

Ensemble Selection using Simulated Annealing Walking

[Zahra Sadat Taghavi - Hedieh Sajedi]

Abstract—Pruning an ensemble of classifiers is one of the most significant and effective issues in ensemble method topic. This paper presents a new ensemble pruning method inspired by upward stochastic walking idea. Our proposed method incorporates simulated annealing algorithm and forward selection method for selecting models through the ensemble according to the probabilistic steps. Experimental comparisons of the proposed method versus similar ensemble pruning methods on a heterogeneous ensemble of classifiers demonstrate that it leads to better predictive performance and small-sized pruned ensemble. One of the reasons of these promising results is more time which our method spends for finding the best models of ensemble compared with rivals.

Keywords—ensemble method, simulated annealing algorithm, forward selection method.

I. Introduction

Ensemble method [1] is one of the most important classification algorithms in machine learning scope. The main purpose of ensemble method is combining of various classification algorithms to benefit from their unique characteristics and also to reduce their errors. Nevertheless, ensemble method has notable characteristics; it is faced with the following problems:

- Low predictive performance
- High computational overhead
- Low diversity
- High communication overhead

To solve these problems, researches in [2, 3-12] beside two main phases of ensemble method, constructing ensemble and combining results, have considered an efficient extra phase for it. This intermediate phase is named ensemble pruning or ensemble selection. An ensemble pruning algorithm is able to discover useless models from initial ensemble and with dropping them leads to more effective pruned ensemble.

Through various ensemble pruning methods which have been proposed in recent years, the methods have been suggested which are based on optimization algorithms [6, 7-9, 11, 12] due to demonstration ensemble pruning as an NP-Complete problem in year 2000 [13]. These methods survey though the initial ensemble greedily, until the best subset of it is found. One defect of the methods in such category is that they always select the best models. In the other words, they never move down and toward the unvalued models. Therefore, the approaches may stick in local optimums.

Zahra Sadat Taghavi

Young Researchers and Elite Club, Qazvin Branch, Islamic Azad University Qazvin, Iran

Hedieh Sajedi

Dept. of Mathematics, Statistics and Computer Science, College of Science, University of Tehran
Tehran, Iran

In this paper, we suggest a new optimization-based ensemble pruning approach based on upward stochastic walking idea. The rationale behind our idea is to give value to stochastic movements and also to accept unvalued models according to an especial evaluation measure. This is a great help to escape from the local optimums. To implement our idea, we incorporate simulated annealing algorithm [14] and forward selection method [2, 6, 7, 8, and 12]. Experimental comparisons on our ensemble pruning method versus three ensemble pruning methods show very promising results.

The rest of this paper includes: Section 2 provides a review of the simulated annealing algorithm, pruned ensemble method, and forward selection method. Section 3 presents overview on previous works. Section 4 introduces our ensemble pruning approach. Section 5 describes implementation requirements and discusses obtained results from experiments. Finally, Section 6 concludes the paper.

II. Background

In the section, we explain the background of three methods, simulated annealing algorithm, pruned ensemble method, and forward selection method.

A. Simulated Annealing Algorithm

Simulated annealing algorithm (SA) [14] is a simple and effective meta-heuristic optimization algorithm. To solve an optimization problem, SA starts from an initial solution, and then by a heterogeneous Markov chain moves to the neighbor solutions until the best solution is found. In this chain, transition probability from a current solution to a next solution depends on an acceptance function. If the next solution according to this function be better than the current solution, transition is done with probability 1; otherwise, it is done with probability $\exp(\Delta E / \Theta)$, where ΔE is difference between values of acceptance function related to the next and the current solution, and Θ is a parameter called temperature. Θ is adjusted from high level of degrees at first and it is reduced with a special cooling schedule. These transitions are done until Θ is equal to the lowest temperature [15].

B. Pruned Ensemble Method

A pruned ensemble method has three different phases as follows. 1) Constructing initial ensemble containing N base classifier algorithms, $\{c_1 \dots c_n\}$. 2) Ensemble pruning that results $\{M_1 \dots M_k\}$. 3) Combining the classification output of selected models.

