



# Stochastic ensemble pruning method via simulated quenching walking

Zahra Sadat Taghavi<sup>1</sup> · Seyed Taghi Akhavan Niaki<sup>2</sup> · Amir Hossein Niknamfar<sup>1</sup>

Received: 8 February 2018 / Accepted: 20 December 2018 / Published online: 2 January 2019  
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

Inspired by an upward stochastic walking idea, a new ensemble pruning method called simulated quenching walking (*SQWALKING*) is developed in this paper. The rationale behind this method is to give values to stochastic movements as well as to accept unvalued solutions during the investigation of search spaces. *SQWALKING* incorporates simulated quenching and forward selection methods to choose the models through the ensemble using probabilistic steps. Two versions of *SQWALKING* are introduced based on two different evaluation measures; *SQWALKING<sub>A</sub>* that is based on an accuracy measure and *SQWALKING<sub>H</sub>* that is based on a human-like foresight measure. The main objective is to construct a proper architecture of ensemble pruning, which is independent of ensemble construction and combination phases. Extensive comparisons between the proposed method and competitors in terms of heterogeneous and homogeneous ensembles are performed using ten datasets. The comparisons on the heterogeneous ensemble show that *SQWALKING<sub>H</sub>* and *SQWALKING<sub>A</sub>* can lead respectively to 5.13% and 4.22% average accuracy improvement. One reason for these promising results is the pruning phase that takes additional time to find the best models compared to rivals. Finally, the proposed *SQWALKINGs* are also evaluated on a real-world dataset.

**Keywords** Ensemble pruning method · Simulated quenching algorithm · Forward selection method · Simulated annealing algorithm · Genetic algorithm

## 1 Introduction

Ensemble methods [1–3] are significant classification algorithms in machine learning scope. The main purpose of the ensemble method is combining various classification algorithms; to benefit from their unique characteristics and also reduce their errors. Nevertheless, the ensemble method has notable characteristics; it is faced with problems such as

reducing the prediction performance and increasing computational and communication overheads [4].

To handle the abovementioned challenges, besides two main phases of ensemble method (constructing ensemble and combining the results); recent researches such as [5–7] have considered an efficient extra phase for the ensemble method. This intermediate phase is named ensemble pruning, ensemble selection, and also ensemble thinning, or selective ensemble. Moreover, a theorem so-called many could be better than all [7], signifies the necessity of this phase. This theorem indicates that applying some classifiers instead of all of them may be a better choice. Ensemble pruning phase is a specific algorithm, which is able to discover redundant and less accurate models from the initial ensemble. By dropping these models, the more effective pruned ensemble is achieved.

Generally, pruning phase has some discussable advantages as follows,

- The initial ensemble may be composed of diverse models with low or high prediction performance. This variety

✉ Amir Hossein Niknamfar  
niknamfar@qiau.ac.ir; niknamfar@yahoo.com

Zahra Sadat Taghavi  
taghavi.cese@gmail.com

Seyed Taghi Akhavan Niaki  
Niaki@Sharif.edu

<sup>1</sup> Young Researchers and Elite Club, Qazvin Branch, Islamic Azad University, Qazvin, Iran

<sup>2</sup> Department of Industrial Engineering, Sharif University of Technology, Azadi Ave., P.O. Box 11155-9414, Tehran 1458889694, Iran

may effect on the overall prediction performance of the initial ensemble and reduce it. However, by applying the pruning phase to the initial ensemble, the prediction performance of the initial ensemble is increased by dropping less accurate models.

- The existence of many models in the initial ensemble constrains high computational overheads, e.g. decision tree models require a lot of memory, or memory-based models have high computational costs during the execution. However, by applying the pruning phase to the initial ensemble, redundant models from the initial ensemble are dropped. Therefore, pruned ensemble leads to lower computational overhead.
- The initial ensemble may consist of very similar models to each other. Pruning the ensemble increases the diversity of ensemble classifiers and also increases the error correction capabilities.
- In the distributed data mining systems in which the ensemble method is applied as a classification algorithm, models are distributed over the network. Therefore, combining these models constrains additional communication overheads [4]. By applying the pruning phase, the overheads may be reduced.

It should be noted that, in the recent years, some novel ensemble combination approaches have been proposed for the ensemble system, e.g. [8–11]. These approaches without pruning initial ensemble, try to produce a reliable ensemble applying novel combination approaches based on weighted voting. These approaches can be applied on the single and multi-objective optimization algorithms [12] to quantify the number of votes for each base classifier. These approaches are adaptive to many ensemble construction methods and can be applied to solve many classification problems. Nevertheless, these approaches improve the overall prediction performance of the ensemble systems; they do not decrease the overall overheads of systems, as they do not eliminate models completely from the ensemble. However, if the key objective is only increasing the overall prediction performance on the given classification problem, these mentioned approaches are proper to use. However, if beside the increasing prediction performance, decreasing overall overheads of the ensemble system is also an objective; these approaches cannot be effective. Nevertheless, these combination methods can be done with many adaptive ensemble pruning approaches. Thus, this accompanying gains these objectives at the expense of spending more time.

Due to the efficiency and applicability of the pruning phase on many classification problems, in this study, an ensemble pruning problem is studied, where a new ensemble pruning approach is proposed. Through various ensemble pruning approaches proposed in recent years, some approaches have been suggested based on optimization

algorithms [5, 13–18], due to the proof of ensemble pruning as an *NP-Complete* problem [19]. These optimization-based ensemble pruning methods survey greedily through the initial ensemble until the best subset is found. One defect of these methods is that they always select models, which are the best based on an evaluation measure. In other words, they never move down and toward the unvalued models. Therefore, the greedy approaches may get stuck in local optima.

In this paper, a new optimization-based ensemble pruning approach is proposed. Our approach is based on the Upward Stochastic Walking idea. The rationale behind this idea is giving value to stochastic movements and also accepting unvalued solutions during the investigation of search space. This is a great help to escape from local optima and to reach the global optimum, almost certainly. To implement the idea, simulated quenching algorithm (*SQ*) [20] and forward selection method (*FS*) [5, 6, 14–17] is developed. This novel hybrid ensemble pruning approach is named *SQWALKING* (Simulated Quenching WALKING). Moreover, two versions of *SQWALKING*, namely *SQWALKING<sub>A</sub>* and *SQWALKING<sub>H</sub>*, are suggested. These versions are based on two different evaluation measures, accuracy, and human-like foresight [15], respectively. Evaluation measures are in fact the main functions for comparing and accepting models.

Here, the main contributions of our work are described.

1. *SQ* and *FS* are incorporated to create a new ensemble pruning algorithm. According to the best of our knowledge, incorporation of *SQ* and *FS* is a novel contribution in the scope.
2. Two different kinds of evaluation measures, namely performance-based and diversity-based measures, are proposed for *SQWALKING*. This procedure leads to create two versions of *SQWALKING*, namely Version *A* based on the accuracy measure and Version *H* based on the *HLF* (Human-like foresight) metric. Thereby, the performance of these measures is evaluated.
3. The structure of the *SQWALKING* is applicable to solve many classification problems. It is noteworthy to say that *SQWALKING* is a parameter-based approach where the parameters must be set meticulously. As such, its results on each classification problem can be dramatically improved when the parameters are calibrated. This certainly takes more time.
4. *SQWALKING* is independent of ensemble construction and combination phases. So, ensemble construction techniques, e.g. homogeneous initial ensemble, heterogeneous initial ensemble, or both can be applied. Moreover, effective combination methods, e.g. [8–11], can be applied.
5. *SQWALKING* can be considered as a tool to recognize more proper classifiers in many problems. That is, firstly,

a heterogeneous initial ensemble is constructed. Then, by applying *SQWALKING*, their best subset is selected. Afterward, the selected models in the pruned ensemble identify which types of classifiers are more effective to solve the given problem.

6. *SQWALKING* is run one time and in an offline manner in order to indicate the most efficient classifiers on a specific classifier problem. Then, the achieved pruned ensemble is applied over and over again.

This paper extends our previous work [16]. It elaborates our proposed ensemble pruning method, comprehensively. For a comprehensive evaluation of *SQWALKING*, this approach is compared with many existing approaches, experimentally. Such comprehensive studies are essential from different viewpoints. According to the viewpoints, the comparisons are categorized into three groups; (1) *Group A*: comparing *SQWALKING* to the other ensemble pruning methods. The objective of *Group A* is to evaluate the procedure of ensemble pruning phases. In this group, both heterogeneous and homogeneous initial ensembles are utilized. (2) *Group B*: comparing *SQWALKING* to the popular ensemble method without the pruning phase. (3) *Group C*: comparing *SQWALKING* to the common combination methods without the pruning phase. *Group B* and *C* help to specify whether the existence of pruning phase is effective or systems without this phase can also work well.

The rest of this paper includes Sect. 2 to provide a review of the simulated quenching algorithm, pruned ensemble method, and forward selection method. Section 3 presents an overview survey of past works in the ensemble pruning scope. In addition, it introduces some new ensemble combination approaches. Section 4 introduces our novel ensemble pruning approach. Section 5 and 6 present the experimental setups and the results, respectively. Section 7 discusses the obtained results from experiments. Section 8 tests the proposed *SQWALKING* on real-world data. Finally, Sect. 9 provides the conclusions and the future works.

## 2 Background

In this section, the background information on the simulated quenching algorithm, the pruned ensemble method, and the forward selection method are presented.

### 2.1 Simulated quenching algorithm

Simulated quenching (*SQ*) [20] is a member of the family of simulated annealing-like (*SA*) algorithms. The algorithm of *SQ* is shown in Fig. 1 [20]. As shown in Fig. 1, *SQ* receives search space of the problem. Then it finds the best solution (say  $S_{BEST}$ ) using stochastic movements and

---

**Function** *SQ* (Problem) **Returns** a state that is a local maximum

**Input:**  
 Problem Size ( $Size_p$ ); Initial Temperature ( $\theta_0$ ); Final Temperature ( $\theta_f$ );  
 Cooling Coefficient ( $\alpha$ ); Number of Iterations ( $Iter$ )

**Output:**  
 Best Solution ( $S_{BEST}$ )

**Process:**

1. Initialize  $\theta_0$
2.  $S_{INITIAL} \leftarrow CreateInitialSolution(Size_p)$
3.  $S_{BEST} \leftarrow S_{INITIAL}$
4.  $\theta_c \leftarrow \theta_0$
5. **while** ( $\theta_c > \theta_f$ ) **do**
  - a.  $i \leftarrow 1$
  - b. **while** ( $i < Iter$ ) **do**
    - i.  $S_{NEIGHBOR} \leftarrow CreateNeighborSolution(S_{BEST})$
    - ii.  $\Delta F \leftarrow Fitness(S_{NEIGHBOR}) - Fitness(S_{BEST})$  //according to the acceptance function
    - iii. **if** ( $\Delta F > 0$ ) **then**
      - $S_{BEST} \leftarrow S_{NEIGHBOR}$
    - iv. **else if** ( $RandomReal(0,1) \leq Exp(\Delta F / \theta)$ ) **then**
      - $S_{BEST} \leftarrow S_{NEIGHBOR}$
    - v.  $i \leftarrow i + 1$
  - c.  $\theta_c \leftarrow \alpha \times \theta_c$
6. **Return**  $S_{BEST}$

---

Fig. 1 The Pseudo-code of the simulated quenching algorithm

giving more value to bad solutions. For the purpose, *SQ* starts from an initial solution (say  $S_{INITIAL}$ ). Then, using the main loop, *SQ* moves toward a neighbor solution (say  $S_{NEIGHBOR}$ ). If  $S_{NEIGHBOR}$  is better than the best obtained solution ( $S_{BEST}$ ), based on a specific acceptance function, *SQ* selects  $S_{NEIGHBOR}$  as  $S_{BEST}$  with acceptance probability 1. Otherwise, this selection is done with acceptance probability around  $Exp(\Delta F / \theta)$  [20] (in the case of maximization problem). In this relation,  $\Delta F$  is the difference between the value of  $S_{NEIGHBOR}$  and  $S_{BEST}$ . And,  $\theta$  is a parameter so-called temperature. Note that, the acceptance function must be selected with respect to the optimization problem. The parameter  $\theta$  is important and effective from 3 viewpoints: (1) the rationale behind *SQ* is based on this parameter; (2) the parameter  $\theta$  determines the number of iterations of the main loop; (3) this parameter affects the acceptance probability of the worse solutions.

