Ensemble Selection for Water Quality Prediction

Ioannis Partalas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, 54124
partalas@csd.auth.gr

Evaggelos V. Hatzikos
Department of Automation
Tech. Educ. Institute of Thessaloniki
Thessaloniki, 57400
hatzikos@csd.auth.gr

Grigorios Tsoumakas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, 54124
greg@csd.auth.gr

Ioannis Vlahavas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, 54124
vlahavas@csd.auth.gr

Abstract

This paper studies the greedy ensemble selection algorithm for ensembles of regression models. 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.

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.

Typically, ensemble methods comprise two phases: a) the production of multiple predictive models, and b) their combination. Recent work [20, 11, 29, 24, 25, 5, 1, 21, 17], 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. This algorithm searches for 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 (NNs) and support vector machines (SVMs).

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. 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.

The rest of the paper is structured as follows: Section 2 reviews related work on ensemble selection in regression problems and on water quality prediction. Section 3 describes the data collection and pre-processing approach that was followed in order to prepare the data for analysis. Section 4 presents a formulation of the general greedy ensemble selection algorithm, as well as the main aspects of it. Section 5 provides information about the experimentation methodology that we followed. Section 6 discusses the results of the experiments and finally, Section 7 concludes this 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. [29] 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.

Rooney et al. [24] 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).

Hernandez et al. [17] 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 pruned ensemble.

Liu and Yao [19] 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 qualified and measured.

2.2 Water Quality Prediction

Reckhow [22] 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. [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.

Lehmann and Rode [18] 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 [23] 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 [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 [9, 10], 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. [14] 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 [12, 13]. 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.

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.

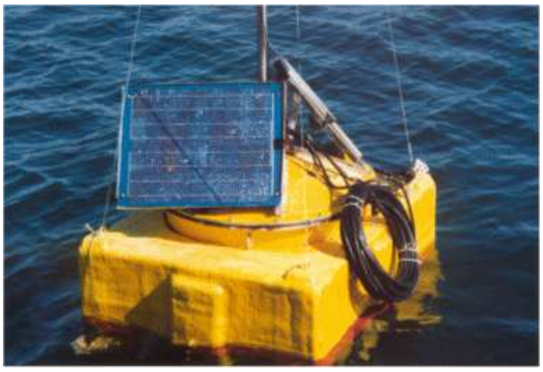

Figure 1. One of the three local measurement stations of the Andromeda system.

Furthermore, the central station hosts an intelligent alerting system [15] 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 Pre-processing

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 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.

We completely removed measurements concerning dissolved oxygen and turbidity, as the correspondind 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 modelling 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* [27]. 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 [16] we set the window to 9 and the lead to 2.

4 The Greedy Ensemble Selection Algorithm

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, $D = \{(x_i, y_i), i = 1, 2, \ldots, N\}$, where each example consist of a feature vector x_i and the true value y_i.

Let $H = \{h_t, t = 1, 2, \ldots, T\}$ 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. 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, $S \subseteq H$. 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 $h_t \in H \backslash S$ 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 models with the method of voting. Figure 2 shows the pseudocode of the forward selection ensemble selection 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
    $h_t = \underset{h \in H \backslash S}{\arg\max} f_{FS}(S, h, D)$;
    $S = S \cup \{h_t\}$;
return $S$
```

Figure 2. The forward selection method in pseudocode

In backward elimination, the current regressor subset S is initialized to the complete ensemble H and the algorithm 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 $S \backslash \{h\}$, 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
    $h_t = \underset{h \in S}{\arg\max} f_{BE}(S, h, D)$;
    $S = S \setminus \{h_t\}$;
return $S$
```

Figure 3. The backward elimination method in pseudocode

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

There are two approaches concerning the evaluation dataset. The first it to use the training dataset for evaluation as it offers the benefit of plenty data, but is suspectible 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 availiable for training.

There are two major categories of evaluation measures: *performance-based* and *diversity-based*. Performance-based metrics include accuracy, root-mean-squared-error (RMSE) and mean cross-entropy. In ensembles of regressors, diversity can be formulated in terms of covariance by decomposing the mean-squared-error (MSE) into three components: bias-variance-covariance [26]. The diversity that optimizes the MSE is that which optimally balances the three components.

5 Experimental Setup

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 consitute 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 [28]. We trained 80 multilayer perceptrons and 120 support vector machines. The different parameters used to train the algorithms were the following (the rest parameters were left unchanged to their default values):

- multilayer perceptrons: we used one hidden layer and 8 values for the nodes in this 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 measure. For the direction parameter we use two values, forward and backward. Evaluation dataset can be instantiated to either the training or the pruning set. The acronyms of the four different algorithms are the following: FT (forward-train), FP (forward-prune), BT (backward-train), BP (backward-prune).

As for the evaluation measure, we use the performance-based RMSE. Let us denote the prediction of the h_t classifier for an intance 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 RMSE for the whole set of instances is:

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

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 (FP, BP) 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.

	o1	o2	o3	o4
FT	7.127	0.240	3.088	2.498
FP	1.356	0.152	1.256	0.839
BT	4.896	0.231	2.473	1.632
BP	4.821	0.145	1.287	0.911

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

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
FT	5	7	7	7
FP	11	6	12	13
BT	25	24	19	28
BP	21	16	32	28

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

Table 3 depicts the average number of NNs and SVMs in the selected ensembles for each target variable. The FT algorithm selects only SVMs while BT contructs ensembles containing 75% of SVMs and 25% of NNs. If we assume that the best models in the case of using the training set are SVMs then we can explain this behaviour. Also, based on this finding we can conclude why both FT and BT are the worst, as they don't manage to select regressors that are diverse enough. On the other hand, FP and BP algorithms select almost equal sizes of NNs and SVMs leading them to higher predictive performance.

	o1		o2		o3		o4	
	NNs	SVMs	NNs	SVMs	NNs	SVMs	NNs	SVMs
FT	0	5	0	7	0	7	0	7
FP	5	6	2	4	6	6	7	6
BT	8	17	9	15	6	13	7	21
BP	10	11	7	9	18	14	16	12

Table 3. Average number of NNs and SVMs in the pruned ensembles.

