Hydrol. Earth Syst. Sci. Discuss., 6, 7055–7093, 2009
www.hydrol-earth-syst-sci-discuss.net/6/7055/2009/

Hydrology and Earth System Sciences Discussions

This discussion paper is/has been under review for the journal Hydrology and Earth System Sciences (HESS). Please refer to the corresponding final paper in HESS if available.

Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology – Part 1: Concepts and methodology

A. Elshorbagy[1], G. Corzo[2], S. Srinivasulu[1], and D. P. Solomatine[2,3]

[1]Centre for Advanced Numerical Simulation (CANSIM), Department of Civil & Geological Engineering, University of Saskatchewan, S7N 5A9 Saskatoon, SK, Canada
[2]Department of Hydroinformatics & Knowledge Management, UNESCO-IHE Institute for Water Education, Delft, The Netherlands
[3]Water Resources Section, Delft University of Technology, Delft, The Netherlands

Received: 29 October 2009 – Accepted: 7 November 2009 – Published: 19 November 2009

Correspondence to: A. Elshorbagy (amin.elshorbagy@usask.ca)

Published by Copernicus Publications on behalf of the European Geosciences Union.

Abstract

A comprehensive data driven modeling experiment is presented in two-part paper. In this first part, an extensive data-driven modeling experiment is proposed. The most important concerns regarding the way data driven modeling (DDM) techniques and data were handled, compared, and evaluated, and the basis on which findings and conclusions were drawn are discussed. A concise review of key articles that presented comparisons among various DDM techniques is presented. Six DDM techniques, namely, neural networks, genetic programming, evolutionary polynomial regression, support vector machines, M5 model trees, and K-nearest neighbors are proposed and explained. Multiple linear regression and naïve models are also suggested as baseline for comparison with the various techniques. Five datasets from Canada and Europe representing evapotranspiration, upper and lower layer soil moisture content, and rainfall-runoff process are described and proposed for the modeling experiment. Twelve different realizations (groups) from each dataset are created by a procedure involving random sampling. Each group contains three subsets; training, cross-validation, and testing. Each modeling technique is proposed to be applied to each of the 12 groups of each dataset. This way, both predictive accuracy and uncertainty of the modeling techniques can be evaluated. The implementation of the modeling techniques, results and analysis, and the findings of the modeling experiment are deferred to the second part of this paper.

1 Introduction

Data driven modeling (DDM) techniques have been in use for nearly two decades for hydrological modeling, prediction, and forecasting. Many articles reporting the application of various techniques to various hydrological case studies are available in literature. Yet, data driven techniques are still facing some classical opposition because of multiple reasons inherit in such techniques (e.g., lack of transparency and difficulty

HESSD

6, 7055–7093, 2009

of reproducing the results). Hydroinformatics researchers started to identify problems of data driven modeling (Maier and Dandy, 2000; Elshorbagy and Parasuraman, 2008; Solomatine and Ostfeld, 2008) and tried to suggest some solutions or modeling guidelines and frameworks. Cherkassky et al. (2006) have listed the quality of the datasets, choosing robust learning methods that can handle heterogeneous data, and the need for uncertainty estimates associated with predictions as some of the main issues and challenges facing computational intelligence in earth sciences.

There is no doubt that more scientific rigour should have been maintained in the applications and use of data driven techniques in earth sciences. Bowden et al., (2005) explored different techniques for input determination for neural network models in water resources applications, showing a comparative performance of different methodologies for determining input variables. Abrahart et al. (2008) have used the example of neural network applications to highlight the shortcomings of the present approach, and how to build stronger foundations. Apparently, their argument can be easily generalized to apply to other data driven and soft computing techniques. In fact, the modeling shortcomings and ambiguity inherit in DDM techniques are less than the ones created because of the way such techniques were presented in earth sciences literature. One of the fundamental means to assess a modeling technique is to evaluate it against other modeling techniques, whether conceptual or data driven ones. One can observe that in the literature of soft computing or data driven hydrology, the modeling comparative studies are usually impaired due to the less-than-comprehensive approach adopted. With few exceptions, the following problems can be noticed: (i) Only one or two modeling techniques have been used at a time in a single study; (ii) if more techniques were employed, then only one or two datasets have been used for the applications. This leads to conclusions that are based on the unique characteristics of such dataset (Abrahart et al., 2008); (iii) Datasets were split into two subsets for training and testing, where the models were tested iteratively using the testing data subset. This means, in fact, that the testing data are used, at least implicitly, during training. In this case, the generalization ability of the developed model is questionable; and (iv) when datasets

HESSD
6, 7055–7093, 2009

were correctly split into three subsets for training, cross-validation, and testing, only one random realization of the three subsets was used. Such use of a single realization of the dataset makes it difficult to assess the predictive uncertainty and the effect of the split approach on the adopted models.

The above-mentioned deficiencies, in addition to other requirements identified by Abrahart et al. (2008) including the need for testing the models over a range of conditions, the reasoning behind the data splitting, and the need for designing repeatable experiments and reproducible findings, are the motives behind this study. The aim of this study is to evaluate and test the predictive abilities of six DDM techniques on five different case studies of rainfall-runoff, evapotranspiration, and soil moisture content. Multiple random realizations of the three subsets of each dataset will be created and used with each and every modeling technique. The techniques will be evaluated against multiple linear regression models and, when applicable, naïve models. Both predictive accuracy and uncertainty will be evaluated. The authors intend to make all datasets used in this study available for all interested researchers to test the results and conduct further studies. The authors hope and aim that this study could serve as a benchmark study for assessing future proposed modeling, optimization, and input processing methods or techniques.

This study is presented in two companion papers. This first part consists of, after this introduction, a section that briefly summarizes some of the key comparative studies in hydrology literature, followed by a section explaining the study methodology and the experimental set up. The fourth section describes the modeling techniques adopted in this study as well as the implementation tools. The fifth section contains a description of study sites, the collected data, and how five different case studies (datasets) representing various hydrological processes were created from three sites. The last section of this first part is a general summary. The second part begins with an introduction section that explains how the methodology was applied and how inputs for the various case studies were selected. The second section reports on the implementation details and parameter values, when applicable, of each modeling techniques for the various

HESSD
6, 7055–7093, 2009

datasets. Results of the various techniques and analysis are presented in the third section. A general discussion and guidelines are presented in Sect. 4, whereas the conclusions and findings of the entire study are presented in the last section.

2 Comparative hydrological modeling studies using data driven modeling techniques

The number of studies that reported some sort of comparison between various DDM techniques in hydrology is very large, and it is beyond the possibility of being summarized here (for presentation of some of the latest advances see, for example, the volume edited by Abrahart et al., 2008). However, some key and representative studies are presented here. Solomatine and Siek (2006) presented an algorithm, which facilitates incorporation of domain knowledge into one particular type of modular model (model tree). They employed the M5flex algorithm to two hourly and daily rainfall-runoff datasets as well as five widely used benchmark datasets – Autompg, Bodyfat, CPU, Friedman, and Housing (Blake and Mertz, 1998). They compared the M5flex method with global ANNs and other local M5 modeling methods (M5, M5opt). They concluded that M5flex delivered high performance because of the use of additional domain knowledge for determining the best split attributes and values. Solomatine and Xue (2004) showed that both M5 model tree technique and ANNs perform similarly for flood forecasting problem in the upper reach of the Huai River in China, but the model trees have certain advantage in terms of transparency in the model structure over ANNs.

Sivapragasm et al. (2007) found that there is no significant difference in the prediction accuracy between GP and ANNs for forecast of daily flows, but GP has an advantage of identifying the optimum inputs. Makkeasorn et al. (2008) compared between genetic programming (GP) and ANN models for forecasting river discharges. The findings indicated that GP-derived streamflow forecasting models were generally favored for forecasting over ANNs. Further, the most forward looking GP-derived models can even perform a 30-day streamflow forecast ahead of time with a reasonable estimation.

HESSD
6, 7055–7093, 2009

Jayawardena et al. (2005) compared the GP technique in modeling rainfall-runoff process to the traditional modelling approaches. They used the GP technique to predict the runoff from three catchments in Hong Kong and two catchments in southern China, and showed that the GP technique evolved simple models that enabled the quantification of the significance of different input variables for prediction. Parasuraman et al. (2007) used two hourly evapotranspiration (ET) datasets to compare between GP and ANNs for prediction of ET. Not much difference was found, with regard to the prediction accuracy, between the two techniques.