Thereupon, similar to *SA* [21], the analogy of *SQ* [20] is the annealing process of metal and their algorithms are the same. However, the main difference between the *SQ* and the *SA* is in the annealing schedule. So, the computational time of *SA* is very high, while *SQ* is high [20]. Both *SQ* and *SA* converge to a close optimal solution, in optimization problems.

### 2.2 Pruned ensemble method

Suppose there are  $N$  base classifier algorithms,  $\{c_1 \dots c_n\}$ . These classifiers are trained to construct initial ensemble. This ensemble is constructed via two main techniques; dataset-based techniques and classifier-based techniques [1, 2]. In the dataset-based techniques, the base classifiers are selected homogeneously. Then they are trained in different training datasets. To produce different training

datasets, instances, attributes, or class labels of original datasets may be manipulated. Bagging [22], Boosting [23], AdaBoost [24], and cross-validation committee [25] are the most famous examples of methods that manipulate the instances of the original datasets for the mentioned purpose. Random Forest [26] is a famous method that applies attributes manipulating technique. Error correcting output coding [1, 2] is another method that uses a class label manipulating technique. In the classifiers-based techniques, a pool of heterogeneous base classifiers, e.g. decision trees, artificial neural networks, and support vector machines, are trained on the same training dataset [1, 2, 5, 14–17].

Then the produced models in the first phase are injected into the pruning algorithm. This algorithm investigates through initial ensemble for finding  $K$  more effective models,  $\{M_1 \dots M_k\}$ . Different ensemble pruning methods have been proposed in recent decades. Each method has a special viewpoint about the ensemble pruning problem and solves it according to a special procedure. In Sect. 3, these pruning methods will be presented. Afterward, the  $K$  selected models classify given testing instance  $x_i$ , separately. Using a specific combining algorithm, the outputs of these models are combined next. Thus, the class label of  $x_i$ ,  $F(x_i)$ , is determined by the combining phase. There are different combining techniques such as plurality voting, majority voting, soft voting, weighted voting, and a mixture of experts [1, 27]. In Sect. 3, some novel methods based on the weighted voting will be presented.

### 2.3 Forward selection method

Forward selection method ( $FS$ ) is the foundation of different ensemble pruning methods such as [5, 6, 14–17]. In general,  $FS$  investigates through initial ensemble greedily for finding the best models. The general algorithm of  $FS$  is shown in Fig. 2. As illustrated in Fig. 2,  $FS$  starts from an empty current ensemble ( $Ens_C = \emptyset$ ). Then, until a stopping criterion is satisfied, it proceeds to obtain the best models (say  $M_{BEST}$ ) and to add them into  $Ens_C$ . In the end,  $Ens_C$  with the highest performance is selected as a pruned ensemble (say  $Ens_{BEST}$ ). At each step of the main loop, one model must be selected that is the best one, according to the specific evaluation measure. For this purpose, different evaluation measures could be applied. These measures are categorized into two main groups; performance-based measures and diversity-based measures. Performance-based measures maximize the performance of pruned ensemble. On the other hand, diversity-based measures emphasize on the diversity of models to reach pruned ensemble with high performance (See Sect. 3).

---

**Function**  $FS$  (Initial Ensemble) **Returns** Pruned Ensemble

**Input:**

Models' Set ( $Set_M$ ); Current Ensemble of Selected Models ( $Ens_C$ ); Performance of Current Ensemble ( $PRF_C$ ); Performance of Pruned Ensemble ( $PRF_{MAX}$ )

**Output:**

Pruned Ensemble ( $Ens_{BEST}$ )

**Process:**

1. Initialize  $Set_M = \text{Initial Ensemble}$ ,  $Ens_C = \emptyset$ ,  $PRF_C = 0$ ,  $PRF_{MAX} = 0$
  2. **Repeat**
    - a.  $M_{BEST} \leftarrow \text{FindBestModel}(Set_M)$  //according to the specific evaluation measure
    - b. Add  $M_{BEST}$  to  $Ens_C$
    - c. Drop  $M_{BEST}$  from  $Set_M$
    - d.  $PRF_C \leftarrow \text{EstimatePerformance}(Ens_C)$
    - e. **if** ( $PRF_C > PRF_{MAX}$ ) **then**
      - $Ens_{BEST} \leftarrow Ens_C$
      - $PRF_{MAX} \leftarrow PRF_C$
  3. **Until** (Stopping criterion is satisfied)
  4. **Return**  $Ens_{BEST}$
- 

Fig. 2 The pseudo-code of the forward selection method

## 3 Related works

In this section, a brief explanation of some ensemble pruning methods is presented. In addition, some novel combination methods will be described.

Xia et al. [28] presented an effective ordering-based ensemble pruning methods, which ranks all the base classifiers with respect to a Maximum Relevancy Maximum Complementary measure ( $MRMC$ ). This measure evaluates the base classifier's classification ability as well as its complementarity to the ensemble, and thereby a set of accurate and complementary base classifiers can be selected [28]. Guo et al. [29] proved that the ensemble pruning method should focus more on the following two factors: (1) examples with small absolute margin, and (2) classifiers that correctly classify more examples and contribute larger diversity. Based on this principle, they proposed a novel metric so-called the Margin and Diversity-based Measure ( $MDM$ ) to explicitly evaluate the importance of individual classifiers. By incorporating ensemble members in a decreasing order based on the  $MDM$ , sub ensembles are formed such that users can select the top  $T$  ensemble members for predictions [29]. Li et al. [30] proposed a novel ensemble pruning algorithm so-called  $RTCRelief-F$  and applied it to facial expression recognition.  $RTCRelief-F$  uses a novel classifier representation method that accounts for the interaction among classifiers and bases the classifier selection upon not only diversity but accuracy.  $RTCRelief-F$  applies the  $Relief-F$  algorithm to evaluate the classifiers' ability and resets the fusion order. Then, the combination of a clustering-based ensemble pruning method and the ordering-based ensemble pruning method can both alleviate the dependence of a selected subset  $S$  on the adopted clustering strategies and guarantee the diversity of the selected subset  $S$  [30].

Onan et al. [31] proposed a hybrid ensemble pruning scheme based on clustering and randomized search for text sentiment classification. The purpose of the method is to establish an effective sentiment classification scheme by pursuing the paradigm of ensemble pruning. The classifiers of the ensemble are initially clustered into groups according to their predictive characteristics. Then, two classifiers from each cluster are selected as candidate classifiers based on their pairwise diversity. The search space of candidate classifiers is explored by the elitist Pareto-based multi-objective evolutionary algorithm [31]. Lin et al. [32] presented a hybrid clustering-based pruning method so-called *D3C*. The main strategy of *D3C* is Overproduce and Choose strategy. According to the strategy, a pool of different classifiers constructs an initial ensemble, firstly. Then, using the *k*-means clustering algorithm, some redundant classifiers are eliminated from the ensemble. Using dynamic selection and circulating combination the most accurate models are chosen. The last choosing phase is done fast and without checking all possible subsets of classifiers. In the method, there are some parameters for the base classifiers that can be carefully set in the training process. Lin et al. [32] suggested cross-validation as automatic parameter tuning to obtain the best values for the parameters. The strategy behind cross-validation is to separate the dataset into two parts, one of which is considered to be unknown. The prediction accuracy obtained from the “unknown” set more precisely reflects the performance on classifying an independent dataset. It is worthwhile to mention, the implementation of the approach is done in Java, *LibD3C*. This package is publicly available at the following URL (<http://lab.malab.cn/soft/LibD3C/weka.html>) [32]. Zhang and Cao [33] described a novel clustering-based pruning method so-called *SC*. It applies the clustering algorithm for partitioning the classifiers in two clusters. This is done based on classifiers similarity. Then, using models of one cluster, *SC* constructs pruned ensemble [33].

Zhou and Tang [18] developed a pruning method so-called *GASEN-b*, which was an optimization-based pruning method. *GASEN-b* was also a feasible and practical version of *GASEN* introduced by Zhou et al. [7]. *GASEN-b* was based on a genetic algorithm with standard operators for pruning the ensemble. The fitness function in this method was the accuracy [18]. On the other hand, Caruana et al. [5] presented an optimization-based pruning method so-called Forward Stepwise Selection (*FSS*). In *FSS*, a library of heterogeneous models was pruned using the method *FS* (see Sect. 2). In *FSS*, the evaluation measures are performance-based measures such as accuracy, root-mean-squared-error, mean-cross-entropy, and lift [5]. Partalas et al. [14] described an optimization-based pruning method, which so-called *UAEP*. This method is aware of the uncertainty of ensemble decisions. *UAEP* applies the directed hill climbing algorithm for pruning a heterogeneous initial ensemble.

In this method, two search directions of the directed hill climbing algorithm, forward selection, and backward elimination, are utilized. For *UAEP*, a diversity-based evaluation measure named Uncertainty Weighted Accuracy (*UWA*) is suggested [14]. Taghavi and Sajedi [15] studied a Human-Inspired Ensemble Pruning method (*HIEP*). *HIEP* applies the directed hill climbing algorithm for pruning a heterogeneous initial ensemble. Moreover, two search directions of the directed hill climbing algorithm are used. Moreover, in [15] a diversity-based evaluation measure named Human-Like Foresight (*HLF*) is suggested. This measure is capable to select more accurate models from the initial ensemble [15]. Taghavi and Sajedi [17] introduced a novel ensemble pruning method named *EPCTS* (Ensemble Pruning via Chained Tabu Searches). *EPCTS* applies a chain of tabu searches to choose several models of ensemble progressively until the best subset of them is found. These tabu searches were customized with the proposed strategy named as Periodic Oblivion. This novel strategy revoked interdict of all tabu answers in the defined periods [17]. Sheen et al. [34] developed an optimization-based pruning method. In [34], harmony search is applied to pruning a heterogeneous ensemble. The pruned ensemble is applied to detect malware. Narassiguin et al. [35] introduced a new approach named Dynamic Ensemble Selection (*DES*). In the method, the authors presented the label dependencies to capture the explicitly and proposed a *DES* method based on probabilistic classifier chains. Their source code is available online at URL ([https://github.com/narasil/pcc\\_des](https://github.com/narasil/pcc_des)). Mozaffari et al. [36] introduced a hybrid intelligent system to estimate sea-ice thickness in the Labrador coast of Canada. Their system consists of two main parts. The first part is a heuristic feature selection algorithm used to process a database for selecting the most effective features. On the other hand, the second part is a Hierarchical Selective Ensemble Randomized Neural Network (*HSE-RNN*) that is used to create a nonlinear map between the selected features and sea-ice thickness. The obtained results demonstrated the computational power of the algorithm for the estimation of sea-ice thickness along the Labrador coast. Ye and Dai [37] proposed a novel greedy randomized Dynamic Ensemble Selection (*DES*) algorithm combined with Path Relinking, Variable Neighborhood Search (*VNS*), Greedy Randomized Adaptive Search Procedure (*GRASP*) algorithms, and abbreviated *DyPREVNS-GraspEnS*. Instead of simply selecting the classifiers with the best competence under certain criterions, *DyPREVNS-GraspEnS* makes a greedy randomized selection of appropriate classifiers to form the ensemble. It also realizes random multi-start search that is capable to easily escape from the local optima, and possesses a high probability to find the optimal subensemble. It effectively strengthens the link between iterations and solves the problem of no memory by integrating the path-relinking technique.

