Application of model trees and other machine learning techniques for algal growth prediction in Yongdam reservoir, Republic of Korea

Nahm-Chung Jung, Ioana Popescu, Peter Kelderman,
Dimitri P. Solomatine and Roland K. Price

ABSTRACT

A promising new approach for eco-environmental modelling, such as algal growth prediction, is the data-driven modeling using machine learning techniques: an artificial neural network (ANN) being a typical method. Another method growing in popularity, based on the M5 model tree (MT) algorithm, is the use of piecewise linear regression models at the leaf nodes of the tree. M5 MTs using partial least-squares regression (PLSR) proposed in this paper were tested on a particular dataset and then compared to M5 MTs, MLF- and RBF-ANN and *k* nearest neighbours (*k*NN). With the dataset partitioned to periods of algal growth and no growth, M5 MTs using PLSR showed better results for algal growth prediction in the reservoir than using the annual dataset and other algorithms. This gives the idea that the M5-PLSR MTs, in spite of the lack of data, more effectively seeks latent vectors between the closely correlated multivariate dataset partitioned using clustering techniques. M5-PLSR MTs is a promising approach when there is a shortage of data required to build a more transparent learning process model, and a combination with clustering is recommended.

Key words | ANN, data-driven modelling (DDM), *k*NN, LSER, M5 MTs, PLSR

Nahm-Chung Jung (corresponding author)
Ioana Popescu
Dimitri P. Solomatine
Roland K. Price
Department of Hydroinformatics and Knowledge Management,
UNESCO-IHE Institute for Water Education,
PO Box 3015, 2601 DA, Delft,
The Netherlands

Nahm-Chung Jung (corresponding author)
Kwater (Korea Water Resources Corporation),
San 6-2, Yeonchuk-dong, Daedeok-gu,
Daejeon 307-711,
Republic of Korea

Nahm-Chung Jung (corresponding author)
Water Resources Section, Civil Engineering and Geoscience,
Delft University of Technology,
Stevinweg 1, 2628 CN, Delft,
The Netherlands
Tel.: +31 015 215 1889
E-mail: *chung@kwater.or.kr*

Peter Kelderman
Department of Environmental Resources,
UNESCO-IHE Institute for Water Education,
PO Box 3015, 2601 DA, Delft,
The Netherlands

INTRODUCTION

The conceptualization of most environmental processes involves the formulation of relationships among variables. These relationships do not necessarily imply that one variable causes another, but that significant associations exist among particular variables. Previously, research on environmental processes used to describe and analyze only univariate and bivariate datasets. Examples of the analysis of univariate datasets include confidence intervals for the mean, and techniques for correlation and regression.

Environmental conditions, such as eutrophication in a reservoir, often involve a large number of variables (attributes: e.g. temperature, pH, inorganic phosphate or Secchi depth) that are considered to be related to a particular dependent variable, such as algal growth. To understand the behavior of a particular phenomenon in the natural environment it has been customary to introduce controls on particular variables, thus reducing the number of variables describing the phenomenon by treating most of them as constants. This can be done, for example, by performing laboratory simulations using constant environment rooms. Such an approach enables the research to focus on a small number of variables, in spite of a large number of variables involved in the particular phenomenon, typically one or two, which can be analyzed using conventional statistical methods.

doi: 10.2166/hydro.2009.004

Data-driven modelling (data mining) is a promising new research direction for interpreting multivariate data (Pedrycz *et al.* 2002; Witten & Frank 2005). In interdisciplinary research detecting patterns and rules in large quantities of data, the terminologies used have varied according to the research fields involved. Data-driven modelling has proven a broader and increasing application tendency over recent years, based on artificial intelligence and/or statistics, complementing or replacing deterministic models in many research fields.

Simply put, a data-driven approach is the process of exploring relationships among a large number of variables and quantities of data in order to extract meaningful information from the data in the form of formulas, computer codes, patterns or rules. The resulting information can be stored as an abstract mathematical model, referred to as a data-driven model, and then new data are examined using the model to see if it fits the established model. From this information, actions can be taken to improve the model. In this sense, the data-driven model can be said to learn. For model learning with respect to environmental data, it is typical to predict a continuous numerical value rather than a discrete category (class) to which an example belongs (Quinlan 1992).

There are many data-driven (machine learning) techniques. These include standard regression, artificial neural network (ANN), k nearest neighbouring (kNN), regression trees, model trees (MTs) and prediction by prediscretization. Each technique has its weaknesses: standard regression is not a very powerful way of representing an induced function because it is restricted to a linear rather than a nonlinear relationship on data with spatial and temporal variation. ANN is more powerful, but suffers from opacity in that it does not disclose any information about the physical processes that it represents (Solomatine & Dulal 2003). kNN can be easily adopted to perform a real-valued prediction, but it uses the "local modeling" approach and lacks the generalization ensured, at least partly, by global models like ANN (Solomatine *et al.* 2007).

Regression tree models are based on an assumption of a linear dependence between inputs and outputs, with averaged numerical values at each leaf node of the tree. Therefore, at the leaves these models capture the linear dependence between one or more independent variables, x_n, and the dependent (or response) variable, y. Unlike regression tree models, MTs are tree-structured regression models that associate leaves with multiple linear regression functions used to calculate numerical values. Therefore, a model tree constructs piecewise linear models at the leaves, but overall it shows a nonlinear behavior. One distinct advantage is that the MT mechanism is more transparent than that of many other machine learning algorithms. Thus, one can easily follow a tree structure to understand how a decision has been made (Pedrycz & Sosnowski 2001). Additionally, when the number of instances (observations) is smaller than the number of attributes (variables) in each instance, a local model with a regression approach is no longer feasible. This problem is common in the partitioning technique when using an MT approach, such as linear regression, as the number of observations in the local model decreases with the expansion of the tree. Partial least-squares regression (PLSR) is an extension of multivariate linear regression, which solves this problem by considering in–out pairs remaining at each internal node and at the final leaf. Specifically, PLSR is effective in situations where use of the traditional least-squares error (LSE)-based multivariate method is severely limited because there are fewer instances than the number of attributes (Baffi *et al.* 1999; Tan *et al.* 2004).

In this paper, the PLSR technique was tested as the regression algorithm used in MT models for algal growth prediction of the Yongdam reservoir, Korea. Using MLP- and RBF-ANN, kNN and M5$'$ methods provided a comparison with results of the proposed M5-PLSR MTs.

