Research Article

Forest Pruning Based on Branch Importance

Xiangkui Jiang,¹ Chang-an Wu,² and Huaping Guo²

¹*School of Automation, Xi'an University of Posts and Telecommunication, Xi'an, Shaanxi 710121, China*

²*School of Computer and Information Technology, Xinyang Normal University, Xinyang, Henan 464000, China*

Correspondence should be addressed to Huaping Guo; hpguo_cm@163.com

Received 19 January 2017; Revised 29 March 2017; Accepted 30 April 2017; Published 1 June 2017

Academic Editor: Michael Schmucker

Copyright © 2017 Xiangkui Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A forest is an ensemble with decision trees as members. This paper proposes a novel strategy to pruning forest to enhance ensemble generalization ability and reduce ensemble size. Unlike conventional ensemble pruning approaches, the proposed method tries to evaluate the importance of branches of trees with respect to the whole ensemble using a novel proposed metric called importance gain. The importance of a branch is designed by considering ensemble accuracy and the diversity of ensemble members, and thus the metric reasonably evaluates how much improvement of the ensemble accuracy can be achieved when a branch is pruned. Our experiments show that the proposed method can significantly reduce ensemble size and improve ensemble accuracy, no matter whether ensembles are constructed by a certain algorithm such as bagging or obtained by an ensemble selection algorithm, no matter whether each decision tree is pruned or unpruned.

1. Introduction

Ensemble learning is a very important research topic in machine learning and data mining. The basic heuristic is to create a set of learners and aggregate the prediction of each learner for classifying examples. Many approaches such as bagging [1], boosting [2], and COPEN [3] have been proposed to create ensembles, and the key to the success of these approaches is that base learners are accurate and diverse [4].

Ensemble methods have been applied to many applications such as image detection [5–7] and imbalanced learning problem [8]. However, an important drawback existing in ensemble learning approaches is that they try to train unnecessarily large ensembles. Large ensembles need a large memory for storing the bases learners and much response time for prediction. Besides, large ensemble may reduce its generalization ability instead of increasing the performance [9]. Therefore, a lot of researches to tackle this problem have been carried out, and the researches mainly focus on ensemble selection: selecting a subset of ensemble members for prediction, such as ordered-based ensemble selection methods [10–12] and greedy heuristic based ensemble selection methods [13–21]. The research results indicate that a well-designed ensemble selection method can reduce ensemble size and improve ensemble accuracy.

Besides ensemble selection, we can prune an ensemble through the following two approaches if ensemble members are decision trees: (1) pruning individual members separately and combining the pruned members together for prediction and (2) repeatedly pruning individual members by considering the overall performance of the ensemble. For the first strategy, many decision tree pruning methods such as those used in CART [22] and C4.5 [23] have been studied. Although pruning can simplify model structure, whether pruning can improve model accuracy is still a controversial topic in machine learning [24]. The second strategy coincides with the expectation of improving model generalization ability globally. However, this method has not been extensively studied. This paper focuses on this strategy and names the strategy as forest pruning (FP).

The major job of forest pruning is to define an effective metric evaluating the importance of a certain branch of trees. Traditional metrics can not be applied to forest pruning, since these metrics just consider the influence on a single decision tree when a branch is pruned. Therefore, we need a new metric for pruning forest. Our contributions in this paper are as follows:

- (i) Introduce a new ensemble pruning strategy to prune decision tree based ensemble;

- (ii) propose a novel metric to measure the improvement of forest performance when a certain node grows into a subtree;
- (iii) present a new ensemble pruning algorithm with the proposed metric to prune a decision tree based ensemble. The ensemble can be learned by a certain algorithm or obtained by some ensemble selection method. Each decision tree can be pruned or unpruned.

Experimental results show that the proposed method can significantly reduce the ensemble size and improve its accuracy. This result indicates that the metric proposed in this paper reasonably measures the influence on ensemble accuracy when a certain node grows into a subtree.

The rest of this paper is structured as follows. Section 2 provides a survey of ensemble of decision trees; Section 3 presents the formal description of forest trimming and the motivation of this study by an example. Section 4 introduces a new forest pruning algorithm. Section 5 reports and analyzes experimental results and we conclude the paper with simple remark and future work in Section 6.

2. Forests

A forest is an ensemble whose members are learned by decision tree learning method. Two approaches are often used to train a forest: traditional approaches and the methods specially designed for forests.

Bagging [1] and boosting [2] are the two most often used traditional methods to build forests. Bagging takes bootstrap samples of objects and trains a tree on each sample. The classifier votes are combined by majority voting. In some implementations, classifiers produce estimates of the posterior probabilities for the classes. These probabilities are averaged across the classifiers and the most probable class is assigned, called “average” or “mean” aggregation of the outputs. Bagging with average aggregation is implemented in Weka and used in the experiments in this paper. Since each individual classifier is trained on a bootstrap sample, the data distribution seen during training is similar to the original distribution. Thus, the individual classifiers in a bagging ensemble have relatively high classification accuracy. The factor encouraging diversity between these classifiers is the proportion of different examples in the training set. Boosting is a family of methods and Adaboost is the most prominent member. The idea is to boost the performance of a “weak” classifier (can be decision tree) by using it within an ensemble structure. The classifiers in the ensemble are added one at a time so that each subsequent classifier is trained on data which have been “hard” for the previous ensemble members. A set of weights is maintained across the objects in the data set so that objects that have been difficult to classify acquire more weight, forcing subsequent classifiers to focus on them.

Random forest [25] and rotation forest [26] are two important approaches specially designed for building forests. Random forest is a variant version of bagging. The forest is built again on bootstrap samples. The difference lies in the construction of the decision tree. The feature to split a node is selected as the best feature among a set of M randomly chosen

features, where M is a parameter of the algorithm. This small alteration appeared to be a winning heuristic in that diversity was introduced without much compromising the accuracy of the individual classifiers. Rotation forest randomly splits the feature set into K subsets (K is a parameter of the algorithm) and Principal Component Analysis (PCA) [27] is applied to each subset. All principal components are retained in order to preserve the variability information in the data. Thus, K axis rotations take place to form the new features and rotation forest building a tree using all training set in the new space defined by a given new feature space.