Ekbal and Saha [8–11] assumed that instead of pruning the initial ensemble and eliminating some models completely, it is better to apply weighted voting and quantify the number of votes for all the classes per classifier. In [8], the authors applied Archived Multi-objective Simulated Annealing (AMOSA) that can be studied in [12, 38] to choose the best weights. This method applied for Named Entity Recognition (NER), utilized several objective functions, e.g. overall recall and precision,  $F$ -measure values, and so on. In [10], single and multi-objective-based methods are suggested for Part-of-Speech (PoS) tagging. In the single objective, SA [21], and in the multi-objective, AMOSA [38] are used as the optimization algorithm. In another approach [9], authors using Genetic Algorithm (GA) [39] attempted to determine the appropriate weights of voting for each class in each classifier. This approach was also proposed for NER. The work of [11] proposed ensemble methods using single and multi-objective optimization algorithms [12]. This work proposed for NER, two different versions are suggested: (1) Binary vote-based method proposed for determining whether particular classifier is allowed to vote for a particular class or not. (2) The real vote-based method proposed for quantifying the number of votes for all classes in each classifier. Its single objective-based applies GA [39], and its multi-objective-based applies the Non-dominated Sorting Genetic Algorithm (NSGA-II) [40]. Wang et al. [41] studied essential relationships between generalization capabilities and fuzziness of fuzzy classifiers in ensemble learning. They showed that higher fuzziness of a fuzzy classifier can promote generalization aspects of the classifier, especially for classification data exhibiting complex boundaries. They also provided some guidelines to improve the generalization aspects of fuzzy classifiers.

## 4 Proposed approach

In this section, firstly, the rationale behind the proposed approach is presented and then the procedure and the main technical contributions are introduced. Then, the proposed algorithm for this approach is developed.

### 4.1 Rationales and procedure

Our proposed ensemble pruning approach is named Simulated Quenching WALKING (SQWALKING). The origin of SQWALKING is Upward Stochastic Walking idea. For realizing this new idea, FS and SQ algorithms are incorporated, as represented in Sect. 2. The reasons to choose FS and SQ are as follows: (1) FS is one of the famous architectures in ensemble pruning scope. (2) Some effective ensemble pruning methods such as [5, 6, 14–17] are based on FS architecture. (3) Although FS has shown that it may yield desired results, it is possible that it gets

stuck in local optima. The reason for this deficiency is the greedy nature of FS. (4) SQ is an effective and fast optimization algorithm. (5) SQ has a stochastic and probabilistic nature. Thus, in the proposed SQWALKING, by incorporating SQ and FS, our new idea, upward stochastic walking, is replaced with the forward greedy stepwise idea of FS. Thereupon, SQWALKING can escape from local optima and also can reach global optimum almost certainly. It is worthwhile to mention that the purpose of the global optimum is to achieve a pruned ensemble with the highest prediction performance and lowest computational overheads.

The proposed procedure of SQWALKING is shown in Fig. 3, graphically. Firstly,  $N$  base classifiers are selected for constructing homogeneous or heterogeneous initial ensemble. These constructed models are transferred to a set named remainder model set. This set always contains unselected models. As shown in Fig. 3, SQWALKING starts from an empty current ensemble. It then continues to reach a full current ensemble of  $N$  models. Thus,  $N$  current ensembles are produced during  $N$  steps of SQWALKING. In each step, using SQ, one model (SM) is selected through the remainder model set. Then, it is added to the previous current ensemble to produce the next new one. At the end of this procedure, the current ensemble along with the highest performance and the smallest size is suggested as a pruned ensemble by SQWALKING. For a better explanation, consider the step  $j + 1$ . In this step, the current ensemble  $j$  has been constructed. The remainder model set also contains  $N - j$  models. SQ finds one model through a remainder model set via its procedure. This model is added to current ensemble  $j$  to produce the current ensemble  $j + 1$ . Afterward, the pruned ensemble is transferred for combining phase. To combine the models of the pruned ensemble, effective combination method can be applied. In the present work, it is preferred to apply the soft voting technique [1, 27].

Considering the mentioned issues, it can be inferred that SQWALKING can lead to increase prediction performance and to decrease computational overheads of ensemble method. First evidence for this inference is that SQWALKING always replaces the current ensemble with the highest performance along with a previous best-obtained current ensemble. Therefore, it always obtains the pruned ensemble with the highest performance. The second evidence is that as the direction of SQWALKING is based on the forward selection method, the pruned ensemble is corresponding to the smallest size current ensemble. Therefore, SQWALKING always produces the smallest size pruned ensemble, which has the highest performance.

Considering the mentioned consequence, the main technical contributions of SQWALKING is specified as:

- The structure of the SQWALKING can be suggested to solve many classification problems. It can yield pruned

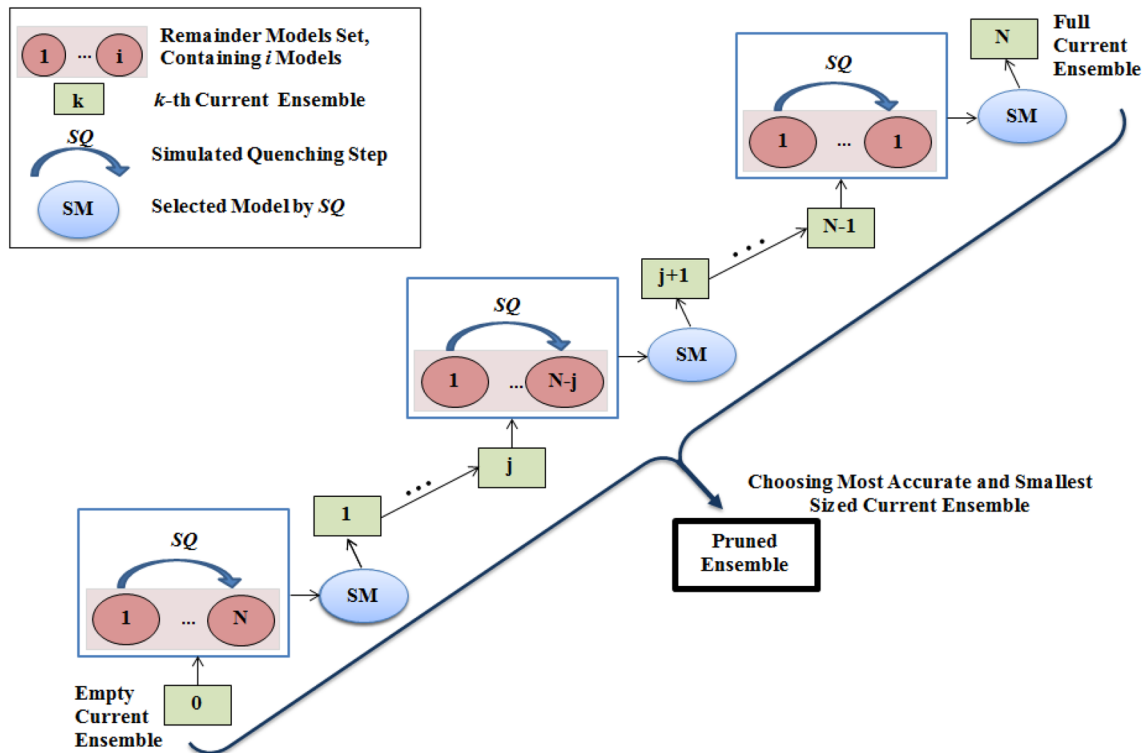


Fig. 3 The procedure of SQWALKING

ensembles with the desired efficiency. The SQWALKING is a parameter-based approach where the parameters must be set meticulously. As such, its results on each classification problem can be dramatically improved when the parameters are calibrated. This certainly takes more time.

- SQWALKING is independent of ensemble construction and combination phase. Therefore, many proper ensemble constructions and combination techniques can be a candidate.
- In heterogeneous initial ensemble mode, SQWALKING can be applied as a tool for recognizing the classification algorithm capability. Consequently, selected classifiers in the pruned ensemble identify which type of classifiers is more effective for a specific problem.
- The SQWALKING can be applied as an effective pruning algorithm for increasing the efficiency of the ensemble method.
- The SQWALKING is run one time and in an offline manner in order to indicate the most efficient classifiers on a specific classifier problem. Then, the achieved pruned ensemble is applied over and over.

### A. The Proposed Algorithm

The pseudo-code of SQWALKING is shown in Fig. 4. As illustrated in Fig. 4, at the first step,  $N$  models of the initial

ensemble are added into remainder model set (say  $Set_M$ ). Then, SQWALKING until  $Set_M$  is becoming empty proceeds to select the best model (say  $M_{BEST}$ ) through  $Set_M$  and adds it into current ensemble (say  $Ens_C$ ). Each  $Ens_C$  must be evaluated in each step. For this purpose, the overall accuracy of the  $Ens_C$  is considered on a separate pruning dataset. If the accuracy of  $Ens_C$  (say  $ACC_C$ ) is greater than the accuracy of the best obtained current ensemble (say  $ACC_{MAX}$ ), it is assigned as the best current ensemble (say  $Ens_{BEST}$ ). In the end, the last  $Ens_{BEST}$  is suggested by SQWALKING as a pruned ensemble.

In each step, for finding  $M_{BEST}$ , SQWALKING starts with one randomly selected model. The model must be evaluated with the acceptance function. In this problem, a special evaluation measure plays this function's role. Two different measures are used in SQWALKING. One is the accuracy ( $ACC$ ) that is a performance-based evaluation measure. The second is the Human-Like Foresight ( $HLF$ ) [15] that is a diversity-based measure. According to this case, it can be said that two different versions of SQWALKING can be developed. One version named  $SQWALKING_A$  is based on  $ACC$ . Another approach is  $SQWALKING_H$  is based on  $HLF$  [15]. Therefore,  $M_{BEST}$  is evaluated with the measure  $ACC$  or  $HLF$  (say  $E_{MAX}$ ).

For each  $\theta$  during the procedure, as an equilibrium condition for  $\theta$  is met, a random model (say  $M_i$ ) is generated. The

---

**Function**  $SQWALKING_H$  (Initial Ensemble) **Returns** Pruned Ensemble

**Input:**  
 $Set_M, Ens_C, \Theta_0, \Theta_F, \alpha, Itr_0, \beta, ACC_C, ACC_{MAX}$

**Output:**  
 $Ens_{BEST}$

**Process:**

1. **Initialize**  $Set_M \leftarrow$  Initial Ensemble,  $Ens_C \leftarrow \emptyset, \Theta_0, \Theta_F, \alpha, Itr_0, \beta,$   
 $ACC_C \leftarrow 0, ACC_{MAX} \leftarrow 0$
2. **while** ( $Set_M \neq \emptyset$ ) **do**
  - A.  $M_{BEST} \leftarrow$  GetRandomModel ( $Set_M$ )
  - B.  $E_{MAX} \leftarrow$  Compute HLF\* ( $Ens_C, M_{BEST}$ )
  - C.  $\Theta_C \leftarrow \Theta_0$
  - D.  $Itr_C \leftarrow Itr_0$
  - E. **while** ( $\Theta_C > \Theta_F$ ) **do**
    - a.  $i \leftarrow 1$
    - b. **while** ( $i < Itr_C$ ) **do**
      - i.  $M_i \leftarrow$  GetRandomModel ( $Set_M$ )
      - ii.  $E_i \leftarrow$  Compute HLF\* ( $Ens_C, M_i$ )
      - iii. **if** ( $E_i > E_{MAX}$ ) **then**
        - $M_{BEST} \leftarrow M_i$
        - $E_{MAX} \leftarrow E_i$
      - iv. **else if** ( $RandomReal(0,1) < \exp((E_i - E_{MAX}) / \Theta_C)$ ) **then**
        - $M_{BEST} \leftarrow M_i$
        - $E_{MAX} \leftarrow E_i$
      - v.  $i \leftarrow i+1$
    - c.  $\Theta_C \leftarrow \alpha \times \Theta_C$
    - d.  $Itr_C \leftarrow (1/\beta) \times Itr_C$
  - F. Add  $M_{BEST}$  to  $Ens_C$
  - G. Drop  $M_{BEST}$  from  $Set_M$
  - H.  $ACC_C \leftarrow$  EstimateOverallAccuracy( $Ens_C$ )
  - I. **if** ( $ACC_C > ACC_{MAX}$ ) **then**
    - a.  $Ens_{BEST} \leftarrow Ens_C$
    - b.  $ACC_{MAX} \leftarrow ACC_C$

