Pattern Recognition 124 (2022) 108493

Contents lists available at ScienceDirect

Pattern Recognition

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

An analysis of heuristic metrics for classifier ensemble pruning based on ordered aggregation

Amgad M. Mohammed [a,b,*], Enrique Onieva [a], Michał Woźniak [c], Gonzalo Martínez-Muñoz [d]

[a] Faculty of Engineering, University of Deusto, Bilbao, Spain
[b] Faculty of Computers and Information, Menoufia University, Menoufia, Egypt
[c] Faculty of Electronics, Wroclaw University of Science and Technology, Wroclaw, Poland
[d] Universidad Autónoma de Madrid, Cantoblanco, Madrid 28049, Spain

ARTICLE INFO

Article history:
Received 10 September 2020
Revised 3 June 2021
Accepted 4 December 2021
Available online 7 December 2021

Keywords:
Heuristic optimization
Ensemble selection
Ensemble pruning
Classifier ensemble
Machine learning
Difficult samples
Ordering-based pruning
Classifier complementariness

ABSTRACT

Classifier ensemble pruning is a strategy through which a subensemble can be identified via optimizing a predefined performance criterion. Choosing the optimum or suboptimum subensemble decreases the initial ensemble size and increases its predictive performance. In this article, a set of heuristic metrics will be analyzed to guide the pruning process. The analyzed metrics are based on modifying the order of the classifiers in the bagging algorithm, with selecting the first set in the queue. Some of these criteria include general accuracy, the complementarity of decisions, ensemble diversity, the margin of samples, minimum redundancy, discriminant classifiers, and margin hybrid diversity. The efficacy of those metrics is affected by the original ensemble size, the required subensemble size, the kind of individual classifiers, and the number of classes. While the efficiency is measured in terms of the computational cost and the memory space requirements. The performance of those metrics is assessed over fifteen binary and fifteen multiclass benchmark classification tasks, respectively. In addition, the behavior of those metrics against randomness is measured in terms of the distribution of their accuracy around the median. Results show that ordered aggregation is an efficient strategy to generate subensembles that improve both predictive performance as well as computational and memory complexities of the whole bagging ensemble.

1. Introduction

The use of multiple classifier systems (MCSs) [1,2] is an alternative to individual models that generally elevates the predictive accuracy of the classification tasks, by exploiting the strengths of individuals. Conceptually, a set of learning models that solve the same problem are consolidated to generate a better composite global model. While the limiting accuracy of an ensemble is achieved by generating a large pool size of classifiers [3]. *Clarifying*, as involving more classifiers in the classification task may provide more discriminating power with an equal or weighted contribution of each classifier to the final decision [4]. For that, the increasing size of the ensemble hardly copes with the increasing demand to speed up the decision and to save computational resources.

It has been reported that complementary members, *in decision space*, are desirable to produce robust systems [5]. This complementary behavior returns to the diversity of the formed ensemble with inductive biases. Bagging ensembles [6,7] achieve a reasonable diversity level by creating different bootstrap samples to train each base model independently. Moreover, the non-sensitivity of bagging and robustness under diverse noise conditions makes it more attractive [8]. To properly measure the diversity, the correct and incorrect predictions of individual members should be considered [9]. In bagging, an independent set of classifiers are generated in random order, the final decision is made by a simple majority voting rule. While, it has been demonstrated that reordering the generated pool and selecting the first subset of classifiers impacts the ensemble size and the composite accuracy positively [4,10–12]. The first subset of classifiers from the ordered list is expected to perform better than aggregating the whole list.

On the contrary, for sequential ensembles, such as *Boosting*, changing this order is not so much influential, as individual members are generated in sequential style by the learning algorithm

* Corresponding author at: Faculty of Engineering, University of Deusto, Bilbao, Spain.
E-mail addresses: amgad.elsayed@deusto.es (A.M. Mohammed), enrique.onieva@deusto.es (E. Onieva), michal.wozniak@pwr.edu.pl (M. Woźniak), gonzalo.martinez@uam.es (G. Martínez-Muñoz).

https://doi.org/10.1016/j.patcog.2021.108493

[10,13]. The sequential mechanism of boosting encourages the complementariness among ensemble members, by focusing on previously misclassified samples. However, in boosting the performance is more sensitive to noisy samples [8] and sometimes overfitting can be observed for large pool size [14]. The main challenge is to design a consolidated ensemble by incorporating ensemble selection techniques.

Ensemble Selection (ES) has been known in the literature as ensemble pruning [15], ensemble thinning [16] and ensemble reduction [17,18]. The selection can be considered as an intermediate process between building the ensemble and aggregating the decisions. Specifically, ES is the strategy of optimizing and selecting the number and the type of individual classifiers in-advance. Collecting the decisions from a reduced number of models speeds up the classification systems and relieves memory storage. In the literature, the selection process can be performed *offline* (static selection [19]) or *online* (dynamic selection [20]). In offline selection, the selected subset of classifiers is determined during the training stage of the system; on the other hand, the base classifiers in the online selection are selected on the fly based on the competence over the local region of the query sample. Without debate, dynamic selection techniques can outperform the static selection methods as experimentally been proved in [20] since the selection is optimized for each test sample independently. The rational in dynamic selection is that each member is an expert in a different local region inside the feature space. However, in the dynamic selection, there is a computational overhead for selecting the subensemble for each test sample. Besides, those techniques do not save memory space as all individual classifiers have to be retained in memory. Additionally, the dynamic selection is affected by the outlier instances around the query sample in the feature space [21]. Due to the mentioned pitfalls, the static selection strategies became an interesting topic in the area of MCSs.

The contribution of this paper can be highlighted in the following points:

1. Focusing on static ensemble selection as an active research topic in MCSs.
2. Analyzing the effectiveness of different heuristic metrics to reorder the randomly bagging ensembles.
3. Separate analysis of those metrics over binary and multiclass classification tasks.
4. As far as we know, this article is the first to group recent and efficient heuristic metrics for reordering bagging ensembles since they were analyzed by Martínez-Muñoz et al. [10] in 2009.

This work is organized as follows: In Section 2, we present the main notations to be used and review the related work. The heuristic metrics in detail are introduced in Section 3. While the experimental results with statistical analysis are presented in Section 4. The list of applications and sample areas are introduced in Section 5. Finally, the conclusions and future works are presented in Section 6.

2. Notations and related works

This section presents the main concepts of static ensemble selection and we highlight the importance of choosing a good heuristic for ordering the base classifiers of the ensemble. The mathematical notation used in this article is summarized in Table 1.

MCSs are composed of three stages: (1) Forming, (2) Selection, and (3) Aggregation [20]. The selection process is optional as it is not embedded in many ensemble systems. However, it has been proved that the generalization performance of a subensemble reports superior results over the traditional combination approaches, such as majority voting of the whole ensemble [10,22]. Pruning down the redundant models reduces the memory burden [17]. Furthermore, ES is a proven mechanism to enhance the efficiency and elevate the efficacy of classification ensemble systems [4,10,11,15]. However, it is not trivial to find the optimal subset of classifiers from a large ensemble as the complexity grows exponentially with the size of the pool. It is known that ES is a combinatorial search problem in which, from a pool size T, $2^T - 1$ nonempty subsets need to be evaluated in order to find the best subensemble [23–25].

Table 1
Mathematical notation to be used in this article.

Symbol	Meaning
ψ_k	The base classifier, $k = \{1, 2, \ldots, T\}$.
T	The ensemble size.
$\hat{T}$	The subensemble size.
D_{pr}	The pruning set, $D_{pr} = \{(\boldsymbol{x_i}, y_i), i = 1, 2, \ldots, N\}$.
N	The size of the pruning set.
$\boldsymbol{x_i}$	Attribute vector values, $\boldsymbol{x_i} = \left[x^{(1)}, \ldots, x^{(d)}\right]$.
y_i	The true class label, $y_i \in \{c_1, c_2, \ldots, c_M\}$.
M	Number of classes.
S_u	The subset of selected classifiers at iteration u.
L_u	The left subset from T at iteration u.
v_{c_M}	The number of votes for specific class.
P	The selection percentage of the subensemble.

To handle this complexity, several attempts ranging from optimized search [17,26], clustering techniques [27–29], and greedy algorithms [10,13,30,31] have been applied over decades. For *optimization-based pruning*, meta-heuristic techniques as genetic algorithm with expensive computational cost has been applied to find near-optimal solution [26]. The optimization function can be the diversity metric [32,33] to return diverse subset, or accuracy metric [34] to reduce the general ensemble error of a particular combination method. In addition, the idea of feature selection has been extended to the domain of ensemble reduction by transforming classifier predictions into artificial features to be reduced and selected by the harmony search algorithm [17]. Recently, the firefly algorithm has been modified to adapt to the area of ensemble selection by focusing on discriminant base classifiers [18]. *Clustering-based pruning* uses clustering techniques as non-supervised methods to group similar classifiers together. A representative classifier from each of the formed cluster is selected to obtain a highly diversified ensemble. This methodology may suffer from cluster instability as mentioned in [27], whereas an alternative solution is to use hybrid clustering techniques [28] to aggregate different clustering results (*consensus clustering*).

Greedy algorithms and heuristic metrics have been proved to be convenient techniques that return near-optimal subsets in fast time. These algorithms start with an empty (or full) initial ensemble and explore different subsets by iteratively expanding (or shrinking) the initial ensemble by an individual classifier. Those techniques comprise dissimilar heuristic measures as: ensemble diversity [15], ensemble margin [11,35], margin hybrid diversity [12], discriminating classifiers [4], ensemble error [13], complementariness of misclassification [35], and relative accuracy with minimum redundancy [4]. In [36], since no widely accepted definition to measure the ensemble diversity exists, five pairwise diversity measures are combined to obtain efficient pruned ensembles. In the literature, those techniques are popular and well known under the name of ordering-based ensemble pruning with the following merits:

- The ordering strategies return subensembles that are close to optimal solution (*Efficacy*) [10].
- Pruning strategies based on base classifier reordering can be easily adjusted to adapt to any given storage and computational restrictions (*Flexibility*) [4,12].

Table 2
Heuristic metrics to guide the ordered bagging ensembles.

Name	Heuristic Measure	Year	Ref.
RE	Reduced Error	1997	[13]
CC	Complementariness of Misclassification	2004	[35]
MDSQ	Supervised Ensemble Margin	2009	[10]
EPIC	Diversity Contribution of Individuals	2010	[15]
UMEP	Unsupervised Ensemble Margin	2013	[11]
MDEP	Margin & Diversity	2018	[12]
MRMR	Max. Relevance & Min. Redundancy	2018	[4]
DISC	Discriminant Classifiers	2018	[4]

- The time complexity of those strategies is low, in comparison with exhaustive or optimization-based search methods (*Efficiency*) [10].

Since the practical analysis of the power of greedy search methods in [10], many research efforts have been directed to propose new heuristic measures to guide the selection of subensemble [4,11,12,15]. Till now and related to our knowledge, no article has considered the analysis of all those promising metrics together. This research covers that gap by comparing all these new techniques with the best performing techniques found in [10] and against other popular baseline metrics [13,35]. Table 2 shows the heuristic metrics to be analyzed in this article over 30 datasets, divided into two parts: one composed of 15 binary datasets and other of 15 multiclass datasets.

3. Ordered aggregation

In bagging, the base classifiers are generated independently based on different bootstrap samples from the training data. The prediction accuracy of the ensemble is positively correlated with the number of aggregated models. Notwithstanding, the accuracy of the ensemble levels off after some point. After this point the inclusion of further models becomes useless. As shown in [10,35], the general accuracy (error) can be maximized (minimized) by changing the order in which the classifiers are aggregated. The authors proved that the first 20% from the modified ordered bagging ensemble was sufficient to speed up the classification decision, to save memory storage, and to get an improved composite prediction. The core component in the ordering strategies is the heuristic metric used to give the ordering process. That metric exploits the aggregation relationship between the classifiers based on maximizing (minimizing) specific measure as in greedy search [4,13,35], or rank the significance of each base classifier in one batch as in [11,12,15]. Those metrics require a selection/pruning set composed of labeled samples, D_{pr}, to validate and guide the ordering process. For that, the pruning set can be an independent part, not used for training, or can be sampled from the original training. Finally, the predictions of the selected classifiers are aggregated by unweighted voting as:

$$\hat{\Psi}(\boldsymbol{x_i}) = \arg\max_{y_i \in \mathcal{M}} \sum_{k=1}^{\hat{T}} [\psi_k(\boldsymbol{x_i}) = y_i] \tag{1}$$

where [ ] denotes Iverson's bracket and $\hat{T}$ represents the subensemble size.

In greedy search, the set of classifiers that are expected to perform better are aggregated first. Sequentially a new subset S_u is constructed from S_{u-1} by incorporating a single classifier from L_{u-1}; $S_u = S_{u-1} \cup \psi_k \mid \psi_k \in L_{u-1}$, where $T = S_u \cup L_u$ and $u = \{1, 2, 3, \ldots, T\}$. Such that the single classifier selection from L_{u-1} is guided upon a heuristic measure to optimize the augmented ensemble S_u. The number of iterations, $\hat{T}$, can be controlled in advance to meet the computational restrictions.

