



A simplified approach to quantifying predictive and parametric uncertainty in artificial neural network hydrologic models

R. K. Srivastav,¹ K. P. Sudheer,¹ and I. Chaubey^{2,3}

Received 18 July 2006; revised 24 May 2007; accepted 26 June 2007; published 5 October 2007.

[1] One of the principal sources of uncertainty in hydrological models is the absence of understanding of the complex physical processes of the hydrological cycle within the system. This leads to uncertainty in input selection and consequently its associated parameters, and hence evaluation of uncertainty in a model becomes important. While there has been considerable interest in developing methods for uncertainty analysis of artificial neural network (ANN) models, most of the methods are relatively complex and/or require assumption about the prior distribution of the uncertain parameters. This paper presents an effective and simple way to perform uncertainty analysis for ANN-based hydrologic model. The method is based on the concept of bootstrap technique and is demonstrated through a case study of the Kolar River basin located in India. The method effectively quantifies uncertainty in the model output and the parameters arising from variation in input data used for calibration. In the current study, the uncertainty due to model architecture and the input vector are not directly considered; they have been minimized during the model calibration. The results from the case study suggest that the sampling variability of the training patterns as well as the initial guess of the parameters of ANN do not have significant impact on the model performance. However, despite good generalization properties for the models developed in this study, most of them fail to capture the hydrograph peak flow characteristics. The proposed method of uncertainty analysis is very efficient, can be easily applied to an ANN-based hydrologic model, and clearly illustrates the strong and weak points of the ANN model developed.

Citation: Srivastav, R. K., K. P. Sudheer, and I. Chaubey (2007), A simplified approach to quantifying predictive and parametric uncertainty in artificial neural network hydrologic models, *Water Resour. Res.*, 43, W10407, doi:10.1029/2006WR005352.

1. Introduction

[2] Hydrologic simulation or modeling is a powerful technique of hydrologic system investigation for the researchers and the engineers involved in the planning and development of integrated approach for water resources management. To date, a wide variety of rainfall-runoff models have been developed and applied for water resources planning. Most of these models are based either on physical considerations or on a system theoretic approach. In the physical approach, the primary motivation is the study of physical phenomena and their understanding, while in the system theoretic approach the concern is with the system operation, not the nature of the system by itself or the physical laws governing its operation.

[3] Most hydrological models (both physical as well as system theoretic) are complex and stochastic in nature and contain certain degree of uncertainty. In the case of physically based or conceptual models the complexity increases as the number of model parameters increase. Further, the understanding of the physics of the processes becomes more predominant in such models. Whereas system theoretic models do not consider the physical characteristic of the parameters, but they map the data from input to output using transfer functions. The uncertainty in hydrological models arises due to the absence of understanding of the complex physical processes of the hydrologic cycle within the system. This lack of understanding leads to uncertainty in selection of the model inputs and consequently its associated parameters. Hydrologists have been inclined to simplify these complex natural processes into simpler way of modeling by assuming the model inputs and the processes to be deterministic. As a consequence most of these models have been applied in a deterministic way [Christiaens and Feyen, 2002] assuming that the input variables and the parameters (after calibration) represent the reality in an accurate way.

[4] The development and application of system theoretic models to hydrological processes have increased in the recent years. The reason for such an increasing interest resides in their intrinsic generality, flexibility, and global

¹Department of Civil Engineering, Indian Institute of Technology Madras, Chennai, India.

²Department of Agricultural and Biological Engineering, Purdue University, West Lafayette, Indiana, USA.

³Also at Department of Earth and Atmospheric Sciences, Purdue University, West Lafayette, Indiana, USA.

performance in most applications where other models either tend to fail or become cumbersome. One of the powerful techniques that have gained momentum in the last few decades is the artificial neural network (ANN), as it has been successfully applied to a wide range of problems in hydrology [Hsu *et al.*, 1995; Sajikumar and Thandaveswara, 1999; Sudheer *et al.*, 2002, 2003]. Generally, in almost all applications, an ANN model is tested for its generalization properties by means of statistical evaluation measures, and no quantification of its predictive uncertainty is reported. The quantification of the uncertainty associated to the results provided by ANN models is essential for their confident and reliable use in practice.

[5] The primary sources of uncertainty are input data, the model parameters, and the structure, in addition to the measured data used during calibration. The uncertainty evaluation provides the degree of behavior of each set of watershed parameters, which in turn are translated into confidence interval estimates on the output of the model [Wagener, 2003]. There are various methods available for quantifying the uncertainty in physical hydrologic models [Christiaens and Feyen, 2002; Beven and Binley, 1992], while little discussion is found in hydrologic literature regarding the uncertainty analysis of the ANN hydrologic models except a few [Kingston *et al.*, 2005; Khan and Coulibaly, 2006; Han *et al.*, 2007].

