Hydrol. Earth Syst. Sci. Discuss., 2, 365–415, 2005
www.copernicus.org/EGU/hess/hessd/2/365/
SRef-ID: 1812-2116/hessd/2005-2-365
European Geosciences Union

Hydrology and Earth System Sciences Discussions

Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation

N. J. de Vos[1] and T. H. M. Rientjes[1, 2]

[1]Section of Water Resources, Delft University of Technology, Delft, The Netherlands
[2]Department of Water Resources, International Institute for Geo-Information Science and Earth Observation, The Netherlands

Received: 4 February 2005 – Accepted: 16 February 2005 – Published: 23 February 2005

Correspondence to: N. J. de Vos (n.j.devos@citg.tudelft.nl)

Abstract

The application of Artificial Neural Networks (ANNs) on rainfall-runoff modelling needs to be researched more extensively in order to appreciate and fulfil the potential of this modelling approach. This paper reports on the application of multi-layer feedforward ANNs for rainfall-runoff modelling in the Geer catchment (Belgium) using both daily and hourly data. The good daily forecast results indicate that ANNs can be considered alternatives for traditional rainfall-runoff modelling approaches. However, investigation of the forecasts based on hourly data reveal a constraint that has hitherto been neglected by hydrologists. A timing error occurs due to a dominating autoregressive component that is introduced by using previous runoff values as ANN model input. The reason for the popular practice of using these previous runoff data is that this information indirectly represents the hydrological state of the catchment. Two possible solutions to this timing problem are discussed. Firstly, several alternatives for representation of the hydrological state are presented: moving averages over the previous discharge and over the previous rainfall, and the output of the simple GR4J model component for soil moisture. A combination of these various hydrological state representators produces good results in terms of timing, but the overall goodness of fit is not as good as the simulations with previous runoff data. Secondly, the use of a combination of multiple measures of model performance during ANN training is suggested, since not all differences between modelled and observed hydrograph characteristics such as timing, volume, and absolute values can be adequately expressed by a single performance measure. The possible undervaluation of timing errors by the commonly-used squared-error-based functions is a clear example of this inability.

1 Introduction

One of the main research challenges in hydrology is the development of computational models that are able to accurately simulate a catchment's response to rainfall. Such models are capable of forecasting future river discharge values, which are needed for hydrologic and hydraulic engineering design and water management purposes. However, simulating the real-world relationships using these Rainfall-Runoff (R-R) models is far from a trivial task since the various interacting processes that involve the transformation of rainfall into discharge are complex and variable. Hydrologists have attempted to address this modelling issue from two different points of view: using knowledge-driven modelling and data-driven modelling.

Knowledge-driven R-R modelling aims to reproduce the real-world hydrological system and its behaviour in a physically realistic manner. This way of R-R modelling is therefore based on detailed descriptions of the system and the processes involved in producing runoff. The best examples of knowledge-driven modelling are so-called physically-based model approaches, which generally use a mathematical framework based on mass, momentum and energy conservation equations in a spatially distributed model domain, and parameter values that are directly related to catchment characteristics. These models require input of initial and boundary conditions since flow processes are described by differential equations (Rientjes, 2004). Examples of physically-based R-R modelling are the Système Hydrologique Européen (SHE) (Abbott et al., 1986a, b) and the Representative Elementary Watershed (REW) (Reggiani et al., 2000; Reggiani and Rientjes, 2005) model approaches. Physically-based modelling suffers from drawbacks due to the complexity of the R-R transformation process in combination with limitations in representing the small-scale spatial variability of meteorological inputs, physiographic characteristics, and initial conditions in the model. Examples of drawbacks are excessive data requirements, large computational demands, overparameterisation effects, and parameter redundancy effects. This is what causes modellers to look for more parsimonious and simple model approaches that incorporate a higher degree of empiricism, but it is (still) not clear how far this empirical approach should be taken (cf. Nash and Sutcliffe, 1970; Beven, 2001a). Conceptual model approaches are a first step from physically-based model approaches in a more empirical direction. These model approaches use the principal of mass conservation in

combination with simplified descriptions of the momentum and energy equations. Conceptual modelling commonly implies that the model domain is represented by storage elements, either in a spatially lumped or semi-distributed manner. Well-studied examples of conceptual modelling are the HBV (Lindström et al., 1997), the TOPMODEL (Beven et al., 1995), and the Sacramento soil moisture accounting (Burnash, 1995) model approaches.

The data-driven approach to forecasting runoff from a catchment is based on extracting and re-using information that is implicitly contained in hydrological data without directly taking into account the physical laws that underlie the R-R processes (the most important of which is the principle of mass conservation). The field of data-driven modelling comprises a plethora of techniques (e.g. time series, empirical regression, fuzzy rule-based systems and Artificial Neural Networks modelling), mostly originating from statistics and artificial intelligence. Data-driven R-R models are generally quickly and easily developed and implemented and do not suffer from most the drawbacks of knowledge-driven models, but they have other disadvantages. Because of their low transparency, which results from the inability to interpret their internal workings in a physically meaningful way, they generally fail to give useful insights into the system under investigation. Furthermore, the range of application is limited because data-driven models only have validity over the range of the specific sample of the hydrological records that is used for model calibration.

A data-driven technique that has gained significant attention in recent years is Artificial Neural Network (ANN) modelling. In many fields, ANNs have proven to be good in simulating complex, non-linear systems. This awareness inspired hydrologists to carry out the earliest experiments using ANNs in the first half of the 1990s. Their promising results lead to the first studies specifically on ANNs in R-R modelling (e.g. Halff et al., 1993; Hjemfelt and Wang, 1993; Karunanithi et al., 1994; Hsu et al., 1995; Smith and Eli, 1995; Minns and Hall, 1996). The ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (ASCE, 2000) and Dawson and Wilby (2001) give good state-of-the-art reviews on ANN modelling in hydrology. The majority of studies have proven that ANNs are able to outperform traditional statistical R-R techniques (e.g. Hsu et al., 1995; Shamseldin, 1997; Sajikumar and Thandaveswara, 1999; Tokar and Johnson, 1999; Thirumalaiah and Deo, 2000; Toth et al., 2000) and produce comparable results to conceptual R-R models (e.g. Hsu et al., 1995; Tokar and Markus, 2000; Dibike and Solomatine, 2001). The field of R-R modelling by means of ANNs is nevertheless still in an early stage of development and remains a topic of continuing interest (see Anctil et al., 2004; Jain and Srinivasulu, 2004; Rajurkar et al., 2004, for example). More research is needed to support the discussion on the value of these techniques in this field, and to help realise its full potential.

In order to add to the knowledge about the relatively young field of ANN R-R modelling, we investigated several ANN design aspects through a case study. Multi-layer feedforward ANN models were developed for forecasting short-term streamflow. Both hourly and daily data sets from the Geer catchment (Belgium) were used to develop and to test these ANN models. We have particularly focused on the representation of the hydrological state (i.e. the amount and distribution of stored water in a catchment) in ANN models. Since the hydrological state for a great part determines the catchment's response to a rainfall event, it is critical as input to an ANN model. Previous discharge values are often used as ANN inputs, since these are indirectly indicative for the hydrological state. In this paper, we discuss the negative consequences of this approach and test several alternatives for state representation. Moreover, we discuss the evaluation of ANN models during the calibration phase and consequently the importance of a good choice of performance measures.

2 Artificial Neural Networks

2.1 History

The first theories on ANN techniques were conceived in the 1940s, and various relatively successful neural computers were built during the following two decades. After a

period of little development, interest in ANNs increased significantly in the late 1980s due to improvements on existing techniques in combination with the increase of computational resources. Since then, the field of ANNs has grown quickly, and the widespread applications of ANNs prove that their potential has been recognised in many fields such as earth sciences, economics, and health sciences.

2.2 Model description

An ANN is a mathematical model that consists of simple, densely interconnected elements known as neurons. These neurons are typically arranged in layers (see Fig. 1). An ANN receives signals through the input units and these signals are consequently propagated and transformed through the network towards the output neuron(s). In some ANNs, information always travels in the direction of the ANN output without delay. These so-called feedforward networks are also used in this study. One of the transformations performed by an ANN is multiplication with weights that express the strength of the connections between neurons. During a calibration procedure known as training, the internal pattern of connectivity between neurons – meaning the weights and biases, and therefore the model's response – is adapted to information that is presented to the network.

Figure 2 explains the transformations that data undergo in an ANN. The inputs of a neuron (either network inputs or output values from a preceding neuron) are multiplied with the weight that accompanies their connection (w). The results are summed and an additional value, a so-called bias (b), is commonly added to this value. The resulting net input (net) is transformed by a transfer function f into an activation value of the neuron, denoted in the diagram as Y. This activation value is then propagated to subsequent neurons.

2.3 Training

ANNs are trained by applying an optimisation algorithm, which attempts to reduce the error in network output by adjusting the matrix of network weights **W** and (optionally) the neuron biases. A common approach to ANN training in function approximation applications such as R-R modelling is to use supervised training algorithms. These algorithms are used in combination with sample input and output data of the system that is to be simulated. The weights are changed according to the optimisation of some performance measure, which is a measure for the degree of fit (or difference) between the network estimates and the sample output values. The alteration of network parameters in the training phase is commonly stopped before the training optimum is found, because the network will start learning the noise in the training data and lose its generalisation capability (overtraining). However, stopping too early means the ANN has not yet learnt all the information from the training data (undertraining). Both situations are likely to result in sub-optimal operational performance of an ANN model. It is for this reason that the available data are often split in three separate data sets: (1) the training set, (2) the cross-validation set, and (3) the validation set. The first provides the data on which an ANN is trained. The second is used during the training phase to reduce the chance of overtraining of the network. The minimisation of the training error is stopped as soon as the cross-validation error starts to increase. This point is considered to lie between undertraining and overtraining an ANN. Stopping earlier means that a network does not take full advantage of the information content of the input signals, and stopping later means that the networks fixates on the training data and loses its capability to generalise. The latter of the three data sets is used to validate the performance of a trained ANN. This so-called split-sampling method is also applied in this study.

2.4 Advantages and disadvantages

ANNs have advantages over many other techniques since they are able to simulate non-linearity in a system. They can also effectively distinguish relevant from irrelevant data characteristics. Moreover, they are non-parametric techniques, which means that ANN models do not necessarily require the assumption or enforcement of constraints or a priori solution structures (French et al., 1992). This, in combination with the fact that ANNs are able to self-adjust to information, makes that little expertise of the problem under consideration is needed for applying them successfully. Lastly, because of their compact and flexible model structure, ANNs have relatively low computational demands and can easily be integrated with other techniques.

