

# Ensemble Selection for Water Quality Prediction

Ioannis Partalas

Evangelos V. Hatzikos

Grigorios Tsoumakas

Ioannis Vlahavas

## 1 Introduction

Ensemble methods [7] 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. In addition, they offer practical solutions to several other problems including *scaling inductive algorithms to large databases, learning from multiple physically distributed data sets* and *learning from concept-drifting data streams*.

[XXX add citations for the above statements XXX]

Typically, ensemble methods comprise two phases: a) the production of multiple predictive models, and b) their combination. Recent work [18, 12, 16, 9, 24, 5, 19, 2, 20], has considered an additional intermediate phase that deals with the reduction of the ensemble size prior to combination. This phase is commonly named *ensemble selection*.

This paper studies the greedy ensemble selection algorithm for ensembles of regression models. The algorithm attempts to find the globally best subset of regressors by making local greedy decisions for changing the current subset. We explore two interesting parameters of this algorithm: a) the direction of search (forward, backward), and b) the performance evaluation dataset (training set, validation set) on a large ensemble (200 models) of neural networks and support vector machines.

Experimental comparison of the different parameters are performed on an application domain with important social and commercial value: water quality monitoring. In specific we experiment on real data collected from an underwater sensor system.

## 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. [27] presented an approach based on a genetic algorithm. More specifically, the genetic algorithm evolve 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 not exhibit a predefined threshold are dropped.

Bakker and Heskes [1]...

Rooney et al. [23] 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

Hernandez et al. [15] introduced an ordered method, where each regressor is ordered according

## 2.2 Water Quality Prediction

[21] 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.

[4] 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 [3] was used for dealing with the above problems.

[8] 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.

[17] 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.

[22] 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.

[6] 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 [10, 11], 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.

[13] 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

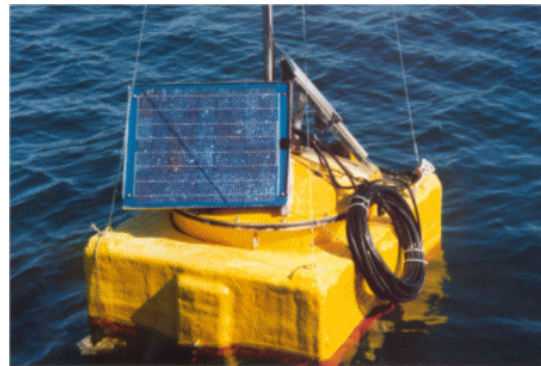
This section describes the system that collected the data used in our study, the pre-processing approach that we followed and initial exploratory data analysis.

#### 3.1 The Andromeda analyzer

The data used in this study have been produced by the Andromeda Analyzer (Hatzikos, 1998; Hatzikos, 2002). 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.



**Figure 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.

Furthermore, the central station hosts an intelligent alerting system [14] that monitors sensor data and reasons about the current level of water suitability for various aquatic uses, such as swimming and piscicultures. The aim of this intelligent alerting system is to help the authorities in the "decision-making" process in the battle against the pollution of the aquatic environment, which is very vital for the public health and the economy of Northern Greece. The expert system determines, using fuzzy logic, when certain environmental parameters exceed certain "pollution" limits, which are specified either by the authorities or by environmental scientists, and flags out appropriate alerts.

### 3.2 Data Preprocessing

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 in the data by the 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.

Based on the above remarks, the communication policy of the data collection system could be altered, in order to save energy if such a system was deployed in an ecosystem with limited sunlight. Instead of transmitting the data every hour, the local stations could transmit the average of their hourly measurements every  $k$  hours. For higher energy efficiency, the sensors themselves could operate every  $k$  hours and send their unique measurement. However, such a policy is less resilient to transmission failures and outliers. A different, adaptive, policy would let the local stations transmit their hourly measurements, only when the difference of at least one of the measurements with the previously transmitted corresponding measurement exceeds a predefined threshold. This would allow to save energy when hour to hour differences are negligible. The central station, assumes that the measurement values are the same if no values are transmitted.

## 4 Methodology

In a regression problem the goal is to learn a mapping from an input space  $X$  to an output value  $y$  using a set of training examples,  $D = \{(x_i, y_i), i = 1, 2, \dots, N\}$ , where each example consist of a feature vector  $x_i$  and the true value  $y_i$ .

Let  $H = \{h_t, t = 1, 2, \dots, T\}$  denote the set of regressors or hypotheses of an ensemble, where each regressor  $h_t$  maps an input vector  $x$  to an output vector  $y$ . Also let us denote as  $S \subseteq H$  the current subensemble during the search in the space of subensembles.

