therefore, removed. Thus, one of the sets of OMs used in this work by the $AC.Rank_A$ method is the one proposed by [17], named here **[TW]**. Since the obtained groups represent the equivalence between the OMs in relation to the ranking, **[TW]** group was constructed by selecting one measure from each cluster, since there is no indication of which group or groups to use, different from the methods described below. The selected OMs, which make up the **[TW]** group, are as follows: Support, Prevalence, K-Measure, Least Contradiction, Confidence, TIC, Leverage, Dir, Loevinger, Odds Ratio, Added Value, Accuracy, Lift, J-Measure, Recall, Specificity, Conditional Entropy, Coverage. The choice of a given measure, within each cluster, was based on its computational cost, opting for those with the lowest cost, as well as their relevance in the literature. It is worth mentioning that three groups presented in [17] were not considered, either because they present measures with high computational cost, or because the cluster, according to the authors themselves, can be joined to another group. Therefore, this set of OMs includes measures with completely different semantics when applied to order a set of ARs, i.e., the context of ACs is not taken into account.

2) [GF] GROUP

Reference [56] present a study similar to the works of [17] and [18], but in the context of ACs when used on imbalanced data. The aim is to analyze the behavior of OMs when applied to ACs. Applying the method they proposed, of the 55 measures given as input, only 26 are selected by their clustering approach, which are divided into two groups: the most appropriate measures to be used for datasets with a distribution smaller than 0.4 ($G_<$) and those with a distribution larger than 0.4 ($G_>$). The measures presented in each cluster are as follows:[3]

- $G_<$: Correlation Coeficient, Collective Strength, Kappa, Piatetsky-Shapiro, Putative Causal Dependency, Zhang, Intensity of Implication, Confirm Causal, Goodman–Kruskal, Entropic Implication Intensity 1, Implication Index, Levarage, Added Value;
- $G_>$: Odd Multiplier, Complement Class Support, Conviction, Yule-Q, Sebag–Schoenauer, Yule-Y, Odds Ratio, Confirm Causal, Confirmed Confidence Causal, Example and Counterexample Rate, Ganascia, J-Measure, Confidence.

[3]In the original work, other OMs are found in the respective groups; however, as shown in [17], some of them are equivalent (i.e., produce the same ranks) and, therefore, were removed.

According to the analyses, the $G_<$ group performs well on extremely imbalanced data, with most of the group's measures performing poorly on balanced data; on the other hand, the $G_>$ group performs well on slightly imbalanced data, with most of the group's measures showing better results when used on more balanced data. Thus, another set of OMs used in this work by the $AC.Rank_A$ method is $G_>$, named here **[GF]**. Therefore, this set of OMs includes the most appropriate measures, according to the authors, to rank a set of rules in the context of ACs in relation to the ranking step. The set $G_<$ was not considered, since this work does not deal with imbalanced datasets (see Section V-A). A brief description of this work, as well as how the clustering is carried out, can be found at [57].

3) [C1], [G1] AND [G2] GROUPS

As with [56] and [57] also aims to cluster OMs according to their performance when applied to ACs. However, the authors' study is aimed at balanced or slightly imbalanced datasets. The work was motivated by some limitations, identified by them, of the work of [56], including the use of the CBA algorithm to generate the rules, which is not viable in the context of imbalanced data. Other algorithms are more appropriate in these cases, such as CBA2 [58]. For the clustering they propose to be viable, the authors modify the CBA algorithm, named by them as CBA', in which all model extraction steps are the same as those of CBA, with the exception of ranking. In this case, given a measure m, a rule r_i precedes a rule r_j ($r_i \succ r_j$) if the value of m in r_i is greater than r_j; in case of a tie, r_i was generated before r_j. The authors state that in this way it is possible to access how much each measure influences the performance of the classifier.

Of the 61 OMs presented in [17], [18], and [57] used 44, since there are equivalences between them (as demonstrated in [17]) and that some of them present a high computational cost. After applying the clustering method they proposed, the 44 OMs were grouped into 15 groups, which were ranked according to their average performance measured via F1-Macro, i.e., through the average of the F1-Macro values associated with the models induced by each OM contained in the group. A comparison of the obtained groups with those presented in [56], as well as the impact of these groups applied in the method proposed by [21], are also presented in the paper. Thus, another set of OMs used in this work by the $AC.Rank_A$ method is the one composed by the measures belonging to the best ranked group, Cluster-1, named here **[C1]**. According to the authors, this group contains the most appropriate measures to rank a set of rules in the context of ACs in relation to the ranking step, since the group presents the best $F1_\mu$. The measures that make up this group are as follows: Odd Multiplier, Complement Class Support, Confidence Causal, Loevinger, Added Value, One Way Support, Comfirmed Confidence Causal, LIFT, Confidence, Putative Causal Dependence, Leverage, Confirm Causal, TIC,

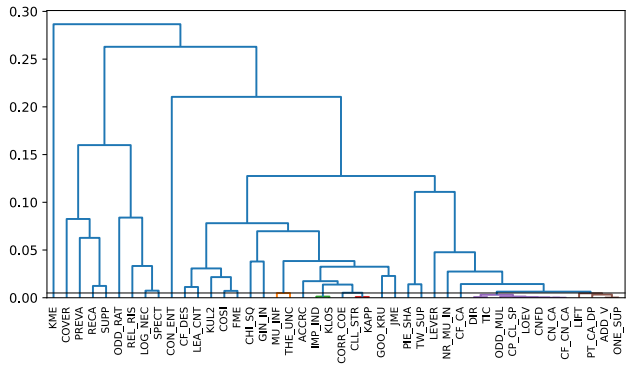


FIGURE 3. Dendrogram obtained after re-executing the experiments presented in [57] considering the modifications made here.

DIR, Normalized Mutual Information. A description of this work, as well as how the clustering is carried out, can be found at [57].

Since $AC.Rank_A$ aims to favor the induction of models that present a better balance between performance and interpretability, we decided to make some adjustments to the method proposed by [57]. Furthermore, it was decided to evaluate, in each group, not only the average F1-Macro, but also the average size of the models. Thus, the following modifications were made:

- the CBA' was modified to incorporate the support measure. In this case, given a measure m, a rule r_i precedes a rule r_j if the value of m in r_i is greater than r_j; in case of a tie, if the support of r_i is greater than r_j; in case of a tie, r_i was generated before r_j. As CBA' is based on CBA, keeping the sorting as in CBA, i.e., changing only the confidence measure for m, the impact of a given measure m on the induction process is better analyzed in relation to CBA;
- aiming to obtain more cohesive groups, a similarity value of 99% was used as a cutoff point in the generated dendrogram (in [57] the value is 95%).

Re-running the experiments, as described in [57], but with the changes presented above, the dendrogram in Fig. 3 was obtained, and the respective groups in Table 2 were found. The groups are ranked according to their average performance ($F1_\mu$), i.e., through the average of the F1-Macro values associated with the models induced by each OM contained in the group. Furthermore, the average size of the models in each group is also presented ($\mathcal{L}_\mu$), which is obtained by averaging the size of the models induced by each OM contained in the group. Unlike the dendrogram presented in [57], it is noted that the cutoff point is close to the zero distance ($\frac{1-0,99}{2} = 0,005 = 0,5\%$), since the similarity value used was high (99%). In this case, 26 groups were obtained.

Thus, the last two sets of OMs used in this work by the $AC.Rank_A$ method are those composed by the measures belonging to the two best ranked groups, Group-1 and Group-2, named here, respectively, **[G1]** and **[G2]**. The two are the ones with the highest average F1-Macro values ($F1_\mu$). However,

TABLE 2. Groups of OMs obtained after re-executing the experiments.

ID	$F1_\mu$	$\mathcal{L}_\mu$	OMs
1	0.7357	136.0328	Complement Class Support; Confidence; Confidence Causal; Confirmed Confidence Causal; DIR; Loevinger; Odd Multiplier; TIC
2	0.7309	98.8442	Added Value; LIFT; One Way Support; Putative Causal Dependency
3	0.7218	186.9953	Leverage
4	0.7202	54.7558	Confirm Causal
5	0.7031	58.486	Normalized Mutual Information
6	0.6928	21.8942	Implication Index; Klosgen
7	0.6886	19.6023	J-Measure
8	0.6878	32.2581	Accuracy
9	0.6748	15.945	Collective Strength; Correlation Coefficient; Kappa
10	0.6612	29.2744	Confirm Descriptive; Least Contradiction
11	0.6494	7.6453	Piatetsky Shapiro; Two Way Support
12	0.6437	11.6581	Mutual Information; Theil Uncertainty Coefficient
13	0.6342	108.9326	K-Measure
14	0.6315	13.0163	Goodman Kruskal
15	0.63	13.0302	Cosine; F-Measure
16	0.6238	11.007	Kulczynski 2
17	0.6151	87.793	Odds Ratio
18	0.614	16.5953	Gini Index
19	0.5916	16.1605	Chi-Square
20	0.5399	25.4605	Relative Risk
21	0.4879	36.9233	Conditional Entropy
22	0.4825	3.2651	Logical Necessity; Specificity
23	0.4366	2.7814	Recall
24	0.4252	2.7163	Support
25	0.4018	2.5	Prevalence
26	0.3606	2.9	Coverage

although **[G1]** presents a small difference in relation to **[G2]** in terms of F1-Macro, in terms of size, **[G2]** obtains a significant advantage over **[G1]**, obtaining an average value around 28% lower, a significant gain compared to a drop of just 0.48% in F1-Macro. For this reason, both groups were selected.

Finally, through Fig. 4 it is possible to visualize the inverse relationship between performance and interpretability, based on the data in Table 2. The figure was constructed via linear regression. The x axis indicates the group ID, which corresponds to a rank value: the higher the group's average performance, the lower its ID. This construction allows visualization of the balance between performance and interpretability. Associated with the average F1-Macro of each group (ID), i.e., $F1_\mu$ (blue line), we have the corresponding average size, i.e., $\mathcal{L}_\mu$ (orange line). It is noted that the regressions generally present an inverse trend to what is understood as ideal for the models, i.e., high $F1_\mu$ and low $\mathcal{L}_\mu$; thus, in this case, it is noted that the greater the value of $F1_\mu$, the greater the value of $\mathcal{L}_\mu$ (and vice versa), i.e., it is difficult to balance performance and interpretability.

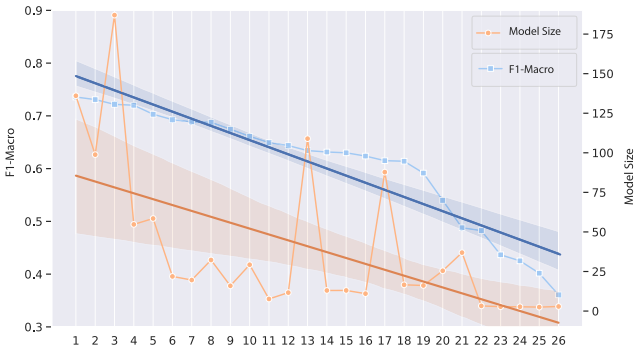


FIGURE 4. Relationship between performance and interpretability based on data from Table 2.