3. Problem Description and Motivation

3.1. Problem Description. Let $D = \{(x_i, y_i) \mid i = 1, 2, \dots, N\}$ be a data set, and let $F = \{T_1, \dots, T_M\}$ be an ensemble with decision tree, T_i , learning from D . Denote by $v \in T_i$ a node in tree T and by $E(v) \in D$, the set of the examples reaching v from the root of T , $\text{root}(T)$. Suppose each node $v \in T$ contains a vector $(p_1^v, p_2^v, \dots, p_K^v)$, where p_k^v is the proportion of the examples in $E(v)$ associated with label k . If $v \in T_i$ is a leaf and $\mathbf{x}_i \in E(v)$, the prediction of T_i on $\mathbf{x}_i$ is

$$T_i(\mathbf{x}_i) = \arg \max_k p_k^v. \quad (1)$$

Similarly, for each example $\mathbf{x}_j$ to be classified, ensemble F returns a vector $(p_{j1}, p_{j2}, \dots, p_{jK})$ indicating that $\mathbf{x}_j$ belongs to label k with probability p_{jk} , where

$$p_{jk} = \frac{1}{M} \sum_{i=1}^M p_{jk}^{(i)}, \quad k = 1, 2, \dots, K. \quad (2)$$

The prediction of F on $\mathbf{x}_j$ is $F(\mathbf{x}_j) = \arg \max_k p_{jk}$.

Now, our problem is, given a forest F with M decision trees, how to prune each tree to reduce F 's size and improve its accuracy, where F is either constructed by some algorithm or obtained by some ensemble selection method.

3.2. Motivation. First, let us look at an example, which shows the possibility that forest trimming can improve ensemble accuracy.

Example 1. Let $F = \{T_0, T_1, \dots, T_9\}$ be a forest with ten decision trees, where T_1 is shown in Figure 1. Suppose that $p_1^v = 0.60$, $p_2^v = 0.40$; $p_1^{v_1} = 1.00$, $p_2^{v_1} = 0.00$; $p_1^{v_2} = 0.20$ and $p_2^{v_2} = 0.80$. Let ten examples $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_9$ reach node v , where $\mathbf{x}_0, \dots, \mathbf{x}_5$ associate with label 1 and $\mathbf{x}_6, \dots, \mathbf{x}_9$ associate with label 2. Assume examples $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_4$ reach leaf node v_1 , and $\mathbf{x}_5, \dots, \mathbf{x}_9$ reach leaf node v_2 .

Obviously, for T_0 , we can not prune the children of node v , since treating v as a leaf would lead to more examples incorrectly classified by T_0 .

Assume that F 's predictions on $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_9$ are as follows:

$$p_{01} = 0.65,$$

$$p_{11} = 0.70,$$

FIGURE 1: Decision tree T_0 . v is a test node and v_1 and v_2 are two leaves.

$$\begin{aligned}
 p_{21} &= 0.70, & p_{31} &= 0.65, \\
 p_{31} &= 0.65, & p_{41} &= 0.75, \\
 p_{41} &= 0.80, & p_{51} &= 0.52, \\
 p_{51} &= 0.49, & p_{61} &= 0.33, \\
 p_{61} &= 0.30, & p_{71} &= 0.22, \\
 p_{71} &= 0.19, & p_{81} &= 0.23, \\
 p_{81} &= 0.20, & p_{91} &= 0.33, \\
 p_{91} &= 0.30, & p_{02} &= 0.40, \\
 p_{02} &= 0.35, & p_{12} &= 0.35, \\
 p_{12} &= 0.30, & p_{22} &= 0.35, \\
 p_{22} &= 0.30, & p_{32} &= 0.35, \\
 p_{32} &= 0.35, & p_{42} &= 0.25, \\
 p_{42} &= 0.20, & p_{52} &= 0.48, \\
 p_{52} &= 0.51, & p_{62} &= 0.67, \\
 p_{62} &= 0.70, & p_{72} &= 0.78, \\
 p_{72} &= 0.81, & p_{82} &= 0.77, \\
 p_{82} &= 0.80, & p_{92} &= 0.67. \\
 p_{92} &= 0.70.
 \end{aligned} \tag{4}$$

(3)

where p_{jk} is the probability of $\mathbf{x}_j$ associated with label k . From F 's predictions shown above, we have that $\mathbf{x}_6$ is incorrectly classified by F . Update T_0 to T'_0 by pruning v 's children and update F to $F' = \{T'_0, T_1, \dots, T_9\}$. A simple calculation tells us that, for the ten examples, F' returns:

$$\begin{aligned}
 p_{01} &= 0.61, \\
 p_{11} &= 0.65, \\
 p_{21} &= 0.65,
 \end{aligned}$$

It is easy to see that F' correctly classifies all of the ten examples.

This example shows that if a single decision tree is considered, maybe it should not be pruned any more. However, for the forest as a whole, it is still possible to prune some branches of the decision tree, and this pruning will probably improve the ensemble accuracy instead of reducing it.

Although the example above is constructed by us, similar cases can be seen everywhere when we study ensembles further. It is this observation that motivates us to study forest trimming methods. However, more efforts are needed to

turn possibility into feasibility. Further discussions about this problem will be presented in the next section.

4. Forest Pruning Based on Branch Importance

4.1. The Proposed Metric and Algorithm Idea. To avoid trapping in detail too early, we assume that $I(v, F, \mathbf{x}_j)$ has been defined, which is the importance of node v when forest F classifies example $\mathbf{x}_j$. If $\mathbf{x}_j \notin E(v)$, then $I(v, F, \mathbf{x}_j) = 0$. Otherwise, the details of the definition of $I(v, F, \mathbf{x}_j)$ are presented in Section 3.2.

Let $T \in F$ be a tree and let $v \in T$ be a node. The importance of v with respect to forest F is defined as

$$I(v, F) = \sum_{\mathbf{x}_j \in D'} I(v, F, \mathbf{x}_j) = \sum_{\mathbf{x}_j \in E(v)} I(v, F, \mathbf{x}_j), \quad (5)$$

where D' is a pruning set and $E(v)$ is the set of the example in D' reaching node v from root(T). $I(v, F)$ reflects the impact of node v on F 's accuracy.

