Data-driven modelling: some past experiences and new approaches

Dimitri P. Solomatine and Avi Ostfeld

ABSTRACT

Physically based (process) models based on mathematical descriptions of water motion are widely used in river basin management. During the last decade the so-called data-driven models are becoming more and more common. These models rely upon the methods of computational intelligence and machine learning, and thus assume the presence of a considerable amount of data describing the modelled system's physics (i.e. hydraulic and/or hydrologic phenomena). This paper is a preface to the special issue on Data Driven Modelling and Evolutionary Optimization for River Basin Management, and presents a brief overview of the most popular techniques and some of the experiences of the authors in data-driven modelling relevant to river basin management. It also identifies the current trends and common pitfalls, provides some examples of successful applications and mentions the research challenges.

Key words | computational intelligence, data-driven modelling, neural networks, river basin management, simulation modelling

Dimitri P. Solomatine (corresponding author)
UNESCO-IHE Institute for Water Education,
PO Box 3015, 2601 DA, Delft,
The Netherlands
E-mail: *d.solomatine@unesco-ihe.org*

Avi Ostfeld
Faculty of Civil and Environmental Engineering,
Technion-Israel Institute of Technology,
Haifa, 32000,
Israel

INTRODUCTION

Modern river basin management is impossible without adequate hydraulic and hydrologic models – used in different tasks from scenario analysis to real-time forecasting (Falconer *et al.* 2005). Numerous papers have been published on using a physically based approach for modelling behaviour of river basins, as well as the ways to classify them. Traditionally the river basin (watershed) was treated as a lumped, time-invariant, linear, deterministic system, resulting in the unit hydrograph theory (Sherman 1932), upon which the Nash linear cascade of reservoirs model (Nash 1957) and the parallel cascade of reservoirs of Diskin (1964) were built. Other models within this category are attributed to Dooge (1959), Diskin & Boneh (1975), Eagleson *et al.* (1966) and others. Later models have expanded the lumped linear deterministic approach to a distributed linear cell approach in which the entire river basin is partitioned into a tree-like structure built of cells, with each cell being a sub-watershed (Diskin *et al.* 1984). Nowadays the models based on partial differential equations incorporate sophisticated solvers and are encapsulated into modelling environments with advanced interfaces and visualisation tools (Abbott 1991; Falconer *et al.* 2005). They vary in complexity and orientation at different tasks from general river basin planning like RIBASIM (2006) to models able to simulate the entire land phase of the hydrologic cycle like MIKE SHE (2006).

During the last 10–15 years, the advances in ICT brought the new tools enhancing data acquisition, data analysis and visualisation; such advances are often associated with Hydroinformatics. A Geographical Information System (GIS) connected to remote sensing tools stepped in for watershed management, providing numerous tools to support modelling. A GIS-based hydrological model couples the descriptions of hydrological features on a spatial scale with the predictive power of models. Several examples of these can be mentioned: SWAT – a river basin scale model developed to quantify the impact of land management practices in large, complex watersheds

doi: 10.2166/hydro.2008.015

(SWAT 2006); BASINS – a multipurpose environmental analysis system for performing watershed and water-quality-based studies, and AVGWLF – a spatial distributed watershed model based on GWLF (Haith & Shoemaker 1987) for simulating runoff, sediment and nutrient loadings from a watershed, given variable-size source areas. The relatively inexpensive satellite technology of today permits using various types of remote sensing data, and associated data analysis and pattern recognition techniques, for water resources management. Hydraulic and hydrologic models are being more and more complemented by the *data-driven models* which are the subject of this paper. Such integrated systems, coupled with GIS and animation tools, being incorporated into social and managerial environments, are often referred to as Hydroinformatics systems, and form powerful data management and modelling instrumentation for water managers and decision-makers.

In order to make a step towards the understanding of data-driven models, it is useful to provide a classification of models for river basin management. The following types of models can be distinguished:

(1) a *physically based* (*process*) model based on the description of the behaviour, typically based on the first-order principles from physics, of a phenomenon or system (also called knowledge-driven or simulation models). In river hydraulics, these are the 1D or 2D hydrodynamic models, and in hydrology the lumped conceptual models or distributed physically based models;

(2) an *empirical*, or *data-driven* (*DD*) model involving mathematical equations assessed not from the physical process in the river basin but from analysis of concurrent input and output time series. Typical examples here are the rating curves, unit hydrograph method and various statistical models (linear regression, multi-linear, ARIMA) and methods of machine learning discussed later.

Data-driven modelling (DDM) is based on the analysis of the data characterising the system under study. A model can then be defined on the basis of connections between the system state variables (input, internal and output variables) with only a limited number of assumptions about the "physical" behaviour of the system. The contemporary methods can go much further than the ones used in conventional empirical modelling in hydraulic engineering and hydrology. They allow for solving numerical prediction problems, reconstructing highly nonlinear functions, performing classification, grouping of data and building rule-based systems.

It should be noted that there is still a certain scepticism about DDM among many hydrologists and water resources specialists. They view the induction of models from datasets as a computational exercise, because in their opinion the derivation is not related to physical principles and mathematical reasoning (See *et al.* 2007). Another issue is the necessity of using sophisticated data-driven models: are they actually needed when traditional statistical models (typically linear regression or ARIMA-class models) are, in many cases, accurate enough? Some of the concerns of this nature are presented, for example, by Gaume & Gosset (2003) and Han *et al.* (2007). In their excellent recent paper, Abrahart & See (2007) address some of these problems and demonstrate that the existing nonlinear hydrological relationships, which are so important when building flow forecasting models for river basin management, are effectively captured by a neural network, the most widely used DDM method. This discussion about what model is the best may continue for a while, but in our view it is important to stress that there are always situations when one model type cannot be applied or suffers from inadequacies and can be well complemented or replaced by another one.

DDM is a common topic of research in the framework of Hydroinformatics (Abbott 1991), and, subsequently, is an important topic at the International Conferences on Hydroinformatics, European Geosciences Union (sub-division on Hydroinformatics), and at other conferences related to water management. During the last decade the number of researchers active in this area has considerably increased, so did the number of publications, and naturally they have the tendency of clustering in the form of volumes or special issues of the journals. An example is this special issue of the *Journal of Hydroinformatics*. Other examples include the edited volume to be published by Springer (Abrahart *et al.* 2008), recent special issues of the *Hydrological Sciences Journal* (2007), *Hydrology and Earth System Sciences* (Abrahart *et al.* 2007*b*) and the *Neural Networks Journal* (Cherkassky *et al.* 2006) where some of

the challenges of DDM that are very relevant for the purpose of this paper are discussed.

This paper presents a general overview and some of the experiences of the authors in data-driven modelling relevant to river basin management as a preface to this special issue on Data Driven Modelling and Evolutionary Optimization for River Basin Management. It also identifies the current trends and common pitfalls, mentions challenges and provides some examples of successful applications.

ESSENCE OF DATA-DRIVEN MODELLING

Definitions

There are a number of (overlapping) areas contributing to DDM: data mining, knowledge discovery in databases, computational intelligence, machine learning, intelligent data analysis, soft computing and pattern recognition. Computational intelligence (CI) incorporates three large areas: neural networks, fuzzy systems and evolutionary computing. Soft computing (SC) is the area that emerged from fuzzy logic, but currently also incorporates many techniques of CI. Machine learning (ML) is an area of computer science that was for a long time considered a sub-area of artificial intelligence (AI) that concentrates on the theoretical foundations of learning from data. Data mining (DM) and knowledge discovery in databases (KDD) used ML methods and are focused typically at very large databases and are associated with applications in banking, financial services and customer resources management.

Data-driven modelling can thus be considered as an approach to modelling that focuses on using the CI (particularly ML) methods in building models that would complement or replace the "knowledge-driven" models describing physical behaviour. DDM uses the methods developed in the fields mentioned above, and the role of a modeller is to tune them to a particular application area. "Modelling" in the name stresses the fact that this activity is close in its objectives to traditional approaches to modelling, and follows the traditionally accepted modelling steps, and that it does not comprise the analysis or mining of data only. Examples of the most common methods used in data-driven modelling of river basin systems are: statistical methods, artificial neural networks and fuzzy rule-based systems.

It is important to note that a component of CI, evolutionary and genetic algorithms (GA), is primarily oriented towards optimisation that can be used in model calibration and model structure optimisation (Savic 2005; Ostfeld & Preis 2005), or in traditional water resources optimization problems like (multi-objective) reservoir optimisation (Kim *et al.* 2006: in this study a popular multi-objective genetic algorithm (NSGA-II) was used).

The main part of data-driven modelling is, in fact, *learning* which incorporates the so-far unknown mappings (or dependencies) between a system's inputs and its outputs from the available data (Mitchell 1997, Figure 1). By data we understand the known samples that are combinations of inputs and corresponding outputs. As such, a dependence (viz. mapping or "model") is discovered (induced), which can be used to operation predict (or effectively *deduce*) the future system's outputs from the known input values.

By data we usually understand a set K of examples (or instances) represented by the duple $\langle \mathbf{x}_k, \mathbf{y}_k \rangle$, where $k = 1, \ldots, K$, vector $\mathbf{x}_k = \{x_1, \ldots, x_n\}_k$, vector $\mathbf{y}_k = \{y_1, \ldots, y_m\}_k$, $n =$ number of inputs and $m =$ number of outputs. The process of building a function (or "mapping", or "model") $\mathbf{y} = f(\mathbf{x})$ is called *training*. Very often only one output is considered, so $m = 1$.

In the context of river basin modelling the inputs and outputs are typically real numbers ($\mathbf{x}_k$, $\mathbf{y}_k \in \mathcal{R}^n$), so the main learning problem solved is numerical prediction (regression). Sometimes the problems of clustering and classification are solved as well (see, for example, Hall & Minns 1999; Hannah *et al.* 2000; Harris *et al.* 2000).