[6] Three main approaches exist for the estimation of accuracy of ANNs: the delta method, based on a Taylor expansion of the regression function $f(x; \hat{w})$ (where x is the vector of input variable and \hat{w} is the weight parameters) upon which process estimation is based [Rivals and Personnaz, 1998; Dybowski and Roberts, 2000]; the bootstrap method, based on a resampling technique [Efron, 1979; Efron and Tibshirani, 1993]; and the Bayesian approach, based on Bayesian statistics to express the uncertainty of the network weights in terms of the probability distributions and integrate them to obtain the probability distribution of the target conditional on the observed training set [Dybowski and Roberts, 2000; Bishop, 1995]. A few applications in hydrology to quantify the uncertainty associated with ANN models have focused on the Bayesian approach [Kingston *et al.*, 2005; Khan and Coulibaly, 2006; Han *et al.*, 2007], which is computationally expensive since it requires Monte Carlo solutions of the integrals. The bootstrap method is the simplest approach since it does not require the complex computations of derivatives and Hessian-matrix involved in the delta method or Monte Carlo solutions involved in the Bayesian approach. Abraham [2003] employed bootstrap technique to continuously sample the input space in the context of rainfall-runoff modeling and reported that it offered marginal improvement in terms of greater accuracies and better global generalizations. He suggested further research involving bootstrap technique for estimating confidence interval of the outputs. To the best of the authors' knowledge, no application has been reported that used bootstrap method to quantify confidence interval of outputs from ANN hydrologic models. The major focus of this paper is to illustrate the use of bootstrap method to quantify uncertainties associated with ANN hydrologic models. The presented method can be employed to quantify the uncertainties in parameters and predictions arising from the choice of data used for the

model calibration, while other sources of uncertainty are assumed to be minimized (through trial and error procedure) during the calibration of the model.

2. Background

[7] The most common method of uncertainty analysis in traditional hydrologic models is to select a set of models (structures and parameters) and evaluate the performance of each of them in terms of some statistical measure, and assign a "degree of believability" to each model based on the performance of the model. This degree of believability is translated into uncertainty estimates on the model output. The approaches differ in the suit of assumptions underlying each technique based on which methods are used to compute the relative degree of believability [Beven and Freer, 2001; Wagener, 2003]. Different methods have evolved in uncertainty analysis in past few decades (e.g., error propagation method [Rogers *et al.*, 1985]; mean value first-order second-moment [Janssen *et al.*, 1992]; advanced first-order second-moment [Morgan and Henrion, 1990]; among many others). These methods have an advantage of having large application area. These methods, however, rely mostly on linear local approximation of the model, their results are biased by the selection of base point, and they may cause possible high costs in model development, implementation, and calculation. Most of these methods are based on a Taylor expansion of the regression function upon which the estimation is based, and require complex computations of derivatives and Hessian-matrix inversion. Hence application of these techniques to ANN uncertainty estimation becomes computationally expensive.

[8] An alternative method is based on Monte Carlo simulation which is cost-effective, suitable for models with many parameters, large application area, simultaneous sampling of parameters, direct estimation of distributions of outputs, and simple to use and implement. Beven and Binley [1992] proposed the Generalized Likelihood Uncertainty Estimation (GLUE), which is a Bayesian Monte Carlo simulation based technique, for uncertainty analysis of hydrological models. It should be noted that the GLUE procedure requires that the sampling ranges be specified for each parameter to be considered. The sampling of the parameter space is generally done by Monte Carlo simulations using uniform random sampling across the specified parameter range. Furthermore, GLUE requires a formal definition of likelihood measure to be used and the criteria for acceptance or rejection of the models, which is a subjective choice [Freer and Beven, 1996]. Consequently the limitations of Monte Carlo simulations, which include computational inefficiency and spurious correlation among parameters due to sampling procedure used in parameter selection, are also associated with the GLUE. Monte Carlo simulation based methods for uncertainty analysis can be applied to ANN, only if the distribution of the weights is known a priori as well as the parameter sampling procedure takes care of the correlation between the weights. When the distribution of weights is not known a priori, prior distribution must be assumed. Recently, Kingston *et al.* [2005] employed Bayesian training approach for accounting the uncertainty in ANN parameters during training. In this approach the distribution of the parameters has been assumed a priori, which is updated to posterior distribution

using likelihood following Baye's theorem while new data are presented. A similar approach is employed by *Khan and Coulibaly* [2006]. While the Bayesian training approach has certain advantages like elimination of over-fitting problems, the success of the approach mainly depends on the selection of suitable prior and noise models and the computation methods employed for integrating the posterior distribution of parameters [*Khan and Coulibaly*, 2006]. Note that poorly designed prior probability distributions can result in serious distortions of results [*Adkison and Pterman*, 1996].

[9] *Shrestha and Solomatine* [2006] proposed a method to compute the predictive uncertainty of a model. The method employs fuzzy *c*-means clustering to partition the input space into different zones having similar model errors; constructs the prediction interval (PI) for each cluster on the basis of empirical distributions of the errors corresponding to all input patterns in each cluster; and translates the PIs of clusters to examples by means of fuzzy computing. While this method has the advantage of not requiring any assumption about the prior distribution of the parameters, it totally ignores the uncertainty associated with the parameter estimation. This is a severe limitation for this method, since ANN has the weakness of "equifinality" problem. It should be noted that the constructed PIs will be biased by the choice of clusters in the input space. *Han et al.* [2007] proposed a method to understand the uncertainty in ANN hydrologic models with the heuristic that the distance between the input vector at prediction and all the training data provide a valuable indication on how well the prediction would be. However, their method did not quantify the uncertainty of the model parameters or the predictions.