A disadvantage of ANNs, however, is that the optimal form or value of most network design parameters (such as the number of neurons in the hidden layer) can differ for each application and cannot be theoretically defined, which is why they are commonly found using trial-and-error approaches. Another important drawback is that the training of the network parameters tends to be problematic, which is due to the following reasons: (1) optimisation algorithms are often unable to find global optima in complex and high-dimensional parameter spaces, (2) overparameterisation effects may occur, and (3) error minimisation in the training phase does not necessarily imply good operational performance. The latter pertains to the representativeness of the training data for the operational phase. For example, the training data should ideally reflect the distribution of variables in the operational situation, and should not contain too many errors.

2.5 Model evaluation

The most important measures for evaluating model performance that are used in this paper are the Root Mean Square Error (RMSE) and the Nash-Sutcliffe coefficient of efficiency (R^2) (Nash and Sutcliffe, 1970). The latter is formulated as:

$$R^2 = 1 - \frac{F}{F_0}, \quad (1)$$

where

$$F = \sum_{k=1}^{K} \left(Q_k - \hat{Q}_k\right)^2; F_0 = \sum_{k=1}^{K} \left(Q_k - \bar{Q}\right)^2. \quad (2)$$

F_0 is the initial variance for discharges and F is the residual model variance. In these equations, K is the total number of data elements, Q_k and $\hat{Q}_k$ are the observed and the computed runoffs at the k^{th} time interval, respectively and $\bar{Q}$ is the mean value of the runoff over time. R^2 values of 1 therefore indicate perfect fits.

Another performance measure that is used is the Persistence Index, PI, defined by Kitanidis and Bras (1980) as:

$$PI = 1 - \frac{F}{F_p}, \quad (3)$$

where

$$F_p = \sum_{k=1}^{K} \left(Q_k - Q_{k-L}\right)^2. \quad (4)$$

The difference with the R^2 is that the scaling of F for the PI involves the last known discharge value (at time k minus the lead time L) instead of the mean flow. This basically means that the model variance is compared with the variance of a very simple model that takes the last observation as a prediction.

At the start of each training trial, ANN weights and biases have to be initialised. The most-often applied method is random initialisation (Zijderveld, 2003). The goal of this randomisation is to force the training algorithm to search other parts of the parameter

space, thereby enabling a more robust overall optimisation procedure and increasing the overall chances of finding a global error minimum. A result of this approach is that the performance of an ANN is often different for each training trial, even if it is trained using the same algorithm. There are three reasons why training algorithms do not find the same parameter set for each training trial when training starts in a different part of the parameter space. First of all, there may be more than one global optima for the training set. Secondly, a training algorithm may not be able to find a global optimum and get stuck in local optima, on flat areas or in ridges on the error surface. Lastly, in case of applying cross-validation to prevent overtraining, the optimum in terms of the training data will probably not coincide with the optimum for the cross-validation set. Therefore, an algorithm might be stopped before finding a global optimum due to increasing cross-validation errors. In the case of random initialisation, the performances of the various training trials yield information on the parameter uncertainty of an ANN model type in combination with a certain training algorithm. Presenting this uncertainty allows for a more reliable and accurate comparison between combinations of ANN model types and training algorithms. Performing and presenting only a single training trial would be based on the assumption that this one trial represents a reliable indicator for the average performance, but our experience teaches us that this assumption is a risky one since ANN performance can be considerably variable between training trials. Gaume and Gosset (2003) were aware of this issue and addressed it by presenting ANN performance using Box-and-Whisker plots of the RMSE over an ensemble of 20 training trials. In our study, we present the mean and standard deviations of the performance measures over an ensemble of 10 training trials. This ensemble size was found to be appropriate for quantifying parameter uncertainty of our ANN models while keeping calculation times acceptable. Time series plots and scatter plots are presented for the median of the ensemble.

3 Application

3.1 Site of study and data

The Geer river (Fig. 3) is located in the north of Belgium, North West Europe, and contributes to the river Meuse. The river's catchment size is 494 km^2. The mean annual rainfall is approximately 810 mm, and the perennial river has discharges ranging from 1.8 m^3/s in dry periods to peaks of roughly 10 m^3/s.

Daily time series of rainfall at stations Waremme, Bierset and Visé, evaporation at Bierset, and streamflow at the catchment outlet at Kanne were available for the periods 1980–1991 and 1993–1997. For each variable, the time series over these two periods were connected into one time series. Except for evaporation, the continuity of the time series was largely preserved because the first period ended and the second period started with a period of low discharge and rainfall. Hourly time series of rainfall at station Bierset and streamflow at Kanne were available for the period 1993–1997. Figure 4 shows the daily catchment discharge in combination with the rainfall at location Bierset for the period 1980–1991. Figure 5 shows the hourly data for the period 1993–1997.

Both the daily and hourly time series were divided into 55% for training, 25% for cross-validation and 20% for validation (see Figs. 4 and 5). All three fragments of the time series start with a period of constant low discharge and rainfall.

3.2 ANN design

The ANN type that we used in this study is the static multi-layer feedforward network. Static networks do not have the dimension of time incorporated in the network architecture, as opposed to dynamic networks, which use feedback connections or local memories in neurons. These static ANNs are nevertheless able to integrate the time dimension in the network by using so-called "tapped delay lines". This method presents a sequence of time series values (e.g. $P(t), P(t-1), \ldots, P(t-m)$) as separate net-

work input signals. $P(t)$ represents an input variable in time and m the size of the window in time. The number of input units thus increases with the size of this window in time.

Increasing the number of parameters of an ANN by adding hidden neurons or layers, complicates network training. Moreover, a large number of parameters increases the chance of overtraining occurring. ANNs with one hidden layer are commonly used in rainfall-runoff modelling (see review by Dawson and Wilby, 2001), since these networks are considered to offer enough complexity to accurately simulate the dynamic and nonlinear properties of the rainfall-runoff transformation. Preliminary test results showed that these ANNs indeed outperformed the networks with two hidden layers. The optimal size of the hidden layer was found by systematically increasing the number of hidden neurons until the trained network performance no longer improved significantly. Figure 6 shows the performance of various ANN architectures in terms of the Nash-Sutcliffe coefficient. The ANN input for these simulations consisted of daily data with a total of 17 signals, concerning evaporation at one station, rainfall at three stations, and previous discharges. The results show that there is indeed a point at which the performance no longer increased (4 hidden neurons). Note that the 95% confidence bounds widen as the number of hidden neurons increases. This implies that the training algorithm is less likely to find optima as the dimensionality of the parameter space increases. We observed that the number of hidden neurons that resulted in parsimonious but well-performing ANN architectures was usually around the square root of the number of input neurons.

ANN architectures with one output neuron were consistently used throughout this study. The output signal from this neuron was the discharge prediction for a certain lead time. In order to make multi-step-ahead predictions (i.e. predictions with a lead time larger than one time step), two methods were available: (1) re-inputting a one-step-ahead prediction into the network, after which it predicts the two-step-ahead prediction, and so forth, and (2) by directly outputting the multi-step-ahead prediction. The first method uses the ANN's own preliminary estimations as a source of information for further predictions, the latter uses only the original data. Our test results showed that for both the daily and hourly data the two methods performed nearly similar up to a lead time of respectively 4 days and 12 h. Because of its simplicity, we have used the direct multi-step-ahead method.

Sigmoid functions are commonly used as transfer functions in hidden layers. We chose the popular hyperbolic tangent function, $a = \tanh(net)$. The identity function, $a = net$, was used as transfer function in the output neuron.

3.3 Input signals

The input signals to an ANN model should comprise all relevant information on the target output, and on the other hand, they should contain as less irrelevant information as possible (Zijderveld, 2003). However, in order to facilitate the training algorithm, largely overlapping information content of input signals should be avoided. Because an increased number of input signals leads to a more complex network structure, the task of training algorithms is complicated, which is likely to have a negative effect on network performance.

The number of input units increases with the memory length of a tapped delay line (as mentioned above). In order to make a parsimonious selection of ANN inputs, we examined the linear correlations between the input and output time series. This simple linear method does not reveal all information content that a non-linear technique such as an ANN is able to make use of, but from experience we know that it is adequate. Figures 7 and 8 show the correlation coefficients between various variables and the daily and hourly time series of runoff at Kanne for various time lags. The minimum and maximum delays were chosen in such a way as to enclose high values of the correlation for each variable, thereby ensuring high information content for each of the signals.

Because the hyperbolic tangent transfer function becomes saturated at a certain range, all input data are linearly scaled to a range of −1 to 1. The output of this transfer function is bounded to the range of −1 to 1, which is why the output data was scaled to

a range of −0.8 to 0.7. The reason for setting these ranges narrow is to enable the ANN to extrapolate beyond the training data range. The output data range is asymmetrical because it is more likely that the upper bound of the training data range is exceeded than the lower bound.

3.4 Training algorithms

The ANNs were trained using supervised training algorithms that tried to minimise an performance measure (often termed objective function from the point of view of calibration), namely the Mean Squared Error (MSE). Some popular algorithms were used for training, such as gradient descent techniques (e.g. backpropagation algorithm) and Newtonian optimisation techniques (e.g. Levenberg-Marquardt algorithm). For details on these and other popular training algorithms, see (Haykin, 1999) or (De Vos, 2003). A so-called batch training approach was used for training the ANNs: the whole training data set is presented once, after which the weights and biases are updated according to the average error.

The algorithms that were tested in this research are the backpropagation (BP), backpropagation with variable learning rate and momentum (BPvm), resilient backpropagation (RBP), Polak-Ribière, Fletcher-Reeves, and Powell-Beale conjugate gradient (CG-P, CG-F, CG-B), Broyden-Fletcher-Goldfarb-Shanno (BFGS), and Levenberg-Marquardt (L-M) algorithms. Table 1 shows the performance of these algorithms in terms of the mean RMSE and R^2, and the number of epochs. The latter gives an indication of the convergence speed of the algorithm.

The L-M algorithm outperformed the other algorithms in terms of accuracy and convergence speed in all test cases. Moreover, the standard deviation of the L-M algorithm was very low: 0.012 for daily data and 0.001 for hourly data. The other algorithms show much more spread in their performance measures (around 5 to 50 times more, depending on the algorithm), which indicates that the L-M algorithm is much more robust.