Ensemble Construction Approaches

An initial pool of N base classifiers is constructed via two main techniques, techniques based on dataset and techniques based on classifier [1]. In the first techniques, the base classifiers are selected homogeneously, then they are trained on different train datasets to construct desired initial ensemble. To produce different train datasets, applying

changes on samples, attributes, or class labels of the original datasets seems essential; e.g. Bagging [16], Boosting [17], and cross validation committee [18] are the most famous examples of methods that manipulate the samples. In the second techniques, a pool of heterogeneous base classifiers is trained on the same train dataset [1, 7, 8, and 12]. Using different classifiers with various parameters causes to benefit from superior characteristics of each of them and reducing predictive performance.

Ensemble Pruning Approaches

At this phase, a specific ensemble pruning algorithm receives N produced models from the previous phase. Then it selects K more proper models among them, $\{M_1 \dots M_k\}$, as pruned ensemble. Different ensemble pruning algorithms have been proposed, in recent decades, ranking-based, clustering-based, and optimization-based ensemble pruning approaches and so on. Each algorithm has own special attitude about this problem and solves it via a special procedure. In Section 3, we will present some of them.

Combination Approaches

At this stage, K models in the pruned ensemble classify given test instance, x_i , separately. Then, these results must be combined using a given combining algorithm for determining the class label of x_i . One of the most famous combination approaches is voting [19]. Voting has different types as follows. 1) Plurality voting; in this method each model M_i , ($i = 1 \dots t$), classifies x_i and specifies a possible class label for it. Afterwards, the class label with the highest vote in comparison to the rest of class labels is declared for x_i by the pruned ensemble. 2) Majority voting; this method is similar to the plurality voting with the difference that the class label is assigned to x_i which is obtained more than half of the overall votes. 3) Soft voting; in this way, each model M_i classifies test instance x_i , but instead of specifying a class label for it, products a probability output for it. The probable output is a vector with N_C elements, based on the number of classes. This job is done directly on the approximation the posterior probability. Then using a simple averaging on the probability vector values per class labels, results are combined. Finally using simple voting approach, the class label of test instance x_i is determined.

C. Forward Selection Method

Forward selection method (*FS*) is foundation of different ensemble pruning approaches such as [2, 6, 7, 8, and 12]. The method *FS* starts from an empty set of models and continues until a set of all models is achieved. At each step, most effective model is selected greedily to accumulate to the previous ensemble. Therefore, during the survey of ensemble search space, N current ensembles are produced, according to the number of initial ensemble classifiers. Finally, one of the current ensembles, which it is the most accurate ensemble among the others is selected as the pruned ensemble.

III. Related Works

The ensemble pruning problem is one of the interesting issues in machine learning scope which has been subject of many researches in recent decades. In this section, we explain a brief study on some of them.

Martinez and Suarez [5] introduce a ranking-based ensemble pruning algorithm called orientation ordering. In

this method, models of initial ensemble are ranked using an evaluation benchmark. Afterward, based on their obtained grade, some models are chosen to construct pruned ensemble. Giacinto et al. [4] prune an ensemble of classifiers using clustering methods. In this method firstly, similar models according to the predictive performance are separated using hierarchical agglomerative clustering method. Then a model from each cluster, which has maximum distance from the rest of clusters, is selected to construct more diverse final ensemble. Masoumi and Sajedi [3] introduce another clustering-based ensemble selection approach called *K*-means Based Classifier Selection (*KSBC*). In this approach, using the Bagging method, an initial ensemble of uniformed classifiers is constructed. Then, *KSBC* partitions them using *K*-means clustering method. Finally, one model is selected from each cluster to construct the effective final ensemble. Zhou and Tang [11] introduce an optimization-based ensemble pruning method, called *GASEN-b*. In this method, using standard operators of genetic algorithm, the ensemble pruning operation is done. Fitness function in this method is accuracy on a separate evaluation dataset. Caruana et al. [8] introduce an ensemble pruning method named forward stepwise selection (*FSS*). In *FSS*, a library of heterogeneous models is pruned using the method *FS*. The performance metrics which are used in *FSS* are accuracy, root-mean-squared-error, mean cross-entropy, lift, and so on. Partalas et al. [7] introduce the ensemble pruning method, which is aware of ensemble decisions uncertainty via directed hill climbing algorithm. In this method, which we called it, *UAEP*, using two search directions of directed hill climbing algorithm; forward selection and backward elimination, the pruning operation is done. In *UAEP* a diversity-based evaluation measure named Uncertainty Weighted Accuracy (*UWA*) is suggested. Taghavi and Sajedi [12] introduce a human-inspired ensemble pruning method using directed hill climbing algorithm. In this method, which we called it, *HIEP*, a diversity-based evaluation measure, named Human-Like Foresight (*HLF*), is suggested. The measure is capable to select more accurate models through initial ensemble, especially in forward selection direction.

