Mach Learn (2010) 81: 257–282
DOI 10.1007/s10994-010-5172-0

An ensemble uncertainty aware measure for directed hill climbing ensemble pruning

Ioannis Partalas · Grigorios Tsoumakas · Ioannis Vlahavas

Received: 12 February 2009 / Revised: 4 March 2010 / Accepted: 8 March 2010 /
Published online: 19 May 2010

Abstract This paper proposes a new measure for ensemble pruning via directed hill climbing, dubbed Uncertainty Weighted Accuracy (UWA), which takes into account the uncertainty of the decision of the current ensemble. Empirical results on 30 data sets show that using the proposed measure to prune a heterogeneous ensemble leads to significantly better accuracy results compared to state-of-the-art measures and other baseline methods, while keeping only a small fraction of the original models. Besides the evaluation measure, the paper also studies two other parameters of directed hill climbing ensemble pruning methods, the search direction and the evaluation dataset, with interesting conclusions on appropriate values.

Keywords Ensemble pruning · Ensemble selection · Ensemble methods

1 Introduction

Ensemble methods (Dietterich 2000) has been a very popular research topic during the last decade. Their success arises largely from the fact that they offer an appealing solution to several interesting learning problems of the past and the present, such as improving predictive performance, scaling inductive algorithms to large databases, learning from multiple physically distributed data sets and learning from concept-drifting data streams.

Typically, ensemble methods comprise two phases: the *production* of multiple predictive models and their *combination*. Recent work (Banfield et al. 2005; Caruana et al. 2004;

Editor: H. Blockeel.

I. Partalas (✉) · G. Tsoumakas · I. Vlahavas
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
e-mail: partalas@csd.auth.gr

G. Tsoumakas
e-mail: greg@csd.auth.gr

I. Vlahavas
e-mail: vlahavas@csd.auth.gr

Margineantu and Dietterich 1997; Giacinto et al. 2000; Fan et al. 2002; Martinez-Munoz and Suarez 2004, 2006; Partalas et al. 2008; Tsoumakas et al. 2005), has considered an additional intermediate phase that deals with the reduction of the ensemble size prior to combination. This phase is commonly called *ensemble pruning*, while other names include *selective ensemble*, *ensemble thinning* and *ensemble selection*.

Ensemble pruning is important for two reasons: *efficiency* and *predictive performance*. Having a very large number of models in an ensemble adds a lot of computational overhead. For example, decision tree models may have large memory requirements (Margineantu and Dietterich 1997) and lazy learning methods have a considerable computational cost during execution. The minimization of run-time overhead is crucial in certain applications, such as in stream mining. In addition, when models are distributed over a network, the reduction of models leads to the reduction of the important cost of communication. Equally important is the second reason, predictive performance. An ensemble may consist of both high and low predictive performance models. The latter may negatively affect the overall performance of the ensemble. Pruning these models while maintaining a high diversity among the remaining members of the ensemble is typically considered a proper recipe for an effective ensemble.

The ensemble pruning problem can be posed as an optimization problem as follows: Find the subset of the original ensemble that optimizes a measure indicative of its generalization performance (for example accuracy on a separate validation set). As the number of subsets of an ensemble consisting of T models is $2^T - 1$ (the empty set is not accountable), exhaustive search becomes intractable for a moderate ensemble size. Several efficient methods that are based on a directed hill climbing search in the space of subsets report good predictive performance results (Banfield et al. 2005; Caruana et al. 2004; Margineantu and Dietterich 1997; Martinez-Munoz and Suarez 2004). These methods start with an empty (or full) initial ensemble and search the space of different ensembles by iteratively expanding (or contracting) the initial ensemble by a single model. The search is guided by an evaluation measure that is based on either the predictive performance or the diversity of the alternative subsets. The evaluation measure is the main component of a directed hill climbing algorithm and it differentiates the methods that fall into this category.

The primary contribution of this work is a new measure for directed hill climbing ensemble pruning (DHCEP) that takes into account the uncertainty of the decision of the current ensemble. Empirical results on 30 data sets show that using the proposed measure to prune a heterogeneous ensemble leads to significantly better accuracy results compared to state-of-the-art measures and other baseline methods, while keeping only a small fraction of the original models. In addition, it is shown that the proposed measure maintains its lead across a variety of pruning levels. The secondary contribution of this work is an empirical study of the main parameters (search direction, evaluation dataset, evaluation measure) of DHCEP methods, leading to interesting conclusions on suitable settings.

This paper extends our previous work (Partalas et al. 2008) in the following respects. First of all, it empirically compares a variety of different combinations of values for the main parameters of DHCEP methods, instead of specific instantiations. This has led to interesting new conclusions, such as the fact that the proposed measure significantly outperforms its rivals when used in a forward search direction. The comparison is based on a much larger number of datasets. Finally, this paper includes an extended description of the proposed measure elaborating on its key issues.

The remainder of this paper is structured as follows: Sect. 2 presents background information on ensemble methods. Section 3 includes an extensive introduction to DHCEP methods and their main parameters. Section 4 presents the proposed measure. Section 5 describes the setup of the empirical study and Sect. 6 presents and discusses the results. Finally, Sect. 7 concludes this work and poses future research directions.

2 Ensemble methods

2.1 Producing the models

An ensemble can be composed of either homogeneous or heterogeneous models. Homogeneous models derive from different executions of the same learning algorithm by using different values for the parameters of the learning algorithm, injecting randomness into the learning algorithm or through the manipulation of the training instances, the input attributes and the model outputs (Dietterich 2000). Two popular methods for producing homogeneous models are bagging (Breiman 1996) and boosting (Schapire 1990).

Heterogeneous models derive from running different learning algorithms on the same dataset. Such models have different views about the data, as they make different assumptions about them. For example, a neural network is robust to noise in contrast to a k-nearest neighbor classifier.

In the empirical evaluation part of this paper we focus on ensembles consisting of models produced by running different learning algorithms, each with a variety of different parameter settings.

2.2 Combining the models

A lot of different ideas and methods have been proposed in the past for the combination of classification models. The necessity for high classification performance in some critical domains (e.g. medical, financial, intrusion detection) has motivated researchers to explore methods that combine different classification algorithms in order to overcome the limitations of individual learning paradigms.

Unweighted and Weighted Voting are two of the simplest methods for combining not only Heterogeneous but also Homogeneous models. In Voting, each model outputs a class value (or ranking, or probability distribution) and the class with the most votes (or the highest average ranking, or average probability) is the one proposed by the ensemble. In Weighted Voting, the classification models are not treated equally. Each model is associated with a coefficient (weight), usually proportional to its classification accuracy.

Let x be an instance and m_i, $i = 1 \ldots k$ a set of models that output a probability distribution $m_i(x, c_j)$ for each class c_j, $j = 1 \ldots n$. The output of the (weighted) voting method $y(x)$ for instance x is given by the following mathematical expression:

$$y(x) = \arg\max_{c_j} \sum_{i=1}^{k} w_i m_i(x, c_j),$$

where w_i is the weight of model i. In the simple case of voting (unweighted), the weights are all equal to one, that is, $w_i = 1, i = 1 \ldots k$.

3 Ensemble pruning via directed hill climbing

Hill climbing search greedily selects the next state to visit from the neighborhood of the current state. States, in our case, are the different subsets of the original ensemble $H = \{h_t, t = 1, 2, \ldots, T\}$ of T models. The neighborhood of a subset of models $S \subseteq H$ comprises those subsets that can be constructed by adding or removing one model from S. We focus on the directed version of hill climbing that traverses the search space from one end (empty

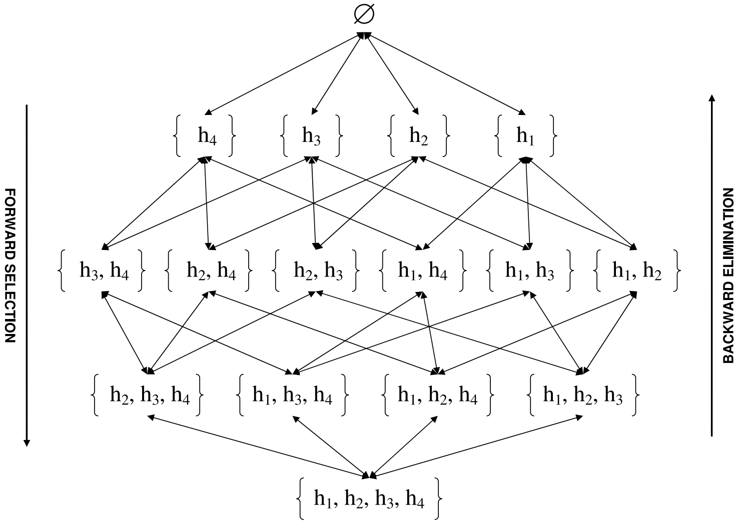

Fig. 1 An example of the search space of DHCEP methods for an ensemble of 4 models

set) to the other (complete ensemble). An example of the search space for an ensemble of four models is presented in Fig. 1.

The key design parameters that differentiate one DHCEP method from the other are (Tsoumakas et al. 2009): (a) the direction of search, (b) the measure and dataset used for evaluating the different branches of the search, and (c) the amount of pruning. The following sections discuss the different options for instantiating these parameters and the particular choices of existing methods.

3.1 Direction of search

Based on the direction of search we have two main categories of DHCEP methods: (a) *forward selection*, and (b) *backward elimination* (see Fig. 1).

In forward selection, the current classifier subset S is initialized to the empty set. The algorithm continues by iteratively adding to S the classifier $h_t \in H \setminus S$ that optimizes an evaluation function. In the past, this approach has been used in Fan et al. (2002), Martinez-Munoz and Suarez (2004), Caruana et al. (2004) and in the Reduced-Error Pruning with Backfitting (REPwB) method in Margineantu and Dietterich (1997).

In backward elimination, the current classifier subset S is initialized to the complete ensemble H and the algorithm continues by iteratively removing from S the classifier $h_t \in S$ that optimizes an evaluation function. In the past, this approach has been used in the AID thinning and concurrency thinning algorithms (Banfield et al. 2005).

In both cases, the traversal requires the evaluation of $\frac{T(T+1)}{2}$ subsets, leading to a time complexity of $O(T^2 g(T, N))$. The term $g(T, N)$ concerns the complexity of the evaluation function, which is linear with respect to N and ranges from constant to quadratic with respect to T, as we shall see in the following sections.

3.2 Evaluation dataset

The evaluation function scores the candidate subsets of models according to an evaluation measure that is calculated based on the predictions of its models on a set of data, which will be called the *pruning set*. The role of the pruning set can be performed by the training set, a separate validation set, or even a set of—naturally existing or artificially produced—instances with unknown value for the target variable. The pruning set will be denoted as $D = \{(\boldsymbol{x_i}, y_i), i = 1, 2, \ldots, N\}$, where $\boldsymbol{x_i}$ is a vector with feature values and y_i is the value of the target variable, which may be unknown.

