Model trees as an alternative to neural networks in rainfall–runoff modelling

DIMITRI P. SOLOMATINE
International Institute for Infrastructural, Hydraulic and Environmental Engineering (IHE), PO Box 3015, NL-2601 DA Delft, The Netherlands
sol@ihe.nl

KHADA N. DULAL
Department of Hydrology and Meteorology, PO Box 406, Kathmandu, Nepal

Abstract This paper investigates the comparative performance of two data-driven modelling techniques, namely, artificial neural networks (ANNs) and model trees (MTs), in rainfall–runoff transformation. The applicability of these techniques is studied by predicting runoff one, three and six hours ahead for a European catchment. The result shows that both ANNs and MTs produce excellent results for 1-h ahead prediction, acceptable results for 3-h ahead prediction and conditionally acceptable result for 6-h ahead prediction. Both techniques have almost similar performance for 1-h ahead prediction of runoff, but the result of the ANN is slightly better than the MT for higher lead times. However, the advantage of the MT is that the result is more understandable and allows one to build a family of models of varying complexity and accuracy.

Key words rainfall; runoff; artificial neural networks; M5 model tree; prediction; committee machine

Arbres de modèles comme alternative aux réseaux de neurones en modélisation pluie–débit

Résumé Cet article étudie de manière comparative les performances de deux techniques de modélisation pluie débit contraintes par les données, en l'occurrence des réseaux de neurones artificiels (RNA) et des arbres de modèles (AM). L'applicabilité de ces techniques est étudiée pour la prévision des débits d'un bassin versant européen pour des anticipations de une, trois et six heures. Les résultats montrent que les RNA et les AM produisent des résultats excellents pour la prévision à une heure, acceptables pour la prévision à trois heures et acceptables sous condition pour la prévision à six heures. Les deux techniques ont des performances presque similaires pour la prévision des débits à une heure, mais les résultats du RNA sont légèrement meilleurs que ceux de l'AM pour les délais plus longs. Néanmoins l'AM présente les avantages de fournir des résultats plus compréhensibles et de permettre la construction d'une famille de modèles de complexité et de précision variables.

Mots clefs pluie; écoulement; réseaux de neurones artificiels; arbre de modèles M5; prévision; fusion de classifications

INTRODUCTION

The prediction of variables such as precipitation, runoff, river stages etc. has always been a major problem in hydrology. Hydrological phenomena are extremely complex, highly nonlinear and exhibit a high degree of spatial and temporal variability. So, hydrological modelling becomes one of the important tasks for planning, operation and control of any water resource project.

In the area of rainfall–runoff modelling, numerous runoff forecasting techniques have been suggested and used in the past. There are basically two approaches for

hydrological modelling: the theory-driven (conceptual and physically-based) approach, and the data-driven (empirical and black-box) approach often associated by practitioners with statistical modelling. Conceptual models represent the general internal sub-processes and physical mechanisms of the hydrological cycle, without looking at the spatial variability and stochastic properties of the rainfall–runoff process. The parameters are generally assumed as lumped representation of the basin characteristics. Physically-based models are based on the understanding of the underlying physical behaviour of the system (hydrological cycle). Typically, they involve the solution of a system of partial differential equations that represent our best understanding of the flow processes in the watershed. Data-driven (black-box) models treat the hydrological system (such as a watershed) as a black box and try to find a relationship between historical inputs (e.g. rainfall, temperature, etc.) and outputs (e.g. runoff). Traditionally, the data-driven models borrow the techniques developed in such (overlapping) areas as statistics, soft computing, computational intelligence, machine learning and data mining.

Among many data-driven techniques, the artificial neural network (ANN) is the most widely used. The concept of ANNs is inspired by the biological neural networks of the human brain. Mathematically, an ANN is a complex nonlinear function with many parameters that are adjusted in such a way that the ANN output becomes similar to the measured output on a known data set. The discovery of the back-propagation algorithm for training an ANN has caused a tremendous growth of interest in this field. Within the last decade, one could see a huge advancement due to the development of more sophisticated algorithms and the emergence of powerful computational tools. The applications of ANNs in hydrology can be found in many papers, such as Minns & Hall (1996), Shamseldin (1997), Abrahart & Kneale (1997), Dawson & Wilby (1998), Dibike & Solomatine (1999), Sajikumar & Thandaveswara (1999), Zealand *et al*. (1999), Abrahart & See (2000), Coulibaly *et al.* (2000), Imrie *et al*. (2000), Govindaraju & Rao (2000), Lekkas *et al*. (2001), Persson & Berndtsson (2001), Rajurkar *et al.* (2002), Shamseldin & O'Connor (2001), and Tayfur (2002). One of the disadvantages of ANNs is that for a decision maker it is very difficult to analyse the structure of the resulting ANN and to relate it to the outputs.