Furthermore, some metrics are proposed to rank all the classifiers in one batch without the sequential search. While, the properties of base classifiers, an individual's accuracy, are not effective to determine this rank [35]. The generated ensemble needs to consider the contribution of both the performance and the diversity of the individual learners to the ensemble [12,15]. It has been confirmed that the weakness of individuals can be compensated by the consensus of correct peers over different samples. Following those ideas, we discuss the heuristic metrics that are shown in Table 2 for reordering bagging ensembles.

3.1. Reduce-Error pruning

Reduce error pruning (RE) was firstly proposed in [13]. The classifier with the highest (lowest) accuracy (error), as estimated on the pruning set D_{pr}, is stored in S_1 as the initial subset to be extended. The sequential addition of more classifiers, one at a time, is performed to get as much (less) accuracy (error) as possible for the combined ensemble. This heuristic incorporates into the subensemble the classifier s_u as:

$$s_u = \arg\max_{k} \sum_{(\boldsymbol{x_i}, y_i) \in D_{pr}} \left[\hat{\Psi}_{S_{u-1} \cup \psi_k}(\boldsymbol{x_i}) = y_i\right] \tag{2}$$

where the index $k \in L_{u-1}$ and $S_u = S_{u-1} \cup \{s_u\}$. That metric has been applied in many articles as a baseline for comparison [4,11] with superior performance over the unpruned ensemble [10].

3.2. Complementariness measure

Complementariness measure (CC) was proposed in [35], and it considers the complementariness between the incorporated models. The first subset, S_1, is initialized by selecting the classifier with the highest accuracy on D_{pr}. Then, the classifier to be nominated is the one with the highest prediction accuracy over the set of instances that are misclassified by S_{u-1}:

$$s_u = \arg\max_{k} \sum_{(\boldsymbol{x_i}, y_i) \in D_{pr}} \left[\hat{\Psi}_{S_{u-1}}(\boldsymbol{x_i}) \neq y_i \wedge \psi_k(\boldsymbol{x_i}) = y_i\right] \tag{3}$$

where $k \in L_{u-1}$ and $S_u = S_{u-1} \cup \{s_u\}$. With this heuristic, the ensemble decision is expected to be shifted towards the correct classification. However, this metric concentrates only on the misclassified samples with no restriction to preserve the previous correct decisions.

3.3. Supervised ensemble margin

Margin distance minimization (MDSQ) is introduced in [10,35], where the decision space of the individual members over the selection set, D_{pr}, is transformed into signature vectors. The signature vector of classifier k, $r^{(k)}$, is defined by an N-dimensional vector whose ith component is calculated as:

$$r_i^{(k)} = 2[\psi_k(\boldsymbol{x_i}) = y_i] - 1, \quad (\boldsymbol{x_i}, y_i) \in D_{pr} \tag{4}$$

The ensemble signature vector, $\langle R \rangle$, is defined as the average sum of all $r^{(k)}$ as:

$$\langle R \rangle = T^{-1} \sum_{k=1}^{T} r^{(k)} \tag{5}$$

The subensemble whose average signature vector $\langle R \rangle$ is in the first quadrant, that is all the components are positive, correctly classifies all the examples in D_{pr}. The objective is to select a subensemble whose $\langle R \rangle$ is as close as possible to a reference vector, O, placed somewhere in the first quadrant. Hence, the reference vector is mathematically represented as:

$$O_i = q \quad with \quad i = \{1, 2, \ldots, N\} \quad and \quad 0 < q < 1 \tag{6}$$

The promoted classifiers are the ones with the minimum distance between their $\langle R \rangle$ and O, and can be selected sequentially by minimizing:

$$s_u = \arg\min_k \; d\left(O, T^{-1}\left(r^{(k)} + \sum_{t=1}^{u-1} r^{(t)}\right)\right) \tag{7}$$

where $k \in L_{u-1}$ and $d(\mathrm{O},\langle R \rangle)$ is the distance, usually Euclidean distance is used. The constant q should be sufficiently small, 0.075, to progressively focus on hard examples to be classified. Therefore, a subensemble with a large number of small positive values in $\langle R \rangle$ is preferred. By contrast, if the value of q is close to 1 the effectiveness of the method will be diminished as the selection will be guided upon the easy samples. In this article, the modified version of this metric is applied with a moving reference point $q^{(u)} = 2\sqrt{2u}/T$ as it was discussed in [10].

3.4. Maximum relevance & minimum redundancy

Maximum Relevance & Minimum Redundancy pruning (MRMR) was recently proposed in [4]. It is inspired by the popular algorithm mRMR [37,38] for reducing redundancy in the feature selection problem. The metric involves two relationships; one is between candidate class and the component class, and the other is between the candidate class and the target class. The candidate class represents the class label output of kth classifier to be included, while the component class represents the class label output of the composite ensemble. The classifier with the highest accuracy, estimated on the pruning set D_{pr}, is stored in S_1 as the initial subset to be extended. The next kth classifier to be incorporated, s_u, is selected according to:

$$s_u = \arg\max_k \left[I(\psi_k; Y) - \frac{1}{u-1} \sum_{\psi_i \in S_{u-1}} I(\psi_k; \psi_i) \right] \tag{8}$$

where $I(m, n)$ is the mutual information of variable m and n; Y is the target class; $k \in L_{u-1}$ and $S_u = S_{u-1} \cup \{s_u\}$. The classifier to be selected is the one with the maximum relevance with the target class, $I(\psi_k; Y)$, and simultaneously with minimum redundancy with S_{u-1}, $\frac{1}{u-1}\sum_{\psi_i \in S_{u-1}} I(\psi_k; \psi_i)$.

3.5. Discriminant classifiers

Discriminant classifiers pruning (DISC) has also been proposed in [4]. The good classifier to be incorporated is the one to compensate the current subensemble, S_{u-1}, taking into account the following two assumptions.

- *Assumption (1)*: Regarding the samples correctly classified by S_{u-1}, a good candidate is expected to do the same decisions on as many of such samples as possible.
- *Assumption (2)*: In relation to the samples misclassified by S_{u-1}, a good candidate is expected to classify correctly as many of those instances as possible.

The first assumption relates the candidate classifier and the composite ensemble, while the second assumption represents how the candidate classifier relates to the target. This metric concentrates on finding the most discriminant classifier, which is relative to both S_{u-1} and Y. The instances are divided into two parts; $\{mis\}$ represents the misclassified set by S_{u-1}, while $\{cor\}$ represents the set which is correctly classified by S_{u-1}. The incorporated classifier is selected as:

$$s_u = \arg\max_k \left[I({\psi_k}^{mis}; Y^{mis}) + \frac{1}{u-1} \sum_{\psi_i \in S_{u-1}} I({\psi_k}^{cor}; \psi_i^{cor}) \right] \tag{9}$$

where $k \in L_{u-1}$ and $S_u = S_{u-1} \cup \{s_u\}$. The first term $I({\psi_k}^{mis}; Y^{mis})$ is the mutual information that ψ_k can gain from the true labels Y according to the mislabeled instances by S_{u-1}. Whereas the second term $\frac{1}{u-1}\sum_{\psi_i \in S_{u-1}} I({\psi_k}^{cor}; \psi_i^{cor})$ is the average mutual information that ψ_k can gain from all ψ_i members of S_{u-1} related to the correct classified samples.

3.6. Diversity contribution of individuals

Ensemble Pruning via Individual Contributions (EPIC) is introduced in [15]. The classifier which is more diverse to the group, over the decision space, receives a higher rank to be in the ordered list. Each classifier divides the samples into four groups based on its decision as:

1. Correct, but ensemble prediction is incorrect.
2. Correct and ensemble prediction is correct.
3. Incorrect, but ensemble prediction is correct.
4. Incorrect and ensemble prediction is incorrect.

Samples in the group (1) are more critical as the individual classifier can contribute to change the ensemble decision. Samples in the group (3) are less harmful to the ensemble, for that the classifier contribution is low. The individual contribution of each classifier, IC_k, is measured $\forall \boldsymbol{x_i}$ based on the above groups and according to:

$$IC_k = \sum_{i=1}^{N} \left(\alpha_{ki} \; (2\nu_{\max}^{(i)} - \nu_{\psi_k(\boldsymbol{x_i})}^{(i)}) + \beta_{ki} \; \nu_{sec}^{(i)} + \theta_{ki} \; (\nu_{y_i}^{(i)} - \nu_{\psi_k(\boldsymbol{x_i})}^{(i)} - \nu_{\max}^{(i)}) \right) \tag{10}$$

Where:

$$\alpha_{ki} = \begin{cases} 1 & \text{if } \psi_k(\boldsymbol{x_i}) = y_i \; \wedge \psi_k(\boldsymbol{x_i}) \text{ is in the minority voting;} \\ 0 & \text{otherwise.} \end{cases}$$

$$\beta_{ki} = \begin{cases} 1 & \text{if } \psi_k(\boldsymbol{x_i}) = y_i \; \wedge \psi_k(\boldsymbol{x_i}) \text{ is in the majority voting ;} \\ 0 & \text{otherwise.} \end{cases}$$

$$\theta_{ki} = \begin{cases} 1 & \text{if } \psi_k(\boldsymbol{x_i}) \neq y_i; \\ 0 & \text{otherwise.} \end{cases}$$

Where $\nu_{\max}^{(i)}, \nu_{\mathrm{sec}}^{(i)}$ are the number of votes for the top two classes over sample $\boldsymbol{x_i}$. While $\nu_{\psi_k(\boldsymbol{x_i})}^{(i)}$ denotes the number of classifiers that agree with the prediction $\psi_k(\boldsymbol{x_i})$ (including itself) and $\nu_{y_i}^{(i)}$ denotes voting of the correct prediction over sample $\boldsymbol{x_i}$. The classifiers which have more samples in the minority group, group (1), bring more diversity contributions to the ensemble and contain more useful knowledge for constructing subensembles [15].

3.7. Unsupervised ensemble margin

Unsupervised Margin based Ensemble Pruning (UMEP) has been proposed in [11] with the focus on classifier properties to correctly classify the hard patterns. The innovation of this metric is based on measuring the margin of $\boldsymbol{x_i}$. The larger the margin of $\boldsymbol{x_i}$ the more certain its classification is. As in boosting [39], the idea of this method is to focus on low margin instances. The absolute margin of $\boldsymbol{x_i}$ can be measured from ensemble decisions as:

$$margin(\boldsymbol{x_i}) = \frac{(\nu_{\max} - \nu_{sec})}{\sum_{i=1}^{M}(\nu_{c_i})} \tag{11}$$

This measure considers only the difference between the votes of the top two classes $(\nu_{\max}, \nu_{\mathrm{sec}})$ over the sample $\boldsymbol{x_i}$. For that, it is an unsupervised measure that does not require the true class label. Here the margin takes a value in the interval [0,1]. The set of samples, $\boldsymbol{x_i} \in D_{pr}$, that are classified correctly by each classifier will

Table 3
Characteristics of the selected datasets for experimentation, *sorted by samples and classes*.

DataSet	#S	#F	#C	R	DataSet	#S	#F	#C	R
Breast-cancer	286	9	2	0.42	Wine	178	13	3	0.676
SPECTF	349	44	2	0.37	Newthyroid	215	5	3	0.2
Ionosphere	351	33	2	0.56	Cmc	1473	9	3	0.529
Wdbc	569	30	2	0.594	Lymphography	148	18	4	0.025
Indian Liver Patient (ILP)	583	10	2	0.401	Vehicle	846	18	4	0.913
Australian	690	14	2	0.802	Wall-Following-Robot (WFR)	5456	24	4	0.149
Wisconsin	699	9	2	0.526	Cleveland	297	13	5	0.081
Blood-transfusion	748	4	2	0.312	Dermatology	358	34	6	0.18
Mammographic	830	5	2	0.944	Flare	1066	11	6	0.13
Tic-tac-toe	958	9	2	0.530	Wine quality-red	1599	11	6	0.015
German	1000	20	2	0.429	Satimage	6435	36	6	0.408
Hill-valley	1212	100	2	1.0	Segment	2310	18	7	1
Kr-vs-kp	3196	36	2	0.915	Led7digit	500	7	10	0.649
spambase	4601	57	2	0.650	Mfeat-Karh	2000	64	10	1
Ringnorm	7400	20	2	0.981	Mfeat-Fourier	2000	76	10	1

Table 4
The average accuracy of the subensemble related to a selection percentage (P) from an inital pool size (T);*SPECTF dataset*.

