International Journal of Applied Metaheuristic Computing
Volume 13 • Issue 1

Metaheuristic Ensemble Pruning via Greedy-Based Optimization Selection

Mergani Ahmed Eltahir Khairalla, Nile Valley University, Sudan*
https://orcid.org/0000-0003-1483-3196

ABSTRACT

Ensemble selection is a crucial problem for ensemble learning (EL) to speed up the predictive model, reduce the storage space requirements, and to further improve prediction accuracy. Diversity among individual predictors is widely recognized as a key factor to successful ensemble selection (ES), while the ultimate goal of ES is to improve its predictive accuracy and generalization of the ensemble. Motivated by the problems stated, the authors have devised a novel hybrid layer-based greedy ensemble reduction (HLGER) architecture to delete the predictor with lowest accuracy and diversity with evaluation function according to the diversity metrics. Experimental investigations are conducted based on benchmark time series data sets. Support vectors regression algorithm utilized as base learner to generate homogeneous ensemble, HLGER uses locally weight ensemble (LWE) strategies to provide a final ensemble prediction. The experimental results demonstrate that, in comparison with benchmark ensemble pruning techniques, HLGER achieves significantly superior generalization performance.

KEYWORDS

Ensemble Pruning, Greedy Ensemble Selection

1. INTRODUCTION

Ensemble learning (EL) has been successfully used as a desirable learning scheme for many regression and classification problems because of its potential to greatly increase predictive accuracy (Idris, Khan, & Lee, 2013; Nadig, Potter, Hoogenboom, & Mcclendon, 2013; P. Sun & Lee, 2014). An ensemble refers to a group of base learners whose decisions are aggregated with the goal of achieving better performance than its constituent members (Caruana, Munson, & Niculescu-Mizil, 2006). Typically, EL algorithm consists of two main stages: first, the production of multiple base classifiers for one specific task, and then the combination of these classifiers to get a final predictive decision (Banfield, Hall, Bowyer, & Kegelmeyer, 2005; Li, Yu, & Zhou, 2012; Tsoumakas, Partalas, & Vlahavas, 2009).

However, it is obvious that combining all of the classifiers in an ensemble adds a lot of computational overheads (Tsoumakas et al., 2009) (Polikar, 2009). Both theoretical and empirical studies have shown that instead of using the whole ensemble, a subset of the ensemble can achieve equivalent or even better generalization performance (Lu, Wu, Zhu, & Bongard, 2010; Y. Zhang, Burer, & Street, 2006). Therefore, an additional intermediate stage that deals with the selection of an appropriate sub-ensemble prior to its combination has to be considered (Dai, 2013a; Dai & Liu, 2013). This stage is generally termed as ensemble pruning (EP) (Lu et al., 2010), selective ensemble

DOI: 10.4018/IJAMC.292501 *Corresponding Author

(C. X. Zhang & Zhang, 2011), ensemble selection(Maskouni, Hosseini, Abachi, Kangavari, & Zhou, 2018), or ensemble thinning (Banfield et al., 2005).

The problem of pruning an ensemble of classifiers has been proven to be NP-complete (Tamon & Xiang, 2000). Finding the best sub-ensemble through exhaustive searching is not feasible for the original ensemble with large or even moderate size (Partalas, Tsoumakas, & Vlahavas, 2008).

EP can be further categorized into two types, namely static pruning (SP) and dynamic pruning (DP) methods (Soto, García-Moratilla, Martínez-Muñoz, Hernández-Lobato, & Suárez, 2017). In SP methods, a fixed set of predictors from an initial pool is selected from the ensemble for all test patterns, while DP methods predictors are selected based on the test pattern. Of the SP methods, search-based (Jiang, Liu, Fu, & Wu, 2017), clustering-based (Fan, Tao, Zhou, & Han, 2017), optimization-based (Lessmann, Caserta, & Arango, 2011), ordered aggregation methods (L. Guo & Boukir, 2013), and other methods exist that have a combination of these categories or use elaborate pruning methods are the most commonly used (C. X. Zhang & Zhang, 2011).

The greedy algorithms (Hernández-Lobato, Martínez-Muñoz, & Suárez, 2011) (Partalas, Tsoumakas, & Vlahavas, 2012) (Martínez-Muñoz, Hernández-Lobato, & Suárez, 2008), also known as hill climbing algorithms, which reduce the search space appropriately, seem to be a good choice. Various greedy ensemble selecting (GES) algorithms have been proposed, just as in Refs. (Banfield et al., 2005) (Martínez-Muñoz & Suárez, 2004) (Abdesslem, Marwa, & Maroua Bencheikh, 2013) (Dai, 2013b) (Gevezes & Pitsoulis, 2015) (Pérez-Gállego, Castaño, Quevedo, & Coz, 2018) (Partalas, Tsoumakas, & Vlahavas, 2010).

There are two key elements for GES algorithms: the search direction and evaluation measure(Partalas et al., 2012) (Partalas et al., 2010) (Baron, 2019). Typically, there are two search directions, i.e., forward expansion and backward shrinkage. Many researchers have compared the effect of the two different search directions in their research works (Banfield et al., 2005) (Martínez-Muñoz & Suárez, 2004) (Partalas et al., 2012) (Partalas et al., 2010) (Q. Sun & Pfahringer, 2012). As to the factor of evaluation measure, various evaluation measures have been proposed to guide the pruning algorithm and have made a success. The measures of uncertainty weighted accuracy (UWA) (Partalas et al., 2012) (Partalas et al., 2010), complementariness (COM) (Martínez-Muñoz & Suárez, 2004) and concurrency (CON) (Banfield et al., 2005), which will be discussed in detail in Section 2, are the three diversity measures particularly suitable for GES algorithm. It is worth to mention that UWA can be seen as an improvement to COM and CON, and it leads to better performance compared to the other two.

It is found through our investigation that, evaluation measures adopted in GES algorithms are usually focused on the ensemble diversity or its accuracy solely, while ignore the counterpart. However, actually, both diversity and accuracy are crucial in the ensemble, and they interrelate with each other closely. A successful ensemble should possess a sufficiently high level of diversity, while an ensemble shows the success in its good predictive accuracy. If an ensemble possesses better predictive accuracy than its constituent members, its diversity is expected to reach a high enough level. Equivalently, if an ensemble does not possess sufficiently high error-correction ability among its components, combination will do little to improve its classification performance. Accordingly, diversity and accuracy should not be considered separately, but rather, they should be taken into account simultaneously. This is the major argument of this study. We found in this work that through considering diversity and accuracy simultaneously for ES, pruned ensemble with excellent classification performance and superior generalization capability can be achieved, which is just the scientific value added of our paper.

Contribution A novel hybrid layered based greedy ensemble reduction (HLDER) approach is proposed for improving the prediction accuracy and generalization of the ensemble, that change the order in which ensemble predictors are combined. The proposed method modifies the order of aggregation through distributing the ensemble selection over the entire training set, which is then dynamically used based on closeness of a test pattern to the training patterns. This dynamic method

is compared, using the Reduced Error Pruning method without back fitting, with other static methods as well as incorporating them in this dynamic approach.

2. RELATED WORK

This section briefly reviews the related work on the ensemble diversity and pruning.

2.1 Background Knowledge of the Diversity and Accuracy

Diversity is an important property for an ensemble of base classifiers, which is usually measured on the basis of a precise pruning dataset (Dai & Liu, 2013). The more uniformly distributed in the errors are, the greater the ensemble diversity could be, and vice versa (Gang, Zhang, Jian, & Cheng, 2011; Li et al., 2012). The property of diversity cares about the characteristics of difference and similarity among the base classifiers in an ensemble.

In (Li, Yu, & Zhou), Zhou et al. theoretically analyzed the effect of diversity on the generalization performance of ensemble combined with voting method in the PAC-learning framework, and concluded that enhancing diversity could be a good way to realize selective ensemble learning. Empirical results have demonstrated that there exists a correlation between the accuracy of an ensemble and the diversity among its base classifiers (Cavalcanti, Oliveira, Moura, & Carvalho, 2016) (H. Guo et al., 2017). However, it is hard to identify this correlation.

Although the difference among individual classifiers is widely recognized to be the key issue to the success of an ensemble, however, diversity cannot guarantee the generalization capability of the final pruned ensemble (Idris et al., 2013). One reason is that there does not exist any generally accepted formal definition of the ensemble diversity in the literature (Li et al., 2012).

Pruning the ensemble according to the accuracy of its base learners while maintaining a sufficiently high level of diversity among them is typically considered a proper approach for effective ensemble selection (Tsoumakas et al., 2009). Both of the two factors of ensemble diversity and accuracy should be considered simultaneously, so as to ensure that the success of the ensemble selection as far as possible. Motivated by the above ideas, in this paper, we propose three different measures which take into account both of the two crucial factors, i.e. diversity and accuracy, at the same time.