However, there are approaches to numerical prediction that often reach accuracy comparable to that of ANNs. They use piece-wise linear approximations that are much easier to interpret. One method of Friedman (1991) is used in his MARS (multiple adaptive regression splines) algorithm implemented as MARS software. Another, used in this paper, is the machine learning method M5 model tree (Quinlan, 1992) implemented in the Cubist and Weka software packages (Witten & Frank, 2000). An earlier method of Breiman *et al*. (1984) of regression trees (implemented in the CART software) should also be mentioned, but it generates zero-order models (constant output values for subsets of input data) rather than first-order (linear) models.

However, in hydrology, M5 model trees (MTs) are practically unknown; only one related paper in Slovene language (Kompare *et al*., 1997) was found by the authors. Solomatine (2002) demonstrated the use of MTs in hydrological and other problems, along with other data-driven models.

The purpose of this paper is to report the application and to investigate the performance of M5 model trees for rainfall–runoff modelling, and to compare this with the performance of ANNs.

M5 MODEL TREE

This machine-learning technique uses the following idea: split the parameter space into areas (subspaces) and build in each of them a linear regression model. In fact the resulting model can be seen as a modular model, or a committee machine, with the linear models being specialized on the particular subsets of the input space. This idea is not new. Combination of specialized models ("local" models) is used in modelling quite often. One can find a clear analogy between MTs and combination of linear models used in dynamic hydrology already in the 1970s—a notable paper on multilinear models is by Becker & Kundzewicz (1987). However, the M5 model tree approach, based on the principle of information theory, makes it possible to split the multi-dimensional parameter space and to generate the models automatically according to the overall quality criterion; it also allows for varying the number of models. The idea of combining several models with the help of computational intelligence techniques, possibly combining theory- and data-driven, is finding more and more supporters in hydrology (see, for example Xiong *et al*., 2001, where the outputs of hydrological models are combined in a fuzzy system).

The splitting in MT follows the idea of a decision tree, but instead of the class labels it has linear regression functions at the leaves, which can predict continuous numerical attributes. Model trees generalize the concepts of regression trees, which have constant values at their leaves (Witten & Frank, 2000). So, they are analogous to piece-wise linear functions (and hence nonlinear). Computational requirements for model trees grow rapidly with dimensionality. Model trees learn efficiently and can tackle tasks with very high dimensionality—up to hundreds of attributes. The major advantage of model trees over regression trees is that model trees are much smaller than regression trees, the decision strength is clear, and regression functions do not normally involve many variables.

The algorithm known as the M5 algorithm is used for inducing a model tree (Quinlan, 1992), which works as follows (Fig. 1). Suppose that a collection T of training examples is available. Each example is characterized by the values of a fixed set of (input) attributes and has an associated target (output) value. The aim is to construct a model that relates a target value of the training cases to the values of their input attributes. The quality of the model will generally be measured by the accuracy with which it predicts the target values of the unseen cases.

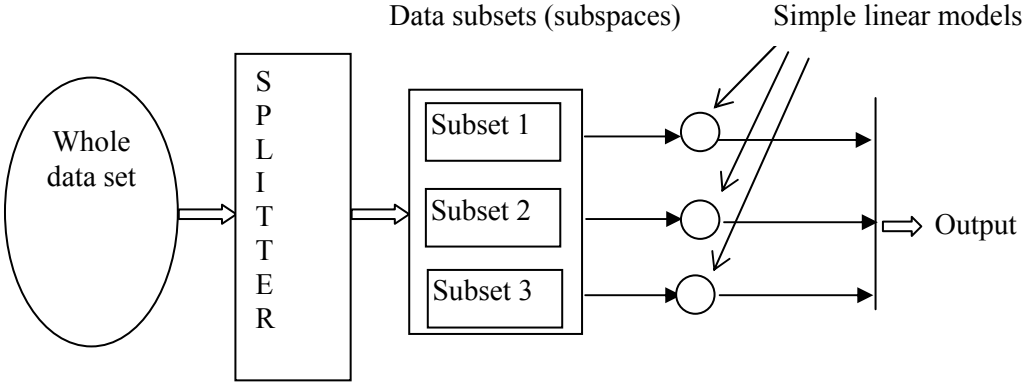

Fig. 1 The induction of a model tree as a modular model (committee machine).

Tree-based models are constructed by a divide-and-conquer method. The set T is either associated with a leaf, or some test is chosen that splits T into subsets corresponding to the test outcomes and the same process is applied recursively to the subsets. The splitting criterion for the M5 model tree algorithm is based on treating the standard deviation of the class values that reach a node as a measure of the error at that node, and calculating the expected reduction in this error as a result of testing each attribute at that node. The formula to compute the standard deviation reduction (*SDR*) is:

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

where T represents a set of examples that reaches the node; T_i represents the subset of examples that have the ith outcome of the potential set; and *sd* represents the standard deviation.

After examining all possible splits, M5 chooses the one that maximizes the expected error reduction. Splitting in M5 ceases when the class values of all the instances that reach a node vary just slightly, or only a few instances remain. The relentless division often produces over-elaborate structures that must be pruned back, for instance by replacing a subtree with a leaf. In the final stage, a smoothing process is performed to compensate for the sharp discontinuities that will inevitably occur between adjacent linear models at the leaves of the pruned tree, particularly for some models constructed from a smaller number of training examples. In smoothing, the adjacent linear equations are updated in such a way that the predicted outputs for the neighbouring input vectors corresponding to the different equations are becoming close in value. Details of this process can be found in Quinlan (1992) and are given by Witten & Frank (2000).