T	Bagging	BSM	RE	CC	MDSQ	MRMR	DISC	EPIC	UMEP	MDEP	P	$\hat{T}$
25	**84.73**	81.87	86.73	86.21	87.23	86.43	**88.43**	86.49	86.65	86.77	20%	5
			87.66	87.11	88.18	87.19	**88.69**	87.77	87.74	87.71	30%	7
51	**85.27**	82.96	89.22	88.41	89.91	88.35	**90.41**	89.65	89.54	89.72	20%	11
			88.97	88.54	89.82	88.62	**90.49**	89.65	89.76	89.80	30%	15
75	**85.48**	82.92	88.89	88.80	90.24	88.12	**91.10**	90.28	90.83	90.66	20%	15
			88.80	88.94	89.69	88.20	**90.89**	90.38	90.32	90.21	30%	23
101	**85.85**	83.09	89.08	89.06	89.90	87.77	**91.21**	90.56	90.22	90.44	20%	21
			88.93	88.80	88.83	88.14	**91.01**	90.84	90.41	90.78	30%	31
151	**85.82**	82.78	89.18	88.89	89.29	88.54	**91.09**	90.58	90.59	90.44	20%	31
			88.82	88.97	88.03	88.91	**91.24**	90.64	90.78	90.70	30%	45
201	**85.84**	83.68	89.12	89.19	88.48	88.82	91.07	91.07	**91.15**	91.10	20%	41
			89.22	88.88	87.36	88.70	**91.49**	91.00	91.04	91.09	30%	61

be considered for calculating it's margin-based information quantity as:

$$UM_k(D_{pr}) = -\frac{1}{N}\sum_i \log\left(margin(\boldsymbol{x_i})\right), \quad \forall i \mid \psi_k(\boldsymbol{x_i}) = y_i \tag{12}$$

The classifiers are ranked based on descending the measured values from Eq. (12). The harder the samples correctly predicted by the classifier, the higher the rank it receives to be included in the subensemble.

3.8. Margin & diversity

Margin and Diversity based Ensemble Pruning (MDEP) [12] considers two aspects to better reorder the set of classifiers: (1) focusing on examples with small absolute margin, and (2) focusing on classifiers with large diversity contribution to the ensemble. The MDEP measures the rank of each classifier via Eq. (13).

$$MDEP(\psi_k) = \sum_i \left[\alpha f_m(\boldsymbol{x_i}) + (1-\alpha) f_d(\psi_k, \boldsymbol{x_i})\right], \quad \forall i \mid \psi_k(\boldsymbol{x_i}) = y_i \tag{13}$$

Where $\alpha \in [0,1]$ represents the balance of importance between the margin of examples and the ensemble diversity. $f_m(\boldsymbol{x_i})$ and $f_d(\psi_k, \boldsymbol{x_i})$ are the log functions of $\boldsymbol{x_i}$'s margin and ψ_k's diversity contribution on $\boldsymbol{x_i}$, respectively:

$$f_m(\boldsymbol{x_i}) = \log\left(\left|\frac{v_{y_i}^{(i)} - v_{\bar{y}_i}^{(i)}}{M}\right|\right) \tag{14}$$

$$f_d(\psi_k, \boldsymbol{x_i}) = \log\left(\frac{v_{y_i}^{(i)}}{M}\right) \tag{15}$$

Where $\bar{y}_i \neq y_i$ is the class that receives the maximum number of votes on $\boldsymbol{x_i}$. The challenge of the MDEP metric is the dependence on the predefined value of α that controls the trade-off between focusing on classifiers that correctly predict hard samples or focusing on classifiers that increase ensemble's diversity.

4. Experimental analysis

This section first introduces the experiment setting and the characteristics of the datasets used in this paper, and then, the experimental results are reported for the comparison among the investigated metrics. The following research questions were posed:

- Q_1. How the initial pool size and the required subensemble size affect on the performance of the heuristics?, Section 4.3.
- Q_2. How the heuristics are affected by the individual classifier type?, Section 4.4.
- Q_3. How the efficacy of the investigated metrics could be affected by binary and multiclass datasets?, Sections 4.5 and 4.6.
- Q_4. What will be the rank of the metrics for analyzing the thirty datasets?, Section 4.7.
- Q_5. How the investigated metrics are potential for prediction consistency?, Section 4.8.
- Q_6. How the efficiency of the metrics differs in terms of time and space complexities?, Section 4.9.

4.1. Set up

The design of experiments has considered the recommendations from Martínez-Muñoz et al. [10] according to the following two issues: (A) The influence of training conditions (B) The influence of the initial pool size. *For training conditions*; the whole

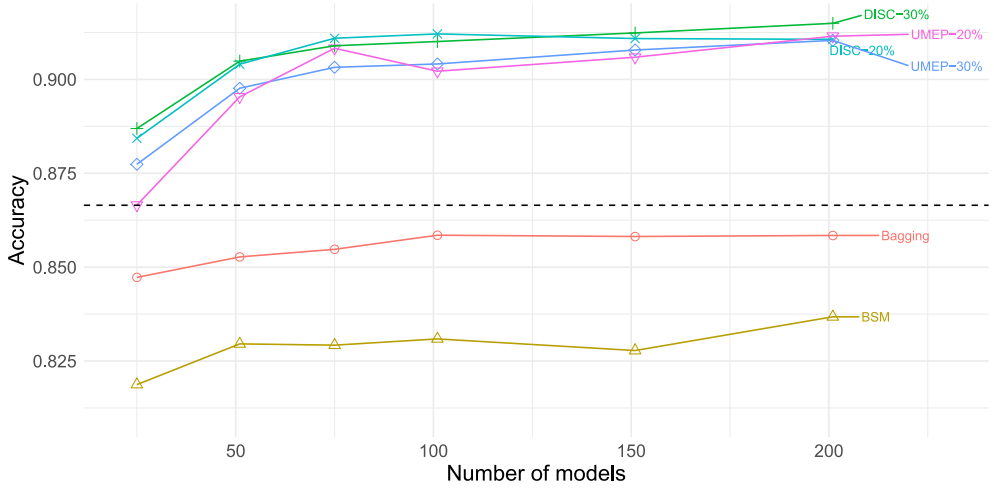

Fig. 1. Influence of pool size and selection size on the general accuracy; *SPECTF dataset*.

Table 5
Average accuracy and standard deviation for Different Classifiers (DC) and Similar Classifiers (SC); T= 101 and P=30%.

	DC				SC			
	Bagging	RE	DISC	UMEP	Bagging	RE	DISC	UMEP
Dataset	$T = 101$	$\hat{T} = 31$			$T = 101$	$\hat{T} = 31$		
Australian	86.70 ± 3.39	**86.97 ± 3.52**	**86.97 ± 3.48**	**87.36 ± 3.35**	86.85 ± 3.43	86.66 ± 3.85	86.34 ± 3.81	**86.95 ±3.67**
Blood	77.37 ± 1.77	77.35 ± 2.68	**77.53 ± 2.74**	77.03 ± 2.73	76.27 ± 0.73	**77.53 ± 2.77**	**77.92 ± 2.47**	**77.14 ± 2.80**
Breast-cancer	73.41 ± 5.73	**73.94 ±6.84**	**73.45 ± 6.03**	72.97 ± 6.90	73.48 ± 5.08	73.02 ± 5.63	73.10 ± 5.58	71.41 ± 6.39
Cmc	53.30 ± 3.74	53.29 ± 3.83	**53.80 ± 3.72**	**53.82 ± 3.69**	53.33 ± 3.84	**53.38 ± 3.41**	**53.57 ± 3.58**	**53.65 ± 3.61**
Mammographic	83.04 ± 4.02	82.72 ± 4.00	82.59 ± 4.35	**83.87 ± 4.02**	83.70 ± 3.41	82.99 ± 3.62	83.19 ± 3.56	83.64 ± 3.53
Wdbc	96.63 ± 2.42	**97.10 ± 2.32**	**97.44 ± 2.03**	**97.54 ± 1.95**	96.58 ± 2.23	96.48 ± 2.40	**96.76 ± 2.32**	**96.67 ± 2.27**
AR-Friedman	5.33	4.42	3.33	**3**	5	5.75	**4.33**	4.83

training data has been used, for both, to train the bagged ensemble and to prune it. *For the initial pool size*; the initial pool should contain a sufficiently large number of classifiers. However, the accuracy gains that could be obtained by expanding the initial pool are negligible after a point. Two ensemble systems have been constructed as a part of the analysis:

1. Heterogeneous (*Different Classifier types-DC*)
   - *Samples*: Bootstrap samples, with replacement, are generated from the training data.
   - *Features*: Sixty percent of features are selected randomly for each classifier.
   - *Classifiers*: Five different classifier models, with their default setup parameters, have been used (*Decision Tree (DT)*[1], *Naíve Bayes (NB)*[2], *rule-based learner (JRip)*[3], *Multinomial Log-linear Models (Multinom)*[4], and *k-Nearest Neighbor (KNN)*[5]) with 20% as a proportional representation by each model from the whole pool size.
2. Homogeneous (*Similar Classifiers-SC*)
   - Using the same configuration to the previous section except that all individual members are of type *Decision Tree*.

Finally, all the datasets are preprocessed by unifying the scales of the features via normalization in order for the features to have zero mean and 1 standard deviation. For each dataset, 10 repetitions of 10 fold cross-validation procedure have been tested to get 100 runs per dataset. In addition, MDEP depends on an internal parameter α; three values for MDEP with different $\alpha \in \{0.1, 0.5, 0.9\}$ are considered, and the best-optimized alpha according to the in train-validation is used to report the test for each dataset separately. The results for Random Forest (RF) [3], Adaboost (AdaB) [39], and the single best model (SBM) from the pool, *according to the measured accuracy of the models on the pruning set*, are included as references in the comparison.

The default set-up for the individual classifiers types, RF, and AdaB are as follows: DT uses C5.0 decision trees of Quinlan [40] in its pruned version. Naíve Bayes applies Laplace smoothing to solve the problem of zero probability. k-Nearest Neighbor is considered with $k = 3$ as the number of neighbors. RF[6] implements Breiman's random forest algorithm with ntrees=T, number of variables that are randomly sampled at each split=$sqrt(ncol(\mathbf{x}))$. AdaB[7] fits the AdaBoost.M1 using classification trees as base classifiers with iterations=T.

4.2. Datasets

A total of 30 datasets that were obtained from OpenML[8] and KEEL[9] are used in this study for experimentation. The characteristics of the datasets are presented in Table 3, where #S, #F, #C, and R represent the number of samples, the number of features, the number of classes and the ratio between the smallest to the largest class for each dataset, respectively. The number of classes varies from 2 to 10, while the maximum number of features is 100.

[1] Package C50: https://cran.r-project.org/web/packages/C50 Decisiont.
[2] Package e1071: https://cran.r-project.org/web/packages/e1071.
[3] Package RWeka: https://cran.r-project.org/web/packages/RWeka.
[4] Package nnet: http://cran.r-project.org/web/packages/nnet.
[5] Package caret: https://cran.r-project.org/web/packages/caret.
[6] randomForest: https://cran.r-project.org/web/packages/randomForest.
[7] Adaboost: https://cran.r-project.org/web/packages/adabag.
[8] Machine Learning Repository: https://www.openml.org.
[9] KEEL Repository: http://www.keel.es.

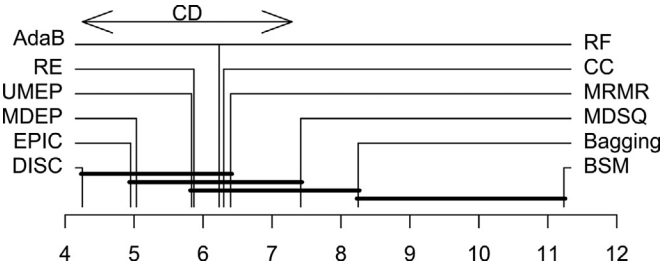

Fig. 2. Comparison of the different metrics over the 30 datasets using the Nemenyi test. Methods not significantly different (α=0.05) are connected together.

4.3. Influence of pool size and selection size

To answer Q_1, we analyze how ordered aggregation is affected by both the initial pool size (T) and the selection percentage (P) simultaneously. A heterogeneous ensemble, see Section (4.1), composed of 201 models is built. The classifiers will be ordered considering only the first 25, 51, 75, 101, 151, and 201 models from ensemble size. This means that the classifiers are nested such that all classifiers in an initial pool are also included in larger pools. The average accuracy over 100 runs is computed for each ensemble size T.

Table 4 shows the analysis of 1200 records ($T = 6 \times$ runs=100 $\times$ $P = 2$) for the *SPECTF* dataset. The last two columns represent the size of the ordered subensemble according to an initial pool of T. For each T and P, the best value is highlighted in bold. The prediction accuracy of the bagging increases monotonically as a function of the initial pool size, while this saturation level sometimes decreases according to the classification task. All the investigated metrics prove their superiority over bagging, *regarding SPECTF dataset*. The poor results reported for BSM, this confirms that the ensemble outperforms all single models of which it is composed of. Regarding the selection percentage of $P = 30\%$, the accuracy of DISC, EPIC, and UMEP keeps on increasing as more classifiers are included in the initial pool. *In general*, the accuracy of the ordered classifiers with a percentage of 30% is better than 20%. In addition, the poorest accuracy by any ordering metric is better than the highest accuracy of bagging. The highest accuracy of 91.49% is reported by DISC using a subensemble composed of 61 classifiers, $\hat{T}$, instead of the 201 classifiers used by bagging.