Because the metabolism of algae is influenced by the season, the training and test datasets were partitioned into two groups: the algal growth and no-growth periods. The modelling results using an annual dataset, covering all two periods, were compared with those using the partitioned dataset, to test and identify if there are any shortcomings in the partitioning performance of MTs on algal growth and hibernation due to temperature variation.

DATA-DRIVEN MODELS

Model trees (MTs)

MTs are not as popular as ANN: they have only recently been introduced in the water sector (Kompare *et al.* 1997*a*;

Solomatine & Dulal 2003) and have not yet been widely applied. Solomatine (2005) demonstrated the application of MTs to hydrological and other problems, along with other data-driven models. The advantages of MTs are that they are more accurate than regression trees, more understandable than ANN, easy to train, and robust when dealing with missing data (Witten & Frank 2005).

There are two basic MT approaches: multiple adaptive regression splines (MARS; Friedman 1991) and M5 MTs (Quinlan 1992). This research used the M5 algorithm for inducing an MT. The MT approach involves two major procedures: building the tree and inferring knowledge from it. In Figure 1, for example, the tree-building procedure involves partitioning the input space into mutually exclusive regions using the linear regression model. In the inference procedure, a new instance is fed into one of the models at the leaves of the tree, according to a splitting condition adopted in the tree-building procedure, and then the predicted output is obtained from the linear model at the leaf.

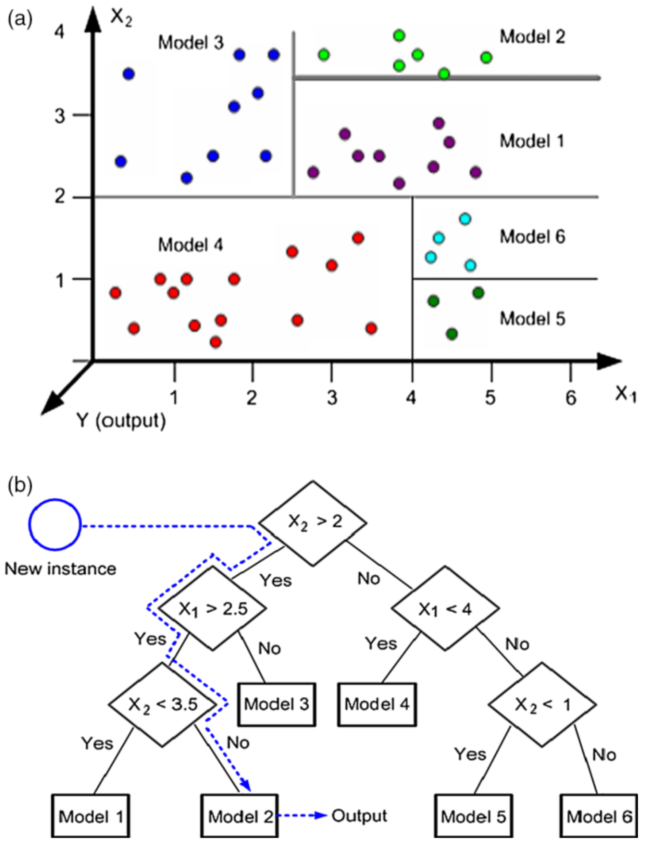


Figure 1 | Splitting of input space and prediction by the model trees for a new dataset. (a) Splitting of the input space such as $X_1 \times X_2$ by M5 model tree algorithm; each model is a linear regression model $y = a_0 + a_1 x_1 + a_2 x_2$. (b) Prediction for new instance by model tree.

There is a version of the M5 algorithm known as the M5′ algorithm proposed by Wang & Witten (1997). This algorithm has a similar structure to the M5 algorithm, but is able to deal effectively with missing values and enumerated attributes. These algorithms have the following three main steps.

Building the tree

The basic tree is formed using the splitting criterion, which treats the standard deviation of the class values that reach a node as a measure of the error at that node, and calculates the expected reduction in error as a result of testing each attribute at that node. The attribute that maximizes the expected error reduction is then selected. The standard deviation reduction (SDR) for M5 is calculated using the formula

$$\mathrm{SDR} = \mathrm{sd}(T) - \sum_i \frac{|T_i|}{|T|} \times \mathrm{sd}(T_i) \tag{1}$$

where sd is the standard deviation of the set of examples T that reach the node and T_i is the set that results from splitting the node according to the chosen attribute. The splitting process ceases when the class values of all the instances that reach a node vary by less than 5% of the standard deviation of the original instance set, or when only a few instances remain.

Pruning the tree

An over-fitting problem can occur during MT construction based on training data. Predictably, the accuracy of the tree for the training examples increases monotonically as the tree grows. However, this increases over-fitting, so that the accuracy measured over the independent test examples first increases, then decreases. A method for reducing this problem is termed “pruning”.

For use in the smoothing process, a linear model is also needed for each interior node of the tree, not just at the leaves. Prior to pruning, a model is calculated for each node of the unpruned tree. The model takes the form

$$y = x_0 + x_1a_1 + x_2a_2 + \dot{c} + x_ka_k$$

where $a_1, a_2, \ldots, a_k$ are attribute values. The weights $x_1, x_2, \ldots, x_k$ are calculated using a standard regression. However, only the attributes tested in the sub-tree below this node are used in the regression, because the other attributes which affect the predicted value have been taken into account in the tests that lead to the node.

The pruning procedure uses an estimate of the expected error that will be experienced at each node for the test data. First, the absolute difference between the predicted value and the actual output value is averaged for each of the training examples that reach the node. Because the trees have been built expressly for this dataset, this average will underestimate the expected error for new cases. To compensate for this, the output value is multiplied by the factor $(n + v)/(n - v)$, where n is the number of training examples that reach the node and v is the number of attributes in the model that represent the output value at that node. Therefore, this multiplication is done to avoid underestimating the error for new data, rather than the data against which it was trained. If the estimated error is lower at the parent, the leaf node can be dropped (Witten & Frank 2005).

Smoothing

A final stage is to use a smoothing process to compensate for sharp discontinuities that inevitably occur between adjacent linear models at the leaves of the pruned tree, particularly for some models constructed from a small number of training instances. The smoothing procedure described by Quinlan (1992) uses the leaf model to compute the predicted value, and that value is then filtered along the path back to the root, smoothing it at each node by combining it with the value predicted by the linear model for that node. This involves the calculation

$$p' = \frac{np + kq}{n + k} \tag{2}$$

where p' is the prediction passed up to the next higher node, p is the prediction passed to this node from below, q is the value predicted by the model at this node, n is the number of training instances that reach the node below and k is a constant. In general, smoothing substantially increases the accuracy of the predictions.