CASE STUDY

Many experiments have been performed by the authors with ANNs, MTs, and other machine learning techniques for numerical prediction on various data sets. The choice of a case study in hydrology was a bit of a problem since the real-life data often suffer from gaps and inaccuracies and could be inappropriate for testing a data-sensitive method in different regimes. That is why a three-month data set was selected. The set is quite old, but of high quality, high frequency, and it was used previously to compare hydrological models (Franchini & Pacciani, 1991). Since the objective was to build a predictive model, the time resolution of the data set was important: it was found that the reaction time of the catchment was 6 h, that is the much larger available daily data sets could not be used.

The area for this study is the Sieve River basin, located in the Tuscany region of Italy, which has a drainage area of 822 km^2. The Sieve is a tributary of the Arno River, having a length of 56 km. The basin covers mostly hills and mountainous areas. The climate of the basin is temperate and humid.

For this basin, three months of hourly discharge (Q), precipitation (R) and evapotranspiration (E) data were available (December 1959–February 1960, 2160 data points). The data represent various types of hydrological conditions, and flows range

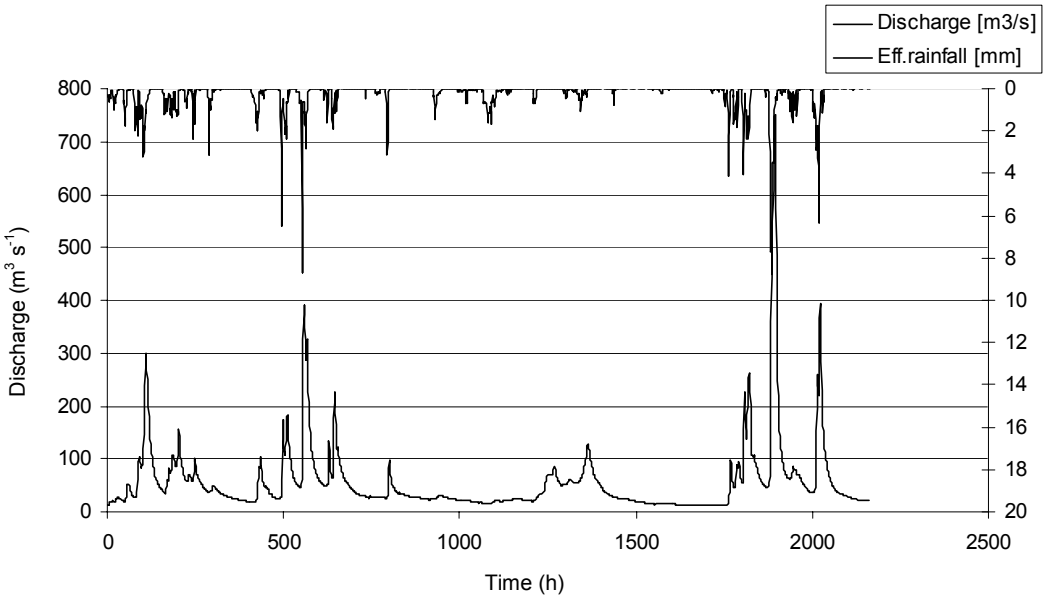

Fig. 2 Sieve catchment: rainfall and discharge data.

from low to very high (Fig. 2). The supplied data were computed as follows: the discharge data were calculated using rating curves and hourly water levels; mean areal precipitation was calculated by Thiessen polygon method using 11 rainfall stations; the hourly areal evapotranspiration data were calculated using the radiation method.

EXPERIMENTS

Input set preparation

In general, the goal of a data-driven model (DDM), for example of a regression model or ANN, is to generalize a relationship of the form:

$$Y^{(m)} = f(X^{(n)}) \tag{2}$$

where $X^{(n)}$ is an n-dimensional input vector consisting of variables x_1, ... x_n, and $Y^{(m)}$ is an m-dimensional output vector consisting of resulting variables of interest y_1, ... y_n. In hydrology, the values of x_i can be causal variables such as rainfall, temperature, previous flows, evaporation etc. and the values of y_i can be hydrological responses such as runoff.

The selection of appropriate model inputs is extremely important in any prediction/forecasting model. However, in many DDM applications not enough attention is given to this task. The philosophy that is generally adopted is to include all input variables that might possibly have an influence on the model outputs and to let the DDM determine which inputs are significant. However, presenting a large number of inputs and relying on the DDM to determine the critical model inputs, usually results in the inclusion of insignificant model inputs.

The choice of input variables is generally based on prior knowledge of causal variables in conjunction with inspections of time series plots of potential inputs and outputs. On top of this, firm understanding of the hydrological system under considera-

tion is an important prerequisite for the successful application of a DDM. After choosing appropriate input variables, the next step is the determination of appropriate lags. Since the flow at any moment is effectively composed of contributions from different sub-areas whose time of travel covers a range of values, the dynamics of the system can be approximated by a set of variables that are properly discretized and lagged with respect to the output to be predicted.