Wu et al. (2007) applied a modular SVM model, termed distributed SVR (D-SVR), with two step Genetic Algorithm parameter optimization method, to carry out prediction of water level in a river. The D-SVR method desegregates the couple of subsets from original training set and then generates a local SVR for each subset independently. Wu et al. (2007) evaluated the performance of D-SVR against the predictions from linear regression (LR), nearest neighbor (NN) method, and genetic algorithm-based ANN (ANN-GA) methods. The proposed D-SVR model can predict the water level better in comparison with the other models. However LR model performed better in comparison with NN, ANN-GA models, which was attributed to highly linear mapping relation between input and output variables that restricts the power of NN and ANN. In their study, Lin et al. (2006) employed an SVM model to predict long-term flow discharges in Manwan Hydropower scheme in Tibet. It was found through comparison of results with ARMA and ANN models that the SVM model can provide more accurate predictions of long term flow discharges. Further, Lin et al. (2006) concluded that SVM has its distinct capabilities and advantages in identifying hydrological time series comprising nonlinear characteristics. In their preliminary study, Çimen (2008) applied SVMs for the estimation of suspended sediment concentration/load. The observed streamflow and suspended sediment data of two rivers in the USA, which have been already used in earlier studies using ANNs, were considered. It was found that the negative sediment estimates, which were encountered using ANNs, did not happen during the application of SVMs. Khan and Coulibaly (2006) examined the application of the SVM

and successfully demonstrated the mean monthly lake water level prediction up to 12 months ahead. SVM was found to be more advantageous than ANNs, which prescribes more number of controlling parameters. Khan and Coulibaly (2006) deduced that SVM proved to be more competitive and promising compared to the widely used ANNs and conventional seasonal multiplicative autoregressive (SAR) models. Behzad et al. (2008) compared SVM with ANN and ANN-GA models for prediction of daily runoff of Bakhtiyari River watershed in Iran. They considered available climate information as model inputs. They concluded that the prediction accuracy of SVM was at least as good as that of ANN and ANN-GA models in some cases, and better in some other cases. Furthermore, Behzad et al. (2008) found that SVM converges considerably faster compared to other models. Wu et al. (2008) demonstrated the feasibility of SVM for forecasting of soil water content in Purple hilly area located in Southwest University in Chongqing. They compared the predictions from SVM with ANNs, and showed that the results from the SVM predictor significantly outperformed the other baseline predictors such as ANNs.

Giustolisi and Savic (2006) found that EPR was more accurate than GP for extracting a symbolic expression for Chezy resistance coefficient. Elshorbagy and El-Baroudy (2009) differentiated between equation-based GP and program-based GP. They further compared GP with EPR technique using a highly nonlinear dataset (soil moisture content). It was found that program-based GP outperformed EPR in its prediction accuracy. More importantly, Elshorbagy and El-Baroudy (2009) demonstrated the need for adopting multiple data driven modeling techniques and tools (modeling environments) to obtain reliable predictions. This brief literature review shows that findings and conclusions were sometimes seemingly contradictory. Apparently such findings should be viewed as data-specific, and thus, lacks generality and strong support for cause-effect relationships.

HESSD
6, 7055–7093, 2009

3 Methodology and experimental setup

In order to achieve the objectives of this paper with regard to the comparative predictive performance of various DDM techniques, first, a set of distinctive modeling techniques were identified. The selected techniques are (i) artificial neural networks (ANNs); (ii) genetic programming (GP); (iii) evolutionary polynomial regression (EPR); (iv) support vector machines (SVM); (v) M5 model trees; and (vi) *K*-nearest neighbors (*K*-nn). To facilitate the comparison and allow for performance evaluation in light of easily understandable and widely recognized techniques, multiple linear regression (MLR) models and/or naïve models were employed as base line references.

Second, five different case studies representing different hydrological processes or variables (actual evapotranspiration, soil moisture content, and rainfall-runoff) were selected. The datasets present a wide range of challenges to data driven techniques because of their various levels of complexity, embedded feedback mechanism, and nonlinearity. The datasets will be explained in more details in a later section of this paper. Third, for each dataset, model inputs were either identified in this research or were pre-selected based on previous studies. Even though appropriate model inputs were secured for this study, the identification of the optimum inputs was not given an extraordinary emphasis since the focus of this research is inter-technique comparison. As long as the inputs are the same for the various modeling techniques, an unbiased analysis can be conducted toward achieving the objectives of this study.

Fourth, split samples from each dataset were prepared for the modeling experiment. Each set of the five datasets was randomly sampled 100 times without replacement, such that every time the dataset is split into three distinct subsets: training, which contains one half of the total data instances; cross-validation, which contains one sixth of the data instances, and testing, which contains one third of the data instances. Twelve different groups (three subsets each) out of the 100 groups were selected based on the statistical properties of the output variable (e.g., runoff). The aim was to select the samples where the mean and the standard deviation values of the three subsets (train-

HESSD
6, 7055–7093, 2009

ing, cross validation, and testing) are similar or, at least, the differences are minimal. The cross-validation subset was used for stopping the model training and selecting the best model, whereas the testing subset was kept completely unseen during the training process. Twelve different models were developed based on the 12 data groups (the best model based on cross-validation was picked every time), and each model was tested using the corresponding testing subset. These procedures were repeated using the six different data driven modeling techniques, applied to each of the five different datasets. The results of this experiment allows for investigating ensemble outputs from each modeling techniques, average and range of possible prediction accuracy, and predictive uncertainty.

Fifth, the predictive accuracy of the various models and techniques were evaluated using the root mean squared error (RMSE), the mean absolute relative error (MARE), the mean bias (MB), and the correlation coefficient (R). The authors believe that these four error statistics, along with the visual comparison between observed and predicted values, are sufficient to reveal any significant differences among the various modeling techniques with regard to their predictive accuracy. The formulae of the error measures are presented in Eqs. (1–4) below.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(O_i - P_i)^2}{N}} \tag{1}$$

$$\mathrm{MARE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{O_i - P_i}{O_i}\right| \tag{2}$$

$$\mathrm{MB} = \frac{1}{N}\sum_{i=1}^{N}(O_i - P_i) \tag{3}$$

HESSD
6, 7055–7093, 2009

$$R = \frac{\sum_{i=1}^{N}(O_i - \overline{O_i})(P_i - \overline{P_i})}{\sqrt{\sum_{i=1}^{N}\left(O_i - \overline{O_i}\right)^2 \sum_{i=1}^{N}\left(P_i - \overline{P_i}\right)^2}} \quad (4)$$

Where N represents the number of instances presented to the model; O_i and P_i represent observed and predicted counterparts; and $\overline{O}$ and $\overline{P}$ represent the mean of the corresponding variables. However, sometimes conflicting results regarding the performance of various models may arise due to the use of various error measures (Dawson et al., 2007; Elshorbagy et al., 2000). In this study, a supplemental error measure that combines the effects of the four error measures in one indicator is proposed. The new indicator, called the ideal point error (IPE) is based on identifying the ideal point in the four dimensional space that each model aims to reach. The coordinates of the ideal point should be: (RMSE=0.0; MARE=0.0; MB=0.0; R=1.0). The IPE (Eq. 5) measures how far the model is from the ideal point. All individual error measures are given equal relative weights, and all are normalized using the maximum error so that the final IPE value for each model ranges between 0.0 for the best model and 1.0 for the worst model.

$$\mathrm{IPE} = \left\{0.25\left[\left(\frac{\mathrm{RMSE}_{ij} - 0.0}{\max\mathrm{RMSE}_{ij}}\right)^2 + \left(\frac{\mathrm{MARE}_{ij} - 0.0}{\max\mathrm{MARE}_{ij}}\right)^2 + \left|\frac{\mathrm{MB}_{ij} - 0.0}{\max|\mathrm{MB}_{ij}|}\right|^2 + \left(\frac{R_{ij} - 1.0}{1/\max R_{ij}}\right)^2\right]\right\}^{1/2} \quad (5)$$

Where i denotes model (i) and j denotes technique (j).

Sixth, the predictive uncertainty of the models was assessed using the model residuals (r values), where r_i is the difference between the observed and the predicted values. For each dataset and each modeling technique, the residuals are computed for all 12 models representing the range of possible residuals. The residuals of the

HESSD
6, 7055–7093, 2009

12 models are merged in one set of presumably random variable, and a probability distribution was fit to this variable.