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. The main aspects of such an algorithm are the direction of search and the evaluation function used for evaluating the different branches of the search.

Based on the direction of search we have two main categories of greedy ensemble selection algorithms: a) *forward selection*, and b) *backward elimination*. In forward selection, the current regressor subset  $S$  is initialized to the empty set. The algorithm continues by iteratively adding to  $S$  the regressor  $h_t \in H \setminus S$  that optimizes an evaluation function  $f_{FS}(S, h, D)$ . This function evaluates the addition of regressors  $h$  in the current subset  $S$  based on the dataset  $D$ . For example,  $f_{FS}$  could return the mean squared error of the ensemble  $S \cup h$  on the data set  $D$  by combining the decisions of the classifiers with the method of voting. Figure 2 shows the pseudocode of the forward selection ensemble selection algorithm.

In backward elimination, the current regressor subset  $S$  is initialized to the complete ensemble  $H$  and the algorithm

**Input:** An ensemble of regressors  $H$ , an evaluation function  $f_{FS}$ , a pruning set  $D$   
**Output:** A subset of regressors  $S$   
 $S = \emptyset$ ;  
**while**  $S \neq H$  **do**  
     $h_t = \arg \max_{h \in H \setminus S} f_{FS}(S, h, D)$ ;  
     $S = S \cup \{h_t\}$ ;  
**return**  $S$

**Figure 2. The forward selection method in pseudocode**

continues by iteratively removing from  $S$  the regressor  $h_t \in S$  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 dataset  $D$ . For example,  $f_{BE}$  could return a measure of diversity for the ensemble  $S \setminus \{h\}$ , calculated based on the data of  $D$ . Figure 3 shows the pseudocode of the backward elimination ensemble selection algorithm.

**Input:** An ensemble of regressors  $H$ , an evaluation function  $f_{BE}$ , a pruning set  $D$   
**Output:** A subset of regressors  $S$   
 $S = H$ ;  
**while**  $S \neq \emptyset$  **do**  
     $h_t = \arg \max_{h \in S} f_{BE}(S, h, D)$ ;  
     $S = S \setminus \{h_t\}$ ;  
**return**  $S$

**Figure 3. The backward selection method in pseudocode**

One of the main components of the greedy ensemble selection algorithm is the evaluation function. This function, consist of two subcomponents:

- evaluation dataset: There are two approaches here. The first it to use the training dataset for evaluation as it offers the benefit of plenty data, but is susceptible to the danger of overfitting. The second approach is to withhold a part of the training set for evaluation. It diminishes the problem of overfitting, but reduces the amount of data that are available for training.
- evaluation measure: it can be clustered in two major categories: those that are based on performance and those on diversity. Performance metrics are ac-

curacy, root-mean-squared-error (RMSE) and mean cross-entropy.

It is generally accepted that an ensemble should contain diverse models in order to achieve high predictive performance. For ensembles of regressors the diversity can be formulated in terms of covariance by decomposing the mean-squared-error (MSE) into three components: bias-variance-covariance [25]. The diversity that optimizes the MSE is that which optimally balances the three components.

In this work we use the performance metric RMSE as the evaluation metric in the greedy selection algorithm. Let us denote the prediction of the  $h_t$  classifier for an instance  $x$  as  $h_t(x)$ . The ensemble output for the instance  $x$  is:

$$h_{ens}(x) = \frac{1}{T} \sum_{t=1}^T h_t(x).$$

The root mean squared error is:

$$e = \sqrt{\frac{1}{N \cdot T} \sum_{t=1}^T \sum_{n=1}^N (h_t(x_n) - y_n)^2}$$

## 5 Experimental Setup

This section provides information about the experimentation methodology that we followed.

### 5.1 Ensemble creation

In order to create an ensemble of regressors we follow the subsequent procedure: Initially, the whole dataset is split in three disjunctive parts, a training set, a pruning set and a test set with  $Tr\%$ ,  $Pr\%$  and  $Ts\%$  percentage form the initial dataset respectively ( $Tr + Pr + Ts = 100$ ).

Then an ensemble production method is used on the training set, in order to produce  $T$  models that constitute the initial ensemble. We experiment with heterogeneous models, where we run different learning algorithms with different parameter configurations.

The WEKA machine learning library was used as the source of the learning algorithms [26]. We trained 80 multi-layer perceptrons and 120 support vector machines (SVMs). The different parameters used to train the algorithms were the following:

- multilayer perceptrons: we used 8 values for the nodes in the hidden layer  $\{1, 2, 4, 8, 16, 32, 64, 128\}$ , 4 values for the momentum term  $\{0.0, 0.2, 0.5, 0.8\}$  and 2 values for the learning rate  $\{0.6, 0.9\}$ .