The above results show that ANN model performance can be be very dependent on the ability of optimisation algorithms to find a good set of weights and biases (also pointed out by, for example, Hsu et al., 1995). However, many studies on ANN R-R models have relied on training algorithms such as the classic backpropagation algorithm, backpropagation variants with momentum and/or variable learning rate, or conjugate gradient-based algorithms (see review by Dawson and Wilby, 2001). In our opinion, many studies using multi-layer feedforward ANNs for R-R modelling would benefit from using more sophisticated algorithms, such as L-M. Other alternatives, which were not tested here, are the LLSSIM algorithm (Hsu et al., 1995) and algorithms based on global optimisation, such as simulated annealing (see Kirkpatrick et al., 1983) and genetic algorithms (see Goldberg, 2000). The merits of using a good algorithm are threefold: (1) better accuracy leads to better ANN performance, (2) faster convergence leads to smaller calculation times, and (3) lower spread in the performance makes it easier and more honest to evaluate and compare ANNs. Unfortunately, few algorithms are able to combine these three merits.

4 Results and discussion

4.1 Main results

Figure 9 shows a scatter plot of the results of a one-day-ahead ($t+1$) prediction of an ANN model using the daily data from the Geer catchment. The input to the network consisted of three previous rainfall values (t to t-2) at the three available measurement stations, the evaporation from t-1 to t-5, and discharge values at the catchment outlet from t to t-2. The ANN architecture was: 17-5-1 (17 input units, 5 hidden neurons, 1 output neuron). A detail of the observed and predicted time series of the daily data is presented in Fig. 10. The ANN model shows to be able to make one-step-ahead forecasts with reasonable accuracy, considering the large ratio between lead time and catchment mean lag time (which is 8 h, see Fig. 8). The biggest drawback is that the model underestimates quite a number of moderate peak flows by up to −40%. Nevertheless, Fig. 10 also shows that the model's timing of the peaks is quite good.

Low flows are mostly well simulated, even though the model's low flow forecasts show more fluctuations than the observed flow pattern. This is most likely due to the model overestimating the effect of small rainfall events.

Scatter plots of simulation results of the ANN models based on hourly data are presented in Figs. 11 and 12. The first shows the results of a one-hour-ahead discharge forecast using an 18-5-1 ANN model with rainfall inputs from t-5 to t-19 and discharge from t to t-2. The latter presents the results of a six-hour-ahead forecast using a similar model and similar input signals (only the rainfall window of time is shifted to t to t-14). Figure 13 shows the mean R^2 for different lead times over an ensemble of ANN models, along with the 95% confidence bounds. The results show that the ANN models are able to make reasonably good forecasts (in terms of the Nash-Sutcliffe coefficient) for several hours ahead. When forecasting 9 or more hours ahead, the performance rapidly deteriorates. This is due to the fact that rainfall up to time t, which are used as input signals, no longer contains significant information on the forecasted discharge, because the catchment's mean lag time is exceeded (cf. Fig. 8).

The scatter plot with low spread, the low RMSE, and the high R^2 coefficient of the one-hour-ahead forecast indicate excellent model performance, but the PI does not (see Fig. 11). Moreover, the multi-step-ahead forecasts are disappointing, especially when compared with the forecast based on daily data. A visual interpretation of the simulation results, a detail of which is presented in Fig. 14, shows why: the prediction of the ANN model is lagged in comparison with the observed time series. This prediction lag effect is the result of using previous discharge values as ANN inputs. The high autocorrelation of the hourly discharge time series makes that the autoregressive model component, which is implicitly contained in ANN models that use previous discharge values, becomes dominant. The ANNs give the most weight to the latest discharge input (usually, Q at t) for calculating the forecast (Q at $t+L$). In other words, the ANN models say that the best forecast for the discharge over a certain lead time is around the value of the current discharge. In terms of most performance measures, this is indeed true for this case. As a consequence, the models underrate the information contained in other input signals.

The prediction lag effect is especially significant in forecasts with small lead times, but more practically relevant forecasts with longer lead times are also noticeably influenced by this effect. However, the longer the lead time L becomes, the lower the correlation between Q at t and Q at $t+L$ will be. As a result, the ANN model will give more weight to the rainfall information, which causes the prediction lags to decrease. Naturally, the overall performance in terms of squared errors also decreases with longer lead times. All this is depicted in Fig. 15, where forecast results for various lead times are evaluated in terms of R^2 (shown on the y-axis), for various shifts in time (shown on the x-axis). The ANN models that were used for these simulations are similar to the ones used to produce Figs. 11 to 13. The R^2 at zero shift corresponds to the actual performance of the forecast. The predicted time series is subsequently shifted in time against the observed time series, after which R^2 is recalculated. The time shift at which the R^2 coefficient is maximised, is an expression for the mean lag in the model forecast. This is done for a number of different lead times (the different lines). The idea for this method of timing evaluation is taken from Conway et al. (1998).

To the authors' knowledge, no previous researchers have appreciated prediction lags in ANN model forecasts and related this effect to the introduction of an autoregressive component by using previous discharge values. The issue has been remarkably overlooked even though various research results indicate that lags indeed occur in the ANN model forecasts (e.g. Campolo et al., 1999; Dawson and Wilby, 1999; Zealand et al., 1999; Thirumalaiah and Deo, 2000; Jain and Srinivasulu, 2004).

The one-day-ahead forecast of the previously discussed daily-data models outperforms the forecasts of the hourly-data models with a lead time of 6 h and more (both in terms of timing and R^2). The reason for this difference in performance is that the cross-correlations between daily rainfalls and discharge series are higher than those of the hourly series, while the autocorrelation of the daily discharge series is lower than that of the hourly series (shown in Figs. 7 and 8). As a result, the information content of the daily input data is more evenly spread over the various input signals and the

autoregressive component of the ANN R-R model does not become so dominant that a large number of lagged forecasts occur. It is important to realise that the importance of the prediction lag effect is thus not always significant.

Two sources of the prediction lag effect can be identified, each of which may be able to suggest other possible solutions. Firstly, there is the matter of ANN model input. If previous discharge values are used for hydrological state representation of the system, pronounced negative effects may be introduced in the form of prediction lags. Secondly, there is the difficulty of evaluating ANN model performance, especially during the training phase. The squared-error-based performance measure that we used for model training and validation is clearly not always strict enough to result in a satisfactory R-R model, since it may undervalue correct timing of the forecast. Both topics are discussed in the following two sections respectively.

4.2 Hydrological state representation

The hydrological state of a river basin prior to a rainfall event is important in governing the processes by which a catchment responds to this rainfall and the proportion of the input volume that appears in the stream as part of the hydrograph (Beven, 2001b). The majority of studies on ANNs in R-R modelling have used input signals that are merely indirectly related to the hydrological state. For example, previous values of discharge or water levels can be considered indirect indicators of the hydrological state of a catchment and are therefore often used as model inputs (e.g. Hsu et al., 1995; Minns and Hall, 1996; Campolo et al., 1999). Our study proves that this may not be a good solution, because the autoregressive model component that is thus introduced can become too dominant, resulting in lagged model forecasts. Another possible source of information for the hydrological state is the (weighted) cumulative rainfall over a preceding period of time (e.g. Shamseldin, 1997; Rajurkar et al., 2004). Air-temperature or (potential) evaporation time series are also often used in combination with rainfall time series (e.g. Zealand et al., 1999; Tokar and Markus, 2000). These evaporation and temperature data can be considered to account for losses in the water balance of a catchment, thereby adding to the information on the hydrological state. More direct indicators of the hydrological state are variables related to soil moisture and groundwater levels. Recent studies by Gautam et al. (2000) and Anctil (2004) have shown that time series of soil moisture measurements and estimations can be successfully used as ANN model input. De Vos (2003) and De Vos et al. (2005) have proved the value of groundwater level time series as ANN inputs.

Three alternatives for hydrological state representation were tested, evaluated and compared in terms of both squared error and timing. Tables 2 and 3, and Figs. 16 and 17 show the results of this. Firstly, a time series of the non-decaying moving average of the discharge (*Qma*) was used as ANN input. A moving average time series of the discharge can also be considered to represent the hydrological state and has the advantage over using discharge time series that the correlation with the ANN output is lower. The near absence of lags in the daily-data model forecasts and the decrease of the prediction lag effect with increased lead times (see Fig. 15) suggested that this approach would improve timing accuracy. We used a memory length of 192 h (eight days) for the moving average of the discharge. Secondly, time series of the non-decaying moving average of the rainfall (*Pma*) were tested. By trial and error, we found that using a memory length of 480 h produced the best results. Lastly, a number of simulations using the simple soil moisture reservoir component of the GR4J lumped conceptual rainfall-runoff model (Edijatno et al., 1999; Perrin et al., 2003) were performed to produce a time series of estimated soil moisture (*SM*). The hourly rainfall time series and temporally downscaled evaporation time series served as input to the GR4J soil moisture model component. The only parameter that needed to be defined is the reservoir's maximum capacity. Of the several values that were tested, a maximum capacity of 400 mm produced the best results. Anctil et al. (2004) have also used the GR4J model component to create soil moisture time series, which too were subsequently used as ANN input. The authors refer to their interesting paper, which gives a more extensive and in-depth presentation on the topic of combining soil moisture modelling with ANN R-R modelling.

Tables 2 and 3 show that the simulations with *Pma* and the *SM* time series are not affected by any prediction lags. The performance as indicated by the R^2 and *PI*, however, is mediocre and only slightly better than using only the *P* time series as ANN input. Using the *Qma* time series results in decreased (but still noticeable) prediction lags compared to the simulations with *Q*, but the R^2 and *PI* also decrease. Similar R^2 and *PI* results are produced by a combination of *Pma*, *Qma* and *SM*, but the prediction lag effect is almost eliminated. It is interesting to note that the test results show that any combination of these variables with *Q* still results in prediction lags, showing that the autoregressive component again dominates as a result of using *Q* as ANN input. In the case of six-hour-ahead forecasts, however, the average prediction lag decreases from -2 to -1 due to the additional information in the *Pma*, *Qma* and *SM* model inputs. This proves that even strongly dominant autoregressive model components can be corrected for timing errors by using additional input signals.