The process of building a data-driven model follows general principles adopted in modelling: study the problem – collect data – select model structure – build the model – test the model and (possibly) iterate. In DDM, not only the model parameters, but also the model structure, is often subject to

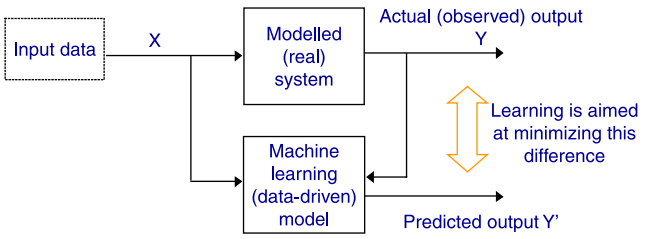

Figure 1 | Learning in data-driven modelling.

optimisation. Generally, following the so-called Occam's razor principle (Mitchell 1997), parsimonious models are valued (as simple as possible, but no simpler). An example of such a parsimonious model could be a linear regression model vs a nonlinear one, or a neural network with a small number of hidden nodes. Such models can be built by the deliberate use of the so-called *regularisation* when the objective function representing the overall model performance includes not only the model error term but also a term that increases in value with the increase of model complexity represented, e.g. by the number terms in the equation or the number of hidden nodes in a neural network.

It is worth mentioning that DDM is sometimes used to build models of models (replicating, for example, physically based models such as 1D hydrodynamic models) rather than models of natural systems; such models are often referred to as *surrogate*, *emulation* or *meta-models* (see, e.g., Solomatine & Torres 1996; Khu *et al.* 2004).

Use of data: methodological issues and trivial pitfalls

There are a couple of methodological issues related to the use of data for building a DDM. They may be considered trivial by experts in machine learning but are not always appreciated by hydraulic engineers or hydrologists building or using such models.

After the model is trained but before it is put into operation, it has to be tested (or verified) by some form of error measurement (e.g. root mean squared error) on the *test dataset*. To test the model during training yet another data set is needed – the *cross-validation* set. As a model gradually improves as a result of the training process, the error on the training data will be decreasing, but the cross-validation error will first be decreasing, but then will start to increase (effect of *overfitting*), so training should be stopped when the error on the cross-validation dataset starts to increase. If these principles are respected, then there is a hope that the model will *generalise* well, that is its prediction error on unseen data will be small.

Note that in an important class of machine learning models – support vector machines – a different approach is taken: it is to build the model that would have the best generalisation ability possible without relying explicitly on the cross-validation set (Vapnik 1998).

In connection to the issues covered above, there are two common pitfalls, especially characteristic of DDM applications where time series are involved, that are worth mentioning herein.

(1) The first pitfall relates to the construction of the three mentioned datasets on the basis of available data. The three sets should be statistically similar, i.e. should have similar distributions or, at least, similar ranges, mean and variance. This can be achieved by careful selection of examples for each dataset to ensure such statistical similarity, by random sampling data from the whole dataset, or employing an optimisation procedure resulting in sets with predefined properties (Bowden *et al.* 2002). A popular approach leading to approximate statistical similarity of training and cross-validation sets is to use the ten-fold validation method when a model is built ten times, trained each time on 9/10th of the whole set of available data and validated on 1/10th (number of runs is not necessarily ten). An extreme version of this method is the "leave-one-out" method when K models are build using K-1 examples and not using one (every time different). The resulting model is either one of the models trained, or an ensemble of all the models built, possibly with the weighted outputs. Note that for generation of the statistically similar training data sets for building a series of similar but different models, one should typically rely on the well-developed statistical resampling methods like the bootstrap originated by B. Tibshirani in the 1970s (see Efron & Tibshirani 1993) where (in its basic form) K data is randomly selected from K original data.The problem is that, if one of these procedures is followed, the data will not always be contiguous, so that, for example, it would not be possible to visualise a hydrograph when the model is fed with the test set. There is nothing wrong with such a model if the "time structure" of all the datasets is preserved. Such models, however, are reluctantly accepted by practitioners since they are so different from the traditional physically based models that always generate contiguous time series. A solution in such a situation is to group the data into hydrological events (i.e. contiguous blocks of data) and to try to ensure the presence of similar events in all the three datasets.

(2) Another pitfall can be encountered when a modeller tries to optimise the model structure (e.g. number of hidden nodes in an ANN) by using the test set. This is, of course, methodologically incorrect since the role of the test set is to judge the final model performance in operation. Our experience is, however, that this principle is not always respected even by experienced modellers; we could refer to one of the recent international competitions of data-driven models of water systems where both training and test sets were given to the contestants (with the best intentions of the organisers), but indeed inevitably pushing some of them to use the test set to optimize the performance of their model.

A question is, of course, what to do if the dataset is not large enough to allow for building all three sets of substantial size. Often the modellers choose not to build a cross-validation set at all, with the hope that the model trained on the training set would perform well on the test set as well. Another option is to perform ten-fold cross-validation.

Data preparation and choice of input variables

In any modelling exercise an important issue is data preparation and the choice of such variables that would be able to represent the modelled system in a best possible way.

An excellent reference to the first issue is the book by Pyle (1999). We cannot provide here the details but one thing has to be stressed: researchers excited by the power of the new modelling techniques often are not spending enough effort on proper data preparation.

An interesting study of the influence of different data transformation methods (linear, logarithmic and seasonal transformations, histogram equalization and a transformation to normality) was undertaken by Bowden *et al.* (2003). They found that the model using the linear transformation resulted in the lowest RMSE and more complex transformations did not improve the model (note, however, that the study is based only on one case study to forecast salinity in a river in Australia 14 days in advance). Our own experience shows that it is sometimes useful to apply the smoothing filters to the hydrological time series.

Choice of variables is an important subject and some studies suffer from the lack of relevant analysis. Apart from the expert judgement and visual inspection, there are formal methods that help in making this choice more justified, and the reader can be directed to the paper by Bowden *et al.* (2005) for an overview of these. Our own experience with using the Average Mutual Information (Solomatine & Dulal 2003) show that this simple and reliable method can help in selection of relevant input variables.

It is our hope that the adequate data preparation and the rational and formalized choice of variables will become a standard part of any modelling study.

POPULAR METHODS AND TYPICAL APPLICATIONS

Most engineering or water management problems are formulated as prediction of real-valued variables; this is a *regression problem* (not to confuse with linear regression, a particular case of regression). Machine learning aims at finding a function that would best approximate some data and this prompts for the use of the corresponding methods already available like linear regression, polynomial functions like splines or orthogonal polynomial functions. Most of the data-driven models use combinations of many simple functions. In essence, training aims at optimising the number of these functions and the values of their parameters (given the functions' class).

Multilayer perceptron (MLP) is a typical example of an artificial neural network (ANN) (Haykin 1999). It consists of several layers of mutually interconnected nodes (neurons), each of which receives several inputs, calculates the weighted sum of them and then passes the result to a non-linear "squashing" function. In this way the inputs to a MLP model are subjected to a multi-parameter nonlinear transformation so that the resulting model is able to approximate complex input–output relationships. Training of MLP is, in fact, solving the problem of minimising the model error (typically, mean squared error) by determining the optimal set of weights.

As the principle of backpropagation for training of MLPs was found and perfected in the 1970–80s (Werbos 1994), this type of ANN has become the most popular machine learning tool. Various types of ANNs are widely used for prediction and classification.

Note that backpropagation is a principle that made it possible to use gradient-based methods for MLP training, and that permits the usage of various optimization algorithms – from the simplistic versions of steepest descent

schemes to much more effective methods like conjugate gradient or Broydon–Fletcher–Goldfarb–Shanno (BFGS) methods (Press *et al.* 2007). In this respect we believe it is not fully correct to name MLP a "backpropagation network" since it can be trained using various methods, for example, direct search methods like GA. The use of these less efficient (much slower) algorithms is justified when gradient-based backpropagation training prematurely converges to the local optimum.

MLP ANNs are known to have several dozens of successful applications in river basin management and related problems, for example:

- Modelling rainfall–runoff processes: Hsu *et al.* (1995); Minns & Hall (1996); Dawson & Wilby (1998); Dibike *et al.* (1999); Abrahart & See (2000); Govindaraju & Ramachandra Rao (2001); Hu *et al.* (2007); Abrahart *et al.* (2007*a*);
- Building an ANN-based intelligent controller for real-time control of water levels in the channels of polders (Lobbrecht & Solomatine 1999);
- Modelling river stage-discharge relationships (Sudheer & Jain 2003; Bhattacharya & Solomatine 2005);
- Building a surrogate (emulation, meta-) model for:
  - replicating the behaviour of hydrodynamic and hydrological models of a river basin where ANNs are used in model-based optimal control of a reservoir (Solomatine & Torres 1996);
  - building an assisting surrogate model in calibration of a rainfall–runoff model (Khu *et al.* 2004);
  - emulating by an MLP network and replacing the hydrologic simulation component of multi-objective decision support model for watershed management (Muleta & Nicklow 2004). In this study an alternative to the backpropagation training was used – a direct search method (evolutionary algorithm) that reportedly allowed for avoiding local minima during training.

Most theoretical problems related to MLP have been solved and it should be seen as a quite reliable, well-understood method.

Radial basis functions (RBF) could be seen as a sensible alternative to the use of complex polynomials. The idea is to approximate some function $y = f(\mathbf{x})$ by a superposition of J functions $F(\mathbf{x}, \sigma)$, where σ is a parameter characterising the span, or "width", of the function in the input space. Functions F are typically "bell-shaped" (e.g. a Gaussian function) so that they are defined in the proximity to some "representative" locations (centres) $\mathbf{w}_j$ in n-dimensional input space and their values are close to zero far from these centres. The aim of learning here is in finding the positions of centres $\mathbf{w}_j$ and the parameters of the functions $f(\boldsymbol{x})$. This can be accomplished by building a *radial-basis function neural network*; its training allows identifying these unknown parameters. The centres $\mathbf{w}_j$ of the RBFs can be chosen using a clustering algorithm, the parameters of the Gaussian can be found based on the spread (variance) of data in each cluster, and it can be shown that the weights can be found by solving a system of linear equations. This is done for a certain number of RBFs, with the exhaustive optimisation run across the number of RBFs in a certain range. Conceptually, RBF networks are close to the modular models considered below. RBF networks were widely used for problems similar to those where MLP networks were used. The following examples could be mentioned:

- Sudheer & Jain (2003) used RBF ANNs for modelling river stage-discharge relationships and showed that, in the considered case study, RBF ANNs were superior to MLPs;
- Moradkhani *et al.* (2004) used RBF ANNs for predicting hourly streamflow hydrographs for the daily flow for a river in the USA as a case study, and demonstrated their accuracy if compared to other numerical prediction models. In this study RBF was combined with the self-organising feature maps used to identify the clusters of data;
- Nor *et al.* (2007) used RBF ANNs for the same purpose; however, just for the hourly flow and considering only storm events in the two catchments in Malaysia as case studies.