IV. Our Approach

In the proposed ensemble pruning approach, we incorporate *FS* and *SA* algorithms. The rationale behind this proposal is that we claim the idea upward stochastic walking is almost more beneficial than the idea forward greedy stepwise, in models selection operations. It is worth to note that nevertheless *FS* is one of the basic methods has been architecture of many ensemble pruning approaches; its greedy nature almost causes to get stuck in local optimums. So, the rationale behind of *FS*, forward greedy stepwise idea, is unsuccessful in models selection. Therefore, we incorporate *SA* with *FS*, for profiting from the probabilistic nature of *SA* during the investigation of model. It causes to escape from the local optimums and to reach global optimum.

Fig. 1 illustrates our ensemble pruning method architecture. As shown in this figure, it starts from an empty current ensemble of base classifiers, Ens_0 , and it continues to reach a current ensemble of N base classifiers, Ens_N . In each step of the procedure, one model, M_{MAX} , is selected among remainder models using *SA* to aggregate to the previous current ensemble. At the end, the most accurate current ensemble, Ens_{BEST} , is suggested the pruned ensemble.

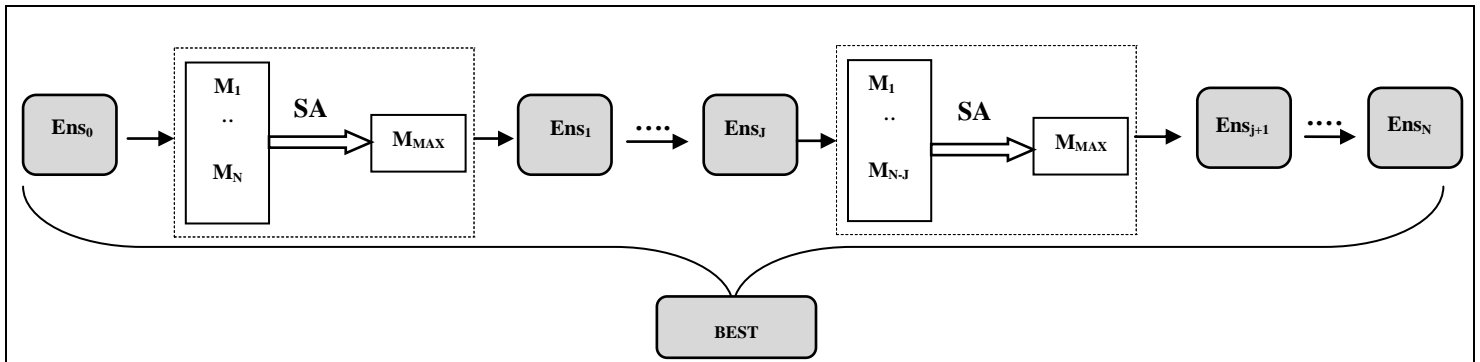


Figure1. Proposed ensemble pruning method architecture

SA is a parametric algorithm, which must be customized for ensemble pruning problem, particularly. For this purpose firstly, we present and adjust required parameters. Then, we explain proposed SA for solving ensemble pruning problem.

- **Neighborhood production.** For benefiting from stochastic movement, neighborhood is started from an initial random model, M_0 , and then is transferred to others, M_i , according to a random movement.
- **Temperature planning.** The parameter Θ is initialized with the value Θ_0 , then with a cooling function equal to Equation (1) it is reduced. The cooling process is continued until Θ becomes less than a specific value Θ_F as stopping criteria [15].

$$\Theta_i = \alpha * \Theta_{i-1} \quad (1)$$

- **Equilibrium condition.** The geometric function is used for determining the number of iterations per Θ . It is equal to Equation (2) and it is initialized with the value $Iter_0$, in the beginning of SA.

$$Iter_i = (1/\beta) * Iter_{i-1} \quad (2)$$

- **Acceptance function.** One of the most effective and significant parameter in SA is acceptance function. In the ensemble pruning problem, an especial evaluation measure plays the acceptance function role. We apply a diversity-based measure *HLF* [12].

