Research Repository UCD

Title	A comparative study of three neural network forecast combination methods for simulated river flows of different rainfall-runoff models
Authors(s)	Shamseldin, Asaad Y., O'Connor, Kieran M., Nasr, Ahmed Elssidig
Publication date	2007-10
Publication information	Shamseldin, Asaad Y., Kieran M. O'Connor, and Ahmed Elssidig Nasr. "A Comparative Study of Three Neural Network Forecast Combination Methods for Simulated River Flows of Different Rainfall-Runoff Models." Taylor & Francis, October 2007. https://doi.org/10.1623/hysj.52.5.896.
Publisher	Taylor & Francis
Item record/more information	http://hdl.handle.net/10197/2421
Publisher's version (DOI)	10.1623/hysj.52.5.896

Downloaded 2026-05-01 23:37:50

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

This is an electronic version of an article published inHydrological Sciences Journal, Volume 52, Issue 5 October 2007 pages 896 - 916. Hydrological Sciences Journal is available online at:
http://www.informaworld.com/smpp/title~db=all~content=t911751996

A Comparative Study of Three Neural Network Forecast Combination Methods for simulated river flows of different Rainfall-runoff models

Asaad Y. Shamseldin[(1)*], Kieran M. O'Connor[(2)] and Ahmed E. Nasr [(3)]

[(1)] Department of Civil and Environmental Engineering, The University of Auckland, Private Bag 92019, Auckland, New Zealand
[(2)] Department of Engineering Hydrology, National University of Ireland, Galway, Galway, Ireland
[(3)]Department of Civil Engineering, University College Dublin, Earlsfort Terrace, Dublin 2, Ireland

Abstract

The performance of three artificial neural network (NN) methods for combining simulated river flows, based on three different neural network structures, are compared. These network structures are; the simple neural network (SNN), the radial basis function neural network (RBFNN) and the multi-layer perceptron neural network (MLPNN). Daily data of eight catchments, located in different parts of the world, and having different hydrological and climatic conditions, are used to enable comparisons of the performances of these three methods. In the case of each catchment, each neural network combination method synchronously uses the simulated river flows of four rainfall-runoff models operating in design non-updating mode to produce the combined river flows. Two of these four models are black-box, the other two being conceptual models. The results of the study show that the performances of all three combination methods are, on average, better than that of the best individual rainfall-runoff model utilized in the combination, i.e. that the combination concept works. In terms of the Nash-Sutcliffe R^2 model efficiency index, the MLPNN combination method generally performs better than the other two combination methods tested. For most of the catchments, the differences in the R^2 values of the SNN and the RBFNN combination methods are not significant but, on average, the SNN form performs marginally better than the more complex RBFNN alternative. Based on the results obtained for the three NN combination methods, the use of the multi-layer perceptron neural network (MLPNN) is recommended as the appropriate NN form for use in the context of combining simulated river flows.

Keywords (neural network combination methods, rainfall-runoff model, river flow simulation, multi-layer perceptron, radial basis function, simple neural network)

1. Introduction

This paper is concerned with the further development of simulated river flow combination methods in which the synchronous simulated river flows from a number of competing rainfall-runoff models are used to produce combined or '*consensus*' simulated flows (see figure 1). The justification for using the combination technique in the context of rainfall-runoff modelling rests on the simple fact that as there is *no perfect model* available at present to truly and consistently define the rainfall-runoff transformation process, the multi-model (consensus/aggregation) approach offers the real prospect of better river flow simulation than that of the best model included in the combination. Such model inadequacy has been demonstrated in numerous rainfall-

* Corresponding author

runoff model inter-comparison studies which showed that rainfall-runoff models are not free from *failure* regardless of their complexity or ascribed physical attributes (cf. WMO, 1992; Ye *et al.*, 1997; Perrin *et al.*, 2001; Butt *et al.*, 2004). As the available simulated river flows resulting from each individual competing model may well capture, to a greater or lesser degree, different specific aspects of the true nature of the catchment response to rainfall, a judicious combination of these flows would logically be expected to provide more reliable flow simulations.

Few hydrological studies have been published which deal extensively with developments and applications of combination methods to models which either have different structures or have the same structure but use different parameter sets. A number of both linear (e.g. regression) and non-linear (e.g. neural network and fuzzy-based) combination methods have been developed (Shamseldin *et al.*, 1997; Shamseldin and O'Connor, 1999; See and Openshaw, 2000; Xiong *et al.*, 2001; Butt *et al.*, 2004; Coulibaly *et al.*, 2005; Oudin *et al.*, 2006; Kim *et al.*, 2006; Fenicia *et al.*, 2007) and applied to produce combined river flows. Such studies have demonstrated the potential capabilities of the combination technique in improving the accuracy and reliability of simulated river flows. In this paper, three forms of neural network combination methods are (NNMs) considered for that purpose.

Over the last decade or more, neural networks have become a very popular mathematical modelling tool in hydrology and water resources. They are generally employed as alternatives to the traditional models (e.g. Hsu *et al.*, 1995 and Karunanithi *et al.*, 1994). However, despite justifiable claims that the neural network models are very flexible and versatile, it has not been convincingly demonstrated that they are universally superior to the traditional models. Moreover, there is no denying the fact that the 'black-box' NNMs are generally non-parsimonious and hence prone to '*over-fitting*' and the phenomenon of '*equifinality*' (Beven and Binley, 1992). To advance this debate, more comprehensive comparisons not only of the simulation performance of various neural network configurations operating as rainfall-runoff models is required but also, as presented in this paper, comparisons of the performance of alternative types of neural networks methods applied in the multi-model/consensus context.

Shamseldin *et al.* (1997) advocated the use of neural network models as an appropriate and sensible method for the combination of simulated river flows of a

suite of rainfall-runoff models. In the combination context, the neural networks may be regarded as non-linear multiple-input single-output black-box models; the inputs at each time step being the constituent model-simulated discharges and the output being the combined '*consensus*' discharge. Similar to the more traditional black-box models used in surface water hydrology, they are trained or calibrated using observed sets of input-output data. The use of neural networks as a model-output combination system is still novel and differs from their other hydrological applications in that the neural network works in concert with, rather than in competition with, the constituent rainfall-runoff models in order to produce better river flow simulations than those of the individual models contributing to the combination.

The results of the neural network combination method (NNM) are found to be generally better than those of the linear weighting (WAM) and fuzzy-based methods (Shamseldin *et al.*, 1997; See and Abrahart, 2001, Xiong et *al.*, 2001). However, these conclusions are dependent not only on the *goodness* of the model simulated discharges included in the combination but also on the type of NNM employed.

Various types of neural network models are available for use in river flow simulation and flow forecasting. The multi-layer perceptron neural network (MLPNN) has been the popular choice in applications of neural networks as river flow forecasting models (cf. Dawason and Wilby, 2001; Maier and Dandy, 2000) and, thus far, it has been the only neural network type used in river flow combination (cf. Shamseldin *et al.*, 1997; See and Abrahart, 2001). Other types of neural networks used as substantive (as distinct from combination) models for simulating river flows include the radial basis function neural networks (RBFNNs), the complex modular neural networks, and the range-dependent/threshold neural networks. Bruen and Yang (2005) applied functional neural networks in the context of real-time river flow forecasting. However, relatively few of these studies have been focussed on comparisons of the performance of different neural network types, and generally such comparisons were conducted using the data of just a few catchments.

For example, Mason *et al.* (1996) concluded that, when used as a rainfall-runoff model, the performance of the RBFNN is comparable to that of the MLPNN. Dawson and Wibly (1999) compared the performance of the MLPNN and the RBFNN using the data of the River Mole in the UK and noted that the performance of the RBFNN is inferior to that of the MLPNN. Hu *et al.* (2001) developed a range (thresholds) dependent network which employs a number of MLPNNs to model the

river flow in different flow bands of magnitude (e.g. high, medium and low). Their results indicated that the range-dependent network performed better than the MLPNN. Zhang and Govindarajui (2000) used a complex modular network to model the rainfall-runoff transformation in four basins. This network is based on dividing the input space into multiple-regions with 'soft boundaries' where a particular external input vector can synchronously belong to more than one region. A number of network modules is used to model different regions of the input space with an additional gating network being used to integrate the results of the different network modules. Their results showed the modular network to be superior to the MLPNN in terms of model simulation performance. However, the range-dependent and the modular neural networks suffer from the disadvantage that they are overly complex, having too many parameters (i.e. not parsimonious) and can therefore be quite difficult to calibrate, a problem due partly to the phenomenon of 'equifinality' (Beven and Binley, 1992) long associated with the traditional conceptual rainfall-runoff models.