- 3. **Return**  $Ens_{BEST}$

---

\* **Note:** HLF is the evaluation measure of version H ( $SQWALKING_H$ )

---

**Fig. 4** The pseudo-code of  $SQWALKING$  (version H)

parameter  $\Theta$  is initialized with the value  $\Theta_0$ . Then, with a geometric cooling function shown in Eq. (1), it is reduced. The cooling process is continued until the parameter  $\Theta$  become less than a specific value  $\Theta_F$  as a stopping criterion.

$$\Theta_i = \alpha \times \Theta_{i-1} \quad 0 < \alpha < 1. \quad (1)$$

For the equilibrium condition, a geometric function given in Eq. (2) is used to determine the number of iterations per  $\Theta$ . It is initialized with the value  $Itr_0$ . Applying this function leads to increase the number of iterations gradually. Therefore, when the parameter  $\Theta$  is reduced, the number of iterations will be increased. Hence, in the final steps of the algorithm, more searches can be implemented.

$$Itr_i = (1/\beta) \times Itr_{i-1} \quad 0 < \beta < 1. \quad (2)$$

Moreover, each  $M_i$  must be evaluated with the measure ACC or HLF (say  $E_i$ ). Afterward, the value of  $\Delta E$ , ( $E_i - E_{MAX}$ ), is calculated. If  $\Delta E$  becomes greater than zero,  $M_i$  and  $E_i$  are assigned as  $M_{BEST}$  and  $E_{MAX}$ , respectively; otherwise, with probability  $\exp(\Delta E / \Theta)$ , the job is done. After reaching the temperature  $\Theta_F$ ,  $M_{BEST}$  is considered as the selected model by  $SQWALKING$ , in one loop.

$SQWALKING$ s have some parameters, which must be adjusted meticulously. The parameters are  $N, \Theta_0, \Theta_F, \alpha, Itr_0$ , and  $\beta$ .

## 5 Experimental setup

In this section, some experiments are presented to evaluate the proposed approach. For a comprehensive evaluation of  $SQWALKING$ , we compare this approach with many existing approaches, experimentally. Such comprehensive studies are essential from different viewpoints. According to the viewpoints, the comparisons are categorized into 3 groups as follows. *Group A:* Evaluating  $SQWALKING$  by comparing to other ensemble methods with the pruning phase. The objective is to evaluate the procedure of ensemble pruning phases. In this group, both heterogeneous and homogeneous initial ensembles are utilized. *Group B:* Evaluating  $SQWALKING$  by comparing to other popular ensemble methods without the pruning phase. On the other hand, *Group C* evaluates  $SQWALKING$  by comparing to combination methods without the pruning phase. It should be noted that the *Groups B* and *C* help specifying whether the existence of the pruning phase is effective, or the systems without this phase can also work well.

### 5.1 Group A: Evaluating $SQWALKING$ by comparing to other ensemble methods with the pruning phase

In this group, the objective is to compare the procedure of ensemble pruning phases. In order to do so, the same conditions of ensemble construction and combination phases should be provided entirely. Thus, their pruning phases are compared. For this purpose, the performance of the proposed  $SQWALKING$  (both versions) is compared with the ones of four ensemble pruning methods, *FSS* [5], *UAEP* [14], *HIEP* [15], and *GASEN-b* [18].

Ten different machine learning problems retrieved from the *UCI* machine learning repository are used, which are available from the URL (<http://archive.ics.uci.edu/ml/datasets.html>). In Table 1, the information of datasets is shown. Moreover, for more compressive evaluation, two different ensemble construction techniques are applied as follows:

- Heterogeneous initial ensemble; a pool of 100 different classifiers with various parameters is used for constructing initial ensemble. The details are shown in Table 2.
- Homogeneous initial ensemble; 100 Decision Trees (*DT*) are used. *DT*s were based on *C4.5* algorithm with default configuration (post pruning trees with a confidence factor equal to 0.2). We employed bootstrap sampling to generate several training datasets, as in Bagging [22].



**Table 1** Details of datasets

| Dataset ID       | UCI folder name                                 | Number of instances | Number of attributes | Number of classes | Area     |
|------------------|---|---------------------|----------------------|-------------------|----------|
| DS <sub>1</sub>  | Audiology (standardized)                        | 226                 | 69                   | 24                | Life     |
| DS <sub>2</sub>  | Balance-scale                                   | 625                 | 4                    | 3                 | Social   |
| DS <sub>3</sub>  | Connectionist bench (sonar, mines versus rocks) | 208                 | 60                   | 2                 | Physical |
| DS <sub>4</sub>  | Contraceptive method choice (CMC)               | 1473                | 9                    | 3                 | Life     |
| DS <sub>5</sub>  | Ecoli   | 336                 | 7                    | 8                 | Life     |
| DS <sub>6</sub>  | Glass identification                            | 214                 | 9                    | 7                 | Physical |
| DS <sub>7</sub>  | Hepatitis                                       | 155                 | 19                   | 2                 | Life     |
| DS <sub>8</sub>  | Iris  | 150                 | 4                    | 3                 | Life     |
| DS <sub>9</sub>  | Primary tumor                                   | 339                 | 17                   | 22                | Life     |
| DS <sub>10</sub> | Statlog (heart)                                 | 270                 | 13                   | 2                 | Life     |

**Table 2** Details Of Base Classifiers For Heterogeneous Initial Ensemble

| Classifier type              | Value of parameters  | No. |
|------------------------------|--|-----|
| Decision tree (DT)           | 10 Post pruning decision tree with<br>Confidence factor: 0.1, 0.2, 0.3, 0.4, 0.5<br>Laplace smoothing: true, false<br>16 Reduced error pruning decision tree<br>Number of fold: 2, 3, 4, 5, 6, 7, 8, 9<br>Laplace smoothing: true, false<br>2 Un-pruned decision tree<br>Minimum object per leaf: 2, 3 | 28  |
| K-nearest neighbor (KNN)     | K = 1 ~ plurality of the training instances (5 different values)<br>Weighting method: no-weighting, similarity weighting   | 10  |
| Multi-layer perceptron (MLP) | Hidden layer: 1, 2, 4, 8, 16<br>Momentum term: 0.2, 0.5, 0.9<br>Learning rate: 0.3, 0.6  | 30  |
| Naive bayes (NB)             | Kernel estimator: true, false  | 2   |
| Support vector machine (SVM) | Complexity parameter: 0.00001, 0.001, 0.1<br>Kernels: polynomial kernel with degree: 2, 3, and, radial kernel<br>with gamma: 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2  | 30  |

Each ensemble pruning method requires some adjustments. In *FSS*, the accuracy is selected as the performance metric. In *UAEP* and *HIEP*, the forward selection method is utilized as the search direction of the directed hill climbing algorithm. In *SQWALKINGs* the parameter  $N$  was set to 100, the parameter  $\theta_0$  was set to 1000, the parameter  $\alpha$  was set to 0.9, the parameter  $\theta_F$  was set to 0.01, the parameter  $Itr_0$  was set to 10, and the parameter  $\beta$  was set to 0.95. In *GASEN-b*, the maximum iteration was set to 150, the population size was set to 30, the crossover probability was set to 0.6, and the mutation probability was set to 0.1. It is noteworthy to say that for *GASEN-b*, the parameters with different values are examined. Nevertheless, improvement in results is not observed. So, we applied these values which led to good answers and a shorter running time.

Regarding ensemble combination phase, the soft voting combination method addressed in [1, 27] is used. Moreover,

regarding the producing required instances 60%, 20%, and 20% of distinct instances of each dataset (in Table 1) are selected for training, pruning, and testing phases, respectively. About methodology, the experiments are carried out 20 times on each dataset. At first, four different situations of dataset instances are produced for training, pruning, and testing datasets. Then each situation is repeated five times for each ensemble pruning method. In the end, the average over the best-gained answers of each situation per ensemble pruning method (on each dataset) is calculated.

## 5.2 Group B: Evaluating *SQWALKING* by comparing to other popular ensemble methods without the pruning phase

In this group, our goal is to compare *SQWALKING* with popular ensemble methods, which do not have the pruning

phase. Thus, the effectiveness and importance of the pruning phase can be evaluated. Bagging [22], AdaBoost (AdaBoost M1) [42], and Random Forest [26] are selected as popular ensemble methods for carrying out the comparisons. The default values of all parameters of these ensemble methods are applied. Moreover, for these methods, 66% and 34% of distinct instances of each dataset (in Table 1) are selected as the training and the testing datasets, respectively. For *SQWALKING*s, all related settings in *Group A* including datasets, constructing homogeneous ensemble, adjusting pruning phase parameters, combining the models, and the methodology are used.

### 5.3 Group C: Evaluating SQWALKING by comparing to combination methods without the pruning phase

In this group, the objective is to compare *SQWALKING*s with popular combination methods. The combination methods attempt to improve the effectiveness of the ensemble by focusing on how the predictions of constructing models should be combined. Thus, the effectiveness and importance of the pruning phase compared to the methods which do not have the pruning phase can be evaluated. Soft voting and majority voting [1, 27] are selected as popular combination methods for carrying out the comparisons. We again use the same setup in *Group B*, for *SQWALKING*s with default values of all parameters of soft voting and majority voting methods. For soft voting and majority voting, we selected 66% and 34% of distinct instances of each dataset (in Table 1), as training and testing datasets, respectively.

## 6 Evaluation results

Here, the experimental results for *Groups A, B, and C*, are presented separately. The results are considered according to the accuracy, size, and the average running time of the ensembles.

### 6.1 Results of Group A

Table 3 shows the accuracy of the six ensemble pruning methods on each dataset (for heterogeneous initial ensemble). For a better clarification, the bold font style is applied for explicit display of the most accurate values across all datasets. Along with the accuracy, the corresponding rank of the methods is listed in Table 3. It helps readers to compare the accuracy of these methods easier [43]. The methods are ranked with the following ranking approach; the most accurate methods are ranked by grade 1.0; the second most accurate methods are ranked by grade 2.0, and so on.