TABLE 3. Alternatives x criteria matrix.

	C_1	C_2	...	C_n
A_1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,n}$
A_2	$x_{2,1}$	$x_{2,2}$	...	$x_{2,n}$
...	...	...	...	...
A_m	$x_{m,1}$	$x_{m,2}$	...	$x_{m,n}$

TABLE 4. Example of a set of alternatives and their respective values by criteria.

W	1	2	0.5

	C_1	C_2	C_3
A_1	0.2	0.67	0.02
A_2	0.1	0.5	0
A_3	0.2	0.33	0.1

B. AGGREGATION METHODS

As already mentioned, for *AC.Rank*$_A$ to work it is necessary to also define the aggregation methods to be used. Previous studies have already used Pareto [20], ensemble of classifiers [21] and optimization through genetic algorithm [22]. However, as seen, they present problems in relation to performance and/or interpretability. Therefore, in this work we chose to use multi-criteria decision analysis (MCDA) and ranking aggregation. The reason is due to the recent use of MCDA methods in the context of ARs, such as ELECTRE I in [24] and [25] and ELECTRE II in [26]. The ranking aggregation methods are similar to those of MCDA and, therefore, were also considered. These methods are proposed to support decision making such as the one proposed in this work. Therefore, this section presents such methods. Finally, before presenting the methods themselves, it is important to mention that:

- The *AC.Rank*$_A$ method is executed in the same way as the aggregation methods described below; however, in this case, the matrices $m \times n$, presented in Tables 3 and 6, are matrices of Rules x OMs, in which the measures that compose them are those belonging to the selected sets, i.e., **[TW]**, **[GF]**, **[C1]**, **[G1]** and **[G2]**;
- All OMs, in MCDA methods, receive the same weight with regard to the W vector.

1) MULTI-CRITERIA DECISION ANALYSIS (MCDA)

MCDA methods were created to assist decision makers in choosing the best options (alternatives) based on a set of criteria, which may or may not have different weights [59]. The use of MCDA methods also fits the task of aggregating OMs, as they can assist in decision making when evaluating several rules (options) on several criteria (OMs).

Due to the large number of existing methods, [59] provides a framework to support the choice of the most appropriate methods for a given problem. This work utilized this framework to select the methods to be used. Among the methods suggested by the framework, TOPSIS [60] was selected, which is commonly cited in the literature. Furthermore, the two most basic and classic MCDA methods were also considered, namely: WSM [61] and WPM [61].

In order to adapt the terms used in this work with those in the MCDA literature, OMs are named criteria and CARs are named alternatives. Thus, the aim is to order the alternatives (rules) according to the established criteria (OMs), in order to prioritize, through the first positions of a given ranking, the best alternatives. In this context, $A = \{A_i | i = 1, \ldots, m\}$ is defined as the set of m alternatives, $C = \{C_j | j = 1, \ldots, n\}$ the set of n criteria and $W = \{W_j | j = 1, \ldots, n\}$ the set of weights associated with the criteria. The data are organized into an m X n matrix, as in Table 3, where x_{ij} indicates the intersection of an alternative i and a criterion j. In order to support the understanding of the methods described below, Table 4 will be used, which contains three alternatives and three criteria, in addition to the weights of the respective criteria (line W).

[WS] — *WSM [61]:* The value V_i of each alternative A_i is given by the sum of the n criteria of the alternative A_i multiplied by their respective weights, where $V_i = \sum_{j=1}^{n} W_j \times x_{ij}$. Using the values in Table 4 we would have the value of 1.55 for the alternative A_1 ($V_1 = 1 \times 0.2 + 2 \times 0.67 + 0.5 \times 0.02$). For alternatives A_2 and A_3 the values of $V_2 = 1.1$ and $V_3 = 0.91$. Considering these values, the alternatives would be ranked in the following order $\{A_1, A_2, A_3\}$, with the alternative A_1 having preference over the others.

[WP] — *WPM [61]:* The value V_i of each alternative A_i is given by the product among the criteria of the alternative A_i, raised to their respective weights, where $V_i = \prod_{j=1}^{n} x_{ij}^{W_j}$. Using the values in Table 4 we would have the value of 0.012 for the alternative A_1 ($V_1 = 0.2^1 \times 0.67^2 \times 0.02^{0.5}$). For alternatives A_2 and A_3 the values of $V_2 = 0$ and $V_3 = 0.006$. Therefore, the alternatives would be ranked in the following order $\{A_1, A_3, A_2\}$, with the alternative A_1 having preference over the others.

[TS] — *TOPSIS [60]:* This method is divided into three parts. In the first part, the matrix values are normalized by dividing each value by the square root of the sum of the squared values, which are, finally, weighted by the weight of the criterion, according to (3). In the second part, for each alternative A_i, the Euclidean distance of its values, in each

TABLE 5. Values from Table 4 after normalization. The largest values in each column are in bold and the smallest in italics.

W	1	2	0.5
	C_1	C_2	C_3
A_1	**0.67**	**1.49**	0.1
A_2	*0.33*	1.11	*0*
A_3	**0.67**	*0.73*	**0.49**

of the criteria, is computed in relation to the best (D_i^{max}) and worst (D_i^{min}) values of each criterion, as seen in (4) and (5). Finally, in the third part, the value V_i is obtained by dividing D_i^{min} by the sum between D_i^{max} and D_i^{min}, as seen in (6).

$$x_{ij}^{normalized} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \times W_j \tag{3}$$

$$D_i^{max} = \sqrt{\sum_{j=1}^{n} (x_{i,j}^{normalized} - \max\{C_j\})^2} \tag{4}$$

$$D_i^{min} = \sqrt{\sum_{j=1}^{n} (x_{i,j}^{normalized} - \min\{C_j\})^2} \tag{5}$$

$$V_i = \frac{D_i^{min}}{D_i^{max} + D_i^{min}} \tag{6}$$

Using the values from Table 4, after the first part, the normalized values would be those presented in Table 5. Equation (7) presents the calculation for $x_{1,1}$. In the second part the distances are computed. Equations (8) and (9) present the values obtained for the alternative A_1. Finally, the value of V_1 is obtained by (10), resulting in a value of 0.68. For alternatives A_2 and A_3 we would have the values of $V_2 = 0.35$ and $V_3 = 0.44$. Therefore, the alternatives would be ranked in the following order $\{A_1, A_3, A_2\}$, with the alternative A_1 having preference over the others.

$$x_{1,1} = \frac{0.2}{\sqrt{0.2^2 + 0.1^2 + 0.2^2}} \times 1 = 0.67 \tag{7}$$

$$D_1^{\max} = \sqrt{(0.67 - 0.67)^2 + (1.49 - 1.49)^2 + (0.1 - 0.49)^2}$$
$$D_1^{\max} = 0.39 \tag{8}$$

$$D_1^{\min} = \sqrt{(0.67 - 0.33)^2 + (1.49 - 0.73)^2 + (0.1 - 0)^2}$$
$$D_1^{\min} = 0.83 \tag{9}$$

$$V_1 = \frac{0.83}{0.39 + 0.83} = 0.68 \tag{10}$$

2) RANKING AGGREGATION

The ranking aggregation methods are similar to those of MCDA, as they aim to evaluate multiple criteria at the same time, and, therefore, were also considered. However, the methods in this family combine the ranks generated by different criteria to generate a final ranking [62]; thus, these methods do not consider the values themselves as in MCDA. Thus, the methods are more robust to *outliers* [63] and invariant to transformations and normalizations, as long as the order of the alternatives is maintained. The Borda [64] methods are the most traditional of this family, being widely used due to their simplicity. Thus, its four versions were selected here.

TABLE 6. Example of a set of alternatives and their respective ranks by criteria.

	C_1	C_2	C_3
A_1	2.5	3	2
A_2	1	2	1
A_3	2.5	1	3

In order to guarantee standardization between the selected methods, both in examples and explanations, the same MCDA nomenclature was adopted. As a reference for the equations, Table 3 was considered. For the examples, Table 6 was constructed from Table 4 modifying its values to rank values. It is worth mentioning that alternatives with equal values receive the average of their ranks. Ranks are expressed in ascending order, in order to preserve the premise that the higher the value, the better the rank.

[BM] — *Borda Arithmetic Mean [63]:* The value V_i of each alternative A_i is given by the arithmetic mean of the values of the alternative in each of the criteria, where $V_i = \frac{\sum_{j=1}^{n} x_{ij}}{n}$. Using the values from Table 6 we would have the value of 2.5 for the alternative A_1 ($V_1 = mean(2.5, 3, 2)$). Being $V_2 = 1.33$ and $V_3 = 2.17$, the ranking is given by $\{A_1, A_3, A_2\}$, with the alternative A_1 having preference over the others.

[BD] — *Borda Median [63]:* The value V_i of each alternative A_i is given by the median of the values of the alternative in each of the criteria, where $V_i = median\{x_{i1}, \ldots, x_{in}\}$. Using the values from Table 6 we would have the value of 2.5 for the alternative A_1 ($V_1 = median(2, 2.5, 3)$). Being $V_2 = 1$ and $V_3 = 2.5$, the ranking is given by $\{A_1, A_3, A_2\}$, with the alternative A_1 having preference over the others.

[BG] — *Borda Geometric Mean [63]:* The value V_i of each alternative A_i is given by the geometric mean, which refers to the product of the values raised to the inverse of the number of criteria, i.e., $V_i = \sqrt[n]{\prod_{j=1}^{n} x_{ij}}$. Using the values from Table 6 we would have the value of 2.47 for the alternative A_1 ($V_1 = \sqrt[3]{2.5 \times 3 \times 2}$). Being $V_2 = 1.26$ and $V_3 = 1.96$, the ranking is given by $\{A_1, A_3, A_2\}$, with the alternative A_1 having preference over the others.

[BL] — *Borda 2-norm [63]:* The value V_i of each alternative A_i is given by $V_i = \sqrt[n]{\frac{\sum_{j=1}^{n} x_{ij}^2}{n}}$, named the Euclidean norm, a special case of p-norm, where $p = 2$. Using the values from Table 6 we would have the value of 1.86 for the alternative A_1 ($V_1 = \sqrt[3]{\frac{6.25+9+4}{3}}$). Being $V_2 = 1.26$ and $V_3 = 1.76$, the ranking is given by $\{A_1, A_3, A_2\}$, with the alternative A_1 having preference over the others.