Let $L(v)$ be the set of leaf nodes of branch(v), the branch (subtree) with v as the root. The contribution of branch(v) to F is defined as

$$I(\text{branch}(v), F) = \sum_{v' \in L(v)} I(v', F), \quad (6)$$

which is the sum of the importance of leaves in branch(v).

Let $v \in T$ be a nonterminal node. The importance gain of v to F is defined by the importance difference between branch(v) and node v , that is,

$$\text{IG}(v, F) = I(\text{branch}(v)) - I(v, F), \quad (7)$$

$\text{IG}(v, F)$ can be considered as the importance gain of branch(v), and its value reflects how much improvement of the ensemble accuracy is achieved when v grows into a subtree. If $\text{IG}(v, F) > 0$, then this expansion is helpful to improve F 's accuracy. Otherwise it is unhelpful to improve or even reduce F 's accuracy.

The idea of the proposed method of pruning ensemble of decision trees is as follows. For each nonterminal node v in each tree T , calculate its importance gain $\text{IG}(v, F)$ on the pruning set. If $\text{IG}(v, F)$ is smaller than a threshold, prune branch(v) and treat v as a leaf. This procedure continues until all decision trees can not be pruned.

Before presenting the specific details of the proposed algorithm, we introduce how to calculate $I(v, F, \mathbf{x}_j)$ in the next subsection.

4.2. $\text{Con}(v, F, \mathbf{x}_j)$ Calculation. Let h be a classifier and let S be an ensemble. Partalas et al. [28, 29] identified that the prediction of h and S on an example $\mathbf{x}_j$ can be categorized into four cases: (1) $e_{tf}: h(\mathbf{x}_j) = y_j \wedge S(\mathbf{x}_j) \neq y_j$, (2) $e_{tt}: h(\mathbf{x}_j) = y_j \wedge S(\mathbf{x}_j) = y_j$, (3) $e_{ft}: h(\mathbf{x}_j) \neq y_j \wedge S(\mathbf{x}_j) = y_j$, (4) $e_{ff}: h(\mathbf{x}_j) \neq y_j \wedge S(\mathbf{x}_j) \neq y_j$. They concluded that considering all four cases is crucial to design ensemble diversity metrics.

Based on the four cases above, Lu et al. [11] introduced a metric, $\text{IC}_i^{(j)}$, to evaluate the contribution of the i th classifier

to S when S classifies the j th instance. Partalas et al. [28, 29] introduced a measure called Uncertainty Weighted Accuracy, $\text{UWA}_D(h, S, \mathbf{x}_j)$, to evaluate h 's contribution when S classifies example $\mathbf{x}_j$.

Similar to the discussion above, we define

$$\begin{aligned} e_{tf}(v) &= \{\mathbf{x}_j \mid \mathbf{x}_j \in E(v) \wedge T(\mathbf{x}_j) = y_j \wedge F(\mathbf{x}_j) \neq y_j\}, \\ e_{tt}(v) &= \{\mathbf{x}_j \mid \mathbf{x}_j \in E(v) \wedge T_i(\mathbf{x}_j) = y_j \wedge F(\mathbf{x}_j) = y_j\}, \\ e_{ft}(v) &= \{\mathbf{x}_j \mid \mathbf{x}_j \in E(v) \wedge T_i(\mathbf{x}_j) \neq y_j \wedge F(\mathbf{x}_j) = y_j\}, \\ e_{ff}(v) &= \{\mathbf{x}_j \mid \mathbf{x}_j \in E(v) \wedge T_i(\mathbf{x}_j) \neq y_j \wedge F(\mathbf{x}_j) \neq y_j\}. \end{aligned} \quad (8)$$

In the following discussions, we assume that $v \in T$ and $\mathbf{x}_j \in E(v)$. Let f_m and f_s be the subscripts of the largest element and the second largest element in $\{p_{j1}, \dots, p_{jK}\}$, respectively. Obviously, f_m is the label of $\mathbf{x}_j$ predicted by ensemble F . Similarly, let $t_m = \arg \max_k (p_1^v, \dots, p_K^v)$. If v is a leaf node, then t_m is the label of $\mathbf{x}_j$ predicted by decision tree T . Otherwise, t_m is the label of $\mathbf{x}_j$ predicted by T' , where T' is the decision tree obtained from T by pruning branch(v). For simplicity, we call t_m the label of $\mathbf{x}_j$ predicted by node v and say node v correctly classifies $\mathbf{x}_j$ if $t_m = y_j$.

We define $I(v, F, \mathbf{x}_j)$ based on the four cases in formula (8), respectively. If $\mathbf{x}_j \in e_{tf}(v)$ or $\mathbf{x}_j \in e_{tt}(v)$, then $\text{Con}(v, F, \mathbf{x}_j) \geq 0$, since v correctly classifies $\mathbf{x}_j$. Otherwise, $\text{Con}(v, F, \mathbf{x}_j) < 0$, since v incorrectly classifies $\mathbf{x}_j$.

For $\mathbf{x}_j \in e_{tf}(v)$, $\text{Con}(v, F, \mathbf{x}_j)$ is defined as

$$I(v, F, \mathbf{x}_j) = \frac{p_{t_m}^v - p_{f_m}^v}{M(p_{jf_m} - p_{jt_m} + 1/M)}, \quad (9)$$

where M is the number of base classifiers in F . Here, $t_m = y_j$ and $f_m \neq y_j$, then $p_{t_m}^v \geq p_{f_m}^v$, $p_{jf_m} \geq p_{jt_m}$, and thus $0 \leq \text{Con}(v, F, \mathbf{x}_j) \leq 1$. Since $p_{f_m}^v$ is the contribution of node v to the probability that F correctly predicates $\mathbf{x}_j$ belonging to class t_m while $p_{f_m}^v$ is the contribution of node v to p_{jf_m} , the probability that F incorrectly predicates $\mathbf{x}_j$ belongs to class f_m , $(p_{t_m}^v - p_{f_m}^v)/M$ can be considered as the net importance of node v when F classifies $\mathbf{x}_j$. $p_{jf_m} - p_{jt_m}$ is the weight of v 's net contribution, which reflects the importance of node v for classifying $\mathbf{x}_j$ correctly. The constant $1/M$ is to avoid $p_{jf_m} - p_{jt_m}$ being zero or too small.