In the past, the training set has been used for evaluation in Martinez-Munoz and Suarez (2004). This approach offers the benefit that plenty of data will be available for evaluation and training, but is susceptible to the danger of overfitting. Withholding a part of the training set for evaluation, has been used in the past in Caruana et al. (2004) and Banfield et al. (2005) and in the REPwB method in Margineantu and Dietterich (1997). This approach is less prone to overfitting, but reduces the amount of data available for training and evaluation compared to the previous approach. It sacrifices both the predictive performance of the ensemble's members and the quantity of the evaluation data for the sake of using unseen data in the evaluation. This method should probably be preferred over the previous one, when there is abundance of training data.

An alternative approach that has been used in Caruana et al. (2006), is based on k-fold cross-validation. For each fold an ensemble is created using the remaining folds as the training set. The same fold is used as the pruning set for models and subensembles of this ensemble. Finally, the evaluations are averaged across all folds. This approach is less prone to overfitting as the evaluation of models is based on data that were not used for their training and at the same time, the complete training dataset is used for evaluation. During testing the above approach works as follows: The k models that were trained using the same procedure (same algorithm, same subset, etc.) form a cross-validated model. When the cross-validated model makes a prediction for an instance, it averages the predictions of the individual models. An alternative testing strategy that we suggest for the above approach is to train an additional single model from the complete training set and use this single model during testing.

3.3 Evaluation measure

The main component that differentiates DHCEP methods is the evaluation measure. Evaluation measures can be grouped into two major categories: those that are based on *performance* and those on *diversity*.

3.3.1 Performance-based measures

The goal of performance-based measures is to find the model that maximizes the performance of the ensemble produced by adding (removing) a model to (from) the current ensemble. Their calculation depends on the method used for ensemble combination, which usually is voting. Accuracy was used as an evaluation measure in Margineantu and Dietterich (1997) and Fan et al. (2002), while Caruana et al. (2004) experimented with several metrics, including accuracy, root-mean-squared-error, mean cross-entropy, lift, precision/recall break-even point, precision/recall F-score, average precision and ROC area. Another measure is benefit, which is based on a cost model and has been used in Fan et al. (2002).

The calculation of performance-based metrics requires the decision of the ensemble on all examples of the pruning set. Therefore, the complexity of these measures is $O(|S|N)$. However, this complexity can be optimized to $O(N)$, if the predictions of the current ensemble are updated incrementally each time a classifier is added to/removed from it.

3.3.2 Diversity-based measures

It is generally accepted that an ensemble should contain diverse models in order to achieve high predictive performance. However, there is no clear definition of diversity, nor a single measure to calculate it. In their interesting study, Kuncheva and Whitaker (2003), could not reach a solid conclusion on how to utilize diversity for the production of effective classifier ensembles. In a more recent theoretical and experimental study on diversity measures (Tang et al. 2006), the authors reached the conclusion that diversity cannot be explicitly used for guiding the process of directed hill climbing methods. Yet, certain approaches have reported promising results (Martinez-Munoz and Suarez 2004; Banfield et al. 2005).

One issue that is worth mentioning here is how to calculate the diversity during the search in the space of ensemble subsets. For simplicity we consider the case of forward selection only. Let S be the current ensemble and $h_t \in H \setminus S$ a candidate classifier to add to the ensemble.

One could compare the diversities of subensembles $S' = S \cup h_t$ for all candidate $h_t \in H \setminus S$ and select the ensemble with the highest diversity. Any pairwise and non-pairwise diversity measure can be used for this purpose (Kuncheva and Whitaker 2003). Pairwise measures calculate the diversity between two models. The diversity of an ensemble of models can be calculated as the mean pairwise diversity of all models in the ensemble. The time complexity of this process is typically $O(|S'|^2N)$. Non-pairwise diversity measures can directly calculate the diversity of an ensemble of models and their time complexity is typically $O(|S'|N)$. A straightforward optimization can be performed in the case of pairwise diversity measures. Instead of calculating the sum of the pairwise diversity for every pair of classifiers in each candidate ensemble S', one can simply calculate the sum of the pairwise diversities only for the pairs that include the candidate classifier h_t. The sum of the rest of the pairs is equal for all candidate ensembles. The same optimization can be achieved in backward elimination too. This reduces their time complexity to $O(|S|N)$.

Several methods use a different approach to calculate diversity during the search. They use pairwise measures to compare the candidate classifier h_t with the current ensemble S, which is viewed as a single classifier that combines the decisions of its members with voting. This way they calculate the diversity between the current ensemble as a whole and the candidate classifier. Such an approach has time complexity $O(|S|N)$, which can be optimized to $O(N)$, if the predictions of the current ensemble are updated incrementally each time a classifier is added to/removed from it. However, these calculations do not take into account the decisions of individual models.

In the past, the widely known pairwise diversity measures *disagreement*, *double fault*, *Kohavi-Wolpert variance*, *inter-rater agreement*, *generalized diversity* and *difficulty* were used for DHCEP in Tang et al. (2006). *Complementariness* (Martinez-Munoz and Suarez 2004) and *concurrency* (Banfield et al. 2005) are two diversity measures designed specifically for ensemble pruning via directed hill climbing. We next introduce some additional notation to uniformly present these two methods.

We can distinguish four events concerning the decision of a classifier h and an ensemble of models S with respect to an example $(\boldsymbol{x_i}, y_i)$:

$$e_{\mathrm{tf}}(h, S, \boldsymbol{x_i}, y_i) : h(\boldsymbol{x_i}) = y_i \wedge S(\boldsymbol{x_i}) \neq y_i$$

$$e_{\mathrm{ft}}(h, S, \boldsymbol{x_i}, y_i) : h(\boldsymbol{x_i}) \neq y_i \wedge S(\boldsymbol{x_i}) = y_i$$

$$e_{\mathrm{tt}}(h, S, \boldsymbol{x_i}, y_i) : h(\boldsymbol{x_i}) = y_i \wedge S(\boldsymbol{x_i}) = y_i$$

$$e_{\mathrm{ff}}(h, S, \boldsymbol{x_i}, y_i) : h(\boldsymbol{x_i}) \neq y_i \wedge S(\boldsymbol{x_i}) \neq y_i$$

The *complementariness* of a model h with respect to an ensemble S and a pruning set D is calculated as follows:

$$COM_D(h,S)=\sum_{i=1}^{N} I(e_{\mathrm{tf}}(h,S,\boldsymbol{x_i},y_i)),$$

where $I(true)=1$, $I(false)=0$.

The complementariness of a model with respect to an ensemble is actually the number of examples of D that are classified correctly by the model and incorrectly by the ensemble. A pruning algorithm that uses the above measure, tries to add (remove) at each step the model that helps the current ensemble classify correctly the examples it gets wrong. Note that in the backward case the removed model is the one that minimizes the measure.

The *concurrency* of a model h with respect to an ensemble S and a pruning set D is calculated as follows:

$$CON_D(h,S)=\sum_{i=1}^{N}\big(2I(e_{\mathrm{tf}}(h,S,\boldsymbol{x_i},y_i))+I(e_{\mathrm{tt}}(h,S,\boldsymbol{x_i},y_i))-2I(e_{\mathrm{ff}}(h,S,\boldsymbol{x_i},y_i))\big)$$

This measure is similar to complementariness, with the difference that it takes into account two extra events and weights them. No specific argument is given for this particular choice of events and weights.

The *margin distance minimization* method (Martinez-Munoz and Suarez 2004) (also specifically designed for DHCEP) follows a different approach for calculating the diversity, which implicitly takes into account the decisions of individual models. For each classifier h_t an N-dimensional vector, c_t, is defined where each element $c_t(i)$ is equal to 1 if the t^{th} classifier classifies correctly example i of the pruning set, and -1 otherwise. The vector, C_S of the ensemble S is the average of the individual vectors c_t, $C_S=\frac{1}{|S|}\sum_{t=1}^{|S|}c_t$. When S classifies correctly all the instances the corresponding vector is in the first quadrant of the N-dimensional hyperplane. The objective is to reduce the Euclidean distance, $d(o,C_S)$, of the current ensemble C_S from a predefined vector with the same components, $o_i=p, i=1,\dots,N, 0<p<1$, placed in the first quadrant of the N-dimensional hyperplane. The value of p is usually between 0.05 and 0.25. The proposed *margin* diversity measure, $MAR_D(h_t,S)$, of a classifier h_t with respect to an ensemble S and a pruning set D is calculated as follows:

$$MAR_D(h_t,S)=d\left(o,\frac{1}{|S|+1}(c_t+C_S)\right)$$

3.4 Amount of pruning

Another issue that pertains to almost all ensemble pruning methods concerns the size of the final ensemble. Two main approaches are followed with respect to this issue: (a) use a fixed user-specified amount or percentage of models, and (b) dynamically select the size based on the predictive performance of candidate ensembles of different size. In the second case the predictive performance of the ensembles encountered during the complete search process from the one end of the search space to the other is recorded and the ensemble with the best performance is selected. If the goal of pruning is to improve efficiency, then the former approach can be used in order to achieve the desired number of models, which may be dictated by constraints (memory and speed) in the application environment. If the goal of pruning is to improve performance, then the latter approach can be used, as it is more flexible and can sacrifice efficiency for effectiveness.

4 The proposed measure

We here propose a new measure starting from the common notation that was introduced in Sect. 3.3 to describe the complementariness and concurrency measures. For simplicity of presentation we slightly abuse the notation by dropping symbols $\boldsymbol{x_i}$ and y_i.

First of all we note that complementariness is based on event $e_{\mathrm{tf}}(h, S)$ only. However, the rest of the events are also plausible indicators of the utility of a candidate classifier h with respect to an ensemble S. For example, event $e_{\mathrm{tt}}(h, S)$ should also add to the utility of h, though potentially not as much as $e_{\mathrm{tf}}(h, S)$.

This is reflected in the concurrency measure, which explicitly takes into account three of the events (implicitly the fourth one as well, since $e_{\mathrm{ft}}(h, S) = 1 - e_{\mathrm{tt}}(h, S) - e_{\mathrm{ff}}(h, S) - e_{\mathrm{tf}}(h, S)$), each with a different weight. Concurrency is considering positively the event $e_{\mathrm{tf}}(h, S)$, negatively the event $e_{\mathrm{ff}}(h, S)$, positively with half the weight of the previous events the event $e_{\mathrm{tt}}(h, S)$ and neutrally the event $e_{\mathrm{ft}}(h, S)$. The contribution of each of the events to the heuristic is rather ad-hoc, as there is no theoretical justification for the particular choice of weights.