[10] Hence a method that works under joint stochastic-deterministic modeling framework, and deals with the uncertainty associated with the model input as well as the parameters that results in an uncertainty band around the deterministic simulations, is to be considered in evaluating the uncertainty associated with ANN. When sufficiently large sets of examples (training patterns) are available, the sampling variability in weights can be approximated by bootstraps [*Stone*, 1974]. The bootstrap is a computational procedure that uses intensive resampling with replacement, in order to reduce uncertainty [*Efron and Tibshirani*, 1993]. In addition, it is the simplest approach since it does not require the complex computations of derivatives and Hessian-matrix inversion involved in linear methods or the Monte Carlo solutions of the integrals involved in the Bayesian approach [*Dybowski and Roberts*, 2000].

[11] It is envisaged that the resampling mimics the random component of the process, and that the variance can be reduced through averaging over numerous different partitions of the data. The bootstrap mechanism is often used to process hundreds or thousands of subsets, such that an empirical estimate of a specified output distribution is produced, and from which certain fundamental characteristics of the population can be calculated, e.g., mean, variances, or cumulants [*Abrahart*, 2003]. Bootstrap technique can also be used to produce statements about probabilities, to generate inferences about true parameters, or to determine confidence intervals. Accordingly, the

current study employed bootstrap technique for uncertainty analysis of ANN models.

3. Methodology

[12] Consider an ANN to be trained for performing the task of nonlinear regression (input-output mapping), i.e., estimating the underlying nonlinear relationship existing between a vector of input variables \mathbf{x} and an output target y (assumed monodimensional for simplicity of illustration) based on a finite set of input/output data examples (patterns), $D \equiv \{(x_n, y_n), n = 1, 2, \dots, n_p\}$. It can be assumed that the target y is related to the input vector \mathbf{x} by an unknown nonlinear deterministic function $\mu_y(x)$ corrupted by a Gaussian white noise $\varepsilon(x)$, such that

$$y = \mu_y(x) + \varepsilon(x); \quad \varepsilon(x) \sim N(0, \sigma_\varepsilon^2(x)). \quad (1)$$

[13] The objective of the regression task is to estimate $\mu_y(x)$ by means of a regression function $f(x; \hat{w})$, dependent on the network parameters \hat{w} to be properly determined on the basis of available data set D . The parameters \hat{w} , called weights, are usually determined by a training procedure which aims at minimizing the quadratic error function,

$$E = \frac{1}{2n_p} \sum_{n=1}^{n_p} (\hat{y}_n - y_n)^2, \quad (2)$$

where $\hat{y}_n = f(x_n; \hat{w})$ is the network output corresponding to input \mathbf{x}_n . If the network architecture and training parameters are suitably chosen and the minimization done to determine the weights values is successful, the resulting function $f(x; \hat{w})$ gives a good estimate of the true but unknown function $\mu_y(x)$. Indeed, it is possible to show that in the ideal case of an infinite training data set and perfect minimization algorithm, a neural network trained to minimize the error function in equation (2) provides a function $f(\cdot)$ which performs a mapping from the input \mathbf{x} into the expected value of target y , i.e., the true deterministic function $E[y|x] = \mu_y(x)$ [*Bishop*, 1995]. In other words, the network averages over the noise on the data discover the underlying deterministic generator. Unfortunately, all the training sets are finite and there is no guarantee that the selected minimization algorithm can achieve the global minimum.

[14] In practical regression problems, there are actually two types of predictions that one may want to obtain in correspondence to a given input \mathbf{x} : an estimate of $f(x; \hat{w})$ of the underlying deterministic function $\mu_y(x)$ and an estimate of the target value y itself as given by equation (1). To these estimates it is crucial to associate their corresponding measure of confidence. This requires that the various sources of uncertainty affecting the determination of weights \hat{w} be properly accounted [*Dybowski and Roberts*, 2000]. It must be considered that from a probabilistic point of view, the data set $D \equiv \{(x_n, y_n), n = 1, 2, \dots, n_p\}$ used for training the network is only one of an infinite number of possible data sets which may be drawn within the given input volume V_x and from the underlying statistical error distribution. In other words, this variability in the training set is due to the variability in the sampling of input vectors $\mathbf{x}_n, n = 1, 2, \dots, n_p$ and in the random variability of the corresponding target output y_n . Each possible training set