The pseudo-code of the proposed algorithm is shown in Fig. 2. In suggested SA, an initial model is selected randomly, M_0 ; and it is evaluated with *HLF* evaluation measure, E_0 . These values are considered as maximum obtained values, M_{MAX} and E_{MAX} , respectively. Then for each Θ during the procedure, as equilibrium condition for Θ is met, the model M_i is generated. It is evaluated with the measure *HLF*, E_i . Afterward, the value of ΔE , ($E_i - E_{MAX}$), is calculated. If ΔE becomes greater than zero, M_i and E_i are assigned as M_{MAX} and E_{MAX} , respectively; otherwise with probability $\exp(\Delta E / \Theta)$, the job is done. After completing SA, its output that is a model, M_{MAX} , is added to pervious current ensemble.

V. Experiments

In this section, we evaluate our proposed method via comparison with three ensemble pruning methods, *FSS*, *UAEP*, and *HIEP*, mentioned in Section 3. For fair comparison between the performances of their pruning phase; we implemented the methods in the entirely same conditions of ensemble construction and combination phases.

First, we present the implementation requirements and then present and discuss the results.

```

Input: The parameter  $\alpha, \beta, Itr_0, \Theta_0, \Theta_F, ModelsSet$ 
Output:  $ENS_{BEST}$ 
Procedure:
while ( $ModelsSet \neq \emptyset$ ) do
   $M_0 = RandomSelection(ModelsSet)$ 
   $E_0 = Estimate\ HLF(Ens_C, M_0)$ 
  Set  $\Theta_C = \Theta_0; Itr_C = Itr_0; E_{MAX} = E_0; M_{MAX} = M_0$ 
  while ( $\Theta_C > \Theta_F$ ) do
     $i = 1$ 
    while ( $i < Itr_C$ ) do
       $M_i = RandomSelection(ModelsSet)$ 
       $E_i = Estimate\ HLF(Ens_C, M_i)$ 
      if ( $E_i > E_{MAX}$ ) then
        Set  $E_{MAX} = E_i; M_{MAX} = M_i$ 
      end
      else if ( $Random\ Real(0,1) \leq Math.exp((E_i - E_{MAX}) / \Theta_C)$ ) then
        Set  $E_{MAX} = E_i; M_{MAX} = M_i$ 
      end
       $i = i + 1$ 
    end of while
    Set  $\Theta_C = \alpha * \Theta_C; Itr_C = (1/\beta) * Itr_C$ 
  end of while
  Add  $M_{MAX}$  to  $Ens_C$ 
  Drop  $M_{MAX}$  from  $ModelsSet$ 
   $ACC_C = Estimate\ OverallAccuracy(Ens_C)$ 
  if ( $ACC_C > ACC_{MAX}$ ) then
    Set  $ACC_{MAX} = ACC_C; Enc_{BEST} = Ens_C$ 
  end
end of while
return  $ENS_{BEST}$ 

```

Figure2. Pseudo-Code of the proposed ensemble pruning algorithm

A. Implementation Requirements

We used 10 different machine learning problem retrieved from *UCI* machine learning repository [20]. In Table 1, the information of these datasets is shown. Note that, to construct train, pruning, and test datasets from each dataset, we needed to 60%, 20%, and 20% of its distinct samples, respectively. In ensemble construction phase, we used a pool of 100 different classifiers with various parameters to construct a heterogeneous ensemble. These classifiers were 28 Decision Tree (*DT*), 10 K-Nearest Neighbor (*KNN*), 30 Multi-Layer Perceptron (*MLP*), 2 Naive Bayes (*NB*), and 30 Support Vector Machine (*SVM*). The detail of the values of the parameters is shown in the Table 2. The unsaid parameters were adjusted with default values. In our proposed method, the parameter $\Theta_0 = 1200$, $\alpha = 0.9$, $\Theta_F = 0.01$, $Itr_0 = 20$, and $\beta = 0.95$. In combination phase, we used the soft voting combination method (mentioned in Section 2).