V. EXPERIMENTAL EVALUATION

In order to evaluate the proposed method, this section presents the experiments carried out, as well as the results and discussions. Since the aim of the work is to propose a rule ranking method via aggregation of OMs that can be

incorporated into ACs induction flows, the experiments were designed to test it in conjunction with different pruning and prediction methods. In this way, several flows were created, using both the CBA ranking, i.e., **[CSC]** (Section II-C, page 88865), and the one proposed here, so that an analysis of its impact could be carried out. All flows were created based on CBA, since it is the most used algorithm in the family [15]. Therefore, this work considers it as a baseline.

Fig. 5 follows the same flow as Fig. 2, i.e., extraction, ranking, pruning and prediction, but it is instantiated with the methods to be explored in each one of the aforementioned steps. Each combination generates a different flow to be used in inducing an AC model. Thus, the following methods are available at each step:

- **Extraction**: Apriori (AP) (Section II-A);
- **Ranking**: $AC.Rank_A$ (Section IV); **[CSC]** (Section II-C, page 88865);
- **Pruning**: **[M1]** (Section II-C, page 88865); **[DN]** (Section III-B, page 88866); **[CV]** (Section III-B, page 88867);
- **Prediction**: **[O]** (Section II-C, page 88865); **[V]** (Section III-C, page 88867); **[R]** (Section III-C; page 88867); **[P]** (Section III-C, page 88867).

Each flow is executed sequentially, where the output of the method of one step is used as input to the method of the next step. It is worth noting that although the ''Set of OMs'' and ''Aggregation Methods'' boxes, belonging to the $AC.Rank_A$ method, have been arranged sequentially, just to indicate the possible combinations, their execution occurs simultaneously. Finally, it is worth mentioning that the methods highlighted in orange in Fig. 5 are those used in CBA; thus, the combination **[CSC]** + **[M1]** + **[O]** refers to its execution.

Considering the possible flows, we obtain a total of 432 possibilities: 420 related to the $AC.Rank_A$ ranking and 12 to the **[CSC]** ranking. Each of them induces a different model, which is evaluated in terms of performance (F1-Macro) and interpretability (model size ($\mathcal{L}$)) (see Section II-D). The 432 flows were explored in 43 datasets, summing up to a total of 185,760 experiments ($432 \times 43 \times 10$), as the measures were estimated, in each dataset, via stratified 10-fold cross-validation. Therefore, more specific details about the experimental setup are presented in Section V-A. In Section V-B the results are presented and discussed. Finally, in Section V-C a complementary analysis, in relation to works in the literature, is carried out.

A. EXPERIMENTAL SETUP DETAILS

Although the methods to be used in each of the steps have already been presented, as well as in Fig. 5, it is necessary that some details, related to the execution of the experiments, be here described.

1) DATASETS

43 datasets, available in the KEEL tool repository,[4] were used. The repository provides 76 datasets in total (section ''Standard classification data sets''). However, of these, only those with balanced, or partially balanced, distribution were selected, i.e., those with proportions of up to 1:2.5 between the minority and majority classes. The reason is due to the fact that the algorithms used here do not deal with the issue of imbalance between classes. Table 7 presents the selected datasets with their respective characteristics, namely: number of transactions (#Transactions); number of features (#Features); number of distinct items after converting the dataset to transactional format, where items are represented by an <attribute = value> pair (#Distinct Items); number of classes (#Classes); the imbalance rate (IR) of the sets ($IR = \frac{\#Maj}{\#Min}$); the minimum support value (**sup-min**) used to perform rule extraction (see below) (#Support).

[4]https://sci2s.ugr.es/keel/datasets.php

2) PREPROCESSING

All datasets were pre-processed. Numerical attributes were discretized, both real and integers containing more than ten unique values. It is worth mentioning that in order to avoid data leakage, discretization was performed only on the folds used as training in a given cross-validation run. In other words, the transformations were performed only on the training set and later applied to the test set. The discretization algorithm used was the one proposed by [65]. After discretization, attributes that presented unique values were excluded.

3) RULE EXTRACTION

Unlike the other steps, the extraction stage was the only one that used just one algorithm, in this case, Apriori. For it to be executed and the rules extracted, it is necessary to set the minimum support (**sup-min**) and minimum confidence (**conf-min**). In order to avoid its impact on the results, we chose to reset the conf-min value to zero, therefore extracting all possible rules within the specified sup-min. In order to avoid a combinatorial explosion in the number of rules, sup-min was set so as not to generate sets of rules above ten thousand rules. The average sup-min values used for each dataset can be found in Table 7 (column ''Support %''). Furthermore, it was defined that the maximum size of the itemset would be equal to 5 items, as frequently used, which implies rules with a maximum of 4 items in the antecedent. An exception was made in the sets *Ionosphere*, *Spambase*, *Texture* and *Satimage*, where the size was set to 3 items, since any extraction with 4 or 5 items resulted in generation of more than 10 thousand rules.

4) EVALUATION CRITERIA

As already mentioned in Section II-D, the measures considered were the F1-Macro, in terms of performance, and the model size $\mathcal{L}$, in terms of interpretability, both estimated via stratified 10-fold cross-validation. In order to compare the obtained results, statistical tests were carried out using the Friedman test with $\alpha = 0.05$ and the Nemenyi post-hoc test (Friedman + Nemenyi), together with the critical difference diagrams (CD). Although other combinations are possible, such as

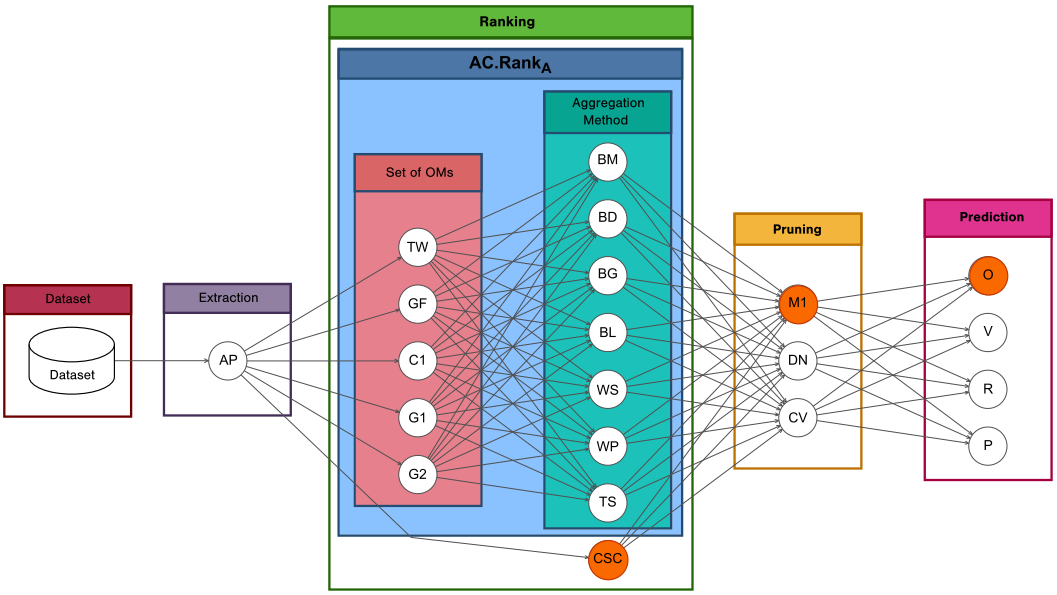


FIGURE 5. Methods used in each induction step, which, together, generate the flows evaluated in this work.

Quade + Nemenyi, the choice was based on [66]. To this end, although good packages are currently available, such as StaTDS,[5] we chose to use the R Stats[6] package, together with scikit-posthocs,[7] as they are more consolidated in the community.

B. RESULTS AND DISCUSSION

The results were organized and analyzed in order to verify the impact of the $AC.Rank_A$ method, in relation to **[CSC]** (baseline), in many ACs induction flows. In order to better understand how the analysis was performed consider Tables 8 and 9. The first presents the results of F1-Macro and the second of model size ($\mathcal{L}$), both referring to the Hayes-roth set with $AC.Rank_A$ instantiated with the set of OMs **[TW]** in the different aggregation methods. Each line contains the result of 8 different flows. Line 1, for example, is associated with the following flows:

- **[CSC]+[M1]+[O]**;
- **[TW+BD]+[M1]+[O]**;
- **[TW+BG]+[M1]+[O]**;
- **[TW+BL]+[M1]+[O]**;
- **[TW+BM]+[M1]+[O]**;
- **[TW+TS]+[M1]+[O]**;
- **[TW+WP]+[M1]+[O]**;
- **[TW+WS]+[M1]+[O]**.

Note that in all of them pruning is instantiated via **[M1]** method and prediction via **[O]**. What changes between flows is the ranking step, which is instantiated by both **[CSC]** and $AC.Rank_A$ in all possible aggregation methods together with the set of OMs **[TW]**. In line 2, for example, all flows have the pruning fixed to **[M1]** and the prediction to **[P]**, with the rankings varying as in line 1. The green column presents the F1-Macro values obtained when flows use **[CSC]** in their ranking step. The columns in blue show the $AC.Rank_A$ method instantiated via MCDA aggregation methods together with the set of OMs **[TW]**. Finally, the columns in yellow show the $AC.Rank_A$ method instantiated via ranking aggregation methods together with the set of OMs **[TW]**.

Analyzing the first line of Table 8, it is noted that CBA, combination **[CSC]** + **[M1]** + **[O]**, can be compared to a CBA' in which all steps are the same as CBA, with the exception of the ranking step, replaced by $AC.Rank_A$. In this case, it is observed that in all combinations in which $AC.Rank_A$ is used, it presents better F1-Macro values, obtaining gains of up to 11% (from 0.7936 (79.36%) in **[CSC]** + **[M1]** + **[O]** to 0.8854 (88.54%) in **[TW+BM]** + **[M1]** + **[O]** and **[TW+WS]** + **[M1]** + **[O]**). The results obtained on all of the 432 explored flows, both in terms of F1-Macro and model size, are available in https://bit.ly/resultados-experimentos-defesa.

Given the above, it is possible to check whether flows that use $AC.Rank_A$ as a ranking method are better or not compared to those that use the standard CBA method (**[CSC]**) in terms of performance and interpretability. The analysis is done by grouping the results by set of OMs, as in Table 8. To this end, statistical tests are performed (Friedman with Nemenyi's post-hoc test (see Section V-A)). The tests are applied to each flow in which the pruning and prediction method is fixed and the ranking method is varied. The aim is to verify whether there are significant differences in relation to F1-Macro and model size between the ranking methods evaluated. Using Table 8 once again, tests are applied to the configurations (flows) present in each row of the table, but

[5]https://github.com/kdis-lab/StaTDS
[6]https://www.rdocumentation.org/packages/stats/
[7]https://github.com/maximtrp/scikit-posthocs

TABLE 7. Characteristics of the datasets used in the experiments.