More important is the fact that, as briefly mentioned in the previous section, concurrency, complementariness and all other diversity measures that are computed based on the decision of the candidate model and the decision of the ensemble as a whole, are agnostic of the decisions of the individual models of the ensemble. We consider this a major limitation of these measures, and justify our claim with an example. Consider two members of the pruning set $(\boldsymbol{x_i}, y_i)$ and $(\boldsymbol{x_j}, y_j)$ that are wrongly classified by candidate classifier h. The first one is wrongly classified by 49% of the members of the ensemble S, while the second one by just 10%. In both cases event $e_{\mathrm{ft}}(h, S)$ is the true one. Should these examples contribute the same value to the measure? The rational answer is no. In the first case, the uncertainty of the ensemble is high, while in the second it is low. In a forward selection scenario, the probability that the ensemble will wrongly classify example i in the future, if it adds h to S, is far greater compared to the probability that it will wrongly classify j.

The above issues motivated us to propose a new measure that takes into account the uncertainty of the ensemble's decision, and at the same time has clear and justified semantics. The following quantities are introduced to allow its definition: NT_i, which denotes the proportion of models in the current ensemble S that classify example $(\boldsymbol{x_i}, y_i)$ correctly, and $NF_i = 1 - NT_i$, which denotes the proportion of models in S that classify it incorrectly. The proposed measure, dubbed Uncertainty Weighted Accuracy (UWA), is defined as follows:

$$\begin{aligned} UWA(h, S) = \sum_{i=1}^{N} \big(& I(e_{\mathrm{tf}}(h, S, \boldsymbol{x_i}, y_i))NT_i - I(e_{\mathrm{ft}}(h, S, \boldsymbol{x_i}, y_i))NF_i \\ & + I(e_{\mathrm{tt}}(h, S, \boldsymbol{x_i}, y_i))NF_i - I(e_{\mathrm{ff}}(h, S, \boldsymbol{x_i}, y_i))NT_i \big) \end{aligned}$$

First of all, note that events e_{tf} and e_{tt} increase the metric, because the candidate classifier is correct, while events e_{ft} and e_{ff} decrease it, as the candidate classifier is incorrect. The strength of increase/decrease depends on the uncertainty of the ensemble's decision. If the current ensemble S is incorrect, then the reward/penalty is multiplied by the proportion of correct models in S. On the other hand, if S is correct, then the reward/penalty is multiplied by the proportion of incorrect models in S.

This complex, at first sight, weighting scheme, actually represents a simple rule: examples for which the ensemble's decision is highly uncertain should influence the metric stronger, while examples where most of the ensemble's members agree should not influence the metric a lot. The rationale of this rule is the following: When most members of the ensemble agree, then this is either a very easy (if the ensemble is correct), or a very hard (if it is wrong) example. Rewarding a candidate classifier that correctly classifies an easy example is of no real value, as is penalizing it for erring on a very hard example. On the other hand, when the ensemble is marginally correct or incorrect, then the decision of the candidate classifier is more important, as it may correct an incorrect decision of the ensemble, or the other way round.

To see exactly how the measure represents the above rule, let's examine each specific event separately, in a forward selection scenario.

- In event e_{tf}, the addition of a correct classifier when the ensemble is wrong contributes a reward proportional to the number of correct classifiers in that ensemble. The rationale is that if the number of correct classifiers is small, then correct classification of this example is hard to achieve and thus the addition of this classifier will not have an important impact on the ensemble's performance. On the other hand, if the number of correct classifiers is large, then the example is marginal and thus the impact of the addition of this classifier is significant.
- In event e_{ft}, the addition of an erring classifier when the ensemble is correct contributes a penalty proportional to the number of erring classifiers in that ensemble. The rationale is that if the number of erring classifiers is small, then the addition of another erring classifier will not influence the ensemble significantly, while if the number of erring classifiers is large, then the example is marginal and thus the classifier could change the ensemble's decision from correct to wrong.
- In event e_{tt}, the addition of a correct classifier when the ensemble is correct contributes a reward proportional to the number of erring classifiers in that ensemble. The rationale is that if the number of erring classifiers is small, then the addition of a correct classifier is not really very useful. The higher the number of erring classifiers the more important is the addition of this candidate classifier.
- In event e_{ff}, the addition of an erring classifier when the ensemble is wrong contributes a penalty proportional to the number of correct classifiers in that ensemble. The rationale is that if the number of correct classifiers is small, then the addition of another erring classifier will not influence the ensemble significantly, as this is a hard to correctly classify example. If the number of correct classifiers is large, then the example is marginal and thus the classifier has a negative effect to the ensemble as it moves it further away from the margin.

We next give a concrete example, in a forward selection scenario, in order to clarify how the proposed measure works. Consider an ensemble that contains 10 classifiers h_1 to h_{10} and two candidate classifiers h'_1 and h'_2. Tables 1 and 2 show whether each of these classifiers is correct or incorrect for a pruning set containing 5 examples $\boldsymbol{x}_1$ to $\boldsymbol{x}_5$. A correct decision is indicated by a plus (+), and a wrong decision by a minus (−). For the ensemble, example $\boldsymbol{x}_1$ is a difficult one, $\boldsymbol{x}_2$ an easy, while the rest of the examples are marginal. The last column of Table 2 shows the value of the proposed measure.

The events that characterize h'_1 in relation to examples $\boldsymbol{x}_1$ to $\boldsymbol{x}_5$ are e_{tf}, e_{ft}, e_{tt}, e_{ft} and e_{ff} respectively. Consequently, the contributions to UWA are $NT_1 - NF_2 + NF_3 - NF_4 - NT_5$, giving $0.2 - 0.2 + 0.6 - 0.4 - 0.6 = -0.4$. The reward from correctly classifying $\boldsymbol{x}_1$ is small, as it is a very easy example. The classifier is penalized stronger for the incorrect classification of marginal examples $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$, compared to the hard example $\boldsymbol{x}_2$. The events

Table 1 An ensemble of 10 classifiers and the decision of its models for the 5 instances of the pruning set

	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$
h_1	$-$	$+$	$-$	$+$	$-$
h_2	$-$	$+$	$+$	$+$	$-$
h_3	$-$	$+$	$-$	$+$	$-$
h_4	$-$	$+$	$-$	$-$	$+$
h_5	$-$	$+$	$+$	$-$	$-$
h_6	$+$	$+$	$+$	$-$	$+$
h_7	$-$	$+$	$+$	$+$	$+$
h_8	$+$	$+$	$-$	$-$	$-$
h_9	$-$	$-$	$+$	$+$	$-$
h_{10}	$-$	$-$	$+$	$+$	$+$
NT	0.2	0.8	0.6	0.6	0.4
NF	0.8	0.2	0.4	0.4	0.6

Table 2 Two candidate classifiers, their decisions for the 5 instances of the pruning set and the value of the proposed measure

	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	UWA
h'_1	$+$	$-$	$+$	$-$	$-$	−0.4
h'_2	$-$	$-$	$-$	$+$	$+$	0

that characterize h'_2 in relation to examples $\boldsymbol{x}_1$ to $\boldsymbol{x}_5$ are e_{ff}, e_{ft}, e_{ft}, e_{tt} and e_{tf} respectively. Consequently, the contributions to UWA are $-NT_1 - NF_2 - NF_3 + NF_4 + NT_5$, giving $-0.2 - 0.2 - 0.6 + 0.4 + 0.6 = 0$. The reward from correctly classifying the marginal examples $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ is large, as is the penalty from incorrect classification of marginal example $\boldsymbol{x}_3$. In contrast, the incorrect classification of the very hard and very easy examples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ incurs a small penalty. Despite that both candidate classifiers make the same number of errors, overall h'_2 is more beneficial to the current ensemble, which is reflected in a higher value of UWA. It is this candidate classifier, that would be added to the ensemble in this forward selection scenario that we examined.

Concluding this section, we explain how the name of the proposed measure was conceived. If we dropped the weights from the measure, then it would be proportional to the accuracy of the candidate classifier, since it would give a reward (penalty) of 1 whenever there is a correct (incorrect) candidate classifier decision, irrespectively of the ensemble's decision. The weights correspond to the ensemble's uncertainty, hence *uncertainty weighted accuracy*.

5 Experimental setup

This section presents information about the datasets that were used for conducting the experiments, the classifiers that comprise the initial ensemble, the ensemble pruning methods that participate in the comparison, as well as the experimentation methodology that was followed.

Table 3 Details of data sets: folder in UCI server, number of instances, classes, continuous and discrete attributes, percentage of missing values

id	UCI folder	Inst	Cls	Cnt	Dsc	MV(%)
d1	Anneal	798	6	9	29	0.00
d2	Balance-scale	625	3	4	0	0.00
d3	Breast-w	699	2	0	2	0.00
d4	Car	1728	4	0	6	0.00
d5	Cmc	1473	3	2	7	0.00
d6	Colic	368	2	7	15	23.80
d7	Credit-g	1000	2	7	13	0.00
d8	Dermatology	366	6	1	33	0.00
d9	Ecoli	336	8	7	0	0.00
d10	Haberman	306	2	3	0	0.00
d11	Heart-h	294	5	6	7	20.46
d12	Heart-statlog	270	2	13	0	0.00
d13	Hill	607	2	100	0	0.00
d14	Hypothyroid	3772	4	7	30	5.4
d15	Ionosphere	351	2	34	0	0.00
d16	Kr-vs-kp	3196	2	0	36	0.00
d17	Mammographic	962	2	5	0	3.3
d18	Mfeat-morphological	2000	10	6	0	0.00
d19	Page-blocks	5473	5	10	0	0.00
d20	Primary-tumor	339	2	0	17	0.00
d21	Segment	2310	7	19	0	0.00
d22	Sick	3772	2	7	23	5.40
d23	Sonar	195	2	60	0	0.00
d24	Soybean	683	19	0	35	0.00
d25	Spambase	4601	2	57	0	0.00
d26	Tic-tac-toe	958	2	0	9	0.00
d27	Vehicle	946	4	18	0	0.00
d28	Vote	435	2	0	16	5.63
d29	Vowel	990	11	3	10	0.00
d30	Waveform-5000	5000	3	21	0	0.00

The source code we developed for conducting the experiments along with a package for performing statistical tests on multiple datasets, can be found at the following URL: http://mlkd.csd.auth.gr/ensemblepruning.html.

5.1 Datasets

We experimented on 30 data sets from the UCI Machine Learning repository (Asuncion and Newman 2007). Table 3 presents the details of these data sets (folder in UCI server, number of instances, classes, continuous and discrete attributes, percentage of missing values). We avoided using datasets with a very small number of examples, so that an adequate amount of data is available for training, evaluation and testing.