- SVMs: we used 12 values for the complexity parameter  $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 10^2, 10^3, 10^4\}$ , and 10 different kernels. We used 2 polynomial kernels (of degree 2 and 3) and 8 radial kernels (gamma  $\{0.001, 0.005, 0.01, 0.1, 0.5, 1, 2\}$ ).

In the next step, we use the general greedy ensemble selection algorithm after setting the parameters of direction, evaluation dataset and metric. For the direction parameter we use two values, forward and backward. Evaluation dataset can be instantiated to either the training or the pruning set. As for the metric, we use the performance metric RMSE.

In order to integrate the estimates of the regressors we use a simple linear combination function, which aggregates the estimates. The final selected subensemble is that with the lowest error on the evaluation set (using linear combination). The resulting ensemble is evaluated on the test set, using linear combination for model combination. The whole experiment is performed 10 times for each dataset and the results are averaged.

We define the size for each of the training, pruning and test set to 40%, 40% and 20% respectively. We choose equal sizes for training and pruning sets in order to provide a fair comparison between the algorithms.

## 6 Results and Discussion

Table 1 presents the average RMSE for each configuration of the greedy ensemble selection algorithm on each dataset. We notice that forward-prune is the best performing configuration in three cases out of four. An interesting fact is that the two configurations that use the pruning set for evaluation (forward-prune, backward-prune) have better performance than the other two that use the training set. This strongly indicates that using a separate dataset for evaluation offers increased predictive accuracy to the greedy ensemble selection algorithm.

As far as the direction parameter is concerned, we can't conclude whether one of the two values (forward, backward) dominates the other. The backward direction exhibits better performance when the training set is used for evaluation, while the forward direction appears to be better when the pruning set is used for evaluation.

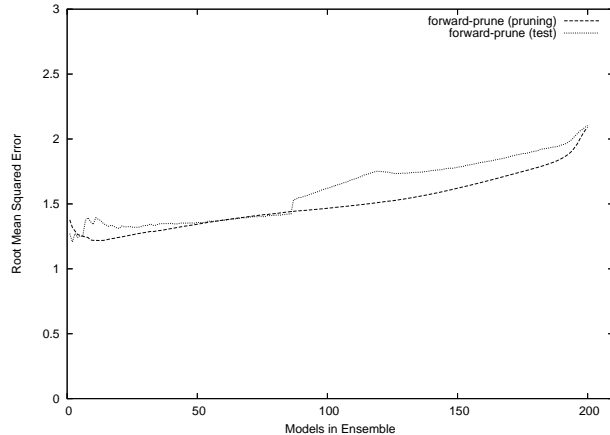
Table 2 shows the average size of the final subensembles that are selected by the different configurations of the greedy ensemble selection algorithm on each dataset. A general remark is that the number of selected models is small compared to the size of the original ensemble. Only 2.5% to 16% of the 200 models are finally selected by the algorithm. Interestingly, configurations that search in the forward direction tend to produce smaller ensembles than those that search in the backward direction.

|                | o1    | o2    | o3    | o4    |
|----------------|-------|-------|-------|-------|
| forward train  | 7.127 | 0.240 | 3.088 | 2.498 |
| forward prune  | 1.356 | 0.152 | 1.256 | 0.859 |
| backward train | 4.896 | 0.231 | 2.473 | 1.632 |
| backward prune | 4.821 | 0.145 | 1.287 | 0.911 |

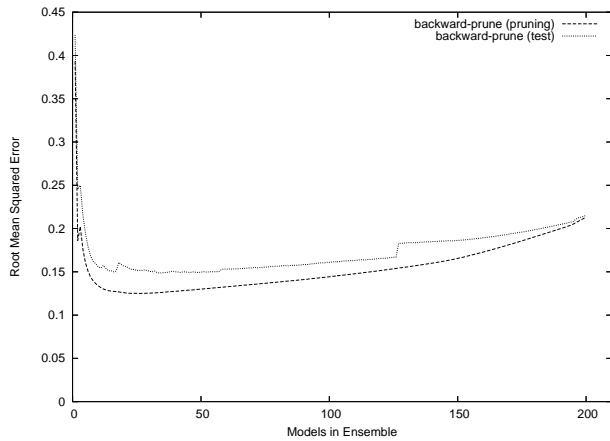
**Table 1. Average errors of each algorithm on each dataset.**

|                | o1 | o2 | o3 | o4 |
|----------------|----|----|----|----|
| forward train  | 5  | 7  | 7  | 7  |
| forward prune  | 11 | 6  | 12 | 13 |
| backward train | 25 | 24 | 19 | 28 |
| backward prune | 21 | 16 | 32 | 28 |

