Greedy Regression Ensemble Selection: Theory and an Application to Water Quality Prediction

Ioannis Partalas [a,*], Grigorios Tsoumakas [a],
Evaggelos V. Hatzikos [b], Ioannis Vlahavas [a]

[a] *Department of Informatics, Aristotle University of Thessaloniki 54124 Thessaloniki, Greece*

[b] *Department of Automation, Tech. Educ. Institute of Thessaloniki*

Abstract

This paper studies the greedy ensemble selection family of algorithms for ensembles of regression models. These algorithms search for the globally best subset of regressors by making local greedy decisions for changing the current subset. We abstract the key points of the greedy ensemble selection algorithms and present a general framework, which is applied to an application domain with important social and commercial value: water quality prediction.

1 Introduction

Ensemble methods [9] has been a very popular research topic during the last decade. It has attracted scientists from several fields including Statistics, Machine Learning, Neural Networks, Pattern Recognition and Knowledge Discovery in Databases. Their success largely arises from the fact that they lead to improved accuracy compared to a single classification or regression model.

Typically, ensemble methods comprise two phases: a) the production of multiple predictive models, and b) their combination. Recent work, has considered an additional intermediate phase that deals with the reduction of the ensemble

* Corresponding author.
Email addresses: partalas@csd.auth.gr (Ioannis Partalas), greg@csd.auth.gr (Grigorios Tsoumakas), hatzikos@teithe.gr (Evaggelos V. Hatzikos), vlahavas@csd.auth.gr (Ioannis Vlahavas).

size prior to combination [23,15,35,29,30,6,1,26,20,34]. This phase is commonly named *ensemble pruning* or *ensemble selection*.

This paper studies the greedy ensemble selection family of algorithms for ensembles of regression models. These algorithms search for the globally best subset of regressors by making local greedy decisions for changing the current subset. We discuss three interesting parameters of these algorithms: a) the direction of search (forward, backward), b) the performance evaluation dataset (training set, validation set) and c) the performance evaluation measure. This way we offer a generalized and comparative perspective on greedy regression ensemble selection that goes beyond the presentation of a single algorithm that instantiates the above parameters.

We consider the application domain of water quality prediction, which is concerned with the construction of models that can provide early warnings of upcoming deterioration of water quality in aquatic systems. Water is a vital substance for most living organisms, including humans. Monitoring the level of pollution in the water is therefore a crucial task for public authorities. In addition, water quality prediction is also important in commercial activities such as irrigation, piscicultures and food industry. In this work, we experiment on real data collected from an underwater sensor system.

Experimental comparison of the different parameters are performed using a large ensemble (200 models) of neural networks (NNs) and support vector machines (SVMs). Results show that using a separate validation set for selection and a balanced mixture of NNs and SVMs leads to successful prediction of water quality variables.

In summary, the main contributions of this paper are the following:

- It describes the main components of greedy ensemble selection algorithms for ensembles of regression models and provides a guideline for building such algorithms.
- It performs a systematic experimental study of the different parameters of these algorithms on real-world data of, and delivers several interesting conclusions.

This paper is an extension of our previous work [25]. We extended the framework of greedy regression ensemble selection algorithms and conducted broader experiments taking into account the parameters of performance evaluation dataset and performance evaluation measure. In addition, this paper presents the experimental results in a methodological way.

The rest of this paper is structured as follows. Section 2 presents related work on ensemble selection in regression problems and on water quality prediction. In Section 3 we describe the data collection and pre-processing steps and in

Section 4 we present the framework of the general greedy ensemble selection algorithm. In Section 5 we describe the experimental setup and in Section 6 we discuss the results. Finally, Section 7 concludes this work and provides directions for future work.

2 Related Work

This section reviews related work on ensemble selection in regression problems, as well as on water quality prediction.

2.1 Ensemble Selection in Regression

Zhou et al. [35] presented an approach based on a genetic algorithm. More specifically, the genetic algorithm evolves a population of weight vectors for the regressors in the ensemble in order to minimize a function of the generalization error. When the algorithm outputs the best evolved weight vector, the models of the ensemble that did not exhibit a predefined threshold are dropped. The authors compared their approach only with ensemble methods and not with ensemble selection methods.

Rooney et al. [29] extended the technique of Stacked Regression to prune an ensemble of regressors using a measure that combines both accuracy and diversity. More specifically, the diversity is based on measuring the positive correlation of regressors in their prediction errors. The authors experimented with small sized ensembles (25 regressors) and they fixed the size of the final pruned to 5 regressors. A drawback of the proposed approach is that the user must define a weighting parameter to balance accuracy and diversity.

Hernandez et al. [20] introduced a greedy algorithm, where each regressor is ordered according to its complementariness, which is measured in terms of biases among the regressors. The algorithm selects a percentage (20%) from the ordered ensemble that consist the final selected ensemble.

Liu and Yao [22] proposed an approach named negative correlation learning, where a collection of neural networks are constructed by incorporating a penalty term to the training procedure. In this way, the models produced, tend to be negatively correlated. The experiments that carried out included small sized ensembles (less than 10 regressors).

Finally, Brown et al. [4] proposed a framework for managing the diversity in regression ensembles. Through the decomposition of bias-variance-covariance, the diversity is explicitly quantified and measured. This work showed that there is a relationship between the error function and the negative correlation algorithm [22]. Another interesting conclusion was that the negative correlation algorithm can be viewed as a framework for application on regression ensembles.

2.2 Water Quality Prediction

Reckhow [27] studied Bayesian probability network models for guiding decision making for water quality of Neuse River in North Carolina. The author focuses both on the accuracy of the model and the correct characterization of the processes, although these two features are usually in conflict with each other.

Blockeel et al [3] studied two problems. The first one concerned the simultaneous prediction of multiple physico-chemical properties of river water from its current biological properties using a single decision tree. This approach is opposed to learning a different tree for each different property and is called predictive clustering. The second problem concerned the prediction of past physico-chemical properties of the water from its current biological properties. The Inductive Logic Programming system TILDE [2] was used for dealing with the above problems.