Dataset	#Transactions	#Features	#Distinct Items	#Classes	IR	Support (%)
Australian	690	14	37	2	1.25	4.4122
Banana	5300	2	12	2	1.23	0.021
Bands	365	19	44	2	1.7	16.4378
Breast	277	9	41	2	2.42	0.8022
Bupa	345	6	12	2	1.38	0.3221
Chess	3196	36	73	2	1.09	36.8377
Contraceptive	1473	9	28	3	1.89	0.611
Crx	653	15	53	2	1.21	4.7304
German	1000	20	74	2	2.33	7.6889
Hayes-roth	160	4	15	3	2.1	0.6944
Heart	270	13	33	2	1.25	3.3333
Housevotes	232	16	32	2	1.15	7.6629
Ionosphere	351	33	125	2	1.79	0.5697
Iris	150	4	11	3	1	0.7407
Led7digit	500	6	13	10	1.55	0.2222
Letter	20000	16	128	26	1.11	0.9872
Magic	19020	10	73	2	1.84	0.6812
Mammographic	830	5	24	2	1.06	0.1339
Marketing	6876	13	75	9	2.49	1.3089
Monk-2	432	6	17	2	1.12	0.2572
Movement libras	360	90	243	15	1.05	5.9259
Mushroom	5644	21	97	2	1.62	11.5363
Optdigits	5620	62	298	10	1.03	9.8754
Penbased	10992	16	162	10	1.09	0.9613
Phoneme	5404	5	30	2	2.41	0.0206
Pima	768	8	19	2	1.87	0.1447
Ring	7400	20	156	2	1.02	4.3799
Saheart	462	9	19	2	1.89	0.2405
Satimage	6435	36	440	6	2.45	1.4331
Segment	2310	18	171	7	1	3.3526
Sonar	208	59	120	2	1.14	2.8311
Spambase	4597	57	158	2	1.54	2.4944
Splice	3190	60	283	3	2.16	3.8384
Tae	151	5	10	3	1.06	0.7358
Texture	5500	40	470	11	1	1.497
Tic-tac-toe	958	9	27	2	1.89	0.9279
Titanic	2201	2	4	2	2.1	4.3671
Twonorm	7400	20	126	2	1	1.3514
Vehicle	846	18	72	4	1.1	6.4485
Vowel	990	12	44	11	1	2.1437
Wdbc	569	30	92	2	1.68	21.6171
Wine	178	13	36	3	1.48	3.4953
Wisconsin	683	9	89	2	1.86	0.488

TABLE 8. F1-Macro results on the hayes-roth set with $AC.Rank_A$ instantiated with the set of OMs [TW] in the different aggregation methods.

Pru	Pre	Ranking							
		[CSC]	[TW]						
			[BD]	[BG]	[BL]	[BM]	[TS]	[WP]	[WS]
[M1]	[O]	0.7936	0.8298	0.8752	0.8695	0.8854	0.8752	0.8752	0.8854
	[P]	0.7936	0.7173	0.7227	0.7215	0.7345	0.7278	0.7227	0.7345
	[R]	0.7936	0.8182	0.7914	0.8152	0.7981	0.7914	0.7914	0.7981
	[V]	0.7623	0.7549	0.6912	0.7029	0.7063	0.6842	0.6912	0.7063
[CV]	[O]	0.7963	0.8459	0.8802	0.8905	0.8905	0.8802	0.8802	0.8905
	[P]	0.8331	0.8234	0.8136	0.8332	0.8176	0.8136	0.8136	0.8176
	[R]	0.8372	0.8367	0.822	0.8322	0.8323	0.8111	0.822	0.8323
	[V]	0.7526	0.7449	0.7036	0.7769	0.7179	0.7033	0.7036	0.7179
[DN]	[O]	0.7874	0.6467	0.8524	0.8334	0.837	0.8366	0.8524	0.837
	[P]	0.655	0.5262	0.6419	0.5618	0.623	0.5857	0.6419	0.623
	[R]	0.764	0.6515	0.8233	0.8334	0.8245	0.8005	0.8233	0.8245
	[V]	0.7021	0.5952	0.7313	0.6688	0.6989	0.6963	0.7313	0.6989

across the 43 datasets. Considering line 1 as an example, the Friedman test is applied to the 8 configurations listed above (bullets) across the 43 datasets, i.e., the test is applied to a 43×8 matrix. The significance level used was 5%. In cases where the null hypothesis is rejected, i.e., there is no difference between the rankings, the Nemenyi post-hoc test is applied,

TABLE 9. Model size results on the hayes-roth set with $AC.Rank_A$ instantiated with the set of OMs [TW] in the different aggregation methods.

Pru	Pre	Ranking							
		[CSC]	[TW]						
			[BD]	[BG]	[BL]	[BM]	[TS]	[WP]	[WS]
[M1]	[O]	28.5	26.8	23.3	21.7	23.6	22.1	23.3	23.6
	[P3]	28.5	26.8	23.3	21.7	23.6	22.1	23.3	23.6
	[R]	28.5	26.8	23.3	21.7	23.6	22.1	23.3	23.6
	[V]	28.5	26.8	23.3	21.7	23.6	22.1	23.3	23.6
[CV]	[O]	131.1	123.8	71.3	69.3	70	67.1	71.3	70
	[P3]	131.1	123.8	71.3	69.3	70	67.1	71.3	70
	[R]	131.1	123.8	71.3	69.3	70	67.1	71.3	70
	[V]	131.1	123.8	71.3	69.3	70	67.1	71.3	70
[DN]	[O]	31.1	26.2	15.4	20	16.9	15.9	15.4	16.9
	[P3]	31.1	26.2	15.4	20	16.9	15.9	15.4	16.9
	[R]	31.1	26.2	15.4	20	16.9	15.9	15.4	16.9
	[V]	31.1	26.2	15.4	20	16.9	15.9	15.4	16.9

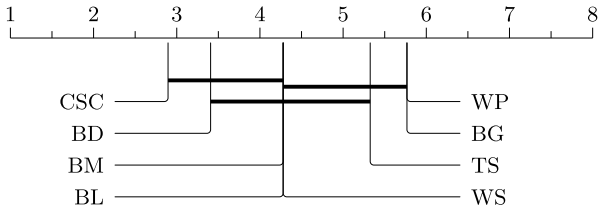


FIGURE 6. Critical difference (CD) diagram, in relation to the F1-Macro values, to compare the 8 configurations (flows) present in line 1 of Table 8 across the 43 datasets.

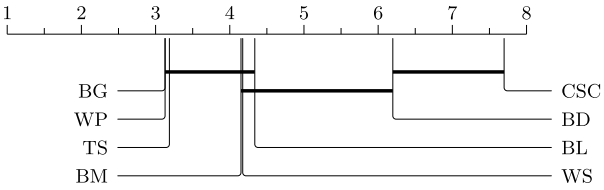


FIGURE 7. Critical difference (CD) diagram, in relation to model sizes, to compare the 8 configurations (flows) present in line 1 of Table 9 across the 43 datasets.

also obtaining the critical difference (CD) diagram, as shown in Fig. 6. This is the diagram, after applying the tests, for the configurations present in line 1 of Table 8 across the 43 datasets in relation to the F1-Macro values. The same procedure is also carried out, under the same conditions, in relation to the size of the model, following Table 9. Fig. 7 shows the generated diagram.

In the CD diagrams, the lines leaving the enumerated axis indicate the average rank of the respective configuration. Lines connected by a bar are equivalent to configurations that do not present critical differences between them. The analysis carried out, for example, in Fig. 6, indicates that the configuration referring to the **[CSC]** method presents an average rank of approximately 2.9, showing no difference in relation to the methods **[BD]**, **[BM]**, **[BL]** and **[WS]**. Thus, analyzing the CD diagrams in Fig. 6 and Fig. 7 at the same time, since a balance is sought between F1-Macro and model size, i.e., performance and interpretability, it is possible to note that:

- In relation to F1-Macro (Fig. 6), although **[CSC]** appears in the first position, standing out with the lowest average rank, it does not present a significant statistical difference against $AC.Rank_A$ instantiated with methods **[BD]**, **[BM]**, **[BL]** and **[WS]** with the set of OMs **[TW]**;
- Regarding model size (Fig. 7), **[CSC]** appears in the last position, standing out with the highest average rank, presenting a significant statistical difference in relation to the $AC.Rank_A$ instantiated with methods **[BG]**, **[WP]**, **[TS]**, **[BM]**, **[WS]** and **[BL]** with the set of OMs **[TW]**;
- Thus, considering both criteria at the same time, it is noted that $AC.Rank_A$, instantiated with the methods **[BM]**, **[BL]** and **[WS]**, maintains the performance of the models in relation to **[CSC]**, but with better interpretability.

This analysis methodology was used in all other combinations, in order to compare the **[CSC]** ranking (baseline) with the $AC.Rank_A$. As the number of CD diagrams generated is large, in order to facilitate analysis, Table 10 was constructed from them. In the first column there is the set of OMs used to instantiate $AC.Rank_A$, in the second the pruning method, in the third the prediction method and in the other columns the aggregation method used to instantiate $AC.Rank_A$. Each cell of the table expresses the existence or not of a critical difference between the rankings **[CSC]** and $AC.Rank_A$, in both criteria analyzed, in a given configuration. The first value in the cell refers to the existence of a difference in relation to the F1-Macro and the second in relation to the size of the model ($\mathcal{L}$). The indication of the existence or not of a statistical difference is represented by three distinct symbols: $\equiv$ for cases where there is no difference; ▲ for cases where there is a difference and it is positive for the ranking $AC.Rank_A$; ▼ for cases where there is a difference and it is negative for $AC.Rank_A$, i.e., it is positive for the ranking **[CSC]**.

To illustrate, consider the CD diagrams in Fig. 6 and Fig. 7, which correspond to the analyzes in the first line of Table 10). As noted previously, the **[BD]**, **[BL]**, **[BM]** and **[WS]** methods, in **[TW]** set, did not present statistical differences in relation to **[CSC]** regarding F1-Macro; therefore, in this table, they all have a $\equiv$ in the first position of their respective cells. On the other hand, **[BG]**, **[TS]** and **[WP]**, as they showed a negative difference in relation to **[CSC]**, have a ▼ in the first position of their respective cells. Regarding the model size criterion,

TABLE 10. Results referring to the critical difference analysis between the *AC.Rank*$_A$ and [CSC] ranking methods in different induction flows.