Fig. 1 shows the complementary perspective about the results. The comparison of DISC-30% with DISC-20% and the comparison of UMEP-30% with UMEP-20% confirm that the ordered subset with a larger number of classifiers returns higher accuracy. Furthermore, the lowest accuracy by UMEP-20%, *horizontal-dashed line*, with only 5 classifiers outperforms the accuracy of bagging for any size. The prediction accuracy of DISC-30% and UMEP-30% keeps on increasing even after bagging has stabilized. In addition, the figure shows that BSM is the worst ensemble selection strategy. We confirm that the main objective of the heuristic-based metrics is to return a subensemble with a reduced number of classifiers while keeping the accuracy unaffected. Going beyond the accuracy of bagging is conditioned by the effectiveness to reorder the ensemble as we will discuss in Sections 4.5 and 4.6.

4.4. Influence of heterogeneous classifiers

To answer Q_2, we analyze how the general accuracy and the performance of the metrics are affected by the type of combined individual models. For that, the initial pool size is fixed with $T =$ 101 as a balance between accuracy and ensemble's complexity. Then the two types of ensembles, *homogeneous and heterogeneous*, described in Section 4.1, are constructed for comparison purposes.

Table 6
Average accuracy and standard deviation over binary datasets for ensemble size $T = 101$ and P=30%. The values that outperform bagging are highlighted in bold.

		$T = 101$	$\hat{T} = 1$	$\hat{T} = 31$							$T = 101$	
#	Datatset	Bagging	BSM	RE	CC	MDSQ	MRMR	DISC	EPIC	UMEP	MDEP	AdaB	RF
D_1	Australian	86.70 ± 3.39	84.03 ± 4.13	**86.97 ± 3.52**	**86.87 ± 3.39**	86.47 ± 3.78	**86.75 ± 3.44**	**86.97 ± 3.48**	**87.29 ± 3.38**	**87.36 ± 3.35**	**87.38 ± 3.32**	**86.94 ± 3.70**	**86.97 ± 3.57**
D_2	Blood-transfusion	77.37 ± 1.77	73.69 ± 4.33	77.35 ± 2.68	**77.70 ± 2.85**	**77.65 ± 2.79**	**77.70 ± 2.80**	**77.53 ± 2.74**	77.34 ± 2.96	77.03 ± 2.73	77.36 ± 2.95	76.09 ± 3.85	75.31 ± 4.10
D_3	Breast-cancer	73.41 ± 5.73	68.64 ± 8.16	**73.94 ± 6.84**	**73.45 ± 5.98**	73.03 ± 4.99	**73.77 ± 6.10**	**73.45 ± 6.03**	73.28 ± 6.63	72.97 ± 6.90	73.32 ± 6.56	68.75 ± 7.62	71.50 ± 8.06
D_4	German	74.98 ± 2.90	69.26 ± 4.43	**75.65 ± 3.24**	**75.48 ± 3.10**	74.67 ± 2.87	**75.19 ± 3.22**	**75.77 ± 3.06**	**75.52 ± 3.24**	**75.77 ± 3.19**	**75.77 ± 3.22**	**75.39 ± 3.56**	**76.09 ± 3.30**
D_5	Hill-valley	67.81 ± 5.09	**86.00 ± 4.66**	**82.80 ± 6.54**	**80.10 ± 7.26**	**78.47 ± 5.23**	**80.04 ± 7.29**	**79.23 ± 5.32**	**74.82 ± 9.06**	**81.71 ± 4.12**	**81.79 ± 4.17**	59.06 ± 3.78	57.58 ± 3.74
D_6	ILP	70.33 ± 5.55	67.22 ± 5.13	69.99 ± 5.20	68.96 ± 5.75	70.28 ± 5.10	68.84 ± 5.76	70.30 ± 5.13	**71.44** ± 4.36	**71.29 ± 5.03**	**71.43 ± 4.59**	**71.59 ± 5.06**	**70.52 ± 5.30**
D_7	Ionosphere	93.37 ± 3.71	89.25 ± 5.54	**93.48 ± 3.81**	**93.54 ± 3.79**	**93.90 ± 3.11**	**93.60 ± 3.89**	**93.80 ± 3.75**	**93.60 ± 3.90**	**93.80 ± 3.89**	**93.85 ± 3.88**	**94.08 ± 3.30**	93.25 ± 4.05
D_8	Kr-vs-kp	96.25 ± 1.20	**96.46 ± 1.46**	**98.42 ± 0.83**	**98.25 ± 0.76**	**97.92 ± 0.88**	**98.20 ± 0.77**	**98.37 ± 0.68**	**96.76 ± 1.23**	**96.73 ± 1.23**	**96.76 ± 1.22**	**99.62 ± 0.33**	**98.67 ± 0.68**
D_9	Mammographic	83.04 ± 4.02	82.46 ± 4.27	82.72 ± 4.00	82.21 ± 4.36	82.35 ± 3.90	82.37 ± 4.38	82.59 ± 4.35	**83.94 ± 3.77**	**83.87 ± 4.02**	**84.00 ± 3.97**	80.73 ± 4.18	81.78 ± 4.09
D_{10}	Ringnorm	96.66 ± 0.60	93.72 ± 2.11	**96.77 ± 0.64**	**96.77 ± 0.59**	95.07 ± 0.69	**96.79 ± 0.60**	**96.85 ± 0.61**	**96.80 ± 0.66**	**96.79 ± 0.68**	**96.79 ± 0.68**	**97.33 ± 0.55**	94.98 ± 0.80
D_{11}	Spambase	94.98 ± 1.07	91.52 ± 1.44	**95.04 ± 1.00**	94.75 ± 1.04	94.72 ± 1.04	94.76 ± 1.01	**94.99 ± 1.15**	94.93 ± 1.06	94.85 ± 1.11	94.91 ± 1.08	**95.72 ± 0.87**	**95.31 ± 0.98**
D_{12}	SPECTF	85.96 ± 5.32	82.18 ± 5.98	**89.17 ± 4.78**	**89.20 ± 4.85**	**88.48 ± 5.50**	**88.64 ± 4.85**	**91.35 ± 4.71**	**90.66 ± 4.75**	**90.84 ± 4.74**	**90.92 ± 4.69**	**91.24 ± 3.96**	**92.39 ± 4.30**
D_{13}	Tic-tac-toe	76.57 ± 2.85	**77.49 ± 4.41**	**86.10 ± 3.14**	**86.18 ± 3.24**	**86.65 ± 3.24**	**86.23 ± 3.17**	**86.53 ± 3.18**	**87.08 ± 2.98**	**85.66 ± 3.22**	**87.06 ± 3.34**	**99.47 ± 0.85**	**98.72 ± 1.18**
D_{14}	Wdbc	96.63 ± 2.42	95.71 ± 2.55	**97.10 ± 2.32**	**97.10 ± 2.24**	**96.89 ± 2.10**	**97.17 ± 2.31**	**97.44 ± 2.03**	**97.42 ± 2.03**	**97.54 ± 1.95**	**97.54 ± 1.94**	**96.82 ± 2.13**	96.14 ± 2.37
D_{15}	Wisconsin	97.31 ± 2.01	95.20 ± 2.33	97.22 ± 1.95	97.22 ± 2.11	97.21 ± 2.15	97.19 ± 2.18	97.18 ± 2.15	97.27 ± 2.04	97.12 ± 1.95	97.22 ± 2.14	96.83 ± 2.13	97.12 ± 1.84
	AR-Friedman	8	10.8	**5.6**	**6.73**	**7.8**	**6.73**	**4.6**	**5.07**	**5.83**	**4**	**5.87**	**6.97**

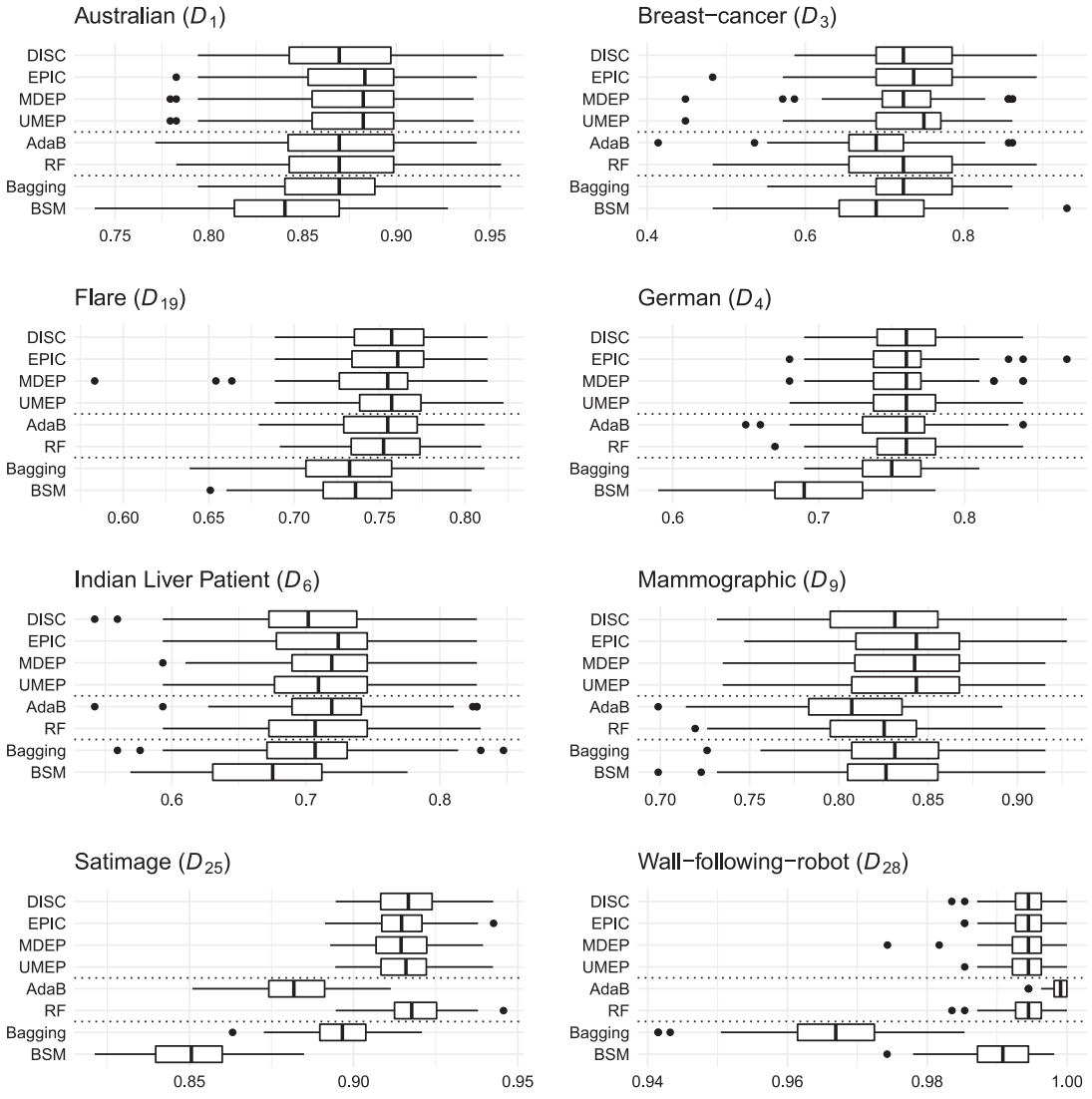

Fig. 3. Distribution of the prediction accuracy.

Table 7
Summary of the Wilcoxon test (for binary datasets). • = the method in the row improves the method of the column. ○ = the method in the column improves the method of the row. Upper diagonal of level significance $\alpha = 0.9$, Lower diagonal level of significance $\alpha = 0.95$.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
Bagging (1)	-	•	○	○		○	○	○	○	○		
BSM (2)	○	-	○	○	○	○	○	○	○	○	○	○
RE (3)	•	•	-	•	•	•						
CC (4)		•		-			○			○		
MDSQ (5)		•	○		-		○		○	○		
MRMR (6)		•	○			-	○			○		
DISC (7)	•	•		•	•		-					
EPIC (8)	•	•						-		○		
UMEP (9)	•	•							-	○		
MDEP (10)	•	•		•	•	•		•	•	-		
AdaB (11)		•									-	
RF (12)		•										-

Table 5 presents the average accuracy and the standard deviation of the ensembles using different and similar individual classifiers over six representative datasets. The results prove that the general accuracy of bagging can be outperformed in both cases by using pruned ensemble, *highlighted bold values in each part*. To differentiate among the methods, the average rank of Friedman test [41] *(AR-Friedman)* is calculated and is shown in the last row of Table 5. The diversity in decision space, which is achieved by different classifiers, guarantees effective performance for the ordering methods. The best ranks, upon the selected datasets, are reported for DC-UMEP, DC-DISC, and DC-RE in comparison with their versions under similar classifiers. We conclude that similar clas-

sifiers produce similar decisions approximately, and the ability to differentiate among them by ordering metrics decreases. For the rest of our experiments, an ensemble of different classifier types with fixed pool size, $T = 101$, will be formed for the evaluation purpose.