5.2 Ensemble construction

We constructed a heterogeneous ensemble of 200 models, by running different learning algorithms with different parameters on the training set. The WEKA machine learning library (Witten and Frank 2005) was used as the source of learning algorithms. We trained 40 multilayer perceptrons (MLPs), 60 k Nearest Neighbors (kNNs), 80 support vector machines (SVMs) and 20 decision trees (DT) using the C4.5 algorithm. The different parameters used to train the algorithms were the following (default values were used for the rest of the parameters):

- MLPs: we used 5 values for the nodes in the hidden layer $\{1, 2, 4, 8, 16\}$, 4 values for the momentum term $\{0.0, 0.2, 0.5, 0.9\}$ and 2 values for the learning rate $\{0.3, 0.6\}$.
- kNNs: we used 20 values for k distributed evenly between 1 and the plurality of the training instances. We also used 3 weighting methods: no-weighting, inverse-weighting and similarity-weighting.
- SVMs: we used 8 values for the complexity parameter $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 100\}$, and 10 different kernels. We used 2 polynomial kernels (of degree 2 and 3) and 8 radial kernels (gamma $\in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2\}$).
- Decision trees: We constructed 10 trees using postpruning with 5 values for the confidence factor $\{0.1, 0.2, 0.3, 0.5\ \}$ and 2 values for Laplace smoothing {true, false}, 8 trees using reduced error pruning with 4 values for the number of folds $\{2, 3, 4, 5\}$ and 2 values for Laplace smoothing {true, false}, and 2 unpruned trees using 2 values for the minimum objects per leaf $\{2, 3\}$.

5.3 Ensemble pruning methods

We compare 16 instantiations of the general DHCEP algorithm, that arise by using all combinations of 4 different values for the evaluation measure parameter and 2 different values for each of the evaluation dataset and search direction parameters. As far as the evaluation measure parameter is concerned, we used the proposed measure, UWA, and the following performance and diversity based measures: Accuracy (ACC) (Caruana et al. 2004), Complementariness (COM) (Martinez-Munoz and Suarez 2004) and Concurrency Thinning (CON) (Banfield et al. 2005). The search was performed in both the forward (F) and the backward (B) directions. The pruning set was instantiated to: (a) the training set (T), which means that all available data are used for both building the models and pruning the ensemble, and (b) a separate validation set (V).

In addition to the above 16 methods we implemented two methods that do not use any diversity measures and prune the ensemble according to a fixed order. The first one is called Random Ordering (RO), which randomly orders the classifiers and the second one is called Greedy Ordering (GO), which orders the classifiers according to their accuracy on the pruning set. Additionally, we implemented two baseline methods, corresponding to two extreme pruning scenarios. The first one selects the best single model (BSM) in the ensemble, according to the performance of the models on the pruning set, while the second one retains all models of the ensemble (ALL). Note that for RO, GO, BSM and ALL, the direction parameter has no meaning and is neglected. Additionally, note that ALL does not require a pruning set, as it does not include any selection process. For this reason, the performance of ALL is calculated only for the case where all available training data are used for training the models of the ensemble.

Voting was used for model combination in all aforementioned algorithms. Additionally, all the algorithms follow the dynamic approach in Caruana et al. (2004), which selects the

Table 4 Acronym, search direction, evaluation dataset and evaluation measure for the different DHCEP, ordering and baseline pruning methods

Acronym	Search direction	Evaluation dataset	Evaluation measure
FTACC	Forward	Training set	ACCuracy
FTCON	Forward	Training set	CONcurrency
FTCOM	Forward	Training set	COMplementariness
FTUWA	Forward	Training set	Uncertainty Weighted Accuracy
FVACC	Forward	Validation set	ACCuracy
FVCON	Forward	Validation set	CONcurrency
FVCOM	Forward	Validation set	COMplementariness
FVUWA	Forward	Validation set	Uncertainty Weighted Accuracy
BTACC	Backward	Training set	ACCuracy
BTCON	Backward	Training set	CONcurrency
BTCOM	Backward	Training set	COMplementariness
BTUWA	Backward	Training set	Uncertainty Weighted Accuracy
BVACC	Backward	Validation set	ACCuracy
BVCON	Backward	Validation set	CONcurrency
BVCOM	Backward	Validation set	COMplementariness
BVUWA	Backward	Validation set	Uncertainty Weighted Accuracy
TGO	–	Training set	–
TRO	–	Training set	–
TBSM	–	Training set	–
VGO	–	Validation set	–
VRO	–	Validation set	–
VBSM	–	Validation set	–

ensemble, during the pruning procedure, with the highest accuracy on the pruning set, instead of using an arbitrary percentage of models. Table 4 shows the acronyms that will be used in the rest of this paper for the different instantiations of the parameters of the general DHCEP algorithm, RO, GO and BSM.

5.4 Methodology

Initially, each dataset is split into three disjunctive parts: D_1, D_2 and D_3, consisting of 60%, 20% and 20% respectively. In the case where a separate validation set is used for pruning, D_1 is used for training the models and D_2 for performing the pruning procedure. In the other case, $D_1 \cup D_2$ is used for both training and pruning. D_3 is always used solely for testing the methods.

The experiment described in this section is performed 10 times for each dataset using a different randomized ordering of its examples. All reported results are averages over these 10 repetitions.

6 Results and discussion

According to Demsar (2006) the appropriate way to compare the effectiveness of multiple algorithms on multiple datasets is based on their average rank across all datasets. On each

Table 5 Average rank, ensemble size and type of the selected models of each method across all datasets

Method	Avg. rank	Avg. size	MLP	*k*NN	SVM	DT
FVUWA	3.96	7.5	2.0	0.9	2.8	1.8
FVCON	6.51	6.8	1.8	1.3	2.4	1.3
FVCOM	7.46	7.3	2.3	2.2	1.8	1.0
FVACC	7.63	7.7	2.8	2.3	1.9	0.7
BVUWA	6.25	32.6	9.5	6.0	11.3	5.8
BVCON	6.58	29.7	8.0	5.9	10.5	5.3
BVCOM	9.65	36.6	10.5	8.7	12.5	4.9
BVACC	12.86	46.5	18.1	15.5	10.4	2.6
FTUWA	18.45	1.69	0.05	0.61	0.98	0.05
FTCON	18.36	1.4	0.03	0.68	0.76	0.01
FTCOM	19.45	1.5	0.06	0.53	0.86	0.07
FTACC	18.11	1.3	0.05	0.51	0.8	0.04
BTUWA	12.65	43.6	7.3	15.9	14.6	5.8
BTCON	13.86	41.6	7.0	15.2	14.1	5.3
BTCOM	14.08	51.8	15.4	18.8	14.7	2.8
BTACC	15.31	29.8	10.6	12.8	5.4	1.1
VRO	12.58	86.1	18.3	26.2	32.1	9.6
VGO	8.80	30.0	8.1	7.6	10.0	4.3
VBSM	7.08	1.0	0.31	0.06	0.48	0.14
TRO	12.90	84.2	14.3	30.6	30.8	8.4
TGO	12.40	52.1	9.1	19.5	17.5	5.8
TBSM	17.58	1.0	0.12	0.79	0.08	0.0
ALL	15.36	200.0	40.0	60.0	80.0	20.0

dataset, the algorithm with the best performance gets rank 1.0, the one with the second best performance gets rank 2.0 and so on. In case two or more algorithms tie, they all receive the average of the ranks that correspond to them.

Table 5 shows the average rank (based on classification accuracy) along with the average size and composition (type of models) of the pruned ensemble, for each method participating in the experiments. DHCEP methods are grouped first by pruning set, then by search direction and finally sorted by evaluation measure. The table continues with RO, GO and BSM grouped by pruning set, and ends with the ALL method. Detailed tables with the classification accuracy, rank and ensemble size of all methods in all datasets can be found in Appendix (Tables 6–9).

We first study the evaluation dataset parameter and its relation to the effectiveness of the ensemble pruning methods. We observe that using a separate validation set leads to better results than using the training set, for all methods. In order to investigate whether the differences in accuracy are statistically significant, we conducted 11 Wilcoxon signed rank tests (Wilcoxon 1945), one for each pair of methods that differ in terms of the evaluation dataset. At a confidence level of 95% all the tests reported significant differences in favor of using a separate validation set. This shows that using unseen data to guide the pruning process is very important, despite the fact that it comes at the expense of available training data for the models of the ensemble. When the training set is used as the pruning set, the pruning process is based on model predictions for data that are known to the models, and is therefore biased towards model subsets that overfit the training data. The negative effect of overfitting is stronger for methods that select a small number of models, such as the baseline BSM method and the DHCEP methods that search in the forward direction.

We next study the effect of the search direction parameter. In light of the results concerning the evaluation dataset, we exclude methods that use the training set from this discussion. We first notice that forward selection leads to better results compared to backward elimination for all evaluation measures. In order to investigate whether the differences in accuracy are statistically significant, we conducted 4 Wilcoxon signed rank tests, one for each pair of DHCEP methods that use a separate validation set for pruning and differ in terms of the search direction. At a confidence level of 95% all tests show significant differences in favor of the forward direction, apart from the pair of BVCON and FVCON.

As far as the size of the pruned ensemble is concerned, we observe that those DHCEP methods that search in the forward direction tend to produce much smaller ensembles than those that search in the backward direction. More specifically, the FVxxx methods achieve an average reduction of 96.33% of the initial ensemble, while the BVxxx ones achieve an average reduction of 81.82%. Note that this trend also holds for DHCEP methods that use the training set to guide the pruning process.

Based on these results, the recommended search direction for DHCEP methods is the forward one. Even though it does not bring a statistically significant benefit for one of the measures (CON), it manages to reduce the initial ensemble substantially more compared to the backward direction. Note that this conclusion assumes the use of a separate validation set for evaluation.

This is an interesting outcome and can be explained by considering the fact that DHCEP methods are based on a comparison of the decisions of a candidate model and an ensemble of models. In the backward direction, DHCEP methods remove those models that are not helpful compared to the current ensemble according to an evaluation measure. However, the current ensemble is initially suboptimal, as it contains all models, both good and bad ones. Therefore, there is a high probability that a good model will be removed in the beginning, just because it does not improve the current suboptimal ensemble. On the other hand, in the forward direction the initial ensemble is empty, and is progressively expanded according to the evaluation measure, so it always contain good models. This explains both the higher performance and the smaller size of the DHCEP methods that search in the forward direction.