Genetic programming (GP) and evolutionary regression. GP is a symbolic regression method in which the specific model structure is not chosen *a priori*, but is a result of the search process. Various elementary mathematical functions, constants and arithmetic operations are combined in one function and the algorithm tries to build a model recombining these building blocks in one formula. The function structure is represented as a tree and since the resulting function is highly nonlinear and often

non-differentiable, it is optimised by a randomised search method – usually a GA. An overview of GP applications in hydrology can be found in Babovic & Keijzer (2005).

One of the criticisms towards GP relates to the fact that the formulae generated on the basis of the combination of multiple elementary functions are often extremely complex and carry no physical insight. To address this issue, an augmented version of GP – a dimensionally aware GP – has been proposed (Keijzer & Babovic 2002). It constraints the search and ensures that the output has the expected physical dimension by allowing only the formulae with variables with particular dimensions (being the combination of length, time, mass, etc.). This leads to the formula(e) with the dimensional semantics and increases the chance of them having some physical meaning. The usefulness of this approach was demonstrated in the experiments of generating a formula for the Chezy coefficient using the data generated by a numerical model of river flow through flexible vegetation in wetlands (Babovic & Keijzer 2000).

Laucelli *et al.* (2007) present an application of GP to the problem of forecasting the groundwater heads in an aquifer in Italy; in this study the authors also employed averaging of several models built on the data subsets generated by bootstrap.

In *evolutionary regression* (Giustolisi & Savic 2006), a method similar to GP, the elementary functions are chosen from a limited set and the structure of the overall function is fixed. Typically, a polynomial regression equation is used and the coefficients are found by GA. This method overcomes some shortcomings of GP, such as the computational requirements, the number of parameters to tune and the complexity of the resulting symbolic models. It was used, for example, for modelling groundwater level (Giustolisi *et al.* 2007*a*) and river temperature (Giustolisi *et al.* 2007*b*) and the high accuracy and transparency of the resulting models were reported.

Fuzzy rule-based systems (FRBS). Fuzzy logic was introduced by Lotfi Zadeh (1965) and since then it has found multiple successful applications, mainly in control theory (e.g. Kosko 1997). Fuzzy rule-based systems can be built by interviewing human experts, or by processing historical data and thus forming a data-driven model. These rules are "patches" of local models overlapped throughout the parameter space, using a sort of interpolation at a lower level to represent patterns in complex nonlinear relationships. The basics of the data-driven approach and its use in a number of water-related applications can be found in Bárdossy & Duckstein (1995).

Typically the following rules are considered:

$$IF\, x_1\, is\, A_{1,r} \quad AND \ldots AND \quad x_n\, is\, A_{n,r} \quad THEN\, y\, is\, \mathrm{B}$$

where $\{x_1, \ldots, x_n\} = \mathbf{x}$ = input vector; A_{im} = fuzzy set; r = index of the rule, $r = 1, \ldots R$. Fuzzy sets A_{ir} (defined as membership functions with values ranging from 0 to 1) are used to partition the input space into overlapping regions (for each input these are intervals). The structure of B in the consequent could be either a fuzzy set (then such a model is called a Mamdani model), or a function $y = f(\mathbf{x})$, often linear (and then the model is referred to as a Takagi–Sugeno–Kang (TSK) model). The model output is calculated as a weighted combination of the R rules' responses. Output of the Mamdani model is fuzzy (a membership function of irregular shape), so the crisp output has to be calculated by the so-called defuzzification operator. Note that in the TSK model, each of the r rules can be interpreted as local models valid for certain regions in the input space defined by the antecedent and overlapping fuzzy sets A_{ir}. Resemblance to the RBF ANN is obvious.

FRBS were effectively used for drought assessment (Pesti *et al.* 1996); prediction of precipitation events (Abebe *et al.* 2000*a*); control of water levels in polder areas (Lobbrecht & Solomatine 1999); modelling rainfall-discharge dynamics (Vernieuwe *et al.* 2005). One of the limitations of FRBS is that the demand for data grows exponentially with an increase in the number of input variables. It is worth mentioning an important area where the principles and methods of fuzzy logic were also successfully used, which is analysis of model uncertainty. The uncertainty of inputs and parameters is described in fuzzy terms (fuzzy numbers) rather than probabilistic ones, and it is possible to generate the membership function (fuzzy number) characterising the output. This approach was applied, for example, in groundwater modelling (Abebe *et al.* 2000*b*) and rainfall–runoff modelling (Maskey *et al.* 2004).

Support vector machines (SVM). This machine learning method is based on the extension of the idea of identifying a hyperplane that separates two classes in classification. It is closely linked to the statistical learning theory initiated by V. Vapnik in the 1970s at the Institute of Control Sciences of the

Russian Academy of Science (Vapnik 1998). Originally developed for classification, it was extended to solving prediction problems, and in this capacity was used in hydrology-related tasks (note that currently some researchers attribute SVM to the group of the so-called kernel machines). Dibike *et al.* (2001) and Liong & Sivapragasam (2002) reported using SVMs for flood management and in prediction of river water flows and stages. Bray & Han (2004) addressed the issue of tuning SVMs for rainfall–runoff modelling. In all cases SVM-based predictors have shown good results in many cases superseding other DDM methods in accuracy (not always, however).

Chaos theory and nonlinear dynamics appear to be useful for time series forecasting when a time series carries enough information about the behaviour of the system (Abarbanel 1996). Let a time series $\{x_1, x_2, \ldots, x_t, \ldots, x_n\}$ be given (e.g. a sequence of water levels). The state of the system at time t can be represented by a vector $\mathbf{y}_t$ in m-dimensional state space $x_t, x_{t-\tau}, \ldots, x_{t-(m-1)\tau}$, where τ is the delay time. The whole time series can then be represented by a sequence of such vectors $\{\mathbf{y}_t\}$: $\{\mathbf{y}_m, \mathbf{y}_{m+1}, \ldots, \mathbf{y}_n\}$. If the original time series exhibits the so-called chaotic properties (manifested by its equivalent trajectory in the phase space following a quasi-periodic pattern), then the methods of chaos theory can be used to predict the future values of $\mathbf{y}$, and hence of x. For this, the so-called local model predicting the future value of $\mathbf{y}$ has to be built in phase space; this is an instance-based learning model, or a regression model (linear or nonlinear) built on the basis of the points representing the "moves in the phase space" of the neighbours of the current $\mathbf{y}$. The predictive capacity of chaos theory, based on an idea that the system behaves in the future in a similar manner as in the (distant) past, supersedes that of the linear models like ARIMA. In practical applications, the delay time τ and the dimension m need to be appropriately chosen (or determined by optimisation, for example minimising the model forecast error by GA) in order to fully capture the dynamic structure of the time series. Multivariate models embody time series representing several variables; they capture the interdependences of these variables and can be interpreted as the input–output data-driven models.

The chaos theory-based approach was used by Babovic *et al.* (2000) for predicting water levels at the Venice lagoon. Solomatine *et al.* (2000) and Velickov *et al.* (2003) used chaos theory to predict the surge water level in the Rijn river estuary and the two-hourly prediction error was at least on par with the accuracy of hydrodynamic models. Phoon *et al.* (2002) employed nonlinear dynamics for forecasting hydrologic time series. Note that the chaos-based methods do not have universal applicability: they can be successfully applied only when time series (or their combination) have certain properties, for example are periodic, or indeed exhibit properties of chaotic behaviour (or close to it) and when time series are of adequate (considerable) length. Note also a certain link between the chaos theory that uses local models and the principles of instance-based learning (considered below).

One of the research challenges relates to a quite a practical issue: the development of more reliable adaptive routines for determining the number of neighbours used in the local models.

Instance-based learning (IBL). In IBL (Mitchell 1997) no model is built: classification or numeric prediction is made directly by combining instances from the training data set that are close (typically in the Euclidean sense) to the new vector $\mathbf{x}_q$ of inputs (query point). In fact, IBL methods construct a local approximation to the modelled function that applies well in the immediate neighbourhood of the new query instance encountered. Thus it describes a very complex target function as a collection of less complex local approximations, and often demonstrates competitive performance when compared, for example, to ANNs.

A typical representative of IBL is the k-nearest neighbour (k-NN) method. For nominal output, the predicted class will just be the most common value among k training examples nearest to the query point $\mathbf{x}_q$. For real valued output, the estimate is the mean value of the k-nearest-neighbouring examples, possibly weighted according to their distance to $\mathbf{x}_q$. Further extensions are known as *locally weighted regression* (*LWR*) when the regression model is built on k nearest instances: the training instances are assigned weights according to their distance to $\mathbf{x}_q$ and the regression equations are generated on the weighted data.

Karlsson & Yakowitz (1987) introduced this method in the context of water issues, focusing, however, only on (single-variate) time series forecasts. Galeati (1990) demonstrated the applicability of the k-NN method (with the vectors composed of the lagged rainfall and flow values) for daily discharge forecasting and favourably compared it to the statistical ARX model. Shamseldin & O'Connor (1996)

used the k-NN method for adjusting the parameters of the linear perturbation model for river flow forecasting. Toth *et al.* (2000) compared the k-NN approach to other time series prediction methods in a problem of short-term rainfall forecasting. Ostfeld & Salomons (2005) developed a hybrid genetic–instance-based learning algorithm through linking a GA with a k-NN scheme for calibrating the 2D surface quantity and water quality model CE-QUAL-W2. Solomatine *et al.* (2007) explored a number of IBL methods and tested their applicability in short-term hydrologic forecasting.

To conclude the coverage of the popular data-driven methods it can be mentioned that most of them are developed in the computational intelligence community. The main challenges for the researchers in hydroinformatics are in testing various combinations of these methods for particular water-related problems, in combining them with the optimisation techniques, in developing the robust modelling procedures able to work with the noisy data, and in developing methods providing the model uncertainty estimates.

SOME OF THE MODERN TRENDS

Data-driven modelling has passed the initial stage in which researchers, excited by the power of new machine learning techniques, rushed to search for all possible data available to feed (often indiscriminately) into a model with the hope of constructing a good predictor. The power of basic data-driven modelling techniques has been already proven and the research community is now working towards development of the optimal model architectures and avenues for making data-driven models more robust, understandable and really useful for managers.

As regards the new modelling architectures, we will address herein an issue of the so-called modular models, being combinations of "local" models, with which we obtained lately some experience.