Dzeroski et al. [10] addressed the problem of inferring chemical parameters of river water quality from biological ones, an important task for enabling selective chemical monitoring of river water quality. They used regression trees with biological and chemical data for predicting water quality of Slovenian rivers.

Lehmann and Rode [21] investigated the changes in metabolism and water quality in the Elbe river at Magdeburg in Germany since the German reunification in 1990. They used weekly data samples collected between the years 1984 and 1996. They used univariate time series models such as autoregressive component models and ARIMA models that revealed the improvement of water quality due to the reduction of waste water emissions since 1990. These models were used to determine the long-term and seasonal behaviour of important water quality parameters.

Romero and Shan [28] developed a neural network based software tool for prediction of the canal water discharge temperature at a coal-fired power plant. The variables considered in this system involve plant operating parameters and local weather conditions, including tide information. The system helps for the optimization of load generation among power plant generation units according to an environmentally regulated canal water discharge temperature limit of 95 Fahrenheit degrees.

Chau [7] presented the application of a split-step particle swarm optimization (PSO) model for training perceptrons in order to predict real-time algal bloom dynamics in Tolo Harbour of Hong Kong. Experiments with different lead times and input variables have been conducted and the results have shown that the split-step PSO-based perceptron outperforms other commonly used optimization techniques in algal bloom prediction, in terms of convergence and accuracy.

The case-based reasoning system, presented in [12,13], copes with water pollution. It specializes in forecasting the red tide phenomenon in a complex and dynamic environment in an unsupervised way. Red tides are the name for the sea water discolorations caused by dense concentrations of microscopic sea plants, known as phytoplankton. The system is an autonomous Case-Based Reasoning (CBR) hybrid system that embeds various artificial intelligence tools, such as case-based reasoning, neural networks and fuzzy logic in order to achieve real time forecasting. It predicts the occurrence of red tides caused by the pseudo-nitzschia spp diatom dinoflagellate near the North West coast of the Iberian Peninsula. Its goal is to predict the pseudo-nitzschia spp concentration (cells/liter) one week in advance, based on the recorded measurements over the past two weeks. The developed prototype is able to produce a forecast with an acceptable degree of accuracy. The results obtained may be extrapolated to provide forecasts further ahead using the same technique, and it is believed that successful results may be obtained. However, the further ahead the forecast is made, the less accurate it may be.

Hatzikos et al. [18] utilized neural networks with active neurons as the modeling tool for the prediction of sea water quality. The proposed approach was concerned with predicting whether the value of each variable will move upwards or downwards in the following day. Experiments were focused on four quality indicators, namely water temperature, pH, amount of dissolved oxygen and turbidity.

3 Data Collection and Pre-Processing

This section describes the system that collected the data used in our study and the pre-processing approach that was followed.

3.1 The Andromeda analyzer

The data used in this study have been produced by the Andromeda analyzer [16,17]. The system is installed in Thermaikos Gulf of Thessaloniki, Greece and consists of three local measurement stations and one central data collection station.

The local measurement stations (see Figure 1) are situated in the sea and serve the purpose of data collection. Each of them consists of the following parts:

- A buoy.
- A number of sensors.
- A reprogrammable logic circuit.
- Strong radio modems.
- A tower of 6 meters height for the placement of an aerial.
- Solar collectors interconnected for more power.
- Rechargeable batteries.

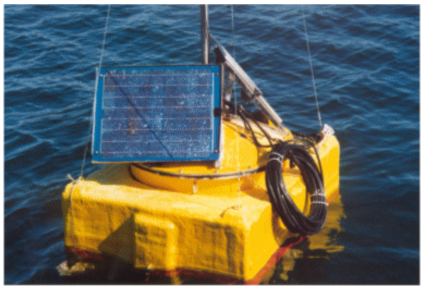


Fig. 1. One of the three local measurement stations of the Andromeda system.

The solar collectors and the batteries provide the electrical power needed by the sensors and electronics. The sensors measure water temperature, pH, conductivity, salinity, amount of dissolved oxygen and turbidity in sea-water at fixed time points. The reprogrammable logic circuit monitors the function of the local measurement station and stores the measurements in its memory. Moreover, it controls the communication via the wireless network and sends the measurements to the central data collection station.

The central data collection station monitors the communication with the local measurement stations and collects data from all of them. Data are stored in a database for the purpose of future processing and analysis. It consists of a Pentium computer operating in SCADA environment. The computer plays the role of *master* and controls the communication with the local measurement stations using the *hand-shake* protocol. The total number of measurements that are collected is between 8 and 24 daily. The frequency of measurements can be increased in case of emergency. This communication policy reduces the consumption of energy by the local stations, since they operate only when they have to send data to the central station.

The data that are studied in this paper were collected from April 14, 2003 until November 2, 2003 at an hourly basis with a sampling interval of 9 seconds. Given that the variation of the measurements from one hour to the next is typically very small, we decided to work on the coarser time scale of 24 hours, by averaging the measurements over days.

Two problems introduced by the data collection process are the following: a) there is a number of missing values due to temporary inefficiency of the sensors as well as problems in the transmission of the data, and b) the occurrence of special events near the local measurement stations, such as the crossing of a boat, have led to the recording of some outliers.

Fortunately, both of these temporary problems are automatically solved through the daily averaging process. During a day, the missing values are typically from 0 to 3, so the rest of the measurements can reliably give a mean estimate for the day. In addition, averaging ameliorates the effect of outliers. Specifically we calculate the median of all daily measurements, which trims away extreme values.

We completely removed measurements concerning dissolved oxygen and turbidity, as the corresponding sensors experienced long-term failures during the data collection period. The remaining 4 variables (temperature, pH, conductivity and salinity) were considered independently as target attributes in the regression modeling task. The input attributes correspond to values of previous days for all variables (including the target one).

Two parameters that are considered in time-series prediction tasks are the *window* or *time lag* and the *time lead* [32]. Window is the number of the preceding days that will be used for generating the prediction model. Lead is the number of the intermediate days between the last day used for generating the attributes and the day we are going to predict the target variable. Based on the findings of a previous study [19] we set the window to 9 and the lead to 2.