In the application of M5, it is not clear how to deal with enumerated attributes and missing values. These factors are critical in real-world datasets that are encountered in practice, and to take account of them the SDR is further modified to

$$\mathrm{SDR} = \frac{m}{|T|} \times \beta(i) \times \left[\mathrm{sd}(T) - \sum_{j \in \{L,R\}} \frac{|T_j|}{|T|} \times \mathrm{sd}(T_j) \right] \tag{3}$$

where m is the number of examples without missing values for that attribute, T is the set of examples that reach this node, $\beta(i)$ is the correction factor calculated for the original attribute to which this synthetic attribute corresponds, and T_L and T_R are sets that result from splitting on this attribute (all attributes are now binary).

In the present paper, the M5′ algorithm with LSE implemented in Weka software (1999–2005) was used.

M5-PLSR model trees

M5-PLSR MTs use the same SDR for tree induction, but use PLSR instead of LSER for multivariate regression at the leaves of the tree. The PLSR is a relatively recent technique that generalizes and combines features from principal component analysis (PCA) and multivariate linear regression (Abdi 2003). Because, instead of finding the hyperplanes of minimum variance, it finds a linear model describing some predicted variables in terms of other observable variables.

A goal of PLSR (or, more precisely, of the principal component analysis part in it) is to deduces orthogonal linear combinations of original predictors that correlate highly with the response variables, while accounting for as much variance in the predictors as possible. This means that the PLSR method balances the objectives of finding latent vectors that explain both the response and predictor variation. Therefore, the PLSR method is well suited to the prediction of regression models when the data is highly correlated, and where there is only a limited number of observations, because the predictor and predicted (response) variables are each considered as a block of variables (Rosipal & Krämer 2006).

PLSR models the relationship between response and predictor variables by means of latent variables. For a dataset with response variable $y \in R^{n \times m}$ and predictor variable $x \in R^{n \times p}$, the PLSR decomposes the variables x and y as follows:

$$\begin{aligned} X &= TP^{\mathrm{T}} + E = \sum_{h=1}^{a} t_h p_h^{\mathrm{T}} + E \\ Y &= UQ^{\mathrm{T}} + F = \sum_{h=1}^{a} u_h q_h^{\mathrm{T}} + F \end{aligned} \tag{4}$$

where T and U are matrices of the extracted score vectors (components, latent vectors), P and Q represent matrices of loadings, and E and F are the matrices of residuals.

The input and output variables are projected onto a subspace of orthogonal latent variables to give the input and output scores, t and u, respectively. The standard algorithm for computing PLS regression components is nonlinear iterative PLS (NIPLS). The NIPLS algorithm starts with a random initialization of the Y score vector u and repeats the sequence of steps below until convergence. After convergence, the loading vectors p and q can be computed. In summary, the NIPLS algorithm is as follows:

(1) set the output scores u equal to a column of Y
(2) compute the input weights w by regressing X on u: $w^{\mathrm{T}} = (u^{\mathrm{T}} \cdot X)/(u^{\mathrm{T}} \cdot u)$
(3) normalize w to unit length: $w = w/(\|w\|)$
(4) calculate the input scores t: $t = (X \cdot w)/(w^{\mathrm{T}} \cdot w)$
(5) compute the output loadings q by regressing Y on t: $q^{\mathrm{T}} = (t^{\mathrm{T}} \cdot Y)/(t^{\mathrm{T}} \cdot t)$
(6) normalize q to unit length: $q = q/(\|q\|)$
(7) calculate the new output scores u: $u = (Y \cdot q)/(q^{\mathrm{T}} \cdot q)$
(8) check the convergence on u–if 'yes' go to (9), otherwise go to (2)
(9) calculate the input loadings p by regressing X on t: $p^{\mathrm{T}} = (t^{\mathrm{T}} \cdot X)/(t^{\mathrm{T}} \cdot t)$
(10) compute the inner model regression coefficient b: $b = (t^{\mathrm{T}} \cdot u)/(t^{\mathrm{T}} \cdot t)$
(11) calculate the input residual matrix: $E = X - t \cdot p^{\mathrm{T}}$
(12) calculate the output residual matrix: $F = Y - b \cdot t \cdot q^{\mathrm{T}}$
(13) if additional PLS dimensions are necessary, replace X and Y by E and F, respectively, and repeat steps 1–13.

In the model prediction step, the prediction $\hat{Y}$ is calculated by the new input $\hat{X}$, and p^{T}, q^{T}, W^{T} and b are calculated in the model construction step.

For PLSR the N-way toolbox of Matlab was used (Andersson & Bro 2000).

ARTIFICIAL NEURAL NETWORK (ANN)

In recent decades, an ANN has been used in many real-world applications and offers an attractive paradigm for a broad range of adaptive complex systems. Their application has proven useful and has been successful in a wide variety of pattern recognition and feature extraction tasks.

Two kinds of ANN were used in this study for comparison with the M5 MTs: supervised backpropagation multilayer perceptron (MLP) and unsupervised feed-forward radial basis function network (RBF). Radial basis function (RBF) neural networks provide a powerful alternative to multilayer perceptron (MLP) neural networks to approximate or to classify a pattern set. RBF differs from MLP in that the overall input–output map is constructed from local contributions of Gaussian functions, requires fewer training samples and trains faster than MLP. The most widely used method to estimate centers and widths consist of using an unsupervised technique called a clustering rule. The centers of the clusters give the centers of the RBF, and the distance between the clusters provides the width of the Gaussians. As the audience of this journal should know the standard type of ANN, it is not described here.

k-nearest neighbor (*k*NN)

k-nearest neighbor is the most basic type of instance-based learning method (Mitchell 1997) which locates k spatial objects to a nearest given query point. In instance-based learning, training instances are stored and a distance function is used to determine which instance of the training set is closest to a new unknown instance (Witten & Frank 2005). The distance between two instances is defined to be d (x_i, y_i) which is an attribute of each instance with N features, such that $x = \{x_1 \cdot \cdots \cdot x_N\}$, $y = \{y_1 \cdot \cdots \cdot y_N\}$, where absolute distance measuring d_{A} is expressed by

$$d_{\mathrm{A}}(x_i, y_i) = \sum_{i=1}^{N} |x_i - y_i| \tag{5}$$

and Euclidean distance d_{E} is expressed by

$$d_{\mathrm{E}}(x_i, y_i) = \sum_{i=1}^{N} \sqrt{x_i^2 - y_i^2} \tag{6}$$

To find the closest instances, it is necessary to pass through the dataset, one instance at a time, and compare it to the query instance. We can represent the dataset as a matrix $D = N \times P$, containing P instances $s^1, \ldots, s^P$, where each instance s^i contains N features $s^i = \{s_1^i, \ldots\ldots, s_N^i\}$. A vector o with length P of output values $o = \{o^i, \ \ldots, o^P\}$ accompanies this matrix, listing the output values o^i for each instance s^i.