The usefulness of a model should be measured not only by its methodological correctness and accuracy, but mainly by the degree to which a model would be able to help a water manager or a decision-maker. In river basin management physically based models are widely applied and typically are found to be useful tools, so one of the challenges here is in the inclusion of DDM into existing decision-making frameworks, while taking into consideration both the system's physics and the data availability.

Another aspect of usefulness is the adequate reflection of reality which is uncertain, and in this respect developing the methods of dealing with the data and model uncertainty is currently an important issue. We will briefly address this issue as well.

Combination of "local" specialized models

Physical processes in rivers and river basins are multi-stationary, are composed of a number of sub-processes (e.g. related to various hydrologic conditions or river flow regimes), and their accurate modelling by the building of one single ("global") model is sometimes not possible. For river basins usually several physically based models are built, each responsible for modelling various aspects of the basin: hydraulic, hydrology, groundwater, etc. Typically, only one comprehensive model is developed for each of these areas. For example, a hydrologic model should be able to represent all complexity of hydrologic processes in the basin. If the processes are described with a sufficient level of detail, and properly encapsulated in the model, such a model may become an accurate representation of reality and is often adequate.

Sometimes, however, such a global model is not capable of describing all the sub-processes adequately and is not equally accurate for all hydrological conditions. In this case an option is to try to identify such sub-processes and to build separate models for each of them. Another approach is to build several similar models for the same process and to combine them in an "ensemble"; an example of such an approach is reported by Xiong *et al.* (2001), where a Takagi–Sugeno fuzzy model is used to combine conceptual rainfall–runoff models, and of course by many researchers using ensembles of meteorological and hydrological models.

In the case of using data-driven models, the situation is similar. A single DDM, e.g. ANN, often is not accurate for all possible situations. The collected data (training set) can be split into a number of subsets and separate models will be trained on these subsets (regions). These models are called *local*, or expert, models and the overall model a *modular model* (MM), or a *committee machine* (Haykin 1999). The way models are

built and combined can be subjected to optimisation, resulting in an overall model with the highest performance.

In the process of building, training and using a MM, two decisions have to be made: (A) which module should receive which training pattern (splitting problem) and (B) how the outputs of the modules should be combined to form the output of the final output of the system (combining problem) (Figure 2). Accordingly, two decision units have to be built, or one unit performing both functions. Such a unit is called an integrating unit or a gating network (a reference to a neural network often used for this purpose). It should be delivered to the user of the final model, along with the trained modules. Functioning of the units A and B could be different during training and operation. Classification of modular models (different from the one of Haykin (1999)) now follows.

Soft splitting of the training set. The group of the statistically driven approaches with "soft" splits of input space are represented by *mixtures of experts* (Jordan & Jacobs 1995), *bagging* (bootstrap aggregating; Breiman 1996) and *boosting* (Freund & Schapire 1997). Here we will briefly introduce boosting only.

Boosting (its advanced version, *AdaBoost*, is described by Freund & Schapire (1997)) can be seen as a method of building a series of modular models using soft splits. In the first iteration the basis model is trained (this will be the first module) on the whole dataset. The probability for each data vector to be selected for the next iteration is adjusted: it is increased if prediction for this data vector was poor. Using this distribution the new dataset of the same size is sampled from the original set and the new model is built. This process is repeated n times, thus resulting in n modules, each trained on different (intersecting) subsets. The combining unit B uses the weighted sum of the modules, where the weight is dependent on the accuracy of the module on the sample used. During training, unit A arranges the recalculation of the distribution and proper resampling, and during operation it simply distributes each new input vector to all modules. Boosting was originally developed for binary classification problems and was later extended to solve multiclass classification problems (*AdaBoost.M2*) and regression problems (*AdaBoost.R*). An improved version of boosting for regression is *AdaBoost.RT* by Shrestha & Solomatine (2006*a*). They demonstrated its advantages in comparison to other boosting algorithms and other learning methods on several benchmarking problems and two problems of river flow forecasting.

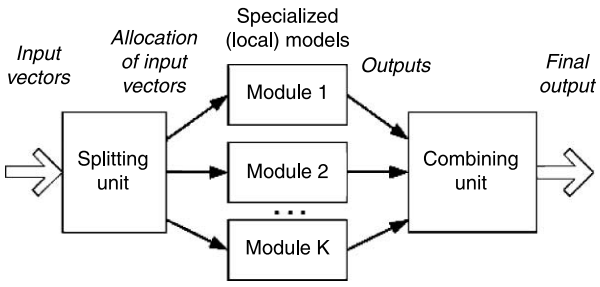

Figure 2 | Combining specialised local models.

Hard splitting of the training set. A number of methods do not combine the outputs of different models but explicitly use only one of them, the most appropriate one (a particular case when the weights of other expert models are zero). Such methods use "hard" splits of input space into regions. Each individual local model is trained individually on the subsets of instances contained in these regions, and finally the output of only one specialized model is taken into consideration. This can be done manually by experts on the basis of domain knowledge. Another way is to use information theory and to perform splitting progressively; examples are decision trees (Quinlan 1986), regression trees (Breiman *et al.* 1984) or M5 model trees (Quinlan 1992).

Several examples of such an approach can be mentioned. See & Openshaw (2000) built different neural networks based on different types of hydrological events. Hsu *et al.* (2002) presented a method of reproducing the catchment response through multiple local linear regression models which are built for specific flow conditions relating to the clusters identified by a Kohonen network. Solomatine & Xue (2004) used M5 model trees and neural networks in a flood-forecasting problem, combining the models valid for particular hydrologic conditions only (see the next subsection). Wang *et al.* (2006) used a combination of ANNs for forecasting flow: different networks were trained on the data subsets determined by applying either a threshold discharge value or clustering in the space of inputs (lagged discharges only but no rainfall data, however). Jain & Srinivasulu (2006) applied a mixture of neural networks and conceptual techniques to model the different segments of a decomposed flow hydrograph. Corzo & Solomatine (2007) used several methods of baseflow separation, built different

models for base and excess flow and combined these models, ensuring optimal overall model performance.

Regression trees and M5 model trees. This class of models is not yet popular in river management, but the known applications to water issues show their high performance (Witten & Frank 2000). These machine-learning techniques use the following idea: split the parameter space into areas (subspaces) and build in each of them a separate regression model of zero or first order (Figure 3). In M5 trees models in leaves are linear. The data set T is either associated with a leaf (where a regression model is built) or with a node (where some test is chosen that splits T into subsets corresponding to the test outcomes). The same process is applied recursively to the subsets. In the case of numeric inputs the Boolean tests a_i at a node used to split the data set have the form "$x_i < C$" where i and C are chosen to minimise the standard deviation in the subsets resulting from the split. Mn are local specialised models built for subsets filtered down to a given tree leaf. The resulting model can be seen as a committee of linear models being specialized on the certain subsets of the training set belonging to particular regions of the input space.

Combination of linear models was used in dynamic hydrology already in the 1970s (e.g. multi-linear models by Becker & Kundzewicz (1987)). The M5 model tree approach advances it further by introducing algorithms based on information theory that makes it possible to automatically split the multi-dimensional parameter space and to generate a range of models according to the overall quality criterion.

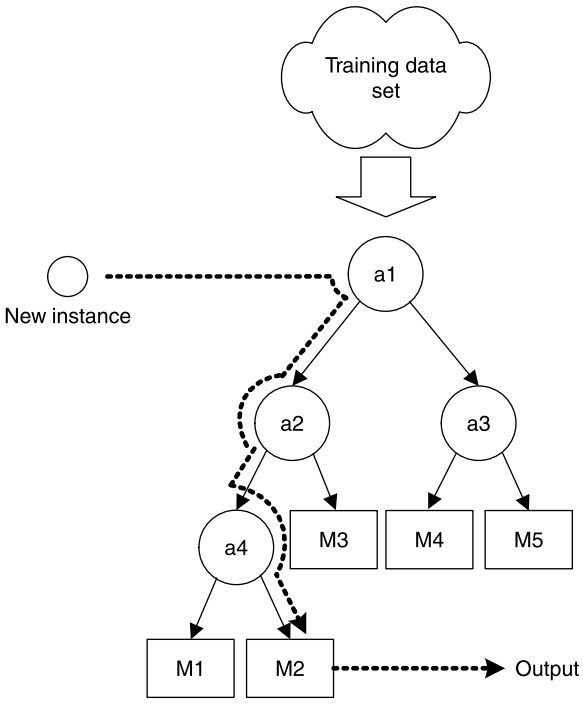

Figure 3 | Building a tree-like modular model (M5 model tree).

MTs may serve as an alternative to nonlinear models like ANNs and are often almost as accurate as ANNs, but have some important advantages: training of MTs is much faster than ANNs, and it always converges, and the results can be easily understood by decision-makers. Moreover, it is easy to generate a range of MTs varying in complexity and accuracy.

An early (if not the first) application of M5 model trees in river flow forecasting was reported by Kompare *et al.* (1997). Solomatine & Dulal (2003) used the M5 model tree in rainfall–runoff modelling of a river sub-basin in Italy. Stravs *et al.* (2006) used M5 trees in modelling the precipitation interception in the context of the Dragonja river basin case study.

It is worth mentioning that the models (modules) on Figure 2 may not be necessarily data-driven ones but rather have various natures, and may include expert judgements. If an overall model uses various types of models, it can be called a *hybrid model*. This is an important emerging research trend and a challenge in modelling of water-related assets.

Inclusion of a human expert and domain knowledge

One of the important challenges in data-driven modelling is incorporation of domain knowledge into the modelling process. A typical machine-learning algorithm minimises the training (cross-validation) error, seeing it as the ultimate indicator of the algorithm's performance, so it is purely data-driven. Domain experts, however, may have other considerations in judging the quality of the model, and want to have certain input into building a model. A human expert always participates in the process of model building, but his/her role could be very different. During the development of a DDM direct inclusion of an expert may increase the model accuracy and trust in the modelling results. An expert can contribute to building a DDM by bringing in the knowledge about the system under question (like is done in the dimensionally aware GP considered above), determining the model structure (as in the M5flex algorithm by Solomatine & Siek (2004)), in performing advanced analysis to select the most relevant variables (Solomatine & Dulal 2003; Bowden *et al.* 2005), and

in deciding what data should be used and how it should be structured (as is done by most modellers).

We will address here a particular problem of including an expert and using the domain knowledge in the process of building modular models (Figure 2). In this context, the role for a human expert could be, for example, in making decisions (A) and (B) (or approving these made by an algorithm) and, of course, in the choice of models used in each unit.