**Table 3** Group A (Heterogeneous Initial Ensemble): Accuracy And Corresponding Rank Of Six Ensemble Pruning Methods, On Each Dataset

| Dataset ID       | Accuracy (%) |              |        |                        |                        |         | Rank     |          |      |                        |                        |         |
|------------------|--------------|--------------|--------|------------------------|------------------------|---------|----------|----------|------|------------------------|------------------------|---------|
|                  | FSS          | HIEP         | UAEP   | SQWALKING <sub>H</sub> | SQWALKING <sub>A</sub> | GASEN-b | FSS      | HIEP     | UAEP | SQWALKING <sub>H</sub> | SQWALKING <sub>A</sub> | GASEN-b |
| DS <sub>1</sub>  | 77.22        | 73.33        | 76.67  | <b>77.78</b>           | 77.22                  | 73.33   | 2        | 4        | 3    | <b>1</b>               | 2                      | 4       |
| DS <sub>2</sub>  | 88.60        | 91.60        | 87.40  | <b>91.80</b>           | 88.80                  | 67.80   | 4        | 2        | 5    | <b>1</b>               | 3                      | 6       |
| DS <sub>3</sub>  | 64.64        | 59.76        | 63.42  | 60.98                  | <b>65.25</b>           | 53.05   | 2        | 5        | 3    | 4                      | <b>1</b>               | 6       |
| DS <sub>4</sub>  | 52.13        | 51.36        | 51.70  | 52.13                  | <b>52.55</b>           | 51.28   | 2        | 4        | 3    | 2                      | <b>1</b>               | 5       |
| DS <sub>5</sub>  | 64.55        | 67.54        | 64.93  | <b>67.91</b>           | 64.93                  | 44.78   | 4        | 2        | 3    | <b>1</b>               | 3                      | 5       |
| DS <sub>6</sub>  | 47.02        | 47.02        | 48.22  | <b>54.17</b>           | 52.38                  | 38.10   | 4        | 4        | 3    | <b>1</b>               | 2                      | 5       |
| DS <sub>7</sub>  | 81.45        | 82.26        | 83.07  | <b>84.68</b>           | 79.84                  | 83.07   | 4        | 3        | 2    | <b>1</b>               | 5                      | 2       |
| DS <sub>8</sub>  | <b>95.00</b> | <b>95.00</b> | 93.34  | <b>95.00</b>           | <b>95.00</b>           | 43.33   | <b>1</b> | <b>1</b> | 2    | <b>1</b>               | <b>1</b>               | 3       |
| DS <sub>9</sub>  | 49.25        | 48.51        | 47.02  | 47.76                  | <b>50.37</b>           | 48.51   | 2        | 3        | 5    | <b>1</b>               | <b>1</b>               | 3       |
| DS <sub>10</sub> | <b>82.41</b> | 80.09        | 78.24  | 81.94                  | 78.70                  | 55.56   | <b>1</b> | 3        | 5    | 2                      | 4                      | 6       |
| Ave              | 70.227       | 69.647       | 69.401 | <b>71.415</b>          | 70.504                 | 55.88   | 2.6      | 3.1      | 3.4  | <b>1.8</b>             | 2.3                    | 4.5     |

Bold values indicate the highest accuracy and the highest rank achieved by the algorithms in 10 datasets

Table 3 shows that  $SQWALKING_H$  achieves the highest accuracy in 5 datasets, exclusively. Meanwhile, the  $SQWALKING_A$  achieves the highest accuracy in 3 datasets, solely.  $FSS$ ,  $HIEP$ ,  $SQWALKING_H$ , and  $SQWALKING_A$  obtain the most accurate pruned ensemble in 1 dataset, similarly.  $FSS$  has just obtained the highest accuracy in 1 dataset, exclusively. Moreover, for general evaluating of the accuracy of the methods, the average corresponding values to each method over all datasets are calculated. These values are shown in the last row of Table 3. These results declare that  $SQWALKING_H$  and  $SQWALKING_A$  attain the first and the second positions compared to

others, with average accuracy 71.415% and 70.504%, and with corresponding average ranks of 1.8 and 2.3, respectively.  $FSS$  attains the third position with average accuracy 70.227% and the corresponding average rank 2.6.  $HIEP$  and  $UAEP$  are in the fourth and the fifth positions because of their average accuracy, 69.647 and 69.401; and their corresponding average ranks of 3.1 and 3.4, respectively. In the end,  $GASEN-b$  is in the last position with the average accuracy of 55.88% and the corresponding average rank of 4.5. In addition, the  $SQWALKING_H$  with the running time of 21 min and 47 s is the slowest algorithm among all.

**Table 4** Group A (heterogeneous initial ensemble): size of pruned ensembles and the number of constitutive models based on the types of classifiers, for the six ensemble pruning methods on each dataset

| Dataset ID       | Size of pruned ensemble |           |           |          |          |            |             |           |           |          |          |          |             |           |           |          |          |           |
|------------------|-------------------------|-----------|-----------|----------|----------|------------|-------------|-----------|-----------|----------|----------|----------|-------------|-----------|-----------|----------|----------|-----------|
|                  | <i>FSS</i>              |           |           |          |          |            | <i>HIEP</i> |           |           |          |          |          | <i>UAEP</i> |           |           |          |          |           |
|                  | $N_{MLP}$               | $N_{KNN}$ | $N_{SVM}$ | $N_{DT}$ | $N_{NB}$ | All        | $N_{MLP}$   | $N_{KNN}$ | $N_{SVM}$ | $N_{DT}$ | $N_{NB}$ | All      | $N_{MLP}$   | $N_{KNN}$ | $N_{SVM}$ | $N_{DT}$ | $N_{NB}$ | All       |
| DS <sub>1</sub>  | 5                       | 0         | 1         | 5        | 0        | <b>11</b>  | 6           | 1         | 0         | 5        | 0        | 12       | 5           | 1         | 0         | 6        | 0        | 12        |
| DS <sub>2</sub>  | 4                       | 0         | 0         | 0        | 0        | <b>4</b>   | 17          | 2         | 5         | 5        | 1        | 30       | 4           | 1         | 0         | 0        | 0        | 5         |
| DS <sub>3</sub>  | 2                       | 0         | 0         | 0        | 0        | <b>2</b>   | 5           | 1         | 3         | 4        | 1        | 14       | 4           | 0         | 0         | 0        | 0        | 4         |
| DS <sub>4</sub>  | 14                      | 3         | 7         | 17       | 1        | 42         | 14          | 4         | 7         | 18       | 1        | 44       | 10          | 3         | 1         | 12       | 1        | <b>27</b> |
| DS <sub>5</sub>  | 1                       | 0         | 0         | 0        | 0        | <b>1</b>   | 4           | 0         | 2         | 0        | 0        | 6        | 1           | 0         | 0         | 0        | 0        | <b>1</b>  |
| DS <sub>6</sub>  | 1                       | 0         | 0         | 0        | 0        | <b>1</b>   | 1           | 0         | 0         | 0        | 0        | <b>1</b> | 11          | 3         | 4         | 1        | 0        | 19        |
| DS <sub>7</sub>  | 1                       | 0         | 0         | 0        | 1        | <b>2</b>   | 0           | 0         | 1         | 0        | 0        | <b>1</b> | 2           | 1         | 1         | 0        | 0        | 4         |
| DS <sub>8</sub>  | 1                       | 0         | 0         | 0        | 0        | <b>1</b>   | 2           | 1         | 0         | 0        | 0        | 3        | 2           | 0         | 0         | 0        | 0        | 2         |
| DS <sub>9</sub>  | 3                       | 1         | 1         | 6        | 2        | 13         | 7           | 2         | 2         | 12       | 0        | 23       | 2           | 1         | 1         | 3        | 2        | <b>9</b>  |
| DS <sub>10</sub> | 1                       | 0         | 0         | 0        | 0        | <b>1</b>   | 1           | 0         | 0         | 0        | 0        | <b>1</b> | 2           | 0         | 0         | 0        | 0        | 2         |
| Avg              | 3.3                     | 0.4       | 0.9       | 2.8      | 0.4      | <b>7.8</b> | 5.7         | 1.1       | 2.0       | 4.4      | 0.3      | 13.5     | 4.3         | 1.0       | 0.7       | 2.2      | 0.3      | 8.5       |

Bold values indicate the smallest size pruned ensembles achieved by the algorithms on 10 datasets

$N_{MLP}$  Number of *MLPs*,  $N_{KNN}$  number of *KNNs*,  $N_{SVM}$  number of *SVMs*,  $N_{DT}$  number of *DTs*,  $N_{NB}$  number of *NBs*

**Table 5** Continued From Table 4

| Dataset ID       | Size of pruned ensemble |           |           |          |          |          |               |           |           |          |          |          |           |           |           |          |          |      |
|------------------|-------------------------|-----------|-----------|----------|----------|----------|---------------|-----------|-----------|----------|----------|----------|-----------|-----------|-----------|----------|----------|------|
|                  | $SQWALKING_H$           |           |           |          |          |          | $SQWALKING_A$ |           |           |          |          |          | $GASEN-b$ |           |           |          |          |      |
|                  | $N_{MLP}$               | $N_{KNN}$ | $N_{SVM}$ | $N_{DT}$ | $N_{NB}$ | All      | $N_{MLP}$     | $N_{KNN}$ | $N_{SVM}$ | $N_{DT}$ | $N_{NB}$ | All      | $N_{MLP}$ | $N_{KNN}$ | $N_{SVM}$ | $N_{DT}$ | $N_{NB}$ | All  |
| DS <sub>1</sub>  | 3                       | 1         | 0         | 8        | 0        | 12       | 4             | 0         | 1         | 7        | 0        | 12       | 10        | 5         | 16        | 15       | 1        | 47   |
| DS <sub>2</sub>  | 4                       | 0         | 1         | 1        | 0        | 6        | 4             | 0         | 0         | 0        | 0        | <b>4</b> | 17        | 6         | 15        | 9        | 1        | 48   |
| DS <sub>3</sub>  | 2                       | 1         | 1         | 1        | 0        | 5        | 2             | 0         | 0         | 0        | 0        | <b>2</b> | 18        | 7         | 13        | 15       | 1        | 54   |
| DS <sub>4</sub>  | 17                      | 5         | 11        | 22       | 1        | 56       | 10            | 3         | 6         | 16       | 1        | 36       | 11        | 4         | 12        | 12       | 1        | 40   |
| DS <sub>5</sub>  | 5                       | 0         | 2         | 1        | 0        | 8        | 1             | 0         | 0         | 0        | 0        | <b>1</b> | 16        | 2         | 11        | 8        | 1        | 38   |
| DS <sub>6</sub>  | 1                       | 0         | 2         | 0        | 0        | 3        | 5             | 2         | 8         | 2        | 0        | 17       | 11        | 3         | 12        | 12       | 1        | 39   |
| DS <sub>7</sub>  | 0                       | 0         | 1         | 0        | 0        | <b>1</b> | 7             | 2         | 7         | 6        | 1        | 23       | 17        | 5         | 11        | 14       | 1        | 48   |
| DS <sub>8</sub>  | 2                       | 0         | 1         | 0        | 0        | 3        | 1             | 0         | 0         | 0        | 0        | 1        | 14        | 7         | 14        | 15       | 1        | 51   |
| DS <sub>9</sub>  | 3                       | 2         | 1         | 10       | 2        | 18       | 5             | 1         | 2         | 8        | 1        | 17       | 14        | 6         | 21        | 11       | 1        | 53   |
| DS <sub>10</sub> | 1                       | 0         | 0         | 0        | 0        | <b>1</b> | 1             | 0         | 1         | 0        | 0        | 2        | 16        | 4         | 11        | 18       | 1        | 50   |
| Avg              | 3.8                     | 0.9       | 2.0       | 4.3      | 0.3      | 11.3     | 4.0           | 0.8       | 2.5       | 3.9      | 0.3      | 11.5     | 14.4      | 4.9       | 13.6      | 12.9     | 1        | 46.8 |

Bold values indicate the smallest size pruned ensembles achieved by the algorithms on 10 datasets

Tables 4 and 5 show the size of the pruned ensemble for the six ensemble pruning methods, on each dataset (for heterogeneous initial ensemble). Note that, all values are rounded to gain integer values. These results show that *FSS* leads to the smallest size pruned ensemble on 7 datasets; *SQWALKING<sub>A</sub>* on 4 datasets, *HIEP* and *UAEP* on 3 datasets, and *SQWALKING<sub>H</sub>* on 2 datasets obtain the smallest size pruned ensemble. Note that these results may overlap on some datasets. At last, *GASEN-b* leads to pruned ensembles with the highest numbers of classifiers over all datasets. Furthermore, for general evaluating, we calculate the average values over all datasets. These values are shown on a separate row in Tables 4 and 5. These results show that *FSS* leads to the smallest size pruned ensembles with the average size of 7.8. *UAEP* with average size 8.5 leads to the second smallest size pruned ensembles. *SQWALKING<sub>H</sub>* and *SQWALKING<sub>A</sub>* are in the third and the fourth positions, with the average size of 11.3 and 11.5, respectively. Finally, *HIEP* and *GASEN-b* with average size of 13.5 and 46.8 lead to the largest size pruned ensembles.