Figures 16 and 17 present details of the forecasted time series using the various hydrological state representators. The simulations with *Pma* show a consistent overestimation of low flows and an inaccurate reproduction of the shape of the recession curves. Moreover, most peak flows are underestimated, especially in the six-hour-ahead forecast. The models with *SM* underestimate high peak flows, but reproduce low flows and recession curves quite well (although there is a slight overestimation). There are abrupt changes in the slope of the recession curve, however, where a more gradual decrease of the discharge would be more accurate. This is a possible result of using the simple GR4J model for creating the *SM* time series, and other soil moisture models or soil moisture measurements might produce better results. The ANNs that used *Qma* as input show good overall performance but contain some inaccuracy due to fluctuations that occurred in periods of low flows. They were best at simulating peak flows, even though more than half of the peaks were still underestimated significantly (by 10% or more). Neither of the three alternatives can be considered very adequate as a sole representator of hydrological state. However, the simulations with all three alternatives for hydrological state representation (i.e. *Pma*, *Qma*, *SM*) show that the

ANN model attempts to combine the best of each alternative. This can be concluded from the good overall performance (resulting from the Qma input) and the correctly timed forecasts (resulting from the Pma and SM inputs). For the one-hour-ahead forecast, the information from all input signals is approximately equally weighted, and the six-hour-ahead forecast is slightly dominated by the information contained in Qma. Figure 18 shows scatter plots of the one-hour-ahead and the six-hour-ahead forecasts for this model type.

Note that in neither of the above simulations extreme peak flows are well approximated. One of the reasons for this is that the ANN models have difficulties dealing with the extremely nonlinear catchment response in the case of wet hydrological catchment state in combination with rainfall events. Another reason is that our ANN models attempt to simulate the complete range of the hydrograph and therefore may undervalue the high peak flow errors, since these flows occur only incidentally. Finally, there are only a few examples of extreme peak flows in the training data, and hence the model has only little information on these types of events, to which it can adapt.

Finding better ways of representing hydrological state is only a first step towards better ANN modelling of R-R processes. The various ANN input signals that serve as state representators can complement each other in terms of information content, but they are also likely to have some information overlap. The ability to exploit the total information content depends strongly on the training algorithm and the performance measure that this algorithm is trying to optimise. The following section will discuss the choice of performance measures in ANN training for R-R modelling.

4.3 Performance measures for ANN training

An ANN can be trained by applying an optimisation algorithm that tries to find parameter values that minimise the distance between model output and target data. This distance is commonly expressed by a single performance measure such as the MSE. Any single performance measure, however, may not adequately measure the ways in which the model fails to match the important characteristics of the target data (Yapo et

al., 1998). For example, our results show that the MSE performance measure may fail to penalise a time shift in time series, while correct timing is of utmost importance in forecasting discharge. Automated calibration algorithms exclude the use of visual assessments of the performance, but using more than one performance measure for evaluating model quality during the calibration procedure (commonly termed multi-objective calibration) may be a good alternative for this. In multi-objective model calibration, a number of performance measures (that are ideally unrelated) are aggregated into a single objective function that is to be minimised. Possible performance measures that can be combined are, for example, measures based on squared error, timing, and volume. An interesting study in relation to this is the one by Conway et al. (1998), who recognised the problem of lagged predictions in solar activity time series forecasting using ANNs. They suggested to train the ANNs using a multi-objective approach that aggregated a squared-error performance measure and a measure for the average prediction lag. The prediction lag effect was successfully eliminated at the cost of a significant increase of the RMSE.

The use of multiple performance measures for model evaluation in the calibration phase has gained growing attention of hydrologists in recent years (e.g. Yapo et al., 1998; Madsen, 2000; Seibert, 2000; Cheng et al., 2002). These applications have been on knowledge-driven hydrological model approaches, but it is likely that data-driven model approaches like ANNs will also benefit from such a calibration approach. The lack of physical laws in data-driven modelling approaches and the fact that they have many non-defined parameters that need calibration makes these models vulnerable to errors. A discussion of the topic of multi-objective calibration of (data-driven) R-R models, however, is outside the scope of this paper. For a thorough discussion on the merits and difficulties of multi-objective calibration in hydrological modelling we refer to Gupta et al. (1998).

5 Summary and conclusions

The purpose of this study was to find whether multi-layer, feedforward ANNs can be effectively used as R-R models, and to investigate the role of hydrological state representation in ANN R-R modelling. The results of the one-day-ahead forecasts using daily data were promising and in accordance with the consensus that (at least in some cases) ANNs are alternatives for traditional R-R modelling approaches. However, the simulations with hourly data were afflicted by a constraint of this modelling approach that has hitherto been neglected by hydrologists: the possibility of lags in the forecasts. Since they are considered indicators of the hydrological state, previous values of discharge are often used as ANN model inputs, which introduces an autoregressive model component in the ANN model. Our results show that this is not necessarily a good solution, because high autocorrelation of the discharge time series may result in an uneven spread of the information content in network input. This leads to the autoregressive model component becoming too dominant and the ANN model producing a forecast that is very similar to the last known discharge, effectively causing timing errors in the predictions. The prediction lag effect is especially significant for short lead times, but more practically relevant forecasts with longer lead times were also affected by it. This issue was discussed from two points of view: (1) hydrological state representation and (2) model performance measures for ANN training. Firstly, instead of representing the hydrological state using previous discharge, we tested a number of alternatives. The best results, in terms of timing and overall fit, were obtained using a combination of multiple hydrological state representators: a moving average over the previous discharge, a moving average over the previous rainfall, and the output of the simple GR4J soil moisture model. The usefulness of the latter proves that complementary conceptual models can be valuable additions to ANN model approaches. Secondly, we conclude that not all differences between modelled and observed hydrograph characteristics such as timing, volume, and absolute values can be adequately expressed by a single performance measure such as the MSE, which was used dur-

ing the automatic training procedure of the ANN. It is therefore our opinion that using multiple performance measures in the training phase is necessary to fully exploit the capabilities of ANNs for R-R modelling. These multiple performance measures should account for a more complete range of aspects on which R-R models are to be evaluated. Another conclusion on ANN training we draw from this study is that the choice of training algorithm greatly affects model performance in terms of accuracy, robustness, and training speed. For this reason, we recommend the use of sophisticated algorithms in hydrological ANN modelling.

References

Abbott, M. B., Bathurst, J. C., Cunge, J. A., O'Connell, P. E., and Rasmussen, J.: An introduction to the european hydrological system – Système Hydrologique Européen, "SHE", 1: History and philosophy of a physically-based, distributed modelling system, J. Hydrol., 87, 45–59, 1986.

Abbott, M. B., Bathurst, J. C., Cunge, J. A., O'Connell, P. E., and Rasmussen, J.: An introduction to the european hydrological system – Système Hydrologique Européen, "SHE", 2: Structure of a physically-based, distributed modelling system, J. Hydrol., 87, 61–77, 1986.

Anctil, F., Michel, C., Perrin, C., and Andrassian, V.: A soil moisture index as an auxiliary ANN input for streamflow forecasting, J. Hydrol., 286, 155–167, 2004.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology: Artificial neural networks in hydrology, II – Hydrologic applications, J. Hydrol. Eng., 5, 2, 124–137, 2000.

Beven, K. J.: How far can we go in distributed hydrological modelling?, Hydrol. Earth Sys. Sci., 5, 1–12, 2001a, SRef-ID: 1607-7938/hess/2001-5-1.

Beven, K. J.: Rainfall-Runoff Modelling: the Primer, John Wiley & Sons, Chichester, 2001b.

Beven, K. J., Lamb, R., Quinn, P. F., Romanowicz, R., and Freer, J.: TOPMODEL, in: Computer Models of Watershed Hydrology, edited by Singh, V. P., Water Resources Publications, Colorado, 627–668, 1995.

Burnash, R. J. C.: The NWS River Forecast System – catchment modeling, in: Computer Mod-

els of Watershed Hydrology, edited by Singh, V. P., Water Resources Publications, Colorado, 311–366, 1995.
Campolo, M., Andreussi, P., and Soldati, A.: River flood forecasting with a neural network model, Water Resour. Res., 35, 4, 1191–1197, 1999.
Cheng, C. T., Ou, C. P., and Chau, K. W.: Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration, J. Hydrol., 268, 72–86, 2002.
Conway, A. J., Macpherson, K. P., and Brown, J. C.: Delayed time series predictions with neural networks, Neurocomputing, 18, 81–89, 1998.
Dawson, C. W. and Wilby, R. L.: A comparison of artificial neural networks used for river flow forecasting, Hydrol. Earth Sys. Sci., 3, 529–540, 1999, SRef-ID: 1607-7938/hess/1999-3-529.
Dawson, C. W. and Wilby, R. L.: Hydrological modelling using artificial neural networks, Prog. Phys. Geog., 25, 80–108, 2001.
De Vos, N. J.: Rainfall-runoff modelling using artificial neural networks. Master's thesis, Delft University of Technology, 2003.
De Vos, N. J., Rientjes, T. H. M., and Pfister, L.: Groundwater levels as state indicator in rainfall-runoff modelling using Artificial Neural Networks, Proceedings NCR-days 2004, in press, 2005.
Dibike, Y. B. and Solomatine, D. P.: River flow forecasting using artificial neural networks, Phys. Chem. Earth (B), 26, 1, 1–7, 2001.
Edijatno, N., Nascimento, O., Yang, X., Makhlouf, Z., and Michel, C.: GR3J: a daily watershed model with three free parameters, Hydrol. Sci. J., 44, 2, 263–277, 1999.
French, M. N., Krajewski, W. F., and Cuykendall, R. R.: Rainfall forecasting in space and time using a neural network, J. Hydrol., 137, 1–31, 1992.
Gaume, E. and Gosset, R.: Over-parameterisation, a major obstacle to the use of artificial neural networks in hydrology?, Hydrol. Earth Sys. Sci., 7, 5, 693–706, 2003, SRef-ID: 1607-7938/hess/2003-7-693.
Gautam, M. R., Watanabe, K., and Saegusa, H.: Runoff analysis in humid forest catchment with artificial neural network, J. Hydrol., 235, 117–136, 2000.
Goldberg, D. E.: Genetic algorithms in search, optimization, and machine learning, Addison-Wesley-Longman, Reading, MA, 2000.
Gupta, H. V., Sorooshian, S., and Yapo, P. O.: Toward improved calibration of hydrologic mod-