Data analysis of the Sieve catchment for preparation of inputs

For data-driven modelling of the Sieve catchment, the causal variable is the rainfall, while the hydrological response is the discharge at the outlet. Visualization of the input and output shows that the maximum value of peak-to-peak time lags of rainfall and runoff is close to 7 h. Additional analysis of lags was performed using the average mutual information and cross-correlation analysis of rainfall and runoff. Figure 3 presents the cross-correlation between the rainfall and runoff and it can be seen that it is increasing with the lag, becomes maximum (0.75) at a lag of 6 h and then starts decreasing. So, average lag time of rainfall for this catchment is considered to be 6 h.

Besides rainfall, the previous flows can also be used as input variables in the DDM, since they also have a high correlation with the future flows (see Fig. 3). An additional reason to consider the past flow is this: as the rainfall data contain a number of zero values, the condition of rainfall and no-rainfall is difficult to identify by DDM with only rainfall time series as inputs. In such a case, the previous river flows provide an indication as to whether rain has occurred or not. Also, such flows add further information in that the longer the interval of zero input, the more the output decreases.

From the cross-correlation analysis between the evaporation and discharge it results that correlation is very low; this indicates that evaporation has little effect on the runoff of the catchment. Therefore, instead of using evaporation and rainfall as

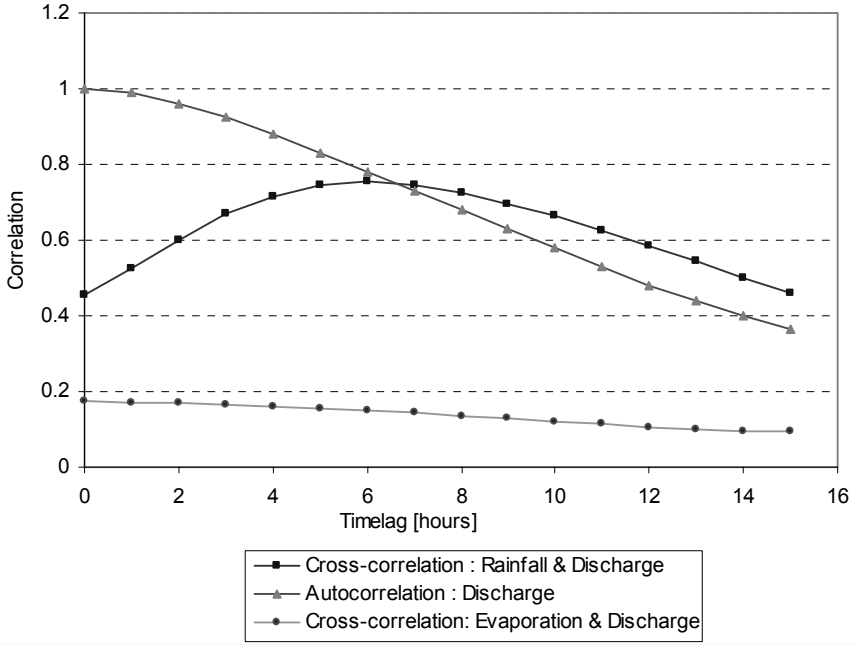

Fig. 3 Correlation analysis.

separate inputs, effective rainfall (rainfall minus evaporation for rainfall greater than evaporation and zero otherwise) will be used as the input in this study.

ANN model set up

The meaning of the symbols used is as follows:

$RE_{t-\tau}$ Effective rainfall, τ = lag time (0, 1, 2, … h)
Q_t Discharge at time t
Q_{t-k} Discharge at time $t-k$, k = 1, ..., 6

Software packages NeuralMachine (2002) and NeuroSolutions (2002) were used for the ANN modelling. A multi-layered perceptron (MLP) network trained with the back-propagation algorithm was used because of its simplicity and capability to learn the rainfall–runoff relationship. A nonlinear activation function, the hyperbolic tangent function (bounded between –1 and +1) was used in the hidden layer. The linear activation function was used in the output layer because it is unbounded and is able, to a certain extent, to extrapolate beyond the range of the training data. The number of hidden nodes was determined by trial-and-error optimization (performed automatically by software). For training, the back-propagation algorithm was used with momentum rule with the stopping after 5000 epochs or when the mean squared error (*MSE*) reached 0.0001. The first 300 points were used for verification (from 1 December 1959, 07:00 to 13 December 1959, 18:00) and the remaining points for training (from 13 December 1959, 19:00 to 28 February 1960, 00:00). It can be seen that the verification data set includes highly variable data with both low and high flows, which is important for the proper model testing.

The summary of the trained ANNs is presented in Table 1.

Table 1 ANN experiment summary.