In addition to the size of the pruned ensemble in Tables 4 and 5, the number of models according to the types of base classifiers is shown.  $N_{MLP}$  is the number of *MLPs*.  $N_{KNN}$  is the number of *KNNs*.  $N_{SVM}$  is the number of *SVMs*.  $N_{DT}$  is the number of *DTs*.  $N_{NB}$  is the number of *NBs*. This study helps readers comprehend which base classifiers are more appropriate for an especial machine learning problem [17]. For a simple example, in dataset *Statlog (Heart)*, *DS<sub>10</sub>*, all ensemble pruning algorithms select *MLP* as the appropriate base classifier for this problem; however, *SQWALKING<sub>A</sub>* selects *MLP* and *SVM*. For further reading, please see [17]. Considering the average related to the base classifiers' types, in the last row of Tables 4 and 5; the percentage of participation of the base classifiers is calculated in obtained pruned

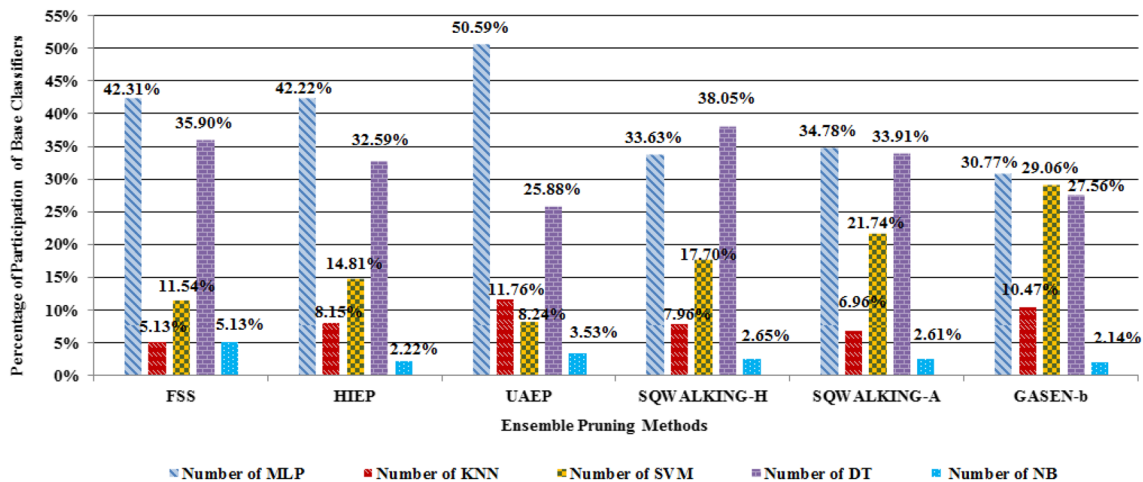
ensembles. These values are illustrated in Fig. 5. This figure shows that *MLPs* have the largest proportion of participation in all pruned ensembles, except in one extracted from *SQWALKING<sub>H</sub>*. Only in *SQWALKING<sub>H</sub>*, *DTs* with 38.05% versus *MLPs* with 33.63% have a higher percentage of participation. In all of the pruning methods, *NBs* have the smallest proportion of participation in the pruned ensembles.

Table 6 shows the average running time of the ensemble pruning methods over all datasets (for heterogeneous initial ensemble). It shows that *FSS* with running time of 0.7636s is the fastest algorithm.

Table 7 shows the accuracy along with the corresponding rank of six ensemble pruning methods on each dataset (for the homogeneous initial ensemble). *SQWALKING<sub>A</sub>* and *HIEP* in 5 datasets are the most accurate methods, while *SQWALKING<sub>H</sub>* is the most accurate method in 8 datasets. *GASEN-b* is the most accurate method in 4 datasets, and both *UAEP* and *FSS* just in 3 datasets are the most accurate methods. These results may overlap in some datasets. According to the average of the accuracy in the last row of Table 7, *SQWALKING<sub>H</sub>*, *HIEP*, and *SQWALKING<sub>A</sub>* attain the first, the second, and the third positions compared to others, with the average accuracy of 77.279%, 76.706%, and 76.353%, respectively. *GASEN-b* with 75.396%, *UAEP* with 74.478%, and *FSS* with 73.053% attain the fourth, the fifth and the sixth positions, respectively. However, according

**Table 6** Group A (heterogeneous initial ensemble): average running time of the six ensemble pruning methods obtained over all datasets

| Running time |             |             |                              |                              |                |
|--------------|-------------|-------------|------------------------------|------------------------------|----------------|
| <i>FSS</i>   | <i>HIEP</i> | <i>UAEP</i> | <i>SQWALKING<sub>H</sub></i> | <i>SQWALKING<sub>A</sub></i> | <i>GASEN-b</i> |
| 0.7636 s     | 1.4512 s    | 1.213 s     | 21 min, 47 s                 | 6 min, 35 s                  | 1 min, 21 s    |



**Fig. 5** Group A (heterogeneous initial ensemble): percentage of participation of base classifiers in pruned ensembles, For the six ensemble pruning methods, average obtained over all datasets

**Table 7** Group A (homogeneous initial ensemble): accuracy and corresponding rank of the six ensemble pruning methods, on each dataset

| Dataset ID       | Accuracy (%) |              |              |                              |                              |                | Rank     |          |          |                              |                              |                |
|------------------|--------------|--------------|--------------|------------------------------|------------------------------|----------------|----------|----------|----------|------------------------------|------------------------------|----------------|
|                  | FSS          | HIEP         | UAEP         | <i>SQWALKING<sub>H</sub></i> | <i>SQWALKING<sub>A</sub></i> | <i>GASEN-b</i> | FSS      | HIEP     | UAEP     | <i>SQWALKING<sub>H</sub></i> | <i>SQWALKING<sub>A</sub></i> | <i>GASEN-b</i> |
| DS <sub>1</sub>  | 73.33        | 73.33        | 75.67        | <b>76.80</b>                 | 73.33                        | <b>76.80</b>   | 3        | 3        | 2        | <b>1</b>                     | 3                            | <b>1</b>       |
| DS <sub>2</sub>  | 88.80        | 89.60        | 86.20        | <b>90.20</b>                 | 88.80                        | 88.80          | 3        | 2        | 4        | <b>1</b>                     | 3                            | 3              |
| DS <sub>3</sub>  | 75.20        | 79.50        | 75.20        | 79.50                        | <b>83.90</b>                 | 79.50          | 3        | 2        | 3        | 2                            | <b>1</b>                     | 2              |
| DS <sub>4</sub>  | 50.40        | 50.40        | 51.50        | <b>53.47</b>                 | 52.00                        | 51.50          | 4        | 4        | 3        | <b>1</b>                     | 2                            | 3              |
| DS <sub>5</sub>  | 64.55        | <b>84.91</b> | 81.93        | <b>84.91</b>                 | <b>84.91</b>                 | 81.93          | 3        | <b>1</b> | 2        | <b>1</b>                     | <b>1</b>                     | 2              |
| DS <sub>6</sub>  | <b>76.40</b> | <b>76.40</b> | <b>76.40</b> | <b>76.40</b>                 | 73.97                        | 69.15          | <b>1</b> | <b>1</b> | <b>1</b> | <b>1</b>                     | 2                            | 3              |
| DS <sub>7</sub>  | 82.50        | 79.14        | 78.30        | <b>89.50</b>                 | 82.84                        | 82.50          | 3        | 4        | 5        | <b>1</b>                     | 2                            | 3              |
| DS <sub>8</sub>  | <b>96.07</b> | <b>96.07</b> | <b>96.07</b> | <b>96.07</b>                 | <b>96.07</b>                 | <b>96.07</b>   | <b>1</b> | <b>1</b> | <b>1</b> | <b>1</b>                     | <b>1</b>                     | <b>1</b>       |
| DS <sub>9</sub>  | <b>43.00</b> | <b>43.00</b> | <b>43.00</b> | <b>43.00</b>                 | <b>43.00</b>                 | <b>43.00</b>   | <b>1</b> | <b>1</b> | <b>1</b> | <b>1</b>                     | <b>1</b>                     | <b>1</b>       |
| DS <sub>10</sub> | 80.28        | <b>84.71</b> | 80.51        | 82.94                        | <b>84.71</b>                 | <b>84.71</b>   | 4        | <b>1</b> | 3        | 2                            | <b>1</b>                     | <b>1</b>       |
| Ave              | 73.053       | 76.706       | 74.478       | 77.279                       | 76.353                       | 75.396         | 2.6      | 2.0      | 2.5      | <b>1.2</b>                   | 1.7                          | 2.0            |

Bold values indicate the the highest accuracy and the highest rank achieved by the algorithms in 10 datasets

to the average rank in Table 7, it can be emphasized that the best performing methods are *SQWALKING<sub>H</sub>* with the average rank 1.2, following with *SQWALKING<sub>A</sub>* with the average rank 1.7, and both *GASEN-b* and *HIEP* with average rank of 2.0.

Table 8 reports the size of the pruned ensemble for six ensemble pruning methods on each dataset (for the homogeneous initial ensemble). All values reported in Table 8 have been rounded to gain integer values. The average obtained over all datasets shows that the *FSS* leads to the smallest size pruned ensembles with average size of 5.7, while the *SQWALKING<sub>A</sub>* and *GASEN-b* are in the second and the third positions, with average sizes of 7.8 and 8.0, respectively. *SQWALKING<sub>H</sub>*, with 8.9, *UAEP* with 11.9, and *HIEP* with the average size of 15.2, lead to the fourth, the fifth, and the sixth positions, respectively.

The results of the average running time of the six ensemble pruning methods obtained over all datasets (for the homogeneous initial ensemble) are presented in Table 9. It can be seen that the pruning methods have the same ranks as compared to the heterogeneous initial ensemble. Here, *FSS* with the running time of 0.751 s is the fastest algorithm. *UAEP* with 1.19 s, *HIEP* with 1.45 s, *GASEN-b* with 1 min and 16 s, and *SQWALKING<sub>A</sub>* with an average time of 6 min are in the subsequent positions, respectively. *SQWALKING<sub>H</sub>* with running time of 20 min and 10 s is the slowest algorithm.

### 6.2 Results of Group B

Table 10 shows the accuracy of the Bagging, AdaBoost, Random Forest, and *SQWALKING<sub>S</sub>* on each dataset. In this table, a bold value shows the most accurate method employed on each dataset. Table 10 shows that *SQWALKING<sub>A</sub>* has the highest accuracy in 5 datasets, while *SQWALKING<sub>H</sub>* is the most accurate method in 7 datasets. Random Forest in 3 datasets and both Bagging and AdaBoost in 2 datasets are the most accurate algorithms. According to the average in the last row in Table 10, it can be found out that *SQWALKING<sub>H</sub>*, with an average accuracy of 77.279% is the most accurate method. *SQWALKING<sub>A</sub>* is the second most accurate method, with an average accuracy of 76.353%. Random Forest with 75.045%, Bagging with 72.749%, and AdaBoost with average 62.231%, achieve the third, the fourth, and the fifth positions, respectively.