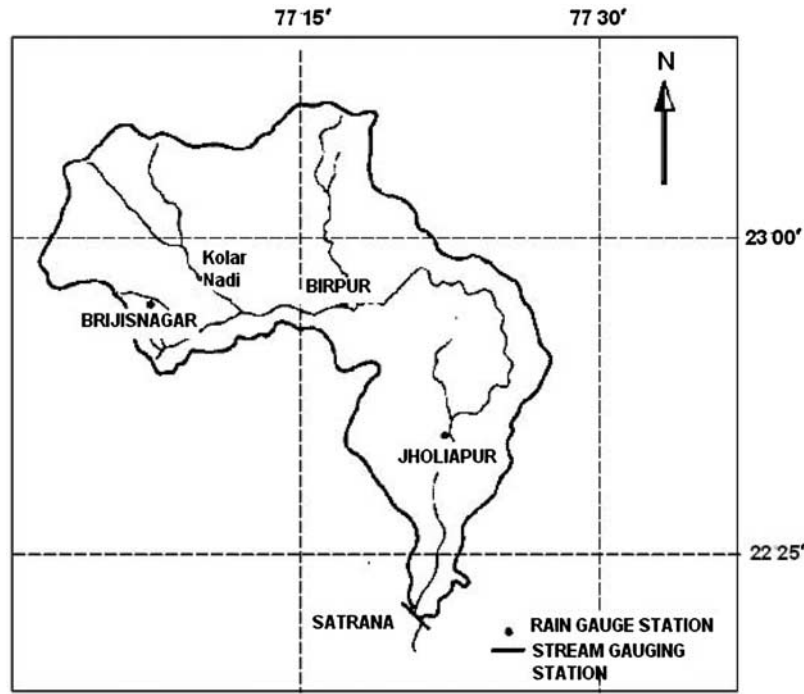


Figure 1. Map of Kolar basin.

can give rise to a different set of network weights \hat{w} . Correspondingly there is a distribution of regression functions $f(x; \hat{w})$ with variance (with respect to the training set D) $E\{[f(x, \hat{w}) - E(f(x, \hat{w}))]^2\}$ in which $E(f(x, \hat{w}))$ is the expected value which is the ensemble average.

[15] Another source of uncertainty in the estimate of $\mu_y(x)$ comes from an inappropriate choice of the network architecture. Indeed, in case of a network with too few nodes (too few parameters), a large bias occurs since the regression function $f(x; \hat{w})$ has insufficient flexibility to model the data adequately, which results in poor generalization properties for the network. On the other hand, excessively increasing the flexibility of the model by introducing too many parameters, e.g., by adding nodes, increases the error variance because the network regression function tend to over-fit the training data. A trade-off is typically achieved by controlling the model complexity by early stopping of the training so as to achieve a good fit of the training data with a reasonably smooth regression function which is not an over-fit to the data [Bishop, 1995]. The quantification of the accuracy of the estimate $f(x; \hat{w})$ of the true deterministic function $\mu_y(x)$ in terms of confidence interval mainly depends on the estimation of its variance:

$$\sigma_f^2(x) = E\{[f(x, \hat{w}) - E(f(x, \hat{w}))]^2\}. \quad (3)$$

[16] The current study employed bootstrap techniques for quantifying the variance. Bootstrap sampling requires that B bootstrap samples be drawn at random with replacement from the original training set of n_p input/output patterns $D \equiv \{x, y\}$. The generic b th sample D_b is constituted by the same number n_p of input/output patterns drawn among those in D although, due to the sampling with replacement, some of the patterns in D will appear more than once in D_b , whereas

some will not appear at all. Each bootstrap set D_b is then used as a data set of training a different neural network to give regression function $\hat{y}_b(x) = f(x; \hat{w}_b)$, where \hat{w}_b is thereby obtained network weight values. Then, in correspondence of a new input x , the bootstrap estimate \hat{y}_{boot} is given by the average of the B regression functions,

$$\hat{y}_{boot}(x) = f(x; \{\hat{w}_b, b = 1, 2, \dots, B\}) = \frac{\sum_{b=1}^B f(x; \hat{w}_b)}{B}, \quad (4)$$

and the bootstrap estimate $\hat{\sigma}_{boot}^2(x)$ of the variance $\sigma_f^2(x)$ is given by

$$\hat{\sigma}_{boot}^2(x) = \frac{\sum_{b=1}^B [\hat{y}_b(x) - \hat{y}_{boot}(x)]^2}{B - 1}. \quad (5)$$

[17] It is worth mentioning that when resorting to bootstrap sampling, particularly on a small data set D , it is important that all the B networks be well trained, in order to avoid that some biased network components of the bootstrap ensemble significantly affect the mean and variance of the estimate.

[18] In the bootstrap method of uncertainty analysis, the total example set is divided into two sets: training sets and validation sets. The validation set is kept aside, and random bootstrapping with replacement is performed on the training set in order to evaluate the variation in performance with varying training sets. The ANN model is trained on the bootstrapped training set (BTS) with fixed initial weights; the remaining patterns in the training set apart from BTS are employed for split sample validation so as to avoid any over-fitting. Note that in order to perform the split sample validation, it is required to keep track of the patterns being picked during bootstrapping. After one such model is