We continue the analysis of the results with a study of the evaluation measure parameter. As before, we exclude methods that use the training set from this discussion. We first notice that the proposed measure, UWA leads to the two best overall results, independently of the search direction. We next proceed to an investigation of whether the proposed measure is significantly better compared to the rest. Taking into account the conclusion concerning the search direction parameter, we performed a Wilcoxon signed rank test between FVUWA and each of FVCON, FVCOM and FVACC. We also performed a Wilcoxon signed rank test between FVUWA and each of VGO, VRO, VBSM and ALL. At a confidence level of 95% all the tests show that FVUWA performs significantly better than its competitors. This shows that taking into account the uncertainty of the ensemble's decision can lead to substantially better results in terms of accuracy. The size of the pruned ensemble is approximately the same as that returned by DHCEP methods using other evaluation measures.

It is also interesting to look at the composition of the pruned ensembles, as shown in the last four columns of Table 5. Focusing on DHCEP methods that search in the forward search direction and use a separate validation set for evaluation, we notice that they produce ensembles with a balanced mixture of different types of models. Different types of models

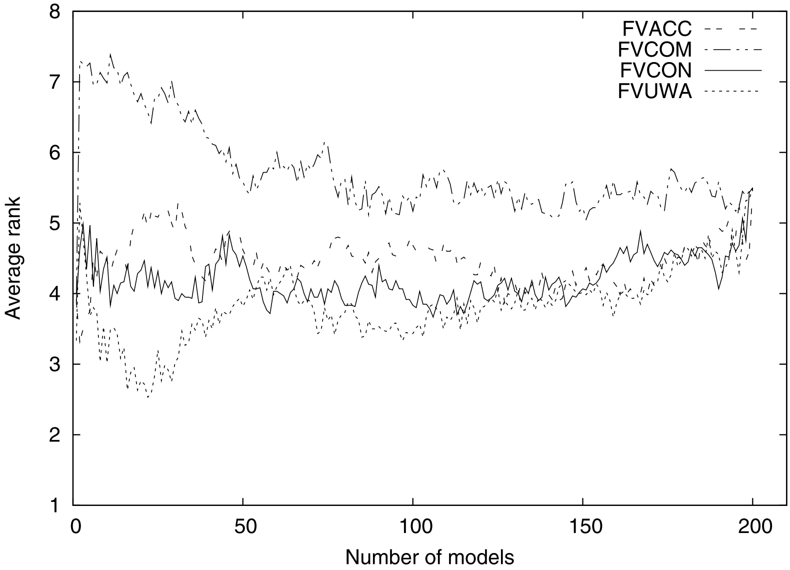

Fig. 2 Performance of the FVxxx methods during the pruning phase. The ranking procedure is performed for the different sizes of the ensemble between 1 and 200

lead to more diverse ensembles, and as a result more accurate ensembles. It is also interesting to look into how the composition of models affects the performance of these 4 methods. It seems that more SVMs and MLPs and less DTs and *k*NN models, lead to better results. MLPs and SVMs are high performance classifiers and thus they dominate the ensemble. Additionally, the selection of DTs and *k*NNs seems to add to the overall diversity, and such models should be present in the pruned ensemble.

So far, we have looked at how methods behave under the setting that the percentage of pruning is automatically determined by the methods themselves, which as we have discussed in Sect. 3 is suitable when the main motivation of the pruning process is to increase the prediction effectiveness. However, if we were primarily interested in assessing the efficiency of the methods, then we should evaluate their accuracy under different percentages of pruning. Still, we should note that the automated pruning methods already achieve a remarkable pruning level, thus they meet the goal of efficiency too.

Figure 2 depicts the average rank of the FVxxx methods during the pruning process. The curves are produced by calculating the ranks for the different sizes of the ensemble between 1 and 200. It is interesting to note that the method using the proposed measure (FVUWA) is the best one for the largest part of the graph. Especially in the area between 2 and 40 models, it clearly outperforms its rivals. These findings show that it can be used for guiding a pruning process, in situations where computational and storage savings is the dominant motivation. Another interesting observation is that although FVCOM has a better average rank compared to FVACC based on the automatic selection of the amount of pruning, it is much worse than FVACC under the same level of pruning for the largest part of the graph.

7 Conclusions and future work

This paper presented a new measure for DHCEP, called Uncertainty Weighted Accuracy (UWA), which takes into account the uncertainty of the decisions of the current ensemble. We compared UWA against state-of-the-art measures using different values for the parameters of search direction and evaluation dataset on ensembles of heterogeneous models. The empirical comparison was carried out on 30 datasets and included 4 additional baseline ensemble pruning methods. The results show that the proposed measure leads to significantly better accuracy results compared to its rivals and it also manages to reduce substantially the size of the original ensemble and thus to minimize the computational complexity.

Several interesting conclusions came up. To begin with, the evaluation dataset parameter plays an important role on the performance of the DHCEP methods. The use of a separate set leads to significantly better results than using all the available data. Also, the direction of search is another important parameter that influences the performance of DHCEP methods. The results showed that the forward direction helps to significantly improve the accuracy.

One interesting future work direction concerns the composition of the pool of models that constitute the initial ensemble. It is interesting to investigate which parameters of the algorithms have proved effective or not (for example a neural network with 10 hidden nodes) and to use this information in order to substitute the specific models with other more effective ones. Another related interesting direction concerns the development of a method that can train and add models in the ensemble incrementally in an active learning fashion. In other words to grow the ensemble dynamically by actively selecting the next most appropriate model to train and/or perhaps dynamically removing inappropriate models that were previously added. This will save the computational costs of training a large number of models from the beginning.

Acknowledgements We would like to sincerely thank the anonymous reviewers for their helpful and to-the-point comments.

Appendix

Table 6 Accuracies of each algorithm on each dataset

id	BVACC	BVCOM	BVCON	BVUWA	BTACC	BTCOM	BTCON	BTUWA	FVACC	FVCOM	FVCON
d1	96.65 ± 1.56	98.21 ± 0.78	98.27 ± 0.85	98.1 ± 0.88	98.55 ± 0.99	94.92 ± 1.74	98.21 ± 0.82	97.99 ± 1.06	98.55 ± 1.03	98.49 ± 0.95	98.49 ± 0.95
d2	90.16 ± 4.89	96.24 ± 2.23	96.96 ± 1.92	96.48 ± 2.0	87.92 ± 3.36	89.6 ± 3.2	89.84 ± 3.27	90.16 ± 2.82	96.8 ± 2.0	96.96 ± 2.06	97.12 ± 2.48
d3	96.12 ± 2.07	95.83 ± 1.72	96.69 ± 1.32	96.62 ± 1.13	96.69 ± 1.37	94.03 ± 2.23	95.83 ± 1.91	95.54 ± 1.76	96.19 ± 2.35	96.4 ± 1.83	96.69 ± 1.37
d4	97.25 ± 0.99	98.32 ± 0.9	99.04 ± 0.92	98.9 ± 0.62	90.96 ± 3.1	92.87 ± 1.99	96.49 ± 0.89	96.41 ± 0.99	99.25 ± 0.95	99.19 ± 0.92	99.42 ± 0.49
d5	51.77 ± 2.78	52.04 ± 2.53	53.57 ± 3.61	53.78 ± 1.91	48.2 ± 2.89	45.88 ± 3.61	48.2 ± 1.49	48.44 ± 1.31	53.74 ± 2.57	52.41 ± 3.06	52.65 ± 2.99
d6	82.74 ± 3.55	83.56 ± 3.93	83.15 ± 4.92	83.97 ± 4.75	79.59 ± 2.99	82.33 ± 3.9	79.18 ± 3.97	79.86 ± 4.09	83.29 ± 4.6	83.15 ± 3.98	82.88 ± 4.93
d7	74.2 ± 3.13	74.3 ± 2.72	74.8 ± 3.17	74.75 ± 3.29	75.45 ± 2.29	75.55 ± 2.25	75.6 ± 2.51	75.35 ± 2.57	74.3 ± 3.7	75.05 ± 3.57	74.85 ± 3.56
d8	94.79 ± 2.95	96.71 ± 2.26	97.67 ± 1.59	97.12 ± 1.2	95.34 ± 3.11	95.48 ± 2.96	96.3 ± 2.67	96.3 ± 2.67	96.58 ± 1.96	96.71 ± 2.35	97.53 ± 1.56
d9	82.54 ± 3.23	83.58 ± 4.82	85.97 ± 5.98	84.48 ± 5.77	81.49 ± 3.87	82.84 ± 2.14	82.99 ± 4.05	82.99 ± 3.67	85.52 ± 4.93	85.22 ± 5.0	85.67 ± 6.26
d10	71.97 ± 5.43	71.97 ± 4.91	71.48 ± 5.64	71.97 ± 5.43	71.64 ± 6.28	71.8 ± 5.88	71.15 ± 6.34	70.98 ± 6.47	71.8 ± 6.68	71.15 ± 5.9	70.98 ± 6.37
d11	83.45 ± 5.34	82.24 ± 5.14	83.28 ± 4.95	83.79 ± 5.15	77.59 ± 4.53	79.31 ± 3.81	78.1 ± 3.36	78.97 ± 3.02	82.93 ± 4.56	83.28 ± 4.81	82.07 ± 5.28
d12	81.48 ± 5.31	81.48 ± 5.09	82.04 ± 5.17	83.33 ± 5.31	82.04 ± 5.09	81.48 ± 4.86	80.93 ± 3.27	80.74 ± 3.29	80.93 ± 4.62	82.59 ± 4.11	81.11 ± 4.6
d13	51.16 ± 5.61	59.09 ± 8.38	59.67 ± 8.67	57.27 ± 7.08	53.31 ± 4.18	53.88 ± 4.64	54.13 ± 4.46	54.13 ± 4.98	57.93 ± 8.41	61.16 ± 8.82	59.83 ± 8.1
d14	94.06 ± 1.18	99.36 ± 0.2	99.42 ± 0.17	99.4 ± 0.19	90.72 ± 1.27	91.91 ± 2.43	91.09 ± 1.7	91.67 ± 2.22	99.4 ± 0.16	99.46 ± 0.18	99.43 ± 0.14
d15	89.14 ± 5.14	92.57 ± 2.76	93.71 ± 3.17	93.86 ± 3.16	87.57 ± 3.02	89.57 ± 4.04	92.57 ± 2.68	92.43 ± 2.86	93.86 ± 2.34	93.86 ± 3.09	93.86 ± 3.37
d16	99.0 ± 0.5	99.23 ± 0.39	99.36 ± 0.41	99.36 ± 0.49	98.26 ± 0.87	99.53 ± 0.22	99.58 ± 0.25	99.58 ± 0.25	99.33 ± 0.48	99.28 ± 0.48	99.31 ± 0.39
d17	79.79 ± 2.79	81.2 ± 2.42	81.82 ± 2.78	80.99 ± 2.84	76.46 ± 2.01	75.78 ± 1.63	76.67 ± 2.42	76.93 ± 2.33	81.56 ± 3.5	80.73 ± 2.69	81.51 ± 3.52
d18	73.15 ± 1.8	73.5 ± 1.44	73.45 ± 1.46	73.93 ± 1.23	65.83 ± 1.21	65.7 ± 1.23	65.8 ± 1.18	65.67 ± 1.18	73.4 ± 1.62	73.65 ± 1.3	73.6 ± 1.31
d19	95.86 ± 0.94	96.8 ± 0.59	96.77 ± 0.58	96.8 ± 0.44	95.74 ± 0.76	95.74 ± 0.77	95.82 ± 0.77	95.77 ± 0.76	97.06 ± 0.49	96.85 ± 0.54	97.01 ± 0.52
d20	42.99 ± 5.96	43.28 ± 4.51	43.88 ± 3.46	44.18 ± 5.41	40.45 ± 3.1	41.64 ± 3.69	41.19 ± 3.46	41.04 ± 3.74	43.43 ± 4.19	42.69 ± 4.78	45.52 ± 4.94
d21	96.1 ± 1.24	96.71 ± 1.1	97.12 ± 0.89	97.1 ± 0.78	96.99 ± 1.11	96.56 ± 1.23	97.32 ± 0.94	97.51 ± 1.01	96.84 ± 1.17	96.9 ± 0.68	97.1 ± 0.8
d22	97.2 ± 0.61	98.54 ± 0.44	98.41 ± 0.52	98.45 ± 0.53	96.1 ± 0.73	97.41 ± 1.21	96.74 ± 0.8	96.84 ± 0.79	98.51 ± 0.44	98.58 ± 0.56	98.51 ± 0.55
d23	81.22 ± 7.09	83.9 ± 5.89	83.41 ± 4.85	85.61 ± 4.8	87.8 ± 3.64	84.63 ± 4.46	85.61 ± 3.72	86.1 ± 3.82	84.15 ± 7.82	83.66 ± 5.64	84.63 ± 5.4