4.5. Analysis over binary datasets

To answer first part of Q_3, for solving binary datasets. Table 6 shows the average accuracy and standard deviation for the different datasets. Adaboost achieves the highest accuracy on six datasets $\{D_6, D_7, D_8, D_{10}, D_{11}, D_{13}\}$ while it uses 101 classifiers. RE and MDEP both of them achieve the highest accuracy for $\{D_3\}$ and $\{D_1, D_9, D_{14}\}$ respectively using 31 classifiers. The highest improvement percentage of 14.25% has been recorded by AdaB in comparison with MDEP for D_{13}. While MDEP recorded the highest improvement over AdaB by 38.49% for D_5. For almost all the datasets, the reordering metrics guarantee higher accuracy and go beyond what can be achieved by bagging. MDSQ is the only metric with a tendency to form the worst subensemble related to the investigated tasks. For the *Hill-valley* dataset, the poor performance of bagging is caused by the uneven response of individual members. While the performance analysis of RE to select a subensemble over the 100 executions, relates to grouping specific individual types. Where DT, Multinom, NB, JRip, and KNN are represented in-order according to the following percentages 33.97%, 28.97%, 20.48%, 10.45%, 6.13% respectively.

The average rank of Friedman test [41] *(AR-Friedman)* is presented in the last row of Table 6 with the best ranks being (in order) MDEP, DISC, EPIC, RE, UMEP, AdaB, CC, and MRMR. Next, the Wilcoxon test [42,43] for pairwise comparisons has been performed to detect significant differences between the two sample means. From Table 7, BSM is the worst ensemble selection strategy. All heuristic metrics, except MDSQ, significantly outperform bagging by 95% or 90%. Furthermore, we notice that MDSQ, AdaB, and RF are at the same level of competence with bagging. While the heterogeneous classifiers selected by RE and CC significantly outperform bagging. Finally, MDEP is the only metric that significantly outperforms both EPIC and UMEP by 95% and is the best metric regarding the investigated tasks.

4.6. Analysis over multiclass datasets

To answer second part of Q_3, to analyze the stability of the different pruning methods in multiclass classification tasks. Table 8 shows the average accuracy and the standard deviation over multiclass datasets. Adaboost and RF achieve the highest accuracy for datasets $\{D_{17}, D_{26}, D_{27}, D_{28}\}$ and $\{D_{16}, D_{25}, D_{29}, D_{30}\}$, respectively using the complete set of 101 classifiers. The highest improvement percentage of Adaboost over DISC has been recorded by 3.94% for D_{27}. While DISC recorded the highest improvement over Adaboost by 12.77% for D_{30}.

For datasets with a large number of samples and classes like $\{D_{22}, D_{23}\}$, as expected, we notice that DISC and MDEP are the best. Our explanation is that for complex decision spaces, it will be preferred to select classifiers that are more discriminant or that have the ability to classify difficult samples. For that, DISC is more promising to acquire complementary information inside the subensemble by reducing the internal conflict. While MDEP is preferred to balance between the individuals' accuracy and ensemble diversity. In addition, DISC is the best for D_{25} as the largest size multiclass dataset.

For datasets with a small number of classes and a small number of instances like $\{D_{16}, D_{21}, D_{24}\}$, the decision space becomes easier as low conflict will exist. For that, except MDSQ for D_{16}, RE proved

Table 8
Average accuracy and standard deviation over multiclass datasets for ensemble size $T = 101$ and P=30%. The values that outperform bagging are highlighted in bold.

		$T = 101$	$\hat{T} = 1$	$\hat{T} = 31$							$T = 101$	
#	Datatset	Bagging	BSM	RE	CC	MDSQ	MRMR	DISC	EPIC	UMEP	MDEP	AdaB	RF
D_{16}	Cleveland	57.07 ± 4.41	51.40 ± 7.76	**57.30 ±5.32**	57.02 ± 4.75	**57.35 ± 4.91**	**57.11 ± 4.66**	**57.38 ± 4.86**	**57.43 ± 5.08**	56.89 ± 4.81	**57.21 ± 5.05**	**57.52 ± 5.43**	**57.72 ± 5.29**
D_{17}	Cmc	53.30 ± 3.74	49.58 ± 3.81	53.29 ± 3.83	53.20 ± 3.66	**53.36 ±3.81**	**53.49 ± 3.97**	**53.80 ± 3.72**	**53.43 ± 3.69**	**53.82 ± 3.69**	53.14 ± 4.27	**56.30 ± 3.44**	**53.98 ± 3.42**
D_{18}	Dermatology	97.66 ± 2.57	93.75 ± 4.85	97.49 ± 2.62	97.52 ± 2.63	97.59 ± 2.44	97.57 ± 2.62	**97.88 ± 2.18**	**97.99 ± 2.14**	**97.96 ± 2.24**	**97.99 ± 2.14**	96.48 ± 2.64	97.65 ± 2.28
D_{19}	Flare	73.23 ± 3.41	**73.59 ± 3.27**	**74.65 ± 2.77**	**75.22 ± 3.03**	**75.31 ± 2.72**	69.89 ± 4.29	**75.60 ± 2.81**	**75.65 ± 2.82**	**75.57 ± 2.77**	**74.78 ± 3.63**	**75.26 ± 2.99**	**75.14 ± 2.45**
D_{20}	Led7digit	69.51 ± 5.28	60.31 ± 5.72	**72.12 ± 5.34**	**70.89 ± 5.69**	**73.17 ± 5.77**	**69.65 ± 6.07**	**71.39 ± 6.11**	**72.11 ± 5.88**	**71.22 ± 5.69**	**70.42 ± 6.52**	**72.66 ± 5.94**	**71.86 ± 5.64**
D_{21}	Lymphography	85.45 ± 8.78	77.45 ± 10.29	**86.34 ± 8.17**	**86.42 ± 8.75**	**86.23 ± 9.58**	**86.57 ± 9.17**	85.35 ± 9.39	84.94 ± 9.27	84.32 ± 9.59	85.30 ± 9.72	85.11 ± 9.01	84.55 ± 9.51
D_{22}	Mfeat-Fourier	83.10 ± 2.20	71.81 ± 2.96	**83.36 ± 2.51**	**83.38 ± 2.34**	**83.19 ± 2.43**	**83.58 ± 2.27**	**83.56 ± 2.46**	**83.20 ± 2.44**	**83.38 ± 2.40**	**83.58 ± 2.36**	81.86 ± 2.56	83.10 ± 2.17
D_{23}	Mfeat-karh	96.87 ± 1.08	90.75 ± 3.61	96.83 ± 1.13	96.78 ± 1.16	96.50 ± 1.41	96.75 ± 1.23	**97.13 ±1.09**	96.74 ± 1.18	96.86 ± 1.17	**97.01 ± 1.11**	95.40 ± 1.47	96.04 ± 1.23
D_{24}	Newthyroid	96.55 ± 3.68	95.43 ± 4.73	**96.88 ± 3.51**	**96.84 ± 3.50**	96.32 ± 4.11	**96.65 ± 3.45**	**96.60 ± 3.69**	**96.69 ± 3.78**	95.91 ± 3.98	96.46 ± 3.92	95.24 ± 4.07	96.08 ± 4.04
D_{25}	Satimage	89.66 ± 1.14	85.09 ± 1.39	**91.48 ± 1.02**	**91.63 ± 1.05**	**91.09 ± 1.04**	**91.61 ± 1.02**	**91.67 ± 1.04**	**91.52 ± 1.02**	**91.54 ± 0.98**	**91.50 ± 0.98**	88.25 ± 1.24	**91.82 ± 0.95**
D_{26}	Segment	97.05 ± 1.07	95.88 ± 1.58	**97.94 ± 0.95**	**98.01 ± 0.90**	**97.49 ± 1.06**	**97.97 ± 0.90**	**98.09 ± 0.98**	**98.13 ± 0.95**	**98.12 ± 0.91**	**98.13 ± 0.92**	**98.55 ± 0.83**	**97.93 ± 1.01**
D_{27}	Vehicle	74.99 ± 4.07	69.23 ± 4.85	**75.56 ± 4.11**	**75.42 ± 3.92**	**75.14 ± 3.90**	**75.48 ± 4.10**	**75.32 ± 3.92**	**75.13 ± 4.03**	**75.14 ± 3.80**	**75.13 ± 3.86**	**78.29 ± 4.10**	**75.13 ± 4.29**
D_{28}	WFR	96.62 ± 0.85	**99.03 ± 0.53**	**99.04 ± 0.41**	**99.06 ± 0.38**	**98.95 ± 0.44**	**99.04 ± 0.43**	**99.40 ± 0.32**	**99.40 ± 0.32**	**99.40 ± 0.31**	**99.38 ± 0.38**	**99.88 ± 0.14**	**99.43 ± 0.32**
D_{29}	Wine	98.11 ± 3.07	94.45 ± 5.44	97.94 ± 3.41	97.94 ± 3.41	97.81 ± 3.33	97.77 ± 3.53	**98.16 ± 3.17**	**98.16 ± 3.17**	**98.16 ± 3.07**	**98.16 ± 3.07**	96.20 ± 4.54	**98.22 ± 3.24**
D_{30}	Winequality-red	63.40 ± 2.97	56.71 ± 4.30	**68.38 ± 2.99**	**68.57 ± 3.01**	**68.30 ± 3.31**	**68.76 ± 3.08**	**68.55 ± 3.08**	**68.44 ± 2.92**	**68.06 ± 2.98**	**67.72 ± 3.13**	60.79 ± 3.30	**70.53 ± 3.44**
	AR-Friedman	8.5	11.67	**6.13**	**5.87**	**7.03**	**6.07**	**3.9**	**4.83**	**5.83**	**6.07**	**6.6**	**5.5**

Table 9
Summary of the Wilcoxon test (for Multiclass datasets). • = the method in the row improves the method of the column. ○ = the method in the column improves the method of the row. Upper diagonal of level significance $\alpha = 0.9$,Lower diagonal level of significance $\alpha = 0.95$.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
Bagging (1)	-	•	○	○	○	○	○	○	○	○		○
BSM (2)	○	-	○	○	○	○	○	○	○	○	○	○
RE (3)	•	•	-									
CC (4)	•	•		-			○					
MDSQ (5)	•	•			-		○					
MRMR (6)		•				-	○					
DISC (7)	•	•					-		•	•		
EPIC (8)	•	•						-				
UMEP (9)	•	•					○		-			
MDEP (10)	•	•								-		
AdaB (11)		•									-	
RF (12)	•	•										-

Table 10
Space and time complexities of different metrics.

Metric	Space complexity	Time complexity
RE	$\mathcal{O}(N \cdot T + M)$	$\mathcal{O}(T^2 \cdot N \cdot M)$
CC	$\mathcal{O}(N \cdot T + M)$	$\mathcal{O}(T^2 \cdot N)$
MDSQ	$\mathcal{O}(N \cdot T)$	$\mathcal{O}(T^2 \cdot N)$
MRMR	$\mathcal{O}(N \cdot T + M^2)$	$\mathcal{O}(T^2 \cdot N \cdot M)$
DISC	$\mathcal{O}(N \cdot T + M^2)$	$\mathcal{O}(T^2 \cdot N \cdot M)$
EPIC	$\mathcal{O}(N \cdot T + M)$	$\mathcal{O}(T \cdot N + T \cdot \log(T))$
UMEP	$\mathcal{O}(N \cdot T)$	$\mathcal{O}(T \cdot N + T \cdot \log(T))$
MDEP	$\mathcal{O}(N \cdot T + M)$	$\mathcal{O}(T \cdot N + T \cdot \log(T))$

its superiority to select effective subensemble based on its general accuracy to outperform margin-based metrics {MDSQ, UMEP, MDEP}. For datasets with a small number of classes and a larger number of instances like $\{D_{17}, D_{28}\}$; DISC, EPIC, and UMEP outperform RE.

For the statistical ranking, *(AR-Friedman)* is presented in the last row of Table 8 with the best ranks scored sequentially by DISC, EPIC, RF, UMEP, CC, MRMR & MDEP, and RE. While Table 9 shows the pairwise comparison between the 3 unpruned ensembles and the 9 pruned ones. Table 9 confirms that BSM is the worst selection (pruning) strategy. All the investigated metrics, except MRMR, significantly outperform bagging by 95% according to both the accuracy and the ensemble size. Adaboost and RF are at the same level of accuracy with all pruning metrics. However, their performance is achieved by combining T=101 classifiers. MDEP waived its rank in favor of DISC and EPIC. While, DISC is the best metric for selecting subensemble according to the investigated datasets. It has been confirmed that ordered bagging based on complementarity decisions works well for multiclass datasets, *confirmed by CC performance*. Next, it will be interesting to combine all the binary and multiclass datasets to analyze ordering-based pruning metrics in general.