For $\mathbf{x}_j \in e_{tt}(v)$, $\text{Con}(v, F, \mathbf{x}_j)$ is defined as

$$I(v, F, \mathbf{x}_j) = \frac{p_{t_m}^v - p_{t_s}^v}{M(p_{jf_m} - p_{jfs} + 1/M)}. \quad (10)$$

Here, $0 \leq \text{Con}(v, F, \mathbf{x}_j) \leq 1$. In this case, both v and F correctly classify $\mathbf{x}_j$. We treat $(p_{t_m}^v - p_{t_s}^v)/M$ as the net

Input: pruning set D' , forest $F = \{T_1, T_2, \dots, T_m\}$, where T_i is a decision tree.

Output: pruned forest F .

Method:

- (1) **for** each $\mathbf{x}_j \in D'$
- (2) Evaluate p_{jk} , $1 \leq k \leq K$;
- (3) **for** each $T_i \in F$ **do**
- (4) **for** each node v in T_i **do**
- (5) $I_v \leftarrow 0$;
- (6) **for** each $\mathbf{x}_j \in D'$ **do**
- (7) $q_{jk} \leftarrow p_{ijk}$, $1 \leq k \leq K$;
- (8) Let P be the path along which $\mathbf{x}_j$ travels;
- (9) **for** each node $v \in P$
- (10) $I_v \leftarrow I_v + I(v, F, \mathbf{x}_j)$;
- (11) PruningTree(root(T_i));
- (12) **for** each $\mathbf{x}_j \in D'$
- (13) $r_{jk} \leftarrow p_{ijk}$, $1 \leq k \leq K$;
- (14) $p_{jk} \leftarrow p_{jk} - q_{jk}/M + r_{jk}/M$

Procedure PruningTree(v)

- (1) **if** v is not a leaf **then**
- (2) $IG \leftarrow I_v$;
- (3) $I_{br(v)} \leftarrow 0$;
- (4) **for** each child c of v
- (5) PruningTree(c);
- (6) $I_{br(v)} \leftarrow I_{br(v)} + I_{br(c)}$;
- (7) $IG = I_{br(v)} - I_v$;
- (8) **if** $IG < \delta$ **then**
- (9) Prune subtree(v) and set v to be a leaf;

ALGORITHM 1: The procedure of forest pruning.

contribution of node v to F and $\mathbf{x}_j$, and $p_{jfm} - p_{jfs}$ as the weight of v 's net contribution.

For $\mathbf{x}_j \in e_{ft}(v)$, $\text{Con}(v, F, \mathbf{x}_j)$ is defined as

$$I(v, F, \mathbf{x}_j) = -\frac{p_{t_m}^v - p_{f_m}^v}{M(p_{jfm} - p_{jfs} + 1/M)}. \quad (11)$$

It is easy to prove $-1 \leq \text{Con}(v, F, \mathbf{x}_j) \leq 0$. This case is opposed to the first case. In this case, we treat $-(p_{t_m}^v - p_{f_m}^v)/M$ as the net contribution of node v to F and $\mathbf{x}_j$, and $p_{jfm} - p_{jfs}$ as the weight of v 's net contribution.

For $\mathbf{x}_j \in e_{ff}(v)$, $\text{Con}(v, F, \mathbf{x}_j)$ is defined as

$$I(v, F, \mathbf{x}_j) = -\frac{p_{t_m}^v - p_{y_j}^v}{M(p_{jfm} - p_{jy_j} + 1/M)}, \quad (12)$$

where $y_j \in \{1, \dots, K\}$ is the label of $\mathbf{x}_j$, $-1 \leq \text{Con}(v, F, \mathbf{x}_j) \leq 0$. In this case, both v and F incorrectly classify $\mathbf{x}_j$, namely, $t_m \neq y_j$ and $f_m \neq y_j$. We treat $-(p_{t_m}^v - p_{y_j}^v)/M$ as the net contribution of node v to F and $\mathbf{x}_j$, and $p_{jfm} - p_{jy_j}$ as the weight of v 's net contribution.

4.3. Algorithm. The specific details of forest pruning (FP) are shown in Algorithm 1, where

D' is a pruning set containing n instances,

p_{jk} is the probability that ensemble F predicts $\mathbf{x}_j \in D'$ associated with label k ,

p_{ijk} is the probability that current tree T_i predicts $\mathbf{x}_j \in D'$ associated with label k ,

I_v is a variant associated with node v to save v 's importance,

$I_{br(v)}$ is a variant associated with node v to save the contribution of branch(v).

FP first calculates the probability of F 's prediction on each instance $\mathbf{x}_j$ (lines (1)~(2)). Then it iteratively deals with each decision tree T_i (lines (3)~(14)). Lines (4)~(10) calculate the importance of each node $v \in T_i$, where $I(v, F, \mathbf{x}_j)$ in line (10) is calculated using one of the equations (9)~(12) based on the four cases in equation (8). Line (11) calls PruningTree(v) to recursively prune T_i . Since forest F has been changed after pruning T_i , we adjust F 's prediction in lines (12)~(14). Lines (3)~(14) can be repeated many times until all decision trees can not be pruned. Experimental results show that forest performance is stable after this iteration is executed 2 times.

The recursive procedure PruningTree(v) adopts a bottom-up fashion to prune the decision tree with v as the root. After pruning branch(v) (subtree(v)), I_v saves the sum of the importance of leaf nodes in branch(v). Then $I(\text{branch}(v), F)$ is equal to the sum of importance of the tree with v as root. The essence of using T_i 's root to call

PruningTree is to travel T_i . If current node v is a nonleaf, the procedure calculates v 's importance gain IG, saves into I_v the importance sum of the leaves of $\text{branch}(v)$ (lines (2)~(7)), and determines pruning $\text{branch}(v)$ or not based on the difference between CG and the threshold value δ (lines (8)~(9)).

4.4. Discussion. Suppose pruning set D' contains n instances, forest F contains M decision trees, and $d_{\max}$ is the depth of the deepest decision tree in F . Let $|T_i|$ be the number of nodes in decision tree T_i , and $t_{\max} = \max_{1 \leq i \leq M} (|T_i|)$. The running time of FP is dominated by the loop from lines (4) to (19). The loop from lines (5) to (7) traverses T_i , which is can be done in $O(t_{\max})$; the loop from lines (8) to (14) searches a path of T_i for each instance in D' , which is complexity of $O(nd_{\max})$; the main operation of $\text{PruningTree}(\text{root}(T_i))$ is a complete traversal of T_i , whose running time is $O(t_{\max})$; the loop from lines (16) to (18) scans a linear list of length n in $O(n)$. Since $t_{\max} \ll nd_{\max}$, we conclude the running time of FP is $O(nMd_{\max})$. Therefore, FP is a very efficient forest pruning algorithm.