With regard to the question of why diversity and accuracy are two crucial factors for ensemble selection, we present the following explanations.

First, diversity has a very significant impact on the ensemble's performance. Suppose that the predictive decisions of base classifiers in the ensemble are very close to each other, then, combination will do little to improve its classification performance, which is against our expectation. Second, accuracy is obviously another essential factor for ensemble selection. The ultimate goal of ensemble pruning is to improve its predictive accuracy. When an ensemble has a high predictive accuracy on a pruning dataset, it can be more likely to correctly classify an unseen sample. Therefore, diversity and accuracy are both crucial for ensemble selection.

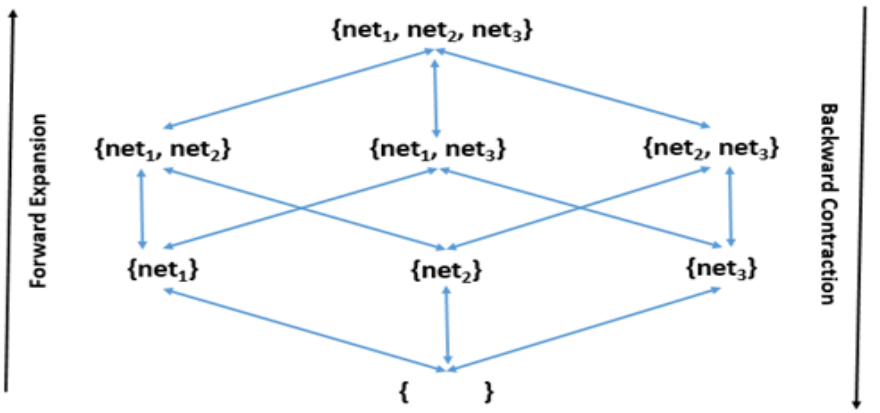
In regard to the other crucial factors for ensemble selection, different diversity measure criterions, evaluation measures for assessing each specific ensemble and pruning approaches utilized, etc., are very important factors. Compared to these decisive factors, there are also some non-crucial ones. For example, it seems that whether base classifiers is heterogeneous or homogeneous is not that vital. In addition, which base learner is utilized to generate the initial ensemble is also not that vital. Finally, size of the original ensemble before pruning is not crucial.

2.2 Ensemble Pruning based on GES Approach

Greedy search is a relatively fast heuristic search algorithm; GES strategies make a locally optimal option with the hope that this option will result in a globally optimal solution. GES schemes, such as the directed hill climbing algorithm, greedily choose, from the neighborhood of the current state, the next state to visit (Dai, 2013b) (Liu, Dai, & Liu, 2014). States, in the EP problem investigated in

this work, are the different subsets of the initial ensemble $ENS \equiv \left\{h_i\left(x\right)\right\}_{i=1}^{L}$ of L member classifiers, where h_i denotes the i^{th} component classifier in ENS, and x denotes an instance (Partalas et al., 2010) (Dai, 2013b) (Liu et al., 2014). The neighborhood of a subset of component classifiers $S \subseteq ENS$ contains those subsets which can be constructed by expanding one component into S or deleting one component from S (Partalas et al., 2010). An example of the search space map for an ensemble initially composed of four component nets is illustrated in Figure 1. As we indicated in the previous section, there are mainly three factors in the classical GES algorithms: the search direction, evaluation dataset and evaluation measure.

Figure 1. GES algorithm for an initial three ensemble component net (Partalas et al., 2010)



2.3 The Two Search Directions of GES Algorithms

Concerning the factor of search direction of the classical GES algorithm, there are mainly two: forward and backward search directions (Partalas et al., 2010) (Liu et al., 2014), as showed in Figure 1. In the forward expanding GES algorithm (Martínez-Muñoz & Suárez, 2004) (Dias & Windeatt, 2014), the subset S is initialized as an empty set. The algorithm is implemented by iteratively expanding to S one component network $h_i \in ENS - S$ in accordance with a specific evaluation measure (Partalas et al., 2010). In contrast, in the backward contracting GES algorithm (Banfield et al., 2005), the subset S is initialized as the whole ensemble ENS, and the algorithm is implemented by iteratively removing from S one network $h_i \in S$ on the basis of a specific evaluation measure (Partalas et al., 2010).

2.4 Evaluation Measures of GES Algorithms

The evaluation measure is computed based upon the classification performance of the candidate subset of a pruning set, which is denoted as $Pr = \left\{\left(x_i, y_i\right), i = 1, 2, ..., N_{Pr}\right\}$, where x_i indicates a feature vector and y_i indicates the value of the corresponding target variable. For both of the two search directions, an amount of $L\left(\mathrm{L} + 1\right)/2$ subsets are required to be evaluated by the searching procedure. Accordingly, the computational complexity of a greedy classical ensemble selection algorithm equals $O\left(L^2 f\left(L, N_{Pr}\right)\right)$ where $f\left(L, N_{Pr}\right)$ indicates the computational complexity of the specific evaluation measure(Dai, 2013b) (Liu et al., 2014) (Dai & Li, 2015). The evaluation measures

utilized in the classical GES algorithms can be roughly separated into two categories: accuracy-focused and diversity-focused measures. Diversity is considered to be a key success factor for ensemble learning method, and a number of diversity-focused evaluation measures have been proposed, including Kappa (Banfield et al., 2005), margin distance minimization (MAR) (Martínez-Muñoz & Suárez, 2004), complementariness (COM) (Martínez-Muñoz & Suárez, 2004) (Partalas et al., 2010) (Partalas et al., 2012) (Gonzalo, Daniel, & Alberto, 2009), concurrency (CON) (Banfield et al., 2005) (Martínez-Muñoz & Suárez, 2004) (Partalas et al., 2012), Kohavi-Wolpert variance (Tang, Suganthan, & Yao, 2006) and inter-rater agreement(Tang et al., 2006). Researchers agree that diversity is proportional to the independent errors that each base classifier makes. However, no clear definition of diversity has been given up to now. The main difference among the classical greedy ensemble selection algorithms is the specific construction of their evaluation measures. In (Banfield et al., 2005), authors analyze the correlation between diversity and accuracy, and they propose a new evaluation measure, concurrency (CON). In (Martínez-Muñoz & Suárez, 2004), Martınez-Munoz and Suárez discuss the correlations among the individual classifiers in bagging, and they propose the measures of Reduce-Error pruning (RE), Complementariness (COM) and Margin Distance Minimization (MAR). In(Partalas et al., 2010), authors analyze the drawback of CON and COM, and they propose a new diversity-focused measure, i.e. Uncertainty Weighted Accuracy (UWA).

2.5 The Correlation and Difference among COM, CON and UWA

As these three diversity-focused measures, which possess similar motives, are used in our experiments, it is necessary to analyze the correlation and difference among them. Firstly, it is important to clarify the four events concerning the decisions of a component net h and a subset S with respect to an instance $\left(x_i, y_i\right)$ (Partalas et al., 2010):

$$e00\left(h,S,\ x_i,y_i\right)\ :\ h\left(x_i\right)\ \neq\ y_i\ and\ S\left(x_i\right)\ \neq\ y_i \tag{1}$$

$$e01\left(h,S,\ x_i,y_i\right)\ :\ h\left(x_i\right)\ \neq\ y_i\ and\ S\left(x_i\right)\ =\ y_i \tag{2}$$

$$e10\left(h,S,\ x_i,y_i\right)\ :\ h\left(x_i\right)\ =\ y_i\ and\ S\left(x_i\right)\ \neq\ y_i \tag{3}$$

$$e11\left(h,S,\ x_i,y_i\right)\ :\ h\left(x_i\right)\ =\ y_i\ and\ S\left(x_i\right)\ =\ y_i \tag{4}$$

The measures COM (Martínez-Muñoz & Suárez, 2004) (Gonzalo et al., 2009) and CON (Banfield et al., 2005) are calculated as follows:

$$COM\left(h,S\right)\ =\sum_{i=1}^{N_{Pr}}\|\left(e10\left(h,S,\ x_i,y_i\right)\right) \tag{5}$$

$$CON\left(h,S\right)\ =\ \sum_{i=1}^{N_{Pr}}\left(2\ \|\ e10\left(h,S,\ x_i,y_i\right)\right)$$

$$+ \left|\left|\left(e11\left(h,S,\ x_i,y_i\right)\right)-\ \left(2\left|\right|e00\left(h,S,\ x_i,y_i\right)\right)\right.\right. \tag{6}$$

where $\left|\right|\left(.\right)$ is an indicator function $\left(\left|\right|\left(true\right)\ =\ 1\right)$ and $\left(\left|\right|\left(false\right)\ =\ 0\right)\right)$ (Liu et al., 2014).