Seventh, the gamma test was conducted to assist in gaining some insight into the predictability of the output variables using nonlinear smooth functions, and possibly some leads into the process of selecting appropriate modeling techniques for a particular case study. The main idea of the gamma test (Γ-test) is estimating the variance of the noise on the output variable, which could be an estimate of the best mean squared error that a smooth model can achieve for the corresponding output. The test was implemented using *winGamma* (Jones et al., 2001) that assumes that non-determinism in a smooth model from inputs to outputs is due to the presence of statistical noise on the outputs:

$$y = f(X_1 \ldots X_m) + \varepsilon \tag{6}$$

Where f is a smooth function and ε is noise, and that the variance of the noise Var(ε) is bounded. The Γ-test is based on $L[i,k]$, which are k nearest neighbors $X_{L[i,k]}$ ($1 \leq k \leq p$) for each vector X_i ($1 \leq i \leq N$) (Stefánsson et al., 1997). Delta (δ) and γ functions can be defined as follows:

$$\delta_N(k) = \frac{1}{N}\sum_{i=1}^{N} |X_{L(i,k)} - X_i|^2 \quad (1 \leq k \leq p) \tag{7}$$

$$\gamma_N(k) = \frac{1}{2N}\sum_{i=1}^{N} |y_{L(i,k)} - y_i|^2 \quad (1 \leq k \leq p) \tag{8}$$

Where $y_{L(i,k)}$ is the corresponding output value for the k nearest neighbors of X_i in Eq. (7) (Stefánsson et al., 1997). A least squares regression line can be constructed for the p points ($\delta_N(k)$, $\gamma_N(k)$) where Γ can be computed:

$$\gamma = A\delta + \Gamma \tag{9}$$

The intercept on the vertical axis is the Γ value (Jones et al., 2001). As $\delta_N(k)$ approaches zero, $\gamma_N(k)$ approaches Var(ε) in probability. In addition to Γ, three other

HESSD
6, 7055–7093, 2009

useful statistics can be calculated: (i) the *gradient*, which is the slope of the regression line that indicates the complexity of the system (steeper gradient indicates greater complexity) (Evans and Jones, 2002), (ii) the *V-ratio*, which is a scale invariant noise estimate where Γ is divided by the variance of the output variable. A *V-ratio* close to zero indicates high degree of predictability of the output variable, and (iii) the M-test, which is the size of data that is possibly required to produce a stable asymptote of Γ. The Γ value might be estimated for scaled or unscaled dataset, but the *gradient* will be more informative if estimated based on scaled dataset. In general, if the inputs have inconsistent units, it is advisable to conduct the Γ-test using the scaled data (Jones et al., 2001).

4 The modeling techniques and tools

4.1 Artificial neural networks (ANNs)

ANN is a method of computation and information processing motivated by the functional units of the human brain, namely neurons. Since abundant information on ANNs is available in literature (e.g., Haykin, 1999; ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000), the description of ANNs herein is brief, and limited to the needs of this study. According to Haykin (1999), a neural network is a massively parallel distributed information processing system that is capable of storing the experiential knowledge gained by the process of learning, and of making it available for future use. Mathematically, ANNs are universal approximators with an ability to solve large-scale complex problems such as time series forecasting, pattern recognition, nonlinear modeling, classification, and control. This is achieved by identifying the relationships among given patterns.

Feedforward neural networks (FFNNs) are the most widely adopted network architecture for the prediction and forecasting of hydrological variables (Minns and Hall, 1996; Maier and Dandy, 2000; Dibike and Solomatine, 2001). Typically, FFNNs con-

sist of three layers: input layer, hidden layer, and output layer. The number of nodes in the input layer corresponds to the number of inputs considered for modeling the output. The input layer is connected to the hidden layer with weights that determine the strength of the connections. The number of nodes in the hidden layer(s) indicates the complexity of the problem being modeled. The hidden layer nodes consist of the activation function, which helps in nonlinearly transforming the inputs into an alternative space where the training samples are linearly separable (Brown and Harris, 1994). Detailed review of ANNs and their application in hydrology can be found in Maier and Dandy (2000) and in ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000).

The FFNNs adopted in this study make use of the tan-sigmoidal activation function in the hidden layer and the linear activation function in the output layer. While the tan-sigmoidal activation function squashes the input between −1 and 1, the linear activation function calculates the neurons output by simply returning the value passed to it. One of the important issues in the development of neural networks model is the determination of optimal number of hidden neurons that can satisfactorily capture the nonlinear relationship existing between the input variables and the output. The number of neurons in the hidden layer is usually determined by trial-and-error method with the objective of minimizing the cost function (typically, the error on cross-validation dataset) (ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000). Levenberg-Marquardt back propagation algorithm is used for training the FFNNs in this study.

4.2 Genetic programming (GP)

Genetic Programming (GP), introduced by Koza (1992), is an evolutionary algorithm based on the concepts of natural selection and genetics. GP extends the search of genetic algorithms for optimal set of parameters search to include the model space, so that both the model structure and the associated model parameters can be optimized simultaneously. Genetic symbolic regression (GSR) is a special application of GP in the

HESSD
6, 7055–7093, 2009

area of symbolic regression, where the objective is to find a mathematical expression in symbolic form, which provides an optimal fit between a finite sample of values of the independent variable and its associated values of the dependent variable (Koza, 1992). GSR can be considered as an extension of numerical regression problems, where the objective is to find the set of numerical coefficients that best fits a predefined model structure (linear, quadratic, or polynomial). Nevertheless, GSR does not require the functional form to be defined a priori, as GSR involves finding the optimal mathematical expression in symbolic form (both the discovery of the correct functional form and the appropriate numerical coefficients) that defines the predictand-predictor relationship. GSR is sometimes referred to as equation-based GP. Another form of GP is program-based GP, where the explicit equation may not be necessarily produced, but rather a program (code) is the final output. Elshorbagy and El-Baroudy (2009) noted that program-based GP can be more effective than equation-based GP with regard to its prediction accuracy. GPLAB (Silva, 2005), a GP toolbox for MATLAB that provides the evolved equation in the form of a parse tree is an example of an equation-based GP tool, whereas Discipulus (Francone, 2001), used in this study, is an example of a program-based GP tool.

Genetic Programming (GP) is a widely used machine learning (ML) technique; it uses a tree-like structure, as decision trees, to represent its concepts and its interpreter as a computer program. Therefore, some authors even considered it to be a superset of all other ML representations; this may enable GP to produce any solution that is produced by any other ML system (Banzhaf et al., 1998). It uses different genetic operators such as crossover and mutation, together with beam search to reach candidate solutions from the overall population of solutions. Although GP is computationally intensive, like most soft-computing techniques, and has its own limitations. The major problem is the deterioration of the prediction ability of the developed model with longer prediction horizon, which is a common problem in any modeling method. The adverse consequences of this problem can be mitigated by combining GP technique with knowledge-based techniques that depend on the accumulated knowledge of the

HESSD
6, 7055–7093, 2009

process under consideration. This will enhance the quality of the developed models and add to the understanding of the complicated hydrological processes (Babovic and Keijzer, 2002).

Several applications of the GP technique in hydrology exist in the literature. Parasuranam et al. (2007a) explored the utility of GP to develop explicit models for eddy covariance-measured actual evapotranspiration. Babovic and Keijzer (2002) addressed the utility of GP in developing rainfall-runoff models on the basis of hydrometeorological data, as well as in combination with other conventional models, i.e. conceptual models. It was reported that the GP models gave more insights into the functional relationships between different input variables resulting in more robust models. Parasuraman et al. (2007b) used GP to evolve pedotransfer functions (PTFs) for estimating the saturated hydraulic conductivity (K_s) from soil texture (sand, silt, and clay) and the bulk density. Similarly, Jayawardena et al. (2005) compared the GP technique in modeling rainfall-runoff process to the traditional modeling approaches. They used the GP technique to predict the runoff from three catchments in Hong Kong and two catchments in southern China, and showed that the GP technique evolved simple models that enabled the quantification of the significance of different input variables for prediction. In literature, there was an emphasis on GP's ability to produce explicit equations, but in this research program-based GP is employed to utilize the full predictive ability of the technique.

For GP implementation, the first step is to define the functional and terminal sets, along with the objective function and the genetic operators. The functional set and the terminal set are the main building blocks of GP, and hence, their appropriate identification is central in developing a robust GP model. The functional set consists of basic mathematical operators $\{+, -, *, /, \sin, \exp, \ldots\}$ that may be used to form the model. The choice of the operators considered in the functional set depends upon the degree of complexity of the problem to be modeled. The terminal set consists of independent variables and constants. The constants can either be physical constants (e.g. Earth's gravitational acceleration, specific gravity of fluid) or randomly generated constants.