B. Results and Discussion

In this section, we present and analyze experiments' results, based on accuracy of the methods, size of pruned ensembles, and averaged running time.

TABLE1. Details ^A of Datasets

Dataset ID	UCI folder	N _I	N _A	N _C
DS ₁	Audiology (Standardized)	226	69	24
DS ₂	Balance-scale	625	4	3
DS ₃	Connectionist Bench	208	60	2
DS ₄	Contraceptive Method Choice (CMC)	1473	9	3
DS ₅	Ecoli	336	7	8
DS ₆	Glass Identification	214	9	7
DS ₇	Hepatitis	155	19	2
DS ₈	Iris	150	4	3
DS ₉	Primary Tumor	339	17	22
DS ₁₀	Statlog (Heart)	270	13	2

A. Dataset id, folder in UCI server, number of instances, attributes, and classes.

TABLE2. Details of base classifiers

Type	Value of Parameters	No
DT	➤ 10 Post Pruning Decision tree with: <ul style="list-style-type: none"> • Confidence Factor: 0.1, 0.2, 0.3, 0.4, 0.5 • Laplace Smoothing: True, False 	28
	➤ 16 Reduced Error Pruning Decision Tree <ul style="list-style-type: none"> • Number of Fold: 2, 3, 4, 5, 6, 7, 8, 9 • Laplace Smoothing: True, False 	
	➤ 2 Un-pruned Decision Tree <ul style="list-style-type: none"> • Minimum Object per Leaf: 2, 3 	
	➤ K=1~Plurality of Train dataset (5 different values)	
KNN	➤ Weighting Method: No-Weighting, Similarity Weighting	10
MLP	➤ Hidden Layer:1, 2, 4, 8, 16	30
	➤ Momentum Term: 0.2, 0.5, 0.9	
	➤ Learning Rate: 0.3, 0.6	
NB	➤ Kernel Estimator: True, False	2
SVM	➤ Complexity Parameter: 0.00001, 0.001, 0.1	30
	➤ Kernels: Polynomial Kernel with Degree: 2, 3, and Radial Kernel with Gamma: 0.001, 0.005, 0.01, 0.05, 0.1,0.5,1,2	

It is necessary to express that we carried out experiments 20 times on each dataset using production of four different situations of its samples for producing train, pruning, and test datasets. Then we repeated each situation five times for each ensemble pruning method. At the end, we averaged over the best gained answers of each situation per method.

Accuracy

Table 3 presents the percentage of accuracy of the four ensemble pruning algorithms, on each datasets. We applied bold font style for explicit display of the highest accuracy across each dataset. This table shows that our proposed method achieves the highest accuracy in 5 datasets. *FSS* just obtains the highest accuracy in 3 datasets. For general evaluating of accuracy of the ensemble pruning algorithms over all datasets, we averaged the obtained values over all datasets. We showed them in the last row of Table 3. According to these results, we can conclude that our proposed approach succeeds in correct recognition local optimums and it can escape from them well.

TABLE3. Accuracy of four ensemble pruning algorithms, on each dataset (%)

Dataset ID	FSS	HIEP	UAEP	Proposed Method
DS ₁	77.22	73.33	76.67	77.78
DS ₂	88.60	91.60	87.40	91.80
DS ₃	64.64	59.76	63.42	60.98
DS ₄	52.13	51.36	51.70	52.13
DS ₅	64.55	67.54	64.93	67.91

DS ₆	47.02	47.02	48.22	54.17
DS ₇	81.45	82.26	83.07	84.68
DS ₈	95.00	95.00	93.34	95.00
DS ₉	49.25	48.51	47.02	47.76
DS ₁₀	82.41	80.09	78.24	81.94
Average	70.227	69.647	69.401	71.415

Size

Tables 4-5 show the size of pruned ensemble via the four ensemble pruning algorithms, on each dataset. Beside the size of pruned ensembles, as shown in Tables 4-5, the number of models according to base classifiers' type were brought; number of *MLPs*, N_{MLP} , number of *KNNs*, N_{KNN} , number of *SVMs*, N_{SVM} , number of *DTs*, N_{DT} , and number of *NBs*, N_{NB} . These results show that *FSS* on 7 datasets, *HIEP* and *UAEP* on 3 datasets, and our proposed method on 2 datasets lead to smallest sized pruned ensemble. Furthermore, like previous criterion, for general evaluating of size of pruned ensemble via the ensemble pruning algorithms across all datasets, we averaged the obtained values over all datasets. Note that all algorithms lead to pruned ensemble in which number of models are less than 14% of size of initial ensemble.