OMs	Pru	Pre	Aggregation Method													
			BD		BG		BL		BM		TS		WP		WS	
			F1	$\mathcal{L}$	F1	$\mathcal{L}$	F1	$\mathcal{L}$	F1	$\mathcal{L}$	F1	$\mathcal{L}$	F1	$\mathcal{L}$	F1	$\mathcal{L}$
TW	M1	O	≡	≡	▼	▲	≡	▲	≡	▲	▼	▲	▼	▲	≡	▲
		V	≡	≡	▼	▲	≡	▲	≡	▲	▼	▲	▼	▲	≡	▲
		R	≡	≡	▼	▲	≡	▲	≡	▲	▼	▲	▼	▲	≡	▲
		P	≡	≡	▼	▲	▼	▲	≡	▲	▼	▲	▼	▲	≡	▲
	CV	O	≡	≡	▼	▲	≡	▲	▼	▲	▼	▲	▼	▲	▼	▲
		V	≡	≡	▼	▲	≡	▲	≡	▲	▼	▲	▼	▲	≡	▲
		R	≡	≡	≡	▲	≡	▲	≡	▲	▼	▲	≡	▲	≡	▲
		P	≡	≡	▼	▲	≡	▲	≡	▲	▼	▲	▼	▲	≡	▲
	DN	O	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		V	≡	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲
		R	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		P	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲
GF	M1	O	≡	≡	≡	▲	≡	▲	≡	≡	≡	▲	≡	▲	≡	≡
		V	≡	≡	≡	▲	≡	▲	≡	≡	≡	▲	≡	▲	≡	≡
		R	≡	≡	≡	▲	≡	▲	≡	≡	≡	▲	≡	▲	≡	≡
		P	≡	≡	≡	▲	≡	▲	≡	≡	≡	▲	≡	▲	≡	≡
	CV	O	≡	≡	≡	≡	≡	▲	≡	≡	≡	▲	≡	≡	≡	≡
		V	≡	≡	≡	≡	≡	▲	≡	≡	≡	▲	≡	≡	≡	≡
		R	≡	≡	≡	≡	≡	▲	≡	≡	≡	▲	≡	≡	≡	≡
		P	≡	≡	≡	≡	≡	▲	≡	≡	≡	▲	≡	≡	≡	≡
	DN	O	▲	≡	≡	▲	▲	▲	▲	▲	≡	▲	≡	▲	▲	▲
		V	≡	≡	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		R	≡	≡	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		P	▼	≡	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲
C1	M1	O	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		V	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		R	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		P	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
	CV	O	≡	≡	≡	▼	≡	≡	≡	≡	≡	▼	≡	▼	≡	≡
		V	▲	≡	≡	▼	≡	≡	≡	≡	≡	▼	≡	▼	≡	≡
		R	≡	≡	≡	▼	≡	≡	≡	≡	≡	▼	≡	▼	≡	≡
		P	≡	≡	≡	▼	≡	≡	≡	≡	≡	▼	≡	▼	≡	≡
	DN	O	▲	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		V	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		R	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		P	▼	≡	≡	≡	▼	≡	▼	≡	▼	≡	≡	≡	▼	≡
G1	M1	O	≡	≡	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼
		V	≡	≡	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼
		R	≡	≡	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼
		P	≡	≡	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼
	CV	O	≡	≡	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼
		V	≡	≡	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼
		R	≡	≡	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼	≡	▼
		P	≡	≡	≡	▼	▲	▼	≡	▼	≡	▼	≡	▼	≡	▼
	DN	O	≡	▲	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		V	≡	▲	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		R	≡	▲	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡
		P	▼	▲	▼	≡	▼	≡	▼	≡	≡	≡	▼	≡	▼	≡
G2	M1	O	≡	≡	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		V	≡	≡	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		R	≡	≡	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		P	≡	≡	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
	CV	O	≡	▼	≡	≡	≡	▲	≡	≡	≡	≡	≡	≡	≡	≡
		V	≡	▼	≡	≡	≡	▲	≡	≡	≡	≡	≡	≡	≡	≡
		R	≡	▼	≡	≡	≡	▲	≡	≡	≡	≡	≡	≡	≡	≡
		P	≡	▼	≡	≡	≡	▲	≡	≡	≡	≡	≡	≡	≡	≡
	DN	O	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		V	▼	▲	≡	▲	▼	▲	▼	▲	▼	▲	≡	▲	▼	▲
		R	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲	≡	▲
		P	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲	▼	▲

[BG], **[BL]**, **[BM]**, **[TS]**, **[WP]** and **[WS]**, as they showed a positive difference in relation to **[CSC]**, have a ▲ in the second position of their respective cells. On the other hand, **[BD]** has a ≡, as it did not present a difference in relation to **[CSC]**.

Finally, some results were highlighted in Table 10. Cells in which there is a positive statistical difference for both criteria were highlighted in dark green (▲ and ▲). Cells in which there is a positive statistical difference in at least one of the criteria, as long as the other does not present a difference, were highlighted in green ((≡ and ▲) and (▲ and ≡)). Both colors indicate favorable results for the $AC.Rank_A$ ranking in relation to **[CSC]**, as they improve both criteria or just one, as long as the other is maintained, ensuring a balance between the two.

On the other hand, cells in which there is a negative statistical difference for both criteria were highlighted in dark red (▼ and ▼).[8] Cells in which there is a negative statistical difference in at least one of the criteria, as long as the other does not present a difference, were highlighted in red ((≡ and ▼) and (▼ and ≡)). Both colors indicate unfavorable results for the $AC.Rank_A$ ranking in relation to **[CSC]**, since they worsen both criteria or just one, as long as the other remains the same. The remaining cases, which represent inversely proportional behaviors, were highlighted in light blue ((▲ and ▼), (▼ and ▲)). Note, in relation to the previously used example (first line), that three instances of $AC.Rank_A$ are highlighted in green in relation to **[CSC]**.

Considering the above, analyzing Table 10, it is possible to note that:

- Regarding the set of OMs used to instantiate $AC.Rank_A$, it is noted that the groups **[TW]**, **[GF]** and **[G2]** were the ones that obtained the most satisfactory results (green cells), but with differences between them:
  - The **[TW]** group presents good results in 45.24% (38 of 84 cells) of the configurations (green cells). However, the group presents the same rate when the evaluation criteria express an inverse relationship, i.e., 45.24% (38) of blue cells, with the models having a smaller F1-Macro (▼) and a better size (▲) compared to **[CSC]**. Therefore, on average, it does not present itself as a good option, since, in most cases, a result at least equal to that of **[CSC]** is not guaranteed;
  - The **[GF]** group presents good results in 47.62% (40) of the configurations (light green cells). Furthermore, in 3.57% (3) of the configurations it presents better results in relation to both evaluated criteria (dark green cells). Finally, it maintains the same performance compared to **[CSC]** in 40.48% (34) of the configurations (yellow cells). Therefore, it presents itself as a good option, as it is possible to improve or maintain results more than 90% of the time;
  - The **[G2]** group is between **[GF]** and **[TW]**. It presents good results in 52.38% (44) of the configurations (green cells). However, although the group maintains the same performance in relation to **[CSC]** in 28.57% (24) of the configurations (yellow cells), it presents a rate of 14.29% (12) when it comes to the inverse relationship between the criteria (blue cells). However, these cases only occur in flows where **[DN]** pruning is used. Thus, disregarding the flows related to **[DN]** (28), it presents results similar to **[GF]**: 50% of green cells (28/56) and 42.86% of yellow cells (24/56).

  Considering the above, the **[GF]** group is the most suitable for instantiating $AC.Rank_A$. It is worth mentioning that the groups **[C1]** and **[G1]** did not obtain good results: **[C1]** presents similar results to **[CSC]** (77.58% of yellow cells (65)), with some lower results (20.24% of light red cells (17)); **[G1]** presents, in more than half of the configurations, lower results than **[CSC]**.
- Regarding the aggregation methods used to instantiate $AC.Rank_A$, it is possible to note that:
  - When combined with the **[TW]** group, **[BL]**, **[BM]** and **[WS]** stand out, all of them covering 75% of the settings (9 out of 12 cells);
  - When combined with the **[GF]** group, **[BL]** and **[TS]** stand out, both covering 91.67% of the settings (11/12);
  - When combined with the **[C1]** and **[G1]** groups, **[BD]** stands out, covering, respectively, 16.67% (2/12) and 25% (3/12) of the settings;
  - When combined with the **[G2]** group, **[BL]** stands out covering 83.33% of the settings (10/12).

  Considering the above, the method that stands out the most is **[BL]**.

Considering the discussion presented above, it is therefore clear that the best instance of $AC.Rank_A$ is the one that combines **[GF]** + **[BL]** and should, therefore, be the one used. However, it is worth mentioning that, in relation to the **[GF]** group, the **[TS]** method also stands out, also presenting itself as a viable alternative (the difference in relation to **[BL]** occurs in just one configuration (**[DN]** + **[O]**)). Both are stable when used together with the **[GF]** group. Nevertheless, **[BL]** was also stable in other sets of OMs.

C. COMPLEMENTARY ANALYSIS

As mentioned, the work most related to the one presented here is that of [22] (MoMAC, Section III-D), since it was designed to simultaneously optimize performance and interpretability. Therefore, this section aims to discuss some aspects related to it when compared to the results presented here.

Initially the idea was to run MoMAC on the same datasets used here and include it in the comparative analysis. Therefore, we downloaded its implementation from GitHub[9] and tried to obtain the same results reported in the paper. However, the output of the method, as it is based on a genetic algorithm, is a collection of individuals, making it necessary to select a specific solution (individual). To do so, a graph is presented to the user, so that he or she can select the desired solution. The

[8] There were no configurations in this condition.

[9] https://github.com/banhdzui/MoMAC-v1

graph presents a set of candidate solutions, all of them aiming to optimize Accuracy and $\mathcal{L}$. In this way, while the x axis of the graph indicates the error (1)-Accuracy) obtained by each of the solutions, the y axis indicates the size of the models ($\mathcal{L}$). The idea is to analyze the graph and choose the individual that produces the best balance between error and $\mathcal{L}$. Therefore, it was not possible to reproduce the results, as the solution they chose to carry out the analyzes and statistical tests is not known. Furthermore, choosing the most appropriate solution to be used is not trivial in some of the datasets explored here. Thus, it was decided to hold a discussion based on the results presented in their paper.

Regarding performance, computed by them via F1-Micro, the authors compare MoMAC with several other algorithms, including CBA and CMAR. In relation to these algorithms MoMAC is superior; however, in relation to other algorithms it does not present a significant statistical difference (the authors used Friedman with Bonferroni-Dunn). However, it is worth mentioning that the average ranks obtained for the explored configurations are close and, depending on the solution (individual) chosen, the results of the statistical tests may change (small differences in values change the ranks).