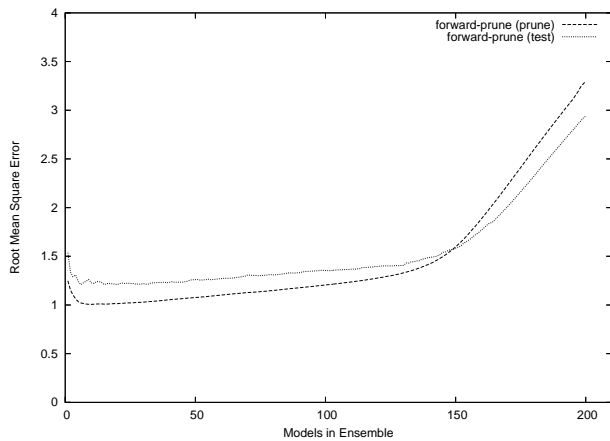
**Table 2. Average size of pruned ensembles for each algorithm on each dataset.**



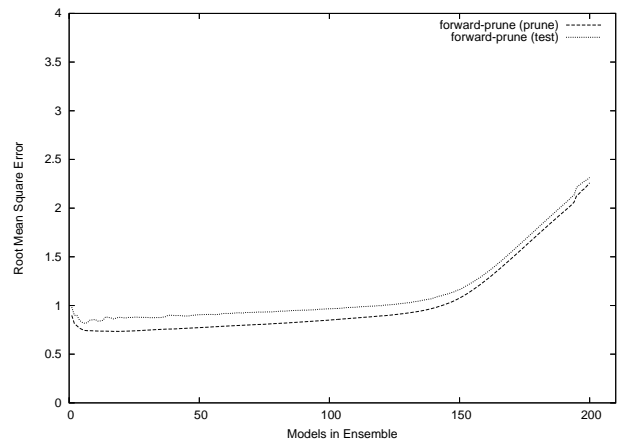
**Figure 4. RMSE of forward-prune method, for the dataset o1, with respect to the number of regressors in the ensemble, for both the pruning and test set.**



**Figure 5. RMSE of backward-prune method, for the dataset o2, with respect to the number of regressors in the ensemble, for both the pruning and test set.**



**Figure 6. RMSE of forward-prune method, for the dataset o3, with respect to the number of regressors in the ensemble, for both the pruning and test set.**



**Figure 7. RMSE of forward-prune method, for the dataset o4, in respect with the number of regressors in the ensemble, for both the pruning and test set.**

Next, we present figures depicting the error curve both on the evaluation set and the test set during ensemble selection. For simplicity, we present these curves for the best performing configuration on each dataset. Figures 4, 5, 6 and 7 plot the RMSE against the different sizes of the ensemble (1-200) for target variables o2, o3 and o4 respectively. In all cases, the error decreases as the algorithm inserts (removes) into (from) the subensemble the most (least) accurate regressors. We notice that a single selected regressor has quite good performance, but the minimum RMSE is achieved for a small number of models. Then we notice a slow linear increase of the error from the minimum RMSE point up to an ensemble with about 3/4 of all the models and a steepest increase of the error up to the original ensemble with all 200 models.

Note that the final subensemble that is selected by the algorithm, is the one that corresponds to the minimum of the pruning set error curve. In the figures we observe that this minimum point corresponds to a near-optimal point in the test set error curve. This shows that the greedy ensemble selection algorithm manages to select an appropriate size for the final subensemble, which allows it to achieve high generalization performance. Furthermore, as we have already seen in Table 2 the number of models selected this way is smaller than using a fixed size of 20% of the models, as in [REF], leading to further reduction of the computational cost of the final subensemble.