HESSD
6, 7055–7093, 2009

Different combinations of functional and terminal sets are used to construct a population of mathematical models (or programs). Each model (individual) in the population can be considered as a potential solution to the problem. Genetic operators include crossover and mutation, and they are discussed in detail later in this section. Once the functional and terminal sets are defined, the next step is to generate the initial population for a given population size. The initial population can be generated in a multitude of ways, including, the full method, grow method, and ramped half-and-half method. The ramped half-and-half method is a combination of the full and the grow methods. For each depth level considered, half of the individuals are initialized using the full method and the other half using the grow method. The ramped half-and-half method is shown to produce highly diverse trees, both in terms of size and shape (Koza, 1992), and thereby provides a good coverage of the search space. More information on the different methods of generating the initial population can be found in Koza (1992). Once initialized, the fitness of each individual (mathematical model) in the population is evaluated based on the selected objective function. The better the fitness of an individual, the greater is the chance of the individual breeding into the next generation. In this study, root mean squared error is used as the objective function, and a lower value of RMSE indicates better fitness. At each generation, new sets of models are evolved by applying the genetic operators: crossover and mutation (Koza, 1992; Babovic and Keijzer, 2000). These new models are termed offspring, and they form the basis for the next generation.

After the fitness of the individual models in the population is evaluated, the next step is to carry out selection. The objective of the selection process is to create a temporary population called the mating pool, which can be acted upon by genetic operators: crossover and mutation. Selection can be carried out by several methods like truncation selection, tournament selection, and roulette wheel selection. As roulette wheel selection is one of the most commonly used methods including Koza (1992), it has been adopted in this study. Roulette wheel is constructed by proportioning the space in a roulette wheel based on the fitness of each model in the population. The selection

HESSD
6, 7055–7093, 2009

process ensures that the models with better fitness have more chance of breeding into the next generation. Crossover is carried out by initially choosing two parent models from the mating pool, and selecting random crossover points for each of the parents. Based on the selected crossover points, the corresponding sub-tree structures are swapped between the parents to produce two different offspring with different characteristics. The number of models undergoing crossover depends upon the chosen probability of crossover (P_c). Mutation involves random alteration of the parse tree at the branch or node level. This alteration is done based on the probability of mutation (P_m). For an overview of different types of computational mutations, readers are referred to Babovic and Keijzer (2000). While the role of the crossover operator is to generate new models, which did not exist in the old population, the mutation operator guards the search against premature convergence by constantly introducing new genetic material into the population.

4.3 Evolutionary polynomial regression (EPR)

Evolutionary Polynomial Regression (EPR) is another data driven and soft computing technique that models time series or regression-type data containing information about physical processes (Giustolisi and Savic, 2006). EPR combines the power of evolutionary algorithms with numerical regression to develop polynomial models combining the independent variables together with the user-defined function as follows (Laucelli et al., 2005):

$$\hat{Y} = \sum_{i=1}^{m} F(X, f(x), a_i) + a_0 \tag{10}$$

where $\hat{Y}$ is the EPR-estimated dependent variable, $F(\cdot)$ is the polynomial function constructed by EPR, X is the independent variables' matrix, $f(\cdot)$ is a user-defined function, a_i is the coefficient of the i-th term in the polynomial, a_0 is the bias and m is the total number of the polynomial terms. Inclusion of the user-defined function is provided to

enhance the characterization of the response (dependant) variable. As the developers of the EPR tool state "EPR is a two-stage technique for constructing symbolic models: (i) structure identification; and (ii) parameter estimation", where it uses genetic algorithm (GA) simple search method to search in the model structure space. EPR uses the least squares (LS) method to estimate the parameters of the selected model structure based on the performed GA search. Applications of EPR are found in Savic et al. (2006), Doglioni et al. (2008), Elshorbagy and el-Baroudy (2009), and Giustolisi et al. (2007). The search proceeds by using the standard GA operators, crossover and mutation; noting that this type of search is not exhaustive as it is practically impossible to conduct such search on an infinite search space (Laucelli et al., 2005). Even though EPR might be viewed as a subset of GP, its reported good performance while emphasizing the polynomial structure makes it a potential candidate for this study.

This study makes use of the EPR toolbox (Laucelli et al., 2005), which is based on "homonymous modeling methodology based on a hybrid evolutionary paradigm". It is a multi-objective implementation of EPR in the sense that it produces several models, which are the best trade-off, considering fitness to training data vs. parsimony. The EPR tool performs three types of regression, i.e. dynamic, static, and classification. Dynamic modeling is used to model systems that have memory, or in other words, when the present state of the system depends on the previous states of other input variables. On the other hand, static systems are systems that are not influenced by the previous states of input variables. Classification modeling is a special type of static modeling in which the model output is an integer (Laucelli et al., 2005). The readers may refer to the user manual for the details of the EPR toolbox and the different components of its simple interface (Laucelli et al., 2005).

4.4 Support vector machine (SVM)

The foundation for the subject of Support Vector Machines has been largely developed by Vapnik in the 1960s and 1970s (Vapnik, 1998; see also Cherkassky and Mulier, 2007) and it is now gaining popularity due to many attractive features. Its formulation

embodies the Structural Risk Minimisation (SRM) principle, which has been shown to be superior to the traditional Empirical Risk Minimisation (ERM) principle, employed by many of the other modelling techniques. SRM minimises an upper bound on the expected risk, as opposed to ERM that minimises the error on the training data. It is this difference that is claimed to provide SVM with a greater ability to generalise, which is a principal goal in statistical learning.

SVM algorithm was first developed to solve the classification problem, but was extended to the domain of regression problems. In regression and time series prediction applications, excellent performances were obtained (Müller et al., 1997; Mattera and Haykin, 1999; Dibike et al., 2001). The goal of ε-SV regression (Vapnik, 1995) is to find a function $f(x)$ that has at most ε deviation from the actually obtained targets y_i for all the training data, and at the same time, is as flat as possible. In case of linear functions f,

$$f(x) = \langle w, x \rangle + b \tag{11}$$

Where $\langle \cdot, \cdot \rangle$denotes the dot product in X. Flatness in this case means seeking small w, which can be ensured by minimizing the Euclidean norm, i.e., $\|w\|^2$. Sometimes, it is not possible to approximate all pairs (x_i, y_i) with ε precision. So, it is possible to allow for some errors in the form of slack variables ζ_i, ζ_i^*. The problem can be written as a convex optimization problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\zeta_i + \zeta_i^*) \\ &\text{subject to} \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \zeta_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* \geq 0 \end{cases} \end{aligned} \tag{12}$$

The constant $C>0$ determines the tradeoff between the flatness of f and the amount up to which deviations larger than ε are tolerated. Figure 1 presents an example of

HESSD
6, 7055–7093, 2009

SV regression with the ε-tube in which errors are ignored – leading to better model generalization.

A Lagrange function from both the objective function and the corresponding constraints can be constructed by introducing a dual set of variables (M"uller et al., 1997):

$$L := \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\zeta_i + \zeta_i^*) - \sum_{i=1}^{l}\alpha_i(\varepsilon + \zeta_i - y_i + \langle w, x_i\rangle + b)$$
$$- \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \zeta_i^* + y_i - \langle w, x_i\rangle - b) - \sum_{i=1}^{l}(\eta_i\zeta_i + \eta_i^*\zeta_i^*) \quad (13)$$

where α_i, α_i^*, η_i, $\eta_i^* \geq 0$. Finally, w can be written as follows:

$$w = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i \quad \text{and therefore} \quad f(x) = \left(\alpha_i - \alpha_i^*\right)\langle x_i, x\rangle + b \quad (14)$$

This is called Support Vector expansion, i.e. w can be completely described as a linear combination of the training patterns x_i. The above discussion is based only on linear SVM regression. For nonlinear regression, the SVM has a great advantage that can represent the nonlinear function in an arbitrary number of dimensions efficiently through a defined Kernel. The idea is to map the training input vector x_i into a higher dimensional space (called feature space) or hyperplane, by the function Φ, while the regression for $x$remains linear. Thus, the procedure is the same as the linear SVM except changing the dot product $\langle x_i, x\rangle$ by $\langle\Phi(x_i), \Phi(x)\rangle$. The Kernel function: $K(x_i, x)=\langle\Phi(x_i), \Phi(x)\rangle$ can assume any form. Many Kernels are being proposed by researchers; however, the most common ones are:

Linear Kernel:	$K(x_i, x) = \langle x_i, x\rangle$	
Polynomial Kernel:	$K(x_i, x) = (\gamma\langle x_i, x\rangle + \tau)^d$,	$\gamma > 0$
Radial basis function Kernel:	$K(x_i, x) = \exp(-\gamma\|x_i - x\|^2)$,	$\gamma > 0$
Sigmoid Kernel:	$K(x_i, x) = \tanh(\gamma\langle x_i, x\rangle + \tau)$,	$\gamma > 0$

HESSD
6, 7055–7093, 2009

Where γ, τ, and d are Kernel parameters.