Unlike traditional metrics such as those used by CART [22] and C4.5 [23], the proposed measure uses a global evaluation. Indeed, this measure involves the prediction values that result from a majority voting of the whole ensemble. Thus, the proposed measure is based on not only individual prediction properties of ensemble members but also the complementarity of classifiers.

From equations (9), (10), (11), and (12), our proposed measure takes into account both the correctness of predictions of current classifier and the predictions of ensemble and the measure deliberately favors classifiers with a better performance in classifying the samples on which the ensemble does not work well. Besides, the measure considers not only the correctness of classifiers, but also the diversity of ensemble members. Therefore, using the proposed measure to prune an ensemble leads to significantly better accuracy results.

5. Experiments

5.1. Experimental Setup. 19 data sets of which the details are shown in Table 1 are randomly selected from UCI repertory [30], where #Size, #Attrs, and #Cls are the size, attribute number, and class number of each data set, respectively. We design four experiments to study the performance of the proposed method (forest pruning, FP):

- (i) The first experiment studies FP's performance versus the times of running FP. Here, four data sets, that is, autos, balance-scale, German-credit, and pima, are selected as the representatives, and each data set is randomly divided into three subsets with equal size, where one is used as the training set, one as the pruning set, and the other one as the testing set. We repeat 50 independent trials on each data set. Therefore a total of 300 trials of experiments are conducted.

TABLE 1: The details of data sets used in this paper.

Data set	#Attrs	#Size	#Cls
Australian	226	70	24
Autos	205	26	6
Backache	180	33	2
Balance-scale	625	5	3
Breast-cancer	268	10	2
Cars	1728	7	4
Credit-rating	690	16	2
German-credit	1000	21	2
Ecoli	336	8	8
Hayes-roth	160	5	4
Heart-c	303	14	5
Horse-colic	368	24	2
Ionosphere	351	35	2
Iris	150	5	3
Lymph	148	19	4
Page-blocks	5473	11	5
Pima	768	9	2
prnn-fglass	214	10	6
Vote	439	17	2

- (ii) The second experiment is to evaluate FP's performance versus FL's size (number of base classifiers). The experimental setup of data sets is the same as the first experiment.
- (iii) The third experiment aims to evaluate FP's performance on pruning ensemble constructed by bagging [1] and random forest [26]. Here, tenfold cross-validation is employed: each data set is divided into tenfold [31, 32]. For each one, the other ninefold is to train model, and the current one is to test the trained model. We repeat 10 times the tenfold cross-validation and thus, 100 models are constructed on each data set. Here, we set the training set as the pruning set. Besides, algorithm rank is used to further test the performance of algorithms [31–33]: on a data set, the best performing algorithm gets the rank of 1.0, the second best performing algorithm gets the rank of 2.0, and so on. In case of ties, average ranks are assigned.
- (iv) The last experiment is to evaluate FP's performance on pruning the subensemble obtained by ensemble selection method. EPIC [11] is selected as the candidate of ensemble selection methods. The original ensemble is a library with 200 base classifiers, and the size of subensembles is 30. The setup of data sets is the same as the third experiment.

In the experiments, bagging is used to train original ensemble, and the base classifier is J48, which is a Java implementation of C4.5 [23] from Weka [34]. In the third experiment, random forest is also used to build forest. In the last three experiments, we run FP two times.

FIGURE 2: Results on data sets. (a) Forest size (node number) versus the times of running FP. (b) Forest accuracy versus the times of running FP.

5.2. Experimental Results. The first experiment is to investigate the relationship of the performance of the proposed method (FP) and the times of running FP. In each trial, we first use bagging to learn 30 unpruned decision trees as a forest and then iteratively run lines (3)~(14) of FP many times to trim the forest. More experimental setup refers to Section 5.1. The corresponding results are shown in Figure 2, where the top four subfigures are the variation trend of forest nodes number with the iteration number increasing, and the bottom four are the variation trend of ensemble accuracy. Figure 2 shows that FP significantly reduces forests size (almost 40%~60% of original ensemble) and significantly improves their accuracy. However, the performance of FP is almost stable after two iterations. Therefore, we set iteration number to be 2 in the following experiments.

The second experiment aims at investigating the performance of FP on pruning forests with different scales. The number of decision trees grows gradually from 10 to 200. More experimental setup refers to Section 5.1. The experimental results are shown in Figure 3, where the top four subfigures are the comparison between pruned and unpruned ensembles with the growth of the number of decision trees, and the bottom four are the comparison of ensemble accuracy. As shown in Figure 3, for each data set, the rate of forest nodes pruned by FP keeps stable and forests accuracy improved by FP is also basically unchanged, no matter how many decision trees are constructed.

The third experiment is to evaluate the performance of FP on pruning the ensemble constructed by ensemble learning method. The setup details are shown in Section 5.1. Tables 2, 3, 4, and 5 show the experimental results of compared methods,

respectively, where Table 2 reports the mean accuracy and the ranks of algorithms, Table 3 reports the average ranks using nonparameter Friedman test [32] (using STAC Web Platform [33]), Table 4 reports the comparing results using post hoc with Bonferroni-Dunn (using STAC Web Platform [33]) of 0.05 significance level, and Table 5 reports the mean node number and standard deviations. Standard deviations are not provided in Table 2 for clarity. The column of “FP” of Table 2 is the results of pruned forest and, “bagging” and “random forest” are the results of unpruned forests constructed by bagging and random forest, respectively. In Tables 3 and 4, Alg1, Alg2, Alg3, Alg4, Alg5, and Alg6 indicate PF pruning bagging with unpruned C4.5, bagging with unpruned C4.5, PF pruning bagging with pruned C4.5, bagging with pruned C4.5, PF pruning random forest, and random forest. From Table 2, FP significantly improves ensemble accuracy in most of the 19 data sets, no matter whether the individual classifiers are pruned or unpruned, no matter whether the ensemble is constructed by bagging or random forest. Besides, Table 2 shows that the ranks of FP always take place of best three methods in these data sets. Tables 3 and 4 validate the results in Table 2, where Table 3 shows that the average rank of PF is much small than other methods and Table 4 shows that, compared with other methods, PF shows significant better performance. Table 5 shows FP is significantly smaller than bagging and random forest, no matter whether the individual classifier is pruned or not.