Regarding model size, MoMAC presents much larger models (more rules) than CBA and other explored algorithms. Thus, even modeling the solution through an optimization problem, MoMAC presents an inverse relationship between performance and interpretability. Finally, it is worth mentioning that the user must specify the maximum error that the model can make, as well as the maximum number of rules that can be contained in the models. There are several choices that affect the results.

Considering the above, it can be said that $AC.Rank_A$ is the one that guarantees a better balance between performance and interpretability. Finally, another criterion not addressed in this work, included in the MoMAC reference, refers to the execution time of the methods. As it is a solution based on neural networks and genetic algorithm, it is, in general, much slower than almost all the algorithms used in the comparative analysis carried out by the authors, including CBA and CMAR (MoMAC just does not lose to one of the selected algorithms). On the other hand, the method proposed here, as it uses less computationally expensive strategies, ends up not being impacted in relation to this aspect. [23] present some experiments in relation to this aspect.

VI. CONCLUSION

This work presented a method for ranking rules via aggregation of OMs, named $AC.Rank_A$, to be incorporated into ACs induction flows, aiming to induce models that present a better balance between performance and interpretability. Previous studies had already used Pareto [20], ensemble of classifiers [21] and multi-objective optimization [22]. However, an inverse relationship between performance and interpretability was verified in them. Therefore, in this work we chose to use multi-criteria decision analysis (MCDA) and ranking aggregation.

For $AC.Rank_A$ to work, it must be instantiated with a set of OMs and an aggregation method. The sets of OMs explored were obtained or generated from works in the literature, namely: **[TW]**, **[GF]**, **[C1]**, **[G1]** and **[G2]**. The selected aggregation methods were **[BD]**, **[BM]**, **[BL]** and **[BG]**, referring to the ranking aggregation family, as well as **[TS]**, **[WS]** and **[WP]**, referring to the MCDA family.

The proposed method was evaluated in several ACs induction flows in terms of performance, measured via F1-Macro, and interpretability, measured via model size. Both criteria were estimated via stratified 10-fold cross-validation. The experiments were performed on 43 datasets. The ranking method used as baseline was the one used in CBA, i.e., **[CSC]**. The analysis was based on statistical tests, comparing the different flows when ranking takes place via **[CSC]** and via $AC.Rank_A$. Regarding the obtained results, it was noted that:

- The $AC.Rank_A$ method, in comparison to other works, is the one that guarantees a better balance between performance and interpretability;
- it was noticed, in several experiments, the inverse relationship between performance and interpretability, being possible to observe in different settings an improvement in the size of the model and a worsening in performance (and vice versa), such as the configurations related to the **[TW]** group;
- The suggested instantiation for $AC.Rank_A$ is the one that combines **[GF]+[BL]**. This combination guarantees a good balance between the criteria analyzed in relation to the **[CSC]** ranking, as it is possible to improve or maintain results more than 90% of the time. In other words, it is possible to improve interpretability without losing performance. It is assumed that the **[GF]** group performed better because it was extracted from sets with different distributions (IR) and that, in a certain way, it contributed to the ordering of the rules in the different classes;
- Among the aggregation methods, **[BL]**, **[BM]**, **[WS]**, **[TS]** and **[BD]** stand out depending on the set of OMs used; however, **[BL]** is what stands out the most.

Since $AC.Rank_A$ is proposed to be instantiated with a set of OMs and an aggregation method, with regard to future work it can be mentioned:

- the exploration of other ranking aggregation methods, such as those available in the FLAGR library [67];
- the exploration of other MCDA methods, as well as ways of assigning weights to measures (criteria), since, in this work, they all received the same weight;
- the investigation and/or proposition of other sets of OMs, since the impact of this parameter on the results was notable, being responsible for major changes in both the interpretability and performance of the model.

Thus, $AC.Rank_A$ can continue to be explored by the community, aiming not only to balance the two criteria, but to improve both.

TABLE 11. Equations of the OMs mentioned and/or used in this work. Adapted from [17]. The measure (ID) highlighted with a "*" was adjusted according to the definition in [68]. The measures, identified via IDs, are as follows: (1) Accuracy; (2) Added Value; (3) Chi-Square; (4) Collective Strength; (5) Complement Class Support; (6) Conditional Entropy (CON_EN); (7) Confidence; (8) Confidence Causal; (9) Confirm Causal; (10) Confirm Descriptive; (11) Confirmed Confidence Causal; (12) Correlation Coefficient; (13) Cosine; (14) Coverage; (15) Dir; (16) F-Measure; (17) Gini Index; (18) Goodman Kruskal; (19) Implication Index; (20) J-Measure; (21) Kappa; (22) Klosgen; (23) K-Measure; (24) Kulczynski 2; (25) Least Contradiction; (26) Leverage; (27) Lift; (28) Loevinger; (29) Logical Necessity; (30) Mutual Information; (31) Normalized Mutual Information; (32) Odd Multiplier; (33) Odds Ratio; (34) One Way Support; (35) Piatetsky Shapiro; (36) Prevalence; (37) Putative Causal Dependency; (38) Recall; (39) Relative Risk; (40) Specificity; (41) Support; (42) Theil Uncertainty Coefficiente; (43) TIC; (44) Two Way Support.