Input variables	Output variable	Hidden nodes
RE_t, RE_{t-1}, RE_{t-2}, RE_{t-3}, RE_{t-4}, RE_{t-5}, Q_t, Q_{t-1}, Q_{t-2}	Q_{t+1}	6
RE_t, RE_{t-1}, RE_{t-2}, RE_{t-3}, Q_t , Q_{t-1}	Q_{t+3}	5
RE_t, Q_t	Q_{t+6}	3

MT model set up

For the model tree (MT), the same sets of input–output, and training and verification data were considered as for the ANN. For MT learning, the Weka software (Witten & Frank, 2000) was used. A summary of the generated MTs is presented in Table 2.

Table 2 MT experiment summary.

Input variables	Output variable	Linear models
RE_t, RE_{t-1}, RE_{t-2}, RE_{t-3}, RE_{t-4}, RE_{t-5}, Q_t, Q_{t-1}, Q_{t-2}	Q_{t+1}	3
RE_t, RE_{t-1}, RE_{t-2}, RE_{t-3}, Q_t, Q_{t-1}	Q_{t+3}	3
RE_t, Q_t	Q_{t+6}	9

RESULTS AND DISCUSSIONS

Table 3 presents the performance statistics, and Fig. 4(a), (b) and (c) shows the predicted and observed hydrographs, for both methods from the verification data set.

Table 3 Comparison of errors for the verification data set.

Prediction	ANN:			MT:		
	RMSE ($m^3 s^{-1}$)	*NRMSE* ($m^3 s^{-1}$)	*COE*	*RMSE* ($m^3 s^{-1}$)	*NRMSE* ($m^3 s^{-1}$)	*COE*
Q_{t+1}	5.175	0.106	0.9886	3.612	0.074	0.9944
Q_{t+3}	11.353	0.234	0.9452	12.548	0.258	0.9331
Q_{t+6}	19.402	0.399	0.8401	21.547	0.443	0.8028

Prediction of runoff one hour ahead

The result of the experiment confirms the fact that the ANN is capable of identifying the rainfall–runoff relationships for a catchment quite well. The error statistics support this statement with high values of the coefficient of efficiency (*COE*), and low values of root mean squared error (*RMSE*) and normalized root mean squared error (*NRMSE*) for verification data, which implies a good model fit. The visualization (Fig. 4(a)) shows that both the low flows and the high flows are reproduced well by the model.

The result of MT modelling for the same set of inputs indicates that the prediction of Q_{t+1} using the MT is performed quite well for the high flows as well as the low flows. The modelling error for verification shows low *RMSE* and *NRMSE* and high values of *COE*, indicating a good model performance.

There were several models built, but even the simplest model with only three equations appeared to be very accurate:

$Q_t \leq 59.4$:
| $Q_t \leq 32.5$: LM1 (1011/1.64%)
| $Q_t > 32.5$: LM2 (396/6.15%)
$Q_t > 59.4$: LM3 (447/23.5%)

The following are the generated linear models:

LM1: $Q_{t+1} = 0.0374 + 0.0181RE_t + 0.0535RE_{t-1} + 0.00873RE_{t-2} + 0.0384RE_{t-3} + 1.01Q_t - 0.0127Q_{t-1} + 0.00311Q_{t-2}$

LM2: $Q_{t+1} = -0.456 + 0.0287RE_t + 1.73RE_{t-1} + 0.0407RE_{t-2} + 7.38RE_{t-3} + 1Q_t - 0.0127Q_{t-1} + 0.00311Q_{t-2}$

LM3: $Q_{t+1} = 2.97 + 2.47RE_t + 4.98RE_{t-1} - 0.0389RE_{t-2} + 1.75Q_t - 1.08Q_{t-1} + 0.265Q_{t-2}$

The performance of both of the techniques for 1-h ahead prediction of runoff is excellent, and comparatively the MT result is slightly better in terms of goodness of fit. It may be noted that the MT model is transparent, i.e. any resulting runoff value can be easily checked. The MT model automatically finds the regimes in the system, which in this case can be interpreted as low (<32.5 $m^3 s^{-1}$), medium (<59.4 $m^3 s^{-1}$) and high flows. Each of the regimes is described by a separate linear model.