It should be noted that the vector o can also be seen as a column matrix: if multiple output values are desired, the width of the matrix may be expanded.

The *k*NN alogorithm consists of the following steps:

(1) store the output values of the M nearest neighbors to query instance q in vector $r = \{r^i, \ \ldots, r^M\}$ by repeating the following loop M times:

  a. go to the next instance s^i in the dataset, where i is the current iteration within the domain $\{1, \ \ldots, P\}$
  b. if q is not set or $q < d(q, s^i)$: $q \leftarrow d(q, s^i)$, $t \leftarrow o^i$
  c. loop until reaching the end of the dataset (i.e. $i = P$)
  d. store q into vector c and t into vector r

(2) calculate the arithmetic mean output $\bar{r}$ across r from

$$\bar{r} = \frac{1}{M} \sum_{i=1}^{M} r_i \tag{7}$$

(3) return $\bar{r}$ as the output value for the query instance q

In Weka software, the *k*NN algorithm is termed the “IBk classifier” (instance-based classifier with *k* neighbors).

CASE STUDY

Site description

The Yongdam dam is located upstream in the Keum River, which flows through the Midwest Korean Peninsula (Figure 2). The Yongdam reservoir was formed as part of the Yongdam impoundment for flood control (volume of flood control: 137 million m^3/yr), for water supply to Jeonju city (700,000 m^3/d with a planned increase to 1,050,000 m^3/d), and for power generation (24,400 kW). The principal morphometric feature of the Yongdam reservoir is that it is a narrow, deep reservoir with a surface area of about 37 km^2, a maximum width of 1 km and a maximum depth of about 70 m. The reservoir has a retention time of 318 d and is served by a catchment of about 928 km^2.

Korea is located in the East Asian monsoon belt. During winter, continental high-pressure air masses develop over Siberia, from which strong northwesterly winds bring dry, cold air over Korea. The summer monsoon brings abundant moisture from the ocean and produces heavy rainfall. About 70% of the annual rainfall occurs between June and September. The winter monsoon is dry, and produces low temperatures and little precipitation except for occasional snowfalls. Normally, less than 10% of the total annual precipitation falls in winter. This means that the pattern of precipitation in Korea is variable both seasonally and spatially, due to effects of the monsoon climate. The mean temperature at the research site during the hottest month (June) was 29.2°C, the mean temperature in the coldest month (January) was -4.9°C and the range (coldest to hottest months) was 34.1°C.

In 2005, the total recorded precipitation was 1,445.8 mm and the mean rainfall on days with precipitation was 10.0 mm in general ranging between 0.1 and 182.8 mm. The mean inflow to the Yongdam reservoir and the inflow range were 28.1 m^3/s and 0.1–1979.8 m^3/s, respectively, and the mean outflow from the reservoir and the outflow range were 31.4 m^3/s and 11.9–705.5 m^3/s, respectively.

During summer, when flood events occur, a massive quantity of non-point source pollutants diluted by precipitation is discharged into the reservoir. Unlike point source

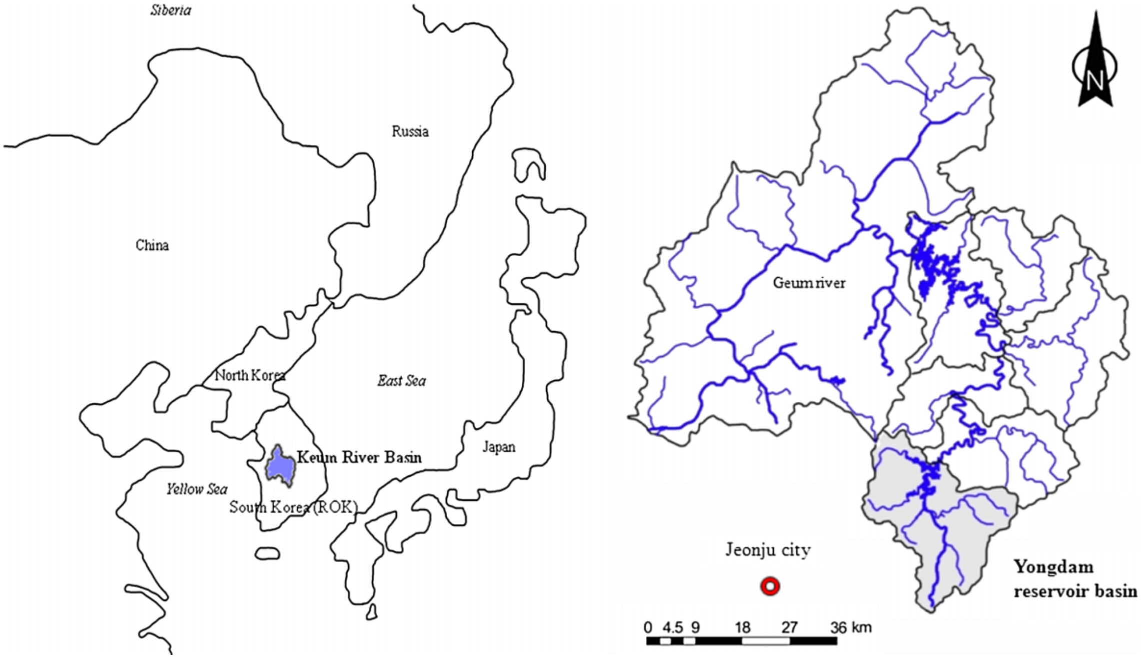


Figure 2 | Location of the Yongdam reservoir on the Keum River, Republic of Korea.

inputs, material from non-point sources is discharged with eroded sediment, resulting in an increase in the total amount of pollutants. Similarly, the eroded particulate matter discharged is in proportion to the flow rate, so that the load increases significantly during flood events. Soil erosion and sediment transport are common phenomena in Korea, and they are closely coupled with deterioration in water quality.

Data acquisition

In order to analyze the water quality of Yongdam reservoir, samples were collected at the nine monitoring stations (R2–R10 in Figure 3) at three depths within the reservoir during the three years (2005–2007). Samples were collected on 60 occasions: 23 in 2005, 23 in 2006 and 14 in 2007. Analysis of the collected samples was done on 16 water quality variables, including water temperature, total phosphorus (TP), inorganic phosphate (PO_4-P) and chlorophyll-a (Chl-a) representing the level of eutrophication, which were used in this study.