Inclusion of a human expert. It is possible to mention a number of studies where an attempt is made to include a human expert in the process of building a modular model. In solving a flow forecasting problem, Solomatine & Xue (2004) introduced a human expert to determine the hydrological conditions for which separate DDMs were built. Solomatine & Siek (2004), Solomatine & Siek (2006) presented an M5flex algorithm, allowing an expert to choose the splitting rules in building M5 trees, directing thus the process of building a DDM, and demonstrated its accuracy in hydrologic modelling. Jain & Srinivasulu (2006) and Corzo & Solomatine (2007) also applied decomposition of the flow hydrograph by a threshold value and then built the separate ANNs for low and high flow regimes. All these studies demonstrated the higher accuracy of the resulting modular models if compared to the models built to represent all possible regimes of the modelled system.

The study by Solomatine & Xue (2004) will be used as an illustration of such an approach. In it, the flow predictions in the Huai river basin (China) were made on the basis of previous flows and precipitation, and a committee hybrid model was built. The problem was to predict flow Q_{t+1} one day ahead. The following notations are used: flows on the previous and the current day as Q_{t-1} and Q_t, respectively; precipitation on the previous day as P_{t-1}; moving average (2 days) of the precipitation two days before as $Pmov2_{t-2}$; moving average (3 days) precipitation four days before as $Pmov3_{t-4}$.

As a first step the domain experts were asked to identify several hydrological conditions (rules), used to split the input space into regions. Some of the rules follow:

(1) $Q_{t-1} \geq 1000\,\mathrm{m^3/s}$ (high flows)
(2) $Q_{t-1} < 1000\,\mathrm{m^3/s}$ AND $Q_t \geq 200\,\mathrm{m^3/s}$ (medium flows)
(3) $P_{t-1} > 50$ AND $Pmov2_{t-2} < 5$ AND $Pmov3_{t-4} < 5$ (flood condition due to the short but intensive rainfall after a period of dry weather).

For each of these conditions separate local models were built (M5 model trees and ANNs). The presented approach demonstrated that combination of several "local" models improves the accuracy of prediction.

Inclusion of domain knowledge in algorithmic form. A human expert can be seen, of course, as a supplier of domain knowledge. However, recently there is an increased interest in exploring the possibilities of encapsulating the domain knowledge in algorithmic form and thus making it part of a data-driven model, thus allowing for performing optimisation of the latter. One such approach is being developed by Corzo & Solomatine (2007) and is used to improve the accuracy of a predictive rainfall – runoff model. In it, separate ANN models for baseflow and excess flow are built. For baseflow separation two methods are used: constant slope method and a recurrent filter. These methods, representing the hydrological knowledge about this phenomenon, are algorithmically implemented and run on the training dataset; then surrogate classifiers are trained to replicate them (since their straightforward implementation needs future data and it is not available during operation). The resulting modular model undergoes an exhaustive optimisation to ensure optimal accuracy. Application of this approach to two catchments demonstrates its value, especially for longer forecasting horizons.

DATA-DRIVEN MODELS OF UNCERTAINTY

Modelling uncertainty was always an issue associated with river basin management, but recently the interest in this problem and, accordingly, the number of publications has dramatically increased. One of the reasons is probably purely technical: computer power and advances in networked computer clusters nowadays allow for running Monte Carlo-based analysis of parametric uncertainty of quite complex models. However, there is, of course, a deeper reason: general recognition of the inadequacy of "point predictions" generated by most water models to the requirements of real-life water management. An important trend of the last several years is complementing the modelling studies of river basins with the sensitivity and uncertainty analysis (Montanari & Brath 2004). Recently we have made a step towards building the data-driven models of uncertainty.

Error prediction models. Consider a model simulating or predicting certain water-related variables (referred to as a primary model). This model's outputs are compared to the recorded data and the errors are calculated. Another model, a data-driven model, is trained on the recorded errors of the primary model and can be used to correct errors of the primary model. In the context of river modelling, this primary model would be typically a physically based model, but can be a data-driven model as well.

Such an approach was employed in a number of studies. Shamseldin & O'Connor (2001) used ANNs to update runoff forecasts: the simulated flows from a model and the current and previously observed flows were used as input, and the corresponding observed flow as the target output. Updates of daily flow forecasts for a lead-time of up to four days were made, and the ANN models gave more accurate improvements than autoregressive models. Lekkas *et al.* (2001) showed that error prediction improves real-time flow forecasting, especially when the forecasting model is poor. Babovic *et al.* (2001) used ANN to predict errors of 2D hydrodynamic models. Abebe & Price (2004) used ANN to correct the errors of a routing model of the River Wye in the UK. Solomatine *et al.* (2007) built an ANN-based rainfall–runoff model whose outputs were corrected by an instance-based learning model.

Uncertainty prediction models. Data-driven (machine-learning) methods may be helpful not only in modelling natural processes, but also in building models of the error probability distributions for physically based models. Recently Shrestha & Solomatine (2006*b*) presented an approach termed UNcertainty Estimation based on local model Errors (UNEEC). It is based on an idea to build local data-driven models predicting the properties of the error distribution, and uses clustering and fuzzy logic. This is a distribution-free, non-parametric method to model the propagation of integral uncertainty through the models and it was tested in forecasting river flows in a flood context.

One of the interesting research directions in building the models of uncertainty is finding the ways of combining the fuzzy and probabilistic descriptors of uncertainty in a data-driven model, and building robust predictors of model uncertainty originating from various sources.

TWO EXAMPLES

We are presenting two examples that illustrate several machine-learning methods used in solving river-basin-related problems. They also demonstrate how data-driven models are built in terms of choosing appropriate inputs, data processing and model optimisation.

DDM for forecasting river flows

Solomatine *et al.* (2007) used decision trees and k-NN in classification of river flow levels according to their severity in a river flood forecasting problem in Nepal. In this problem a medium-sized foothill-fed river in the Bagmati basin was considered, having an area of about 3700 km^2. Time series data of rainfall at three stations within the basin with daily sampling over eight years (1988–1995) were collected. Daily flows were recorded at one station so this precluded modelling the routing. Weight factors were calculated using the Thiessen polygon. The daily evapotranspiration was computed using the modified Penman method recommended by FAO.

Generally a rainfall–runoff data-driven model predicting flow T days ahead was sought in the form presented in Table 1.

First, dependence analysis of input and output variables was accomplished by visual inspection. Then the

Table 1 | Forecasting data-driven hydrologic model (Bagmati catchment)

Available data	**Measured rainfalls R_t, flows Q_t, T 1, … , E**
Inputs (L and M are to identified as a result of model optimization)	Lagged rainfalls $R_{t-\tau}$, $\tau = 0, \dots, L$
Forecasting model (forecast horizon T)	Lagged flows $Q_{t-\psi}$, $\psi = 0, \dots, M$
(F is typically multiple linear regression model, ANN, SVM, or M5 model tree)	$Q_{t+T} = f\,(R_t, R_{t-1}, \dots, R_{t-L}, Q_t, \dots, Q_{t-M})$

interdependences between variables and the lags τ were established using correlation and average mutual information (AMI) analyses (Solomatine & Dulal 2003; Bowden *et al.* 2005). By visual inspection of several precipitation events the maximum value of peak-to-peak time lags of rainfall and runoff was found to be close to one day. The cross-correlation analysis of the rainfall and runoff gave a maximum correlation of 0.78 for one day lag, so this lag was accepted as the average lag time of rainfall. This value of this lag was also consistent with AMI analysis. The autocorrelation function of runoff drops rapidly within three time steps (days). As a result, the model predicting flow one day ahead on the basis of five variables was set to be of the form

$$Q_{t+1} = f(RE_{t-2}, RE_{t-1}, RE_t, Q_{t-1}, Q_t) \quad (1)$$

An important problem is splitting the data into training and testing datasets. The ways to do it and the possible problems have been mentioned previously. In this study we used two approaches – a method based on randomization to create statistically similar training, cross-validation and testsets, and a method based on hydrological analysis of data to generate three contiguous datasets, trying to ensure at the same time at least some statistical resemblance of these sets. In the latter one eight years of data sets (2919 records) were split as follows: the first 919 records were used as testing data set and the remaining records as training and cross-validation data. Each instance was represented by a vector in five-dimensional space (since there are five inputs) accompanied by the associated value of its output variable.

In the study of Solomatine *et al.* (2007) several methods of instance-based learning were applied (including local weighted regression), along with ANN, M5 model tree models and a lumped conceptual model. The results show high accuracy of all data-driven methods, especially the weighted local regression. Corzo & Solomatine (2007) also applied a modular ANN model to the same case study, and have shown that building two separate models related to baseflow and excess flow, with the global optimisation of the resulting model structure, increases the prediction accuracy, if compared to a single model. The mentioned papers provide more details of the experiments conducted and the visualisation of the results.

GA-based optimization of M5 model trees for predicting river basin output flow and contaminant transport

This example application is based on Preis *et al.* (2006) and Ostfeld & Preis (2005) for the flow and the contaminant predictions at Lake Kinneret (the Sea of Galilee) watershed, located in northern Israel. The Lake Kinneret watershed is about 2730 km^2 (2070 in Israel, with the rest in Lebanon), is inhabited by about 200 000 people, organised into 25 municipalities, and three cities (in the Israeli part). The watershed outlet is Lake Kinneret, which is the most important surface water resource in Israel, providing approximately 35% of its annual drinking water demand.

Factors such as the rapid increase in Israel's population over the last decade along with an increase in its standard of living, the Israeli peace agreement with Jordan and the increasingly frequent droughts in the region are consistently intensifying the demand for freshwater, and hence the need to remove larger volumes of water from the lake. These factors further increase the likelihood of water quality decline; thus preserving the lake from further pollution is a foremost concern.