Running Time

Table 6 shows running time of the methods which were averaged over all datasets. It shows that *FSS* with time 0.76 seconds is the fastest algorithm, and our proposed method with time 26 minutes and 42 seconds is the slowest algorithm.

The results show that our proposed method leads to increase predictive performance of the initial ensemble. Also, it succeeds in correct recognition local optimums and it can escape from them well, compared with other methods. Furthermore, it leads to reduce computational overhead, with dropping more redundant and useless models through the ensemble. It is necessary to note that pruning phase like training phase is accomplished one time and offline. Therefore, spending more time constrains more computational costs one time and more important, it causes to further search through initial ensemble.

VI. Conclusions

In this paper, a new ensemble pruning approach was proposed. In this method, forward selection method and simulated annealing algorithm are incorporated to reach the goal of "upward stochastic walking". Our method with aid of probabilistic steps selects models progressively until the best subset of them is found.

The proposed method was evaluated via comparison with three analogous ensemble pruning methods for pruning a heterogeneous initial ensemble of 100 classifiers on ten machine learning problems. The empirical experiments on the ensemble pruning approaches were accomplished and analyzed according to three important criteria, accuracy, size of pruned ensembles, and running time. Considering the accuracy criterion, the experimental results demonstrated that our method generally leads to most accurate pruned ensemble on 70% datasets. Considering the size criterion, our method generally leads to small-sized pruned ensemble, which includes less than 14% of initial ensemble models, in the case of averaging over all datasets. Finally, according to the

running time criterion, our method is time consuming. However, it is reasonable; especially when taking more time causes to find more effective subset from the initial ensemble.

[1]. P. N. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining," In Pearson Addison-Wesley, Upper Saddle River, NJ, 2006.

References

TABLE4. Size of pruned ensembles and their number of models according to the type of classifiers, for four ensemble pruning methods, on each dataset

DataSet ID	FSS						HIEP						UAEP					
	N_{MLP}	N_{KNN}	N_{SVM}	N_{DT}	N_{NB}	ALL	N_{MLP}	N_{KNN}	N_{SVM}	N_{DT}	N_{NB}	ALL	N_{MLP}	N_{KNN}	N_{SVM}	N_{DT}	N_{NB}	ALL
DS ₁	5	0	1	5	0	11	6	1	0	5	0	12	5	1	0	6	0	12
DS ₂	4	0	0	0	0	4	17	2	5	5	1	30	4	1	0	0	0	5
DS ₃	2	0	0	0	0	2	5	1	3	4	1	14	4	0	0	0	0	4
DS ₄	14	3	7	17	1	42	14	4	7	18	1	44	10	3	1	12	1	27
DS ₅	1	0	0	0	0	1	4	0	2	0	0	6	1	0	0	0	0	1
DS ₆	1	0	0	0	0	1	1	0	0	0	0	1	11	3	4	1	0	19
DS ₇	1	0	0	0	1	2	0	0	1	0	0	1	2	1	1	0	0	4
DS ₈	1	0	0	0	0	1	2	1	0	0	0	3	2	0	0	0	0	2
DS ₉	3	1	1	6	2	13	7	2	2	12	0	23	2	1	1	3	2	9
DS ₁₀	1	0	0	0	0	1	1	0	0	0	0	1	2	0	0	0	0	2
Avg.	3.3	0.4	0.9	2.8	0.4	7.8	5.7	1.1	2.0	4.4	0.3	13.5	4.3	1.0	0.7	2.2	0.3	8.5