The main objective of the present paper is, therefore, a comparison of the performance of three neural network structures, in the context of combining the simulated river flows, in order to establish if the use of neural network types other than the multi-layer perceptron (*the only type hitherto used for simulated river flow combination*) can improve on the performance achieved by the MLPNN. In the present neural network inter-comparison study, the daily data of eight catchments are used, these being located in different parts of the world and having different hydrological and climatic features. A brief description of these catchments is shown in Table (1).

The three neural network structures selected for the present study are the standard multi-layer perceptron neural network (MLPNN), the simple multi-layer perceptron neural network (SNN) and the radial basis function neural network (RBFNN). These neural networks are used to combine the simulated discharges provided by the four selected rainfall-runoff models operating in '*the design non-updating mode'*. These models are: the Linear Perturbation Model (LPM), the Linearly Varying Gain Factor Model (LVGFM), the Soil Moisture Accounting and Routing (SMAR) Model and the Probability-Distributed Interacting Storage Capacity (PDISC) model. The first two of these models are 'black-box' models, the LPM exploiting seasonality and the LVGFM employing a storage-based coefficient of runoff. The remaining two are conceptual models. Further details on these models

and their applications are given by Nash and Barsi (1983), Moore (1985), Kachroo (1992), Ahsan and O'Connor (1994), Tan and O'Connor, (1996), Shamseldin *et al.* (1997), Senbeta *et al.* (1999), and O'Connor *et al.* (2002).

2. Neural Network Combination Methods

2.1 The Multi-Layer Perceptron Neural Network MLPNN

The MLPNN used in this study is composed of three neuron layers, namely, the *input layer*, the *output layer* and the *hidden layer* (see figure 2). Although the MLPNN can have more than one hidden layer, having more than one hidden layer is rarely beneficial (Masters, 1993) and can lead to gross over-parameterization.

For a particular time period *i*, the input layer of the MLPNN used for river flow combination receives the external input vector $\boldsymbol{Z_i}$ having the synchronous simulated discharges of the different models for that period as its elements. This external input vector $\boldsymbol{Z_i}$ has the form

$$\mathbf{Z}_i = \left(\hat{Q}_{i,1}, \hat{Q}_{i,2}, ..., \hat{Q}_{i,N}\right)^T \quad (1)$$

in which $\hat{Q}_{j,i}$ is the estimated discharge of the *j-th* rainfall-runoff model for the *i-th* time period and *N* is the number of constituent individual models chosen for combination. Each of the individual model-estimated discharges is assigned as an input to one neuron in the input layer. Thus, the total number of the neurons in the input layer is equal to the number *(N)* of individual models chosen for combination. The input-output transformation for each input-layer neuron is achieved by a direct identity transformation in which the output of each neuron is equal to its external input.

A hidden layer is an intermediate layer, which adds a degree of flexibility to the performance of the neural network enabling it to deal efficiently with complex non-linear problems. Each neuron in the single hidden layer receives the same input vector of *N* elements from the neurons of the input layer, as defined by equation (1), and produces a single output. The input-output transformation in each hidden neuron is achieved by a mathematical non-linear transfer (or activation) function which, in the context of the MLPNN river flow combination, can be expressed as

$$\mathrm{Y}_{i,k} = \mathrm{f}(\sum_{j=1}^{\mathrm{N}} w_{j,k} Q_{i,j} + w_{\mathrm{o},k}) \qquad (2)$$

$Y_{i,k}$ being the output of the k^{th} hidden neuron for the i-th time period, $f(\,)$ the non-linear transfer function, $w_{j,k}$ the connection weight, which is assigned to the connection pathway between the k^{th} hidden neuron and the j^{th} neuron in the previous (input) layer, and $w_{o,k}$ the threshold (or bias) of the k^{th} hidden neuron. The same non-linear mathematical transformation function is generally used for all of the hidden neurons. A number of non-linear functions have been suggested to be used in conjunction with the hidden neurons (cf. Fausett, 1994, pp. 17-19; Masters, 1993, pp. 81-82). However, the most widely used non-linear transfer function in neural network applications is the logistic function (Blum, 1992, p. 39). This sigmoid function, as used in the present study, has the form:

$$\mathrm{Y}_{i,k} = \left[1 + Exp\left(- \sum_{j=1}^{\mathrm{N}} Q_{i,j} w_{j,k} - w_{\mathrm{o},k} \right) \right]^{-1} \qquad (3)$$

which is bounded between 0 and +1. In contrast to the input layer, the number of hidden neurons *M* is unknown *a priori* and can only be decided upon by trial and error.

The output layer is the last layer, having a single neuron which produces the final network output. This single output neuron receives an input array $\tilde{Z}_i = (Y_{i,1}, Y_{i,2}, ..., Y_{i,M})^T$ having the outputs of the *M* hidden neurons as its elements. The input-output transformation for this single neuron is similar to that of the hidden neurons, i.e. that based on the logistic function. Accordingly, the final output of the network, $Y_{f,i}$, for the i^{th} time step has the form

$$Y_{f,i} = \left[1 + Exp(- \sum_{k=1}^{M} w_k Y_{i,k} - w_o) \right]^{-1}$$

$$= \left[1 + Exp\left\{ - \sum_{k=1}^{M} w_k \left(1 + Exp\left(- \sum_{j=1}^{N} Q_{i,j} w_{j,k} - w_{o,k} \right) \right)^{-1} - w_o \right\} \right]^{-1} \qquad (4)$$

where w_k is the connection weight between the k^{th} neuron in the hidden layer and the single output neuron and w_o being the corresponding output neuron threshold value.

The connection weights and the threshold values of the hidden and output neurons network, which are effectively the parameters of the network, are to be estimated by training (i.e. calibration).

The use of the logistic function in conjunction with the output neuron implies that the final network output $Y_{f,i}$ lies within the bounded range [0,1] of the logistic function. However, as the actual observed discharges are outside this range, a rescaling scheme for the observed discharges is therefore required in order to compare the actual observed discharges and the network-estimated outputs. A linear rescaling function, which is commonly adopted in similar studies (e.g. Shamseldin *et al.*, 1997; Shamseldin *et al.*, 2002) is used for rescaling the observed discharges. This has the form

$$Q_{si} = a + b(\frac{Q_i}{Q_{max}}) \tag{5}$$

where Q_{si} is the rescaled discharge, a and b are the parameters of the rescaling function, Q_i is the observed discharge and $Q_{\max}$ is the maximum actual observed discharge in the calibration period. In the present study, the values of the rescaling parameters a and b are set equal to 0.1 and 0.75 respectively, resulting in the rescaled discharge Q_{si} range bounded by 0.1 and 0.85. The rationale behind using such a reduced rescaling range is to (i) avoid the occurrence of extreme values which adversely affect the performance of derivative-based calibration algorithms, and more importantly, (ii) to enable the network to simulate discharge values greater than those occurring in the calibration period.

In some applications where the external input variables to the network have different orders of magnitude and measurement units, the rescaling of these input variables would also be necessary in order to avoid numerical problems during the neural network calibration (Masters, 1993, pp. 254-267). As the external input variables used in the present study are simulated discharges of different models, having the same order of magnitude and the same measurement units, such rescaling of the external input variables is not needed in this study.

2.2 **The Simple Neural Network**

The Simple Neural Network (SNN) is introduced in the present study as a naïve NNM but also to test whether or not the use of a non-complex neural network can be effective as a river flow forecast combination method. The SNN is a special limiting case of the MLPNN having input and output layers only, and no hidden/intermediate layer(s). The roles of the input and the output layers in the SNN are the same as those of the MLPNN. The input layer transmits the external input vector $\mathbf{Z}_i = \left(\hat{Q}_{i,1}, \hat{Q}_{i,2}, ..., \hat{Q}_{i,N}\right)^T$ to the single neuron of the output layer using the direct linear '*identity*' transformation, the elements of which are subsequently assigned individual weights w_j, as in the case of the MLPNN, with a constant threshold w_o. The output neuron transforms the outputs received from the input layer into the final network output using the non-linear logistic function. This input-output transformation has the form

$$Y_{f,i} = \left[1 + Exp\left(-\sum_{j=1}^{N} w_j \hat{Q}_{i,j} - w_o\right)\right]^{-1} \tag{6}$$

Similar to the MLPNN, the final network output $Y_{f,i}$ of the SNN will be bounded in the range [0,1]. Accordingly to facilitate the calibration of the SNN and the comparison of network outputs with the actual observed discharges equation, (5) is used in the SNN for rescaling the observed discharges Q_i. The N connection weights w_j and the neuron threshold value w_o are effectively the parameters of the SNN to be estimated by calibration.