4.7. General analysis over all datasets

To answer Q_4, a nonparametric statistical test is conducted over the thirty datasets, D_1 to D_{30}, to check if there is a significant difference between the performance of the ordered bagging methods or not. Using the methodology proposed by Demšar [44], Fig. 2 shows the Nemenyi post hoc test for $\alpha = 0.05$. Methods that are connected are not significantly different based on the absolute difference in the average rankings. The Critical Difference is shown (CD=3.06 for 12 methods, 30 datasets, $\alpha = 0.05$). The analysis shows that DISC, EPIC, and MDEP significantly outperform bagging by 95%. While, the inferior performance is recorded by BSM, Bagging, and MDSQ.

4.8. Prediction consistency

To answer Q_5, after statistical analysis and according to the Nemenyi test, we selected {DISC, EPIC, MDEP, UMEP, AdaB, RF, Bagging, BSM} and analyzed the distribution of their prediction accuracy over the 100 executions. Fig. 3 presents the range of the prediction accuracy around the median and how the heuristic metrics realize robust and stable predictions, with less number of internal classifiers, for a set of representative datasets $\{D_1, D_3, D_4, D_6, D_9, D_{19}, D_{25}, D_{28}\}$. The conclusion is that the ordering metrics are more effective to select a promising subensemble and their behavior reduces the performance variance.

4.9. Efficiency analysis

As demonstrated in [10], the efficiency of reordering metrics can be evaluated according to the following three aspects: the computational cost to extract the pruned subensemble, the required memory space to store the pruned ensembles, and the classification speed. While, some steps can be performed in parallel: the generation of the initial pool of classifiers, and the retrieving of classification decisions from the selected classifiers.

4.9.1. Space and time complexities

To answer Q_6, space and time complexities of the different heuristic metrics are summarized in Table 10 in terms of T, N, M. The memory requirements are estimated assuming that the decisions of the classifiers are stored in a matrix of size $N \times T$. For large datasets, it might be difficult to store this matrix in memory. In such a case, the whole matrix can be stored to a secondary memory device like the hard disk [10]. This would reduce the memory requirements to $\mathcal{O}(N)$, but the required disk access will slow down the classification process.

To empirically investigate how the heuristic measures depend on T, a series of experiments over the *SPECTF* dataset are performed. Table 11 presents the execution time of several heterogeneous classifiers with initial bagging of 25, 51, 75, 101, 151, and

Table 11
Average execution time in seconds (s) for the *SPECTF* dataset.

Metric/T	25	51	75	101	151	201
RE	0.09	0.26	0.78	1.48	3.95	5.91
CC	0.06	0.26	0.68	1.20	2.85	4.87
MDSQ	0.05	0.19	0.32	0.48	1.13	2.96
MRMR	0.10	0.91	1.10	2.83	4.09	7.66
DISC	0.10	0.58	1.43	2.17	3.64	7.01
EPIC	0.01	0.03	0.04	0.05	0.09	0.11
UMEP	0.01	0.02	0.03	0.04	0.07	0.10
MDEP	0.02	0.04	0.05	0.07	0.12	0.15

Table 12
The best and the average classification times as a percentage of the complete bagging time, $T = 101$ and P=30%.

#	Datatset	RE	CC	MDSQ	MRMR	DISC	EPIC	UMEP	MDEP	Time
D_1	Australian	18.7%	**13.0**%	14.3%	15.6%	**13.0**%	14.3%	16.5%	15.4%	Best
		30.1%	**24.2**%	**24.2**%	24.9%	25.4%	25.8%	27.0%	26.8%	Avg.
D_2	Blood-transfusion	19.4%	14.3%	14.3%	14.3%	14.3%	9.5%	11.9%	**7.1**%	Best
		30.5%	28.8%	28.3%	**28.2**%	29.6%	29.1%	29.3%	29.8%	Avg.
D_3	Breast-cancer	**15.7**%	19.4%	20.3%	19.4%	20.0%	20.0%	20.5%	22.9%	Best
		31.6%	32.7%	32.5%	33.0%	32.1%	34.0%	33.0%	33.5%	Avg.
D_4	German	15.0%	12.2%	**11.3**%	12.2%	12.8%	11.7%	12.8%	12.8%	Best
		18.5%	17.0%	16.6%	17.3%	17.4%	**16.2**%	17.9%	17.6%	Avg.
D_5	Hill-valley	22.2%	24.1%	16.9%	23.0%	10.9%	**5.4**%	6.8%	6.8%	Best
		34.2%	39.0%	21.4%	39.0%	16.3%	14.0%	**10.3**%	**10.3**%	Avg.
D_6	ILP	19.7%	13.6%	16.4%	18.2%	**9.1**%	**9.1**%	13.9%	13.9%	Best
		30.4%	26.7%	**23.8**%	26.7%	25.2%	**23.8**%	25.8%	26.0%	Avg.
D_7	Ionosphere	26.1%	26.5%	27.0%	**26.0**%	31.3%	34.8%	30.8%	30.8%	Best
		33.3%	35.2%	34.2%	33.9%	39.2%	42.6%	42.0%	42.1%	Avg.
D_8	Kr-vs-kp	15.6%	17.2%	18.3%	17.4%	**14.7**%	16.4%	16.1%	16.1%	Best
		22.4%	23.0%	24.8%	21.8%	22.0%	20.8%	**19.9**%	20.0%	Avg.
D_9	Mammographic	22.5%	19.1%	19.2%	19.1%	19.1%	**18.2**%	20.0%	20.0%	Best
		31.8%	27.7%	27.6%	27.4%	**26.9**%	28.2%	30.2%	30.6%	Avg.
D_{10}	Ringnorm	18.2%	**15.8**%	16.5%	**15.8**%	17.1%	16.7%	16.3%	16.3%	Best
		21.9%	**19.3**%	**19.3**%	**19.3**%	19.7%	24.1%	23.3%	23.2%	Avg.
D_{11}	Spambase	30.1%	33.8%	35.7%	33.8%	22.3%	18.4%	**16.4**%	**16.4**%	Best
		33.9%	38.0%	38.5%	38.1%	27.4%	19.9%	**18.6**%	**18.6**%	Avg.
D_{12}	SPECTF	**23.9**%	25.4%	24.8%	25.7%	28.4%	30.9%	30.9%	30.9%	Best
		33.0%	33.8%	**32.8**%	33.7%	38.4%	39.7%	40.0%	40.0%	Avg.
D_{13}	Tic-tac-toe	16.1%	14.5%	17.7%	**12.9**%	**12.9**%	17.7%	18.9%	18.9%	Best
		25.7%	23.4%	24.1%	**22.9**%	23.5%	23.3%	24.7%	24.6%	Avg.
D_{14}	Wdbc	19.9%	19.6%	21.5%	20.3%	18.4%	16.5%	**13.3**%	13.9%	Best
		27.7%	28.4%	30.4%	28.7%	25.2%	25.4%	**20.7**%	**20.7**%	Avg.
D_{15}	Wisconsin	20.4%	17.1%	16.7%	18.1%	**16.1**%	18.5%	25.0%	25.0%	Best
		28.2%	25.7%	**25.2**%	25.9%	26.9%	25.5%	33.0%	32.9%	Avg.
D_{16}	Cleveland	19.5%	16.1%	18.9%	16.1%	**15.2**%	16.0%	21.0%	21.0%	Best
		27.5%	**25.2**%	26.3%	25.7%	26.1%	26.7%	32.1%	31.8%	Avg.
D_{17}	Cmc	11.8%	12.5%	10.6%	10.5%	11.1%	**8.7**%	12.3%	12.5%	Best
		18.9%	17.2%	**15.3**%	18.8%	16.4%	16.0%	17.3%	18.0	Avg.
D_{18}	Dermatology	23.1%	26.0%	25.9%	23.1%	**14.8**%	20.1%	17.8%	16.6%	Best
		34.5%	33.9%	33.6%	32.2%	**21.5**%	27.2%	25.5%	25.2%	Avg.
D_{19}	Flare	24.6%	14.5%	17.0%	31.4%	15.1%	18.0%	**13.7**%	**13.7**%	Best
		34.9%	**20.2**%	22.4%	37.3%	23.1%	26.7%	23.3%	23.2%	Avg.
D_{20}	Led7digit	24.4%	7.8%	**6.2**%	25.6%	8.5%	17.1%	12.1%	11.1%	Best
		34.4%	19.1%	16.4%	38.1%	**14.9**%	24.2%	17.8%	17.5%	Avg.
D_{21}	Lymphography	21.9%	22.3%	17.7%	22.3%	22.0%	18.4%	**9.5**%	**9.5**%	Best
		30.3%	30.7%	**26.8**%	30.2%	29.5%	30.8%	29.5%	29.5%	Avg.
D_{22}	Mfeat-Fourier	**18.3**%	19.5%	20.4%	19.4%	20.0%	21.2%	20.8%	20.8%	Best
		22.1%	**21.5**%	22.9%	21.7%	21.7%	23.8%	23.1%	23.0%	Avg.
D_{23}	Mfeat-karh	17.5%	18.4%	13.3%	18.0%	6.4%	5.6%	**5.3**%	5.8%	Best
		22.8%	23.0%	15.1%	22.6%	8.6%	7.8%	**7.6**%	**7.6**%	Avg.
D_{24}	Newthyroid	16.7%	16.7%	22.2%	16.7%	**15.6**%	25.0%	20.8%	18.1%	Best
		31.0%	**30.0**%	35.1%	**30.0**%	34.8%	38.0%	38.5%	38.2%	Avg.
D_{25}	Satimage	**15.7**%	18.3%	24.5%	17.6%	18.9%	16.7%	17.4%	17.6%	Best
		18.4%	20.6%	26.2%	20.0%	20.7%	19.6%	20.5%	20.6%	Avg.
D_{26}	Segment	20.6%	19.2%	30.3%	20.2%	17.3%	17.5%	**16.4**%	17.2%	Best
		27.9%	24.3%	35.1%	25.6%	22.4%	21.8%	21.6%	**21.5**%	Avg.
D_{27}	Vehicle	22.7%	20.0%	**18.7**%	21.7%	20.7%	23.8%	24.9%	25.3%	Best
		29.5%	28.9%	29.9%	**27.8**%	28.9%	32.7%	32.8%	33.0%	Avg.
D_{28}	WFR	19.9%	17.3%	27.2%	21.1%	**15.5**%	16.0%	16.0%	16.2%	Best
		25.3%	25.0%	31.3%	24.2%	19.2%	**19.1**%	19.2%	19.3%	Avg.
D_{29}	Wine	18.4%	17.1%	22.2%	18.4%	13.4%	**8.8**%	12.3%	12.3%	Best
		30.6%	30.4%	33.6%	30.8%	**22.9**%	26.9%	30.3%	29.4%	Avg.
D_{30}	Winequality-red	12.9%	**8.8**%	**8.8**%	9.5%	**8.8**%	**8.8**%	12.9%	13.6	Best
		20.6%	**18.5**%	19.8%	18.8%	19.6%	18.6%	21.0%	21.5%	Avg.
Total Average Time		19.7%	18%	19.2%	19.4%	**16.1**%	16.7%	16.8%	16.6%	Best
		28.1%	26.4%	26.5%	27.5%	**24.2**%	25.1%	25.2%	25.2%	Avg.

201. The values of the remaining parameters are $M = 2$, $N_{train} = 314$, $D_{pr} = N_{train}$, $P = 20\%$. The times are averaged over 100 executions in Intel(R) Core(TM) i5-7200 CPU @ 2.50GHZ with 8 GB of RAM.

Results from Table 11 show that UMEP is the fastest method because the rank is calculated based on the classifier's correct classified samples from the pruning set D_{pr}. Additionally, the computation time of EPIC and MDEP is also rather fast with an increasing complexity approximately log-linear with respect to T. While, the execution time of RE, CC, MDSQ, MRMR, and DISC is quadratic in T. Both MRMR and DISC are much slower, as the selection of each classifier is conditioned by more internal calculations.

Besides the efficacy of the ensemble pruning metrics, other main benefits concern the efficiency as classification speed and the storage requirements. The classification speed depends on both (1) the size of the pruned ensemble, and (2) the complexity of the base classifiers. While, the allocated memory space for complete bagging will be released to store the pruned subensemble.

Table 13
The best and the average memory space requirements as a percentage of storage space of the complete bagging, $T = 101$ and P=30%.