In this study, the SVM implementation within WEKA 3.6.0 Software (Bouckaert et al., 2008; Witten and Frank, 2005) has been used.

4.5 Model trees

Model trees (or M5 model trees) are relatively new machine learning technique introduced by Quinlan (1992) who also suggested the algorithm that uses information theory to build them – the M5 algorithm. This is effectively a piece-wise linear regression model. A complex modelling problem can be solved by dividing it into a number of simple tasks and building simple model for each of them.

A model tree (MT) belongs to a class of modular models, which uses the "hard" (i.e. yes-no) splits of input space into regions progressively narrowing the regions of the input space. Thus model tree is a hierarchical (or tree-like) modular model that has splitting rules in non-terminal nodes and the expert models at the leaves of the tree. In M5 model trees, the expert models are simple linear regression equation derived by fitting to the non-intersecting data subsets. Once these models are formed recursively in the leaves of the hierarchical tree, then prediction with the new input vector consists of the two steps: (i) attributing the input vector to a particular subspace by following the tree; and (ii) running the corresponding model. Brief description of model tree algorithm is presented below.

The M5 algorithm for inducing a model tree was developed by Quinlan (1992). The first step in building a model tree is to determine which input variable is the best to split the training set. The splitting criterion (i.e. selection of the input variable and splitting value of the input variable) is based on treating the standard deviation of the target values that reach a node as a measure of the error at that node, and calculating the expected reduction in error as a result of testing each input variable at that node. The expected error reduction, which is called standard deviation reduction, SDR, is

HESSD
6, 7055–7093, 2009

calculated by

$$\mathrm{SDR} = \mathrm{sd}(T) - \sum_i \frac{|T_i|}{|T|}\mathrm{sd}(T_i) \quad (15)$$

where, T represents set of examples that reach the splitting node, T_1, T_2, ..., represents the subset of T that results from splitting the node according to the chosen input variable, sd represents standard deviation, $|T_i|/|T|$ is the weight that represents the fraction of the examples belonging to subset T_i.

After examining all possible splits by exhaustive search, M5 chooses the one that maximizes SDR. The splitting of the training examples is done recursively to the subsets. The splitting process terminates when the target values of all the examples that reach a node vary only slightly, or only a few instances remain (this is a user-defined parameter). This division often produces over-elaborate structures leading to overfitting models. They can be pruned back, for instance by replacing a subtree with a single model in a leaf. Additionally, "smoothing" may be also performed to compensate for the sharp discontinuities that will inevitably occur between the adjacent linear models at the leaves of the pruned tree. In smoothing, the outputs from adjacent linear equations are updated in such a way that their difference for the neighboring input vectors belonging to the different leaf models will be smaller. Details of the pruning and smoothing process can be found in Witten and Frank (2000). Figure 2 presents an example of model tree.

As compared to other machine learning techniques, model tree learns efficiently and can tackle tasks with very high dimensionality – up to hundreds of variables. The main advantage of model tree is that results are transparent and interpretable. During the last years several authors have shown the effectiveness of the M5 machine learning method in rainfall-runoff modelling (see, e.g., Solomatine and Dulal, 2003; Solomatine and Siek, 2006; Stravs and Brilly, 2007).

HESSD
6, 7055–7093, 2009

4.6 *K*-nearest neighbors

The *K*-nearest neighbors (*K*-nn) technique is one of the simplest forms of instance-based learning, which can be treated as plain memorization (Witten and Frank, 2005). Once a set of training instances has been memorized, one encountering a new (testing) instance, the memory is searched for the training instance that most closely resembles the testing instance. Instead of creating rules (or continuous function approximation surface), *K*-nn technique works directly from the examples themselves. Each new instance is compared with existing ones using a distance metric, and the closest existing distance is used to assign the output to the new instance. Usually, more than one nearest neighbors is used. Standard Euclidean distance (or any other distance measure) is used as a metric to represent "resemblance". When multiple nearest neighbors are employed, the output of the testing instance can be based either on simple average, weighted average, or any more sophisticated function. In this study, the simplest method, which is the average value of the *K*-nearest neighbors, is used. An apparent drawback to instance-based representation is that it does not make explicit the structures that are learnt. Instances do not really describe the patterns in data. Karlsson and Yakowitz (1987); Parasuraman and Elshorbagy (2007); and Solomatine et al. (2008) presented some hydrological prediction case studies using *K*-nn technique.

5 Datasets

5.1 Actual evapotranspiration

The eddy covariance (EC)-measured actual evapotranspiration data from the South West Sand Storage (SWSS) facility, located near Ft. McMurray, Alberta, Canada, is considered in this study. The SWSS is currently the largest operational tailings dam in the world, holding approximately 435 million cubic meters of material, covering 25 km^2, and standing approximately 40 m high with a 20H:1V side-slope ratio. Soils consist of

HESSD
6, 7055–7093, 2009

mine tailings sand overlain with 0.4 to 0.8 m of topsoil that is a mixture of peat and secondary mineral soil with a clay loam texture. Both vegetation species and composition vary across the SWSS, with dominant groundcover including horsetail (*Equisetum arvense*), fireweed (*Epilobium angustifolia*), sow thistle (*Sonchus arvense*), and white and yellow sweet clover (*Melilotus alba*, *Melilotus officinalis*). Tree and shrub species include Siberian larch (*Larix siberica*), hybrid poplar (*Populus* sp. hybrid), trembling aspen (*Populus tremuloides*), white spruce (*Picea glauca*), and willow (*Salix* sp.). For the SWSS facility, the ground-water table is located well below the rooting zone, at a depth between 0.8–1.0 m, and hence do not directly contribute to the evapotranspiration process. Accurate estimation of actual evapotranspiration from the reconstructed watersheds is of vital importance as it plays a major role in the water-balance of the system, which links directly to ecosystem restoration strategies. The weather station located on top of the SWSS facility measured the air temperature (AT) (°C), ground temperature (GT) (°C), net radiation (NR) (W/m^2), relative humidity (RH), and wind speed (WS) (m/s). Turbulent fluxes of heat and water vapor were measured using a CSAT3 sonic anemometer and thermometer (Campbell Scientific) and an LI-7500 CO2/H2O gas analyzer (Li-Cor). Ground heat flux was measured using a CM3 radiation and energy balance (REBS) ground heat flux plate placed at 0.05 m depth. In EC technique, the covariance of vertical wind speed with temperature and water vapor is used to estimate the sensible heat (H) and latent heat (LE) fluxes (Parasuraman and Elshorbagy, 2008). More information on the EC technique can be found in Drexler et al. (2004). Raw turbulence measurements were made at 10 Hz and fluxes were calculated using 30 min block averages with a 2-D coordinate rotation.

The half hourly EC-measured LE flux (the product of the latent heat of vaporization and evapotranspiration) at the SWSS facility for two growing seasons (from 3 May to 21 September 2005 and from 27 May to 9 September 2006) is considered in this study. The total precipitation during the two seasons is 275 mm and 265 mm, respectively and the average day-time reference evaporation rate is 0.27 mm/h. Nevertheless for modeling purposes, the day time (08:00h–20:00 h) evapotranspiration alone is considered.

HESSD
6, 7055–7093, 2009

After eliminating records of missing data, the remaining number of data instances were 5307 data points. Since evapotranspiration is commonly perceived as being highly dependent on climatic variables, the EC-measured LE flux is modeled as a function of NR, AT, GT, RH, and WS, as well as possible combinations of these variables. The descriptive statistics of the datasets used for training, cross validation, and testing are presented in Table 1. The coefficient of variation (CV) of different variables during training, cross validation, and testing are comparable.