An effectual measure for EP via DHCEP termed as UWA, proposes in (Partalas et al., 2010), which takes into account the uncertainty of the decision of the current subset and has clear semantics. It defines as following:

$$\begin{aligned} UWA\left(h,S\right)\ &=\ \sum_{i=1}^{N_{Pr}}\left\|\left(e10\left(h,S,\ x_i,y_i\right)\right)NT_i\right. \\ &-\ \left(\left\|\ e00\left(h,S,\ x_i,y_i\right)NT_i\right.\right) \\ &+\ \left(\left\|\ e11\left(h,S,\ x_i,y_i\right)NF_i\right.\right) \\ &-\left(\left\|\ e01\left(h,S,\ x_i,y_i\right)NF_i\right.\right) \end{aligned} \tag{7}$$

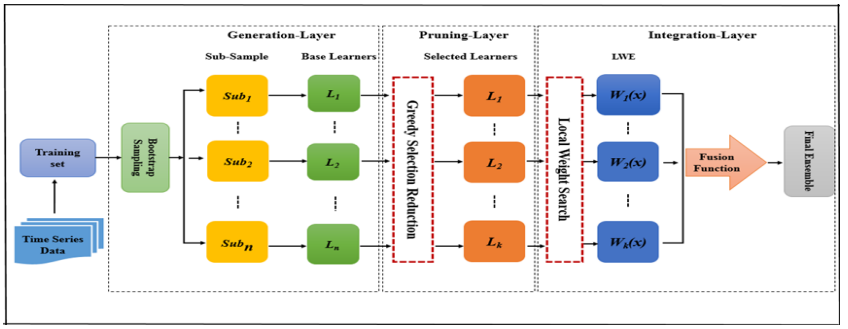
Where NT_i represents the proportion of component nets in the current subset S that correctly classify instance $\left(x_i,y_i\right)$, and $NF_i\ =\ 1\ -\ NT_i$, represents the proportion of nets in S that that provide wrong classification decision to it (Partalas et al., 2010). $\left|\right|\left(.\right)$ is an indicator function; Pr denotes the pruning dataset (Dai, 2013b) (Liu et al., 2014) (Dai & Li, 2015) (Dai & Han, 2016).

The CON measure (Banfield et al., 2005) attaches importance to the concurrence between the current candidate classifiers and the current sub-ensemble. While the COM measure attaches importance to the classifiers whose performance is complementary to the current sub-ensemble (Dai & Han, 2016) (Martínez-Muñoz et al., 2008). And UWA keeps a watchful eye on the instances on which the classifiers within the sub-ensemble greatly differ in their opinions. According to the presentation of authors in (Partalas et al., 2010), UWA embodies a very simple idea of them. Instances on which the member classifiers' decisions highly disagree with each other should be paid more attention to.

3. MATERIALS AND METHODS

Motivated by the problems stated in the previous section, we have devised a novel hybrid layered based greedy ensemble reduction (HLGER) architecture for TSP. In this section, we describe the details of HLGER algorithm. The objective of HLGER is twofold, to improve the ac curacy and diversity of the base predictors, and to improve forecasting accuracy. HLGER technique employs a layer-wise mechanism by which important lag or time window for a time series is identified first in layer one. Then using this lag information, design an ensemble where the base predictors has higher accuracy and diversity. Finally, HLGER uses a powerful combination algorithm to provide a final forecast. The significant problem of ES research is how to design practical algorithms leading to sub- ensembles without sacrificing or even improving the generalization performance contrasting to all-member ensembles. The major steps of HLGER can be described by Figure 2., which explain further as following:

Figure 2. Model of a hybrid layered-based greedy ensemble reduction (HLGER) scheme



3.1 Ensemble Generation

The process of generation of the initial set (the pool) of the base models. When all models are generated having the same induction algorithm, the approach is called homogeneous. Otherwise, it names heterogeneous. The heterogeneous approach is claimed to obtain models with higher diversity [36,37], which is important to increase the accuracy of the ensemble.

3.2 Step1: Training Set Generation

A training set in the form $\left\{X_i;\ Y_i\right\}_{i=1}^{N}$ is necessary for obtaining optimal or near optimal weight of a RBF network using the SVR algorithm. Here X_i represents input to the network and may have several components i.e., $X_i = x_{11},\ x_{12},\ \ldots,\ x_{1d}$. For the sake of simplicity and without loss of generality, we assume that the output Y_i has one component. To generate the training set for a TSP problem, we need some extra effort because only data points of a given time series are available. The parameter needed in such generation is the lag (time window), which determines how many previous data points will influence the next point.

An appropriate lag of a time series is not known in advance. As mentioned before, the aim of our ensemble generation step 1 is to find the appropriate lag. Lacking of knowledge about such a lag enforces HLGER to vary the lag from $1\ to\ l_{max}$. And HLGER generates a different training set using each of different lags. Let the lag equal to 5 and the data points $d_1,\ d_2,\ldots,\ d_k$ are used for generating the training set. The generation process takes the lag as a window and shifts it in generating the training set. That is, $X_1 = d_1 \ldots d_5$ and $Y_1 = d_6$, $X_2 = d_2 \ldots d_6$ and $Y_2 = d_7$, this process continues until the Y_i reaches at the end of the series i.e., d_k dk. It is now clear that it is possible to get a different training set by using a different lag.

3.3 Step 2: Base Predictors Generation

In the ensemble generation step 2, the training sets for base predictors are generated in two steps. At the first step, using the lag obtained from the ensemble generation step 1 and the data points $d_1,\ d_2,\ldots,\ d_k$, HLGER generates the training set, D_{tr}. In the second step, bootstrapped sampling is applied on D_{tr} for generating N training sets, one for each base predictor in the ensemble generation step 2.

3.4 Model Selection for Ensemble

GES algorithm attempts to find the globally best subset of ensembles by taking local greedy decision for changing the current subset. In this study, backward elimination is exploited in the greedy selection. Firstly, the current classifier subset S is initialized to the complete base learner set H. Then, at each iteration, the classifier h_t, S that optimizes the evaluation function f will be removed from S to improve the forecasting accuracy. With the idea of greedy methods, the evaluation function f selects the classifier h_t which has the smallest diversity in the current subset S. The iterative process will not stop until the error of ensemble subset S starts to increase. This process is named as back-forward reduction of diversity (BRD). The description of BRD is illustrated in Algorithm 1.

Table 1. Algorithm 1: Forward-backward search reduction

Algorithm 1: Forward-Backward Search Reduction (FBSR)

Input: Training set D_{train} ,Test set D_{test} ,
Original base learners $H = \{h_1, h_2, ..., h_t\}$;
Output: Selected base learners $S = \{s_1, s_2, ..., s_n\}$, $n < t$;
Begin

01 $D_{train} = \left\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\right\}$
of N samples;
02 for $i = 1$ *to* t **do**
03 $D_i = Bootstrap(D_{train})$;
04 Train $h_i \in H$ with D_i ;
05 end for;
06
07 Calculate DIV using Eq.() for H ;
08 $S = H;$
09 $E_0 = Error\left(ensemble(S)\right);$
10 While $(\mathrm{S} \neq \phi)$ do
11 $h = \arg \min_{hk\hat{I}s} f(S, d_k);$
12 $E_1 = Error\left(Ensemble\left(S - \{h\}\right)\right);$
13 if $E_1 <= E_0$ then
14 $E_0 = E_1;\ S = S - \left\{h\right\};$
15 end if;
16 Return S
End;

3.5 Evaluation Measures

The evaluation measures can be categorized such as performance-based and diversity-based measure. The goal of performance-based measures is to find the model that optimizes the performance of the ensemble produced by adding (removing) a model to (from) the current ensemble. Performance-based measure includes the root-mean-squared-error RMSE is calculated in the forward selection for a model h with respect to the current sub-ensemble S and the set of examples D as follows:

$$RMSE_{FS}(S,h,D) = \sqrt{\frac{1}{N|S|}\sum_{i=1}^{T}\sum_{j=1}^{N}(h_i(x_i)-y_j)^2 + \frac{1}{N}\sum_{j=1}^{N}(h(x_j)-y_i)^2} \tag{11}$$

The calculation of performance-based metrics requires the decision of the ensemble on all examples of the pruning dataset.