2.3 Radial Basis Function Neural Network

The Radial Basis Function Neural Network (RBFNN) was suggested as an alternative to the MLPNN for solving complex modelling problems (Luo and Unbehaven, 1997). In principle, the configuration of the RBFNN is quite similar to that of the MLPNN. However, there are differences between the RBFNN and the MLPNN regarding the mathematical operations involved in the input-output transformation. The RBFNN has only three layers (input layer, intermediate/radial-basis layer and output layer). Hence, in contrast to the MLPNN, which can have more

than one intermediate (hidden) layer, the RBFNN is designed to have one intermediate layer only.

The input layer receives the external input vector $Z_i = \left(\hat{Q}_{i,1},\hat{Q}_{i,2},...,\hat{Q}_{i,N}\right)^T$ defined by equation (1) and, similar to the input layer of the MLPNN, it transmits this external vector into the network using the direct linear '*identity*' transformation.

Each k^{th} neuron in the intermediate (radial basis) layer receives the external input vector transmitted from the input layer. Similar to the MLPNN, the appropriate number (M) of the neurons in this layer is unknown *a priori* and must be determined by trial and error.

The process of the input-output transformation in the neurons of the hidden layer is achieved using the non-linear radial basis function. There are a number of radial basis functions, such as the Gaussian function, the multi-quadratic function, and the inverse multi-quadratic functions, which can be used in conjunction with the neurons in the intermediate layer (Mason *et al.*, 1996). However, the most widely used radial basis function, chosen also for the purpose of the present study, is the Guassian function. This function is symmetrical and has only positive values bounded between 0 and 1.

The operation of the radial basis function, for the *i-th* time interval, depends on the calculation of the Euclidean norm (i.e. the distance $d_{i,k}$ between the input vector $\boldsymbol{Z_i}$ and the neuron centre vector C_k), defined by the equation:

$$d_{i,k} = ||Z_i\text{-}C_k||^2 = \left(\sum_{j=1}^{N}(\hat{Q}_{i,j} - C_{j,k})^2\right)^{1/2} \qquad (6)$$

where $d_{i,k}$ is the Euclidian distance of the k^{th} intermediate neuron and $\boldsymbol{C}_k$ is the corresponding centre vector having N elements, $C_{1,k}$, $C_{2,k}$, $C_{3,k}$……and $C_{N,k}$. In the case of the Gaussian function, the distance $d_{i,k}$ is standardised by dividing by the neuron scale parameter, σ_k, and the output $Y_{i,k}$ of the intermediate layer neuron is given by

$$Y_{i,k} = e^{-\left(\frac{d_{i,k}}{\sigma_k}\right)^2} = e^{-\left(\frac{\sum_{i=1}^{N}(\hat{Q}_{j,k} - C_{j,k})^2}{\sigma_k^2}\right)} \qquad (7)$$

Similar to the MLPNN and SNN, the output layer of the RBFNN has a single neuron. This single output neuron receives an input array $\tilde{Z}_i = \left(Y_{i,1},Y_{i,2},...,Y_{i,M}\right)^T$

from the intermediate layer, the elements of which are the *M* outputs of the neurons in that layer.

In contrast to the non-linear logistic function used as the transfer function for the output neuron of the MLPNN and the SNN, the simpler linear identity transfer function is adopted for the output neuron of the RBFNN, the final network output, $Y_{f,i}$, being simply

$$Y_{f,i} = \sum_{k=1}^{M} w_k Y_{i,k} + w_o \qquad (8)$$

The main advantage of using such a linear transformation in the output layer of the RBFNN is that there is no need to rescale the external outputs of the network to facilitate the network calibration.

3. Calibration of the MLPNN, SNN and RBFNN

The calibration of the three selected neural networks requires the estimation of the network parameters as well as the determination of the appropriate number neurons in the hidden/intermediate layer in the case of MLPNN and RBFNN. In the case of the MLPNN and SNN, the network parameters are the connection weights between adjacent layers and the corresponding neuron threshold value. In the case of the RBFNN, the centre vectors, the neuron scaling parameters of the radial basis transfer function, the connection weights and the neuron threshold value of the output neuron, constitute the parameters of the network to be estimated by calibration.

In the case of the MLPNN and RBFNN, the determination of the appropriate number of neurons in the hidden layer is essential for their success. The use of too few neurons in the hidden layer will have adverse effects on the performance of the network, i.e. the network will be prove inadequate and fail to capture the essence of the input-output transformation. On the other hand, the use of too many neurons in the hidden layer results in a complex network having too many parameters leading to over-fitting the training/calibration data set, with perhaps poor simulation performance in the independent verification period. The optimum number of hidden neurons is estimated in this study by a sequential trial and error procedure, which involves successive calibration of the network with an increasing number of hidden neurons. The number finally adopted as the *'optimum'* is that which has near-maximum performance in calibration with as few neurons as necessary to achieve that

condition, thereby striking a pragmatic balance between the network performance and its complexity, i.e. the virtue of parameter parsimony is acknowledged.

The parameters of the three selected neural networks are estimated by a sequential search procedure, which minimizes the least-squares objective function (i.e. sum of the squares of the errors between the estimated and rescaled observed outputs). This sequential procedure involves the use of the genetic algorithm (GA, a probabilistic global search method) and the conjugate gradient method (CG, a deterministic local search method). In this sequential GA-CG search procedure, the final optimized parameter values of the genetic algorithm are used as the initial starting parameter values for the conjugate gradient method. The motivation for using the sequential procedure is basically to enhance the prospects of finding what are ultimately considered to be the global optimum parameter values.

The genetic algorithm (GA), developed by Holland (1975), has been used for the calibration of rainfall-runoff models by Wang (1991) and also by Franchini (1996). A global heuristic search method, the operation of the GA is broadly based on the Darwinian theory of '*survival of the fittest*', as potential solutions to a given optimization problem contend and '*mate*' with each other to evolve improved solutions (van Rooij *et al.*, 1996, pp. 18-24). The GA codes the parameter values as genes in a chromosome and uses probabilistic rules to advance the search process (cf. Khosla and Dillon, 1997, p. 46; Wang, 1991). Although now applied as a general optimization algorithm in disciplines far removed from its origins, use of the original terminology of the GA persists. The starting point for the operation of the GA is the random generation of an initial population of parameter sets. From this initial population, pairs of parameters sets are randomly chosen depending on their fitness evaluated on the basis of the value of the selected objective function. The chosen pairs are subsequently used to generate a new population of parameters sets (i.e. the next generation) utilizing the genetic operators of '*crossover'* and '*mutation'* to generate '*offspring'* (Goldberg, 1989). The newly generated population is anticipated to be better than the older one. The process of generating new populations continues until a pre-specified '*stopping-criterion*' is fulfilled (e.g. when the specified number of function evaluations or the selected degree of convergence of successive values of the objective function is reached).

In contrast to the GA, the conjugate gradient (CG) optimisation method is a local search method but, like the GA, is still viewed as being suitable for solving complex large-scale optimisation problems involving the estimation of a large number of parameters (Haykin, 1999, p. 236). It is considered to be more robust and efficient than the back-propagation method which is commonly used for training (i.e. calibrating) the MLFFNN (Masters, 1993, pp. 105-111). The operation of the conjugate gradient method requires information on the gradient (i.e. the first-order derivative) vector of the objective function for each value of the objective function. The parameter values are adjusted iteratively in the method and this adjustment process depends on the specification of a search direction vector (cf. Haykin, 1999, p. 243: Press *et al.*, 1992, pp. 413-418). At any particular iteration, the search direction vector is constructed so that it is conjugate to (i.e. linearly independent of) the previous search directions. The parameter values are systematically adjusted by conducting line searches along the search direction vector. As in the case of the GA, the parameter adjustment process ends when a predetermined '*stopping-criterion*' is activated.

4. **Application of the three Neural Network Combination Methods**

The four selected individual rainfall-runoff models (i.e. LPM, LVGFM, PDISC and SMAR) are applied to the daily data of the eight selected catchments. The chosen catchments are from different geographical locations in the world and have various sizes and climatic conditions (see table 1). The available data for these catchments are split into two non-overlapping parts. The first part, containing the early years of the available record, is the *calibration period* which is used to estimate the parameters of the model by minimizing an objective function reflecting the sum of the squares of the differences between the observed and the model estimated discharges. The second part, contains the subsequent remaining years of the record, is the *verification period* which is used for the purpose of inferring whether the level of accuracy obtained during the calibration period would persist in other data samples using the parameter set obtained from the calibration period. For consistency in assessing the improvement in the accuracy of the daily discharge forecasts obtained using the three neural network combination methods, the individual rainfall-runoff

models and the combination methods are calibrated and verified using the same calibration and verification periods.