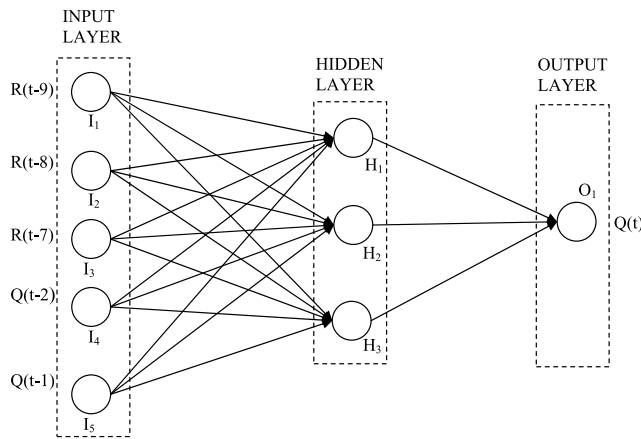


Figure 2. Artificial neural network (ANN) model structure.

trained, another training set is drawn from the pool using bootstrapping, and the network is trained on the new BTS using the same initial weights used earlier. A sufficiently large number of networks are trained using this procedure (300 in the current study). All the networks so developed are evaluated on the validation set kept aside by computing various performance indices.

[19] The variation in the weights of the network and the output of the network over the whole trained network is a measure of the uncertainty in the model parameters and predictions, respectively, that are coming from the variation in the training data set. Given a test input vector (\mathbf{x}), a prespecified prediction interval for output (\mathbf{y}) is an interval $[L, U]$, such that $P(L \leq \mathbf{y} \leq U) = C$, where C is typically 0.95 or 0.99, and the probability is computed over repeated random selection of the calibration set and repeated observations of \mathbf{y} , given the test input \mathbf{x} .

4. Case Example

4.1. Study Area and Data

[20] In order to demonstrate the proposed method of uncertainty analysis of ANN models, a case study on an Indian River basin is presented herein. For the bulk of the study, hourly data of rainfall and runoff from Kolar basin (Figure 1) in India is used. An ANN model for forecasting the river flow at 1-hour lead time has been developed in this study and analyzed for uncertainty. Data are collected during monsoon season (July, August, and September) for 3 years (1987–1989). Note that areal average values of rainfall data for three rain gauge stations have been used in the study. The Kolar River is a tributary of the river Narmada that drains an area about 1350 km² before its confluence with Narmada near Neelkant. In the present study the catchment area up to the Satrana gauging site is considered, which constitutes an area of 903.87 km². The 75.3-km-long river course lies between north latitude 21°09′–23°17′ and east longitude 77°01′–77°29′. More details on the basin are given by *Nayak et al.* [2005].

4.2. Modeling Exercise

[21] One of the most important steps in the ANN hydrologic model development process is the determination of significant input variables [Bowden et al., 2004a, 2004b].

Generally some degree of a priori knowledge is used to specify the initial set of candidate inputs [e.g., Campolo et al., 1999; Thirumalaiah and Deo, 2000]. However, the relationship between the variables is not clearly known a priori, and hence often an analytical technique, such as cross correlation, is used [e.g., Sajikumar and Thandaveswara, 1999; Luk et al., 2000; Silverman and Dracup, 2000; Sudheer et al., 2002]. The major disadvantage associated with using cross correlation is that it is only able to detect linear dependence between two variables, while the modeled relationship may be highly nonlinear. Nonetheless, the cross-correlation methods represent the most popular analytical techniques for selecting appropriate inputs [Bowden et al., 2004a]. The current study used a statistical approach suggested by Sudheer et al. [2002] to identify the appropriate input vector. The method is based on the heuristic that the potential influencing variables corresponding to different time lags can be identified through statistical analysis of the data series that uses cross correlations, autocorrelations, and partial autocorrelations between the variables in question.

[22] In order to ensure good generalization ability by an ANN model, a number of empirical relationships between the number of training samples and the number of connection weights have been suggested in the literature [Maier and Dandy, 2000]. However, network geometry is generally highly problem dependent and these guidelines do not ensure optimal network geometry, where optimality is defined as the smallest network that adequately captures the relationships in the calibration data (principle of parsimony). In addition, there is quite a high variability in the number of hidden nodes suggested by various rules. While research is being conducted in this direction by scientists working with ANNs, it may be noted that traditionally optimal network geometries have been found by trial and error [Maier and Dandy, 2000]. Consequently in the current application the number of hidden neurons in the network, which is responsible for capturing the dynamic and complex relationship between various input and output variables, was identified by various trials.

[23] The trial and error procedure started with two hidden neurons initially, and the number of hidden neurons was increased up to 10 with a step size of 1 in each trial. For each set of hidden neurons, the network was trained in batch mode (offline learning) to minimize the mean-square error at the output layer. In order to check any over-fitting during training, a split sample validation was performed by keeping track of the efficiency of the fitted model. The procedure of split sample validation is to check the performance of the model on the split-sample validation set at every iteration of the training, and suggest an early stopping when there is no

Table 1. Summary of Statistics of the Performance Measures for 300 Models

| Performance Measures | Statistical Measures | | | |
|----------------------------|----------------------|--------|--------------------|-------------------------|
| | Mean | Median | Standard Deviation | Coefficient of Skewness |
| Root-mean-square error | 58.222 | 58.258 | 8.838 | −1.073 |
| Coefficient of efficiency | 0.902 | 0.905 | 0.027 | 0.232 |
| Mean bias error | −1.880 | −1.93 | 0.600 | 0.731 |
| Coefficient of correlation | 0.957 | 0.961 | 0.013 | −0.281 |