#	Datatset	RE	CC	MDSQ	MRMR	DISC	EPIC	UMEP	MDEP	Space
D_1	Australian	**16.5**%	28.7%	25.2%	23.2%	23.2%	16.9%	17.1%	17.1%	Best
		26.1%	32.9%	33.4%	32.9%	29.5%	25.8%	**23.1**%	**23.1**%	Avg.
D_2	Blood-transfusion	**26.9**%	32.9%	32.2%	32.9%	30.8%	32.9%	28.8%	30.8%	Best
		34.2%	37.9%	38.2%	37.8%	37.9%	37.9%	35.6%	36.5%	Avg.
D_3	Breast-cancer	29.6%	35.0%	38.5%	36.9%	36.8%	39.0%	**28.7**%	29.6%	Best
		34.9%	42.7%	44.8%	42.8%	43.1%	44.6%	35.1%	36.2%	Avg.
D_4	German	26.4%	34.4%	39.3%	34.4%	28.6%	29.2%	**20.8**%	**20.8**%	Best
		36.7%	43.3%	46.6%	43.5%	35.9%	35.4%	**26.9**%	27.4%	Avg.
D_5	Hill-valley	9.8%	**8.5**%	28.7%	**8.5**%	29.6%	25.7%	33.2%	33.2%	Best
		17.7%	**15.8**%	33.6%	15.9%	34.9%	43.0%	41.5%	41.2%	Avg.
D_6	ILP	**13.2**%	17.2%	22.4%	17.9%	17.4%	19.4%	17.8%	17.8%	Best
		21.8%	25.5%	28.9%	25.5%	27.5%	26.8%	25.1%	25.1%	Avg.
D_7	Ionosphere	20.7%	20.9%	21.8%	20.5%	12.1%	**11.3**%	11.7%	11.7%	Best
		24.5%	24.7%	24.5%	24.8%	15.6%	12.9%	**12.7**%	**12.7**%	Avg.
D_8	Kr-vs-kp	22.4%	19.5%	23.8%	27.0%	20.9%	**19.4**%	**19.4**%	**19.4**%	Best
		31.1%	35.7%	**31.0**%	35.4%	35.0%	33.0%	33.7%	33.7%	Avg.
D_9	Mammographic	**20.6**%	28.5%	30.1%	30.1%	28.8%	28.3%	21.0%	21.0%	Best
		30.3%	36.2%	35.6%	36.7%	37.3%	34.3%	**28.7**%	29.6%	Avg.
D_{10}	Ringnorm	10.0%	8.3%	31.2%	8.3%	5.5%	**2.5**%	**2.5**%	**2.5**%	Best
		15.9%	13.4%	32.8%	13.5%	8.0%	**2.7**%	**2.7**%	**2.7**%	Avg.
D_{11}	Spambase	29.0%	38.9%	38.1%	42.1%	48.7%	5.9%	**2.6**%	**2.6**%	Best
		37.3%	45.5%	43.6%	45.8%	52.7%	15.7%	7.0%	**6.8**%	Avg.
D_{12}	SPECTF	17.6%	19.6%	23.9%	20.0%	10.5%	**10.4**%	**10.4**%	**10.4**%	Best
		23.6%	25.3%	28.5%	25.2%	12.3%	**11.3**%	**11.3**%	**11.3**%	Avg.
D_{13}	Tic-tac-toe	**29.3**%	33.8%	31.4%	33.9%	33.9%	33.7%	29.4%	29.4%	Best
		37.7%	39.9%	40.0%	39.9%	39.8%	39.4%	**37.2**%	**37.2**%	Avg.
D_{14}	Wdbc	25.8%	24.7%	**20.8**%	26.5%	22.2%	25.1%	22.7%	22.7%	Best
		31.0%	31.2%	**28.6**%	30.9%	31.8%	29.6%	31.3%	31.5%	Avg.
D_{15}	Wisconsin	21.4%	22.1%	25.2%	22.1%	16.6%	20.5%	**16.5**%	16.7%	Best
		29.4%	32.0%	31.3%	31.6%	30.4%	29.7%	**24.0**%	24.1%	Avg.
D_{16}	Cleveland	19.0%	20.1%	16.5%	20.3%	18.1%	19.1%	14.9%	**13.9**%	Best
		22.0%	23.5%	20.0%	24.1%	20.9%	21.0%	**16.8**%	**16.8**%	Avg.
D_{17}	Cmc	29.5%	29.7%	32.3%	30.1%	**25.6**%	29.0%	25.7%	25.7%	Best
		34.6%	35.6%	35.3%	35.1%	33.5%	34.5%	32.3%	**31.9**%	Avg.
D_{18}	Dermatology	26.8%	28.6%	27.3%	28.7%	31.4%	**22.4**%	25.7%	25.7%	Best
		31.1%	31.6%	31.9%	32.0%	37.0%	32.4%	34.0%	34.2%	Avg.
D_{19}	Flare	18.5%	29.5%	30.0%	**13.5**%	29.6%	29.7%	32.4%	30.8%	Best
		26.2%	39.5%	38.5%	**21.2**%	39.5%	36.3%	40.4%	40.0%	Avg.
D_{20}	Led7digit	23.3%	27.0%	28.9%	**18.1**%	32.9%	20.7%	32.9%	33.4%	Best
		30.3%	35.8%	39.5%	**23.9**%	42.3%	35.7%	39.4%	39.3%	Avg.
D_{21}	Lymphography	**25.8**%	27.8%	26.2%	26.9%	27.6%	27.5%	30.8%	30.8%	Best
		30.9%	32.0%	33.5%	31.8%	32.9%	33.7%	34.5%	34.6%	Avg.
D_{22}	Mfeat-Fourier	6.8%	10.5%	4.4%	11.0%	12.7%	**4.3**%	6.1%	6.1%	Best
		14.9%	16.5%	6.2%	16.1%	18.1%	**4.9**%	8.5%	8.5%	Avg.
D_{23}	Mfeat-karh	27.0%	29.3%	**24.6**%	31.2%	60.8%	52.4%	52.4%	52.4%	Best
		33.4%	33.8%	**26.8**%	36.8%	66.3%	61.7%	61.7%	62.0%	Avg.
D_{24}	Newthyroid	28.4%	28.3%	**23.0**%	26.6%	23.1%	23.9%	24.5%	24.5%	Best
		30.9%	31.0%	31.4%	**30.4**%	32.7%	31.5%	32.8%	32.9%	Avg.
D_{25}	Satimage	9.6%	16.5%	9.5%	14.2%	16.6%	**5.0**%	9.6%	9.6%	Best
		13.8%	20.5%	**11.0**%	19.2%	20.5%	14.6%	16.4%	16.6%	Avg.
D_{26}	Segment	**8.9**%	15.0%	13.0%	13.4%	10.5%	12.4%	12.3%	12.3%	Best
		18.8%	19.6%	17.2%	18.8%	**16.1**%	17.0%	17.1%	17.2%	Avg.
D_{27}	Vehicle	15.0%	14.9%	11.4%	15.0%	10.2%	**8.5**%	**8.5**%	**8.5**%	Best
		20.5%	20.1%	14.9%	21.0%	16.1%	**9.8**%	9.9%	9.9%	Avg.
D_{28}	WFR	11.4%	9.2%	18.2%	8.9%	**1.9**%	2.0%	2.0%	2.0%	Best
		17.3%	17.1%	20.6%	17.8%	3.4%	**2.1**%	**2.1**%	**2.1**%	Avg.
D_{29}	Wine	30.1%	30.1%	**28.3**%	29.4%	31.1%	30.8%	29.0%	30.3%	Best
		31.5%	31.4%	33.4%	**31.0**%	34.8%	35.3%	33.9%	34.1%	Avg.
D_{30}	Winequality-red	14.8%	16.4%	16.5%	16.2%	15.1%	15.2%	**12.0**%	**12.0**%	Best
		18.2	19.3%	18.3%	20.1%	17.4%	19.1%	**15.2**%	**15.2**%	Avg.
Total Average Space		20.5%	23.5%	24.8%	22.9%	23.8%	20.8%	**20**%	20.1%	Best
		26.9%	29.6%	30%	28.7%	30.1%	27.1%	**25.7**%	25.8%	Avg.

To empirically investigate the reduction in the classification time, Table 12 shows the best and the average classification times as a percentage of the time employed by the complete bagging. Regarding that, the classification time has reduced dramatically by 75% of the complete bagging time. Table 13 shows the best and the average memory space requirements for the analyzed methods as a percentage of the storage space required by the complete bagging. We conclude that the pruned ensemble saves, often, close to 70% from reserved space by complete bagging. The reported values from Tables 12 and 13 are calculated for 20 executions.

5. Applications

Classifier ensemble pruning can be applied in many areas that use multiple classifier systems [1]. Among those areas are: *Remote sensing* to identify the materials that cover a surface area [45], and to detect the change of land cover in time series [46]; *Secure in-*

formation systems to provide predictive solutions with the aim of avoiding distributed denial of service (DDoS) attacks [47], malicious code and spyware program [48], wireless sensor networks (WSNs) attacks [49]; *Banking and economical systems* for tracking complex and large volumes of online transactions with the aim of predicting any suspicious attempts [50]; *Medical systems* to help in medical decisions about treatment and prognosis [51], also can be used for drug development [52], designing of medical robotics; *Data driven-based modeling* to predict machine failure and modeling the remaining useful life (RUL) of engines [53]; *Recommender systems* predict users' behavior to be capable of detecting their interests and needs [54]; *Emotion recognition* is the process of identifying the human emotion, and it works best when combining multiple modalities in context [55].

For sample areas, ensemble pruning is practically embedded in multi-sensor based human activity recognition (HAR) systems [4,56–58]. Those systems have gained popularity in human-centered applications: intelligent interactive, health monitoring, elderly assisted living, sports activity, computer interaction, fall detection, security monitoring, etc. While some sensors may induce a negative effect (noisy or corrupted signals) in the learning. It becomes crucial to select the number and the type of multi-sensor, appropriately, to save the communication bandwidth, to reduce the power waste, and to optimize the multi-sensor fusion without degrading the performance accuracy.

In online classification learning, ensemble pruning has been applied for target tracking in streamed video sequences [59]. The tracking problem can be formulated as a binary classification to distinguish the target from the background. Regarding that, ensemble of classifiers can be trained on heterogeneous features (e.g. color features, texture, motion, etc.) to improve the robustness of the system. While ensemble pruning is integrated to speed up the tracking process via reducing the number of classifiers.

In natural language processing and text mining, ensemble pruning has been incorporated in [28] to extract subjective information from online text documents. Ensemble pruning proved its capability to establish an effective sentiment classification scheme with high predictive accuracy and efficiency.

6. Conclusions

Multiple classifier systems are superior, in general, to any of the single classifiers they are composed of. However, three main drawbacks are reported for those systems: (1) a large pool of classifiers should be built, (2) a large memory space should be reserved to store those models, and (3) a large classification time will be consumed to combine multiple decisions. To alleviate these drawbacks, this article discussed the concept and the benefits of ensemble pruning to select a subset of the original base classifiers of the ensemble. An effective and fast heuristic metrics are analyzed to reorder the classifiers in the generated random bagging. The main conclusions from this study are as follows:

- In general, the larger the size of the initial pool of classifiers, the better the accuracy but also the complexity of the selected subensemble. In any case, we found that it is enough to use rather small subensembles (20–30 members) to achieve very good performance.
- The analyzed metrics applied to small-size bagged ensembles can easily outperform the large-size ensembles.
- The performance of tested methods can keep on improving even after the bagging ensemble accuracy stabilizes.
- The combination of heterogeneous classifiers has been found to produce more promising subensembles in terms of diversity and complementarity, than the subensembles of homogeneous classifiers.
- For tested binary datasets, the results of the investigated metrics are significantly better than those of bagging, with $\alpha = 0.95$, in terms of accuracy. The method that obtained the best results for these problems is MDEP.
- The strategy that selects the best single model from a group of classifiers is the worst approach in terms of accuracy.
- For multiclass datasets, most of the investigated metrics significantly outperform bagging, with $\alpha = 0.95$, in terms of accuracy. The methods that obtained the best results for these problems are DISC and EPIC.
- Considering the 30 datasets, there are three metrics that stand out from the rest. Specifically, DISC, EPIC, and MDEP get the best results, which are significantly better than those of bagging at $\alpha = 95\%$.
- The variability of the tested heuristics has been analyzed by displaying the prediction accuracies of the subensembles around their median value. The results show the robustness and stability of the metrics.
- Regarding computational efficiency, UMEP is the fastest heuristic with a log-linear performance with respect to the number of base classifiers. On the other hand, MRMR and DISC are the slowest of the investigated methods (quadratic in number of classifiers).
- The general analysis over the 30 datasets proves that the ordering metrics are comparable with Random Forest and Adaboost in terms of accuracy, but they are much better in terms of ensemble size.
- Pruning metrics have great impact to speed up the classification time, to reduce memory requirements, and to improve performance accuracy. For all these reasons, embedding pruning ensemble techniques in real classification systems can be critical.

In summary, ordering-based pruning metrics are more accurate, computationally faster, and have lower memory requirements than bagging ensembles. The tested heuristics generally find near-optimum subensembles with respect to the generalization performance. Metrics based on removing the redundancy of classifiers, as MRMR, affect the majority voting procedure. Among the tested heuristics, UMEP, EPIC, and MDEP have the lowest computational cost for obtaining the pruned ensemble, which is linear with respect to ensemble size. The other investigated metrics are also rather efficient, but they present a larger computational cost (quadratic in T).