els: Multiple and noncommensurable measures of information, Wat. Resour. Res., 34, 4, 751–763, 1998.
Halff, A. H., Halff, H. M., and Azmoodeh, M.: Predicting from rainfall using neural networks, Proc. Engrg. Hydrol., 760–765, 1993.
Haykin, S.: Neural networks, a comprehensive foundation. Prentice Hall, Upper Saddle River, NJ, 1999.
Hjemfelt, A. T. and Wang, M.: Artificial neural networks as unit hydrograph applications, Proc. Engrg. Hydrol., 754–759, 1993.
Hsu, K. L., Gupta, H. V., and Sorooshian, S.: Artificial neural network modeling of the rainfall-runoff process, Water Resour. Res., 31, 10, 2517–2530, 1995.
Jain, A. and Srinivasulu, S.: Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques, Water Resour. Res., 40, 4, 2004.
Karunanithi, N., Grenney, W. J., Whitley, D., and Bovee, K.: Neural network for river flow prediction, J. Comput. Civil Eng., 8, 2, 201–220, 1994.
Kirkpatrick, S., Gerlatt Jr., C. D., and Vecchi, M. P.: Optimization by simulated annealing, Science, 220, 671–680, 1983.
Kitanidis, P. K. and Bras, R. L.: Real-time forecasting with a conceptual hydrologic model, 2, applications and results, Wat. Resour. Res., 16, 6, 1034–1044, 1980.
Lindström, G., Johansson, B., Persson, M., Gardelin, M., and Bergström, S.: Development and test of the distributed HBV-96 hydrological model, J. Hydrol., 201, 272–288, 1997.
Madsen, H.: Automatic calibration of a conceptual rainfall-runoff model using multiple objectives, J. Hydrol., 235, 276–288, 2000.
Minns, A. W. and Hall, M. J.: Artificial neural networks as rainfall-runoff models, Hydrolog. Sci. J., 41, 3, 399–417, 1996.
Nash, J. E. and Sutcliffe, J. V.: River flow forecasting through conceptual models; part I – a discussion of principles, J. Hydrol., 10, 282–290, 1970.
Perrin, C., Michel, C., and Andréassian, V.: Improvement of a parsimonious model for streamflow simulation, J. Hydrol., 279, 275–289, 2003.
Rajurkar, M. P., Kothyari, U. C., and Chaube, U. C.: Modeling of the daily rainfall-runoff relationship with artificial neural network, J. Hydrol., 285, 96–113, 2004.
Reggiani, P. and Rientjes, T. H. M.: Flux parameterization in the Representative Elementary Watershed (REW) approach: application to a natural basin, Water Resour. Res., accepted,

2005.

Reggiani, P., Sivapalan, M., and Hassanizadeh, S. M.: Conservation equations governing hill-slope responses: exploring the physical basis of water balance, Water Resour. Res., 36, 7, 1845–1863, 2000.

Rientjes, T. H. M.: Inverse modelling of the rainfall-runoff relation; a multi objective model calibration approach, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, 2004.

Sajikumar, N. and Thandaveswara, B. S.: A non-linear rainfall-runoff model using an artificial neural network, J. Hydrol., 216, 32–55, 1999.

Seibert, J.: Multi-criteria calibration of a conceptual runoff model using a genetic algorithm, Hydrol. Earth Syst. Sc., 4, 2, 215–224, 2000, SRef-ID: 1607-7938/hess/2000-4-215.

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

Smith, J. and Eli, R. N.: Neural-network models of rainfall-runoff process, J. Water Resour. Plng. and Mgmt., 121, 6, 499–508, 1995.

Thirumalaiah, K. and Deo, M. C.: Hydrological forecasting using neural networks, J. Hydrol. Eng., 5, 2, 180–189, 2000.

Tokar, A. S. and Johnson, P. A.: Rainfall-runoff modeling using artificial neural networks, J. Hydrol. Eng., 4, 3, 232–239, 1999.

Tokar, A. S. and Markus, M.: Precipitation-runoff modeling using artificial neural networks and conceptual models, J. Hydrol. Eng., 5, 2, 156–160, 2000.

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

Yapo, P. O., Gupta, H. V., and Sorooshian, S.: Multi-objective global optimization for hydrologic models, J. Hydrol., 204, 83–97, 1998.

Zealand, C. M., Burn, D. H., and Simonovic, S. P.: Short term streamflow forecasting using artificial neural networks, J. Hydrol., 214, 32–48, 1999.

Zijderveld, A.: Neural network design strategies and modelling in hydroinformatics, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, 2003.

Table 1. ANN model performance using various training algorithms.

	Daily data			Hourly data		
Algorithm	RMSE	R^2	Epochs	RMSE	R^2	Epochs
BP	1.275	–1.868	1000	0.572	0.411	800
BPvm	0.926	–0.568	140	0.948	–0.502	20
RBP	0.690	0.223	30	0.279	0.871	80
CG-P	0.770	0.010	25	0.206	0.929	60
CG-F	0.519	0.519	60	0.185	0.941	80
CG-B	0.425	0.706	50	0.164	0.956	90
BFGS	0.567	0.427	30	0.182	0.942	100
L-M	0.339	0.815	20	0.151	0.963	40

Table 2. ANN model performance for one-hour-ahead forecast using various methods of hydrological state representation.

Input	Time window	Architecture	Mean R^2	St. dev. R^2	Mean PI	St. dev. PI	Avg. lag
P	–5 to –19	15–4–1	0.513	0.047	–10.676	1.134	0.1
P Q	–5 to –19 0 to –2	18–5–1	0.963	0.001	0.121	0.020	–1.0
P Qma	–5 to –19 0 to –2	18–5–1	0.803	0.020	–3.557	0.494	–1.0
P Pma	–5 to –19 0 to –2	18–5–1	0.479	0.057	–11.403	1.398	0.0
P SM	–5 to –19 0 to –2	18–5–1	0.560	0.022	–9.540	0.535	0.0
P Qma Pma SM	–5 to –19 0 to –2 0 to –2 0 to –2	24–5–1	0.656	0.044	–7.238	1.054	–0.1
P Q Qma Pma SM	–5 to –19 0 to –2 0 to –2 0 to –2 0 to –2	27–5–1	0.964	0.002	0.133	0.035	–1.0

Table 3. ANN model performance for six-hour-ahead forecast using various methods of hydrological state representation.

Input	Time window	Architecture	Mean R^2	St. dev. R^2	Mean PI	St. dev. PI	Avg. lag
P	0 to –14	15–4–1	0.491	0.032	–0.258	0.079	0.0
P Q	0 to –14 0 to –2	18–5–1	0.791	0.006	0.482	0.015	–2.0
P Qma	0 to –14 0 to –2	18–5–1	0.682	0.012	0.213	0.029	–-0.8
P Pma	0 to –14 0 to –2	18–5–1	0.521	0.061	–0.185	0.150	0.0
P SM	0 to –14 0 to –2	18–5–1	0.558	0.054	–0.092	0.134	0.0
P Qma Pma SM	0 to –14 0 to –2 0 to –2 0 to –2	24–5–1	0.688	0.016	0.229	0.039	–0.1
P Q Qma Pma SM	0 to –14 0 to –2 0 to –2 0 to –2 0 to –2	27–5–1	0.806	0.014	0.518	0.035	–1.0

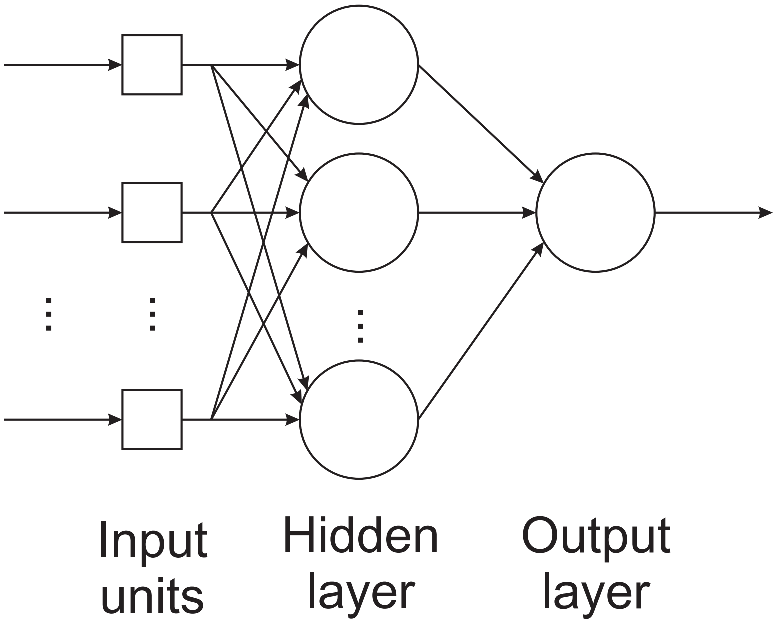

Fig. 1. An exemplary feedforward ANN with one hidden layer. The input units are not considered neurons since they do not transform data and merely pass information to the network.

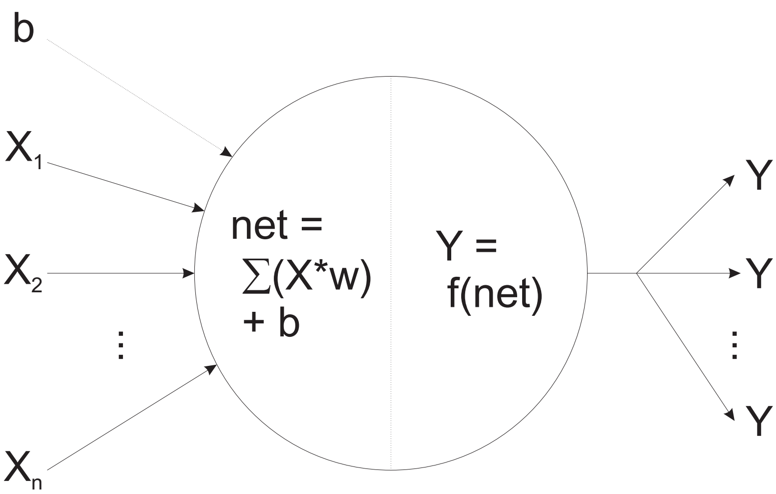

Fig. 2. Schematic representation of the transformations inside artificial neurons.