4 The Greedy Ensemble Selection Algorithm

The general greedy ensemble selection algorithm attempts to find the globally best subset of regressors by making local greedy decisions for changing the current subset. In the following subsections we present the main aspects of the greedy ensemble selection algorithm: the direction of search, the function

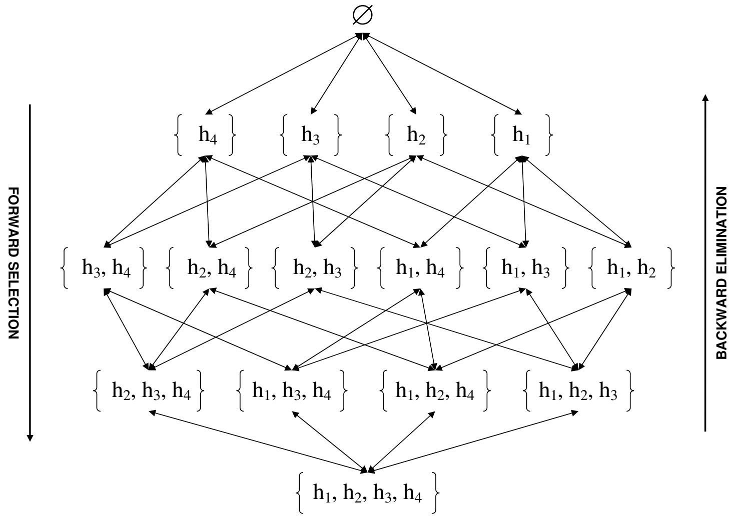


Fig. 2. An example of the search space of greedy ensemble selection algorithms.

that is used to evaluate the different branches of the search and the size of the selected subensemble.

Before abstracting the main characteristics of the greedy ensemble selection algorithm, we provide the notation that will be used in the rest of this paper. Let $D = \{(x_i, y_i), i = 1, 2, \dots, N\}$ denote a set of labeled training examples where each example consists of a feature vector x_i and the true value y_i. Also, let $H = \{h_t, t = 1, 2, \dots, T\}$ be the set of base regressors that constitute the ensemble. Each regressor h_t maps an input vector x to an output value y. Finally, we denote as $S \subset H$ the current subensemble during the search in the space of subensembles.

4.1 Direction of Search

Based on the direction of search there are two different groups of ensemble selection algorithms: *forward selection* and *backward elimination*. Figure 2 shows an example of the search space of greedy ensemble selection algorithms along with the two search directions.

In forward selection, the current subset S is initialized to the empty set, and the algorithm at each step appends to S the regressor $h \in H \setminus S$ that optimizes an evaluation function $f_{FS}(S, h, D)$. This function evaluates the addition of h to S based on the labeled data D. For example, it could return the root-mean-squared error of the subensemble $S \cup \{h\}$ on the data set D by simply averaging the decisions of the regressors. Table 1 shows the forward ensemble selection algorithm in pseudocode.

In backward elimination, the current subset of regressors S is initialized to

```
Require: Ensemble of regressors $H$, evaluation function $f_{FS}$ and set $D$
$S = \emptyset$
while $S \neq H$ do
    $h_t = \underset{h \in H \setminus S}{\arg\max} f_{FS}(S, h, D)$
    $S = S \cup \{h_t\}$
end while
```

Table 1
The forward selection method in pseudocode

the complete ensemble H and the algorithm continues by iteratively removing from S the regressor $h \in S$ that optimizes the evaluation function $f_{BE}(S, h, D)$. This function evaluates the removal of regressor h from the current subset S based on the labeled data of D. For example, f_{BE} could return a measure of diversity for the ensemble $S \setminus \{h\}$, calculated based on the labeled data D. Table 2 shows the pseudocode of the backward elimination ensemble selection algorithm.

```
Require: Ensemble of regressors $H$, evaluation function $f_{BE}$ and set $D$
$S = H$
while $S \neq \emptyset$ do
    $h_t = \underset{h \in S}{\arg\max} f_{BE}(S, h, D)$
    $S = S \setminus \{h_t\}$
end while
```

Table 2
The backward elimination method in pseudocode

The complexity of greedy ensemble selection algorithms for traversing the space of subensembles is $O(T^2 g(T, N))$. The term $g(T, N)$ concerns the complexity of the evaluation function, which is discussed in the following subsections.

4.2 Evaluation Function

The evaluation function is one of the most important components in an ensemble selection algorithm. Its purpose is to evaluate the alternative branches during the search in the space of the subensembles, and affects the selection of the final subensemble and subsequently the performance of the whole algorithm. Given a subensemble S and a model h, the evaluation function estimates the utility of inserting (removing) h into (from) S using an appropriate *evaluation measure*, which is calculated on an *evaluation dataset.*

4.2.1 Evaluation Dataset

We can distinguish three approaches for the evaluation dataset. The first one is to use the training dataset for evaluation. This approach offers the advantage of using plenty of data for evaluation and training, but it is susceptible to the danger of overfitting. This approach was used in [20].

Another approach is to withhold a part of the training set for evaluation as in [6,1,25]. This approach is less prone to overfitting, but reduces the amount of data that are available for training and evaluation compared to the previous approach.

A third approach that has been used in [5], is based on $k-$fold cross$-$validation. For each fold an ensemble is created using the remaining folds as the training set. The same fold is used as the evaluation dataset for models and subensembles of this ensemble. Finally, the evaluations are averaged across all folds. This approach is less prone to overfitting as the evaluation of the models is based on the data that were not used for their training and at the same time, the complete training dataset is used for evaluation.

During testing the above approach works as follows: the k models that were trained using the same procedure (same algorithm, same subset, etc.) form a cross-validated model. When the cross-validated model makes a prediction for an instance, it averages the predictions of the individual models.