TABLE5. Continued from Table 4

Dataset ID	Proposed Method					
	N_{MLP}	N_{KNN}	N_{SVM}	N_{DT}	N_{NB}	ALL
DS ₁	3	1	0	8	0	12
DS ₂	4	0	1	1	0	6
DS ₃	2	1	1	1	0	5
DS ₄	17	5	11	22	1	56
DS ₅	5	0	2	1	0	8
DS ₆	1	0	2	0	0	3
DS ₇	0	0	1	0	0	1
DS ₈	2	0	1	0	0	3
DS ₉	3	2	1	10	2	18
DS ₁₀	1	0	0	0	0	1
Avg.	3.8	0.9	2.0	4.3	0.3	11.3

TABLE6. Averaged running time of four ensemble pruning methods over all datasets

FSS	HIEP	UAEP	Proposed Method
0.76 sec	1.45 sec	1.21 sec	26 min, 42 sec

[2]. D. Margineantu, and T. Dietterich, "Pruning Adaptive Boosting," In Proceedings of the 14th International Conference on Machine Learning, 1997.

[3]. E. Masoumi-Nogorabi and H. Sajedi, "Increasing Classifier Ensemble Efficiency using KSBC Algorithm", In International Journal of Computer Applications, vol. 70, no.13, May, 2013.

[4]. G. Giacinto, F. Roli, G. Fumera, "Design of Effective Multiple Classifier Systems by Clustering of Classifiers," In 15th International Conference on Pattern Recognition, ICPR, 2000.

[5]. G. Martinez-Munoz, and A. Suarez, "Pruning in Ordered Bagging Ensembles," In 23rd International Conference in Machine Learning, ICML, 2006.

[6]. G. Martinez-Munoz, and A. Suarez, "Aggregation Ordering in Bagging," In International Conference on Artificial Intelligence and Applications (IASTED), 2004.

[7]. I. Partalas, G. Tsoumakas, and I. Vlahavas, "An Ensemble Uncertainty Aware Measure for Directed Hill Climbing Ensemble Pruning," In Machine Learning, Springer, vol. 81, pp. 257-282, 2010.

[8]. R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble Selection from Libraries of Models," In Proceedings of the 21st International Conference on Machine Learning, 2004.

[9]. R. E. Banfield, L.O. Hall, K.W. Bowyer, and W.P. Kegelmeyer, "Ensemble Diversity Measures and Their Application to Thinning," In Information Fusion, vol. 6, no. 1, pp. 49-62, 2005.

[10]. Z. H. Zhou, J.Wu, and W.Tang, "Ensemble Neural Networks: Many Could Be Better Than All," In Artificial Intelligence, vol. 137, no. 1-2, pp. 239-263, 2002.

[11]. Z. H. Zhou, and W. Tang, "Selective Ensemble of Decision Trees," In Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC, pp. 476-483, 2003.

[12]. Z. S. Taghavi and H. Sajedi, "Human- Inspired Ensemble Pruning using Hill Climbing Algorithm," In 5th Robocup IranOpen International Symposium and 3rd Joint Conference of Robotices & AI, April,2013.

[13]. C. Tamon, and J. Xiang, "On the Boosting Pruning Problem," In 11th European Conference on Machine Learning, ECML, Springer, 2000.

[14]. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing," In Science Journal, Vol.220, No. 4598, pp. 671-680, 1983.

[15]. E. Talbi, "Metaheuristics: From Desing to Implementation," In John Wiley and sons, Hoboken, 2009.

[16]. L. Breiman, "Bagging Predictors," In Machine Learning, vol. 24, pp. 123-140, 1996.

[17]. R. E. Schapire, "The Strength of Weak Learnability," In Machine Learning, vol. 5, pp. 197-227, 1990.

[18]. B. Parmanto, P.Munro, H. Doyle, "Improving Committee Diagnosis with Resampling Techniques," In D. Touretzky, M. Mozer, M. Hasselmo (Ed.): Advances in Neural Information Processing Systems, Vol. 8, 882-888, 1996.

[19]. Z. H. Zhou, "Ensemble Methods: Foundations and Algorithms," CRC press, Taylor & Francis Group, LLC, 2012.

[20]. The UCI machine learning raspository, URL: {http://archive.ics.uci.edu/ml/datasets.html}.