The four models are calibrated using a global objective function which directly reflects the least-squares criterion (i.e. the sum of the squares of deviations between the model-estimated and the observed discharges). Having calibrated the four individual rainfall-runoff models to the data of the eight test catchments, their estimated discharges are generated. In the case of each catchment, the estimated discharges of these four models for a particular day are the inputs to each of the three neural network combination methods for that day.

The three neural network combination methods are calibrated by minimizing the least-squares objective function using the sequential optimisation procedures described earlier. The calibration of both the MLPNN and the RBFNN requires the *a priori* specification of the number of hidden neurons. In this study, the optimum number of hidden neurons is estimated by a sequential trial and error procedure in which the MLPNN and the RBFFN are successively calibrated using two and three hidden neurons. Figures (3) and (4) show comparison of the R^2 values of the MLPNN and the RBFFN combination methods for two and three hidden neurons respectively for the calibration and the verification periods. Examination of the figures show that, for most of the catchments, there is no substantial gain in terms of network performance by using more than two hidden/intermediate neurons. In the verification period, the R^2 values obtained using three hidden neuron are in some cases actually lower than those obtained using two hidden neurons. This may be an indication of over-fitting the data in the three-neuron case. On the basis of the calibration period results, the number of hidden neurons is fixed at two in the present study.

In addition, numerical experiments are conducted to test the stability of the results of the sequential optimization method used to calibrate the three neural network combination methods. In these numerical experiments, the calibration process is repeated five times. The results of these experiments show that the R^2 values of each of the three methods show little variation between these repeated calibrations. However, the optimum parameter sets obtained are not the same. This

suggests the existence of equifinality in the sense that different parameter sets yield similar model performance.

In the present study, the performances of the four selected models, as well as those of the three neural network combination methods, are evaluated quantitatively using the following numerical model assessment criteria:

1. The R^2 criterion (Nash and Sutcliffe, 1970) which enables the evaluation of the performance of models operating on different catchments as well as on the same catchment. The criterion directly reflects the least squares objective function F (i.e. sum of the squares of the differences between the simulated and observed discharges) which is usually used in model calibration. This criterion may be expressed mathematically as;

$$R^2 = \left(\frac{F_o - F}{F_o} \right) \times 100 \qquad (9)$$

where F_o is the 'initial (*no-model* or *naïve-model*) variance', which is defined as the sum of squares of differences between the observed discharges and the corresponding naïve-model forecasts (these being the long-term mean discharge estimated from the calibration period). For a highly seasonal catchment, a more appropriate naïve-model forecast would be "*the seasonal forecast for a given date*", as obtained from the calibration period (Garrick *et al.*, 1978), whereas for the evaluation of a modified model structure under test, it would be the simulated flow of the original model structure. For the real-time forecasting scenario, it would simply be the observed discharge at the start of the forecast lead-time, which is a primitive form of autoregressive model lead-time forecast. Although three of the catchments considered in this study show evidence of seasonality, a characteristic specifically addressed by the LPM, for comparison purposes the standard Nash-Sutcliffe R^2 index (based on the calibration mean) is used for all eight catchments.

For the calibration period, R^2 is identical to the coefficient of determination of linear regression analysis. In contrast, it differs from the coefficient of determination when evaluated in the verification period.

This is because the F_0 of the verification period is calculated using the mean of the calibration period, rather than the mean of the verification period, which is the mean that would be used in computing the coefficient of determination. The mean of the calibration period is used in estimating R^2 because the criterion attempts to compare the performance of the substantive model with that of a naïve (primitive) model having the long-term mean value of the discharge record, as estimated from the calibration period, as its discharge forecast.

2. The Average Relative Error (ARE) of peaks over a threshold is used in the present study as a basic measure of the goodness of the model peak discharge forecasts, this being an especially important issue in the context of real-time flood forecasting. The (ARE) over the relevant period (i.e. either the calibration or verification period) is defined as

$$ARE = \frac{1}{N_p}\left(\sum_{i=1}^{Np}\frac{\left|\Delta Q_{p_i}\right|}{Q_{p_i}}\right)\times 100 = \frac{1}{N_p}\left(\sum_{i=1}^{Np}\frac{\left|\hat{Q}_{p_i} - Q_{p_i}\right|}{Q_{p_i}}\right)\times 100 \qquad (10)$$

where N_p is the number of the selected peaks over the chosen discharge threshold, $\hat{Q}_{p_i}$ and Q_{p_i} are i^{th} model estimated and observed discharge peaks over the threshold, respectively. In general, the lower the relative error, the better is the performance of the model. A value of ARE of zero indicates a perfect matching of the peaks while large values of ARE would be an indication of the failure of the model to reproduce the peaks over the selected threshold. The calculation of the ARE requires the *a priori* specification of the discharge threshold. In the present study, the discharge value which is exceeded 25% of the time, in the calibration period, is chosen subjectively as the discharge threshold value to be used in conjunction with the ARE index. The use of such a discharge threshold value enables the evaluation of the efficacy of the model in producing the relatively high peak discharges.

Table (2) shows the R^2 values of the four rainfall-runoff models and the three neural network combination methods for the eight test catchments. Examination of the table reveals that for catchments characterised by significant seasonality (i.e.

Brosna, Sunkosi-1 and Yanbian) the LPM, with its inbuilt seasonal component, outperforms the LVGFM. For the large catchments (namely, Bahie, Kelantan, Sunkosi-1 and Shiquan-1), the R^2 values of the SMAR model, the PDISC model and the LVGFM generally have the same order of magnitude in the calibration period. On the basis of the average R^2 values in the calibration period, considering all the eight test catchments, the seasonally-based black-box LPM model has the worst performance among the four rainfall-runoff models. However, in the verification period, the black-box LVGFM is on average the worst model. Likewise, on the basis of the average R^2 values for the eight test catchments, the SMAR conceptual model has the best performance in the calibration period while in the verification period the PDISC conceptual model has the best performance.

Further analysis of Table (2) reveals that, in the calibration period, all three neural network combination methods have higher R^2 values than those obtained by the best of the four rainfall-runoff models. However, in the case of the verification period, the results are quite variable. In four of the catchments, namely, Brosna, Chu, Shiquan-1 and Yanbian, all three combination methods perform better than the best individual model. In the case of the Nan catchment, two out of the three combination methods perform better than the best individual model. For the remaining three catchments, only one of the four individual models performs better than all three combination methods, i.e. the PDISC model for the Baihe and Kelantan catchments and the LPM in the case of the Sunkosi-1 catchment.

On the basis of the average R^2 values of the eight test catchments, for the verification period, all three combination methods have better performance than the best of the four individual rainfall-runoff models. Referring still to the verification period, the SNN, in terms of R^2, is superior to all four individual rainfall-runoff models in the case of the five catchments noted above (Brosna, Chu, Shiquan-1 and Yanbian). However, for the remaining three catchments, the SNN performs worse than the best individual model but still better than the other three individual models tested. Likewise, in the verification period, the RBFNN and the MLPNN have higher R^2 values than those of the four individual models in the case of only four catchments, namely, the Brosna, the Chu, the Shiquan-1 and the Yanbian catchments. Comparison of the results of the three neural network combination methods shows

that, in the calibration period, the performance of the MLPNN is consistently better than those of the other two combination methods. The performance of the RBFNN ranks second best in the case of four catchments while, for the remaining four catchments, it is the SNN that ranks second best. Generally, the differences in the R^2 values of the SNN and the RBFNN are not significant.

In the case of the verification period, the SNN combination method has higher R^2 values than the other two combination methods in six of the eight test catchments, the exceptions being the Kelantan and Baihe catchments which have R^2 values better than those of the MLPNN only. The performance of the MLPNN is better than that of the RBFNN in the case of half of the test catchments while the converse is true in the case of the other half.

Table (3) shows the Average Relative Error (*ARE*) of peaks over a threshold for the four individual models and the three neural network combination methods. In table 3, models with high ARE value and with high ARE inverse rank values have the worst results. In the case of the individual models, the LPM and the LVGFM are on average the worst two models in terms of reproducing the observed peaks, as they have the highest average ARE values in both the calibration and the verification periods. However, in the case of the Sunkosi-1 catchment, which is characterized by strong seasonality, the LPM has the third lowest ARE value in the calibration period and the lowest in the verification period. In the calibration period, the LVGFM has the highest or the second highest ARE values among the four individual models in the case of six catchments, namely, Brosna, Kelantan, Nan, Shiquan-1, Sunkosi-1 and Yanbian. With the exception of the Kelantan catchment, this also holds for the verification period. In the case of the Bahie and Chu catchments in calibration, the LVGFM has the second lowest ARE values among the four individual models. However, In the case of the Kelantan (the last of the three remaining catchments), the LVGFM has the highest ARE value in the calibration period and yet the lowest in the verification period.