4.2.2 Evaluation Measure

The evaluation measures can be categorized into two different groups: *performance-based* and *diversity-based.*

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 metrics include mean-squared-error (MSE), root-mean-squared-error (RMSE), and correlation coefficient.

The RMSE (which is used in the experimental section) is calculated in forward selection for a model h with respect to the current subensemble S and the set of examples D as follows:

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

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

It is generally accepted that an ensemble should contain diverse models in order to achieve high predictive performance. In their study, Brown et al. [4], formulate diversity in terms of covariance between the regressors by decomposing the mean-squared-error (MSE). The diversity that optimizes the MSE is that which optimally balances the three components: bias-variance-covariance [31].

In the experimental section we use the diversity measure proposed by Hernandez-Lobato et al. [20], which we shall call Regression Diversity (RDIV) in the rest of the paper.

The RDIV is calculated in forward selection for a model h with respect to the current subensemble S and the set of examples D as follows:

$$RDIV_{FS}(S,h,D) = \frac{1}{|S|}\left(\sum_{i=1}^{|S|}\sum_{j=1}^{|S|} C_{h_i h_j} + 2\sum_{i=1}^{|S|} C_{h_i h} + C_{hh}\right)$$

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

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

The time-complexity of calculating the above measure is $O(|S|)$ as we can pre-calculate the matrix C which has a cost of $O(T^2 \cdot N)$.

4.3 Size of the Selected Ensemble

Another issue that concerns greedy ensemble selection algorithms, is when to stop the search process, or in other words how many models should the final ensemble include.

One solution is to perform the search until all models have been added into (removed from) the ensemble and select the subensemble with the lowest error on the evaluation set [6]. This approach has been used in [25]. Others prefer to select a predefined number of models, expressed as a percentage of the original ensemble [23,11,20,1].

5 Experimental Setup

We experimented with the four datasets that were produced from the preprocessing step described in Section 3.2. Each dataset deals with the prediction of a different water quality variable ($o1$: temperature, $o2$: pH, $o3$: conductivity, $o4$: salinity).

Initially, each dataset is split into three disjunctive parts: a training set D_{Tr}, a selection set D_S and a test set D_{Te}, consisting of 60%, 20% and 20% of the examples in this dataset respectively. Then an ensemble production method is used, in order to create an ensemble of 200 models. We experiment with heterogeneous models, where we run different learning algorithms with different parameter configurations.

The WEKA machine learning library is used as the source of learning algorithms [33]. We train 90 multilayer perceptrons (MLPs) and 110 support vector machines (SVMs). The reason for restricting our study to these two types of learning algorithms, is that they are known to produce among the most accurate models for regression tasks. The different parameters that are used to train these algorithms are the following (default values are used for the rest of the parameters):

- MLPs: we use 6 values for the nodes in the hidden layer {1, 2, 4, 6, 8, 16}, 5 values for the momentum term {0.0, 0.2, 0.5, 0.7, 0.9} and 3 values for the learning rate {0.4, 0.6, 0.9}.
- SVMs: we use 11 values for the complexity parameter {10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 100, 1000}, and 10 different kernels. We use 2 polynomial kernels (degree 2 and 3) and 8 radial kernels (gamma {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2}).

In order to evaluate the utility of using a separate selection dataset for the evaluation function, we train two different ensembles of models: In the first ensemble, D_{Tr} is used for training the models. D_S is then used for evaluation and D_{Te} for testing. In the second ensemble, the dataset $D_{Tr} \bigcup D_S$ is used for both training and evaluation. As in the previous case, D_{Te} is then used for testing. In this way, we make a fair comparison between using just the training set and using a separate dataset for evaluation.

In the next step, we use the greedy ensemble selection algorithm 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: root-mean-squared-error (`RMSE`) and regression diversity (`RDIV`).

Table 3 shows the acronyms for the different instantiations of the greedy ensemble selection algorithm. In order to fuse the estimates of the regressors for the calculation of RMSE, we use a simple linear function, which aggregates the estimates. The ensemble output for an instance x is the following:

$$h_S(x) = \frac{1}{|S|} \sum_{i=1}^{|S|} h_i(x). \tag{1}$$

The final selected subensemble is the one with the lowest RMSE on the evaluation set (using Equation 1 for model combination). We record the size of the resulting ensemble and its error on the test set, using Equation 1 for model combination. The whole experiment is performed 10 times and the results are averaged.

We also calculate the performance of the complete ensemble of 200 regressors (`ALL`) using Equation 1 for model combination and the best single model (`BSM`) based on the performance of the regressors on the evaluation dataset.

Table 3
Acronym, search direction, evaluation dataset and evaluation measure for the different instantiations of the greedy ensemble selection algorithm.

Acronym	Direction	Evaluation Dataset	Evaluation Measure
BSRDIV	Backward	Selection	Regression Diversity
BSRMSE	Backward	Selection	Root Mean Squared Error
BTRDIV	Backward	Training	Regression Diversity
BTRMSE	Backward	Training	Root Mean Squared Error
FSRDIV	Forward	Selection	Regression Diversity
FSRMSE	Forward	Selection	Root Mean Squared Error
FTRDIV	Forward	Training	Regression Diversity
FTRMSE	Forward	Training	Root Mean Squared Error

6 Results and Discussion

In this section we present and discuss the results from the perspectives of predictive performance, final ensemble size and the relationship between them. Note that we do not argue for general conclusions that can be generalized beyond the specific datasets of this water quality monitoring application. Yet, some of the issues discussed here should be useful to researchers and practitioners working on different applications as well.

Tables 4 and 5 present the root-mean-squared-error and the corresponding rank respectively for each algorithm on each dataset, as well as the average error and rank across all datasets.