The shape of the reservoir meant that the sampling stations were located in three zones: the riverine zone (R2, R3, R8), the transition zone (R4, R5, R6, R7) and the lacustrine (lake) zone (R9, R10). However, the preferred sampling station for the modelling data should be stable and in a well-mixed zone, with respect to the reservoir characteristics, because the data derived from such a site would be expected to be more reliable (i.e. have lower error values) than from other sampling stations. The sampling station chosen was R9 in the lacustrine zone. This sampling station is also important, because it is located near the intake tower for water supply. Figure 4 shows the annual variations in TP and Chl-a concentration during the sampling period. Note that variation patterns (rise and fall) of TP and Chl-a concentration are not in accord with each other.

The water quality datasets had some limitations due to the small number of samples taken. The average number of samples is 2 each month due to a limited budget as well as the difficulty of accessing the site during the monsoon. The field observation datasets for 2005 and 2006 were used as the training data, and the dataset for 2007 was used for testing the trained models.

Variables like water temperature, pH, DO, TP, PO_4-P and Chl-a concentration are correlated with algal

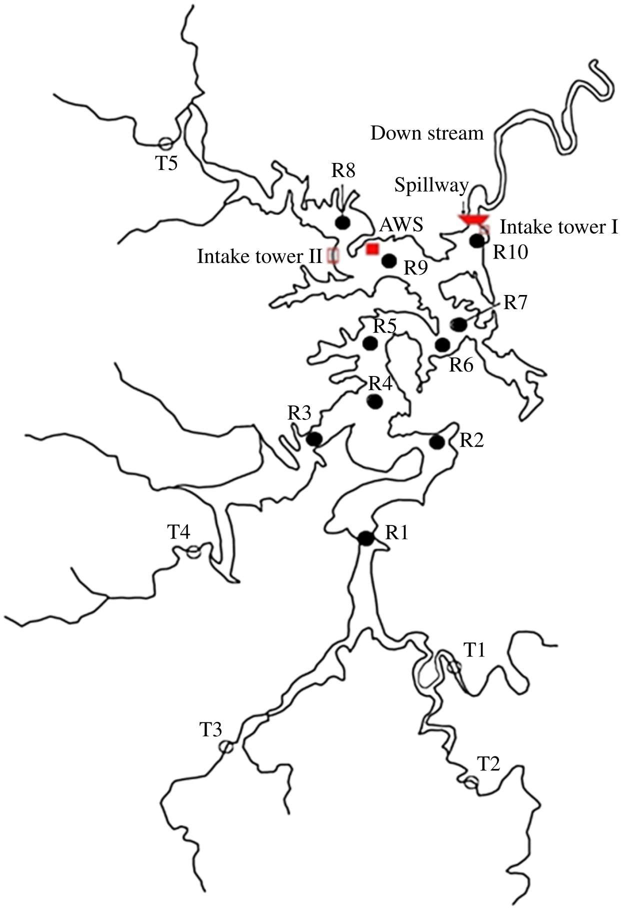


Figure 3 | Location of the sampling stations.

concentration, and were used to build a Chl-a prediction model. Because algal blooms change the pH and DO but these variables do not affect the emergence and growth of the algae, we developed a Chl-a prediction model based only on water temperature, TP, PO_4-P and Chl-a. These four variables are the parameters well known to be important in the development of a eutrophication model under limiting phosphate conditions. The season also influences the metabolism of algae. Therefore, the training and test datasets were partitioned into two groups: the algal growth period (May–November) and the no-growth period (December–April) in Korea. The modeling results using an annual dataset covering both periods were compared with those using the partitioned dataset.

In summary, the analyses involved in this study were the ten cases identified in Table 1.

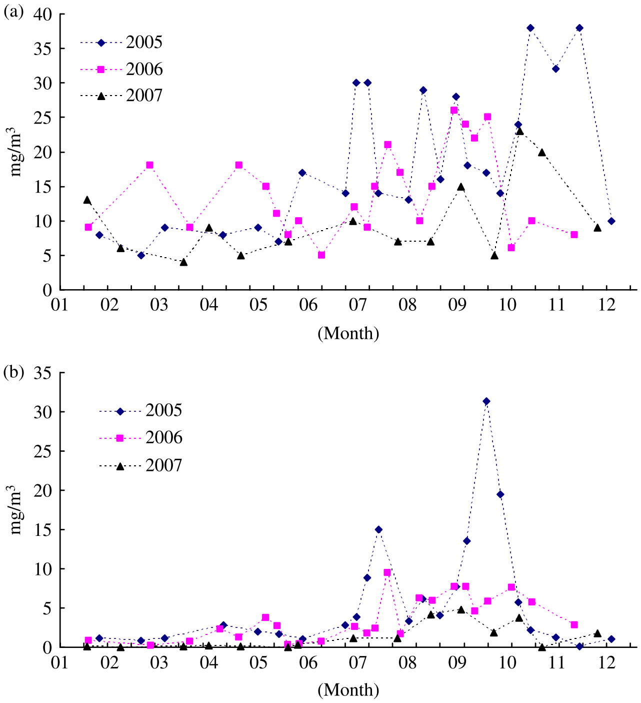


Figure 4 | Variations of TP and Chl-a concentrations at sampling station R9 for three years (2005–2007). (a) Total phosphorus. (b) Chlorophyll-a.