ID	Equation
1	$\mathrm{P(AB)} + \mathrm{P}(\overline{\mathrm{AB}})$
2	$\mathrm{P(B|A)} - \mathrm{P(B)}$
3	$\frac{(\mathrm{P(AB)} - \mathrm{P(A)P(B)})^2 \mathrm{N}}{\mathrm{P(A)P}(\overline{\mathrm{A}})\mathrm{P(B)P}(\overline{\mathrm{B}})}$
4	$\frac{\mathrm{P(AB)} + \mathrm{P}(\overline{\mathrm{AB}})}{\mathrm{P(A)P(B)} + \mathrm{P}(\overline{\mathrm{A}})\mathrm{P}(\overline{\mathrm{B}})} \frac{1 - \mathrm{P(A)P(B)} - \mathrm{P}(\overline{\mathrm{A}})\mathrm{P}(\overline{\mathrm{B}})}{1 - \mathrm{P(AB)} - \mathrm{P}(\overline{\mathrm{AB}})}$
5	$\frac{\mathrm{P}(\mathrm{A}\overline{\mathrm{B}})}{\mathrm{P}(\overline{\mathrm{B}})}$
6	$-\mathrm{P(B|A)} \log_2 \mathrm{P(B|A)} - \mathrm{P}(\overline{\mathrm{B}}|\mathrm{A}) \log_2 \mathrm{P}(\overline{\mathrm{B}}|\mathrm{A})$
7	$\mathrm{P(B|A)}$
8	$\frac{1}{2}(\mathrm{P(B|A)} + \mathrm{P}(\overline{\mathrm{A}}|\overline{\mathrm{B}}))$
9	$\mathrm{P(AB)} + \mathrm{P}(\overline{\mathrm{AB}}) - 2\mathrm{P}(\mathrm{A}\overline{\mathrm{B}})$
10	$\mathrm{P(AB)} - \mathrm{P}(\mathrm{A}\overline{\mathrm{B}})$
11	$\frac{1}{2}(\mathrm{P(B|A)} + \mathrm{P}(\overline{\mathrm{A}}|\overline{\mathrm{B}})) - \mathrm{P}(\overline{\mathrm{B}}|\mathrm{A})$
12	$\frac{\mathrm{P(AB)} - \mathrm{P(A)P(B)}}{\sqrt{\mathrm{P(A)P(B)P}(\overline{\mathrm{A}})\mathrm{P}(\overline{\mathrm{B}})}}$
13	$\frac{\mathrm{P(AB)}}{\sqrt{\mathrm{P(A)P(B)}}}$
14	$\mathrm{P(A)}$
15	$\begin{cases} -\infty & \text{if } \mathrm{P(B)} = 1 \\ 0 & \text{if } \mathrm{P(B)} \leq \frac{1}{2} \;\&\; \mathrm{P(B|A)} \leq \frac{1}{2} \\ 1 - \mathrm{CON\_ENT} & \text{if } \mathrm{P(B)} \leq \frac{1}{2} \;\&\; \mathrm{P(B|A)} > \frac{1}{2} \\ 1 - \frac{1}{\mathrm{CON\_ENT}} & \text{if } \mathrm{P(B)} > \frac{1}{2} \;\&\; \mathrm{P(B|A)} \leq \frac{1}{2} \\ 1 + \frac{\mathrm{CON\_ENT}}{\log_2 \mathrm{P(B)}^{\mathrm{P(B)}} \mathrm{P}(\overline{\mathrm{B}})^{\mathrm{P}(\overline{\mathrm{B}})}} & \text{if } \mathrm{P(B)} > \frac{1}{2} \;\&\; \mathrm{P(B|A)} > \frac{1}{2} \end{cases}$
16	$\frac{2\mathrm{P(A|B)P(B|A)}}{\mathrm{P(A|B)} + \mathrm{P(B|A)}}$
17	$\mathrm{P(A)}\left(\mathrm{P(B|A)}^2 + \mathrm{P}(\overline{\mathrm{B}}|\mathrm{A})^2\right) + \mathrm{P}(\overline{\mathrm{A}})\left(\mathrm{P(B|}\overline{\mathrm{A}})^2 + \mathrm{P}(\overline{\mathrm{B}}|\overline{\mathrm{A}})^2\right) - \mathrm{P(B)}^2 - \mathrm{P}(\overline{\mathrm{B}})^2$
18	$\frac{\max\left(\mathrm{P(AB)}, \mathrm{P}(\mathrm{A}\overline{\mathrm{B}})\right) + \max\left(\mathrm{P}(\overline{\mathrm{AB}}), \mathrm{P}(\overline{\mathrm{AB}})\right)}{2 - \max(\mathrm{P(A)}, \mathrm{P}(\overline{\mathrm{A}})) - \max(\mathrm{P(B)}, \mathrm{P}(\overline{\mathrm{B}}))} + \frac{\max\left(\mathrm{P(AB)}, \mathrm{P}(\overline{\mathrm{AB}})\right) \max\left(\mathrm{P}(\mathrm{A}\overline{\mathrm{B}}), \mathrm{P}(\overline{\mathrm{AB}})\right) -}{2 - \max(\mathrm{P(A)}, \mathrm{P}(\overline{\mathrm{A}})) - \max(\mathrm{P(B)}, \mathrm{P}(\overline{\mathrm{B}}))} + \frac{\max(\mathrm{P(A)}, \mathrm{P}(\overline{\mathrm{A}})) - \max(\mathrm{P(B)}, \mathrm{P}(\overline{\mathrm{B}}))}{2 - \max(\mathrm{P(A)}, \mathrm{P}(\overline{\mathrm{A}})) - \max(\mathrm{P(B)}, \mathrm{P}(\overline{\mathrm{B}}))}$
19	$\sqrt{\mathrm{N}} \frac{\mathrm{P}(\mathrm{A}\overline{\mathrm{B}}) - \mathrm{P(A)P}(\overline{\mathrm{B}})}{\sqrt{\mathrm{P(A)P}(\overline{\mathrm{B}})}}$
20	$\mathrm{P(AB)} \log_2 \frac{\mathrm{P(B|A)}}{\mathrm{P(B)}} + \mathrm{P}(\mathrm{A}\overline{\mathrm{B}}) \log_2 \frac{\mathrm{P}(\overline{\mathrm{B}}|\mathrm{A})}{\mathrm{P}(\overline{\mathrm{B}})}$
21	$\frac{\mathrm{P(B|A)P(A)} + \mathrm{P}(\overline{\mathrm{B}}|\overline{\mathrm{A}})\mathrm{P}(\overline{\mathrm{A}}) - \mathrm{P(A)P(B)} - \mathrm{P}(\overline{\mathrm{A}})\mathrm{P}(\overline{\mathrm{B}})}{1 - \mathrm{P(A)P(B)} - \mathrm{P}(\overline{\mathrm{A}})\mathrm{P}(\overline{\mathrm{B}})}$
22	$\sqrt{\mathrm{P(A)}}(\mathrm{P(B|A)} - \mathrm{P(B)})$
23	$\mathrm{P(B|A)} \log_2 \frac{\mathrm{P(B|A)}}{\mathrm{P(B)}} + \mathrm{P}(\overline{\mathrm{B}}|\overline{\mathrm{A}}) \log_2 \frac{\mathrm{P}(\overline{\mathrm{B}}|\overline{\mathrm{A}})}{\mathrm{P}(\overline{\mathrm{B}})} - \mathrm{P(B|A)} \log_2 \frac{\mathrm{P(B|A)}}{\mathrm{P}(\overline{\mathrm{B}})} - \mathrm{P}(\overline{\mathrm{B}}|\overline{\mathrm{A}}) \log_2 \frac{\mathrm{P}(\overline{\mathrm{B}}|\overline{\mathrm{A}})}{\mathrm{P(B)}}$
24	$\frac{\mathrm{P(AB)}}{\mathrm{P}(\mathrm{A}\overline{\mathrm{B}}) + \mathrm{P}(\overline{\mathrm{AB}})}$
25	$\frac{\mathrm{P(AB)} - \mathrm{P}(\mathrm{A}\overline{\mathrm{B}})}{\mathrm{P(B)}}$
26	$\mathrm{P(B|A)} - \mathrm{P(A)P(B)}$
27	$\frac{\mathrm{P(B|A)}}{\mathrm{P(B)}}$
28	$1 - \frac{\mathrm{P}(\mathrm{A}\overline{\mathrm{B}})}{\mathrm{P(A)P}(\overline{\mathrm{B}})}$
29	$\frac{\mathrm{P}(\overline{\mathrm{A}}|\mathrm{B})}{\mathrm{P}(\overline{\mathrm{A}}|\overline{\mathrm{B}})}$
30	$\mathrm{P(AB)} \log_2 \frac{\mathrm{P(AB)}}{\mathrm{P(A)P(B)}} + \mathrm{P}(\mathrm{A}\overline{\mathrm{B}}) \log_2 \frac{\mathrm{P}(\mathrm{A}\overline{\mathrm{B}})}{\mathrm{P(A)P}(\overline{\mathrm{B}})} + \mathrm{P}(\overline{\mathrm{AB}}) \log_2 \frac{\mathrm{P}(\overline{\mathrm{AB}})}{\mathrm{P}(\overline{\mathrm{A}})\mathrm{P(B)}} + \mathrm{P}(\overline{\mathrm{AB}}) \log_2 \frac{\mathrm{P}(\overline{\mathrm{AB}})}{\mathrm{P}(\overline{\mathrm{A}})\mathrm{P}(\overline{\mathrm{B}})}$
31	$\frac{\mathrm{MUINF}}{-\mathrm{P(A)} \log_2 \mathrm{P(A)} - \mathrm{P}(\overline{\mathrm{A}}) \log_2 \mathrm{P}(\overline{\mathrm{A}})}$
32	$\frac{\mathrm{P(AB)P}(\overline{\mathrm{B}})}{\mathrm{P(B)P}(\mathrm{A}\overline{\mathrm{B}})}$
33	$\frac{\mathrm{P(AB)P}(\overline{\mathrm{AB}})}{\mathrm{P}(\mathrm{A}\overline{\mathrm{B}})\mathrm{P}(\overline{\mathrm{AB}})}$
34	$\mathrm{P(B|A)} \log_2 \frac{\mathrm{P(B|A)}}{\mathrm{P(B)}}$
35	$\mathrm{N}(\mathrm{P(AB)} - \mathrm{P(A)P(B)})$
36	$\mathrm{P(B)}$
37	$\frac{1}{2}(\mathrm{P(B|A)} - \mathrm{P(B)}) + (\mathrm{P}(\overline{\mathrm{A}}|\overline{\mathrm{B}}) - \mathrm{P}(\overline{\mathrm{A}})) - (\mathrm{P}(\overline{\mathrm{B}}|\mathrm{A}) - \mathrm{P}(\overline{\mathrm{B}})) - (\mathrm{P(A|}\overline{\mathrm{B}}) - \mathrm{P(A)})$
38	$\mathrm{P(A|B)}$
39	$\frac{\mathrm{P(B|A)}}{\mathrm{P(B|}\overline{\mathrm{A}})}$
40	$\mathrm{P}(\overline{\mathrm{B}}|\overline{\mathrm{A}})$
41	$\mathrm{P(AB)}$
42	$\frac{\mathrm{MUINF}}{-\mathrm{P(B)} \log_2 \mathrm{P(B)} - \mathrm{P}(\overline{\mathrm{B}}) \log_2 \mathrm{P}(\overline{\mathrm{B}})}$
43*	$\sqrt{\mathrm{DIR}(\mathrm{A} \Rightarrow \mathrm{B})\,\mathrm{DIR}(\overline{\mathrm{B}} \Rightarrow \overline{\mathrm{A}})}$
44	$\mathrm{P(AB)} \log_2 \frac{\mathrm{P(B|A)}}{\mathrm{P(B)}}$

APPENDIX A
ABBREVIATIONS

The most used acronyms in this work are presented below:

$\mathcal{L}$	Model Size.
AC(s)	Associative Classifier(s).
AP	Apriori.
AR(s)	Association Rule(s).
BD	Borda Median.
BG	Borda Geometric Mean.
BL	Borda 2-norm.
BM	Borda Arithmetic Mean.
C1	Set of OMs C1.
CAR(s)	Classification Association Rule(s).
CBA	Classification Based on Association Rules.
CD	Critical Difference Diagram.
Conf	Confidence.
CSC	Confidence, Support, Cardinality.
CV	Coverage Pruning.
DN	Dynamic Pruning.
G1	Set of OMs G1.
G2	Set of OMs G2.
GF	Set of OMs GF.
IR	Imbalanced Ratio.
M1	M1 Pruning.
MCDA	Multi-criteria Decision Analysis.
O	Ordinal Prediction.
OM(s)	Objective Measure(s).
P	Probabilistic Prediction.
R	Rank-based Prediction.
Sup	Support.
TS	Topsis.
TW	Set of OMs TW.
V	Voting Prediction.
WP	WPM.
WS	WSM.

APPENDIX B
DEFINITION OF OBJECTIVE MEASURES

This appendix presents in Table 11 the equations of the OMs mentioned and/or used in this work.

REFERENCES

[1] M. S. Basha, "Early prediction of cardio vascular disease by performing associative classification on medical datasets and using genetic algorithm," in *Intelligent Computing and Innovation on Data Science*. Singapore: Springer, 2021, pp. 393–402.

[2] O. Haas, A. Maier, and E. Rothgang, "Rule-based models for risk estimation and analysis of in-hospital mortality in emergency and critical care," *Frontiers Med.*, vol. 8, Nov. 2021, Art. no. 785711.

[3] D. Sen, S. Paladhi, J. Frnda, S. Chatterjee, S. Banerjee, and J. Nedoma, "Associative classifier coupled with unsupervised feature reduction for dengue fever classification using gene expression data," *IEEE Access*, vol. 10, pp. 88340–88353, 2022.

[4] R. M. A. Mohammad, "An improved multi-class classification algorithm based on association classification approach and its application to spam emails," *IAENG Int. J. Comput. Sci.*, vol. 47, no. 2, pp. 187–198, 2020.

[5] B. S. Kumar and V. Ravi, "A hybrid approach using topic modeling and class-association rule mining for text classification: The case of malware detection," in *Proc. IEEE 17th Int. Conf. Cognit. Informat. Cognit. Comput. (ICCI CC)*, Jul. 2018, pp. 261–268.

[6] A. Al-Hawari, H. Najadat, and R. Shatnawi, "Classification of application reviews into software maintenance tasks using data mining techniques," *Softw. Qual. J.*, vol. 29, no. 3, pp. 667–703, Sep. 2021.

[7] Y. Shao, B. Liu, S. Wang, and G. Li, "Software defect prediction based on correlation weighted class association rule mining," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105742.

[8] F. Padillo, J. M. Luna, and S. Ventura, "LAC: Library for associative classification," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105432.

[9] P. K. Shukla and S. P. Tripathi, "A review on the interpretability-accuracy trade-off in evolutionary multi-objective fuzzy systems (EMOFS)," *Information*, vol. 3, no. 3, pp. 256–277, Jul. 2012.

[10] T. Mori and N. Uchihira, "Balancing the trade-off between accuracy and interpretability in software defect prediction," *Empirical Softw. Eng.*, vol. 24, no. 2, pp. 779–825, Apr. 2019.

[11] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019.

[12] M. Hahsler, I. Johnson, T. Kliegr, and J. Kuchař, "Associative classification in R: Arc, arulesCBA, and rCBA," *R J.*, vol. 11, no. 2, pp. 254–267, 2019.

[13] M. Azmi and A. Berrado, "RCAR framework: Building a regularized class association rules model in a categorical data space," in *Proc. 13th Int. Conf. Intell. Syst., Theories Appl.* New York, NY, USA: Association for Computing Machinery, Sep. 2020, pp. 1:1–1:16.

[14] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proc. 4th Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1998, pp. 27–31.

[15] J. Filip and T. Kliegr, "Classification based on associations (CBA)—A performance analysis," in *Proc. Doctoral Consortium Challenge @ RuleML+RR Hosted 2nd Int. Joint Conf. Rules Reasoning (RuleML+RR)*, vol. 2204, Luxembourg, Sep. 2018, p. 9.

[16] K. Bong, M. Joest, C. Quix, T. Anwar, and S. Manickam, "Selection and aggregation of interestingness measures: A review," *J. Theor. Appl. Inf. Technol.*, vol. 59, pp. 146–166, Jan. 2014.