The last experiment is to evaluate the performance of FP on pruning subensembles selected by ensemble selection method EPIC. Table 6 shows the results on the 19 data sets, where left and right are the accuracy and size, respectively. As shown in Table 6, FP can further significantly improve the

FIGURE 3: Results on data sets. (a) Forest size (node number) versus the number of decision trees. (b) Forest accuracy versus the number of decision trees. Solid curves and dash curves represent the performance of FP and bagging, respectively.

TABLE 2: The accuracy of FP, bagging, and random forest. • represents that FP outperforms bagging in pairwise t -tests at 95% significance level and denotes that FP is outperformed by bagging.

Dataset	Unpruned C4.5		Pruned C4.5		PF	RF
	PF	Bagging	PF	Bagging		
Australian	87.14 (2.0)	86.09 (5.0)*	86.80 (3.0)	85.86 (6.0)*	87.21 (1.0)	86.14 (4.0)*
Autos	74.40 (2.0)	73.30 (4.0)*	74.20 (3.0)	73.20 (5.0)*	74.72 (1.0)	73.10 (6.0)*
Backache	85.07 (3.0)	83.17 (5.5)*	85.89 (1.0)	83.17 (5.5)*	85.21 (2.0)	83.22 (4.0)*
Balance-scale	78.89 (3.0)	75.07 (6.0)*	79.79 (1.0)	76.64 (4.0)*	79.65 (2.0)	76.32 (5.0)*
Breast-cancer	69.98 (2.0)	67.10 (5.0)*	69.97 (3.0)	66.58 (6.0)*	70.11 (1.0)	68.88 (4.0)*
Cars	86.51 (4.0)	86.78 (2.0)	86.88 (1.0)	86.28 (5.0)	86.55 (3.0)	86.11 (6.0)
Credit-rating	86.44 (2.0)	85.54 (4.0)*	86.34 (3.0)	85.43 (5.0)*	86.82 (1.0)	85.42 (6.0)*
German-credit	75.33 (1.0)	73.83 (4.0)*	74.86 (3.0)	73.11 (6.0)*	75.22 (2.0)	73.18 (5.0)*
Ecoli	84.47 (2.0)	83.32 (6.0)*	84.20 (3.0)	83.40 (5.0)*	84.52 (1.0)	83.89 (4.0)*
Hayes-roth	78.75 (3.0)	78.63 (5.0)	78.77 (1.0)	76.31 (6.0)*	78.76 (2.0)	77.77 (4.0)
Heart-c	80.94 (2.0)	80.34 (5.0)	81.01 (1.0)	80.27 (6.0)	80.90 (3.0)	80.87 (4.0)
Horse-colic	84.52 (1.0)	83.29 (6.0)*	84.33 (2.0)	83.42 (5.0)*	84.31 (3.0)	83.99 (4.0)
Ionosphere	93.99 (1.0)	93.93 (2.0)	93.59 (6.0)	93.71 (4.0)	93.87 (3.0)	93.56 (5.0)
Iris	93.55 (6.0)	94.24 (4.0)	94.52 (3.0)	94.53 (2.0)	94.21 (5.0)	94.62 (1.0)
Lymphography	83.81 (5.0)	83.43 (6.0)	84.55 (2.0)	84.53 (3.0)	84.38 (4.0)	84.82 (1.0)
Page-blocks	97.03 (4.5)	97.04 (2.5)	97.04 (2.5)	97.06 (1.0)	97.03 (4.5)	97.01 (6.0)
Pima	75.09 (3.0)	74.27 (4.0)*	75.46 (1.0)	74.06 (5.0)*	75.43 (2.0)	73.21 (6.0)*
prnn-fglass	78.14 (4.0)	78.46 (1.0)	77.62 (6.0)	77.84 (5.0)	78.18 (3.0)	78.32 (2.0)
Vote	95.77 (1.0)	95.13 (6.0)*	95.67 (3.0)	95.33 (4.0)	95.72 (2.0)	95.31 (5.0)

TABLE 3: The ranks of algorithms using Friedman test, where Alg1, Alg2, Alg3, Alg4, Alg5, and Alg6 indicate PF pruning bagging with unpruned C4.5, bagging with unpruned C4.5, PF pruning bagging with pruned C4.5, bagging with pruned C4.5, PF pruning random forest, and random forest.

Algorithm	Alg5	Alg3	Alg1	Alg2	Alg6	Alg4
Ranks	2.39	2.50	2.71	4.32	4.42	4.66

TABLE 4: The testing results using post hoc, Alg1, Alg2, Alg3, Alg4, Alg5, and Alg6 indicate PF pruning bagging with unpruned C4.5, bagging with unpruned C4.5, PF pruning bagging with pruned C4.5, bagging with pruned C4.5, PF pruning random forest, and random forest.

Comparison	Statistic	<i>p</i> value
Alg1 versus Alg2	2.64469	0.04088
Alg3 versus Alg4	3.55515	0.00189
Alg5 versus Alg6	3.33837	0.01264

TABLE 5: The size (node number) of PF and bagging. • denotes that the size of PF is significantly smaller than the corresponding comparing method.