Similarly, on the basis of the mean ARE values for the eight test catchments, the PDISC is best and SMAR is second best among the four individual models for both the calibration and verification periods. For the calibration period, the PDISC has the highest ARE value only in the case of the Sunkosi-1 catchment and only in the

case of the Kelantan catchment for the verification period. The SMAR model has the highest ARE value in the case of the Chu catchment only for the verification period.

In the calibration period, all three of the neural network combination methods have better performance, in terms of the ARE value, than the best of the individual models in six of the eight test catchments, i.e. excluding the Shiquan-1 and Sunkosi-1 catchments. However, in the verification period, all three of the three neural network combination methods have a better performance than the best individual model in the case of three catchments, namely, Baihe, Brosna, Nan. Two of the combination methods are better in the case of Chu, and none in the case of the other three catchments, namely, Shiquan-1, Sunkosi-1 and Ynabian-1. However, on the basis of the mean ARE values for the eight test catchments, all of the three combination methods have better performance than that of the best individual model for both periods. However, such improvements in performance by the combination methods may not be considered as being significant in real-time operational conditions.

Comparison of the ARE values of the three combination methods reveals that for the calibration period, the MLPNN has lowest ARE values for five catchments, namely, Brosna, Chu, Nan, Kelantan and Yanbian, the second lowest in the case of the Baihe and Sunkosi-1 catchments, being the lowest in the case of the Shiquan catchment. For the verification period, the MLPNN has lowest ARE values for four catchments, namely, Brosna, Chu, Kelantan and Yanbian, the second lowest in the case of the Shiquan-1 and Sunkosi-1 catchments, and the lowest in the case of the Baihe and Nan catchments.

Likewise, in the calibration period, the SNN has the lowest ARE value for Baihe, Shiquan-1 and Sunkosi-1 and the highest ARE values for Nan and Yanbian. However, in the verification period, among the combination methods, the SNN has the lowest ARE values for Nan, Shiquan-1 and Sunkosi-1 and the second lowest ARE values for the remaining catchments, apart from the Yanbian for which the SNN has the highest ARE value. In the calibration period, the RBFNN has the highest ARE values for Baihe, Brosna, Chu, Kelantan and Sunkosi-1 and the second highest ARE values among the combination methods for the remaining catchments. In the verification period, the ARE values of the RBFNN are either the highest or the second highest for all of the test catchments, except Bahie, where the RBFNN has the lowest

ARE value. Figure (3) shows the observed and the estimated flood peak hydrographs for some selected peaks in the Brosna, Shiquan-1, Nan and Chu catchments for the three combination methods and the best individual model.

5. Conclusions

In the present study, the performances of three neural network combination methods, which use three different neural network structures, are compared using the daily data of eight catchments. These three methods are based on the structures of the simple perceptron neural network (SNN), the radial basis function neural network (RBFNN) and the multi-layer perceptorn neural network (MLPNN). For each catchment, the three combination methods are used to combine the synchronous daily discharge simulations of four rainfall-runoff models, these simulations being used as inputs to the three neural networks. The four rainfall-runoff models are the Linearly Varying Gain Factor Model (LVGFM), the Linear Perturbation Model (LPM), the Probability-Distributed Interacting Storage Capacity (PDISC) Model and the Soil Moisture and Accounting Procedure (SMAR) Model.

The results of the individual models indicate that the SMAR and the PDISC have on average better simulation performance than the other two rainfall-runoff models. In the Sunkosi-1 catchment, with marked seasonality, the LPM has the best performance of the individual models. The LVGFM and the LPM are on average the worst models in terms of reproducing the peak hydrographs.

The results of the study showed that all of the three combination methods have, on average, better performance than that of the best individual model in terms of the R^2 values and the Average Relative Error (*ARE*) of peaks over a threshold. However, in terms of the ARE values, the MLPNN combination method has a better performance than that of the other two methods for seven of the eight catchments in the calibration period and for five catchments in the case of the verification period. The results also reveal that, in the calibration period, the performance of the MLPNN combination method is consistently better than those of the other two combination methods, in terms of the R^2 values, for all of the eight catchments. In general, the differences in the R^2 values of the SNN and the RBFNN combination methods are not significant for most of the eight test catchments. However, the differences in the R^2 values of these three methods are quite marginal in some cases. This would suggest that, in

these cases, the use of a complicated neural network for river flow combination is not justified.

In future applications of the NNM methods, consideration should be given to different network structures over a wider range of catchments to test whether similar conclusions to those obtained in this study can be drawn. When the NNMs are used in real-time forecasting systems, the addition of an updating component to the simulation mode results of the combination methods would significantly improve the reliability of the real-time river flow forecasts. However, as the lead-time increases the reliability of the real-time river flow forecasts would be more dependent on the reliability of the design mode simulated river flows rather than the versatility of the updating procedure.

Acknowledgements

The authors gratefully acknowledge the comments and constructive criticisms of the two unknown reviewers which helped them to improve the quality of the manuscript.

References

Ahsan, M. and O'Connor, K.M., 1994. A simple non-linear rainfall-runoff model with a variable gain factor. J. Hydrology, 155: 151-183.

Beven K.J. and Binley, A., 1992. The future of distributed models: model calibration and uncertainty prediction. Hydrol. Proc. 6, pp. 279-298. John Wiley, London.

Blum, A., 1992. Neural networks in C++; an object-oriented framework for building connectionist systems. John Wiley and Sons., Inc., USA.

Bruen, M. and , Yang, J., 2005. Functional networks in real-time flood forecasting — a novel application. Advances in Water Resources, 28, 899-909.

Butts, M.B., Payne,J.T., Kristensen M. and Madsen, H., 2004. An evaluation of the impact of model structure on hydrological modelling uncertainty for streamflow simulation. J. Hydrology, 298:242-266.

Coulibaly, P., Hache, M., Fortin, V. and Bobée, B., 2005. Improving Daily reservoir inflow forecasts with model combination. ASCE Journal of Hydrologic Engineering, 10(2): 91-99.

Dawson, C.W. and Wilby, R., 1999. A Comparison of Artificial Neural Networks used for River Flow Forecasting. Hydrology and Earth System Sciences, 3(4): 529-540.

Dawson, C.W. and Wilby, R.L., 2001. Hydrological modelling using artificial neural networks. Progress in Physical Geography, 25(1): 80-108.

Fausett, L., 1994. Fundamentals of neural networks: architectures, algorithms, and applications. Prentice-Hall International., Inc., USA.

Fenicia, F., Solomatine, D.P., Savenije H.H. and Matgen G.P., 2007. Soft combination of local models in a multi-objective framework. Hydrology and Earth System Sciences Discussions, 4, 91-123, SRef-ID: 1812-2116/hessd/2007-4-91.

Franchini, M., 1996. Use of a genetic algorithm combined with a local search method for the automatic calibration of conceptual rainfall-runoff models. Hydrol. Sci. J., 41: 21-39.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Kluwer Academic Publishers, Boston, MA.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, NJ.

Holland, J.H., 1975. Adaptation in natural and artificial systems. University Michigan Press, Ann Arbor.

Hsu, K., Gupta, H.V. and Sorooshian, S., 1995. Artificial neural network modelling of the rainfall-runoff process, Water Resour. Res., 31(10): 2517-1530.

Hu, T.S., Lam, K.C. and Ng, S.T., 2001. River flow time series prediction with a range-dependent neural network. Hydrological Sciences, 46(5): 729–745.

Kachroo, R.K., 1992. River Flow forecasting. Part 5. Applications of a conceptual model. J. Hydrology, 133: 141-178.

Karunanithi, N., Grenney, W.J., Whitley, D. and Bovee, K., 1994. Neural networks for river flow prediction. J. Computing in Civil Eng., 8(2): 201-220.

Khosla, R. and Dillon, T., 1997. Engineering Intelligent Hybrid Multi-Agent Systems. Kluwer Academic Publishers, Norwell, MA 02061, USA.

Kim, Y.-Oh, Jeong, D. and Ko, I. H., 2006. Combining rainfall-runoff model outputs for improving ensemble streamflow prediction. Journal of Hydrologic Engineering, 11(6): 578-588.