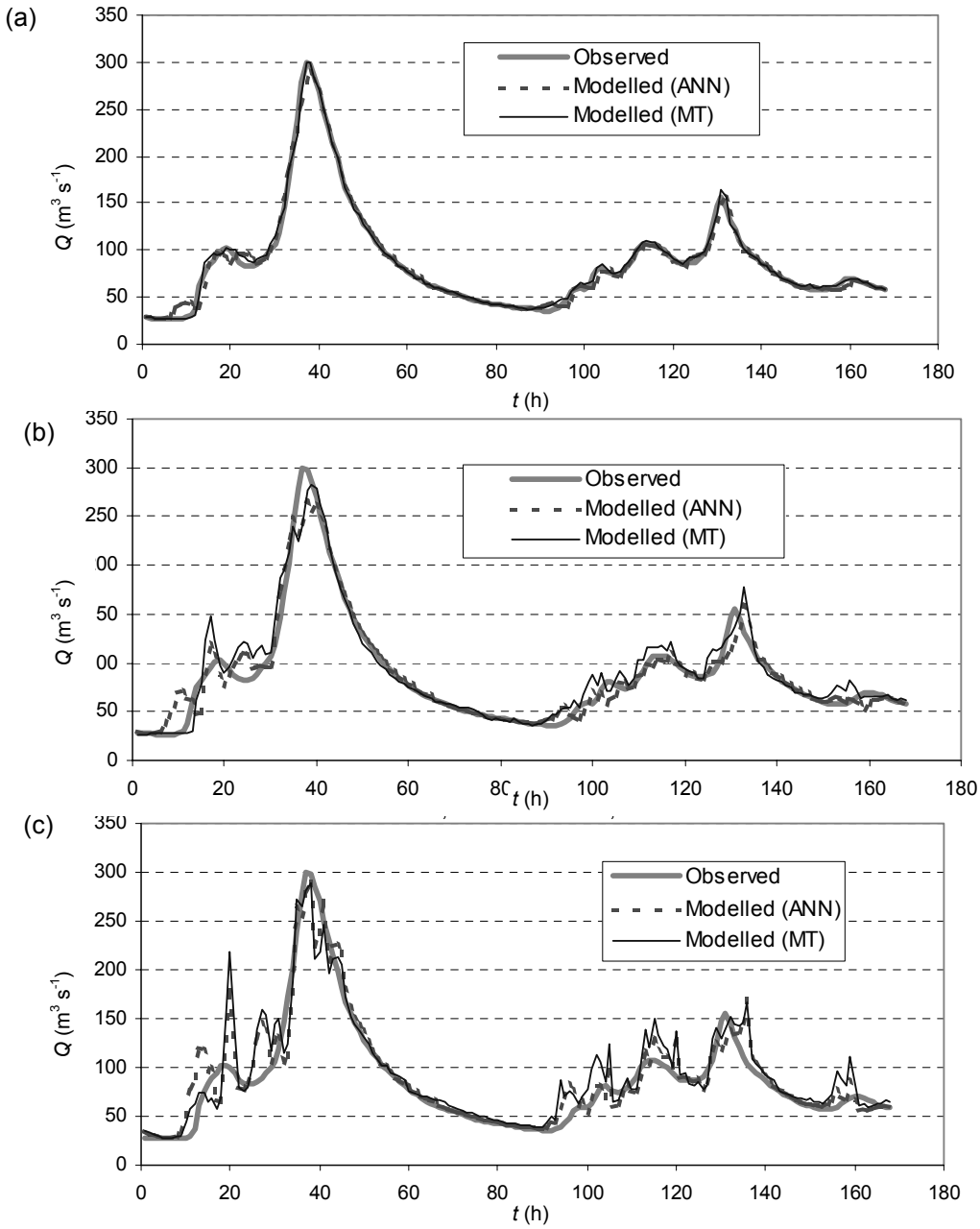

Fig. 4 Prediction of (a) Q_{t+1}; (b) Q_{t+6}; and (c) Q_{t+3} using ANNs and MTs.

Prediction of runoff three hours ahead

In the second experiment of runoff modelling, an ANN was applied to the prediction of the runoff of the Sieve River 3 h ahead (Q_{t+3}). The visualization (Fig. 4(b)) shows that the low-flow values are estimated well, while the ANN model encounters some difficulties in predicting peak flows. It seems that the runoff values computed by the ANN for medium peaks are more affected by the value of RE_t. As RE_t increases, Q_{t+3} also increases and *vice versa*. That may be the reason for the oscillations in the medium peaks. For the highest peak, both rising and falling limbs of the hydrograph are captured well, but the peak magnitude is slightly underestimated.

The MT result (Fig. 4(b)) makes it clear that the result is good for low flows, but the peak values are not predicted well. As in the case of the ANN, there are oscillations in the prediction of medium peaks and the highest peak is underestimated.

The following is the generated model tree with the pruning factor 4:

$Q_t \leq 51.2$:
| $Q_t \leq 28.7$: LM1 (903/5.66%)
| $Q_t > 28.7$: LM2 (379/13.1%)
$Q_t > 51.2$: LM3 (572/66.7%)

The following are the generated linear models:

LM1: $Q_{t+3} = -0.0118 + 0.317RE_t + 0.124RE_{t-1} + 0.0844RE_{t-2} - 0.109RE_{t-3} + 1.09Q_t - 0.0826Q_{t-1}$

LM2: $Q_{t+3} = -0.262 + 11.9RE_t + 0.182RE_{t-1} + 8.9RE_{t-2} - 0.198RE_{t-3} + 3.66Q_t - 2.67Q_{t-1}$

LM3: $Q_{t+3} = 15.5 + 25.7RE_t + 7.59RE_{t-1} - 0.0923RE_{t-3} + 1.44Q_t - 0.732Q_{t-1}$

Comparatively, the result of the ANN is slightly better than the MT (*RMSE* is 9.5% lower) with less oscillations in the medium peaks. However, the highest peak is underestimated by both of the methods.

Prediction of runoff six hours ahead

As the average lag time of the catchment was found to be 6 h, in the last experiment the runoff was predicted with a lead time of maximum 6 h. Naturally, as the prediction horizon increases, the prediction becomes less accurate. This was also the case in this ANN experiment. The result is poor for high flows with much more noise in the peaks (Fig. 4(c)), the *RMSE* and *MAE* produced by the model for verification is higher with *COE* in the range of 0.8, and the low flow values are also overestimated. Therefore, the prediction of runoff 6 h ahead is not very good.