In the future, it would be interesting to analyze other combination functions different than majority voting, in order to maximize the accuracy of the subensembles. In addition, another interesting approach would be to analyze the performance of the investigated metrics in human-centered applications. Moreover, we will analyze a hybrid of ordering-based pruning metrics. In that case, the classifiers' ranks via each metric could be aggregated to determine the proper subensemble. Multi-objective optimization could be further applied to find pareto optimum subensembles in terms of size and accuracy. One more interesting research line, is to characterize the difficulty of a classification problem via data complexity measures, and analyze this information with the investigated pruning metrics.

Declaration of Competing Interest

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

Acknowledgements

We would like to thank the anonymous reviewers for their diligent work and efficient efforts. This project has received funding

from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement N° 665959. Besides, this work was supported in part by the LOGISTAR project, funded by the European Union Horizon 2020 Research and Innovation Programme grant agreement No. 769142. Michal Wozniak was partially supported by the Polish National Science Center under the grant No. 2017/27/B/ST6/01325. Gonzalo Martínez-Munoz was partially supported by PID2019-106827GB-I00 / AEI / 10.13039/501100011033.

References

[1] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17.
[2] A.M. Mohammed, E. Onieva, M. Woźniak, Training set selection and swarm intelligence for enhanced integration in multiple classifier systems, Appl. Soft Comput. 95 (2020) 106568.
[3] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[4] J. Cao, W. Li, C. Ma, Z. Tao, Optimizing multi-sensor deployment via ensemble pruning for wearable activity recognition, Inf. Fusion 41 (2018) 68–79.
[5] N. Li, Y. Yu, Z.-H. Zhou, Diversity regularized ensemble pruning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2012, pp. 330–345.
[6] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[7] A.M. Mohammed, E. Onieva, M. Woźniak, Vertical and horizontal data partitioning for classifier ensemble learning, in: International Conference on Computer Recognition Systems, Springer, 2019, pp. 86–97.
[8] S. González, S. García, J. Del Ser, L. Rokach, F. Herrera, A practical tutorial on bagging and boosting based ensembles for machine learning: algorithms, software tools, performance study, practical perspectives and opportunities, Inf. Fusion 64 (2020) 205–237.
[9] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, John Wiley & Sons, 2014.
[10] G. Martínez-Muñoz, D. Hernández-Lobato, A. Suarez, An analysis of ensemble pruning techniques based on ordered aggregation, IEEE Trans. Pattern Anal. Mach. Intell. 31(2) (2009) 245–259.
[11] L. Guo, S. Boukir, Margin-based ordered aggregation for ensemble pruning, Pattern Recognit. Lett. 34 (6) (2013) 603–609.
[12] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, M. Xu, Margin & diversity based ordering ensemble pruning, Neurocomputing 275 (2018) 237–246.
[13] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: ICML, vol. 97, Citeseer, 1997, pp. 211–218.
[14] G. Rätsch, T. Onoda, K.-R. Müller, Soft margins for adaboost, Mach. Learn. 42 (3) (2001) 287–320.
[15] Z. Lu, X. Wu, X. Zhu, J. Bongard, Ensemble pruning via individual contribution ordering, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 871–880.
[16] R.E. Banfield, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, Ensemble diversity measures and their application to thinning, Inf. Fusion 6 (1) (2005) 49–62.
[17] R. Diao, F. Chao, T. Peng, N. Snooke, Q. Shen, Feature selection inspired classifier ensemble reduction, IEEE Trans. Cybern. 44 (8) (2013) 1259–1268.
[18] L. Zhang, W. Srisukkham, S.C. Neoh, C.P. Lim, D. Pandit, Classifier ensemble reduction using a modified firefly algorithm: an empirical evaluation, Expert Syst. Appl. 93 (2018) 395–422.
[19] X. Zhu, Z. Ni, L. Ni, F. Jin, M. Cheng, J. Li, Improved discrete artificial fish swarm algorithm combined with margin distance minimization for ensemble pruning, Comput. Ind. Eng. 128 (2019) 32–46.
[20] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, Dynamic classifier selection: recent advances and perspectives, Inf. Fusion 41 (2018) 195–216.
[21] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, T.I. Ren, Meta-des: a dynamic ensemble selection framework using meta-learning, Pattern Recognit. 48 (5) (2015) 1925–1935.
[22] X. Xia, T. Lin, Z. Chen, Maximum relevancy maximum complementary based ordered aggregation for ensemble pruning, Appl. Intell. 48 (9) (2018) 2568–2579.
[23] C. Tamon, J. Xiang, On the boosting pruning problem, in: European Conference on Machine Learning, Springer, 2000, pp. 404–412.
[24] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, Front. Comput. Sci. 14 (2) (2020) 241–258.
[25] G. Tsoumakas, I. Partalas, I. Vlahavas, An ensemble pruning primer, in: Applications of Supervised and Unsupervised Ensemble Methods, Springer, 2009, pp. 1–13.
[26] M.N. Adnan, M.Z. Islam, Optimizing the number of trees in a decision forest to discover a subforest with high ensemble accuracy using a genetic algorithm, Knowl. Based Syst. 110 (2016) 86–97.
[27] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, Q. Zou, Libd3c: ensemble classifiers with a clustering and dynamic selection strategy, Neurocomputing 123 (2014) 424–435.
[28] A. Onan, S. Korukoğlu, H. Bulut, A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification, Inf. Process. Manage. 53 (4) (2017) 814–833.
[29] P. Zyblewski, M. Woźniak, Novel clustering-based pruning algorithms, Pattern Anal. Appl. (2020) 1–10.
[30] S. Mao, L. Jiao, L. Xiong, S. Gou, Greedy optimization classifiers ensemble based on diversity, Pattern Recognit. 44 (6) (2011) 1245–1261.
[31] I. Partalas, G. Tsoumakas, I. Vlahavas, An ensemble uncertainty aware measure for directed hill climbing ensemble pruning, Mach. Learn. 81 (3) (2010) 257–282.
[32] C.A. Shipp, L.I. Kuncheva, Relationships between combination methods and measures of diversity in combining classifiers, Inf. Fusion 3 (2) (2002) 135–148.
[33] M. Aksela, Comparison of classifier selection methods for improving committee performance, in: International Workshop on Multiple Classifier Systems, Springer, 2003, pp. 84–93.
[34] E.M. Dos Santos, R. Sabourin, P. Maupin, Overfitting cautious selection of classifier ensembles with genetic algorithms, Inf. Fusion 10 (2) (2009) 150–162.
[35] G. Martınez-Munoz, A. Suárez, Aggregation ordering in bagging, in: Proc. of the IASTED International Conference on Artificial Intelligence and Applications, Citeseer, 2004, pp. 258–263.
[36] G.D. Cavalcanti, L.S. Oliveira, T.J. Moura, G.V. Carvalho, Combining diversity measures for ensemble pruning, Pattern Recognit. Lett. 74 (2016) 38–45.
[37] C.O. Sakar, O. Kursun, F. Gurgen, A feature selection method based on kernel canonical correlation analysis and the minimum redundancy–maximum relevance filter method, Expert Syst. Appl. 39 (3) (2012) 3432–3437.
[38] A. Unler, A. Murat, R.B. Chinnam, Mr2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, Inf. Sci. 181 (20) (2011) 4625–4641.
[39] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[40] J. Quinlan, C4. 5: Programs for Machine Learning, Elsevier, 2014.
[41] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (200) (1937) 675–701.
[42] F. Wilcoxon, Individual comparisons by ranking methods, in: Breakthroughs in Statistics, Springer, 1992, pp. 196–202.
[43] S. Garcia, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, J. Mach. Learn. Res. 9 (Dec) (2008) 2677–2694.
[44] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[45] V.L. Diengdoh, S. Ondei, M. Hunt, B.W. Brook, A validated ensemble method for multinomial land-cover classification, Ecol. Inform. 56 (2020) 101065.
[46] A. Wang, Y. Wang, Y. Chen, Hyperspectral image classification based on convolutional neural network and random forest, Remote Sens. Lett. 10 (11) (2019) 1086–1094.
[47] S. Das, A.M. Mahfouz, D. Venugopal, S. Shiva, DDoS intrusion detection through machine learning ensemble, in: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), IEEE, 2019, pp. 471–477.
[48] D. Gupta, R. Rani, Improving malware detection using big data and ensemble learning, Comput. Electr. Eng. 86 (2020) 106729.
[49] S. Otoum, B. Kantarci, H.T. Mouftah, A novel ensemble method for advanced intrusion detection in wireless sensor networks, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020, pp. 1–6.
[50] S. Bagga, A. Goyal, N. Gupta, A. Goyal, Credit card fraud detection using pipeling and ensemble learning, Procedia Comput Sci 173 (2020) 104–112.
[51] J. Lu, E. Song, A. Ghoneim, M. Alrashoud, Machine learning for assisting cervical cancer diagnosis: an ensemble approach, Future Gener. Comput. Syst. 106 (2020) 199–205.
[52] G. Adam, L. Rampášek, Z. Safikhani, P. Smirnov, B. Haibe-Kains, A. Goldenberg, Machine learning approaches to drug response prediction: challenges and recent progress, npj Precis. Oncol. 4 (1) (2020) 1–10.
[53] Z. Li, K. Goebel, D. Wu, Degradation modeling and remaining useful life prediction of aircraft engines using ensemble learning, J. Eng. Gas Turbine Power 141 (4) (2019).
[54] S. Forouzandeh, K. Berahmand, M. Rostami, Presentation of a recommender system with ensemble learning and graph embedding: a case on MovieLens, Multimed. Tools Appl. 80 (5) (2021) 7805–7832.
[55] M. Hao, W.-H. Cao, Z.-T. Liu, M. Wu, P. Xiao, Visual-audio emotion recognition based on multi-task and ensemble learning with multiple features, Neurocomputing 391 (2020) 42–51.
[56] Y. Tian, J. Zhang, Optimizing sensor deployment for multi-sensor-based HAR system with improved glowworm swarm optimization algorithm, Sensors 20 (24) (2020) 7161.
[57] Y. Tian, J. Zhang, L. Chen, Y. Geng, X. Wang, Selective ensemble based on extreme learning machine for sensor-based human activity recognition, Sensors 19 (16) (2019) 3468.
[58] Y. Tian, X. Wang, L. Chen, Z. Liu, Wearable sensor-based human activity recognition via two-layer diversity-enhanced multiclassifier recognition method, Sensors 19 (9) (2019) 2039.
[59] I. Visentini, L. Snidaro, G.L. Foresti, Diversity-aware classifier ensemble selection via f-score, Inf. Fusion 28 (2016) 24–43.

Amgad M. Mohammed received his Master degree in evolutionary computation from Menoufia University, Egypt, in 2015. Received the PhD degree in information engineering from Universidad de Deusto, Spain, in 2021. He was funded by the European Commission through Marie Curie scholarship program during his doctorate study. His main research interests are on Ensemble of Classifiers, Artificial Intelligence, Evolutionary Computation, Information Fusion.

Enrique Onieva is currently a Professor in artificial intelligence, machine learning, and big data with the University of Deusto, as well as a Researcher of intelligent transportation systems applications related to data processing, mobility, and smart city solutions in the DeustoTech-Mobility Research Unit. He has participated in more than 25 research projects, including, CYBERCARS-2 (FP6), ICSI (FP7), and PostLowCit (CEF-Transport). He is responsible for the Artificial Intelligence Work Package of the Project TIMON (H2020) and a Project Coordinator of the LOGISTAR Project (H2020). He has authored more than 100 scientific articles. His research interest includes the application of artificial intelligence to intelligent transportation systems, including fuzzylogic based decision, evolutionary optimization, and machine learning.

Michał Wózniak is a Professor of Computer Science in the Department of Systems and Computer Networks, Wroclaw University of Technology, Poland. He received an MSc degree in Biomedical Engineering from the Wroclaw University of Technology in 1992, and PhD and DSc (habilitation) degrees in Computer Science in 1996 and 2007, respectively. Professor Woózniak has published over 180 papers and two books, and edited eight books. He has been involved in several research projects related to the abovementioned topics and has been a consultant on several commercial projects for well-known Polish companies and the public administration. Professor Woózniak is a senior member of the IEEE and a member of the International Biometric Society. Fields of interest - machine learning, especially inductive learning, data and web mining, learning on distributed and streaming data - pattern recognition, especially combined and compound classifiers, concept drift, recognition with context - telemedicine and medical decision support - computer and networks security, especially IDS, IPS, and anti-spam filters design - distributed algorithms.

Gonzalo Martínez-Muñoz received the university degree in Physics (1995) and Ph.D. degree in Computer Science (2006) from the Universidad Autóonoma de Madrid (UAM). From 1996 to 2002 he worked in industry. Until 2008 he was an interim assistant professor in the Computer Science Department of the UAM. During 2008/2009, he worked as a Fulbright postdoc researcher at Oregon State University in the group of Professor Thomas G. Dietterich. He is currently a professor at Computer Science Department at UAM. His research interests include machine learning, computer vision, pattern recognition, neural networks, decision trees, and ensemble learning