The developed data-driven model is aimed at predicting flow and contaminant transports within the watershed,

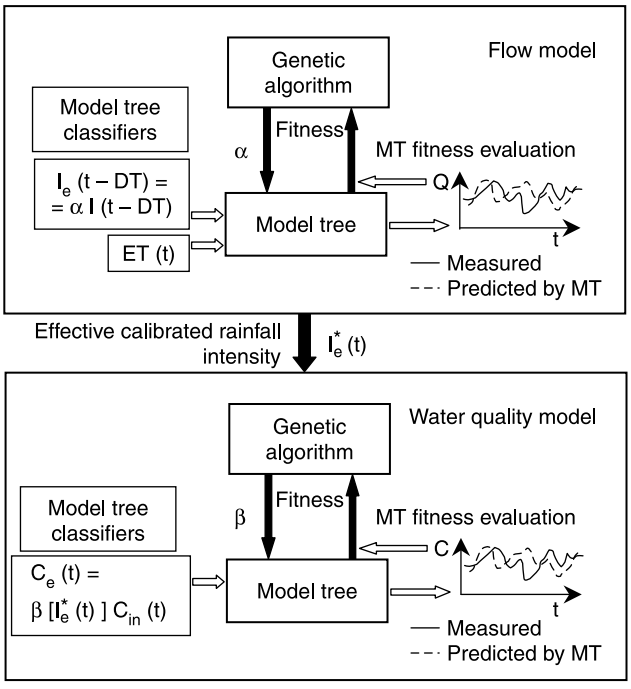

Figure 4 | Schematics of the hybrid model tree – genetic algorithm scheme.

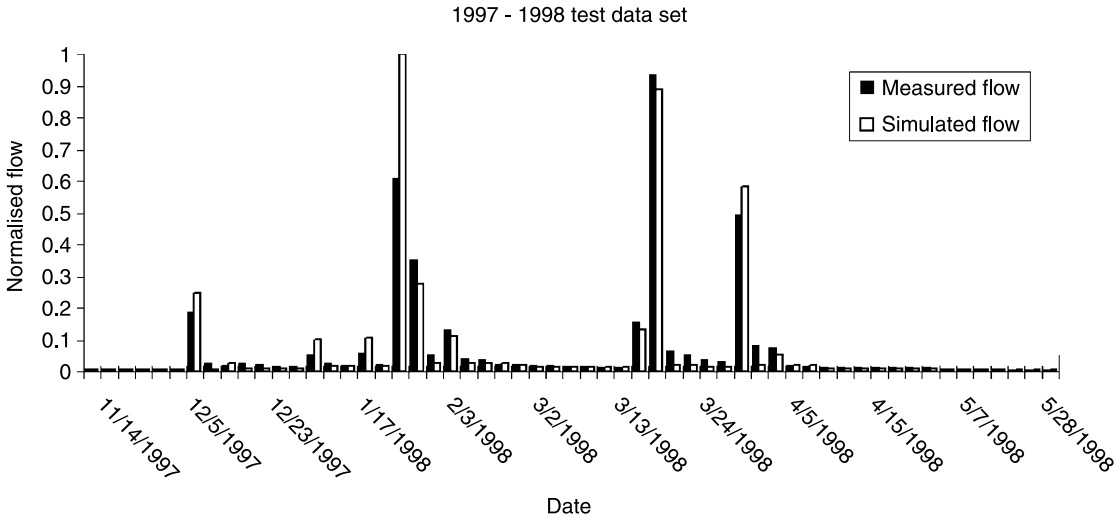

Figure 5 | Measured and simulated flow (normalised), test data, 1997–1998.

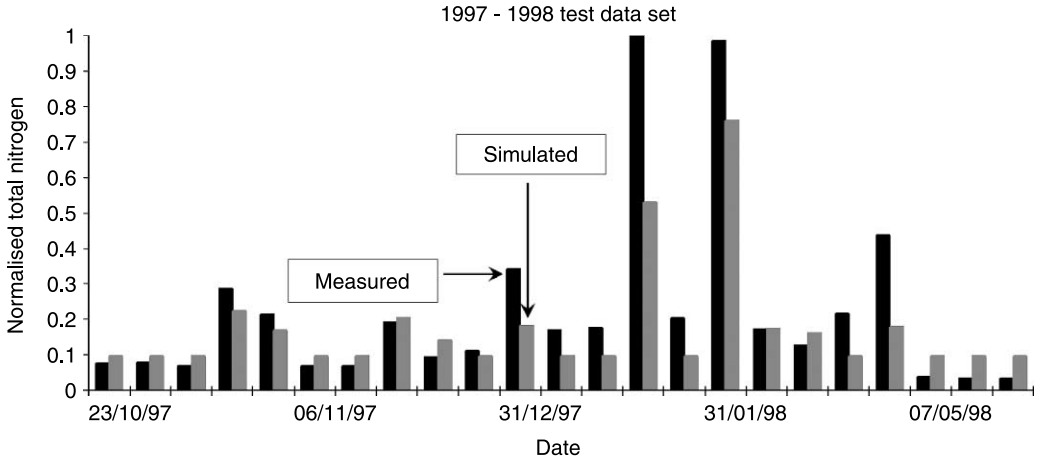

Figure 6 | Measured and simulated total nitrogen (normalised), test data, 1997–1998.

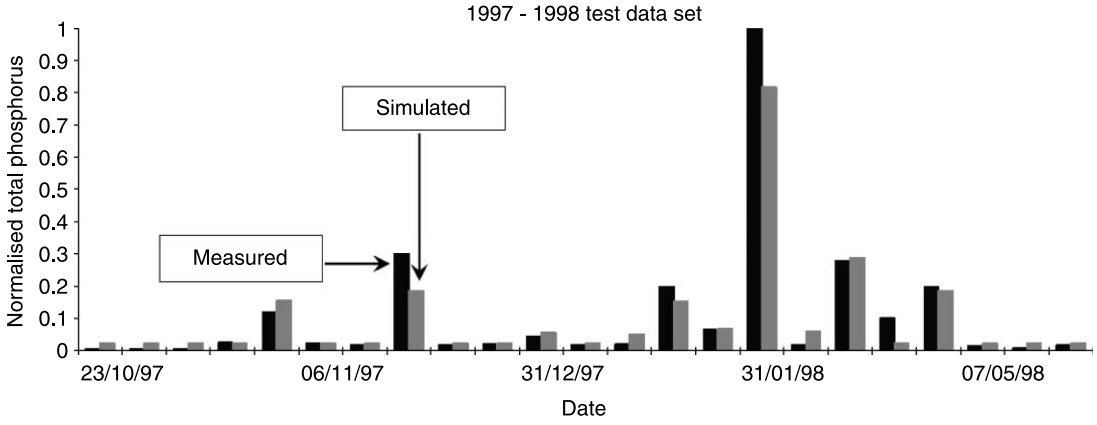

Figure 7 | Measured and simulated total phosphorus (normalised) test data, 1997–1998.

down to its outlet – Lake Kinneret. The model, entitled KWAT (Kinneret Watershed Analysis Tool), follows a hybrid set-up combining GA and model trees (MT) (Figure 4). The objective of the Flow section of the model is to tune the values of a vector of coefficients α that multiply the average rainfall time series intensity $I(t)$ (the input) imposed on a watershed so as to calibrate its outlet flows $Q(t)$. The Water quality section of the model then uses these optimal flows $Q(t)$ and the effective optimal rainfall intensities $I_e^*(t)$ to adjust the values of a vector of coefficients β so as to calibrate the watershed outlet concentrations $C(t)$.

In the Flow section the following variables are used: $t =$ time; DT = lag in time (e.g. the concentration time of the watershed); $\alpha =$ vector of coefficients (the GA decision variables); $I(t)$ = time series of the average rainfall intensity imposed on the watershed (e.g. using the average rainfall Theisen method); ET(t) = evapotranspiration time series; $I_e(t)$ = time series of the effective rainfall; $Q(t)$ = flow time series at the watershed outlet; and Fitness = the fitness of the model tree (MT) outcome analysis estimated through a least square type equation.

In the Water quality section the variables used are: $I_e^*(t)$ = the optimal effective rainfall intensity time series (i.e. the outcome of the quantity model); $C_{in}(t)$ = the input concentration time series imposed on the watershed; β = vector of coefficients (the GA decision variables); $C_e(t)$ = an "effective" resultant concentration time series; and $C(t)$ = concentration time series at the watershed outlet.

To reduce the computational complexity and to increase the model robustness, the dimension of the α and β coefficient vectors are set to be much less than the dimension of t. This is accomplished by dividing the time series of the rainfall intensity $I(t)$ to a set of category domains of no more than six (i.e. the rainfall intensity is divided into six categories, with α and β values assigned to each).

Figures 5–7 show results for the flow and water quality models as applied to a sub-watershed of Lake Kinneret (Meshushim watershed – 140 km^2, Ostfeld & Preis (2005)). Figure 5 shows the results for the flow model test data set for 1997–1998. Figures 6 and 7 describe the results for the water quality model test data set for 1997–1998 for predicting total nitrogen and total phosphorus, respectively.

It can be seen from Figures 5–7 that the predictions received by the developed flow and water quality models were, in general, in good agreement with the measurements. However, the models were less successful in predicting high flows and water quality concentrations. This is an inherent limitation of a data-driven technique whose accuracy is primarily dependent on the quality of the dataset used for training. The larger a dataset, the greater is the chance to have better predictions. It is anticipated that increasing the number of training instances for the proposed model will also improve its prediction accuracy.

CONCLUSIONS

Data-driven modelling and computational intelligence methods have proven their applicability to various problems related to river basin management: modelling, short-term forecasting, classification of hydrology-related data, and even automated generation of flood inundation maps based on aerial photos (not discussed in this paper due to lack of space, see, e.g., Velickov *et al.* (2000)), etc. A particular problem will benefit from data-driven modelling if: (1) there is a considerable amount of data available; (2) there are no considerable changes to the system during the period covered by the model; (3) it is difficult to build adequate knowledge-driven simulation models due to the lack of understanding and/or to the ability to satisfactorily construct a mathematical model of the underlying processes. Of course, data-driven models can also be useful when there is a necessity to validate the simulation results of physically based models with other types of models.

It can be said that it is practically impossible to recommend one particular type of data-driven model for a given problem. Since water-related applications are often characterised by the data being noisy and of poor quality, it is advisable to apply various types of techniques and to compare and/or combine the results. For example, M5 model trees, combining local and global properties, could very well complement ANNs, and be more easily accepted by decision-makers due to their reliance on simple linear models.

We have considered and demonstrated some of the new trends in data-driven modelling and mentioned a number of research challenges. It is worth mentioning one challenge of

a general nature: development of *hybrid* models by combining the models of different types and following different modelling paradigms, including the combination of data-driven physically based models, and finding effective ways of including of a human expert in the modelling cycle.

ACKNOWLEDGEMENTS

This work was partly supported by the EU project "Integrated Flood Risk Analysis and Management Methodologies" (FLOODsite), contract GOCE-CT-2004-505420, and the Delft Cluster Research Programme of the Dutch Government (project 4.30 "Safety against flooding"). The authors are grateful to the three reviewers for their valuable comments.