Table 2. Summary Statistics of the Weights and Biases for 300 Models

| Weight | Mean | SD | Minimum | Maximum |
|-------------------------------|----------|---------|----------|---------|
| W ₁ H ₁ | 2.2606 | 6.8558 | 0.0400 | 51.8120 |
| W ₁ H ₂ | -1.4271 | 1.2574 | -7.9018 | 0.5149 |
| W ₁ H ₃ | 0.1696 | 0.2485 | -0.3696 | 1.4233 |
| W ₂ H ₁ | 0.1581 | 0.5680 | -2.5079 | 7.9536 |
| W ₂ H ₂ | -0.6799 | 0.5564 | -2.5378 | 0.5543 |
| W ₂ H ₃ | 0.0775 | 0.2187 | -1.1424 | 1.3929 |
| W ₃ H ₁ | 0.5042 | 1.6997 | -0.0382 | 23.1163 |
| W ₃ H ₂ | -1.2952 | 0.7605 | -2.8182 | 0.5544 |
| W ₃ H ₃ | 0.0714 | 0.2243 | -0.3753 | 1.4606 |
| W ₄ H ₁ | -2.6998 | 9.6876 | -74.4496 | 19.3496 |
| W ₄ H ₂ | 31.8005 | 16.1115 | -2.2918 | 59.9072 |
| W ₄ H ₃ | 2.9275 | 4.0943 | -1.2747 | 41.9548 |
| W ₅ H ₁ | 5.1304 | 8.3442 | -8.7833 | 73.1378 |
| W ₅ H ₂ | -26.9738 | 13.2200 | -50.6115 | 3.8400 |
| W ₅ H ₃ | 2.1293 | 4.4042 | -30.9881 | 5.0820 |
| B ₁ | -6.9587 | 6.2993 | -49.5073 | -4.2341 |
| B ₂ | -2.0137 | 3.2215 | -10.6204 | 3.1343 |
| B ₃ | -0.3610 | 2.4702 | -14.6202 | 1.8279 |
| W _H O ₁ | 23.2133 | 8.5931 | 0.7727 | 35.5384 |
| W _H O ₂ | 3.3487 | 7.6715 | 0.5128 | 52.7662 |
| W _H O ₃ | 8.2925 | 6.1327 | 1.1218 | 33.6254 |
| B ₄ | -9.6577 | 6.9008 | -48.3798 | -3.0453 |

significant improvement in the model performance. Note that while training the network using split sample validation, care has been taken to avoid any premature stopping of the training. The parsimonious structure that resulted in minimum error and maximum efficiency during training as well as split sample validation was selected as the final form of the ANN model.

[24] A sigmoid function is used as the activation function in the hidden layer and a linear transfer function at the output layer. As the sigmoid function has been used in the model, the input-output variables have been scaled appropriately to fall within the function limits using the range of the data. A standard back propagation algorithm [Rumelhart *et al.*, 1986] has been employed to estimate the network parameters. Adaptive learning and momentum rates [Nayak *et al.*, 2005] have been employed for the model training.

[25] On the basis of the methodology suggested by Sudheer *et al.* [2002], the following inputs have been identified for the ANN model: $R(t-9)$, $R(t-8)$, $R(t-7)$, $Q(t-2)$, $Q(t-1)$, where $R(t)$ represents the rainfall and $Q(t)$ represents the runoff at any time period t . The output of the network was considered as $Q(t)$. From the total available data for 3 years, 6525 patterns (input-output pairs) were identified for the study and were split into training (5500 sets) and validation (1025 sets) data sets. Note that the 1025 sets considered for validation were corresponding to a continuous hydrograph. A single hidden layer is considered in the study based on various research studies conducted on this basin [Nayak *et al.*, 2005; Chetan and Sudheer, 2006]. Out of the 5500 training examples, 4500 patterns were randomly bootstrapped every time. The optimal number of hidden neurons was found to be three by trial and error procedure. Note that the number of hidden neurons was varied for only the first model, and for the subsequent models the number was fixed as that was found optimal for the first model so as to maintain consistency. It should be noted that earlier studies using

bootstrap have revealed that variation in forecast due to changes in structure or architecture are small in comparison with those that arise from sample splitting [LeBaron and Weigend, 1998]. Thus the uncertainty arising from the architecture has not been considered in this study. In this study a total of 300 networks have been trained for the analysis.