In addition to the accuracy, the size of the pruned ensembles via *SQWALKING<sub>S</sub>* is brought in separate columns in Table 10. Note that, Bagging, AdaBoost, and Random Forest use all 100 models of the initial ensemble. According to the average size of pruned ensemble in the last row of Table 10, *SQWALKING<sub>A</sub>* leads to the smaller size pruned ensemble compared to *SQWALKING<sub>H</sub>*. It is obvious that

**Table 8** Group A (Homogeneous Initial Ensemble): Size Of Pruned Ensembles, For The Six Ensemble Pruning Methods On Each Dataset

| Dataset ID       | Size of pruned ensemble |             |             |                              |                              |                |
|------------------|-------------------------|-------------|-------------|------------------------------|------------------------------|----------------|
|                  | <i>FSS</i>              | <i>HIEP</i> | <i>UAEP</i> | <i>SQWALKING<sub>H</sub></i> | <i>SQWALKING<sub>A</sub></i> | <i>GASEN-b</i> |
| DS <sub>1</sub>  | <b>4</b>                | 20          | 19          | 8                            | 7                            | 8              |
| DS <sub>2</sub>  | 6                       | 41          | 5           | <b>3</b>                     | 4                            | 11             |
| DS <sub>3</sub>  | <b>1</b>                | 20          | 4           | <b>1</b>                     | <b>1</b>                     | <b>9</b>       |
| DS <sub>4</sub>  | 13                      | 54          | 31          | 29                           | 17                           | 9              |
| DS <sub>5</sub>  | 6                       | <b>1</b>    | <b>1</b>    | 4                            | 6                            | 12             |
| DS <sub>6</sub>  | 6                       | <b>1</b>    | 31          | 18                           | 22                           | 8              |
| DS <sub>7</sub>  | 3                       | <b>1</b>    | <b>1</b>    | 11                           | 5                            | 11             |
| DS <sub>8</sub>  | <b>1</b>                | <b>1</b>    | <b>1</b>    | <b>1</b>                     | <b>1</b>                     | <b>1</b>       |
| DS <sub>9</sub>  | 10                      | 12          | 14          | 12                           | 10                           | <b>9</b>       |
| DS <sub>10</sub> | 7                       | <b>1</b>    | 12          | 2                            | 5                            | 2              |
| Ave              | <b>5.7</b>              | 15.2        | 11.9        | 8.9                          | 7.8                          | 8.0            |

Bold values indicate the smallest size pruned ensembles achieved by the algorithms on 10 datasets

**Table 9** Group A (Homogeneous Initial Ensemble): Average Running Time Of The Six Ensemble Pruning Methods, Obtained Over All Datasets

| Running time |             |             |                              |                              |                |
|--------------|-------------|-------------|------------------------------|------------------------------|----------------|
| <i>FSS</i>   | <i>HIEP</i> | <i>UAEP</i> | <i>SQWALKING<sub>H</sub></i> | <i>SQWALKING<sub>A</sub></i> | <i>GASEN-b</i> |
| 0.751 s      | 1.45 s      | 1.19 s      | 20 min, 10 s                 | 6 min                        | 1 min, 16 s    |

*SQWALKING<sub>S</sub>* compared to unpruned ensemble methods can lead to low computational overheads.

Table 11 shows the average running time of these ensemble methods obtained over all datasets. *SQWALKING<sub>S</sub>* are very time consuming methods compared to Bagging, AdaBoost, and Random Forest. These ensemble methods are implemented about less than 2 s; however *SQWALKING<sub>S</sub>* take more than several minutes.

**Table 10** Groups B and C: Accuracy Of Seven Ensemble Methods, On Each Dataset, And The Size Of Pruned Ensembles Via *SQWALKING<sub>S</sub>*

| Dataset ID       | Accuracy (%)                                   |              |               |   |                              |   |             | Size of pruned ensembles via <i>SQWALKING<sub>S</sub></i> |                              |
|------------------|--|--------------|---------------|---|------------------------------|---|-------------|---|------------------------------|
|                  | Popular ensemble methods without pruning phase |              |               | Our ensemble methods with pruning phase |                              | Combination methods without pruning phase |             | <i>SQWALKING<sub>H</sub></i>                              | <i>SQWALKING<sub>A</sub></i> |
|                  | Bagging  | AdaBoost     | Random forest | <i>SQWALKING<sub>H</sub></i>            | <i>SQWALKING<sub>A</sub></i> | Majority voting                           | Soft voting |   |                              |
| DS <sub>1</sub>  | 72.02  | 45.86        | <b>83.12</b>  | <b>76.80</b>                            | 73.33                        | 71.43                                     | 71.43       | 8   | <b>7</b>                     |
| DS <sub>2</sub>  | 78.78  | 68.32        | 76.89         | <b>90.20</b>                            | 88.80                        | 72.64                                     | 71.70       | <b>3</b>  | 4                            |
| DS <sub>3</sub>  | 78.87  | <b>83.90</b> | 76.46         | 79.50                                   | <b>83.90</b>                 | 63.38                                     | 63.38       | <b>1</b>  | <b>1</b>                     |
| DS <sub>4</sub>  | 52.49  | 43.91        | 51.67         | <b>53.47</b>                            | 52.00                        | 54.69                                     | 54.29       | 29  | <b>17</b>                    |
| DS <sub>5</sub>  | <b>84.91</b>                                   | 57.91        | 81.46         | <b>84.91</b>                            | <b>84.91</b>                 | 78.95                                     | 78.95       | 4   | 6                            |
| DS <sub>6</sub>  | 63.01  | 41.09        | 73.97         | <b>76.40</b>                            | 73.97                        | 57.53                                     | 57.53       | <b>18</b>   | 22                           |
| DS <sub>7</sub>  | 84.74  | 77.27        | 86.90         | <b>89.50</b>                            | 82.84                        | 79.25                                     | 79.25       | 11  | <b>5</b>                     |
| DS <sub>8</sub>  | <b>96.07</b>                                   | <b>96.07</b> | <b>96.07</b>  | <b>96.07</b>                            | <b>96.07</b>                 | 96.07                                     | 96.07       | <b>1</b>  | <b>1</b>                     |
| DS <sub>9</sub>  | 38.46  | 24.48        | 39.20         | <b>43.00</b>                            | <b>43.00</b>                 | 39.13                                     | 40.86       | 12  | <b>10</b>                    |
| DS <sub>10</sub> | 78.14  | 83.50        | <b>84.71</b>  | 82.94                                   | <b>84.71</b>                 | 76.09                                     | 76.09       | <b>2</b>  | 5                            |
| Ave              | 72.749   | 62.231       | 75.045        | <b>77.279</b>                           | 76.353                       | 68.916                                    | 68.955      | 8.9   | 7.8                          |

Bold values indicate the highest accuracy and the smallest size pruned ensembles achieved by the algorithms

**Table 11** Groups B and C: Average Running Time Of Seven Ensemble Methods Obtained Over All Datasets

| Running time |          |               |                              |                              |                 |             |
|--------------|----------|---------------|------------------------------|------------------------------|-----------------|-------------|
| Bagging      | AdaBoost | Random forest | <i>SQWALKING<sub>H</sub></i> | <i>SQWALKING<sub>A</sub></i> | Majority voting | Soft voting |
| 1.794 s      | 1.782 s  | 1.688 s       | 20 min, 10 s                 | 6 min                        | 1.34 s          | 1.48 s      |

### 6.3 Results of Group C

Table 10 shows the accuracy of majority voting, soft voting, and *SQWALKINGs*, on each dataset. Table 10 shows that *SQWALKING<sub>A</sub>* has the highest accuracy in 5 datasets, while *SQWALKING<sub>H</sub>* in 7 datasets. Majority voting in 2 datasets and soft voting in 1 dataset achieve the most accurate ensembles. On average, *SQWALKING<sub>H</sub>*, with an average accuracy of 77.279% is the most accurate method. *SQWALKING<sub>A</sub>* is the second most accurate method, with average accuracy 76.353%. Soft voting with 68.955% and majority voting with average accuracy 68.916%, achieve the third and the fourth positions, respectively. Soft voting and majority voting contain all 100 models of the initial ensemble; thus, comparing to *SQWALKINGs*, which use a subset of models in final ensembles, they lead to higher computational overheads. The strong point of these voting methods is the running time (as shown in Table 11). The majority of voting runs about 1.34 s, while the soft voting runs about 1.48 s; however, *SQWALKINGs* take several minutes.

## 7 Discussion

The *SQWALKING* procedure is investigated by comparing to ensemble pruning methods (*Group A*), popular ensemble methods without pruning phase (*Group B*), and combination methods without pruning phase (*Group C*), in the entirely same conditions.

In *Group A*, for the case of the heterogeneous initial ensemble, it can be observed that *FSS*, *HIEP*, *UAEP*, and *SQWALKING* (both versions) lead to more accurate pruned ensembles compared to *GASEN-b* based on the average calculated over all datasets. Moreover, they lead to pruned ensembles in which the number of models is less than 15% of the number of models in the initial ensemble. For the reason that, except *GASEN-b*, all above-mentioned methods are based on the forward selection architecture, it can be suggested that the architecture of the forward selection is efficient in practice. In *Group A*, for the case of the homogeneous initial ensemble, almost all methods lead to accurate pruned ensembles and the number of models in all pruned ensembles is less than 16% of the number of models in the initial ensemble. It is noteworthy that *GASEN-b* shows better results on the case of a homogeneous initial ensemble than the heterogeneous initial ensemble. Generally, on average, it can be observed that *SQWALKING<sub>H</sub>* and *SQWALKING<sub>A</sub>* lead to the most accurate pruned ensembles compared to the other pruning methods in *Group A*, respectively. It indicates the strength of the prediction of *SQWALKING*, especially *SQWALKING<sub>H</sub>*. Regarding the size of the pruned ensembles, however, *SQWALKING<sub>H</sub>* and *SQWALKING<sub>A</sub>* do not

lead to the smallest size pruned ensembles; as they lead to the acceptable size of pruned ensembles.

In *groups B* and *C*, *SQWALKINGs* is compared with popular ensemble methods, which do not have pruning phase and they apply all models of the initial ensemble in the predictions. Therefore, the computational overheads of these ensemble methods are more than *SQWALKINGs*, which use a subset of the initial ensemble. Moreover, an average *SQWALKINGs* (especially, *SQWALKING<sub>H</sub>*) leads to more accurate ensembles compared to other methods. These results can show that the pruning phase is an effective middle phase for an ensemble method; with decreasing computational overheads and increasing overall accuracy of the ensemble.

As seen in *Group C*, *SQWALKINGs* are more effective than the soft voting combination method. Moreover, as mentioned in Sect. 5, soft voting applies in *SQWALKINGs* as the combination method. So, it can be concluded that with accompanying the soft voting and *SQWALKING*, the effectiveness of the soft voting will be increased. Therefore, when novel combination methods such as [8–11] increase the overall prediction performance of the initial ensemble, this increment can be more, if these methods [8–11] are accompanied by ensemble pruning methods like *SQWALKINGs*. Thus, by applying the pruning phase and eliminating some redundant and useless models, computational overheads of these combination methods may be decreased.

Generally, these promising results prove that our proposed Upward Stochastic Walking idea works in practice effectively. Moreover, as seen on 10 datasets, *SQWALKING<sub>H</sub>* compared to *SQWALKING<sub>A</sub>* reaches to valuable results in terms of the accuracy and the size of pruned ensembles. However, the disadvantage of *SQWALKING<sub>H</sub>* is that it is time-consuming. Note that, the difference between *SQWALKING<sub>H</sub>* and *SQWALKING<sub>A</sub>* is the evaluation measures; *HLF* is the diversity-based evaluation measure in *SQWALKING<sub>H</sub>* and *ACC* is a performance-based evaluation measure in *SQWALKING<sub>A</sub>*. So, it can be concluded that the mentioned successes of *SQWALKING<sub>H</sub>* are related to the measure *HLF* (human-like foresight) [15] and even being time-consuming of *SQWALKING<sub>H</sub>* is due to *HLF*. This time consuming is reasonable from 2 points of views:

1. Taking more time for running is due to more searches done by *SQWALKINGs*. These more searches will lead to finding the most accurate and small size subset of classifiers.
2. As explained in Sect. 4, our proposed pruning phase is done one time and in an offline manner. So, taking more time and additional computational costs only once.

Regarding the efficiency of *SQWALKING*, especially *SQWALKING<sub>H</sub>*, it can be suggested as a tool to recognize

the most proper classification algorithms for a given classification problem. So that firstly, the more proper types of classification algorithms are selected. Then, *SQWALKING<sub>H</sub>* prunes the initial ensemble of these classifiers. Thus, by studying the type of classifiers in the pruned ensemble, *SQWALKING<sub>H</sub>* can find the most capable types of classifiers. Subsequently, these selected types of classifiers can be applied as singular classifiers or ensemble methods classifiers for a given problem.