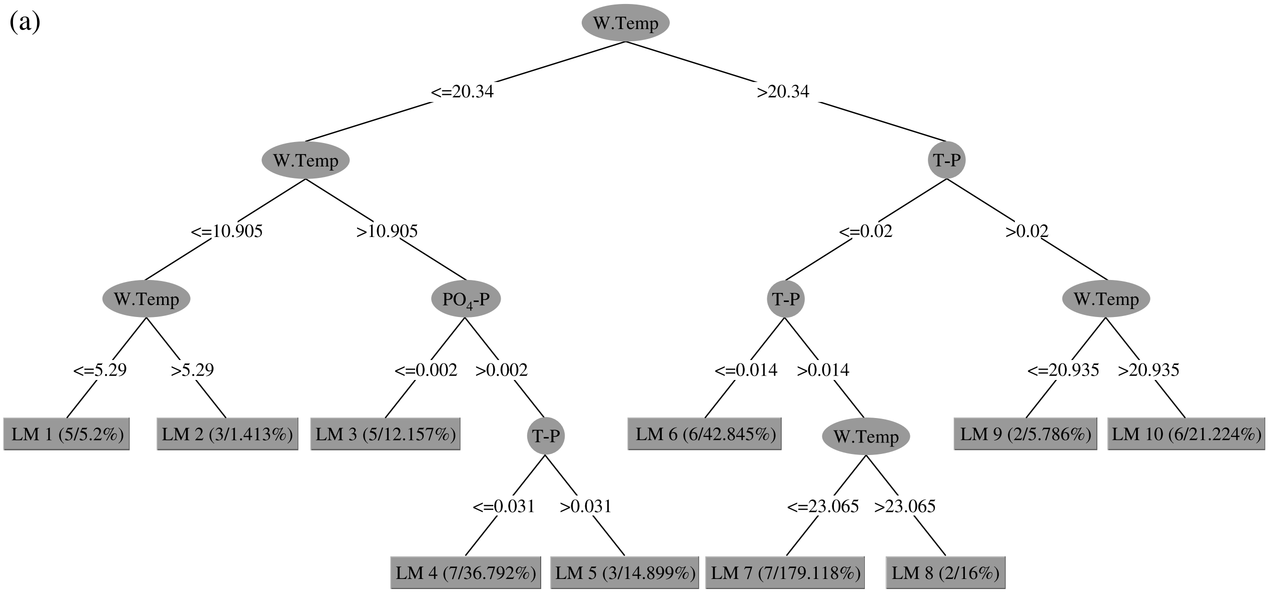
MODEL APPLICATION

Learning conditions

Modelers typically seek to develop techniques for finding and describing information in data as aids to explain the data and to make predictions from it. In the machine learning phase, each data-driven technique uses predictor variables (W. Temp., TP and PO_4-P) and response variable (Chl-a) for training; in the prediction phase, new instances of only predictor variables are fed into the trained model, and then the predicted values are obtained by inference procedures or from the models at the leaves.

Table 1 | Application cases for data-driven models according to annual and partitioned dataset

Annual dataset (January–December)		**Partitioned datasets (algal growth/no-growth period)**	
Case 1	ANN-MLP	Case 6	ANN-MLP
Case 2	ANN-RBF	Case 7	ANN-RBF
Case 3	*k*NN	Case 8	*k*NN
Case 4	M5′	Case 9	M5′
Case 5	M5 + PLSR	Case 10	M5 + PLSR



Tree structure

(b)

LM 1: Chl-a = 0.2147 × W.Temp – 0.5628	LM 6: Chl-a = 0.1528 × W.Temp + 178.8629 × T-P + 1.2346
LM 2: Chl-a = 0.2163 × W.Temp – 0.5554	LM 7: Chl-a = 0.1528 × W.Temp + 156.505 × T-P + 2.2642
LM 3: Chl-a = 0.1899 × W.Temp + 0.0081	LM 8: Chl-a = 0.1528 × W.Temp + 156.505 × T-P + 2.1297
LM 4: Chl-a = 0.1899 × W.Temp + 0.0803	LM 9: Chl-a = 0.1528 × W.Temp + 3.8093
LM 5: Chl-a = 0.1899 × W.Temp + 0.0294	LM 10: Chl-a = 0.1528 × W.Temp + 3.7242

Linear models

Figure 5 | Structure of model trees constructed by SDR and linear models at sampling station R9.

For the LSER model trees, M5′ MTs were built using Weka software. The minimum number of instances was set to 8 for the highest correlation coefficient between measured and predicted data, and the smoothed linear model option was applied for each interior node of the un-pruned tree (see Figure 5). For the M5-PLSR MTs presented in this paper, the minimum number of instances was set to 15, and the smoothed linear model option was applied for each interior node of the un-pruned tree.

ANN was built using the Weka software that uses the Levenberg–Marquardt algorithm. The number of epochs used for training was 700. A classifier of MLP-ANN uses back-propagation to classify instances. The number of hidden nodes was two, which was found by trial-and-error analysis of the ANN performance on the validation set (see Figure 6). The optimal number of clusters for RBF-ANN was nine in this study. ANN models were run on the same training and test datasets as for the MTs.

For *k*NN, three instance-based neighbors and the Euclidean distance metrics were used; the outputs of the neighbors were additionally weighted by the inverse distance to give higher weights to the closest neighbors.

Methods of error analysis

The error analysis methods applied in this study were the correlation coefficient (CC), root mean square error (RMSE) and root mean relative error (RRSE). The details of their applications are explained below.

The correlation coefficient (CC) measures the strength and direction of a linear relationship between two random variables (the predicted and observed values). The CCs were

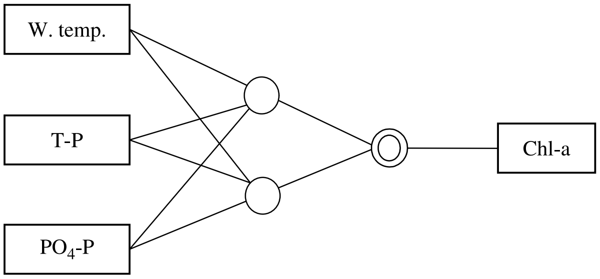


Figure 6 | Structure of ANN with two hidden layers applied at sampling station R9.

obtained by dividing the covariance of the two variables by the product of their standard deviations. The CC ranges from -1 to 1, with negative values indicating that the observed and predicted values tend to vary inversely. It should be noted that, even if the correlation is close to 1, the predicted and observed values may not be similar, but only tend to vary in a similar way.

The root mean square error (RMSE) measures the discrepancies between predicted and observed values. To calculate the RMSE the individual errors are squared, added together, divided by the number of individual errors and the square root of the resulting value is determined (Equation (8)):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)^2} \qquad (8)$$

where n = number of observations; $O_i = i$th of n observations; $P_i = i$th of n predictions; $\bar{O}$ and $\bar{P}$ = observation and prediction averages, respectively. Values near zero indicate a close match. The RMSE overcomes the shortcoming of the average error by considering the magnitude rather than the sign of each discrepancy.

The root relative square error (RRSE) is relative to what it would have been if an observation had been used. Thus, the relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the observation. By taking the square root of the relative error, one reduces the error to the same dimensions as the quantity being predicted:

$$\mathrm{RRSE} = \sqrt{\sum_{i=1}^{n}(P_i - O_i)^2 / \sum_{i=1}^{n}(O_i - \bar{O})^2}. \qquad (9)$$

The RRSE index ranges from 0 to infinity, with 0 corresponding to the ideal situation. The RRSE exaggerates situations where the prediction error is significantly greater than the mean error. Consequently, in spite of the data fluctuation, the RRSE explains the normalized relative error levels.