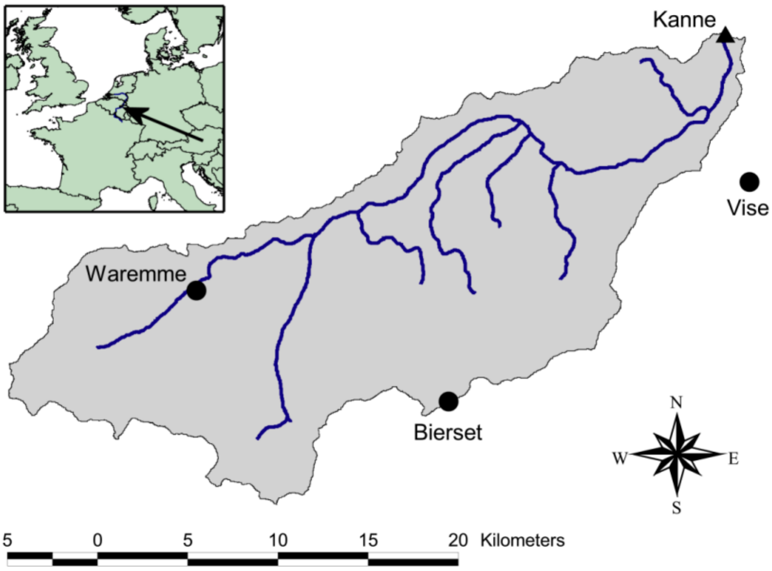

Fig. 3. Map of the Geer catchment, showing various measurement stations.

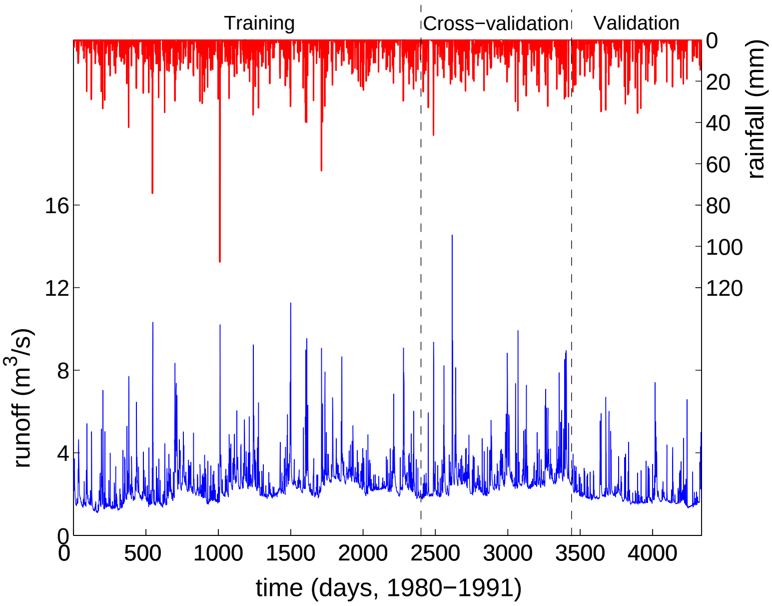

Fig. 4. Daily runoff (Kanne) and rainfall (Bierset) from 1980 to 1991.

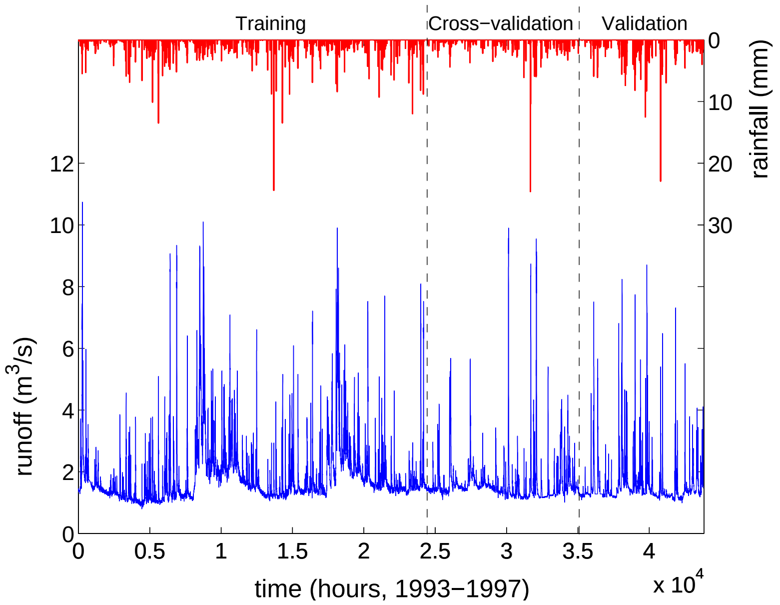

Fig. 5. Hourly runoff (Kanne) and rainfall (Bierset) from 1993 to 1997.

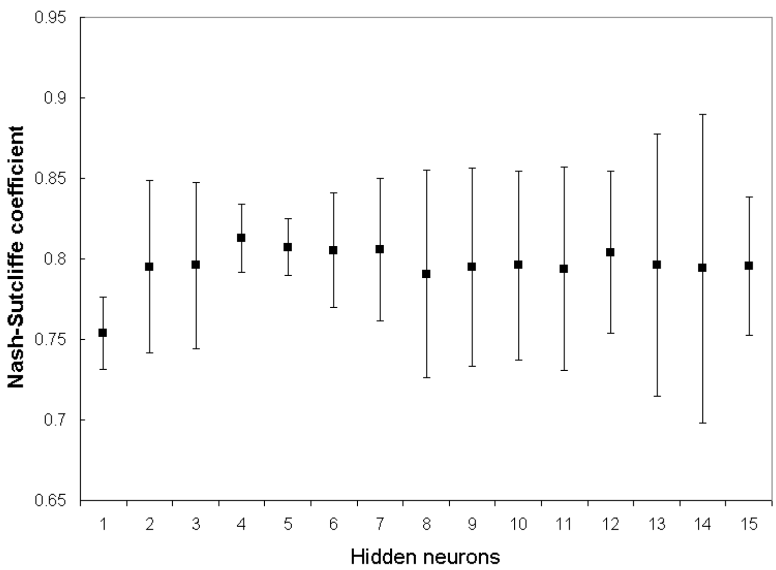

Fig. 6. ANN performance for various hidden layer sizes. The squares represent the mean Nash-Sutcliffe coefficients R^2, and the bars depict the 95% confidence bounds.

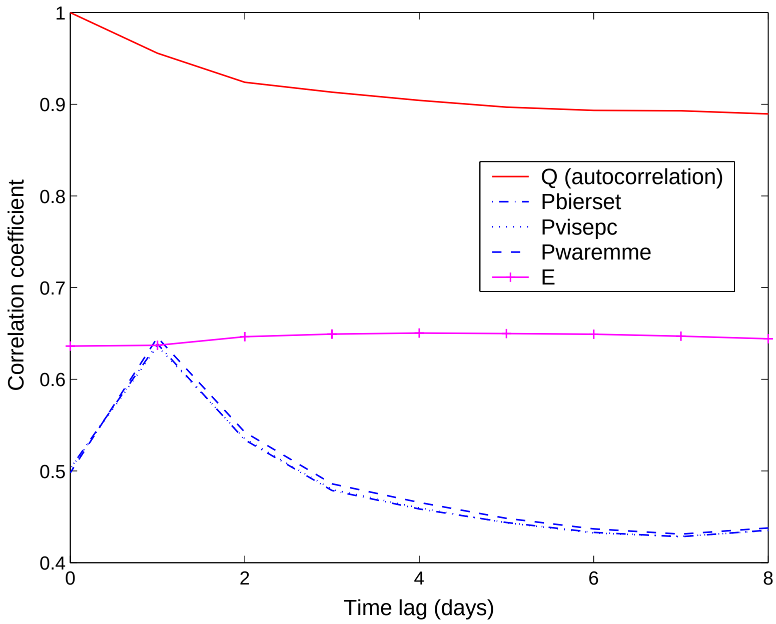

Fig. 7. Correlation with the daily runoff time series for various variables and lags.

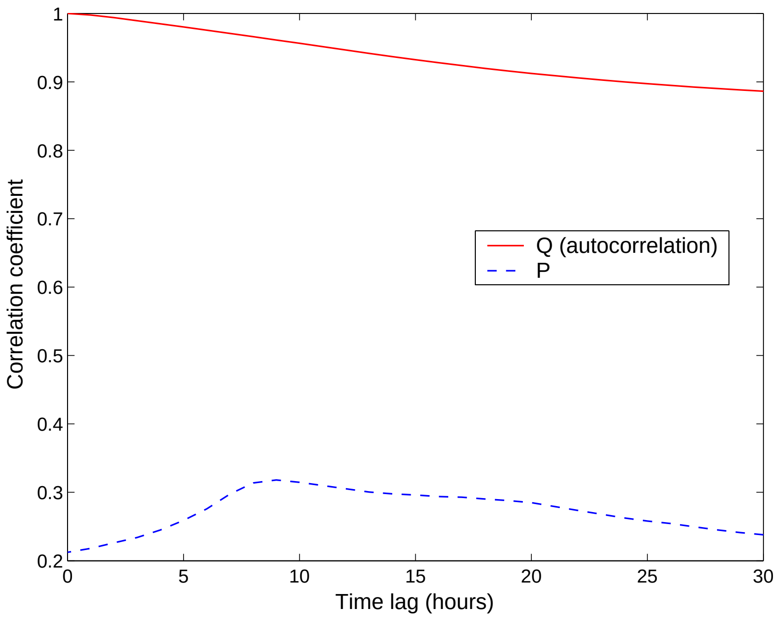

Fig. 8. Correlation with the hourly runoff time series for rainfall and runoff and for various lags.

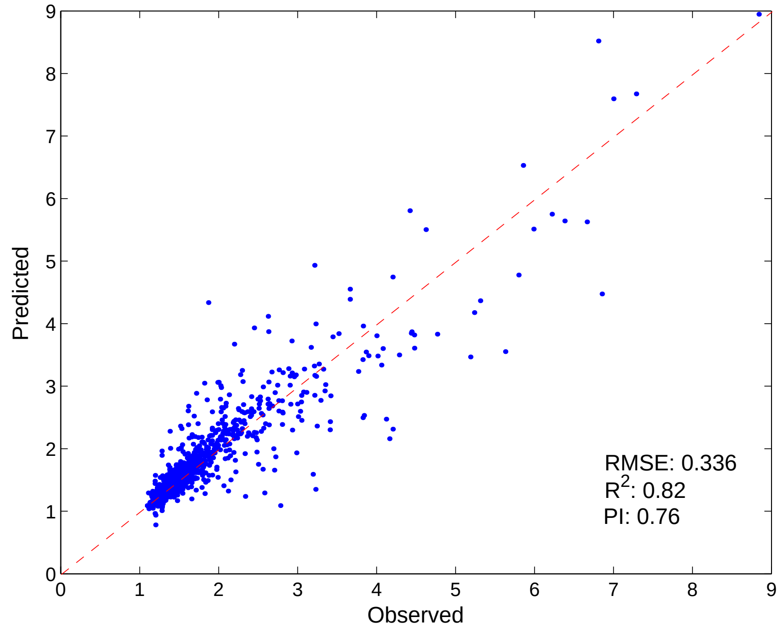

Fig. 9. Scatter plot of predicted versus observed daily discharges for a one-day-ahead forecast.