It is generally accepted that an ensemble should contain diverse models in order to achieve high predictive performance. In the experimental section we use the diversity-based measure (DIV) proposed in (Hernandez-Lobato, Martinez-Munoz, & Suarez, 2006). The DIV is calculated in the forward selection for model h with respect to the current sub-ensemble S and the set of examples D as follows:

$$DIV_{FS}(S,h,D) = \frac{1}{|S|}\left(\sum_{i=1}^{|S|}\sum_{j=1}^{|S|}C_{h_ih_j} + 2\sum_{i=1}^{|S|}C_{h_ih} + C_{hh}\right) \tag{12}$$

where $C_{h_ih_j}$ expresses the correlation between the two learners h_i and h_j. The value $C_{h_ih_j}$ is computed as follows:

$$C_{h_ih_j} = \frac{1}{N}\sum_{n=1}^{N}(h_i(x_n)-y_n)(h_j(x_n)-y_n) \tag{13}$$

4. EXPERIMENTAL SETUP

4.1 Description of Experimental Data

In this sub-section, the proposed GES algorithm evaluated based on the analysis of hyper-parameters setting and the influence of these hyper-parameters in SVR performance. Well-known datasets were chosen as a benchmark to assess the performance of the suggested model. The benchmark datasets downloaded from (www.kaggle.com) include US air passengers, Korean won against US dollar exchange rate, gasoline retail price in New York city, IBM stock time series, American electric power consumption and gold price, the descriptions of all datasets are outlined in Table 1, including the notation, names of the datasets, time stamp type, time period, total size, size of sample data used in the experiments for train and test.

4.2 Base Predictors

The only experimental study of ensemble selection algorithms considering homogeneous models by using Weka tool. The optimization of SVR hyper-parameters is performed by using GES approach in the training set data using 10-fold cross validation which gives the smallest value of RMSE error, we selected parameters range of the SVR ensemble model as in Table 2. The regularization constant (the complexity parameter C), ε parameters of the ε - insensitive loss function, kernel parameters of the degree d of the polynomial kernel, the width of RBF kernel γ, and PUK kernel parameters σ were optimized see Table 2.

Table 2. data sets description

Notation	Data set	Time	Period	Size	Train	Test
D1	Air Passengers	Monthly	1949 to 1960	144	101	43
D2	Exchange Rate	Daily	2002 to 2017	4584	3209	1375
D3	Gasoline Price	Weekly	2000 to 2012	901	631	270
D4	IBM Stock	Daily	2005 to 2017	3272	2290	982
D5	Electric Consumption	Hourly	2002 to 2018	143207	100245	42962
D6	Gold Price	Weekly	2013 to 2017	247	173	74

Table 3. Parameters range for the SVR-based all kernel model.

Parameter	Value
Kernel	Poly, PUK, RBF
Regularization constant (C)	1-1000
ε-insensitive (ε)	0.1: 10^{-9}
Kernel parameter (λ)	10E-5:10E6

In the next step, we use the greedy ensemble selection algorithm (GES) after setting the parameters of direction, evaluation dataset and evaluation measure. We experiment with the direction parameter using both forward (F) and backward (B) as values. For the evaluation dataset, we use both the training set (T) and a separate selection set (S) as the evaluation dataset, as explained in the previous paragraph. Concerning the evaluation measure, we use the following 2 measures: RMSE and DIV. Table 3 shows the acronyms for the different instantiations of the greedy ensemble selection algorithm.

Table 4. *Kernel, search direction, evaluation dataset, evaluation measure, and acronym for the different instantiations of the greedy ensemble selection algorithm*

Kernel	Direction	Dataset	Measure	Acronym
Poly	Backward	Selection	Diversity	POBSD
			RMSE	POBSR
		Training	Diversity	POBTD
			RMSE	POBTR
	Forward	Selection	Diversity	POFSD
			RMSE	POFSR
		Training	Diversity	POFTD
			RMSE	POFTR
RBF	Backward	Selection	Diversity	RBSD
			RMSE	RBSR
		Training	Diversity	RBTD
			RMSE	RBTR
	Forward	Selection	Diversity	RFSD
			RMSE	RFSR
		Training	Diversity	RFTD
			RMSE	RFTR
PUK	Backward	Selection	Diversity	PUBSD
			RMSE	PUBSR
		Training	Diversity	PUBTD
			RMSE	PUBTR
	Forward	Selection	Diversity	PUFSD
			RMSE	PUFSR
		Training	Diversity	PUFTD
			RMSE	PUFTR

5. RESULTS AND DISCUSSION

In this section we present and discuss results concerning homogeneous models are analyzed from the perspectives of predictive performance, final ensemble size and the relationship between them.

5.1 Predictive Performance

Table 4 and 5 presents the RMSE and the corresponding rank respectively for each algorithm on each dataset, as well as the average error and rank across all datasets. We start the performance analysis of the different algorithms based on their average rank across all datasets. We first notice, that the best performing algorithm is FSR, obtaining the best performance on all datasets, followed by FSD. Figure 3 presents aggregates of the mean ranks for the different values of the search direction (3a), evaluation dataset (3b) and evaluation measure (3c) parameters. Additionally, Figure 3d–f present aggregates for the different values of parameter pairs.

Table 5. Average errors for the different algorithms on each predicted data set

Base Learner	D1	D2	D3	D4	D5	D6	Avg. Error
Poly kernel algorithms							
POBSD	0.094	0.165	0.167	0.127	0.135	0.144	0.139
POBSR	0.096	0.093	0.112	0.102	0.148	0.128	0.113
POBTD	0.096	0.855	0.179	0.302	0.222	0.228	0.314
POBTR	0.089	0.100	0.119	0.281	0.130	0.149	0.145
POFSD	0.083	0.172	0.083	0.107	0.113	0.130	0.117
POFSR	0.067	0.091	0.101	0.101	0.119	0.112	**0.098**
POFTD	0.611	0.291	0.241	0.112	0.274	0.303	0.305
POFTR	0.222	0.222	0.212	0.136	0.248	0.293	0.222
RBF kernel algorithms							
RBSD	0.815	0.856	0.014	0.105	0.249	0.249	0.381
RBSR	0.087	0.091	0.094	0.099	0.101	0.103	0.096
RBTD	0.928	0.049	0.152	0.303	0.392	0.464	0.382
RBTR	0.099	0.103	0.114	0.116	0.125	0.134	0.115
RFSD	0.086	0.086	0.091	0.095	0.099	0.103	0.093
RFSR	0.075	0.076	0.077	0.076	0.078	0.076	**0.076**
RFTD	0.790	0.956	0.250	0.375	0.589	0.780	0.623
RFTR	0.509	0.555	0.616	0.625	0.677	0.688	0.612
PUK kernel algorithms							
PUBSD	0.658	0.732	0.853	0.011	0.094	0.191	0.423
PUBSR	0.079	0.081	0.085	0.091	0.093	0.094	0.087
PUBTD	0.646	0.429	0.235	0.424	0.652	0.814	0.533
PUBTR	0.150	0.151	0.168	0.172	0.180	0.192	0.169
PUFSD	0.077	0.077	0.083	0.081	0.082	0.082	0.080
PUFSR	0.069	0.072	0.075	0.079	0.083	0.085	**0.077**
PUFTD	0.610	0.642	0.806	0.740	0.822	0.851	0.745
PUFTR	0.326	0.818	0.537	0.686	0.808	0.963	0.690

Note: The bold number indicates the best accuracy value.

Based on Figure 3a we notice that the algorithms that search in the forward direction obtain slightly better mean rank (4.42) for Poly kernel, (4.3) for PUK kernel, and (4.46) for RBF kernel than those that search in the backward direction (4.5) for Poly kernel, (4.7) for PUK kernel, and (4.5) for RBF kernel. We therefore conclude that the search direction does not significantly affect the performance of the ensemble selection algorithms in this application.

In Figure 3c we observe a very interesting fact, as the mean rank of the algorithms that use the selection set (2.8) for Poly kernel, (3) for PUK kernel, (3) for RBF kernel for evaluation is considerably larger than the mean rank of those that use the training set (6.1, 6, 6) for Poly, PUK and RBF kernels respectively. This finding indicates a clear superiority of the xxxSx algorithms and leads to the conclusion that using a separate selection set improves the efficiency of the algorithms.