RESULTS AND DISCUSSION

The prediction results for the M5′ MTs and M5 MTs using PLSR knowledge inferences (M5-PLSR MTs), and the MLP and RBF ANN, are shown in Figure 7(a, b). It shows quite different results for each modelling experiment.

The modelling results using two groups of data were compared using the three error measures (see Table 2). Table 2 shows that the CCs have quite high values except for Case 8. Therefore, all models except for *k*NN show a fair correlation between predicted and measured values of Chl-a concentrations. Although Cases 5 and 6 have relatively high CC values, the RMSE and RRSE values are greater rather than the others. This means that the predictions are relatively distant from the measured values. What is important in the statistical error analysis is the imprecision that is intrinsic in human cognition. There are several outliers in terms of goodness-of-fit; however,

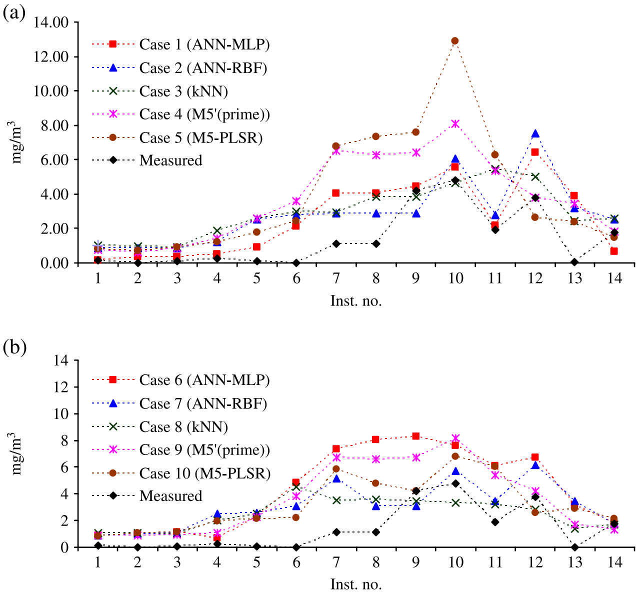


Figure 7 | Chl-a concentrations at sampling station R9 predicted using data-driven models: M5′ MTs, PLSR-M5 MTs, MLP-ANNs, RBF-ANNs and *k*NN. (a) Annual dataset. (b) Partitioned dataset.

Table 2 | Results of error analysis on each case: CC, RMSE and RRSE

	Error terms		
Cases	**CC**	**RMSE**	**RRSE**
Case 1	0.74	1.78	1.09
Case 2	0.75	2.67	1.63
Case 3	0.74	1.65	1.01
Case 4	0.69	2.91	1.77
Case 5	0.73	3.63	2.21
Case 6	0.71	3.56	2.17
Case 7	0.69	2.16	1.31
Case 8	0.43	1.91	1.16
Case 9	0.71	2.87	1.75
Case 10	0.61	2.39	1.46

there is always a possibility that in the natural environment outliers exist.

Based on Figure 7, ANNs and the *k*NN algorithm showed better results on the annual dataset than on the partitioned ones. RBF-ANN for Case 2 shows a better result than those of MLP-ANN and other algorithms. The reason may be that it uses a clustering rule partitioning the annual dataset into nine clusters bearing a stronger resemblance. However, RBF-ANN for Case 7 using the partitioned dataset became worse than Case 2. This may be because of over-fitting. The M5′ MT algorithm showed similar results on both sets. M5-PLSR MTs showed unfair binary splitting on the annual dataset. The reason is that the binary splitting rule compares the SDR between the values of only one attribute in spite of other attributes in each instance. Eco-environmental data show irregular variations between attributes of each instance like the non-linearity between TP and Chl-a concentrations shown in Figure 4. In contrast, M5-PLSR MTs gave a little better result on the partitioned dataset for Case 10 than on the annual one for Case 5. This is explained by the partitioning (clustering) of the annual dataset into the two datasets of the algal growth and no-growth periods brought some instances to bear a stronger resemblance to each other than the annual dataset. Like that, the use of the PLSR algorithm, which identifies the latent vector that explains variations in both response and predictor variables on the partitioned dataset than variations in both response and predictor variables on the annual dataset.

This was not observed when using ANN, *k*NN and M5′. ANN and *k*NN show similar or worse predictions when using the partitioned dataset. This means that ANN and *k*NN show certain limitations in modeling of water quality related to biological processes, especially for high dimensional data. For this reason, many studies have examined simple relationships between Chl-a and DO or pH (e.g. Schladow & Hamilton 1997; Heiskary & Markus 2001) for use in algal growth prediction models. The reason for this is that DO and pH are dependent on algal growth and decay, and can be easily measured.

Eco-environmental processes are often affected by unknown factors such as retention times, reservoir flow and/or mixing conditions (Lawrence *et al.* 2000). These multiple factors affect eutrophication processes in the reservoir, being a second- or third-order activity, which was not the case in the laboratory experiment. This indicates that data-driven models built on eco-environmental data affected by these natural processes should be evolved into various algorithms.

Although M5′ and M5-PLSR MTs for Cases 4 and 5 show relatively greater errors than those of other algorithms, they have the advantage of generating transparent models as shown in Figure 5. M5′ and M5-PLSR MTs have improved prediction with the partitioned dataset. This means that M5-PLSR MTs will show better prediction using more closely correlated multivariate datasets. Clustering can be used to group data that seem to fall similarly together.

CONCLUSION AND FUTURE RESEARCH

A reliable prediction model using data-driven modeling techniques typically needs a considerable amount of data. As eco-environmental data have a periodic cycle and time lags occur between nutrient uptake by algae and subsequent growth, it may be meaningless to seek relationships between algal growth and water quality parameters using bivariate analysis without more sophisticated analysis techniques of time series data. However, it is difficult to acquire adequate time series data in reality. In most eco-environmental research, the number of sampling occasions generally does not exceed 30 per year, which is the minimum number for ensuring a normal distribution in a dataset (Pentecost 1999). In the present study, the number of instances considered was 46 for the training dataset and 14 for the test dataset for three years, from 2005–2007. To overcome this small dataset size, the present study implemented M5 MTs with the PLSR algorithm, and in the future, the additional input data, each year, can improve the prediction efficiency of this model.