5.2 Soil moisture content

Over the years, several large scale soil cover (reconstructed watersheds) experiments are being conducted to assess the performance of different reclamation strategies in northern Alberta, Canada, by studying the basic mechanisms that control the moisture movement within these covers. In particular, three experimental soil covers (D1, D2, and D3) were established in the year 1999. The experimental covers were constructed over the saline-sodic overburden with thickness of 0.50 m, 0.35 m, and 1.0 m, comprising a thin layer of peat mineral mix over varying thickness of secondary (glacial/till) soil. Cover D1 consists of 20 cm of peat overlying 30 cm of till, and it is considered for this study. The soil cover has an area of 1 ha (approximately 200 m long and 50 m wide), with a 5:1 slope (5 horizontal to 1 vertical). This reconstructed watershed, compared to natural watersheds, is not stable during their initial stages, and hence evolves over time to achieve hydro-sustainability. In order to track the evolution (hydrological changes) of the watershed, intensive instrumentations were installed in the watershed. Each watershed has an individual soil station located at the middle of the slope, which measures the volumetric soil moisture content of the upper peat (SMP) and the lower till (SMT) layers, twice a day. Soil moisture is measured using TDR principles with model CS615 (Boese, 2003). The TDR sensors were installed laterally into the soil profile. Watershed D1 has eight TDR sensors installed over a depth range of 0.05 m to 1.00 m. Hourly values of soil temperature of peat (STP) and till (STT) layers are measured using thermisters buried in the watershed at the depth ranges corresponding to the TDR

HESSD
6, 7055–7093, 2009

sensors. Consequently, D1 has eight soil temperature sensors. A weather station located in the mid-slope measures air temperature (AT), and precipitation (P). Similarly, Bowen station located at the mid-slope measures net-radiation (NR) and energy fluxes. All the meteorological variables are measured in an hourly scale. More details on the field instrumentation program and the data collected can be found in Boese (2003) and Elshorbagy et al. (2007).

Average daily values of precipitation, air temperature, soil temperature (STP and STT), net radiation (NR), soil moisture (SMP and SMT) as well as possible combinations of them, are considered for modeling purposes. The ground temperature and soil moisture contents are depth averaged for each layer (upper peat and lower till). As the soil stratum is frozen during the winter, only summer (May–September) time data of years 2000 till 2006 are considered. The total number of instances available for modeling purposes was 972 data points. As the reconstructed watersheds evolve over time to achieve hydro-sustainability, the freeze-thaw cycles and decomposition of highly organic peat layer increases the porosity of the soil and consequently increasing infiltration rates (Haigh, 2000). Hence, modeling the moisture dynamics of such evolving watersheds would be adding to the already challenging task of modeling soil moisture. The descriptive statistics of the datasets used for training, cross validation, and testing are presented in Table 2 for the peat and the till layer datasets, respectively. For modeling purposes, two datasets were generated from the site; one for predicting SMP and the other for SMT. The same set of inputs was used in both datasets. The coefficient of variation (CV) of different variables during training, cross validation, and testing are comparable (Table 2).

5.3 Rainfall-runoff

The rainfall-runoff dataset used in this study is taken from the Ourthe subcatchment, which is a sucatchment of River Meuse The river basin covers part of France, Belgium and The Netherlands (Fig. 3). The area analyzed in this research is approximately 22 000 km^2, from Borgharen-dorp (Jeker basin on the Netherlands border) to Meuse

HESSD
6, 7055–7093, 2009

source-St Mihiel (Lorraine basin in France). This meso-scale catchment system has been widely explored with data driven and expert knowledge (de Wit, 2001; Tu et al., 2005).

The greater part of the discharge of the River Meuse is supplied by its tributaries. Groundwater, precipitation and artificial extractions constitute the discharge to a smaller extent (de Wit, 2001). The Meuse has a great number of tributaries, varying greatly in their sizes. The largest is the Ourthe, with a contributing area of 3626 km^2. The Ourthe subcatchment has great discharges rising fast. Through its nature and situation, close to the Dutch border, the Ourthe is also the most important Meuse tributary for flood forecasts. In its upper course, the Ourthe consists of two branches: the Ourthe Occidentale and the Ourthe Orientale, merging near Nisramont. Near Comblain-au-Pont, the Amblève joins the Ourthe and near Angleur the Ourthe also receives the Vesdre. Measuring from the source of the Ourthe Occidentale, the Ourthe is approximately 175 km long.

The average travel time from upstream to downstream is one day Berger (1992). Mode information about the hydrological properties of the basin and the data validation are referred to Berger (1992) and De Wit (2007). The daily rainfall and runoff data of the Ourthe subcatchment from 11 January 1988 till 31 December 1998 (4008 data points) were used for modeling purposes in this study. Two distinct datasets were created: (i) the first is a dataset where only rainfall data were used as model inputs to predict the runoff; and (ii) the second is the same dataset but the preceding time step ($t-1$) runoff, in addition to the rainfall data, were used as inputs to predict the runoff at time t. The descriptive statistics of the variables that are used as model outputs in this study are presented in Table 3.

6 Summary

Data driven modeling techniques, and in particular soft computing techniques, have addressed and solved many issues in hydrological modeling but also caused ques-

tions and concerns to be raised. The most important concerns are regarding the way DDM techniques are handled, compared, and evaluated and the basis on which findings and conclusions were drawn. The sub-optimal approach in designing modeling experiments, the use and the split of datasets, the exclusive use of techniques and case studies, and writing research articles from the standpoint of advocating certain techniques have contributed to the problem. In this first part of two-part paper, a concise but comprehensive review of key articles that presented comparisons among various data driven modeling techniques was summarized. It was concluded that findings were usually dataset-specific, to some extent contradictory, and thus, difficult to generalize. A comprehensive data driven modeling experiment was proposed and explained. Six data driven modeling techniques, namely, neural networks, genetic programming, evolutionary polynomial regression, support vector machines, M5 model trees, and *K*-nearest neighbors were proposed and briefly explained. Multiple linear regression and naïve models were also suggested as baseline for comparison with the various techniques.

Five different case studies representing three different hydrological processes or variables (evapotranspiration, soil moisture, and rainfall-runoff) from Canada and Europe were described and proposed for the modeling experiment. The central step of the methodology is creating 12 different realizations (groups) from each dataset by random sampling. Each group contains three subsets; training, cross-validation, and testing. Each technique was proposed to be applied to each of the 12 groups of each dataset. This methodology was designed to evaluate both predictive accuracy and uncertainty of the various techniques on a wide range of possibilities that allow for comprehensive testing the modeling capabilities of these techniques. The second paper addresses the application of the proposed methodology through the input selection and the implementation of the various techniques. Results, analysis, and discussion of the findings of this study are presented in the second paper as well.

HESSD

6, 7055–7093, 2009

References

Abrahart, R., See, L., and Solomatine, D.: Practical Hydroinformatics. Computational Intelligence and Technological Developments in Water Applications, Springer-Verlag, Berlin, Heidelberg, 505 pp., 2008.

Abrahart, R., See, L., and Dawson, C.: Neural network hydroinformatics: maintaining scientific Rigour, in: Practical Hydroinformatics, in: Computational Intelligence and Technological Developments in Water Applications, edited by: Abrahart, R., See, L., and Solomatine, D., Springer-Verlag, Berlin, Heidelberg, 33–47, 2008.

ASCE Task Committee on Application of Artificial Neural Networks in hydrology: artificial neural networks in hydrology. I: Preliminary concepts, J. Hydrol. Eng., 5(2), 115–123, 2000.

Babovic, V. and Keijzer, M.: Rainfall-runoff modelling based on genetic programming, Nordic Hydrol. J., 33(5), 331–346, 2002.

Babovic, V. and Keijzer, M.: Genetic programming as model induction engine, J. Hydroinform., 2(1), 35–60, 2000.

Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D.: Genetic Programming – an Introduction: On the Automatic Evolution of Computer Programs and its Applications, Morgan Kaufmann Publishers, Inc., 1998.

Behzad, M., Asghari, K., Eazi, M., and Palhang, M.: Generalization performance of Support Vector Machines and Neural Networks in Runoff Modeling. Expert Systems with Applications, 36(4), 7624–7629, 2009.

Berger, H. E. J.: Flow Forecasting for the River Meuse, Ph.D. Thesis, Technische Universiteit Delft, 1992.

Boese, K.: The design and installation of a field instrumentation program for the evaluation of soil-atmosphere water fluxes in a vegetated cover over saline/sodic shale overburden, M.Sc. thesis, University of Saskatchewan, Saskatoon, Sask., 2003.

Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., and Scuse, D.: WEKA Manual for version 3.6.0. University of Waikato, Hamilton, New Zealand, 2008.

Brown, M. and Harris, C.: Neurofuzzy Adaptive Modeling and Control, Prentice Hall, New York, 1994.

Cherkassky, V., Krasnopolsky, V., Solomatine, D., and Valdes, J.: Computational intelligence in earth sciences and environmental applications: issues and challenges, Neural Networks, 19, 113–121, 2006.