Dataset	Unpruned C4.5		Pruned C4.5		PF-RF	RF
	PF	Bagging	PF	Bagging		
Australian	4440.82 ± 223.24	5950.06 ± 210.53*	2194.71 ± 99.65	2897.88 ± 98.66*	1989.67 ± 99.65	2653.88 ± 99.61*
Autos	1134.83 ± 193.45	1813.19 ± 183.49*	987.82 ± 198.22	1523.32 ± 193.22*	954.26 ± 198.22	1429.12 ± 182.21*
Backache	1162.79 ± 96.58	1592.80 ± 75.97*	518.77 ± 40.49	764.24 ± 37.78*	522.74 ± 40.49	789.23 ± 45.62*
Balance-scale	3458.52 ± 74.55	4620.58 ± 78.20*	3000.44 ± 71.76	3762.60 ± 65.55*	2967.44 ± 71.76	3763.19 ± 79.46*
Breast-cancer	2164.64 ± 156.41	3194.20 ± 144.95*	843.96 ± 129.44	1189.33 ± 154.08*	886.66 ± 129.44	1011.21 ± 148.92*
Cars	1741.68 ± 60.59	2092.20 ± 144.95*	1569.11 ± 57.55	1834.91 ± 46.80*	1421.32 ± 56.65	1899.92 ± 68.88*
Credit-rating	4370.65 ± 219.27	5940.51 ± 223.51*	2168.11 ± 121.51	2904.40 ± 99.73*	2015.21 ± 140.58	2650.40 ± 102.13*
German-credit	9270.75 ± 197.62	11464.19 ± 168.63*	4410.11 ± 114.94	5421.60 ± 107.24*	4311.54 ± 124.68	5340.60 ± 217.48*
Ecoli	1366.62 ± 61.68	1736.52 ± 64.91*	1304.30 ± 54.39	1611.02 ± 56.31*	1324.30 ± 54.42	1820.02 ± 88.74*
Hayes-roth	498.65 ± 28.99	697.58 ± 40.87*	272.30 ± 45.11	308.48 ± 53.86*	264.24 ± 46.46	299.48 ± 63.84*
Heart-c	1503.46 ± 65.47	1946.94 ± 62.52*	647.89 ± 102.15	974.93 ± 129.83*	647.89 ± 102.15	1032.93 ± 111.57*
Horse-colic	2307.67 ± 106.99	3625.23 ± 116.63*	684.29 ± 106.35	974.93 ± 129.83*	647.89 ± 102.15	743.25 ± 120.43*
Ionosphere	552.49 ± 61.41	680.43 ± 69.95*	521.83 ± 58.01	634.73 ± 64.44*	542.58 ± 96.02	665.84 ± 66.44*
Iris	168.46 ± 111.12	222.66 ± 150.42*	144.52 ± 97.26	191.84 ± 133.12*	133.24 ± 98.32	212.55 ± 129.47*
Lymphography	1089.87 ± 67.16	1394.37 ± 61.85*	711.62 ± 37.61	856.44 ± 30.83*	724.53 ± 37.61	924.33 ± 50.78*
Page-blocks	1420.05 ± 278.51	2187.45 ± 555.02*	1394.11 ± 600.06	2092.93 ± 403.79*	1401.11 ± 588.03	2134.40 ± 534.97*
Pima	2202.41 ± 674.18	2776.77 ± 852.95*	2021.19 ± 698.02	2481.64 ± 747.19*	1927.67 ± 625.27	2521.43 ± 699.82*
prnn-fglass	1219.98 ± 39.85	1398.62 ± 36.29*	1145.20 ± 39.76	1269.28 ± 35.52*	1098.18 ± 34.26	1314.05 ± 60.97*
Vote	303.06 ± 124.00	527.80 ± 225.05*	174.04 ± 77.61	276.00 ± 127.46*	182.14 ± 76.21	288.33 ± 113.76*

accuracy of subensembles selected by EPIC and reduce the size of the subensembles.

6. Conclusion

An ensemble with decision trees is also called forest. This paper proposes a novel ensemble pruning method called forest pruning (FP). FP prunes trees' branches based on the proposed metric called branch importance, which indicates the importance of a branch (or a node) with respect to the whole ensemble. In this way, FP achieves reducing ensemble size and improving the ensemble accuracy.

The experimental results on 19 data sets show that FP significantly reduces forest size and improves its accuracy in most of the data sets, no matter whether the forests are the ensembles constructed by some algorithm or the

subensembles selected by some ensemble selection method, no matter whether each forest member is a pruned decision tree or an unpruned one.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is in part supported by the National Natural Science Foundation of China (Grant nos. 61501393 and 61402393), in part by Project of Science and Technology Department of Henan Province (nos. 162102210310, 172102210454, and 152102210129), in part by Academics

TABLE 6: The performance of FP on pruning subensemble obtained by FP on bagging. • represents that FP is significantly better (or smaller) than EPIC in pairwise t -tests at 95% significance level and denotes that FP is significantly worse (or larger) than EPIC.

Dataset	Error rate		Size	
	PF	EPIC	PF	EIPC
Australian	86.83 $\pm$ 3.72	86.22 $\pm$ 3.69*	2447.50 $\pm$ 123.93	3246.16 $\pm$ 116.07*
Autos	84.83 $\pm$ 4.46	82.11 $\pm$ 5.89*	708.01 $\pm$ 54.55	931.44 $\pm$ 51.16*
Backache	84.83 $\pm$ 4.46	82.11 $\pm$ 5.89*	708.01 $\pm$ 54.55	931.44 $\pm$ 51.16*
Balance-scale	79.74 $\pm$ 3.69	78.57 $\pm$ 3.82*	3277.76 $\pm$ 85.07	4030.82 $\pm$ 94.67*
Breast-cancer	70.26 $\pm$ 7.24	67.16 $\pm$ 8.36*	843.96 $\pm$ 129.44	1189.33 $\pm$ 154.08*
Cars	87.02 $\pm$ 5.06	86.83 $\pm$ 5.04	178.32 $\pm$ 60.44	2022.81 $\pm$ 53.19*
Credit-rating	86.13 $\pm$ 3.92	85.61 $\pm$ 3.95*	2414.60 $\pm$ 123.66	3226.25 $\pm$ 131.46*
German-credit	74.98 $\pm$ 3.63	73.13 $\pm$ 4.00*	4410.11 $\pm$ 114.94	6007.28 $\pm$ 124.30*
Ecoli	83.77 $\pm$ 5.96	83.24 $\pm$ 5.98*	1498.86 $\pm$ 62.27	1806.26 $\pm$ 70.98*
Hayes-roth	78.75 $\pm$ 9.57	76.81 $\pm$ 9.16*	275.09 $\pm$ 47.90	311.32 $\pm$ 57.05*
Heart-c	81.21 $\pm$ 6.37	79.99 $\pm$ 6.65*	1230.14 $\pm$ 54.80	1510.57 $\pm$ 52.56*
Horse-colic	84.53 $\pm$ 5.30	83.80 $\pm$ 6.11*	940.07 $\pm$ 66.64	1337.60 $\pm$ 75.73*
Ionosphere	93.90 $\pm$ 4.05	94.02 $\pm$ 3.83	590.63 $\pm$ 65.62	706.79 $\pm$ 73.17*
Iris	94.47 $\pm$ 5.11	94.47 $\pm$ 5.02	152.58 $\pm$ 108.04	197.80 $\pm$ 141.31*
Lymphography	81.65 $\pm$ 9.45	81.46 $\pm$ 9.39	858.42 $\pm$ 46.50	1022.67 $\pm$ 39.68*
Page-blocks	97.02 $\pm$ 0.74	97.07 $\pm$ 0.69	1396.63 $\pm$ 237.03	2086.89 $\pm$ 399.10*
Pima	74.92 $\pm$ 3.94	74.03 $\pm$ 3.58*	2391.95 $\pm$ 764.16	2910.31 $\pm$ 936.70*
prnn-fglass	78.13 $\pm$ 8.06	77.99 $\pm$ 8.44	1280.14 $\pm$ 43.85	1410.84 $\pm$ 39.59*
Vote	95.70 $\pm$ 2.86	95.33 $\pm$ 2.97	177.36 $\pm$ 86.10	281.62 $\pm$ 140.60*