Table 6. Average ranks for the different algorithms on each predicted data set

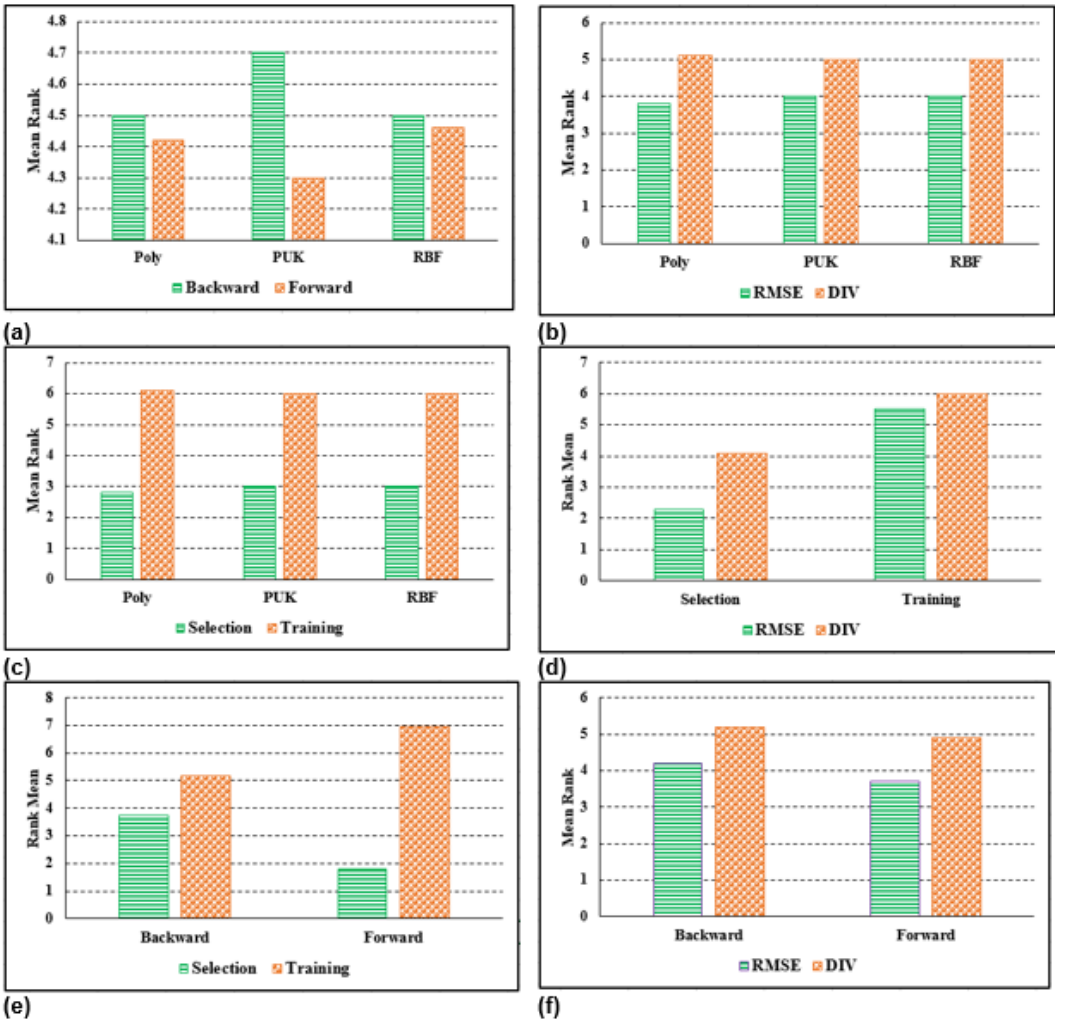
Base Learner	D1	D2	D3	D4	D5	D6	Avg. Rank
Poly kernel algorithms							
POBSD	4.0	4.0	5.0	5.0	4.0	4.0	4.3
POBSR	5.0	2.0	3.0	2.0	5.0	2.0	3.2
POBTD	5.0	8.0	6.0	8.0	6.0	6.0	6.5
POBTR	3.0	3.0	4.0	7.0	3.0	5.0	4.2
POFSD	2.0	5.0	1.0	3.0	1.0	3.0	2.5
POFSR	1.0	1.0	2.0	1.0	2.0	1.0	**1.3**
POFTD	8.0	7.0	8.0	4.0	8.0	8.0	7.2
POFTR	7.0	6.0	7.0	6.0	7.0	7.0	6.7
RBF kernel algorithms							
RBSD	7.0	7.0	1.0	4.0	5.0	5.0	4.8
RBSR	3.0	4.0	4.0	3.0	3.0	3.0	3.3
RBTD	8.0	1.0	6.0	6.0	6.0	6.0	5.5
RBTR	4.0	5.0	5.0	5.0	4.0	4.0	4.5
RFSD	2.0	3.0	3.0	2.0	2.0	2.0	2.3
RFSR	1.0	2.0	2.0	1.0	1.0	1.0	**1.3**
RFTD	6.0	8.0	7.0	7.0	7.0	8.0	7.2
RFTR	5.0	6.0	8.0	8.0	8.0	7.0	7.0
PUK kernel algorithms							
PUBSD	8.0	7.0	8.0	1.0	4.0	4.0	5.3
PUBSR	3.0	3.0	3.0	4.0	3.0	3.0	3.2
PUBTD	7.0	5.0	5.0	6.0	6.0	6.0	5.8
PUBTR	4.0	4.0	4.0	5.0	5.0	5.0	4.5
PUFSD	2.0	2.0	2.0	3.0	1.0	1.0	1.8
PUFSR	1.0	1.0	1.0	2.0	2.0	2.0	**1.5**
PUFTD	6.0	6.0	7.0	8.0	8.0	7.0	7.0
PUFTR	5.0	8.0	6.0	7.0	7.0	8.0	6.8

Note: The bold number indicates the best rank value.

Algorithms that use the training set for evaluation run the risk of overfitting which leads to low performance. On the other hand, the algorithms that use a separate selection set have better generalization performance as they are more robust to unseen data and resilient to noise. This behavior is also noticed in Figure 3e where the xxBTx algorithms have mean rank 5.17 the xxBSx algorithms 3.75, and the xxFTx, xxFSx 6.97 and 1.81 correspondingly.

Concerning the evaluation measures, the mean ranks of the algorithms are 3.8 for RMSE and 5.13 for DIV in case of Poly kernel, while are 4 for RMSE and 5 for DIV in case of PUK and RBF kernels. We notice that RMSE obtain the best performance despite its simplicity. For the DIV measure we can conclude that it does not succeed to select learners with a high diversity degree. The strength of the RMSE measure can be verified if we compare the ranks of the pairs of algorithms that use the same value for the direction and evaluation parameters, and different value for the evaluation measure.

Figure 3. Mean rank across all datasets for different values of parameters and parameter pairs



5.2 Statistical Analysis

In order to statistically evaluate differences between base learners, Friedman rank test (Cavalcanti et al., 2016) at significance level α=0.05, and critical difference (CD), CD = 2.90. The CD is the minimum required difference of the average ranks of two algorithms, so that their difference can be deemed significant. The best ranks are to the right and the groups of algorithms that are not significantly different are connected with a bold line.

Figure 4 graphically represents the results of the test with Poly kernel, the obtained output as one can see, POFSR (which are gathered near rank 1.) outperform all base learners. POFTD was confirmed to be the most and least accurate learner. We see there is a significant difference between POFSR and POBTD (p-value = 0.000725), POFTR (p-value = 0.001442), and POFTD (p-value = 0.000217). Similarly, POFSD and POBTD (p-value = 0.028489), POFTR (p-value = 0.046961), and POFTD (p-value = 0.011498) are significantly different, POBSR and POFTD (p-value = 0.039906) are significantly different.

Figure 4. Critical difference (CD) diagram of the post-hoc Nemenyi test (α = 0.05) between Poly kernel algorithms

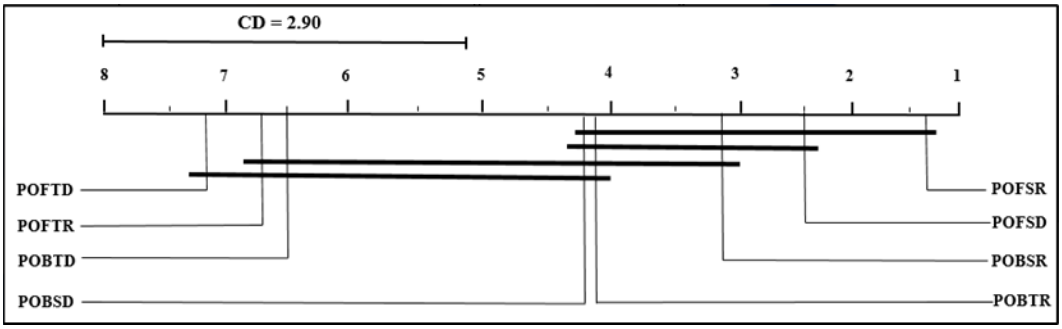


Figure 5 graphically represents the results of the test with PUK kernel, the obtained output as one can see, PUFSR (which are gathered near rank 1.) outperform all base learners. PUFTD was confirmed to be the most and least accurate learner. We see there is a significant difference between PUFSR and PUBTD (p-value = 0.045718), PUFTR (p-value = 0.004136), and PUFTD (p-value = 0.002617). Similarly, POFSD and PUFTR (p-value = 0.009848), and PUFTD (p-value = 0.006433) are significantly different.