The prediction of Q_{t+6} of the MT gives similar result; it is seen from the plot that there are large errors in the high flows (Fig. 4(c)). The modelling error for verification is also high. So, the result for a 6-h ahead prediction of runoff can be considered as acceptable only conditionally.

The following is the generated model tree. It is larger than the previous ones because an attempt to prune it to a smaller size led to the deterioration of accuracy.

$Q_t \leq 37$:
| $RE_t \leq 0.0614$: LM1 (879/3.51%)
| $RE_t > 0.0614$:
| | $RE_t \leq 0.384$:
| | | $Q_t \leq 22.8$: LM2 (82/7.08%)
| | | $Q_t > 22.8$: LM3 (50/14.5%)
| | $RE_t > 0.384$: LM4 (89/60.5%)
$Q_t > 37$:

| $RE_t \leq 0.382$:
| | $Q_t \leq 70.2$: LM5 (356/24.4%)
| | $Q_t > 70.2$
| | | $Q_t \leq 103$: LM6 (145/33.9%)
| | | $Q_t > 103$: LM7 (80/29.1%)
| $RE_t > 0.382$:
| | $RE_t \leq 2.04$: LM8 (135/160%)
| | $RE_t > 2.04$: LM9 (38/362%)

The following are the generated linear models:

LM1: $Q_{t+6} = 0.622 + 1.02RE_t + 0.947Q_t$
LM2: $Q_{t+6} = 3 + 4.08RE_t + 0.842Q_t$
LM3: $Q_{t+6} = 17.8 + 4.08RE_t + 0.398Q_t$
LM4: $Q_{t+6} = -32.3 + 23.5RE_t + 2.23Q_t$
LM5: $Q_{t+6} = 5.15 + 43.6RE_t + 0.829Q_t$
LM6: $Q_{t+6} = 42.9 + 34.5RE_t + 0.314Q_t$
LM7: $Q_{t+6} = 45.7 + 34.2RE_t + 0.38Q_t$
LM8: $Q_{t+6} = 33.8 + 64.3RE_t + 0.353Q_t$
LM9: $Q_{t+6} = 91.8 + 39.5RE_t + 0.535Q_t$

Although the performance of both of the techniques for the prediction of Q_{t+6} is quite poor, the ANN has slightly lower error than the MT.

CONCLUSIONS

For the runoff modelling experiments of the Sieve catchment using ANNs and MTs, it was found that both techniques performed very well for runoff prediction with short lead times (1 and 3 h), while both failed to produce good results for runoff prediction with higher lead times (6 h). The prediction of runoff 1 h ahead is satisfactory because the input space contains the most recent information as well as appropriately lagged information. For instance, the lagged rainfall values provide the model with information about when the rain started, and the recent runoff values tell the model where the hydrograph starts rising, reaches the peak and starts falling. It is noteworthy for runoff prediction with a lead time of 6 h that peaks are predicted poorly with either overestimation or underestimation of the observed hydrographs.

In this study, the higher errors for the higher prediction lead times may be due to the following factors:

– Inadequacy in the information carrying capacity of the available data: the data set contains many values at the low flow part which are reproduced well by the model, whereas it contains less information for high flow parts, which may be insufficient to learn the flood hydrograph part.
– Inability of the model to identify saturation excess runoff and infiltration excess runoff: for example, in the prediction of Q_{t+6}, it was seen that the medium peaks are overestimated with noisy oscillations. This is due to the fact that the runoff

generated by the model is increased immediately when RE_t is increased and *vice versa*. In reality, it should be saturation excess runoff, and the increase in Q should start after 6 h as a lagged effect of rainfall.

– Lack of recent information in model structure: for the prediction of Q_{t+3} and Q_{t+6}, the latest information is the information at time t. Although there are lagged rainfall values up to 6 h, there are no recent values of runoff to provide the information about the proper rising and falling time of the hydrograph, as in the case of Q_{t+1} prediction. That is why the model cannot appropriately learn the relationship in high flow parts.

Looking at the relative performance of the two techniques (ANNs and MTs) for runoff modelling of the Sieve catchment, the following was found:

– For 1-h ahead prediction of runoff, the performance of both techniques was almost the same. In this particular experiment, the MT was slightly more accurate. It shows that MTs are also able to learn rainfall–runoff relationships like ANNs when the dataset contains all possible information.
– For a 3- and 6-h ahead prediction of runoff, the results of the ANN were slightly better than those of the MT in terms of both performance measures and fitness of predicted hydrograph. This discrepancy may be due to the splitting criteria of the input space to build linear models at the leaves. Model trees do not use all available attributes to make linear models at any leaf. Only those attributes which fulfil the conditions of certain criteria (standard deviation reduction in this case) go under one sub-tree, terminating to a leaf. It is likely that the influencing attribute in reality may not be there which may be the reason for poorer performance. On the other hand, learning in ANNs is different from MTs, i.e. the iterative learning process allows for ultimate model refining.