REFERENCES

Abarbanel, H. D. I. 1996 *Analysis of Observed Chaotic Data*. Springer-Verlag, Berlin.

Abbott, M. B. 1991 *Hydroinformatics: Information Technology and the Aquatic Environment*. Avebury Technical, Aldershot.

Abebe, A. J., Solomatine, D. P. & Venneker, R. 2000*a* Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events. *Hydrol. Sci. J.* **45** (3), 425–436.

Abebe, A.J., Guinot, V. & Solomatine, D.P. 2000*b* Fuzzy alpha-cut vs. Monte Carlo techniques in assessing uncertainty in model parameters. *Proc. 4th Int. Conf. on Hydroinformatics, Cedar Rapids*. Available online: http://www.ihe.nl/hi/sol/papers/HI2000-AlphaCut.pdf

Abebe, A. J. & Price, R. K. 2004 Information theory and neural networks for managing uncertainty in flood routing. *ASCE J. Comput. Civil Engng.* **18** (4), 373–380.

Abrahart, R. J., Heppenstall, A. J. & See, L. M. 2007*a* Timing error correction procedure applied to neural network rainfall-runoff modelling. *Hydrol. Sci. J.* **52** (3), 414–431.

Abrahart, R.J., See, L.M., Solomatine, D.P. & Toth, E (Eds.) 2007*b* Data-driven approaches, optimization and model integration: hydrological applications. *Hydrol Earth Syst. Sci*. **11** (Special issue)

Abrahart, R. J. & See, L. 2000 Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecast in two contrasting catchments. *Hydrol. Process.* **14**, 2157–2172.

Abrahart, R. J. & See, L. M. 2007 Neural network modelling of non-linear hydrological relationships. *Hydrol. Earth Syst. Sci.* **11**, 1563–1579.

Abrahart, B., See, L. M. & Solomatine, D. P. 2008 *Hydroinformatics in Practice: Computational Intelligence and Technological Developments in Water Applications*. Berlin, Springer-Verlag. In press.

AVGWLF ArcView Generalized Watershed Loading Function. Available at: http://www.avgwlf.psu.edu/

Babovic, V., Canizares, R., Jensen, H. R. & Klinting, A. 2001 Neural networks as routine for error updating of numerical models. *ASCE J. Hydraul. Engng.* **127** (3), 181–193.

Babovic, V. & Keijzer, M. 2000 Genetic programming as a model induction engine. *J. Hydroinf.* **2**, 35–60.

Babovic, V. & Keijzer, M. 2005 Rainfall runoff modelling based on genetic programming. In *Encyclopedia of Hydrological Sciences*, **vol 1.** (ed. Andersen, M.G.). John Wiley & Sons, New York, Doi: 10.1002/0470848944.hsa017.

Babovic, V., Keijzer, M. & Stefansson, M. 2000 Optimal embedding using evolutionary algorithms. In: *Proc. 4th Int. Conference on Hydroinformatics*, *Cedar Rapids*.

Bárdossy, A. & Duckstein, L. 1995 *Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological and Engineering Systems*. CRC Press, Boca Raton, FL.

BASINS Better Assessment Science Integrating Point and Nonpoint Sources. Available at: http://www.epa.gov/OST/BASINS/

Becker, A. & Kundzewicz, Z. W. 1987 Nonlinear flood routing with multilinear models. *Wat. Res. Res.* **23**, 1043–1048.

Bhattacharya, B. & Solomatine, D. P. 2005 Neural networks and M5 model trees in modelling water level – discharge relationship. *Neurocomputing* **63**, 381–396.

Bowden, G. J., Dandy, G. C. & Maier, H. R. 2003 Data transformation for neural network models in water resources applications,. *J. Hydroinf.* **5**, 245–258.

Bowden, G. J., Dandy, G. C. & Maier, H. R. 2005 Input determination for neural network models in water resources applications. Part 1–Background and methodology. *J. Hydrol.* **301**, 75–92.

Bowden, G. J., Maier, H. R. & Dandy, G. C. 2002 Optimal division of data for neural network models in water resources applications. *Wat. Res. Res.* **38** (2), 1–11.

Bray, M. & Han, D. 2004 Identification of support vector machines for runoff modelling. *J. Hydroinf.* **6**, 265–280.

Breiman, L. 1996 Bagging predictors. *Machine Learning* **24** (2), 123–140.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. 1984 *Classification and Regression Trees*. Wadsworth International, Belmont.

Cherkassky, V., Krasnopolsky, V., Solomatine, D.P. & Valdes, J (Eds.) 2006 Computational intelligence in earth sciences and environmental applications. *Neural Networks J.* **19** (2)

Corzo, G. & Solomatine, D. P. 2007 Baseflow separation techniques for modular artificial neural network modelling in flow forecasting. *Hydrol. Sci. J.* **52** (3), 491–507.

Dawson, C. W. & Wilby, R. 1998 An artificial neural network approach to rainfall-runoff modelling. *Hydrol. Sci. J.* **43** (1), 47–66.

Dibike, Y., Solomatine, D. P. & Abbott, M. B. 1999 On the encapsulation of numerical-hydraulic models in artificial neural network. *J. Hydraul. Res.* **37** (2), 147–161.

Dibike, Y. B., Velickov, S., Solomatine, D. P. & Abbott, M. B. 2001 Model induction with support vector machines: introduction and applications. *ASCE J. Comput. Civil Engng.* **15** (3), 208–216.

Diskin, M. H. 1964 *A basic study of the linearity of the rainfall – runoff process in watersheds*. PhD thesis University of Illinois. Urbana, Champaign.

Diskin, M. H. & Boneh, A. 1975 Determination of an optimal IUH for linear time invariant systems from multi-storm records. *J. Hydrol.* **24**, 57–76.

Diskin, M. H., Wyseure, G. & Feyen, J. 1984 Application of a cell model to the Bellebeek watershed. *Nordic Hydrol.* **15**, 25–38.

Dooge, J. C. I. 1959 A general theory of the unit hydrograph. *J. Geophys. Res.* **64** (2), 241–256.

Eagleson, P. S., Mejia, R. & March, F. 1966 Computation of optimum realizable unit hydrographs. *Wat. Res. Res.* **2** (4), 755–764.

Efron, B. & Tibshirani, R. J. 1993 *An Introduction to the Bootstrap*. Chapman & Hall, New York.

Falconer, R., Lin, B. & Harpin, R. 2005 Environmental modelling in river basin management. *J. River Basin Mngmnt.* **3** (3), 169–184.

Freund, Y. & Schapire, R. 1997 A decision-theoretic generalisation of on-line learning and an application of boosting. *J. Comput. System Sci.* **55** (1), 119–139.

Galeati, G. 1990 A comparison of parametric and non-parametric methods for runoff forecasting. *Hydrol. Sci. J.* **35** (1), 79–94.

Gaume, E. & Gosset, R. 2003 Over-parameterisation, a major obstacle to the use of artificial neural networks in hydrology? *Hydrol. Earth Syst. Sci.* **7** (5), 693–706.

Giustolisi, O., Doglioni, A., Savic, D. A. & di Pierro, F. 2007*a* An evolutionary multiobjective strategy for the effective management of groundwater resources. *Wat. Res. Res.* doi:10.1029/2006WR005359.

Giustolisi, O., Doglioni, A., Savic, D. A. & Webb, B. W. 2007*b* A multi-model approach to analysis of environmental phenomena. *Environ. Modell. Syst. J.* **22** (5), 674–682.

Giustolisi, O. & Savic, D. A. 2006 A symbolic data-driven technique based on evolutionary polynomial regression. *J. Hydroinf.* **8** (3), 207–222.

Govindaraju, R.S., & Ramachandra Rao, A. (Eds.) 2001, *Artificial Neural Networks in Hydrology*. Kluwer, Dordrecht.

Haith, D. A. & Shoemaker, L. L. 1987 Generalized watershed loading functions for stream flow nutrients. *Wat. Res. Bull.* **23** (3), 471–478.

Hall, M. J. & Minns, A. W. 1999 The classification of hydrologically homogeneous regions. *Hydrol. Sci. J.* **44**, 693–704.

Han, D., Kwong, T. & Li, S. 2007 Uncertainties in real-time flood forecasting with neural networks. *Hydrol. Process.* **21** (2), 223–228.

Hannah, D. M., Smith, B. P. G., Gurnell, A. M. & McGregor, G. R. 2000 An approach to hydrograph classification. *Hydrol. Process.* **14** (2), 317–338.

Harris, N. M., Gurnell, A. M., Hannah, D. M. & Petts, G. E. 2000 Classification of river regimes: a context for hydrogeology. *Hydrol. Process.* **14**, 2831–2848.

Haykin, S. 1999 *Neu Networks: A Comprehensive Foundation*. McMillan, New York.

Hsu, K. L., Gupta, H. V., Gao, X., Sorooshian, S. & Imam, B. 2002 Self-organizing linear output map (SOLO): an artificial neural network suitable for hydrologic modeling and analysis. *Wat. Res. Res.* **38** (12), 1–17.

Hsu, K. L., Gupta, H. V. & Sorooshian, S. 1995 Artificial neural network modelling of the rainfall-runoff process. *Wat. Res. Res.* **31** (10), 2517–2530.

Hu, T., Wu, F. & Zhang, X. 2007 Rainfall–runoff modeling using principal component analysis and neural network. *Nordic Hydrol.* **38** (3), 235–248.

Jain, A. & Srinivasulu, S. 2006 Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques. *J. Hydrol.* **317**, 291–306.

Jordan, M. I. & Jacobs, R. A. 1995 Modular and hierarchical learning systems. In *The Handbook of Brain Theory and Neural Networks* (ed. M. Arbib). MIT Press, Cambridge, MA.

Karlsson, M. & Yakowitz, S. 1987 Nearest neighbour methods for non-parametric rainfall runoff forecasting. *Wat. Res. Res.* **23** (7), 1300–1308.

Keijzer, M. & Babovic, V. 2002 Declarative and preferential bias in GP-based scientific discovery. *Genetic Programming and Evolvable Machines* **3** (1), 41–79.