[26] The ANN network structure in the present study is shown in Figure 2. The following nomenclature is used to represent the links in this paper: the link connecting I_1 (input

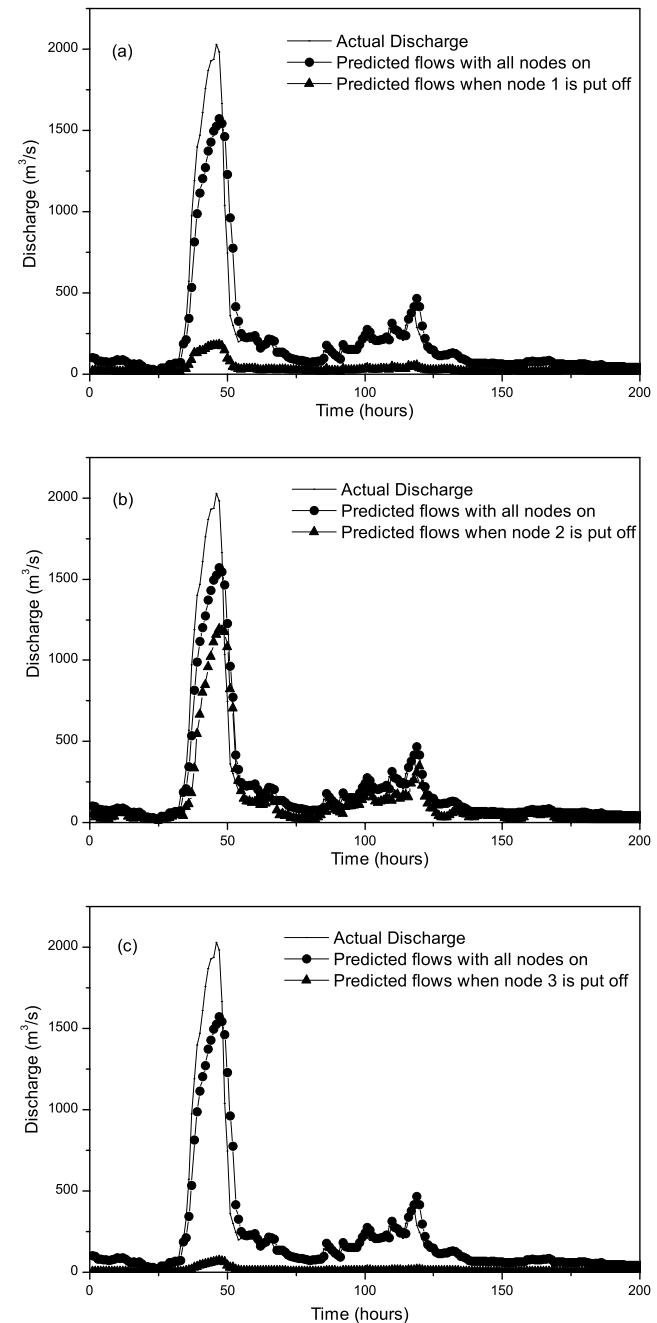


Figure 3. ANN-computed flows for a typical event during validation (mean of 300 simulations): (a) when hidden node 1 (H_1) is put off, (b) when hidden node 2 (H_2) is put off, and (c) when hidden node 3 (H_3) is put off.

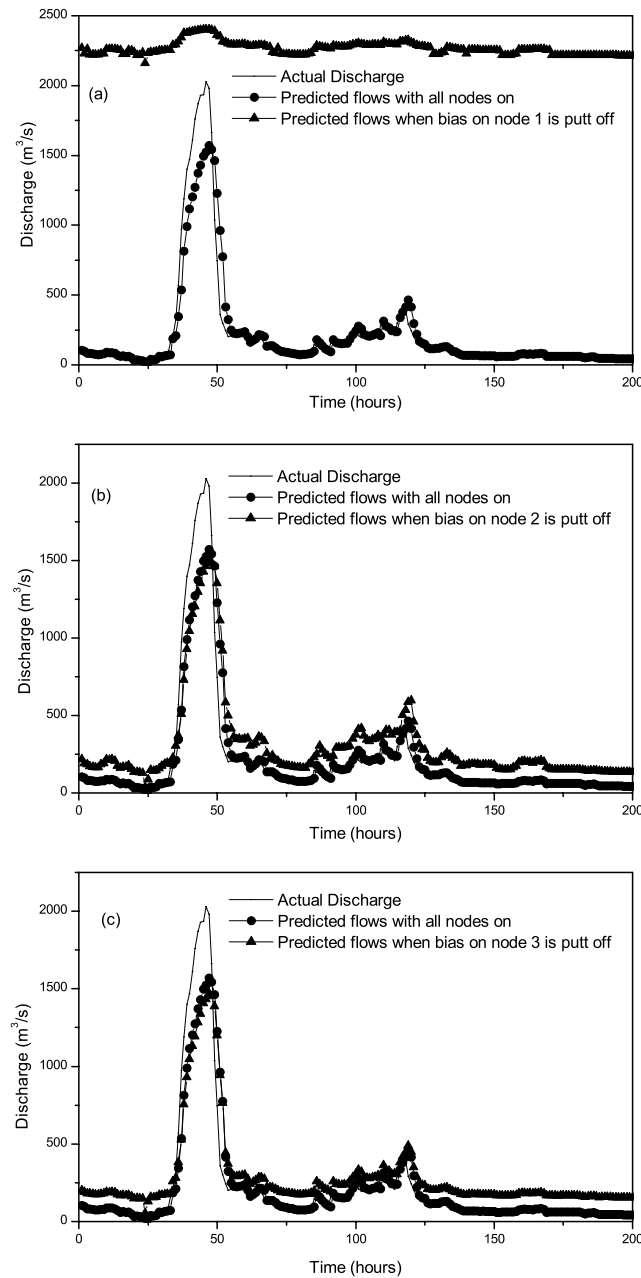


Figure 4. ANN-computed flows for a typical event during validation (mean of 300 simulations): (a) when bias parameter on hidden node 1 (H_1) is put off, (b) when bias parameter on hidden node 2 (H_2) is put off, and (c) when bias parameter on hidden node 3 (H_3) is put off.