Cherkassky, V. S. and Mulier, F.: Learning from Data: Concepts, Theory, and Methods, 2nd edn., John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.
Çimen, M.: Estimation of daily suspended sediments using support vector machines, Hydrol. Sci. J., 53(3), 656–666, 2008.
Dibike, Y. B., Velickov, S., Solomatine, D. P., and Abbott, M. B. Model induction with support vector machines: introduction and applications, ASCE J. Comput. Civil Eng., 15(3), 208–216, 2001.
Dibike, Y. B. and Solomatine, D. P.: River flow forecasting using artificial neural networks, J. Phys. Chem. Earth B: Hydrol. Oceans Atmos., 26(1), 1–8, 2001.
Doglioni, A., Giustolisi, O., Savic, D. A., and Webb, B. W.: An evolutionary approach to stream temperature analysis, Hydrol. Process. J., 22(3), 315–326, 2007.
Drexler, J. Z., Snyder, R. L., Spano, D., and Paw, K. T.: A review of models and micrometeorological methods used to estimate wetland evapotranspiration, Hydrol. Process., 18, 2071–2101, 2004.
Elshorbagy, A. and El-Baroudy, I.: Investigating the capabilities of evolutionary data-driven techniques using the challenging estimation of soil moisture content, J. Hydroinfo., 11(3–4), 237–251, 2009.
Elshorbagy, A. and Parasuraman, K.: Toward bridging the gap between data-driven and mechanistic models: cluster-based neural networks for hydrologic processes. in: Practical Hydroinformatics. Computational Intelligence and Technological Developments in Water Applications, edited by: Abrahart, R., See, L., and Solomatine, D., Springer-Verlag, Berlin, Heidelberg, 389–403, 2008.
Elshorbagy, A., Jutla, A., and Kells, J.: Simulation of the hydrological processes on reconstructed watersheds using system dynamics, Hydrol. Sci. J., 52, 538–562, 2007.
Giustolisi, O., Doglioni, A., Savic, D. A., and Webb, B. W.: A multi-model approach to analysis of environmental phenomena. Environ. Modell. Softw., 22(5), 674–682, 2007.
Evans, D. and Jones, A. J.: A proof of the gamma test, Proc. Roy. Soc. A, 458, 2759–2799, 2002.
Giustolisi, O. and Savic, D. A.: A symbolic data-driven technique based on evolutionary polynomial regression, J. Hydroinf., 8(3), 207–222, doi:10.2166/hydro.2006.020, 2006.
Haigh, M. J.: The aims of land reclamation, Land Reconstruction and Management, A. A. Balkema Publishers, Rotterdam, The Netherlands, 1, 1–20, 2000.
Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn, MacMillan, New York,

1999.
Jayawardena, A. W., Muttil, N., and Lee: J. H. W.: Comparative analysis of data-driven and GIS-based conceptual rainfall-runoff model, J. Hydrolog. Eng., 11(1), 1–11, 2006.
Jayawardena, A. W., Muttil, N., and Fernando, T. M. K. G.: Rainfall-runoff modelling using genetic programming, MODSIM 2005 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand, edited by: Zerger, A. and Argent, R. M., December 2005, 1841–1847. ISBN: 0-9758400-2-9, 2005.
Jones, A. J., Margetts, S., and Durrant, P.: The winGamma^{TM} User Guide. University of Wales, Cardiff, 2001.
Khan, M. S. and Coulibaly, P.: Application of support vector machine in lake water level prediction, J. Hydrol. Eng., 11(3), 199–205, 2006.
Koza, J. R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, The MIT Press, Cambridge, 1992.
Laucelli, D., Berardi, L., and Doglioni, A.: Evolutionary polynomial regression toolbox: version 1.SA, Department of Civil and Environmental Engineering, Technical University of Bari, Bari, Italy. Available from: http://www.hydroinformatics.it/prod02.htm, 2005.
Maier, H. and Dandy, G.: Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications, Environ. Modell. Softw., 15(1), 101–124, 2000.
Makkeasorn, A., Chang, N. B., and Zhou, X.: Short-term streamflow forecasting with global climate change implications – A comparative study between genetic programming and neural network models, J. Hydrol., 352, 336–354, 2008.
Mattera, D. and Haykin, S.: Support vector machines for dynamic reconstruction of a chaotic system, in: Advances in Kernel Methods – Support Vector Learning, edited by: Schölkopf, B., Burges, C. J. C., and Smola, A. J., 211–242, MIT Press, Cambridge, 1999.
Minns, A. W. and Hall, M. J.: Artificial neural networks as rainfall-runoff models, Hydrol. Sci. J., 41, 399–417, 1996.
Müller, K. R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., and Vapnik, V.: Predicting time series with support vector machines, in: Artificial Neural Networks – ICANN'97, edited by: Gerstner, W., Germond, A., Hasler, M., and Nicoud, J. D., 999–1004, Springer Lecture Notes in Computer Science, Vol. 1327, Berlin, 1997.
Karlsson, M. and Yakowitz, S.: Nearest neighbour methods for nonparametric rainfall–runoff forecasting, Water Resour. Res., 23(7), 1300–1308, 1987.

Parasuraman, K. and Elshorbagy, A.: Cluster-based hydrologic prediction using genetic algorithm-trained neural networks, J. Hydrol. Eng., ASCE, 12(1), 52–62, 2007.
Parasuraman, K., Elshorbagy, A., and Carey, S. K.: Modelling dynamics of the evapotranspiration process using genetic programming, Hydrol. Sci. J., 53(3), 563–578, 2007a.
Parasuraman, K., Elshorbagy, A., and Si, B. C.: Estimating saturated hydraulic conductivity using genetic programming, Soil Sci. Soc. Am. J., 71, 1676–1684, 2007b.
Parasuraman, K. and Elshorbagy, A.: Model structure uncertainty and its quantification using ensemble-based genetic programming framework, Water Resour. Res., 44, W12406, doi:10.1029/2007WR006451, 2008.
Rabuñal, J. R., Puertas, J., Suárez, J., and Rivero, D.: Determination of the unit hydrograph of a typical urban basin using genetic programming and artificial neural networks, Hydrol. Process., 21, 476–485, 2007.
Savic, D. A., Giustolisi, O., Berardi, L., Shepherd, W., Djordjevic, S., and Saul, A.: Sewers failure analysis using evolutionary computing, Water Manage. J., 159(2), 111–118, doi:10.1680/wama.2006.159.2.111, 2006.
Silva, S.: GPLAB – a genetic programming toolbox for MATLAB, http://gplab.sourceforge.net, 2005.
Sivapragasam, C., Vincent, P., and Vasudevan, G.: Genetic programming model for forecast of short and noisy data, Hydrol. Process., 21, 266–272, 2007.
Solomatine, D. P. and Dulal, K. N.: Model trees as an alternative to neural networks in rainfall-runoff modelling, Hydrol. Sci. J., 48(3), 399–411, 2003.
Solomatine, D. P., Maskey, M., and Shrestha, D. L.: Instance-based learning compared to other data-driven methods in hydrological forecasting, Hydrol. Process., 22, 275–287, 2008.
Solomatine, D. P. and Siek, M. B.: Modular learning models in forecasting natural phenomena, Neural Networks, 19, 225–235, 2006.
Solomatine, D. P. and Xue, Y.: M5 model trees and neural networks: application to flood forecasting in the upper reach of the Huai River in China, J. Hydrol. Eng., 9(6), 491–501, 2004.
Solomatine, D. P. and Ostfeld, A.: Data-driven modelling: some past experiences and new approaches, J. Hydroinf., 10(1), 3–22, 2008.
Stefánsson, A, Konèar, N., and Jones, A. J.: A note on the gamma test, Neural Comput. Appl., 5, 131–133, 1997.
Stravs, L. and Brilly, M.: Development of a low-flow forecasting model using the M5 machine learning method, Hydrol. Sci. J., 52(3), 466–477, 2007.

Vapnik, V.: The Nature of Statistical Learning Theory, Springer, New York, 1995.
Wit, de M. J. M.: Effect of Climate Change on the Hydrology of the River Meuse. RIVM, National Institute op Public Health and the Environment, 2001.
Witten, I. H. and Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn., Morgan Kaufmann, San Francisco, 2005.
Wu, C. L., Chau, K. W., and Li, Y. S.: River stage prediction based on a distributed support vector regression, J. Hydrol., 358, 96–111, 2008.
Wu, W., Wang, X., Xie, D., and Liu, H.: Soil water content forecasting by support vector machine in Purple Hilly Region, Comput. Comput. Technol. Agr., 1, 223–230, 2008.
Zhang, B. and Govindaraju, S.: Prediction of watershed runoff using Bayesian concepts and modular neural networks, Water Resour. Res., 36(3), 753–762, 2000.

Table 1. Descriptive statistics of the AET dataset.