403

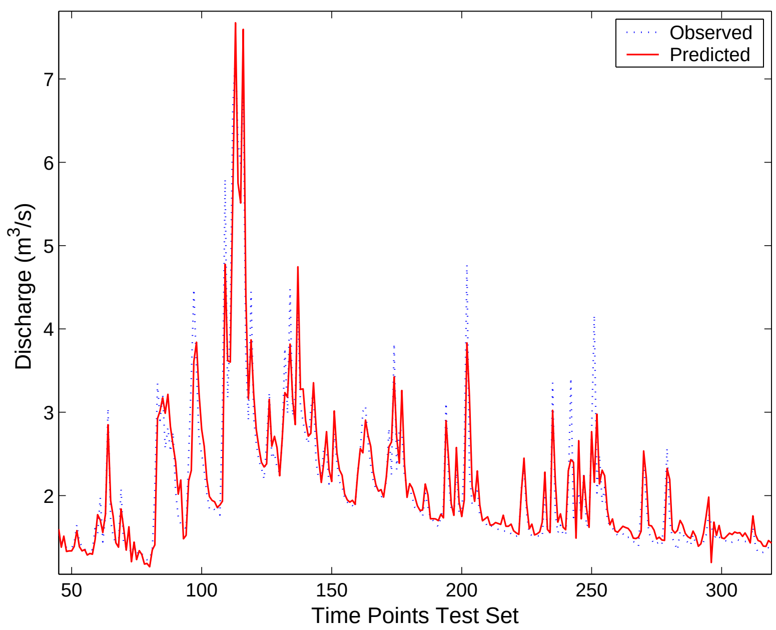

Fig. 10. Observed and predicted daily time series for a one-day-ahead forecast (detail).

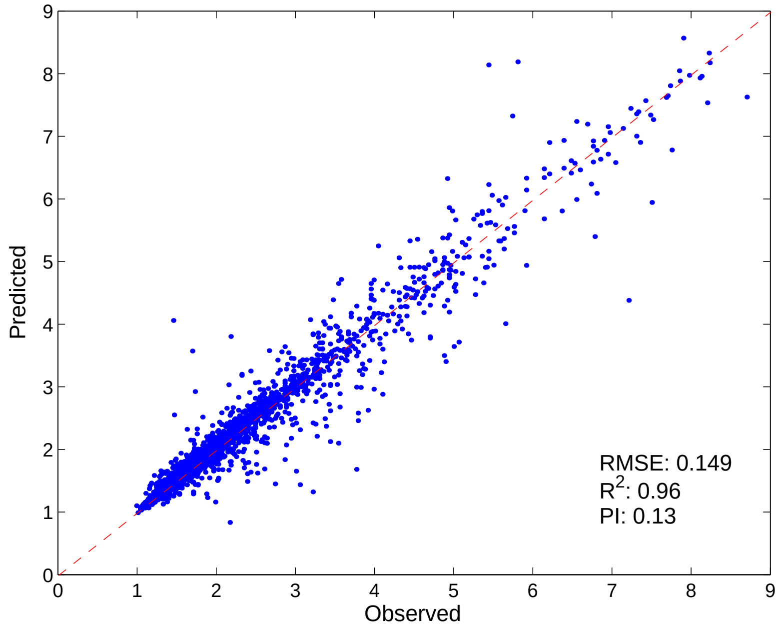

Fig. 11. Scatter plot of predicted versus observed hourly discharges for a one-hour-ahead forecast based on historical rainfall and discharge values.

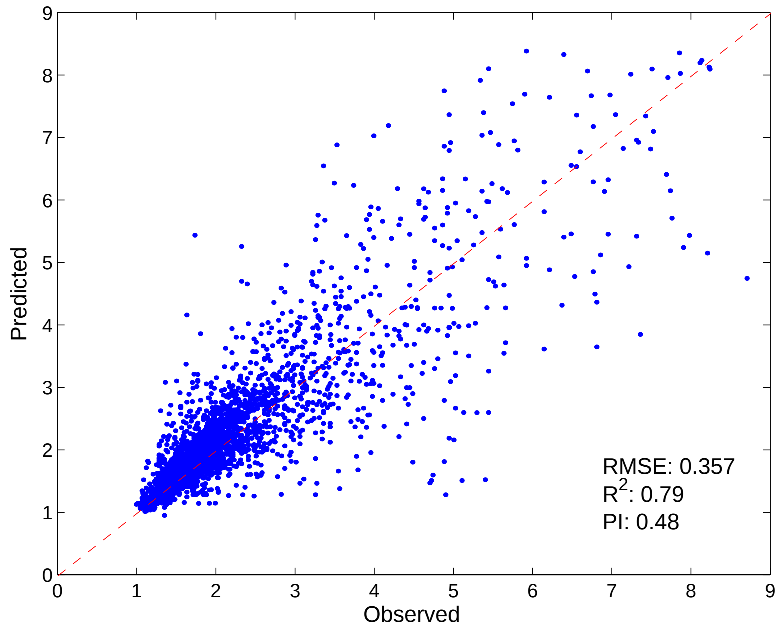

Fig. 12. Scatter plot of predicted versus observed hourly discharges for a six-hour-ahead forecast based on historical rainfall and discharge values.

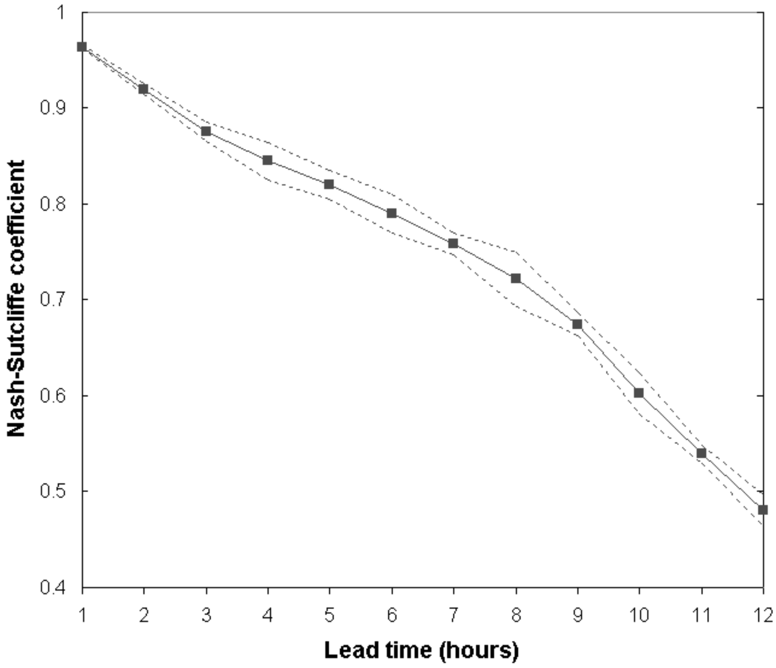

Fig. 13. ANN performance for multi-step-ahead predictions, in terms of the Nash-Sutcliffe coefficient R^2, with 95% confidence bounds.

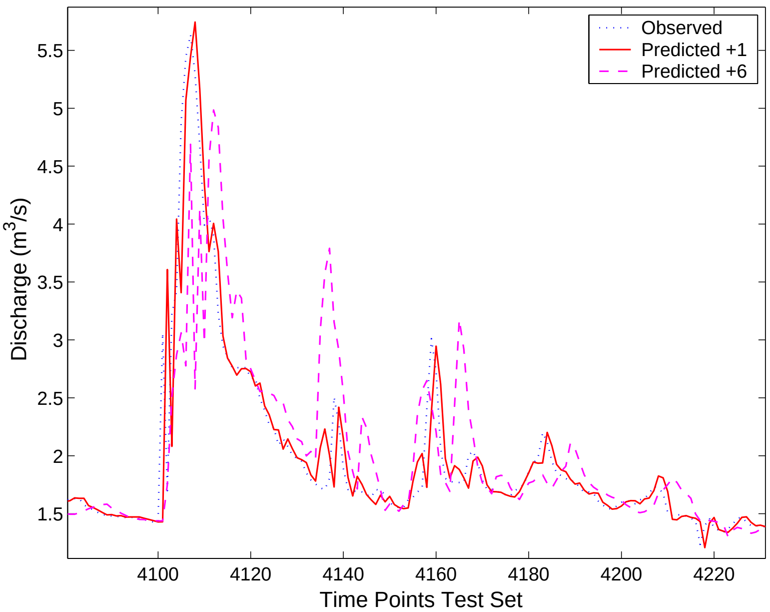

Fig. 14. Observed and predicted hourly time series for a one-hour-ahead and a six-hour-ahead forecast (detail).

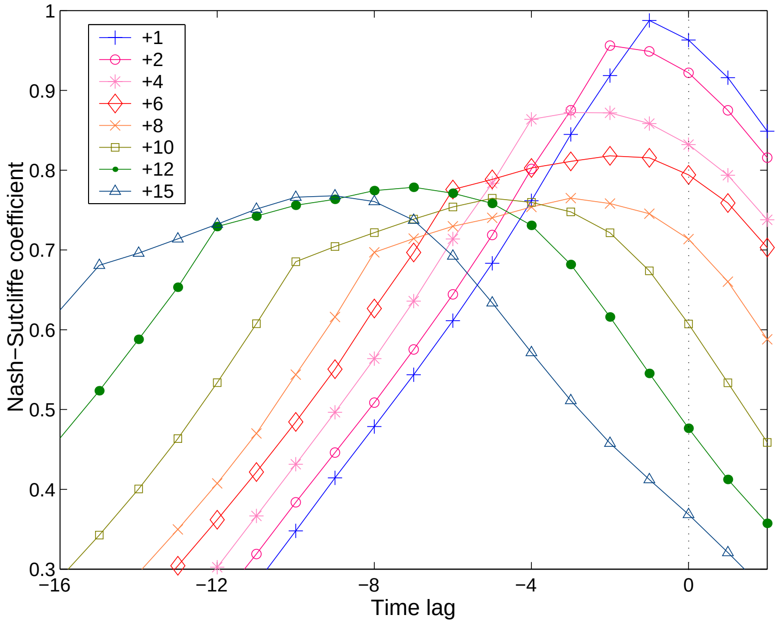

Fig. 15. ANN multi-step-ahead forecast performances (in terms of the Nash-Sutcliffe coefficient) for various shifts in time.