Figure 5. Critical difference (CD) diagram of the post-hoc Nemenyi test (α = 0.05) between PUK kernel algorithms

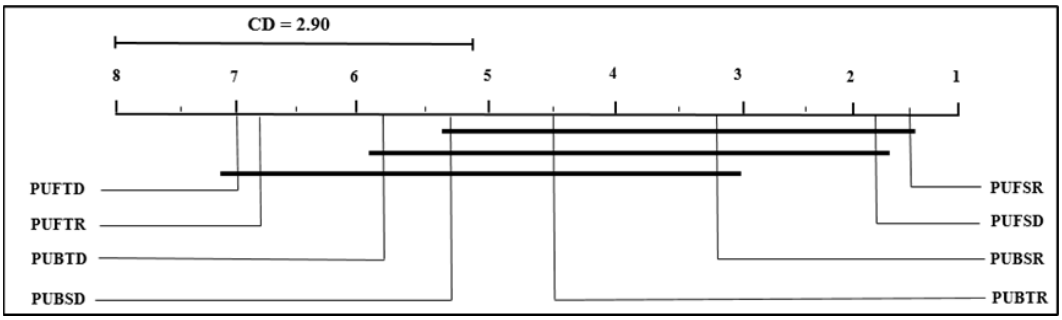


Figure 6. Critical difference (CD) diagram of the post-hoc Nemenyi test (α = 0.05) between Poly kernel algorithms

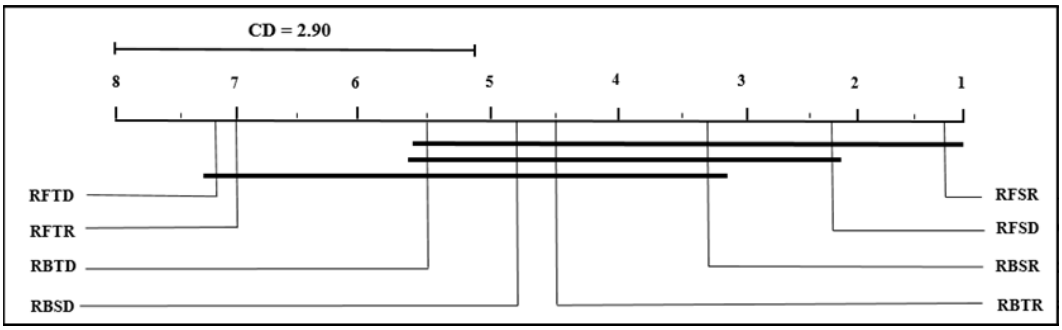


Figure 6 graphically represents the results of the test with RBF kernel, the obtained output as one can see, RFSR (which are gathered near rank 1.) outperform all base learners. RFTD was confirmed to be the most and least accurate learner. We see there is a significant difference between RFSR and RFTR (p-value = 0.001631), and RFTD (p-value = 0.001001). Similarly, RFSD and RFTR (p-value = 0.02196), and RFTD (p-value = 0.01483) are significantly different.

5.3 Ensemble Size

Table 6 shows the average size of the selected ensembles for each algorithm on each predicted variable. Figure 7 presents aggregates of the mean size of the selected ensemble for the different values for the search (7a), evaluation dataset (7b) parameters, evaluation measures (7c) as well as for pairs of values of the direction and evaluation dataset parameters (7d).

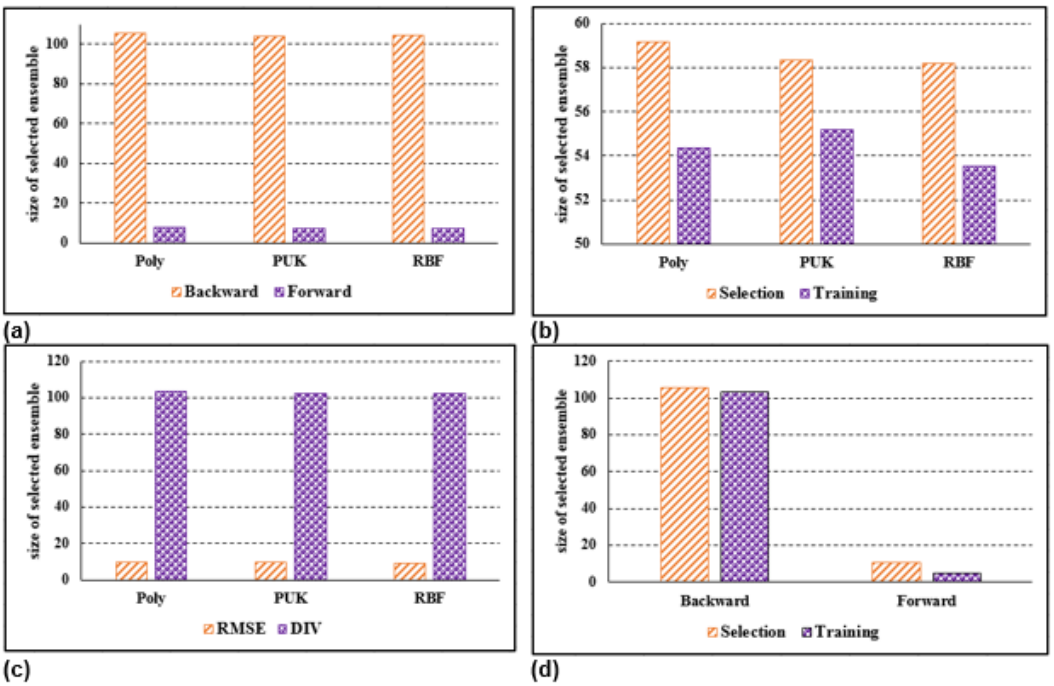
Table 6. Average size of the selected ensembles for the different algorithms on each predicted data set

Base Learner	D1	D2	D3	D4	D5	D6	Avg. Size
Poly kernel algorithms							
POBSD	199.0	199.0	199.0	199.0	199.0	199.0	199.0
POBSR	16.2	11.5	17.7	14.5	15.5	17.5	15.5
POBTD	199	193.1	199	199	199	199	198.0
POBTR	2.4	12.4	6.2	11.9	10.9	12.9	**9.5**
POFSD	7.2	8.9	15.2	20.5	11.5	12.5	12.6
POFSR	4.9	10.9	12.5	11.1	9.1	8.2	**9.5**
POFTD	2.4	4.9	4.7	4.4	5.6	6.6	4.8
POFTR	2.7	5.4	6.5	4.1	6.4	5.9	5.2
RBF kernel algorithms							
RBSD	199.0	196.0	199.0	194.0	199.0	195.0	197.0
RBSR	17.2	10.5	17.5	13.4	15.2	16.5	15.05
RBTD	199.0	193.1	192.0	199.0	194.0	199.0	196.0
RBTR	2.4	12.4	6.2	11.9	8.9	12.9	**9.1**
RFSD	7.2	8.9	16.0	18.4	10.5	12.3	12.2
RFSR	4.9	10.9	12.5	11.1	7.1	8.2	**9.1**
RFTD	2.6	4.9	4.7	4.4	5.5	3.6	4.3
RFTR	2.4	5.1	4.5	4.1	6.4	5.3	4.6
PUK kernel algorithms							
PUBSD	198.0	196.0	194.0	195.0	199.0	193.0	195.8
PUBSR	16.2	11.5	17.7	14.5	15.5	17.5	15.5
PUBTD	197	193.1	199	194	199	195	196.2
PUBTR	2.4	12.4	6.2	11.9	8.9	12.9	**9.1**
PUFSD	7.2	8.9	16	20.5	8.5	12.3	12.2
PUFSR	4.9	8.9	12.5	11.1	9.1	8.2	**9.1**
PUFTD	2.4	5.9	3.7	4.5	5.3	4.6	4.4
PUFTR	3.4	5.6	3.5	4.2	4.4	5.8	4.5

A remarkable observation in Figure 5a is that the algorithms that search in the backward direction produce larger ensembles (105.49) than those that search in the forward direction (8.0) in case of Poly kernel, (104.15) than those that search in the forward direction (7.55) in case of PUK kernel, and (104.30) than those that search in the forward direction (7.56) in case of RBF kernel. Based on the previous finding, that the direction parameter does not affect significantly the performance of the greedy ensemble selection algorithm, we conclude that the advisable direction for an ensemble selection algorithm is the forward direction.