Table 4
Average errors for the different algorithms on each predicted variable.

	o1	o2	o3	o4	Average Error
BSRDIV	2.044	0.308	2.906	2.889	2.037
BSRMSE	1.388	0.363	1.142	0.838	0.933
BTRDIV	1.956	2.018	2.927	2.562	2.366
BTRMSE	6.189	0.248	3.75	2.349	3.134
FSRDIV	1.267	0.335	1.198	0.796	0.899
FSRMSE	1.227	0.12	1.118	0.758	0.806
FTRDIV	6.189	0.266	4.009	3.022	3.371
FTRMSE	6.189	0.263	4.0	3.088	3.385

Table 5
Average ranks for the different algorithms on each predicted variable.

	o1	o2	o3	o4	Average Rank
BSRDIV	5.0	5.0	4.0	6.0	5.0
BSRMSE	3.0	7.0	2.0	3.0	3.75
BTRDIV	4.0	8.0	5.0	5.0	5.5
BTRMSE	7.0	2.0	6.0	4.0	4.75
FSRDIV	2.0	6.0	3.0	2.0	3.25
FSRMSE	1.0	1.0	1.0	1.0	1.0
FTRDIV	7.0	4.0	8.0	7.0	6.5
FTRMSE	7.0	3.0	7.0	8.0	6.25

Following the recommendation of Demsar [8], 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 `FSRMSE`, obtaining the best performance on all datasets, followed by `FSRDIV`.

Figure 3 presents aggregates of the mean ranks for the different values of the search direction 3(a), evaluation dataset 3(b) and evaluation measure 3(c) parameters. Additionally, Figures 3(d) to 3(f) present aggregates for the different values of parameter pairs.

Based on Figure 3(a) we notice that the algorithms that search in the forward direction obtain slightly better mean rank (4.25) than those that search in the backward direction (4.75). We therefore conclude that the search direction does not significantly affect the performance of the ensemble selection algorithms in this application.

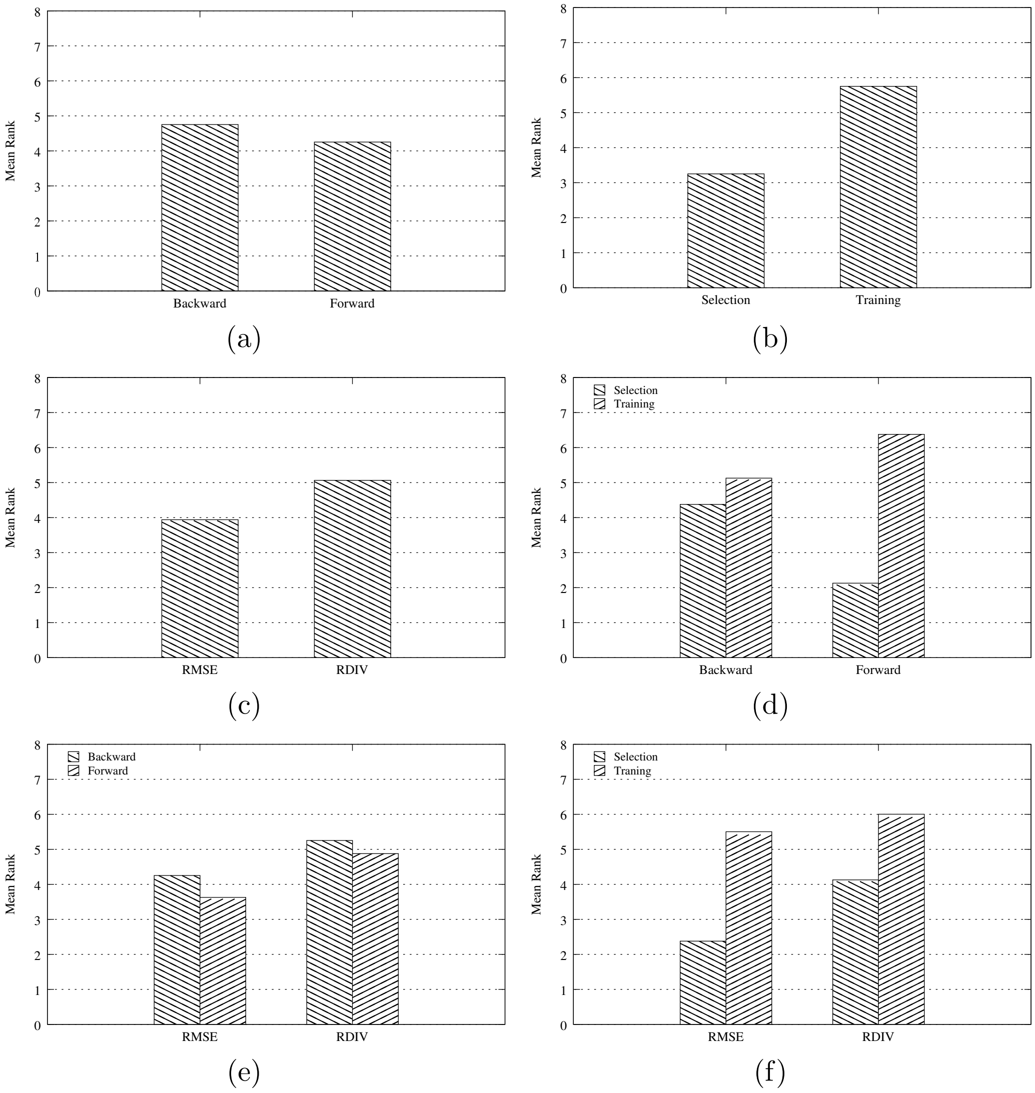
In Figure 3(b) we observe a very interesting fact, as the mean rank of the algorithms that use the selection set (3.25) for evaluation is considerably larger than the mean rank of those that use the training set (5.75). This finding indicate a clear superiority of the xSxxxx algorithms and leads to the conclusion that using a separate selection set improves the efficiency of the algorithms.

The algorithms that use the training test 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 3(d) where the BTxxxx algorithms have mean rank 5.125 the BSxxxx algorithms 4.375, and the FTxxxx, FSxxxx 6.375 and 2.125 correspondingly.

Concerning the evaluation measures, the mean ranks of the algorithms are 3.9375 for RMSE and 5.0625 for RDIV. We notice that RMSE obtains the best performance despite its simplicity. For the RDIV measure we can conclude that it doesn't manage to select regressors 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. RDIV has been successful in the past [20], but seems to have problems in the case of heterogeneous problems.