Table 6 (*Continued*)

id	BVACC	BVCOM	BVCON	BVUWA	BTACC	BTCOM	BTCON	BTUWA	FVACC	FVCOM	FVCON
d24	89.04 ± 4.28	90.96 ± 2.35	91.4 ± 2.25	91.4 ± 2.5	87.94 ± 4.07	85.74 ± 4.95	91.32 ± 1.95	89.85 ± 2.4	91.69 ± 2.4	91.47 ± 2.28	91.54 ± 2.14
d25	93.89 ± 0.68	94.27 ± 0.44	94.55 ± 0.51	93.98 ± 1.01	90.88 ± 0.81	91.26 ± 0.67	90.88 ± 0.81	91.91 ± 1.45	94.37 ± 0.79	94.14 ± 0.86	94.72 ± 0.61
d26	97.33 ± 0.76	97.85 ± 0.8	97.75 ± 0.86	97.75 ± 0.99	97.91 ± 0.92	97.96 ± 0.84	97.96 ± 0.84	97.96 ± 0.84	97.54 ± 0.82	97.64 ± 0.93	97.7 ± 1.02
d27	79.64 ± 3.56	83.91 ± 1.82	84.56 ± 3.1	84.67 ± 3.04	73.85 ± 3.3	78.11 ± 3.65	75.44 ± 3.91	77.1 ± 3.97	83.61 ± 2.8	83.02 ± 3.55	84.2 ± 3.22
d28	96.09 ± 2.04	95.4 ± 2.42	95.98 ± 2.18	95.86 ± 2.18	95.17 ± 3.38	96.21 ± 2.43	96.9 ± 1.88	97.13 ± 1.9	96.09 ± 2.49	96.09 ± 2.67	95.98 ± 2.56
d29	96.41 ± 0.84	96.41 ± 2.16	97.93 ± 0.69	97.68 ± 0.72	85.76 ± 5.12	86.21 ± 4.25	82.58 ± 2.4	84.8 ± 2.87	97.58 ± 1.68	97.98 ± 0.98	97.98 ± 0.98
d30	85.75 ± 0.69	85.04 ± 1.01	85.71 ± 1.18	86.0 ± 0.91	80.63 ± 2.98	84.13 ± 1.09	82.43 ± 0.95	82.8 ± 0.91	85.73 ± 0.98	85.53 ± 0.99	85.73 ± 1.2

Table 7 Accuracies of each algorithm on each dataset (continues from Table 6)

id	FVUWA	FTACC	FTCOM	FTCON	FTUWA	VBSM	VGO	VRO	TBSM	TGO	TRO	ALL
d1	98.72 ± 0.75	96.2 ± 1.6	96.2 ± 1.6	96.2 ± 1.6	96.2 ± 1.6	98.49 ± 0.95	98.04 ± 1.16	97.93 ± 0.88	99.05 ± 0.79	98.16 ± 0.95	92.91 ± 4.06	94.36 ± 1.22
d2	97.68 ± 1.86	96.32 ± 1.61	96.32 ± 1.61	96.32 ± 1.61	96.32 ± 1.61	97.6 ± 2.44	96.32 ± 2.54	89.36 ± 5.09	84.4 ± 3.44	89.84 ± 3.06	89.44 ± 3.35	88.88 ± 3.85
d3	96.76 ± 1.32	95.18 ± 1.44	95.18 ± 1.44	95.18 ± 1.44	95.18 ± 1.44	96.76 ± 1.32	96.55 ± 1.11	96.04 ± 1.49	95.18 ± 1.32	95.25 ± 1.98	95.54 ± 1.72	95.4 ± 1.77
d4	99.33 ± 0.51	70.38 ± 2.63	70.38 ± 2.63	70.38 ± 2.63	70.38 ± 2.63	99.45 ± 0.46	98.75 ± 0.96	97.1 ± 1.53	93.94 ± 2.58	96.35 ± 0.96	90.75 ± 2.27	89.25 ± 1.68
d5	52.24 ± 3.24	43.71 ± 2.32	43.71 ± 2.32	43.71 ± 2.32	43.71 ± 2.32	53.37 ± 3.62	53.2 ± 2.05	51.12 ± 3.3	43.71 ± 2.32	49.08 ± 1.64	49.08 ± 2.69	50.85 ± 2.0
d6	83.56 ± 5.08	64.79 ± 4.66	64.79 ± 4.66	64.79 ± 4.66	64.79 ± 4.66	82.19 ± 4.47	83.42 ± 4.59	82.74 ± 4.44	77.12 ± 4.52	80.27 ± 4.48	83.56 ± 3.29	82.88 ± 3.72
d7	75.1 ± 3.49	70.4 ± 3.28	70.4 ± 3.28	70.4 ± 3.28	70.4 ± 3.28	73.35 ± 3.76	74.9 ± 3.1	74.35 ± 2.95	71.9 ± 1.61	75.4 ± 2.54	72.4 ± 4.03	70.55 ± 3.29
d8	97.53 ± 1.56	31.92 ± 2.59	31.92 ± 2.59	31.92 ± 2.59	31.92 ± 2.59	97.26 ± 1.83	97.53 ± 2.02	95.21 ± 3.99	96.44 ± 2.43	96.3 ± 2.67	96.3 ± 2.67	96.3 ± 2.67
d9	84.63 ± 4.67	79.7 ± 5.03	79.7 ± 5.03	79.7 ± 5.03	79.7 ± 5.03	85.67 ± 5.5	84.18 ± 6.79	82.54 ± 4.88	79.7 ± 5.03	83.28 ± 4.03	82.09 ± 4.87	81.64 ± 5.03
d10	71.8 ± 5.67	66.56 ± 6.79	66.56 ± 6.79	66.56 ± 6.79	66.56 ± 6.79	71.48 ± 6.19	70.66 ± 4.73	72.3 ± 5.65	66.56 ± 6.79	70.82 ± 6.22	71.48 ± 5.25	72.46 ± 5.56
d11	82.59 ± 5.3	78.62 ± 2.72	77.59 ± 3.45	77.76 ± 4.34	78.28 ± 3.47	82.93 ± 4.56	82.59 ± 4.56	83.1 ± 4.65	77.07 ± 3.9	78.79 ± 4.07	82.59 ± 4.41	82.93 ± 5.17
d12	83.52 ± 4.97	76.48 ± 5.92	76.48 ± 5.92	76.48 ± 5.92	76.48 ± 5.92	81.85 ± 4.6	82.78 ± 5.09	82.22 ± 6.19	73.89 ± 4.49	81.11 ± 3.47	83.52 ± 4.66	82.59 ± 5.18
d13	58.68 ± 7.8	53.55 ± 4.28	53.55 ± 4.28	53.55 ± 4.28	53.55 ± 4.28	58.02 ± 7.51	56.78 ± 6.96	55.62 ± 7.9	53.55 ± 4.28	53.8 ± 4.62	52.15 ± 7.72	48.6 ± 6.11
d14	99.48 ± 0.22	90.72 ± 1.27	90.72 ± 1.27	90.72 ± 1.27	90.72 ± 1.27	99.43 ± 0.14	99.31 ± 0.25	96.49 ± 3.66	90.72 ± 1.27	92.81 ± 2.45	94.4 ± 2.42	92.29 ± 1.1
d15	94.57 ± 3.0	94.0 ± 2.92	94.0 ± 2.92	94.0 ± 2.92	94.0 ± 2.92	93.57 ± 2.36	93.57 ± 2.36	89.43 ± 3.51	87.43 ± 2.59	92.43 ± 2.7	88.57 ± 3.56	87.29 ± 3.19
d16	99.33 ± 0.5	96.7 ± 0.83	96.7 ± 0.83	96.7 ± 0.83	96.7 ± 0.83	99.34 ± 0.48	99.3 ± 0.49	99.34 ± 0.4	96.67 ± 2.1	99.59 ± 0.24	99.34 ± 0.33	99.01 ± 0.44
d17	81.61 ± 2.93	76.15 ± 1.99	74.74 ± 1.94	76.04 ± 2.43	75.78 ± 1.6	80.73 ± 2.93	80.52 ± 3.37	79.53 ± 2.43	74.79 ± 1.97	77.14 ± 2.15	79.95 ± 2.37	80.0 ± 2.95
d18	73.98 ± 1.57	65.55 ± 1.27	65.55 ± 1.27	65.55 ± 1.27	65.55 ± 1.27	73.6 ± 1.16	73.58 ± 1.56	72.28 ± 3.01	65.55 ± 1.27	66.03 ± 1.14	67.38 ± 1.87	69.2 ± 2.44
d19	97.03 ± 0.49	95.72 ± 0.72	95.72 ± 0.72	95.72 ± 0.72	95.72 ± 0.72	97.02 ± 0.5	96.96 ± 0.41	96.43 ± 0.52	95.72 ± 0.72	95.77 ± 0.77	96.71 ± 0.51	92.71 ± 0.69
d20	45.97 ± 4.92	41.19 ± 3.24	41.04 ± 2.84	40.9 ± 3.16	40.75 ± 3.79	44.18 ± 4.29	44.18 ± 5.72	42.99 ± 3.78	41.04 ± 5.13	41.34 ± 4.22	42.84 ± 6.01	44.18 ± 5.5
d21	97.45 ± 0.8	96.93 ± 1.11	96.93 ± 1.11	96.93 ± 1.11	96.93 ± 1.11	96.8 ± 0.76	96.56 ± 0.86	96.36 ± 1.36	96.93 ± 1.11	97.66 ± 1.08	97.58 ± 0.66	96.8 ± 1.09
d22	98.55 ± 0.54	96.1 ± 0.73	96.1 ± 0.73	96.1 ± 0.73	96.1 ± 0.73	98.55 ± 0.55	98.46 ± 0.62	95.41 ± 2.29	96.1 ± 0.73	97.31 ± 0.95	97.02 ± 1.84	93.93 ± 0.58
d23	85.37 ± 5.63	79.27 ± 7.38	79.27 ± 7.38	79.27 ± 7.38	79.27 ± 7.38	85.61 ± 4.94	82.44 ± 6.17	81.46 ± 8.23	85.61 ± 4.22	85.61 ± 3.72	83.9 ± 4.33	80.24 ± 5.33
d24	91.32 ± 2.37	26.91 ± 4.23	26.91 ± 4.23	26.91 ± 4.23	26.91 ± 4.23	91.54 ± 2.38	91.32 ± 2.72	88.97 ± 3.53	90.81 ± 2.03	89.04 ± 2.91	88.09 ± 2.59	81.32 ± 2.6
d25	94.36 ± 0.77	90.9 ± 0.83	90.9 ± 0.83	90.9 ± 0.83	90.9 ± 0.83	93.33 ± 0.93	93.78 ± 0.92	93.58 ± 0.72	90.9 ± 0.83	90.88 ± 0.81	94.1 ± 1.08	87.52 ± 1.1