In addition to the above results, some important issues are sensible. In some datasets, heterogeneous initial ensembles are more effective than homogeneous initial ensembles. Adjusting the values of the pruning parameters may affect the results. Moreover, it is possible the other ensemble combination methods are selected. Therefore, it can be concluded that except the pruning procedure, the other issues such as: (1) Related information to the classification problem, number of features, classes, and instances; (2) Type of initial ensemble; heterogeneous, homogeneous, or a combination of both; (3) Type of base classifiers and their number in the initial ensemble; (4) Number of instances in the training dataset; (5) Type of the pruning procedure, and adjusting the parameters; (6) Number of instances in the pruning dataset; (7) Type of the combination methods may affect reaching the success on each classification problem.

## 8 Testing *SQWALKING* on a real-world data

In this section, *SQWALKING* is tested on a real-world dataset named “human miRNA” [44]. The dataset can be downloaded from the following URL (<http://lab.malab.cn/soft/LibD3C/data.html>). The 73% and 27% of instances in training dataset file are used as training and pruning datasets, respectively; and all instances in testing dataset file are used as testing dataset. For this test, we again applied heterogeneous initial ensemble, *SQWALKING* parameter tuning, and ensemble combination method, mentioned in Sect. 5 [*Group A* (heterogeneous initial ensemble)].

As shown in Table 12, *SQWALKING<sub>A</sub>* achieves the high accuracy (97.91%) on the dataset, while *SQWALKING<sub>H</sub>* achieves less accuracy (97.59%). The size of the final ensembles of two methods is large and as expected the running time of the two methods is relatively high. These results are

reasonable when the overall accuracy is more important, compared to other factors, on this dataset.

## 9 Conclusion and future works

In this paper, a new ensemble pruning approach named *SQWALKING* (Simulated Quenching *WALKING*) was proposed. In *SQWALKING*, simulated quenching and forward selection methods were incorporated to reach the goal of the upward stochastic walking idea. *SQWALKING* with the aid of probabilistic steps select the models progressively until the best subset is found. Moreover, two versions of *SQWALKING* were suggested, namely *SQWALKING<sub>A</sub>* and *SQWALKING<sub>H</sub>*. These versions were based on two different evaluation measures, accuracy, and human-like foresight, respectively.

*SQWALKING* (both versions) was evaluated via comparison with four analogous ensemble pruning methods for pruning heterogeneous and homogeneous initial ensembles of 100 classifiers, on ten machine learning problems. The related results in the heterogeneous case demonstrated that *SQWALKING<sub>A</sub>* led to the most accurate pruned ensemble on 40% of the datasets, while *SQWALKING<sub>H</sub>* on 60% of the datasets. Besides, it was demonstrated in the homogeneous case that *SQWALKING<sub>A</sub>* led to the most accurate pruned ensemble on 50% of the datasets, while *SQWALKING<sub>H</sub>* on 80% of the datasets. Moreover, *SQWALKING<sub>S</sub>* led to small size pruned ensembles included less than 12% of models of the initial ensemble, in the case of averaging over all datasets (for both cases). However, the results showed that *SQWALKING<sub>S</sub>* were time-consuming, especially version *H*. However, due to the fact that the pruning operation in the paper was applied offline, extra computational costs were constrained one time. Thus, the costs were reasonable; especially when taking more time led to finding a more effective subset of the initial ensemble (as *SQWALKING<sub>H</sub>* shown). Therefore, it can be concluded that *SQWALKING<sub>H</sub>* was more efficient than *SQWALKING<sub>A</sub>*. It was also shown that the human-like foresight evaluation measure was more efficient than the accuracy evaluation measure. Moreover, extensive experimental comparisons of *SQWALKING<sub>S</sub>* against popular ensemble methods and combination methods showed very promising

**Table 12** Accuracy, Size of pruned ensemble, and average running time of *SQWALKING<sub>S</sub>*, On “Human miRNA” dataset

| Pruned method                | Accuracy (%) | Size of pruned ensemble |           |           |          |          |           | Average running time (min) |
|------------------------------|--------------|-------------------------|-----------|-----------|----------|----------|-----------|----------------------------|
|                              |              | $N_{MLP}$               | $N_{KNN}$ | $N_{SVM}$ | $N_{DT}$ | $N_{NB}$ | All       |                            |
| <i>SQWALKING<sub>H</sub></i> | 97.59        | <b>30</b>               | <b>0</b>  | <b>30</b> | <b>0</b> | <b>0</b> | <b>60</b> | 116                        |
| <i>SQWALKING<sub>A</sub></i> | <b>97.91</b> | 23                      | 10        | 12        | 28       | 2        | 75        | <b>39</b>                  |

Bold values indicate the higher accuracy, the smaller size pruned ensemble, and the less average running time achieved by the two algorithm



results. It indicated that the pruning phase is important for improving the efficiency of ensemble systems.

*SQWALKINGs* could increase the prediction performance of ensemble systems and decrease the computational overheads. So, they could be utilized in the systems in which the overhead of the system was as important as prediction performance. *SQWALKINGs* were independent of the ensemble construction and combination phases. Hence, proper ensemble construction and effective combination methods can be applied. *SQWALKINGs* are introduced as ensemble pruning methods to solve some classification problems in this field. Moreover, they were introduced as a tool to recognize the capabilities of classification algorithms. *SQWALKINGs* can be applied in distributed data mining systems in order to observe their capability in reducing communication overheads.

For future work, two new extensions of our method can be considered. Regarding the remarkable success of deep learning in data mining and machine learning fields, it is suggested utilizing deep learning to introduce novel ensemble pruning methods. Firstly, deep learning can be applied as an expert fitness function in *SQWALKING* architecture to find the best classifier in each step of the method. Secondly, deep learning and forward selection method can be incorporated for selecting models through the initial ensemble. Developing the learning-based ensemble pruning approach may lead to effective results for future research. Moreover, *SQWALKING* can be suggested to solve the classification problems, especially for real-world classification problems.

**Acknowledgements** The authors would like to acknowledge the efforts and the consideration of the editor and all reviewers for their valuable comments and suggestions to improve the quality of the paper.

## References

- Kuncheva LI (2004) Combining pattern classifiers, methods and algorithms. Wiley, Hoboken
- Tan PN, Steinbach M, Kumar V (2006) Introduction to data mining. Addison-Wesley Longman Publishing Co., Inc., Boston
- Dietterich TG (2000) Ensemble methods in machine learning. Springer, Berlin
- Prodromidis A, Chan P (2000) Meta-learning in distributed data mining systems: issues and approaches. In: Advances of distributed data mining. AAAI Press, Palo Alto
- Caruana R et al (2004) Ensemble selection from libraries of models. In: Proceedings of the 21st international conference on machine learning, ACM: Banff, Canada. p 137–144
- Margineantu D, Dietterich T (1997) Pruning adaptive boosting. In: Proceedings of the 14th international conference on machine learning. Morgan Kaufmann Publishers Inc. p 211–218
- Zhou ZH, Wu J, Tang W (2002) Ensemble neural networks: many could be better than all. *Artif Intell* 137(1–2):239–263
- Ekbal A, Saha S (2011) A multiobjective simulated annealing approach for classifier ensemble: named entity recognition in Indian languages as case studies. *Expert Syst Appl* 38(12):14760–14772
- Ekbal A, Saha S (2011) Weighted vote-based classifier ensemble for named entity recognition: a genetic algorithm-based approach. *ACM Trans Asian Lang Inf Process* 10(2):1–37
- Ekbal A, Saha S (2013) Simulated annealing based classifier ensemble techniques: application to part of speech tagging. *Inf Fusion* 14(3):288–300
- Saha S, Ekbal A (2013) Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data Knowl Eng* 85:15–39
- Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Hoboken, p 518
- Banfield RE et al (2005) Ensemble diversity measures and their application to thinning. *Inf Fusion* 6(1):49–62
- Partalas I, Tsoumakas G, Vlahavas I (2010) An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. *Mach Learn* 81(3):257–282
- Taghavi ZS, Sajedi H (2013) Human-inspired ensemble pruning using hill climbing algorithm. In: The 5th Robocup Iranian International Symposium and the 3rd Joint Conference of Robotics and AI. IEEE: Qazvin-Iran. p 1–7
- Taghavi ZS, Sajedi H (2014) Ensemble selection using simulated annealing walking. In: The proceedings of the international conference on advances in computing, electronics and electrical technology. Institute of research engineers and doctors: Kuala Lumpur, Malaysia
- Taghavi ZS, Sajedi H (2015) Ensemble pruning based on oblivious Chained Tabu searches. *Int J Hybrid Intell Syst* 12(3):131–143
- Zhou ZH, Tang W (2003) Selective ensemble of decision trees. Springer, Berlin
- Tamon C, Xiang J (2000) On the boosting pruning problem. In: The 11th European conference on machine learning, ECML. Springer, Berlin, p 404–412
- Ingber L (1993) Simulated annealing: Practice versus theory. *Math Comput Model* 18(11):29–57
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Breiman L (1996) Bagging predictors. *Mach Learn* 24(2):123–140
- Schapire RE (1990) The strength of weak learnability. *Mach Learn* 5(2):197–227
- Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55(1):119–139
- Parmanto B, Munro P, Doyle H (1996) Improving committee diagnosis with resampling techniques. *Adv Neural Inf Process Syst* 8:882–888
- Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
- Zhou ZH (2012) Ensemble methods: foundations and algorithms. CRC Press, Boca Raton
- Xia X, Lin T, Chen Z (2018) Maximum relevancy maximum complementary based ordered aggregation for ensemble pruning. *Appl Intell* 48:2568–2579
- Guo H et al (2018) Margin and diversity based ordering ensemble pruning. *Neurocomputing* 275:237–246
- Li D et al (2018) RTCRelief-F: an effective clustering and ordering-based ensemble pruning algorithm for facial expression recognition. *Knowl Inf Syst*. <https://doi.org/10.1007/s10115-018-1176-z>
- Onan A, Korukoğlu S, Bulut H (2017) A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification. *Inf Process Manag* 53(4):814–833
- Lin C et al (2014) LibD3C: ensemble classifiers with a clustering and dynamic selection strategy. *Neurocomputing* 123:424–435

33. Zhang H, Cao L (2014) A spectral clustering based ensemble pruning approach. *Neurocomputing* 139:289–297
34. Sheen S, Anitha R, Sirish P (2013) Malware detection by pruning of parallel ensembles using harmony search. *Pattern Recognit Lett Innov Knowl-based Tech* 34(14):1679–1686
35. Narassiguin A, Elghazel H, Aussem A (2017) Dynamic ensemble selection with probabilistic classifier chains. In: *Machine learning and knowledge discovery in databases*, p 169–186
36. Mozaffari A et al (2017) A hierarchical selective ensemble randomized neural network hybridized with heuristic feature selection for estimation of sea-ice thickness. *Appl Intell* 46(1):16–33
37. Ye R, Dai Q (2018) A novel greedy randomized dynamic ensemble selection algorithm. *Neural Process Lett* 47(2):565–599
38. Bandyopadhyay S et al (2008) A simulated annealing based multi-objective optimization algorithm: AMOSA. *IEEE Trans Evol Comput* 12(3):269–283
39. Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, p 372
40. Deb K et al. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2): 181–197
41. Wang X et al (2015) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. *IEEE Trans Fuzzy Syst* 23(5):1638–1654
42. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: *Proceedings of the 13th International conference on international conference on machine learning*. Morgan Kaufmann Publishers Inc., Bari, Italy. p 148–156
43. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
44. Wei L et al (2014) Improved and promising identification of human micrnas by incorporating a high-quality negative set. *IEEE/ACM Trans Comput Biol Bioinform* 11(1):192–201

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.