Luo, F. and Unbehauen, R., 1997. Applied Neural Networks For Signal Processing. Cambridge University Press.

Maier, H. R. and Dandy, G.C., 2000. Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. Environmental Modelling & Software, 15: 101-124.

Mason, J.C., Price, R.K. and Tem'Me, A., 1996. A neural network model of rainfall-runoff using radial basis functions. J. Hydraulic Research, 34: 537-548.

Masters, T., 1993. Practical neural networks recipes in C++. Academic Press, Inc., USA.

Moore, R.J., 1985. The probability-distributed principle and runoff production at point and basin scales. Hydrol. Sc. J., 30 (2), 273-297.

Nash, J.E. and Barsi, B.I., 1983. A hybrid model for flow forecasting on large catchments. J. Hydrol., 65: 125-137.

Nash, J.E. and Sutcliffe, J.V., 1970. River flow forecasting through conceptual models. Part 1. A discussion of principles. J. Hydrol., 10: 282-290.

O'Connor, K.M., Goswami, M. and Shamseldin, A.Y., 2002. Comparative performance evaluation of rainfall-runoff models, six of black-box type and one of conceptual type, from the Galway low Forecasting System (GFFS) package, applied on two Irish catchments. The XXVII Assembly of the European Geophysical Union (EGS), Nice, France, April 2002.

Oudin, L., Andréassian, V., Mathevet, T., Perrin, C. and Michel, C., 2006. Dynamic averaging of rainfall-runoff model simulations from complementary model parameterization. Water Resources Research, 42, W07410, doi: 10.1029/2005WR004636.

Perrin, C., Michel, C. and Andréassian, V., 2001. Does a large number of parameters enhance model performance ? Comparative assessment of common catchment model structures on 429 catchments. J. Hydrology, 242(3-4): 275-301.

Press, W.H., Flannaery, B.P., Teukolsky, S.A. and Vetterling, W.T., 1992. Numerical Recipes. Cambridge University press, New York, pp. 301-306.

See, L. and Abrahart, R. J., 2001 Multi-model data fusion for hydrological forecasting. Computers & Geosciences 27(8), 987–994.

See, L. and Openshaw, S., 2000. A hybrid multi-model approach to river level forecasting. Hydrological Sciences Journal, 45(4): 523-536.

Senbeta, D.A., Shamseldin, A.Y. and O'Connor, K.M., 1999. Modification of the probability distributed interacting storage capacity model, J. Hydrology 224(3-4): 149-168.

Shamseldin, A.Y., 2004. Hybrid neural network models. Chapter 4 in: Abrahart, R.J., Kneale, P.E. and See, L.M. (eds.) *Neural Networks for Hydrological Modelling*. Rotterdam: A.A. Balkema Publishers

Shamseldin, A.Y., Ahmed, E.N. and O'Connor, K.M., 2002. Comparison of different forms of the Multi-layer Feed-Forward Neural Network Method used for River Flow Forecast Combination. Journal of Hydrology and Earth System Sciences, 6(4): 671-684.

Shamseldin, A.Y. and O'Connor, K.M., 1999. A Real-time Combination Method for the Outputs of Different Rainfall-Runoff Models. Hydrological Science Journal, 44(6): 895-912.

Shamseldin, A.Y., O'Connor, K.M. and Liang, G.C., 1997. Methods for Combining the Outputs of Different Rainfall-Runoff Models. J. Hydrology, 197: 203-229.

Tan, B. Q. and O'Conner, K. M., 1996. Application of an Empirical Equation in the SMAR Conceptual Model. J. Hydrology, 185:275-295.

van Rooij, A.J.F., Jain, L.C. and Johnson, R.P., 1996. Neural Network Training Using Genetic Algorithms. Machine Perception & Artificial Intelligence, 26, World Scientific Publishing Co.

Wang, Q.J., 1991. The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. Water Resour. Res., 27(9): 2467-2471.

World Meteorological Organization (WMO), 1992. Simulated real-time intercomparison of hydrological models. Oper Hydrol. Rep. 38, WMO No. 779, Geneva.

Xiong, L., Shamseldin, A.Y. and 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. Hydrology, 245: 196-217.

Ye, W., Bates, B. C., Viney, N. R., Sivapalan, M. and Jakeman, A.J., 1997. Performance of conceptual rainfall-runoff models in low-yielding ephemeral catchments, Water Resour. Res., 33: 153–166.

Zhang, B. and Govindaraju, R.S., 2000. Prediction of watershed runoff using Bayesian concepts and modular neural networks. Water Resour. Res., 36(3): 753-762.

Table 1: Summary Description of the test catchments.

Catchments	Country	Area Km^2	Memory length (*day*)	Average* Rainfall (*mm/day*)	Average Evaporation (*mm/day*)	Average Discharge (*mm/day*)	Calibration Period (*years*)	Verification Period (*years*)	Data Starting date
Sunkosi-1	Nepal	18000	30	4.65	3.30	3.63	6	2	1 Jan. 1975
Baihe	China	61780	15	2.59	2.89	1.04	6	2	1 Jan. 1972
Brosna	Ireland	1207	30	2.20	1.31	0.98	8	2	1 Jan. 1969
Chu	Vietnam	2370	15	3.78	2.54	1.64	8	2	1 Jan. 1965
Kelantan	Malaysia	12867	20	6.58	4.84	3.50	6	2	1 Jan. 1975
Nan	Thailand	4609	20	3.89	3.33	1.82	6	3	1Apr. 1978
Shiquan-1	China	23805	15	2.55	2.68	1.08	6	2	1 Jan. 1973
Yanbian	China	2350	30	3.28	5.79	2.55	6	2	1 Jan. 1978

* note that these averages are calculated using the calibration data only.

Table 2: The R^2 efficiency results of the four Models and of the three Methods of Combining the various Model Outputs.

Calibration period

Model	Baihe		Brosna		Chu		Kelantan		Nan		Shiquan-1		Sunkosi-1		Yanbian		Average in test catchments	
	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2	R^2 rank
LPM	74.26	7	70.28	6	63.10	7	76.71	7	75.81	7	71.94	7	91.96	4	83.00	6	75.88	7
LVGFM	87.16	5	41.37	7	83.18	4	78.91	6	76.53	6	85.17	5	88.63	7	78.05	7	77.38	6
PDISC	88.07	4	85.70	5	79.91	5	88.92	5	65.13	5	85.61	4	90.72	5	87.84	4	83.99	5
SMAR	83.29	6	85.83	4	75.90	6	89.24	4	84.02	4	84.23	6	89.81	6	85.87	5	84.77	4
SNN	90.43	3	91.13	3	85.6	3	91.33	2	84.88	2	89.54	2	93.15	2	88.84	3	89.36	2
RBFNN	90.68	2	91.58	2	85.94	2	89.63	3	86.21	3	89.13	3	92.47	3	89.06	2	89.34	3
MLPNN	91.99	1	92.45	1	90.10	1	91.52	1	87.19	1	90.96	1	93.66	1	89.26	1	90.86	1

Verification Period

Model	Baihe		Brosna		Chu		Kelantan		Nan		Shiquan-1		Sunkosi-1		Yanbian		Average in test catchments	
	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2 (%)	R^2 rank	R^2	R^2 rank
LPM	73.17	4	77.53	6	70.28	6	37.89	7	75.73	7	57.63	6	90.49	1	79.21	6	70.24	6
LVGFM	61.45	7	48.06	7	75.34	4	43.84	6	69.56	6	73.62	5	83.59	7	76.38	7	66.48	7
PDISC	80.16	1	86.21	4	64.44	7	51.85	1	75.26	5	78.06	4	85.42	6	85.97	4	75.92	4
SMAR	72.79	6	85.39	5	71.82	5	50.55	5	83.70	3	71.24	7	85.19	5	83.93	5	75.58	5
SNN	75.25	3	89.25	2	77.70	1	50.64	4	84.67	1	82.23	1	88.49	2	87.06	2	79.41	1
RBFNN	80.02	2	88.73	3	77.22	2	50.80	2	83.79	2	80.01	3	87.86	4	86.78	3	79.40	2
MLPNN	73.01	5	91.11	1	75.04	3	50.78	3	79.70	4	80.85	2	88.19	3	88.24	1	78.35	3

Table 3: Average Relative Error of the four Models and of the three Methods of Combining the various Model Outputs.