Table 7 (*Continued*)

id	FVUWA	FTACC	FTCOM	FTCON	FTUWA	VBSM	VGO	VRO	TBSM	TGO	TRO	ALL
d26	98.06 ± 1.02	65.39 ± 4.03	65.39 ± 4.03	65.39 ± 4.03	65.39 ± 4.03	97.59 ± 0.75	97.59 ± 0.75	97.43 ± 0.8	97.43 ± 1.03	97.96 ± 0.84	93.46 ± 2.53	85.39 ± 2.86
d27	84.79 ± 3.07	72.78 ± 6.17	72.78 ± 6.17	72.78 ± 6.17	72.78 ± 6.17	83.73 ± 3.4	83.91 ± 2.58	79.53 ± 4.03	70.41 ± 3.71	76.51 ± 4.89	75.74 ± 2.95	75.44 ± 5.06
d28	96.21 ± 2.49	93.68 ± 5.06	95.06 ± 4.37	94.71 ± 4.44	93.45 ± 4.97	96.21 ± 2.6	96.44 ± 2.13	95.86 ± 2.49	95.29 ± 1.26	96.9 ± 1.63	96.55 ± 2.03	96.44 ± 2.06
d29	98.03 ± 1.02	99.34 ± 0.63	99.34 ± 0.63	99.34 ± 0.63	99.34 ± 0.63	97.98 ± 0.98	97.93 ± 0.87	94.44 ± 4.05	99.14 ± 0.79	86.31 ± 3.1	88.48 ± 3.8	84.9 ± 2.88
d30	86.19 ± 1.09	84.73 ± 1.4	84.73 ± 1.4	84.73 ± 1.4	84.73 ± 1.4	85.69 ± 1.28	85.87 ± 1.2	85.91 ± 1.24	73.45 ± 1.26	82.7 ± 1.16	84.61 ± 1.29	85.65 ± 1.17

Table 8 Rank of each algorithm on each dataset

id	BVACC	BVCOM	BVCON	BVUWA	BTACC	BTCOM	BTCON	BTUWA	FVACC	FVCOM	FVCON	FVUWA	FTACC	FTCOM	FTCON	FTUWA	VBSM	VGO	VRO	TBSM	TGO	TRO	ALL
d1	16.0	9.5	8.0	12.0	3.5	21.0	9.5	14.0	3.5	6.0	6.0	2.0	18.5	18.5	18.5	18.5	6.0	13.0	15.0	1.0	11.0	23.0	22.0
d2	14.5	13.0	4.5	7.0	22.0	18.0	16.5	14.5	6.0	4.5	3.0	1.0	10.0	10.0	10.0	10.0	2.0	10.0	20.0	23.0	16.5	19.0	21.0
d3	10.0	12.5	4.0	6.0	4.0	23.0	12.5	14.5	9.0	8.0	4.0	1.5	20.0	20.0	20.0	20.0	1.5	7.0	11.0	20.0	17.0	14.5	16.0
d4	10.0	9.0	6.0	7.0	17.0	16.0	12.0	13.0	4.0	5.0	2.0	3.0	21.5	21.5	21.5	21.5	1.0	8.0	11.0	15.0	14.0	18.0	19.0
d5	10.0	9.0	3.0	1.0	16.5	18.0	16.5	15.0	2.0	7.0	6.0	8.0	21.0	21.0	21.0	21.0	4.0	5.0	11.0	21.0	13.5	13.5	12.0
d6	11.5	3.0	7.5	1.0	17.0	13.0	18.0	16.0	6.0	7.5	9.5	3.0	21.5	21.5	21.5	21.5	14.0	5.0	11.5	19.0	15.0	3.0	9.5
d7	15.0	13.5	10.0	11.0	3.0	2.0	1.0	5.0	13.5	7.0	9.0	6.0	21.5	21.5	21.5	21.5	16.0	8.0	12.0	18.0	4.0	17.0	19.0
d8	19.0	7.5	1.0	6.0	17.0	16.0	13.0	13.0	9.0	7.5	3.0	3.0	21.5	21.5	21.5	21.5	5.0	3.0	18.0	10.0	13.0	13.0	13.0
d9	14.5	9.0	1.0	7.0	18.0	13.0	11.5	11.5	4.0	5.0	2.5	6.0	21.0	21.0	21.0	21.0	2.5	8.0	14.5	21.0	10.0	16.0	17.0
d10	4.0	4.0	11.0	4.0	9.0	7.0	13.5	15.5	7.0	13.5	15.5	7.0	21.0	21.0	21.0	21.0	11.0	18.0	2.0	21.0	17.0	11.0	1.0
d11	2.0	12.0	3.5	1.0	21.5	14.0	19.0	15.0	7.0	3.5	13.0	10.0	17.0	21.5	20.0	18.0	7.0	10.0	5.0	23.0	16.0	10.0	7.0
d12	12.0	12.0	8.5	3.0	8.5	12.0	16.5	18.0	16.5	5.5	14.5	1.5	20.5	20.5	20.5	20.5	10.0	4.0	7.0	23.0	14.5	1.5	5.5
d13	22.0	4.0	3.0	8.0	20.0	13.0	11.5	11.5	7.0	1.0	2.0	5.0	17.0	17.0	17.0	17.0	6.0	9.0	10.0	17.0	14.0	21.0	23.0
d14	12.0	8.0	5.0	6.5	20.5	15.0	17.0	16.0	6.5	2.0	3.5	1.0	20.5	20.5	20.5	20.5	3.5	9.0	10.0	20.5	13.0	11.0	14.0
d15	19.0	13.5	10.0	7.5	21.0	17.0	13.5	15.5	7.5	7.5	7.5	1.0	3.5	3.5	3.5	3.5	11.5	11.5	18.0	22.0	15.5	20.0	23.0
d16	17.0	15.0	5.5	5.5	18.0	4.0	2.5	2.5	10.5	14.0	12.0	10.5	20.5	20.5	20.5	20.5	8.0	13.0	8.0	23.0	1.0	8.0	16.0
d17	12.0	5.0	1.0	6.0	17.0	20.5	16.0	15.0	3.0	7.5	4.0	2.0	18.0	23.0	19.0	20.5	7.5	9.0	13.0	22.0	14.0	11.0	10.0
d18	10.0	7.0	8.0	2.0	15.0	17.0	16.0	18.0	9.0	3.0	4.5	1.0	21.0	21.0	21.0	21.0	4.5	6.0	11.0	21.0	14.0	13.0	12.0
d19	12.0	7.5	9.0	7.5	16.5	16.5	13.0	14.5	1.0	6.0	4.0	2.0	20.0	20.0	20.0	20.0	3.0	5.0	11.0	20.0	14.5	10.0	23.0
d20	10.5	9.0	7.0	4.5	23.0	14.0	16.5	19.0	8.0	13.0	2.0	1.0	16.5	19.0	21.0	22.0	4.5	4.5	10.5	19.0	15.0	12.0	4.5
d21	23.0	19.0	6.0	7.5	9.0	20.5	5.0	3.0	16.0	15.0	7.5	4.0	12.0	12.0	12.0	12.0	17.5	20.5	22.0	12.0	1.0	2.0	17.5
d22	12.0	4.0	9.0	8.0	18.5	10.0	15.0	14.0	5.5	1.0	5.5	2.5	18.5	18.5	18.5	18.5	2.5	7.0	22.0	18.5	11.0	13.0	23.0
d23	18.0	12.5	15.0	5.0	1.0	9.5	5.0	2.0	11.0	14.0	9.5	8.0	21.5	21.5	21.5	21.5	5.0	16.0	17.0	5.0	5.0	12.5	19.0
d24	13.5	10.0	5.5	5.5	17.0	18.0	8.0	12.0	1.0	4.0	2.5	8.0	21.5	21.5	21.5	21.5	2.5	8.0	15.0	11.0	13.5	16.0	19.0
d25	9.0	5.0	2.0	8.0	21.0	14.0	21.0	13.0	3.0	6.0	1.0	4.0	17.0	17.0	17.0	17.0	12.0	10.0	11.0	17.0	21.0	7.0	23.0