In Figure 7b, we notice that the average size of the selected ensembles for the xxxSx (59.14) algorithms is slightly larger than the xxxTx (54.35) algorithms in case of Poly kernel, (58.35) algorithms is slightly larger than the xxxTx (55.2) algorithms in case of PUK kernel, and (58.17) algorithms is slightly larger than the xxxTx (53.55) algorithms in case of RBF kernel. This observation is also verified if we look at Figure 7d. We can assume that the xxxTx algorithms contain stronger learners than the xxxSx algorithms and select fewer learners in order to achieve the maximum performance. But the performance of the xxxTx algorithm is worse than the performance of the xxxSx algorithms which means that those strong models are over trained.

Figure 7. Mean size of selected ensemble based on (a) search method, (b) evaluation dataset parameters, (c) evaluation measures (d) pairs of values of the direction and evaluation dataset parameters



5.4 Predictive Performance vs. Ensemble Size

Figure 8a and b present the RMSE curve both on the train and the test set during the ensemble selection for one data set (D1). Firstly, in Figure 8a and b we notice that the ensemble selection procedure improves the RMSE using a small number of learners. Note that the final sub-ensemble that is selected, is the one that corresponds to the minimum of the evaluation set RMSE curve. In the figures we observe that this minimum point corresponds to a near optimal point in the test set RMSE

curve. This observation shows that the greedy ensemble selection algorithm manages this way to select an appropriate size for the ultimate sub-ensemble, which allows it to achieve high predictive performance.

In Figure 9a and b we notice that the FSD and BSD algorithms respectively, fail to select a good sub-ensemble. More specifically, in the case of FSD the DIV measure guides ineffectively the algorithm, as at the first steps it inserts for learners that have bad performance. The BSD algorithm seems to remove continually the superior learners from the ensemble, leading it to increase the error.

Figure 8. Predictive performance of FSR and BSR against ensemble size

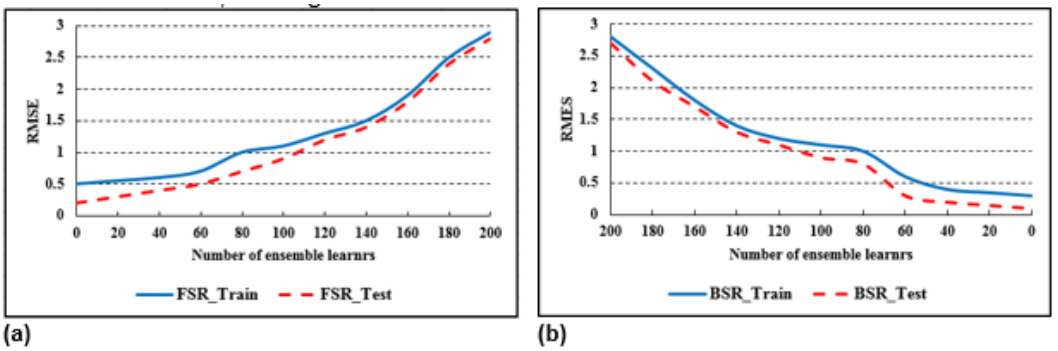
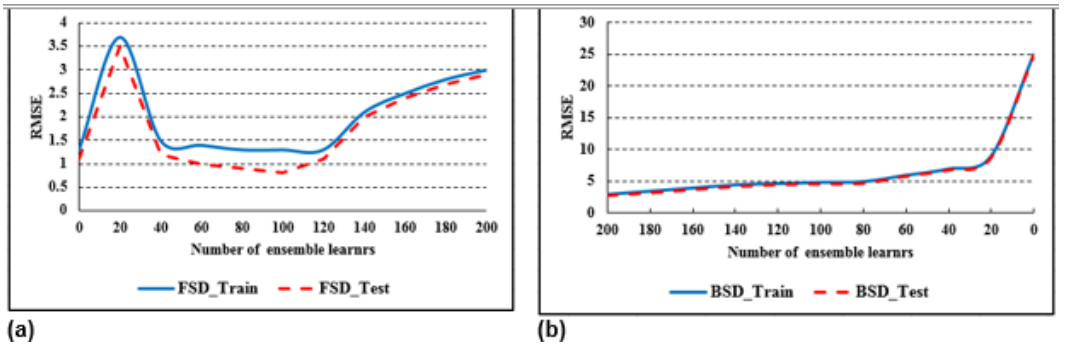


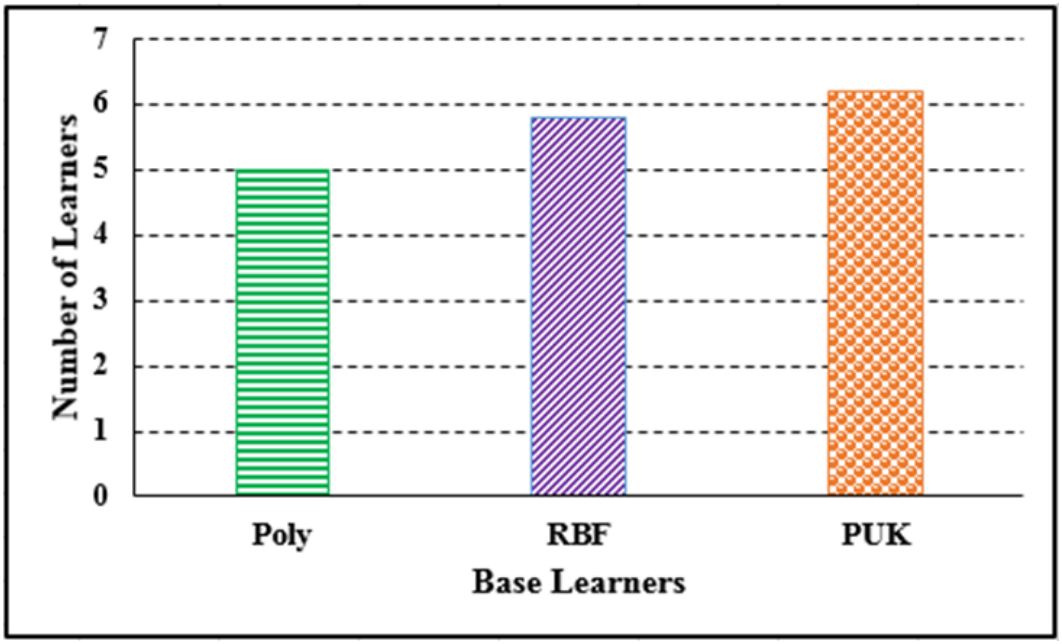
Figure 9. Predictive performance of FSD and BSD against ensemble size



5.5 Type of Models

Figure 10 presents aggregates concerning the type of models that are selected across all the predicted data sets. We focus on the results of the four best performing algorithms (FSR, FSD, BSR, BTR). The algorithms select equal sizes of both SVR-based Poly learner (5.0), SVR-based PUK learner (5.8) and SVR-based RBF learner (6.2).

Figure 10. Selected models by the best performing algorithms (FSR, FSD, BSR and BTR)



6. CONCLUSION

In this study, we presented a novel framework for a hybrid layered based greedy ensemble reduction (HLDER) approach is proposed for improving the prediction accuracy and generalization of the ensemble. We decomposed HLDER algorithm into different layers and we emphasized the various options for pruning layer. Additionally, we applied the framework of real data concerning time series data, and experimented with an ensemble of 200 learners consisting of SVR based (Poly, PUK and RBF) kernels. The results have shown that using a separate unseen set for the evaluation, leads the algorithm to increase its performance. Also, the algorithm is able to select an appropriate size for the final selected ensemble achieving a near-optimal performance. In this way there is no necessity to predefine the percentage of the models that must be pruned from the initial ensemble. Finally, as far as the direction parameter is concerned, we concluded that it does not affect importantly the performance of the HLDER algorithm, in both cases, the direction parameter does not affect significantly the performance and based on this conclusion we suggest the use of the forward direction as it produces smaller ensembles than the backward direction. Concerning the size of the ensemble that is selected, we concluded that selecting it based on the maximum accuracy on the evaluation set, leads to small ensembles with superior predictive performance.

REFERENCES

Abdesslem, L., Marwa, S., & Maroua Bencheikh, E. (2013). A new greedy randomised adaptive search procedure for Multiple Sequence Alignment. *International Journal of Bioinformatics Research and Applications*, *9*(4), 323–335. doi:10.1504/IJBRA.2013.054695 PMID:23797992

Banfield, R. E., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2005). Ensemble diversity measures and their application to thinning. *Information Fusion*, *6*(1), 49–62. doi:10.1016/j.inffus.2004.04.005

Baron, G. (2019). *Greedy Selection of Attributes to Be Discretised.* Academic Press.