In conclusion, it is important to stress that the ANN is not the only data-driven model that can be used in hydrology or elsewhere. Attention to ANNs is without any doubt justifiable, but other models deserve attention as well. (The situation can be compared to a genetic algorithm that is widely and successfully used but is not the most efficient multi-extremum optimization technique.) One feature of ANNs that is often criticized by some hydrologists is that they do not reveal anything about the structure of the function that they represent. It is believed that the physics is locked up in the ANN model in the connection weights and threshold values, but these are not easily interpretable.

An MT approach can partly resolve this problem. The model setting is very easy, the training is very fast, and the generated result (simple linear equations) is understandable. The equations are not really physically interpretable but they allow for a quick check of the calculation of the predicted flow. The rules and equations can be easily implemented in a spreadsheet and they use the language of statistics that could be more appealing to some hydrologists than neural networks or soft computing. Another feature of MTs is that they generate "local" models that are, in principle, more accurate since they correspond to different flow regimes. An MT approach allows one to generate a family of models with different complexity and accuracy.

The next step, which is currently being undertaken, is to try to build a modular model (mixture of models, or committee machines) by using the M5 algorithm for splitting the space but using nonlinear local models, for example ANNs, in the leaves of a resulting model tree.

Acknowledgements Part of this work was performed in the framework of the project "Data mining, knowledge discovery and data-driven modelling" of the Delft Cluster research programme supported by the Dutch government. The authors are grateful to Prof. Marco Franchini for providing the data on the Sieve catchment.

REFERENCES

Abrahart, R. J. & Kneale, P. E. (1997) Exploring neural network rainfall–runoff modelling. In: Proc. Sixth National Hydrology Symposium (Manchester, UK, 15-18 September 1997), 9.35–9.44. University of Salford, UK.

Abrahart, R. J. & See, L. (2000) Comparing neural network and auto regressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrol. Processes* **14**, 2157–2172.

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

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

Coulibaly, P., Anctil, F., Rasmussen, P. & Bobée, B. (2000) A recurrent neural network approach using indices of low frequency climatic variability to forecast regional annual runoff. *Hydrol. Processes* **14**, 2555–2577.

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

Dibike, Y. B. & Solomatine, D. P. (1999) River flow forecasting using artificial neural network. In: *Geophysical Research Abstracts*, vol. 1(1). European Geophysical Society XXIV General Assembly, The Hague, The Netherlands. (Published also in extended form in *J. Phys. Chem. Earth*, Part B: *Hydrology, Oceans and Atmosphere*, 2001, **26**(1), 1–8.)

Franchini, M. & Pacciani, M. (1991) Comparative analysis of several conceptual rainfall–runoff models. *J. Hydrol.* **122**, 161–219.

Friedman, J. H. (1991) Multivariate adaptive regression splines. *Ann. Statist.* **19**, 1–141.

Govindaraju, R. S. & Rao, A. R. (eds) (2000) *Artificial Neural Network in Hydrology.* Kluwer, Dordrecht, The Netherlands.

Imrie, C. E., Durucan, S. & Korre, A. (2000) River flow prediction using artificial neural network generalisation beyond calibration range. *J. Hydrol.* **233**, 138–153.

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 Hydrotechnica* (in Slovene) **16/17**, 79-94.

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

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

NeuralMachine (2002) Software package. http://www.data-machine.com.

NeuroSolutions (2002) Software package. http://www.nd.com.

Persson, M. & Berndtsson, R. (2001) Using neural networks for calibration of time-domain reflectometry measurements. *Hydrol. Sci. J.* **46**(3), 389–398.

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

Rajurkar, M. P., Kothyari , U. C. & Chaube, U. C. (2002) Artificial neural networks for daily rainfall–runoff modelling. *Hydrol. Sci. J.* **47**(6), 865–877.

Sajikumar, N. & Thandaveswara, B. S. (1999) A non-linear rainfall–runoff model using artificial neural network. *J. Hydrol.* **214**, 32–48.

Shamseldin, A. Y. (1997) Application of a neural network technique to rainfall–runoff modelling. *J. Hydrol.* **199**, 272–294.

Shamseldin, A. Y. & O'Connor, K. M. (2001) A non-linear neural network technique for updating river flow forecasts. *Hydrol. Earth System Sci*. **5**(4), 577–597.

Solomatine, D. P. (2002) Data-driven modelling: paradigm, methods, experiences. In: *Proc. Fifth Int. Conf. on Hydroinformatics* (Cardiff, UK, July 2002). IWA Publishing, London, UK.

Tayfur, G. (2002) Artificial neural networks for sheet sediment transport. *Hydrol. Sci. J.* **47**(6), 879–892.

Witten, I. H. & Frank, E. (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, San Francisco, USA.

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.

Zealand, C. M., Burn, D. H. & Simonovic, S. P. (1999) Short term streamflow forecasting using artificial neural network. *J. Hydrol.* **216**, 32–55.

Received 6 June 2002; accepted 20 February 2003