	NR W/m^2	AT °C	GT °C	RH	WS m/s	LE W/m^2
			Training dataset			
Minimum	–189.6	–3.4	4.1	0.14	0.4	–80.2
Maximum	875.4	33.9	27.2	0.96	10.2	503.8
Mean	229.7	18.7	16.7	0.5	2.8	144.9
SD	189.4	5.5	3.8	0.2	1.7	90.0
CV	0.82	0.29	0.23	0.34	0.62	0.62
			Cross validation dataset			
Minimum	–119.8	–3.2	3.7	0.16	0.4	–42.2
Maximum	729.5	33.7	26.4	0.95	11	405.6
Mean	224.1	18.7	16.9	0.5	2.8	145.9
SD	181.9	5.6	3.8	0.2	1.7	88.7
CV	0.81	0.30	0.23	0.33	0.60	0.61
			Testing dataset			
Minimum	–414.6	–4.3	3.3	0.15	0.4	–56.3
Maximum	801.6	33.8	27.2	0.96	12.3	425.8
Mean	226.9	18.5	16.6	0.5	2.9	143.8
SD	188.9	5.5	3.7	0.2	1.8	89.9
CV	0.83	0.30	0.22	0.34	0.63	0.63

HESSD
6, 7055–7093, 2009

Table 2. Descriptive statistics of the daily peat and till moisture datasets.

	P mm	AT °C	NR W/m^2	STP °C	STT °C	SMP	SMT
			Training dataset				
Minimum	0.00	–6.30	–10.40	0.50	–0.50	0.304	0.240
Maximum	43.70	25.20	204.40	18.20	16.30	0.539	0.316
Mean	1.54	13.63	90.64	11.71	10.48	0.442	0.288
SD	4.20	6.10	50.22	3.79	3.49	0.055	0.018
CV	2.72	0.45	0.55	0.32	0.33	0.124	0.062
			Cross validation dataset				
Minimum	0.00	–3.90	0.00	0.50	–0.70	0.305	0.241
Maximum	27.18	22.90	226.10	18.20	16.10	0.542	0.316
Mean	1.68	13.80	92.96	11.75	10.32	0.440	0.289
SD	3.99	4.96	49.98	4.03	4.17	0.055	0.018
CV	2.38	0.36	0.54	0.34	0.40	0.125	0.062
			Testing dataset				
Minimum	0.00	–6.80	0.00	–0.10	–0.60	0.306	0.241
Maximum	23.60	25.80	223.60	18.20	16.10	0.543	0.316
Mean	1.48	14.07	96.94	11.88	10.45	0.440	0.288
SD	3.32	5.96	50.91	3.77	3.56	0.054	0.018
CV	2.25	0.42	0.53	0.32	0.34	0.123	0.061

HESSD
6, 7055–7093, 2009

Table 3. Descriptive statistics of the output variables of all datasets.

	Evapotranspiraion	Peat moisture	Till moisture	Runoff
Count	5307	972	972	4008
Minimum	–80.20	0.30	0.24	1.07
Median	133.09	0.45	0.29	11.39
Average	144.52	0.44	0.29	21.91
Maximum	503.77	0.54	0.32	370.63
St. deviation	89.79	0.05	0.02	29.93
CV	0.62	0.12	0.06	1.37
Skew	0.51	–0.72	–1.33	4.06

HESSD
6, 7055–7093, 2009

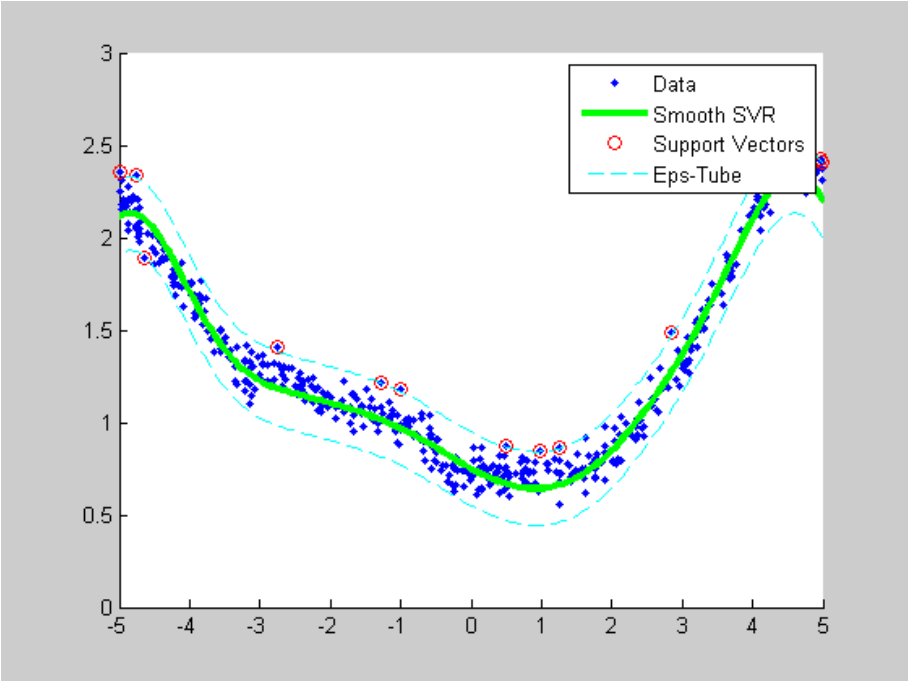

Fig. 1. Illustration of SV regression. Model errors inside the ε-tube are ignored.

HESSD

6, 7055–7093, 2009

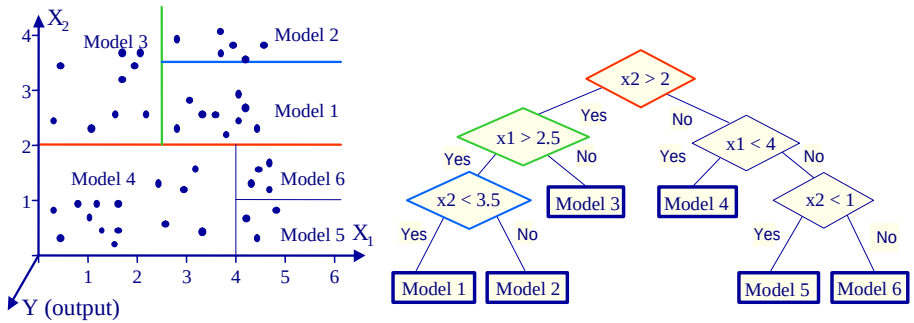

Fig. 2. Illustration of splitting in a model tree.

HESSD
6, 7055–7093, 2009

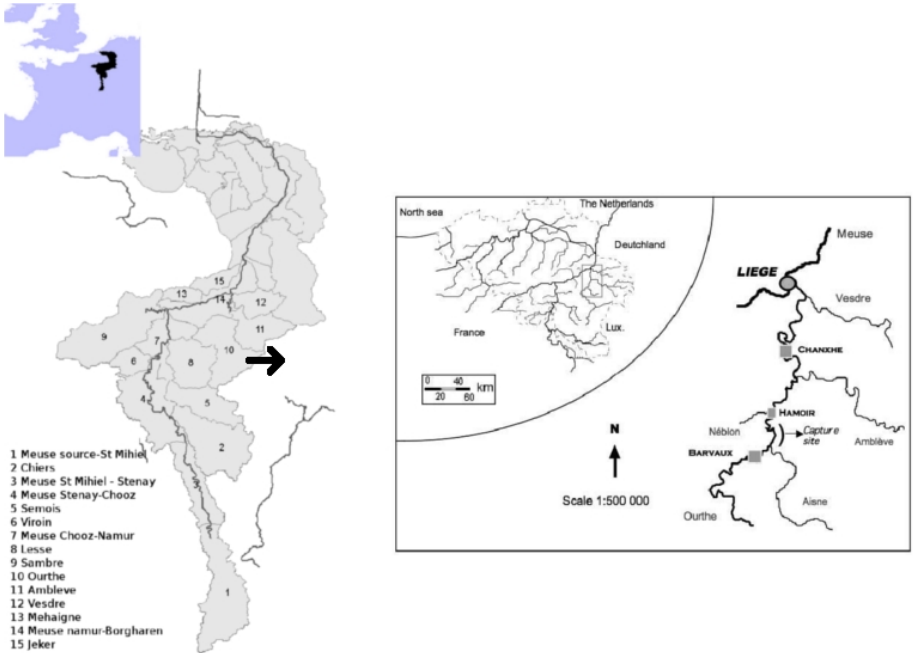

Fig. 3. The Meuse river basin and the sub-basins upstream of Borgharen. Sub-basin 10 (Ourthe) is used in the case study.

HESSD
6, 7055–7093, 2009