Calibration period

Model	Baihe		Brosna		Chu		Kelantan		Nan		Shiquan-1		Sunkosi-1		Yanbian		Average in test catchments	
	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank
LPM	36.99	7	22.25	6	89.53	7	21.41	4	36.41	5	46.86	7	14.22	3	28.62	6	37.04	7
LVGFM	30.38	5	37.07	7	60.76	5	26.81	7	42.90	7	44.41	6	15.96	6	29.94	7	36.03	6
PDISC	30.08	4	19.89	5	44.82	4	21.83	6	41.74	6	30.98	1	18.61	7	23.12	4	28.88	4
SMAR	32.02	6	17.32	4	69.67	6	20.59	5	34.35	4	39.52	5	15.67	5	23.68	5	31.60	5
SNN	26.86	1	16.27	2	46.33	2	17.87	2	34.14	3	36.64	2	13.34	1	22.36	3	26.73	2
RBFNN	27.41	3	17.18	3	46.39	3	19.26	3	33.17	2	36.93	3	15.42	4	22.01	2	27.23	3
MLPNN	27.09	2	15.96	1	42.72	1	17.55	1	32.18	1	37.80	4	13.94	2	21.82	1	26.18	1

Verification Period

Model	Baihe		Brosna		Chu		Kelantan		Nan		Shiquan-1		Sunkosi-1		Yanbian		Average in test catchments	
	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank	ARE (%)	ARE Inv. rank
LPM	59.05	7	27.18	6	55.23	6	28.82	2	41.97	6	55.66	7	20.84	1	31.05	6	39.98	7
LVGFM	37.79	4	40.79	7	46.10	5	24.97	1	45.19	7	52.32	6	28.98	6	39.77	7	39.49	6
PDISC	38.95	5	22.59	5	41.98	3	34.41	7	37.63	5	35.56	1	32.58	7	22.14	1	33.23	4
SMAR	39.69	6	21.73	4	56.28	7	30.52	6	35.68	4	40.96	5	25.83	4	23.38	4	34.26	5
SNN	26.93	2	20.80	2	40.88	2	29.34	4	30.77	1	37.91	2	23.74	2	23.91	5	29.29	1
RBFNN	26.45	1	21.42	3	43.19	4	30.06	5	31.90	2	40.56	4	26.91	5	22.87	3	30.42	2
MLPNN	33.51	3	20.09	1	40.60	1	29.04	3	33.27	3	40.49	3	25.40	3	22.51	2	30.67	3

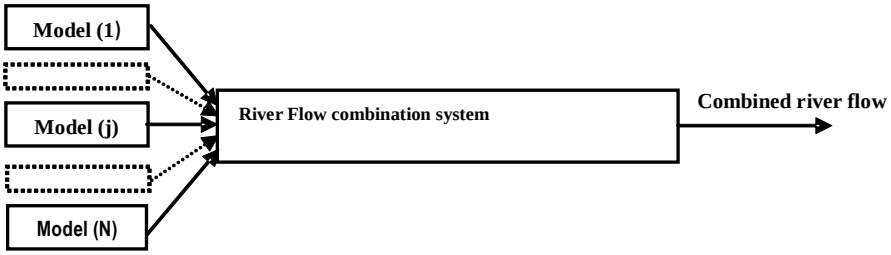


Figure 1: River flow combination system (adapted from Shamseldin, 2004)

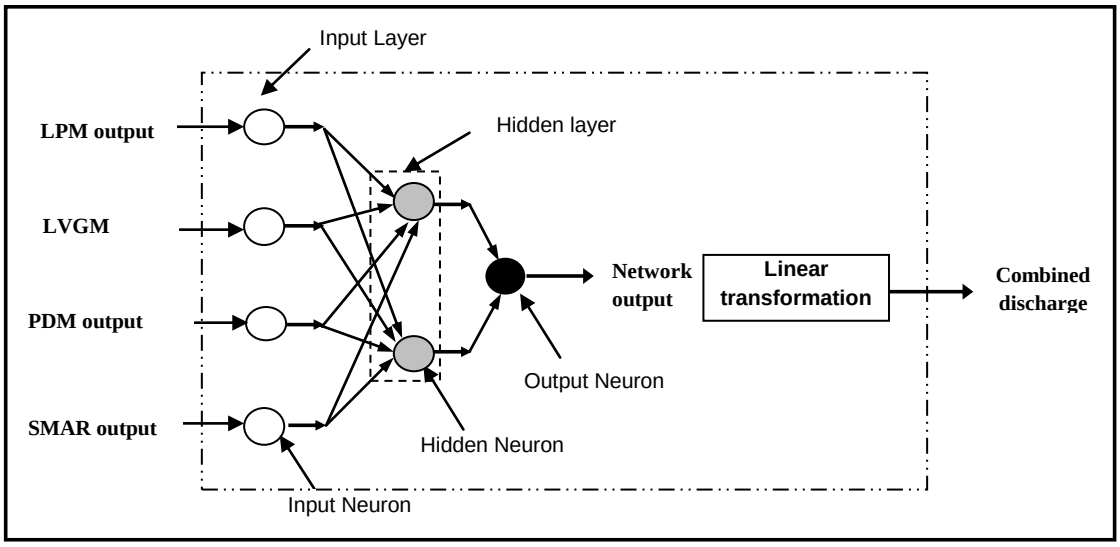
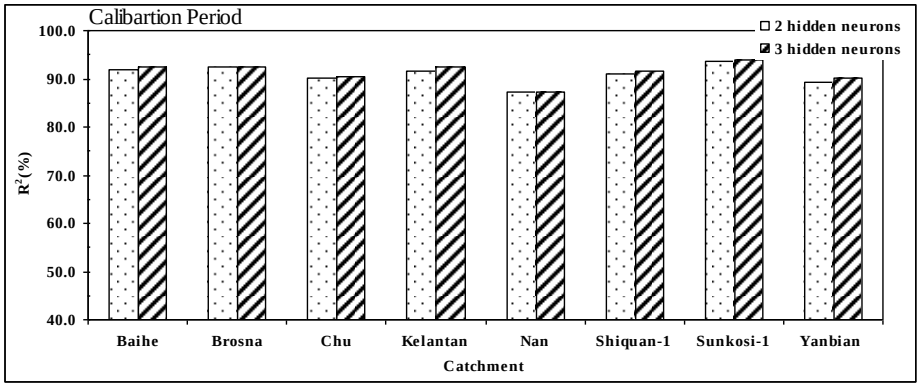


Figure 2: Neural Network Multi-Layer Perceptron Structure (after Shamseldin *et al.*, 1997).



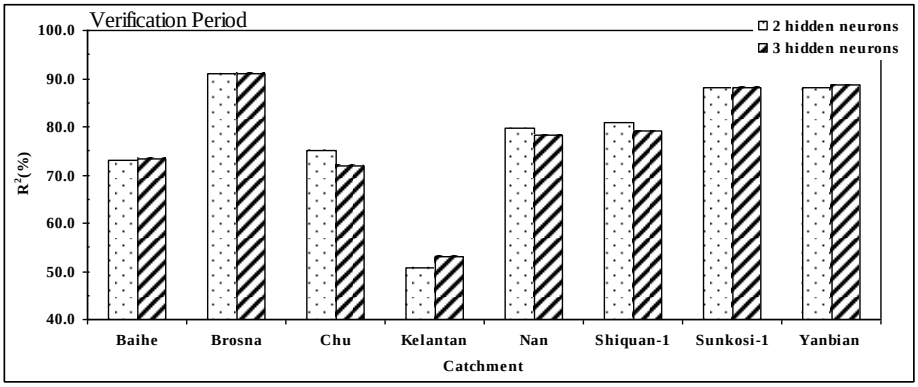
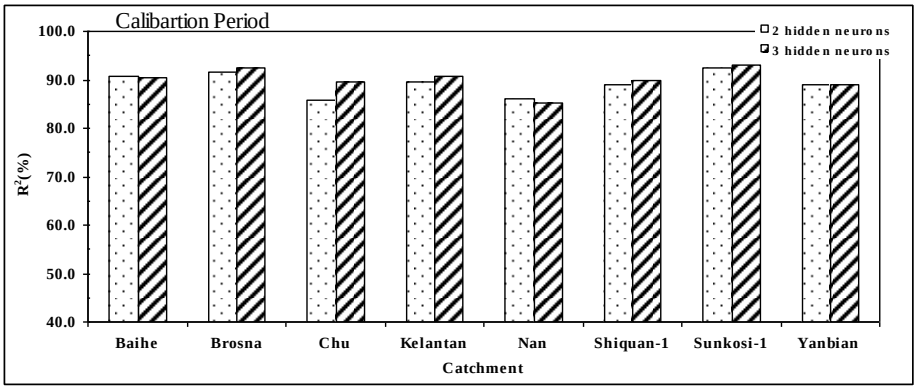


Figure 3 Comparison of the R^2 values of the MLPNN combination method for different numbers of hidden neurons.