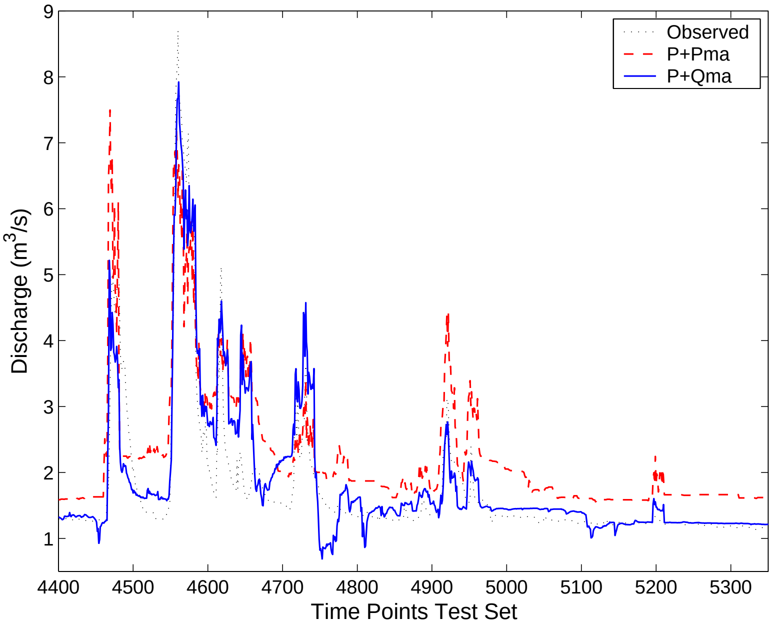

Fig. 16. (a) For the caption, please see next page.

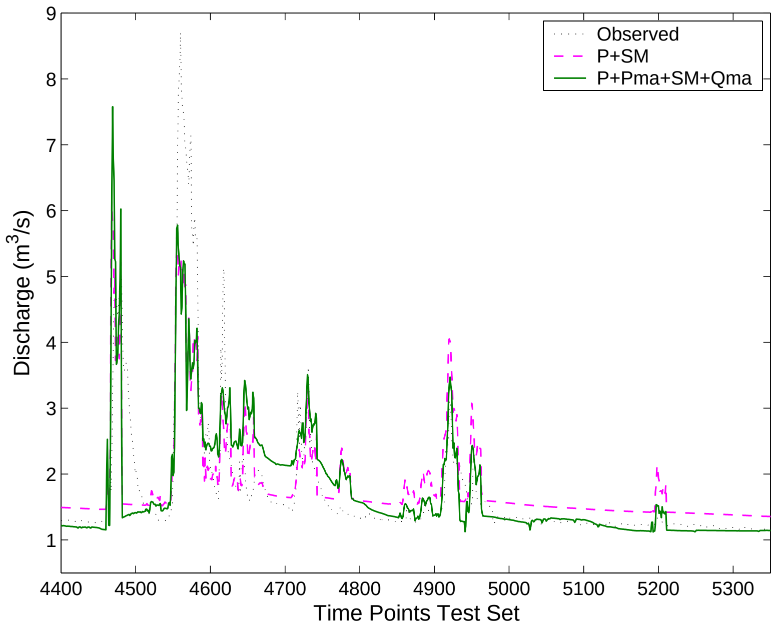

Fig. 16. **(b)** Details of ANN model results for one-hour-ahead forecasted time series using various methods of hydrological state representation.

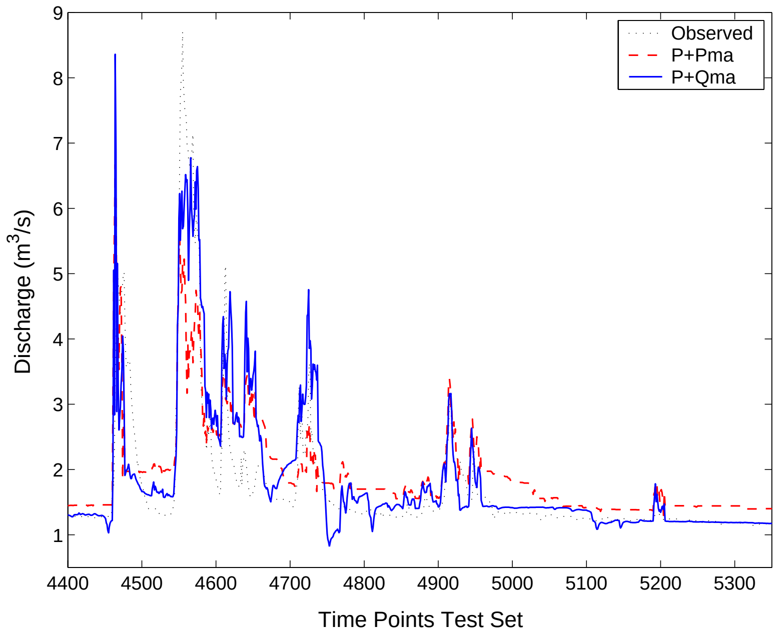

Fig. 17. **(a)** For the caption, please see next page.

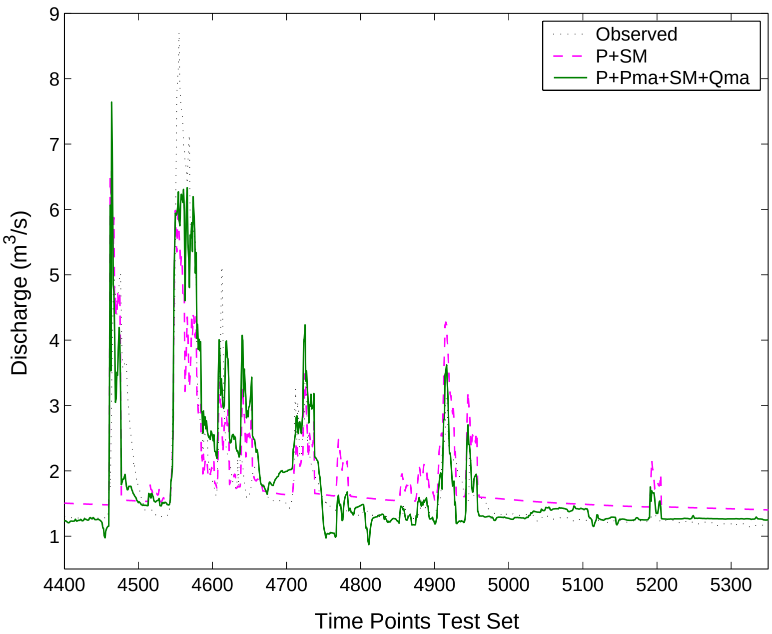

Fig. 17. (b) Details of ANN model results for six-hour-ahead forecasted time series using various methods of hydrological state representation.

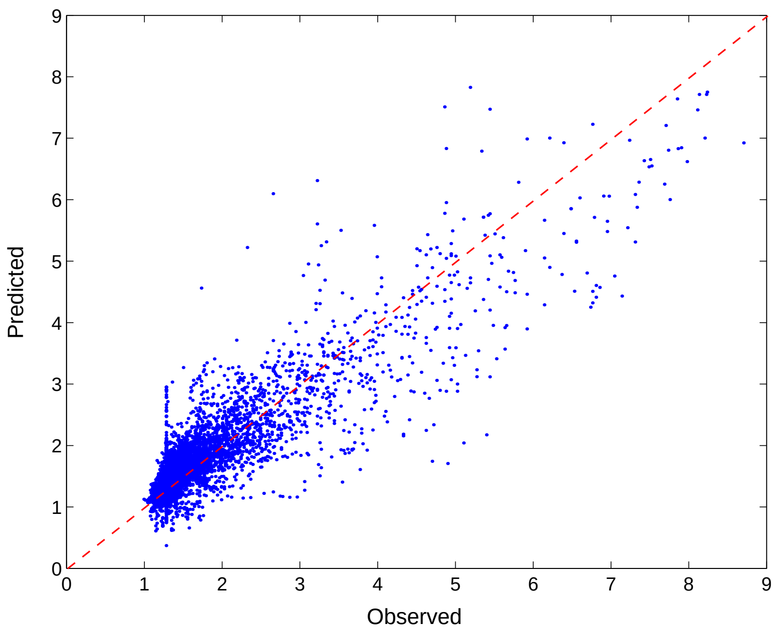

Fig. 18. (a) For the caption, please see next page.

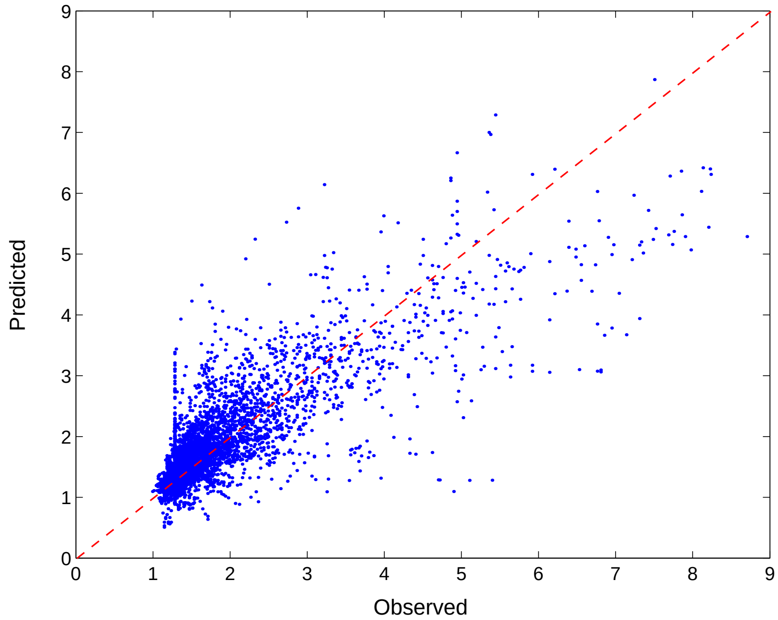

Fig. 18. Scatter plots of predicted versus observed hourly discharges for **(a)** a one-hour-ahead and **(b)** a six-hour-ahead forecast based on P, Pma, SM, and Qma inputs.