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 o1, o2, o3 and o4 respectively.

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

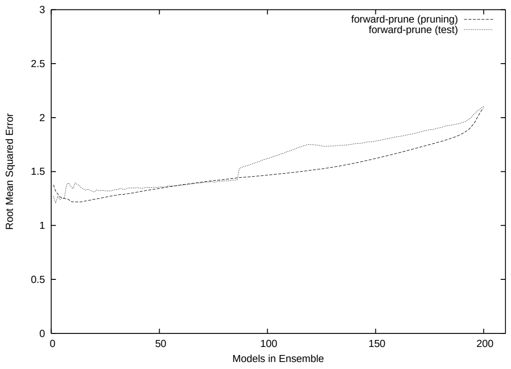

Figure 4. RMSE of forward-prune method, for the dataset o1, with respect to the number of regressors in the ensemble.

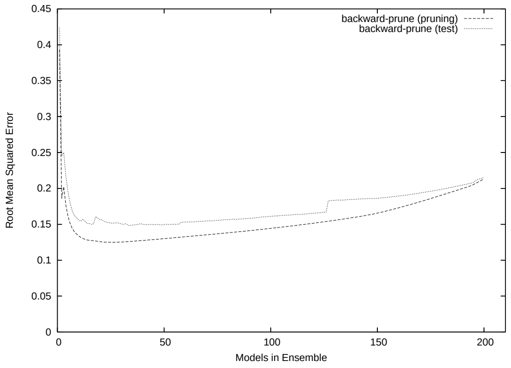

Figure 5. RMSE of backward-prune method, for the dataset o2, with respect to the number of regressors in the ensemble.

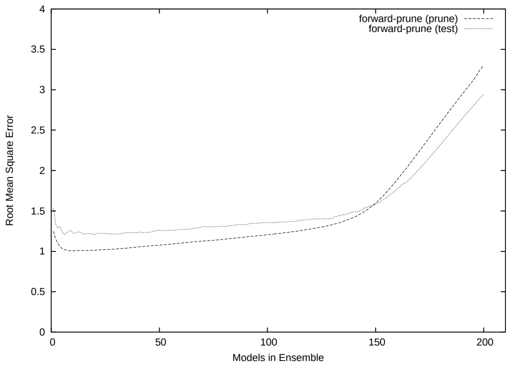

Figure 6. RMSE of forward-prune method, for the dataset o3, with respect to the number of regressors in the ensemble.

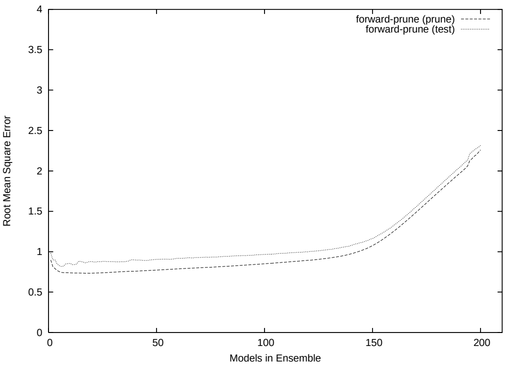

Figure 7. RMSE of forward-prune method, for the dataset o4, in respect with the number of regressors in the ensemble.

smaller than using a fixed size of 20% of the models, as in [17], leading to further reduction of the computational cost of the final subensemble.

7 Conclusions and Future Work

In this paper we presented an application of the greedy ensemble selection algorithm on real data concerning water quality monitoring. We explored two important parameters of the general algorithm, the direction of search and the evaluation set. 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, leads the algorithm to improve its performance. Also, the algorihm 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. In addition the algorithm selects a balanced mixture of NNs and SVMs that leads to increased diversity.

For future work, we intend to investigate other evaluation metrics, like diversity measures following the bias-variance-covariance decomposition originally proposed in [26].

Acknowledgements

This work has been (partially) co-funded by the European Social Fund & National Resources - EPEAEK II - ARCHIMIDIS.

References

[1] 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.

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

[3] 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.

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

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

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

[11] 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.

[12] 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, October 1998.

[13] E. Hatzikos. A fully automated control network for monitoring polluted water elements. In *Proceedings of the 4th Conference on Technology and Automation*, Thessaloniki, Greece, October 2002.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

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

[20] D. Margineantu and T. Dietterich. Pruning adaptive boosting. In *Proceedings of the 14th International Conference on Machine Learning*, pages 211–218, 1997.

[21] 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.

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

[23] 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.

[24] N. Rooney, D. Patterson, and C. Nugent. Reduced ensemble size stacking. In *16th International Conference on Tools with Artificial Intelligence*, pages 266–271, 2004.

[25] 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.

[26] N. Ueda and R. Nakano. Generalization error of ensemble estimators. In *Proceeding of International Joint Conference on Neural Networks*, pages 90–95, 1996.

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

[28] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition, 2005.

[29] Z.-H. Zhou, J. Wu, 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.