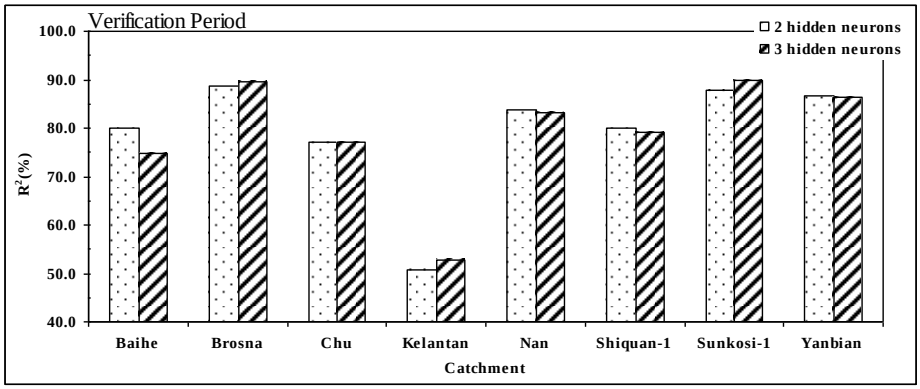
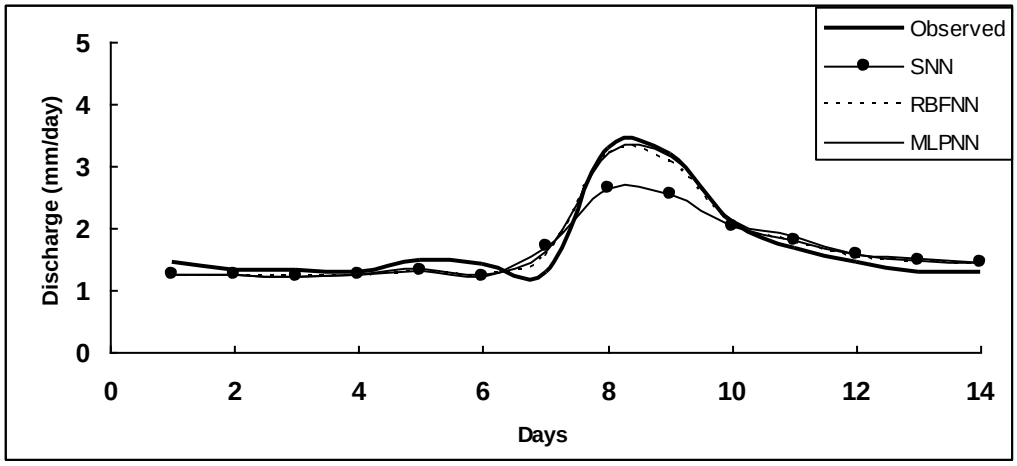


Figure 4: Comparison of the R^2 values of the RBFNN combination method for different numbers of hidden neurons.

Catchmnet: Brosna



Catchment: Shiquan-1

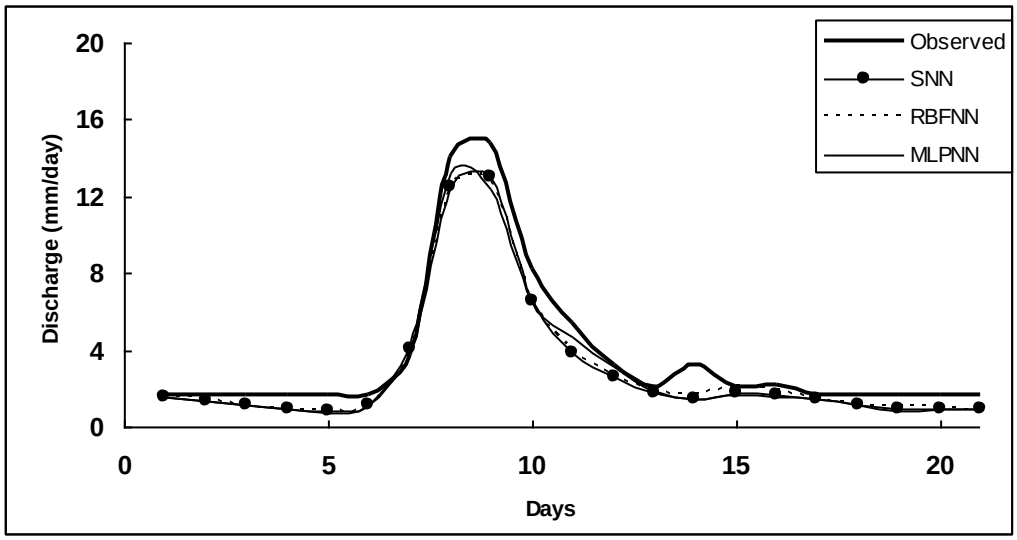
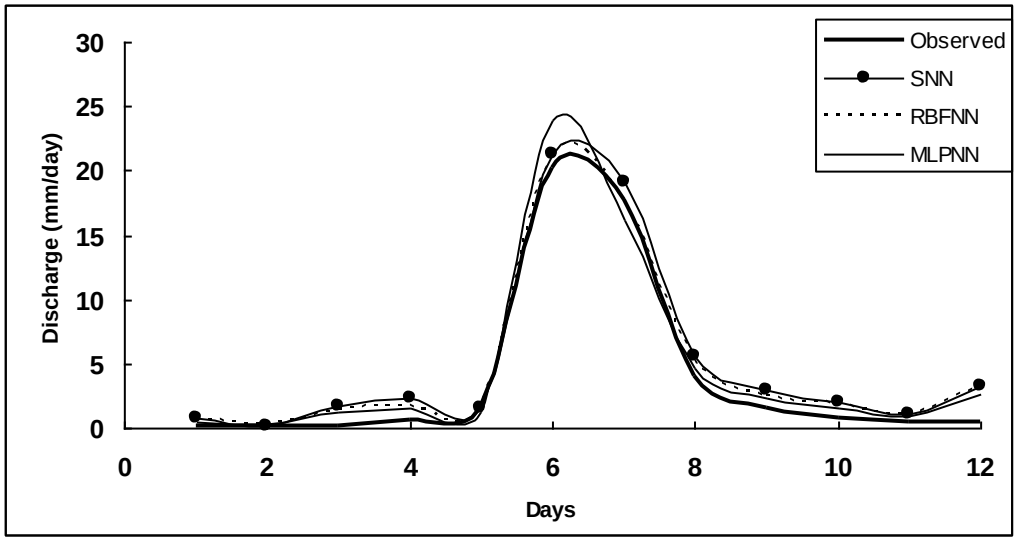


Figure (5): Comparison of the observed and the estimated flood hydrographs of the combination method.

Catchment: Chu



Catchment: Nan

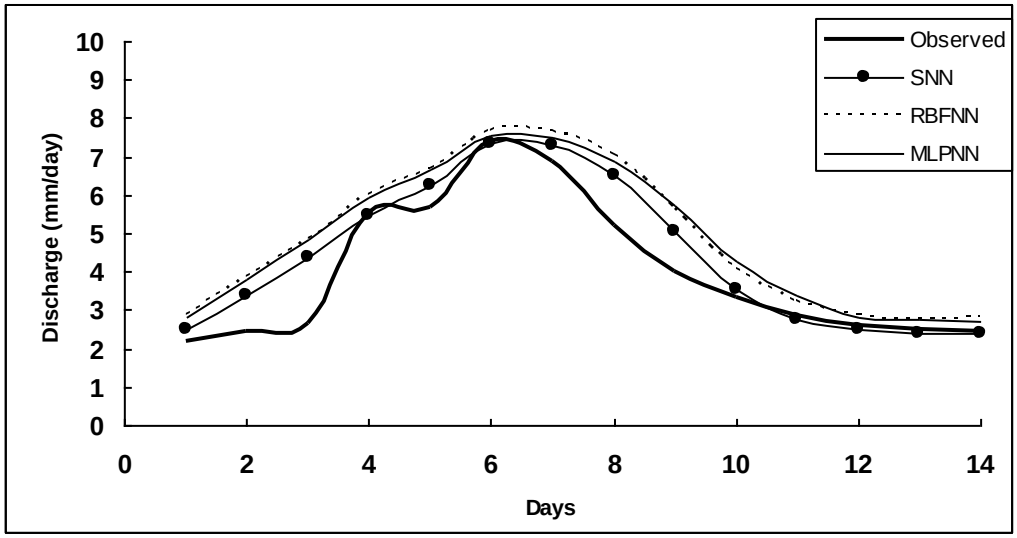


Figure (5): contd.