[17] C. Tew, C. Giraud-Carrier, K. Tanner, and S. Burton, "Behavior-based clustering and analysis of interestingness measures for association rule mining," *Data Mining Knowl. Discovery*, vol. 28, no. 4, pp. 1004–1045, Jul. 2014.

[18] R. Somyanonthanakul and T. Theeramunkong, "Scenario-based analysis for discovering relations among interestingness measures," *Inf. Sci.*, vol. 590, pp. 346–385, Apr. 2022.

[19] R. Sharma, M. Kaushik, S. A. Peious, S. B. Yahia, and D. Draheim, "Expected vs. unexpected: Selecting right measures of interestingness," in *Proc. 22nd Int. Conf. Big Data Anal. Knowl. Discovery*, in Lecture Notes in Computer Science, vol. 12393, Bratislava, Slovakia. Cham, Switzerland: Springer, Sep. 2020, pp. 38–47.

[20] M. F. Da Silva and V. O. De Carvalho, "Aggregating interestingness measures in associative classifiers," in *Proc. Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, Oct. 2018, pp. 70–81.

[21] M. Dall'Agnol and V. O. de Carvalho, "Objective measures ensemble in associative classifiers," in *Proc. 22nd Int. Conf. Enterprise Inf. Syst.* Setúbal, Portugal: SciTePress, 2020, pp. 83–90.

[22] D. Bui-Thi, P. Meysman, and K. Laukens, "MoMAC: Multi-objective optimization to combine multiple association rules into an interpretable classification," *Appl. Intell.*, vol. 52, no. 3, pp. 3090–3102, Feb. 2022.

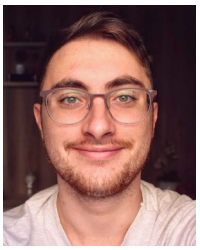
[23] M. Dall'Agnol and V. O. De Carvalho, "Ranking rules in associative classifiers via Borda's methods," in *Proc. 18th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2023, pp. 1–6.

[24] F. Z. El Mazouri, M. C. Abounaima, and K. Zenkouar, "A selection of useful patterns based on multi-criteria analysis approach," in *Proc. 2nd Int. Conf. Comput. Wireless Commun. Syst.*, Nov. 2017, pp. 1–6.

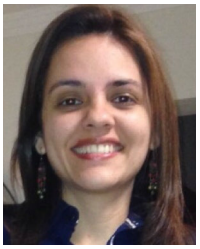
[25] A. Dahbi, S. Jabri, Y. Balouki, and T. Gadi, "The selection of the relevant association rules using the ELECTRE method with multiple criteria," *IAES Int. J. Artif. Intell.*, vol. 9, no. 4, p. 638, Dec. 2020.

[26] F. Z. El Mazouri, M. C. Abounaima, and K. Zenkouar, "Data mining combined to the multicriteria decision analysis for the improvement of road safety: Case of France," *J. Big Data*, vol. 6, no. 1, p. 5, Dec. 2019.

[27] M. Dall'Agnol, "Agrupamento e agregação de medidas objetivas para ranqueamento de regras em classificadores associativos," M.S. thesis, Departamento de Estatística, Matemática Aplicada e Computação, Universidade Estadual Paulista, São Paulo, Brazil, 2024.
[28] J. Mattiev and B. Kavsek, "Distance based clustering of class association rules to build a compact, accurate and descriptive classifier," *Comput. Sci. Inf. Syst.*, vol. 18, no. 3, pp. 791–811, 2021.
[29] J. Mattiev, M. Davityan, and B. Kavsek, "ACMKC: A compact associative classification model using K-modes clustering with rule representations by coverage," *Mathematics*, vol. 11, no. 18, p. 3978, Sep. 2023.
[30] J. Mattiev, C. Meza, and B. Kavsek, "The effect of 'directness' of the distance metric to produce compact and accurate associative classification models," *Appl. Sci.*, vol. 12, no. 18, p. 9055, 2022.
[31] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. 20th Int. Conf. Very Large Data Bases*. San Mateo, CA, USA: Morgan Kaufmann, 1994, pp. 487–499.
[32] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed. Pearson, 2019.
[33] J. M. Luna, M. Ondra, H. M. Fardoun, and S. Ventura, "Optimization of quality measures in association rule mining: An empirical study," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 1, pp. 59–78, 2018.
[34] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," *ACM Comput. Surv.*, vol. 38, no. 3, p. 9, Sep. 2006.
[35] W. Li, J. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in *Proc. IEEE Int. Conf. Data Mining*, 2001, pp. 369–376.
[36] X. Yin and J. Han, "CPAR: Classification based on predictive association rules," in *Proc. 3rd SIAM Int. Conf. Data Mining*, D. Barbará and C. Kamath, Eds. San Francisco, CA, USA: SIAM, May 2003, pp. 331–335.
[37] F. Thabtah, "A review of associative classification mining," *Knowl. Eng. Rev.*, vol. 22, no. 1, pp. 37–65, Mar. 2007.
[38] V. Margot and G. Luta, "A new method to compare the interpretability of rule-based algorithms," *AI*, vol. 2, no. 4, pp. 621–635, Nov. 2021.
[39] C. Molnar, *Interpretable Machine Learning*, 2nd ed. 2022.
[40] N. Sood and O. Zaiane, "Building a competitive associative classifier," 2020, *arXiv:2007.01972*.
[41] K. D. Rajab, "New associative classification method based on rule pruning for classification of datasets," *IEEE Access*, vol. 7, pp. 157783–157795, 2019.
[42] H. Lakkaraju, S. H. Bach, and J. Leskovec, "Interpretable decision sets: A joint framework for description and prediction," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1675–1684.
[43] A. Bechini, F. Marcelloni, and A. Segatori, "A MapReduce solution for associative classification of big data," *Inf. Sci.*, vol. 332, pp. 33–55, Mar. 2016.
[44] W. Cao, Q. Zhong, H. Li, and S. Liang, "A novel approach for associative classification based on information entropy of frequent attribute set," *IEEE Access*, vol. 8, pp. 140181–140193, 2020.
[45] T. T. N. Le, H. X. Huynh, and F. Guillet, "Finding the most interesting association rules by aggregating objective interestingness measures," in *Knowledge Acquisition: Approaches, Algorithms and Applications*. Springer, 2009, pp. 40–49.
[46] S. Bouker, R. Saidi, S. B. Yahia, and E. M. Nguifo, "Mining undominated association rules through interestingness measures," *Int. J. Artif. Intell. Tools*, vol. 23, no. 4, Aug. 2014, Art. no. 1460011.
[47] A. Dahbi, S. Jabri, Y. Balouki, and T. Gadi, "A new method for ranking association rules with multiple criteria based on dominance relation," in *Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2016, pp. 1–7.
[48] M. Rogers, M. Bruen, and L.-Y. Maystre, *ELECTRE and Decision Support: Methods and Applications in Engineering and Infrastructure Investment*. Boston, MA, USA: Springer, 2000.
[49] M. Azmi and A. Berrado, "Class-association rules pruning using regularization," in *Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2016, pp. 1–7.
[50] S. Sivanantham, V. Mohanraj, Y. Suresh, and J. Senthilkumar, "Rule precision index classifier: An associative classifier with a novel pruning measure for intrusion detection," *Pers. Ubiquitous Comput.*, vol. 27, no. 3, pp. 1395–1403, Jul. 2021.
[51] J. Mattiev and B. Kavšek, "A compact and understandable associative classifier based on overall coverage," *Proc. Comput. Sci.*, vol. 170, pp. 1161–1167, 2020.
[52] T. Kliegr and E. Izquierdo, "QCBA: Improving rule classifiers learned from quantitative data by recovering information lost by discretisation," 2017, *arXiv:1711.10166*.
[53] J. Li and O. R. Zaiane, "Exploiting statistically significant dependent rules for associative classification," *Intell. Data Anal.*, vol. 21, no. 5, pp. 1155–1172, Oct. 2017.
[54] S. Ayyat, J. Lu, and F. A. Thabtah, "Class strength prediction method for associative classification," in *Proc. 4th Int. Conf. Adv. Inf. Mining Manag.*, 2014, pp. 20–24.
[55] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[56] G. Yang and X. Cui, "A study of interestingness measures for associative classification on imbalanced data," in *Trends and Applications in Knowledge Discovery and Data Mining*. Springer, 2015, pp. 141–151.
[57] M. Dall'Agnol and V. O. De Carvalho, "Clustering the behavior of objective measures in associative classifiers," in *Proc. 18th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2023, pp. 1–6.
[58] B. Liu, Y. Ma, and C.-K. Wong, "Classification using association rules: Weaknesses and enhancements," in *Data Mining for Scientific and Engineering Applications*. Boston, MA, USA: Springer, 2001, pp. 591–605.
[59] J. Wątróbski, J. Jankowski, P. Ziemba, A. Karczmarczyk, and M. Zioło, "Generalised framework for multi-criteria method selection," *Omega*, vol. 86, pp. 107–124, Jul. 2019.
[60] C.-L. Hwang and K. Yoon, *Methods for Multiple Attribute Decision Making*. Berlin, Germany: Springer, 1981.
[61] E. Triantaphyllou, *Multi-Criteria Decision Making Methods: A Comparative Study*. Boston, MA, USA: Springer, 2000.
[62] C. Dwork, M. Naor, and D. Sivakumar, "Rank aggregation revisited," Tech. Rep., Mar. 2003.
[63] S. Lin, "Rank aggregation methods," *WIREs Comput. Statist.*, vol. 2, no. 5, pp. 555–570, Sep. 2010.
[64] J.-C. de Borda, "Mémoire sur les élections au scrutin," in *Histoire Del'Académie Royale Des Sciences*, 1781.
[65] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 1993, pp. 1022–1029.
[66] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.
[67] L. Akritidis, M. Alamaniotis, and P. Bozanis, "FLAGR: A flexible high-performance library for rank aggregation," *SoftwareX*, vol. 21, Feb. 2023, Art. no. 101319.
[68] J. Blanchard, F. Guillet, R. Gras, and H. Briand, "Mesurer la qualité des règles et de leurs contraposées avec le taux informationnel TIC," in *Proc. Extraction et Gestion des Connaissances (EGC)*, in Revue des Nouvelles Technologies de l'Information, Clermont-Ferrand, France. Cépaduès-Éditions, 2004, pp. 287–298.



MAICON DALL'AGNOL received the master's degree in computer science from Unesp, Brazil. He has published conference papers. His research interests include data mining and machine learning.



VERONICA OLIVEIRA DE CARVALHO received the Ph.D. degree in computer science from USP, Brazil. She is currently a Professor of computer science with Unesp. She has published papers at several international conferences. Her research interests include data mining and machine learning.