Caruana, R., Munson, A., & Niculescu-Mizil, A. (2006). *Getting the Most Out of Ensemble Selection.* Paper presented at the International Conference on Data Mining. doi:10.1109/ICDM.2006.76

Cavalcanti, G. D. C., Oliveira, L. S., Moura, T. J. M., & Carvalho, G. V. (2016). Combining diversity measures for ensemble pruning. *Pattern Recognition Letters*, *74*, 38–45. doi:10.1016/j.patrec.2016.01.029

Dai, Q. (2013a). A competitive ensemble pruning approach based on cross-validation technique. *Knowledge-Based Systems*, *37*(2), 394–414. doi:10.1016/j.knosys.2012.08.024

Dai, Q. (2013b). A novel ensemble pruning algorithm based on randomized greedy selective strategy and ballot. *Neurocomputing*, *122*(122), 258–265. doi:10.1016/j.neucom.2013.06.026

Dai, Q., & Han, X. (2016). An efficient ordering-based ensemble pruning algorithm via dynamic programming. *Applied Intelligence*, *44*(4), 816–830. doi:10.1007/s10489-015-0729-z

Dai, Q., & Li, M. (2015). Introducing randomness into greedy ensemble pruning algorithms. *Applied Intelligence*, *42*(3), 406–429. doi:10.1007/s10489-014-0605-2

Dai, Q., & Liu, Z. (2013). ModEnPBT: A Modified Backtracking Ensemble Pruning algorithm. *Applied Soft Computing*, *13*(11), 4292–4302. doi:10.1016/j.asoc.2013.06.023

Dias, K., & Windeatt, T. (2014). *Dynamic Ensemble Selection and Instantaneous Pruning for Regression Used in Signal Calibration*. Paper presented at the International Conference on Artificial Neural Networks. doi:10.1007/978-3-319-11179-7_60

Fan, Y., Tao, L., Zhou, Q., & Han, X. (2017). Cluster Ensemble Selection with Constraints. *Neurocomputing*, *235*, 59–70. doi:10.1016/j.neucom.2017.01.001

Gang, Z., Zhang, S., Jian, Y., & Cheng, L. (2011). *Regularization based ordering for ensemble pruning.* Paper presented at the Eighth International Conference on Fuzzy Systems & Knowledge Discovery.

Gevezes, T., & Pitsoulis, L. (2015). A greedy randomized adaptive search procedure with path relinking for the shortest superstring problem. *Journal of Combinatorial Optimization*, *29*(4), 859–883. doi:10.1007/s10878-013-9622-z

Gonzalo, M. M. O., Daniel, H. L., & Alberto, S. (2009). An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 245–259. doi:10.1109/TPAMI.2008.78 PMID:19110491

Guo, H., Liu, H., Li, R., Wu, C., Guo, Y., & Xu, M. (2017). Margin & diversity based ordering ensemble pruning. *Neurocomputing*, 275.

Guo, L., & Boukir, S. (2013). Margin-based ordered aggregation for ensemble pruning. *Pattern Recognition Letters*, *34*(6), 603–609. doi:10.1016/j.patrec.2013.01.003

Hernandez-Lobato, D., Martinez-Munoz, G., & Suarez, A. (2006). *Pruning in Ordered Regression Bagging Ensembles.* Paper presented at the International Joint Conference on Neural Networks.

Hernández-Lobato, D., Martínez-Muñoz, G., & Suárez, A. (2011). Empirical analysis and evaluation of approximate techniques for pruning regression bagging ensembles. *Neurocomputing*, *74*(12), 2250–2264. doi:10.1016/j.neucom.2011.03.001

Idris, A., Khan, A., & Lee, Y. S. (2013). Intelligent churn prediction in telecom: Employing mRMR feature selection and RotBoost based ensemble classification. *Applied Intelligence*, *39*(3), 659–672. doi:10.1007/s10489-013-0440-x

Jiang, Z., Liu, H., Fu, B., & Wu, Z. (2017). *A Novel Bayesian Ensemble Pruning Method*. Paper presented at the IEEE International Conference on Data Mining Workshops.

Lessmann, S., Caserta, M., & Arango, I. M. (2011). Tuning metaheuristics: A data mining based approach for particle swarm optimization. *Expert Systems with Applications*, *38*(10), 12826–12838. doi:10.1016/j.eswa.2011.04.075

Li, N., Yu, Y., & Zhou, Z. H. (n.d.). Diversity regularized ensemble pruning. In *Machine Learning and Knowledge Discovery in Databases*. Springer. doi:10.1007/978-3-642-33460-3_27

Li, N., Yu, Y., & Zhou, Z. H. (2012). *Diversity Regularized Ensemble Pruning*. Springer Berlin Heidelberg.

Liu, Z., Dai, Q., & Liu, N. (2014). Ensemble selection by GRASP. *Applied Intelligence*, *41*(1), 128–144. doi:10.1007/s10489-013-0510-0

Lu, Z., Wu, X., Zhu, X., & Bongard, J. (2010). *Ensemble pruning via individual contribution ordering.* Paper presented at the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. doi:10.1145/1835804.1835914

Martínez-Muñoz, G., Hernández-Lobato, D., & Suárez, A. (2008). An Analysis of Ensemble Pruning Techniques Based on Ordered Aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 245–259. doi:10.1109/TPAMI.2008.78 PMID:19110491

Martínez-Muñoz, G., & Suárez, A. (2004). Aggregation ordering in bagging. *Munoz*, 258-263.

Maskouni, M. A., Hosseini, S., Abachi, H. M., Kangavari, M., & Zhou, X. (2018). *Auto-CES: An Automatic Pruning Method Through Clustering Ensemble Selection*. Paper presented at the Australasian Database Conference. doi:10.1007/978-3-319-92013-9_22

Nadig, K., Potter, W., Hoogenboom, G., & Mcclendon, R. (2013). Comparison of individual and combined ANN models for prediction of air and dew point temperature. *Applied Intelligence*, *39*(2), 354–366. doi:10.1007/s10489-012-0417-1

Partalas, I., Tsoumakas, G., & Vlahavas, I. (2010). An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. *Machine Learning*, *81*(3), 257–282. doi:10.1007/s10994-010-5172-0

Partalas, I., Tsoumakas, G., & Vlahavas, I. (2012). *A Study on Greedy Algorithms for Ensemble Pruning*. Academic Press.

Partalas, I., Tsoumakas, G., & Vlahavas, I. P. (2008). *Focused Ensemble Selection: A Diversity-Based Method for Greedy Ensemble Selection.* Paper presented at the Conference on Ecai: European Conference on Artificial Intelligence.

Pérez-Gállego, P., Castaño, A., Quevedo, J. R., & Coz, J. J. D. (2018). Dynamic Ensemble Selection for Quantification Tasks. *Information Fusion*, 45.

Polikar, R. (2009). Ensemble learning. *Scholarpedia*, *4*(1), 1–34. doi:10.4249/scholarpedia.2776

Soto, V., García-Moratilla, S., Martínez-Muñoz, G., Hernández-Lobato, D., & Suárez, A. (2017). A Double Pruning Scheme for Boosting Ensembles. *IEEE Transactions on Cybernetics*, *44*(12), 2682–2695. doi:10.1109/TCYB.2014.2313638 PMID:24802114

Sun, P., & Lee, S. R. (2014). *Red tides prediction system using fuzzy reasoning and the ensemble method*. Kluwer Academic Publishers.

Sun, Q., & Pfahringer, B. (2012). *Bagging Ensemble Selection for Regression*. doi:10.1007/978-3-642-35101-3_59

Tamon, C., & Xiang, J. (2000). *On the Boosting Pruning Problem*. Paper presented at the European Conference on Machine Learning.

Tang, E. K., Suganthan, P. N., & Yao, X. (2006). *An analysis of diversity measures*. Kluwer Academic Publishers. doi:10.1007/s10994-006-9449-2

Tsoumakas, G., Partalas, I., & Vlahavas, I. (2009). An Ensemble Pruning Primer. *Studies in Computational Intelligence*, *245*, 1–13.

Zhang, C. X., & Zhang, J. S. (2011). A Survey of Selective Ensemble Learning Algorithms. *Chinese Journal of Computers*, *34*(8), 1399–1410. doi:10.3724/SP.J.1016.2011.01399

Zhang, Y., Burer, S., & Street, W. N. (2006). Ensemble Pruning Via Semi-definite Programming. *Journal of Machine Learning Research*, *7*(3), 1315–1338.

Mergani A. E. Khairalla received his doctorate degree from the school of computer science and technology, Wuhan University of Technology, China in 2019. He is currently an Assistant Professor at Faculty of Computer Science and Information Technology of Nile Valley University, Sudan. His main research areas are data mining, time series data, text mining, machine learning, and deep learning.