We next proceed in statistical tests in order to investigate whether there are significant differences among the performance of the algorithms. We initially employ Friedman's test [14], which shows critical differences among the algorithms. Following the recommendation of Demsar [8], we proceed in the post-hoc Nemenyi test [24]. Figure 4 graphically represents the results of the test with 90% confidence, $q_{0.10} = 2.78$ and critical difference, $CD = 4.815$. 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. We notice that there are two groups of similar algorithms. The statistically significant differences are those of FSRMSE over FTRDIV and FTRMSE.

We also calculate the performance of the simpler ensemble methods ALL and BSM. Table 6 shows the average errors of these methods on each target variable. We notice that fusing all models does not exhibit good performance, since the



(a) (b) (c) (d) (e) (f)

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

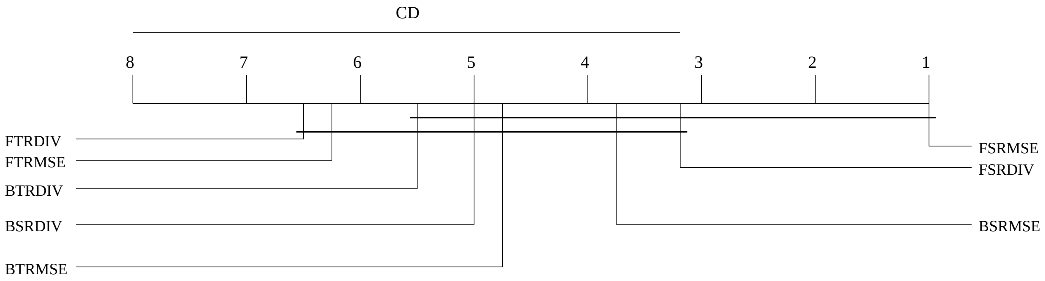


Fig. 4. Graphical representation of the Nemenyi test for all the algorithms.

ensemble is composed of both well and bad performing models. Using just the best single model on the other hand performs quite well and outperforms most of the ensemble selection methods apart from (`FSRMSE`). This shows that using the best model should be used as a strong baseline method in comparative experiments involving ensemble selection.

Table 6
Average errors for single best and all regressors on each predicted variable.

	o1	o2	o3	o4	Average Error
ALL	2.037	0.307	2.894	2.655	1.973
BSM	1.336	0.138	1.312	0.810	0.899

6.2 Ensemble Size

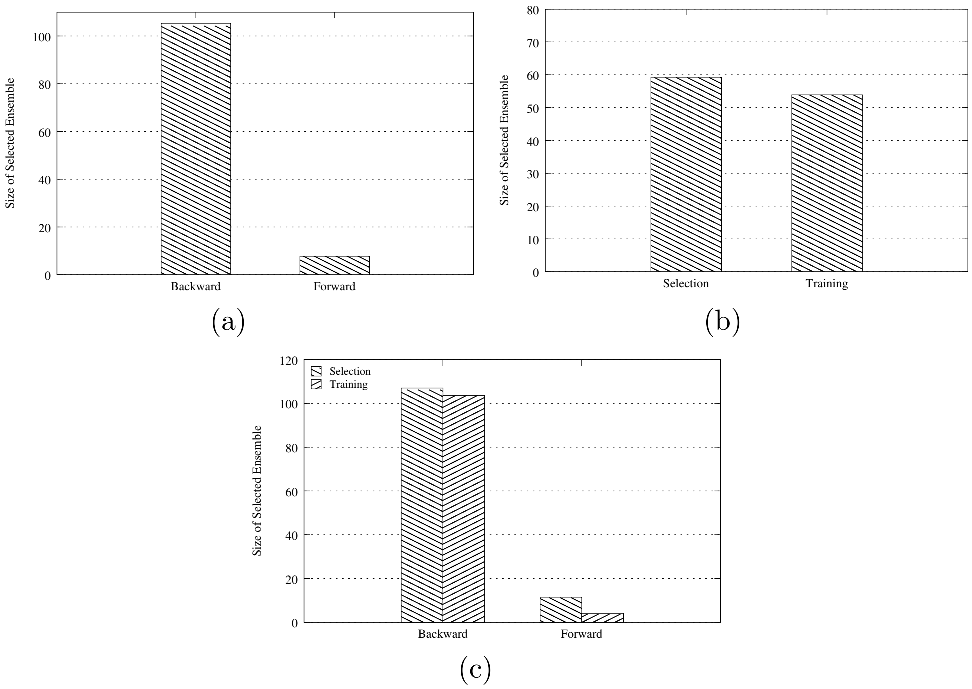
Table 7 shows the average size of the selected ensembles for each algorithm on each predicted variable. Figure 5 presents aggregates of the mean size of the selected ensemble for the different values for the search 5(a), evaluation dataset 5(b) parameters, as well as for pairs of values of the direction and evaluation dataset parameters 5(c).

Table 7
Average size of the selected ensembles for the different algorithms on each predicted variable.

	o1	o2	o3	o4	Average Size
BSRDIV	199.0	199.0	199.0	199.0	199.0
BSRMSE	16.2	11.5	17.7	14.5	14.975
BTRDIV	199.0	193.1	199.0	199.0	197.525
BTRMSE	2.4	13.4	6.2	16.9	9.725
FSRDIV	7.2	8.9	16.0	20.5	13.15
FSRMSE	4.9	10.4	12.5	11.1	9.725
FTRDIV	2.4	4.9	4.7	4.4	4.1
FTRMSE	2.4	5.1	4.5	4.1	4.025

A remarkable observation in Figure 5(a) is that the algorithms that search in the backward direction produce larger ensembles (105.3) than those that search in the forward direction (7.75). 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 5(b) we notice that the average size of the selected ensembles for the `xSxxxx` (59.21) algorithms is slightly larger than the `xTxxxx` (53.84) algorithms. This observation is also verified if we heed Figure 5(c). We can assume that the `xTxxxx` algorithms contain stronger regressors than those that manipulate the `xSxxxx` algorithms and select less regressors in order to achieve the maximum performance. But the performance of the `xTxxxx` algorithms is worst than the performance of the `xSxxxx` algorithms which means that those strong models are overtrained.