Propulsion Technology Transfer projects of Xi'an Science and Technology Bureau [CX1516(6)], and in part by Nanhu Scholars Program for Young Scholars of XYNU.

References

- [1] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [2] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, part 2, pp. 119–139, 1997.
- [3] D. Zhang, S. Chen, Z. Zhou, and Q. Yang, "Constraint projections for ensemble learning," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI '08)*, pp. 758–763, Chicago, Ill, USA, July 2008.
- [4] T. G. Dietterich, "Ensemble methods in machine learning," in *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pp. 1–15, Cagliari, Italy, June 2000.
- [5] Z. Zhou, Y. Wang, Q. J. Wu, C. N. Yang, and X. Sun, "Effective and efficient global context verification for image copy detection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 48–63, 2017.
- [6] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.
- [7] Z. Zhou, C.-N. Yang, B. Chen, X. Sun, Q. Liu, and Q. M. J. Wu, "Effective and efficient image copy detection with resistance to arbitrary rotation," *IEICE Transactions on information and systems*, vol. E99-D, no. 6, pp. 1531–1540, 2016.
- [8] W. M. Zhi, H. P. Guo, M. Fan, and Y. D. Ye, "Instance-based ensemble pruning for imbalanced learning," *Intelligent Data Analysis*, vol. 19, no. 4, pp. 779–794, 2015.
- [9] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.
- [10] G. Martinez-Muñoz, D. Hernández-Lobato, and A. Suarez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 245–259, 2009.
- [11] Z. Lu, X. D. Wu, X. Q. Zhu, and J. Bongard, "Ensemble pruning via individual contribution ordering," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pp. 871–880, Washington, DC, USA, July 2010.
- [12] L. Guo and S. Boukir, "Margin-based ordered aggregation for ensemble pruning," *Pattern Recognition Letters*, vol. 34, no. 6, pp. 603–609, 2013.
- [13] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.
- [14] B. Krawczyk and M. Woźniak, "Untrained weighted classifier combination with embedded ensemble pruning," *Neurocomputing*, vol. 196, pp. 14–22, 2016.
- [15] C. Qian, Y. Yu, and Z. H. Zhou, "Pareto ensemble pruning," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 2935–2941, Austin, Tex, USA, January 2015.
- [16] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "Ensemble diversity measures and their application to thinning," *Information Fusion*, vol. 6, no. 1, pp. 49–62, 2005.
- [17] W. M. Zhi, H. P. Guo, and M. Fan, "Energy-based metric for ensemble selection," in *In Proceedings of 14th Asia-Pacific Web Conference*, vol. 7235, pp. 306–317, Springer Berlin Heidelberg, Kunming, China, April 2012.

- [18] Q. Dai and M. L. Li, "Introducing randomness into greedy ensemble pruning algorithms," *Applied Intelligence*, vol. 42, no. 3, pp. 406–429, 2015.
- [19] I. Partalas, G. Tsoumakas, and I. Vlahavas, "A Study on greedy algorithms for ensemble pruning," Tech. Rep. TR-LPIS-360-12, Dept. of Informatics, Aristotle University of Thessaloniki, Greece, 2012.
- [20] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *Proceedings of the 14th International Conference on Machine Learning*, pp. 211–218, Nashville, Tenn, September 1997.
- [21] Q. Dai, T. Zhang, and N. Liu, "A new reverse reduce-error ensemble pruning algorithm," *Applied Soft Computing*, vol. 28, pp. 237–249, 2015.
- [22] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, Calif, USA, 1984.
- [23] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, Calif, USA, 1993.
- [24] G. I. Webb, "Further experimental evidence against the utility of occam's razor," *Journal of Artificial Intelligence Research*, vol. 4, pp. 397–417, 1996.
- [25] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [26] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: a new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006.
- [27] C. Yuan, X. Sun, and R. Lv, "Fingerprint liveness detection based on multi-scale LPQ and PCA," *China Communications*, vol. 13, no. 7, pp. 60–65, 2016.
- [28] I. Partalas, G. Tsoumakas, and I. P. Vlahavas, "Focused ensemble selection: a diversity-based method for greedy ensemble selection," in *In Proceeding of the 18th European Conference on Artificial Intelligence*, pp. 117–121, Patras, Greece, July 2008.
- [29] I. Partalas, G. Tsoumakas, and I. Vlahavas, "An ensemble uncertainty aware measure for directed hill climbing ensemble pruning," *Machine Learning*, vol. 81, no. 3, pp. 257–282, 2010.
- [30] A. Asuncion and D. Newman, "UCI machine learning repository," 2007.
- [31] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 6, pp. 1–30, 2006.
- [32] S. García and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.
- [33] I. Rodríguez-Fdez, A. Canosa, M. Mucientes, and A. Bugarín, "STAC: a web platform for the comparison of algorithms using statistical tests," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1–8, Istanbul, Turkey, August 2015.
- [34] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, Calif, USA, 2005.

Hindawi

Submit your manuscripts at
<https://www.hindawi.com>