As the number of predictors in the LSE regression rapidly decreases as the tree expands, the M5 algorithm based on the PLS regression can help overcome the effects of insufficient data for cases involving new or recently established reservoirs. In this study, the use of the M5 MTs with the PLSR algorithm gave better results than the M5′ in the case of the partitioned dataset, probably because the former has a function identifying the latent vectors, which explain variations in the response and predictor variables in algal growth prediction from the instance-based partitioned dataset.

There have been reported only a few similar applications of a multivariate (at least four variables) tree structure and knowledge inference for predicting algal growth in water quality management research (Kompare *et al.* 1997*a*,*b*; Dzěroski 2001). The MTs have many advantages over other data-driven models, including a more explicit tree structure involving classification rules and linear models at the leaves. However, for the MT algorithm for eco-environmental data with more than three water quality variables, it may be recommended to use a clustering rule instead of binary splitting based on the standard deviation reduction for tree generation. As the PLSR algorism seeks latent vectors among response and predictor variables, a minimum number of partitions reflecting seasonal variations will show better results. Note that if there are many leaves in the trees in the case of small datasets, PLSR can be unreliable and result in negative coefficients and consequently in negative predicted values.

As eco-environmental data have a periodic cycle, it should be emphasized that implementation of long-term monitoring strategies is important for providing adequate datasets, such as the more partitioned dataset according to environmental conditions in this study. The nonlinear composite model of M5-PLSR MTs in this study will contribute to development of a decision support tool for the management of the reservoir water quality and/or as a predictor of environmental processes.

Finally, the MTs-PLSR technique is a promising approach when there is a shortage of data required to build a more transparent learning process model, and the combination with the clustering technique is recommended for follow-up research.

ACKNOWLEDGEMENTS

The authors acknowledge help from Dr. J. K. Shin (Korea Water Resources Corporation) and thank Professor M. G. Chun (Chungbuk National University, Cheongju, Korea) for providing the water quality data and the PLS algorithm.

REFERENCES

Abdi, H. 2003 Partial least squares (PLS) regression. In *Encyclopedia of Social Sciences Research Methods* (eds. M. Lewis-Beck, A. Bryman & T. Futing), pp. 1–17. Sage, Thousand Oaks, CA.

Andersson, C. A. & Bro, R. 2000 The N-way toolbox for MATLAB. *Chemomet. Intell. Lab. Syst.* **52**, 1–4.

Baffi, G., Martin, E. B. & Morris, A. J. 1999 Non-linear projection to latent structures revisited (the neural network PLS algorithm). *Comput. Chem. Eng.* **23**, 1293–1307.

Dzěroski, S. 2001 Applications of symbolic machine learning to ecological modelling. *Ecol. Modell.* **146**, 263–273.

Friedman, H. 1991 Multivariate adaptive regression splines. *Ann. Stat.* **19**, 1–141.

Heiskary, S. & Markus, H. 2001 Establishing relationships among nutrient concentrations, phytoplankton abundance, and biochemical oxygen demand in Minnesota, USA rivers. *J. Lake Reserv. Manage.* **17** (4), 251–262.

Kompare, B., Dzžroski, S. & Karalič, A. 1997*a* Identification of the Lake of Bled ecosystem with the artificial intelligence tools M5 and FORS. In *Proceedings of the 4th International Conference on Water Pollution*. Computational Mechanics Publications, Southampton, pp. 789–798.

Kompare, B., Dzžroski, S. & Križman, V. 1997*b* Modeling the growth of algae in the Lagoon of Venice with the artificial intelligence tool GoldHorn. In *Proceedings of the 4th International Conference on Water Pollution*. Computational Mechanics Publications, Southampton, pp. 799–808.

Lawrence, I., Bormans, M., Oliver, R., Ransom, G., Sherman, B., Ford, P. & Wasson, B. 2000 *Physical and Nutrient Factors Controlling Algal Succession and Biomass in Burrinjuck Reservoir*. Technical Report, January, CRCFE, Canberra, Canada.

Mitchell, T. M. 1997 *Machine Learning*. McGraw-Hill, New York.

Pedrycz, W. & Sosnowski, Z. A. 2001 The design of decision trees in the framework of granular data and their application to software quality models. *Fuzzy Sets Syst.* **123**, 271–290.

Pedrycz, W., Succi, G. & Chun, M. G. 2002 Association analysis of software measures. *Int. J. Softw. Eng. Knowl. Eng.* **12** (3), 291–316.

Pentecost, A. 1999 *Analysing Environmental Data*. Addison Wesley Longman, Singapore.

Quinlan, J. R. 1992 Learning with continuous classes. In *Proc. AI'92* (eds. A. Adams & L. Sterling), pp. 343–348. World Scientific, Singapore.

Rosipal, R. & Krämer, N. 2006 Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection (SLSFS) 2005, Lecture Notes in Computer Science (LNCS) 3940*. Springer-Verlag, Berlin, pp. 34–51.

Solomatine, D. P. & Dulal, K. N. 2003 Model trees as an alternative to neural networks in rainfall-runoff modeling. *Hydro. Sci. J.* **48** (3), 399–411.

Schladow, S. G. & Hamilton, D. P. 1997 Prediction of water quality in lakes and reservoirs: part II–model calibration, sensitivity analysis and application. *Ecol. Modell.* **96**, 111–123.

Solomatine, D. P. 2005 *Data-Driven Modelling and Computational Intelligence Methods in Hydrology* (ed. M. G. Andersen),. *Encyclopedia of Hydrological Sciences*, Vol. 1, pp. 3–22. John Wiley & Sons, New York.

Solomatine, D. P., Maskey, M. & Shrestha, D. L. 2007 Instance-based learning compared to other data-driven methods in hydrologic forecasting. *Hydro. Process.* **21** (2), 275–287.

Tan, Y., Shi, L., Tong, W., Gene Hwang, G. T. & Wang, C. 2004 Multi-class tumor classification by discriminant partial least squares using microarray gene expression data and assessment of classification models. *Comput. Biol. Chem.* **28**, 235–244.

Wang, Y. & Witten, I. H. 1997 Inducing model trees for continuous classes. In *Proceedings of the Poster Papers of the 9th European Conference on Machine Learning (ECML 97)* (eds. M. van Someren & G. Widmer), pp. 128–137. Prague, Czech Republic.

Weka Software 1999–2005 *Environment for Knowledge Analysis, Version 3.4.7*. University of Waikato, New Zealand.

Witten, I. H. & Frank, E. 2005 *Data Mining–Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco.

First received 11 December 2008; accepted in revised form 14 April 2009. Available online 21 November 2009