Khu, S.-T., Savic D., Liu, Y. & Madsen, H. 2004 A fast evolutionary-based meta-modelling approach for the calibration of a rainfall-runoff model. In: *Trans. 2nd Biennial Meeting of the International Environmental Modelling and Software Society*, iEMSs, Manno, Switzerland. Available online: http://www.iemss.org/iemss2004/pdf/evocomp/khuafas.pdf

Kim, T., Heo, J. -H. & Jeong, C. -S. 2006 Multireservoir system optimization in the Han River basin using multi-objective genetic algorithms. *Hydrol. Process.* **20** (9), 2057–2075.

Kompare, B., Steinman, F., Cerar, U. & Dzeroski, S. 1997 Prediction of rainfall runoff from catchment by intelligent data analysis with machine learning tools within the artificial intelligence tools. *Acta Hydrotech.* **16/17** (79–94(in Slovene)).

Kosko, B. 1997 *Fuzzy Engineering*. Prentice-Hall, Englewood Cliffs, NJ.

Laucelli, D., Giustolisi, O., Babovic, V. & Keijzer, M. 2007 Ensemble modeling approach for rainfall/groundwater balancing. *J. of Hydroinf.* **9** (2), 95–106.

Lekkas, D. F., Imrie, C. E. & Lees, M. J. 2001 Improved non-linear transfer function and neural network methods of flow routing for real-time forecasting. *J. Hydroinf.* **3** (3), 153–164.

Liong, S. Y. & Sivapragasam, C. 2002 Flood stage forecasting with SVM. *J. AWRA* **38** (1), 173–186.

Lobbrecht, A. H. & Solomatine, D. P. 1999 Control of water levels in polder areas using neural networks and fuzzy adaptive systems. In *Water Industry Systems: Modelling and Optimization Applications* (ed. in D. Savic & G. Walters), pp. 509–518. Research Studies Press, Baldock.

Maskey, S., Guinot, V. & Price, R. K. 2004 Treatment of precipitation uncertainty in rainfall-runoff modelling: a fuzzy set approach. *Adv. Wat. Res.* **27** (9), 889–898.

MIKE SHE 2006 *MIKE SHE Integrated Modelling Tool*. Available at: http://www.dhisoftware.com/mikeshe/ (last accessed on 1 July 2006).

Minns, A. W. & Hall, M. J. 1996 Artificial neural network as rainfall-runoff model. *Hydrol. Sci. J.* **41** (3), 399–417.

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

Montanari, A. & Brath, A. 2004 A stochastic approach for assessing the uncertainty of rainfall-runoff simulations. *Wat. Res. Res.* **40**, W01106 doi:10.1029/2003WR002540.

Moradkhani, H., Hsu, K. L., Gupta, H. V. & Sorooshian, S. 2004 Improved streamflow forecasting using self-organizing radial basis function artificial neural networks. *J. Hydrol.* **295** (1), 246–262.

Muleta, M. K. & Nicklow, J. W. 2004 Joint application of artificial neural networks and evolutionary algorithms to watershed management. *J. Wat. Res. Mngmnt.* **18** (5), 459–482.

Nash, J. E. 1957 The form of the instantaneous unit hydrograph. *IASH Publ.* **45** (3), 114–121.

Nor, N. A., Harun, S. & Kassim, A. H. 2007 Radial basis function modeling of hourly streamflow hydrograph. *J. Hydrol. Engng.* **12** (1), 113–123.

Ostfeld, A. & Preis, A. 2005 A data driven model for flow and contaminants runoff predictions in watersheds. In *River Basin Restoration and Management* (ed. in A. Ostfeld & J. M. Tyson), pp. 62–70. Water and Environmental Management Series. IWA Publishing, London.

Ostfeld, A. & Salomons, S. 2005 A hybrid genetic–instance based learning algorithm for CE-QUAL-W2 calibration. *J. Hydrol.* **310**, 122–142.

Pesti, G., Shrestha, B. P., Duckstein, L. & Bogárdi, I. 1996 A fuzzy rule-based approach to drought assessment. *Wat. Res. Res.* **32** (6), 1741–1747.

Phoon, K. K., Islam, M. N., Liaw, C. Y. & Liong, S. Y. 2002 A practical inverse approach for forecasting nonlinear hydrological time series. *ASCE J. Hydrol. Engng.* **7** (2), 116–128.

Preis, A., Tubaltzev, A. & Ostfeld, A. 2006 Kinneret Watershed Analysis Tool (KWAT) - a cell based decision tree model for watershed flow and pollutants predictions. *Wat. Sci. Technol.* **53** (10), 29–35.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. 2007 *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. Cambridge University Press, Cambridge.

Quinlan, J. R. 1986 Induction of decision trees. *Machine Learning* **1** 81–106.

Quinlan, J. R. 1992 Learning with continuous classes. In *Proc. AI'92, 5th Australian Joint Conference on Artificial Intelligence* (ed. A. Adams & L. Sterling), World Scientific, Singapore. pp. 343–348.

RIBASIM 2006 *RIBASIM River Basin Planning and Management Tool*. at: http://www.wldelft.nl/soft/ribasim/int/index.html (last accessed on 1 July 2006).

Savic, D. 2005 Evolutionary computing in hydro-geological systems. In *Encyclopedia of Hydrological Sciences*, **vol 1.** (ed. M. G. Andersen), John Wiley & Sons, New York, Doi: 10.1002/0470848944.hsa016.

See, L., Openshaw, S. 2000 A hybrid multi-model approach to river level forecasting. *Hydrological Sciences J.* **45** (3), 523–536.

See, L. A., Solomatine, D. P., Abrahart, R. & Toth, E. 2007 Hydroinformatics: computational intelligence and technological developments in water science applications – Editorial. *Hydrol. Sci J.* **52** (3), 391–396.

Shamseldin, A. Y. & O'Connor, K. M. 1996 A nearest neighbour linear perturbation model for river flow forecasting. *J. Hydrol.* **179**, 353–375.

Shamseldin, A. Y. & O'Connor, K. M. 2001 A non-linear neural network technique for updating of river flow forecasts. *Hydrol. Earth Syst. Sci.* **5** (4), 557–597.

Sherman, L. K. 1932 Stream flow from rainfall by the unit graph method. *Engng. News-Record* **108**, 501–505.

Shrestha, D. L. & Solomatine, D. P. 2006*a* Experiments with AdaBoostRT, an improved boosting scheme for regression. *Neural Comput.* 2006, 17.

Shrestha, D. L. & Solomatine, D. P. 2006*b* Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks* **19** doi:10.1016/j.neunet.2006.01.012.

Solomatine, D. P. & Dulal, K. N. 2003 Model tree as an alternative to neural network in rainfall-runoff modelling. *Hydrol. Sci. J.* **48** (3), 399–411.

Solomatine, D.P., Rojas, C., Velickov, S. & Wust, H. 2000 Chaos theory in predicting surge water levels in the North Sea. In: *Proc. 4th Int. Conf. on Hydroinformatics*, *Cedar Rapids*.

Solomatine, D. P., Shrestha, D. L. & Maskey, M. 2007 Instance-based learning compared to other data-driven methods in hydrological forecasting. *Hydrol. Process.* 10.1002/hyp.6592.

Solomatine, D. P. & Siek, M. B. 2004 Flexible and optimal M5 model trees with applications to flow predictions. In: *Proc. 6th Int. Conf. on Hydroinformatics*. World Scientific, Singapore.

Solomatine, D. P. & Siek, M. B. 2006 Modular learning models in forecasting natural phenomena. *Neural Networks J.* **19** (2), 225–235.

Solomatine, D. P. & Torres, L. A. 1996 Neural network approximation of a hydrodynamic model in optimizing reservoir operation. In: *Proc. 2nd Int. Conf. on Hydroinformatics*, Balkema, Rotterdam. pp. 201–206.

Solomatine, D. P. & Xue, Y. 2004 M5 model trees and neural networks: application to flood forecasting in the upper reach of the Huai River in China. *ASCE J. Hydrol. Engng.* **9** (6), 491–501.

Stravs, L., Brilly, M. & Sraj, M. 2006 Precipitation interception modelling using machine learning methods – the Dragonja River basin case study. In: *Hydroinformatics in Practice: Computational Intelligence and Technological Developments in Water Applications* (ed. B. Abrahart, L. M. See & D. P. Solomatine), Springer-Verlag, Berlin.

Sudheer, K. P. & Jain, S. K. 2003 Radial basis function neural network for modeling rating curves. *ASCE J. Hydrol. Engng.* **8** (3), 161–164.

SWAT 2006 *SWAT Soil and Water Assessment Tool*. Available at: http://www.brc.tamus.edu/swat/ (last accessed on 1 July 2006).

Toth, E., Brath, A. & Montanari, A. 2000 Comparison of short-term rainfall prediction models for real-time flood forecasting. *J. Hydrol.* **239**, 132–147.

Vapnik, V. N. 1998 *Statistical Learning Theory*. John Wiley & Sons, New York.

Velickov, S., Solomatine, D. & Price, R. K. 2003 Prediction of nonlinear dynamical systems based on time series analysis: issues of entropy, complexity and predictability. In: *Proc. of the XXX IAHR Congress*. Thessaloniki, Greece.

Velickov, S., Solomatine, D.P., Yu, X. & Price, R.K. 2000 Application of data mining techniques for remote sensing image analysis. In: *Proc. 4th Int. Conf. on Hydroinformatics*, *Cedar Rapids*.

Vernieuwe, H., Georgieva, O., De Baets, B., Pauwels, V. R. N., Verhoest, N. E. C. & De Troch, F. P. 2005 Comparison of data-driven Takagi–Sugeno models of rainfall–discharge dynamics. *J. Hydrol.* **302** (1–4), 173–186.

Wang, W., van Gelder, P. H. A. J. M., Vrijling, J. K. & Ma, J. 2006 Forecasting daily streamflow using hybrid ANN models. *J. Hydrol.* **324** (1–4), 383–399.

Werbos, P.J. (1974/1994). *The Roots of Backpropagation*. John Wiley & Sons (Includes Werbos's 1974 Harvard Ph.D. thesis, *Beyond Regression*).

Witten, I. H. & Frank, E. 2000 *Data Mining*. Morgan Kaufmann, San Mateo, CA.

Xiong, L. H., Shamseldin, A. Y. & O'Connor, K. M. 2001 A non-linear combination of the forecasts of rainfall–runoff models by the first-order Takagi-Sugeno fuzzy system. *J. Hydrol.* **245** (1–4), 196–217.

Zadeh, L. A. 1965 Fuzzy sets. *Inf. Control* **8**, 338–353.

First received 27 April 2007; accepted in revised form 13 November 2007