(a) (b)

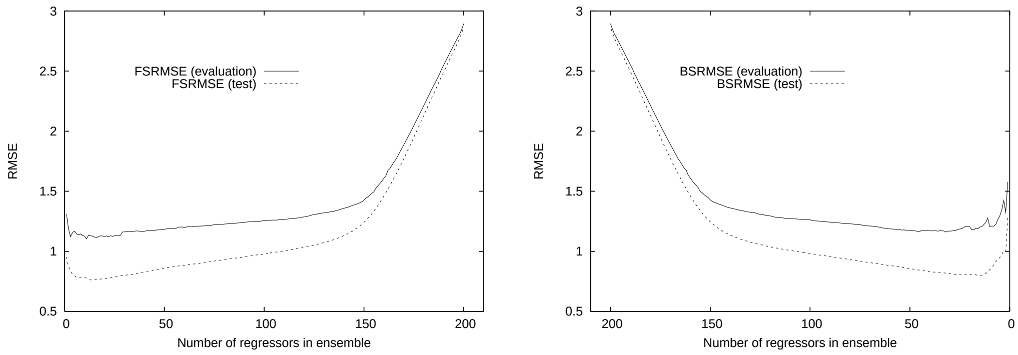
(c)

Fig. 5. Mean size of selected ensemble across all datasets and respective algorithms.

6.3 Predictive Performance vs. Ensemble Size

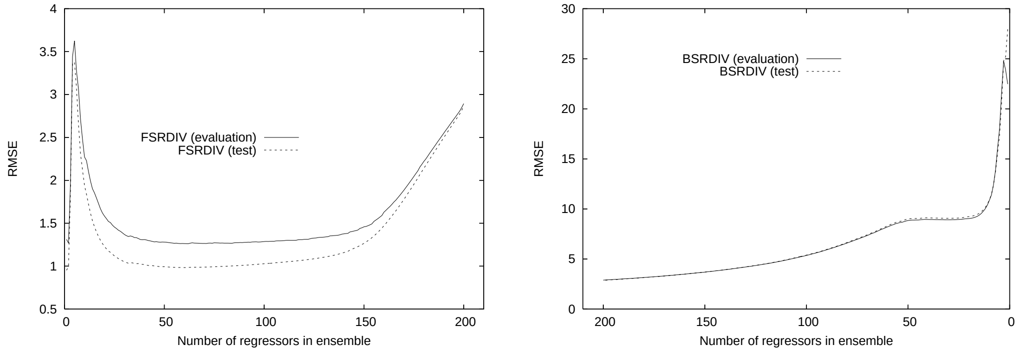
Figures 6(a), 6(b) present the RMSE curve both on the evaluation and the test set during the ensemble selection for one indicative variable (o2). Firstly, in Figures 6(a) and 6(b) we notice that the ensemble selection procedure improves the RMSE using a small number of regressors. Note that the final subensemble 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 final subensemble, which allows it to achieve high predictive performance.

In Figures 7(a) and 7(b) we notice that the `FSRDIV` and `BSRDIV` algorithms respectively, fail to select a good subensemble. More specifically, in the case of `FSRDIV` the RDIV measure guides ineffectively the algorithm as at the first steps inserts regressors that have bad performance. The `BSRDIV` algorithm seems to remove continually the good regressors from the ensemble, leading it to increase the error.



(a) (b)

Fig. 6. RMSE of FSRMSE and BSRMSE against the number of models.



(a) (b)

Fig. 7. RMSE of FSRDIV and BSRDIV against the number of models.

6.4 Type of Models

Figure 8 presents aggregates concerning the type of models that are selected across all the predicted variables. We focus on the results of the four best performing algorithms (FSRMSE, FSRDIV, BSRMSE, BTRMSE). The algorithms select equal sizes for both the SVM regressors (6.3) and NN regressors (5.5).

7 Conclusions and Future Work

In this paper we presented a general framework for the greedy ensemble selection algorithm. We decomposed the general ensemble selection algorithm into different parts and we accented the various options for this parts. Additionally, we applied the framework of the greedy ensemble selection algorithm on real data concerning water quality monitoring. We experimented with an ensemble of 200 regressors consisting of NNs and SVMs.

The results have shown that using a separate unseen set for the evaluation,

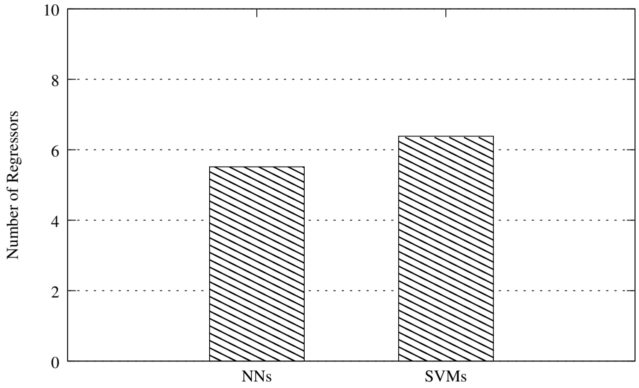


Fig. 8. Aggregates concerning the type of models that are selected by FSRMSE, FSRDIV, BSRMSE and BTRMSE.

leads the algorithm to improve its performance. Also, the algorithm manages 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 concerns, we concluded that it does not affect importantly the performance of the greedy ensemble selection algorithm.

References

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

[2] H. Blockeel, L. De Raedt, Top-down induction of first order logical decision trees, Artificial Intelligence 101 (1–2) (1998) 285–297.

[3] H. Blockeel, S. Dzeroski, J. Grbovic, Simultaneous prediction of multiple chemical parameters of river water quality with tilde, in: J. Zytkow, J. Rauch (eds.), Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery, Springer-Verlag, 1999, pp 32–40.