## 7 Conclusions and Future Work

### Acknowledgements

Evaggelos Hatzikos is supported by the European Social Fund & National Resources - EPEAEK II - ARCHIMIDIS

### References

- [1] B. Bakker and T. Heskes. Clustering ensembles of neural network models. *Neural Networks*, 16(2):261–269, 2003.
- [2] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. Ensemble diversity measures and their application to thinning. *Information Fusion*, 6(1):49–62, 2005.
- [3] H. Blockeel and L. De Raedt. Top-down induction of first order logical decision trees. *Artificial Intelligence*, 101(1–2):285–297, 1998.
- [4] H. Blockeel, S. Dzeroski, and J. Grbovic. Simultaneous prediction of multiple chemical parameters of river water quality with tilde. In *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery*, volume 1704 of *LNAI*, pages 32–40. Springer-Verlag, 1999.
- [5] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning*, page 18, 2004.
- [6] K. Chau. A split-step pso algorithm in prediction of water quality pollution. In *Proceedings of the 2nd International Symposium on Neural Networks*, pages 1034–1039, 2005.
- [7] T. G. Dietterich. Machine-learning research: Four current directions. *AI Magazine*, 18(4):97–136, 1997.
- [8] S. Dzeroski, D. Demsar, and J. Grbovic. Predicting chemical parameters of river water quality from bioindicator data. *Applied Intelligence*, 13(7–17), 2000.
- [9] W. Fan, F. Chu, H. Wang, and P. S. Yu. Pruning and dynamic scheduling of cost-sensitive ensembles. In *Eighteenth national conference on Artificial intelligence*, pages 146–151. American Association for Artificial Intelligence, 2002.
- [10] F. Fdez-Riverola and J. Corchado. Cbr based system for forecasting red tides. *Knowledge-Based Systems*, 16(321–328), 2003.
- [11] F. Fdez-Riverola and J. Corchado. Fsfrrt: Forecasting system for red tides. *Applied Intelligence*, 21(251–264), 2004.
- [12] G. Giacinto, F. Roli, and G. Fumera. Design of effective multiple classifier systems by clustering of classifiers. In *15th International Conference on Pattern Recognition, ICPR 2000*, pages 160–163, 3–8 September 2000.
- [13] E. Hatzikos, L. Anastasakis, N. Bassiliades, and I. Vlahavas. Simultaneous prediction of multiple chemical parameters of river water quality with tilde. In *Proceedings of the 2nd International Scientific Conference on Computer Science*, pages 114–119. IEEE Computer Society, Bulgarian Section, 2005.
- [14] E. Hatzikos, N. Bassiliades, L. Asmanis, and I. Vlahavas. Monitoring water quality through a telematic sensor network and a fuzzy expert system. *Expert Systems*, 24(4):(to appear), 2007.
- [15] D. Hernandez-Lobato, G. Martinez-Munoz, and A. Suarez. Pruning in ordered regression bagging ensembles. In *Proceedings of the IEEE World Congress on Computational Intelligence (IJCNN)*, pages 1266–1273, 2006.
- [16] A. Lazarevic and Z. Obradovic. Effective pruning of neural network classifiers. In *2001 IEEE/INNS International Conference on Neural Networks, IJCNN 2001*, pages 796–801, 15–19 July 2001.
- [17] A. Lehmann and M. Rode. Long-term behaviour and cross-correlation water quality analysis of the river elbe, germany. *Water Research*, 35(9):2153–2160, 2001.
- [18] D. Margineantu and T. Dietterich. Pruning adaptive boosting. In *Proceedings of the 14th International Conference on Machine Learning*, pages 211–218, 1997.
- [19] G. Martinez-Munoz and A. Suarez. Aggregation ordering in bagging. In *International Conference on Artificial Intelligence and Applications (IASTED)*, pages 258–263. Acta Press, 2004.
- [20] I. Partalas, G. Tsoumakas, I. Katakis, and I. Vlahavas. Ensemble pruning via reinforcement learning. In *4th Hellenic Conference on Artificial Intelligence (SETN 2006)*, pages 301–310, May 18–20 2006.
- [21] K. Reckhow. Water quality prediction and probability network models. *Canadian Journal of Fisheries and Aquatic Sciences*, 56:1150–1158, 1999.
- [22] C. Romero and J. Shan. Development of an artificial neural network-based software for prediction of power plant canal water discharge temperature. *Expert Systems with Applications*, 29:831–838, 2005.
- [23] N. Rooney, D. Patterson, and C. Nugent. Reduced ensemble size stacking. In *16th International Conference on Tools with Artificial Intelligence*, pages 266–271, 2004.
- [24] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective Voting of Heterogeneous Classifiers. In *Proceedings of the 15th European Conference on Machine Learning, ECML2004*, pages 465–476, 2004.
- [25] N. Ueda and R. Nakano. Generalization error of ensemble estimators. In *Proceeding of International Joint Conference on Neural Networks*, pages 90–95, 1996.
- [26] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition, 2005.
- [27] J. W. Z.-H. Zhou and W. Tang. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 2002, 137(1-2): 239-263, 137(1-2):239–263, 2002.