Table 8 (*Continued*)

id	BVACC	BVCOM	BVCON	BVUWA	BTACC	BTCOM	BTCON	BTUWA	FVACC	FVCOM	FVCON	FVUWA	FTACC	FTCOM	FTCON	FTUWA	VBSM	VGO	VRO	TBSM	TGO	TRO	ALL
d26	17.0	7.0	8.5	8.5	6.0	3.5	3.5	3.5	14.0	11.0	10.0	1.0	21.5	21.5	21.5	21.5	12.5	12.5	15.5	15.5	3.5	18.0	19.0
d27	10.0	5.5	3.0	2.0	18.0	12.0	16.5	13.0	8.0	9.0	4.0	1.0	20.5	20.5	20.5	20.5	7.0	5.5	11.0	23.0	14.0	15.0	16.5
d28	11.0	17.0	13.5	15.5	19.0	8.0	2.5	1.0	11.0	11.0	13.5	8.0	22.0	20.0	21.0	23.0	8.0	5.5	15.5	18.0	2.5	4.0	5.5
d29	14.5	14.5	10.5	12.0	20.0	19.0	23.0	22.0	13.0	8.0	8.0	6.0	2.5	2.5	2.5	2.5	8.0	10.5	16.0	5.0	18.0	17.0	21.0
d30	5.0	12.0	8.0	2.0	22.0	18.0	21.0	19.0	6.5	11.0	6.5	1.0	14.5	14.5	14.5	14.5	9.0	4.0	3.0	23.0	20.0	17.0	10.0

Table 9 Size of the pruned ensemble for each algorithm on each dataset

id	BVACC	BVCOM	BVCON	BVUWA	BTACC	BTCOM	BTCON	BTUWA	FVACC	FVCOM	FVCON	FVUWA	FTACC	FTCOM	FTCON	FTUWA	VGO	VRO	TGO	TRO
d1	36.2	64.0	58.2	62.7	4.4	44.7	43.1	45.3	1.8	2.8	2.1	1.0	1.0	1.0	1.0	1.0	57.8	95.8	51.6	94.7
d2	69.0	5.6	6.2	6.3	25.8	44.4	50.5	52.2	3.4	2.9	2.6	2.3	1.0	1.0	1.0	1.0	3.8	130.5	53.1	90.4
d3	72.8	48.3	44.5	64.1	10.6	21.7	8.5	12.1	1.3	1.9	2.0	1.2	1.0	1.0	1.0	1.0	63.7	111.6	18.3	67.3
d4	6.2	30.4	16.1	18.4	54.3	107.9	108.9	109.5	2.7	4.6	1.2	1.6	1.0	1.0	1.0	1.0	21.3	54.6	112.7	126.7
d5	16.3	8.4	19.6	22.9	46.1	17.6	33.1	33.3	11.6	13.3	11.8	22.7	1.0	1.0	1.0	1.0	9.2	19.2	36.9	45.1
d6	109.9	41.4	54.9	52.4	5.1	20.5	17.9	17.8	16.3	10.6	5.3	8.3	1.0	1.0	1.0	1.0	53.8	118.4	21.0	60.9
d7	10.6	28.1	32.8	26.7	18.1	65.2	63.6	64.1	10.4	14.4	14.6	14.6	1.0	1.0	1.0	1.0	11.1	29.3	64.3	76.6
d8	97.3	81.8	68.2	75.7	198.4	198.6	197.9	197.9	15.6	7.5	14.1	3.4	1.0	1.0	1.0	1.0	72.2	110.0	198.8	191.5
d9	38.9	36.0	15.5	41.9	10.5	14.7	27.2	28.8	2.6	2.0	6.4	5.4	1.0	1.0	1.0	1.0	20.8	99.7	39.3	69.3
d10	104.2	65.9	61.7	41.5	24.0	39.1	34.5	34.9	8.0	16.3	12.7	7.0	1.0	1.0	1.0	1.0	57.4	181.6	35.2	111.1
d11	97.2	56.3	39.6	34.7	6.2	14.7	11.5	15.6	1.6	5.1	4.9	3.1	3.3	3.6	3.2	4.0	45.5	161.1	13.0	59.1
d12	93.9	62.0	35.6	46.3	24.7	46.2	43.5	43.3	2.9	11.1	4.7	5.2	1.0	1.0	1.0	1.0	48.7	108.1	44.3	76.3
d13	20.0	4.9	5.9	9.8	60.5	49.2	10.6	17.4	13.4	5.5	4.3	4.7	1.0	1.0	1.0	1.0	24.8	76.6	34.3	78.2
d14	2.8	26.8	36.5	47.7	2.4	9.4	6.0	6.1	2.1	2.3	1.0	1.0	1.0	1.0	1.0	1.0	42.1	108.7	10.0	100.5
d15	20.7	26.6	15.4	18.4	7.6	60.4	54.6	54.8	1.8	2.1	3.2	5.1	1.0	1.0	1.0	1.0	17.1	89.7	55.5	103.1
d16	44.6	52.6	32.0	28.5	25.6	62.8	62.2	62.2	3.6	9.3	7.3	8.2	1.0	1.0	1.0	1.0	40.3	100.6	66.3	124.6
d17	45.0	54.7	42.8	43.7	5.2	4.9	6.9	9.3	25.6	17.3	21.3	13.8	2.8	1.8	5.8	4.3	30.7	50.7	11.8	31.1
d18	12.1	8.3	5.2	3.4	22.1	7.3	9.1	7.0	8.0	1.2	1.3	1.7	1.0	1.0	1.0	1.0	3.9	45.6	18.6	28.7
d19	2.3	14.5	29.0	36.1	5.3	5.8	4.6	4.6	5.7	5.8	6.1	27.6	1.0	1.0	1.0	1.0	29.3	41.2	9.3	22.7
d20	39.8	44.6	33.2	20.6	3.0	11.8	13.2	15.4	11.7	7.1	9.3	9.5	8.1	6.6	6.8	14.2	27.0	72.8	32.5	45.0
d21	47.7	34.6	16.6	27.6	15.7	22.3	11.5	13.2	17.0	15.3	5.7	13.0	1.0	1.0	1.0	1.0	11.8	63.4	36.0	81.8
d22	5.8	14.3	20.4	23.9	6.0	10.4	6.0	6.2	3.3	4.3	3.6	12.6	1.0	1.0	1.0	1.0	12.7	131.5	8.1	62.6
d23	40.0	49.8	38.0	44.8	33.7	151.0	139.8	139.6	12.1	20.9	11.2	7.6	1.0	1.0	1.0	1.0	35.0	104.4	142.3	146.6
d24	24.2	35.5	9.6	9.4	28.5	71.8	37.7	59.2	6.8	1.4	3.0	1.8	1.0	1.0	1.0	1.0	9.2	58.0	76.9	86.1
d25	9.9	12.4	26.5	40.3	7.2	6.1	6.0	6.0	7.2	15.0	18.4	6.9	1.0	1.0	1.0	1.0	10.4	20.9	6.0	33.0

Table 9 (*Continued*)

id	BVACC	BVCOM	BVCON	BVUWA	BTACC	BTCOM	BTCON	BTUWA	FVACC	FVCOM	FVCON	FVUWA	FTACC	FTCOM	FTCON	FTUWA	VGO	VRO	TGO	TRO
d26	116.4	32.6	23.8	30.6	12.4	112.3	94.4	97.2	1.3	4.5	3.7	5.1	1.0	1.0	1.0	1.0	50.3	114.7	100.1	119.3
d27	21.0	6.1	10.1	7.5	17.7	58.8	38.1	43.7	7.6	6.1	6.2	6.5	1.0	1.0	1.0	1.0	9.2	39.4	61.9	82.9
d28	81.4	76.1	60.6	64.8	11.3	44.3	19.1	12.6	2.2	1.2	1.8	1.9	1.7	7.8	3.0	2.2	57.8	150.7	37.3	89.6
d29	59.4	16.6	11.4	12.3	170.7	177.8	62.8	73.4	2.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	11.9	72.4	132.9	168.8
d30	50.3	58.8	22.5	16.2	33.0	52.5	25.9	27.0	23.4	8.0	14.1	22.2	1.0	1.0	1.0	1.0	12.7	24.2	34.8	53.6

References

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Banfield, R. E., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2005). Ensemble diversity measures and their application to thinning. *Information Fusion*, *6*(1), 49–62.

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

Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A. (2004). Ensemble selection from libraries of models. In *Proceedings of the 21st international conference on machine learning*, p. 18.

Caruana, R., Munson, A., & Niculescu-Mizil, A. (2006). Getting the most out of ensemble selection. In *Proceedings of the international conference on data mining (ICDM)*, pp. 828–833.

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

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proceedings of the 1st international workshop in multiple classifier systems* (pp. 1–15).

Fan, W., Chu, F., Wang, H., & Yu, P. S. (2002). Pruning and dynamic scheduling of cost-sensitive ensembles. In *Eighteenth national conference on artificial intelligence, American association for artificial intelligence* (pp. 146–151).

Giacinto, G., Roli, F., & Fumera, G. (2000). Design of effective multiple classifier systems by clustering of classifiers. In *15th international conference on pattern recognition, ICPR, 2000* (pp. 160–163).

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, *51*(2), 181–207.

Margineantu, D., & Dietterich, T. (1997). Pruning adaptive boosting. In *Proceedings of the 14th international conference on machine learning* (pp. 211–218).

Martinez-Munoz, G., & Suarez, A. (2004). Aggregation ordering in bagging. In *International conference on artificial intelligence and applications (IASTED)* (pp. 258–263). Calgary: Acta Press.

Martinez-Munoz, G., & Suarez, A. (2006). Pruning in ordered bagging ensembles. In *23rd international conference in machine learning (ICML-2006)* (pp. 609–616). New York: ACM Press.

Partalas, I., Tsoumakas, G., & Vlahavas, I. (2008). Focused ensemble selection: a diversity-based method for greedy ensemble selection. In M. Ghallab, C. D. Spyropoulos, N. Fakotakis, & N. M. Avouris (Eds.), *Frontiers in artificial intelligence and applications: Vol. 178. ECAI 2008—18th European conference on artificial intelligence*, Patras, Greece, July 21–25, 2008 (pp. 117–121). Amsterdam: IOS Press.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227.

Tang, E. K., Suganthan, P. N., & Yao, X. (2006). An analysis of diversity measures. *Machine Learning*, *65*(1), 247–271.

Tsoumakas, G., Angelis, L., & Vlahavas, I. (2005). Selective fusion of heterogeneous classifiers. *Intelligent Data Analysis*, *9*(6), 511–525.

Tsoumakas, G., Partalas, I., & Vlahavas, I. (2009). An ensemble pruning primer. In O. Okun, & G. Valentino (Eds.), *Applications of supervised and unsupervised ensemble methods* (pp. 1–13). Berlin: Springer.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, *1*, 80–83.

Witten, I. H., & Frank, E. (2005). *Data mining: practical machine learning tools and techniques*. San Mateo: Morgan Kaufmann.