[4] G. Brown, J. L. Wyatt, P. Tino, Managing diversity in regression ensembles, Journal of Machine Learning Research 6 (2005) 1621–1650.

[5] R. Caruana, A. Munson, A. Niculescu-Mizil, Getting the most out of ensemble selection, in: J. Liu, B. W. Wah (eds.), Sixth International Conference in Data Mining (ICDM '06), IEEE Computer Society, 2006, pp 828–833.

[6] R. Caruana, A. Niculescu-Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, in: C. E. Brodley (ed.), Proceedings of the 21st International Conference on Machine Learning, ACM, 2004, pp 137–144.

[7] K. Chau, A split-step pso algorithm in prediction of water quality pollution, in: J. Wang, X. Liao, Z. Yi (eds.), Proceedings of the 2nd International Symposium on Neural Networks, Springer-Verlag, 2005.

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

[9] T. G. Dietterich, Machine-learning research: Four current directions, AI Magazine 18 (4) (1997) 97–136.

[10] S. Dzeroski, D. Demsar, J. Grbovic, Predicting chemical parameters of river water quality from bioindicator data, Applied Intelligence 13 (1) (2000) 7–17.

[11] W. Fan, F. Chu, H. Wang, P. S. Yu, Pruning and dynamic scheduling of cost-sensitive ensembles, in: R. Dechter, M. Kearns, R. Sutton (eds.), 18th National Conference on Artificial intelligence, American Association for Artificial Intelligence, 2002, pp 146–151.

[12] F. Fdez-Riverola, J. Corchado, CBR based system for forecasting red tides, Knowledge-Based Systems 16 (2003) 321–328.

[13] F. Fdez-Riverola, J. Corchado, FSfRT: Forecasting system for red tides, Applied Intelligence 21 (2004) 251–264.

[14] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Annals of Mathematical Statistics 11 (1940) 86–92.

[15] G. Giacinto, F. Roli, G. Fumera, Design of effective multiple classifier systems by clustering of classifiers, in: B. Werner (ed.), 15th International Conference on Pattern Recognition, 2000, pp 160–163.

[16] E. Hatzikos, The andromeda network for monitoring the quality of water and air elements, in: Proceedings of the 2nd Conference on Technology and Automation, Thessaloniki, Greece, 1998, pp 63–68.

[17] E. Hatzikos, A fully automated control network for monitoring polluted water elements, in: Proceedings of the 4th Conference on Technology and Automation, Thessaloniki, Greece, 2002, pp 443–448.

[18] E. Hatzikos, L. Anastasakis, N. Bassiliades, I. Vlahavas, Applying neural networks with active neurons to sea-water quality measurements, in: B. Plamenka, V. Chatzis (eds.), Proceedings of the 2nd International Scientific Conference on Computer Science, IEEE Computer Society, Bulgarian Section, 2005, pp 114–119.

[19] E. Hatzikos, G. Tsoumakas, G. Tzanis, N. Bassiliades, I. Vlahavas, An empirical study of sea water quality prediction, Technical report tr-lpis-231-07, Aristotle University of Thessaloniki, available at http://mlkd.csd.auth.gr/publications.asp (2007).

[20] D. Hernandez-Lobato, G. Martinez-Munoz, A. Suarez, Pruning in ordered regression bagging ensembles, in: G. G. Yen (ed.), Proceedings of the IEEE World Congress on Computational Intelligence, 2006, pp 1266–1273.

[21] A. Lehmann, M. Rode, Long-term behaviour and cross-correlation water quality analysis of the river Elbe, Germany, Water Research 35 (9) (2001) 2153–2160.

[22] Y. Liu, X. Yao, Ensemble learning via negative correlation, Neural Networks 12 (10) (1999) 1399–1404.

[23] D. Margineantu, T. Dietterich, Pruning adaptive boosting, in: D. H. Fisher (ed.), Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann, 1997, pp 211-218.

[24] P. B. Nemenyi, Distribution-free multiple comparisons, Ph.D. thesis, Princeton University (1963).

[25] I. Partalas, E. Hatzikos, G. Tsoumakas, I. Vlahavas, Ensemble selection for water quality prediction, in: K. Margaritis, L. Iliadis (eds.), 10th International Conference on Engineering Applications of Neural Networks, 2007, pp 428–435.

[26] I. Partalas, G. Tsoumakas, I. Katakis, I. Vlahavas, Ensemble pruning using reinforcement learning, in: G. Antoniou (ed.), 4th Hellenic Conference on Artificial Intelligence (SETN 2006), 2006, pp 301–310.

[27] K. Reckhow, Water quality prediction and probability network models, Canadian Journal of Fisheries and Aquatic Sciences 56 (1999) 1150–1158.

[28] C. Romero, J. Shan, Development of an artificial neural network-based software for prediction of power plant canal water discharge temperature, Expert Systems with Applications 29(4) (2005) 831–838.

[29] N. Rooney, D. Patterson, C. Nugent, Reduced ensemble size stacking, in: T. M. Khoshgoftaar (ed.) 16th International Conference on Tools with Artificial Intelligence, 2004, pp 266–271.

[30] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective Voting of Heterogeneous Classifiers, in: J.-F. Booulicaut, F. Esposito, F. Giannotti, D. Pedreschi (eds.), Proceedings of the 15th European Conference on Machine Learning, 2004, pp 465–476.

[31] N. Ueda, R. Nakano, Generalization error of ensemble estimators, in: B. Wah (ed.), Proceeding of IEEE International Conference on Neural Networks, 1996, pp 90–95.

[32] S. Weiss, N. Indurkhya, Predictive Data Mining: A practical guide, Morgan Kaufmann, 1997.

[33] I. Witten, E. Frank, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2005.

[34] Y. Zhang, S. Burer, W. N. Street, Ensemble pruning via semi-definite programming, J. Mach. Learn. Res. 7 (2006) 1315–1338.

[35] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: Many could be better than all, Artificial Intelligence 137 (1-2) (2002) 239–263.