1) to H_1 (hidden node 1) is WI_1H_1 ; I_1 to H_2 is WI_1H_2 ; I_1 to H_3 is WI_1H_3 ; and so on for other inputs. Links connected from hidden node to the output node are designated as WH_1O_1 (H_1 to O_1), WH_2O_1 (H_2 to O_1), and WH_3O_1 (H_3 to O_1), and so on.

5. Results and Discussions

[27] As discussed earlier, the uncertainty in parameter estimation and the predictive uncertainty have been evaluated for the ANN model developed for forecasting the river

flow for Kolar Basin at a lead time of 1 hour. The results of the study are discussed in detail in the following sections.

5.1. ANN Models' Performance

[28] The performance of the models (300 independent models) has been evaluated using various statistical indices such as root-mean-square error (RMSE), Nash-Sutcliffe efficiency [Nash and Sutcliffe, 1970], and the coefficient of correlation between the measured and computed flow values. The summary statistics of the performance indices across the 300 models are presented in Table 1. It is observed that the variation of RMSE for most of the models is not very significant. The RMSE statistic has a mean value of $58.22 \text{ m}^3/\text{s}$ with a standard deviation of $8.84 \text{ m}^3/\text{s}$. It is noted that 90% of the models produce an RMSE within a band of $\pm 5\%$ around the mean value, suggesting that the impact of training samples does not have a significant effect on the model predictions. It is found that most of the models forecast the flows with a correlation (between computed and measured flow) varying from 0.94 to 0.96. A similar behavior is observed in the case of efficiency statistic. The models possess efficiency with a mean value of 90%, which according to Shamseldin [1997] is very satisfactory. The deviation of efficiency from mean value is only to the tune of $\pm 5\%$. In addition, the mean value of the mean bias error is -1.88 and standard deviation is 0.6, which is also very satisfactory. Overall, the results indicate that the variation in training patterns do not have a significant effect on the overall model performance.

5.2. Uncertainty in Model Parameters

[29] The network architecture being 5-3-1 (Figure 2), the model has 22 parameters including the bias terms. A summary of statistics of the variation of all the 22 parameters are presented in Table 2. It can be noted from Table 2 that some of the input-hidden weights (WI_1H_3 , WI_2H_3 , WI_3H_3) do not have much variation as their range is very small (fewer than three units). It is interesting to note that all these weights are connections from rainfall information, where as the connections from discharge nodes (WI_4H_1 , WI_5H_1 , WI_4H_2 , WI_5H_2 , WI_4H_3 , WI_5H_3) have high variation and large values as is evident from Table 2. It is also observed that the weight pairs (WI_4H_1 , WI_5H_1), (WI_4H_2 , WI_5H_2), and (WI_4H_3 , WI_5H_3) are strongly correlated to each other with a correlation coefficient of -0.989 , -0.989 , and -0.793 , respectively. However, it is to be noted that the strength of the relationship between the weights from rainfall inputs and weights from flow inputs is found to be weak. Hence flow inputs are much more significant in generating the predictions compared with the rainfall inputs and therefore flow inputs require larger weights (see Table 2).

[30] It is noted from Table 2 that the mean value of weight connection between first hidden node and the output is relatively high (23.21 units) compared to the weights from other hidden nodes, indicating that the response from first hidden node plays a major role in the model output. However, this behavior may also be due to over-parameterization of the network since we employed linear transfer function at the output node and the ANN may be trying to overcome this over-parameterization by saturating the hidden node. In order to confirm the reason behind this behavior of the first hidden node, an empirical trial was

Table 3. Parameters of Probability Distribution of Weights and Biases Obtained From 300 ANN Models^a

| ANN Parameters | Best Describing Distribution | Parameters of the Distribution |
|--------------------------------|------------------------------|-----------------------------------|
| WI ₁ H ₁ | Pearson5 | 0.8612, 0.1207 |
| WI ₁ H ₂ | logistic | -1.2538, 0.5635 |
| WI ₁ H ₃ | extreme value | 0.0625, 0.1763 |
| WI ₂ H ₁ | log logistic | -6.5902, 6.7340, 54.9310 |
| WI ₂ H ₂ | Weibull | 6.2909, 3.2272 |
| WI ₂ H ₃ | log logistic | -8.3324, 8.4019, 88.4230 |
| WI ₃ H ₁ | log logistic | -0.0503, 0.3360, 4.4742 |
| WI ₃ H ₂ | log logistic | -3.6077, 2.1983, 5.1720 |
| WI ₃ H ₃ | lognormal | 0.9071, 0.2252 |
| WI ₄ H ₁ | Weibull | 86.7450, 585.2700 |
| WI ₄ H ₂ | triangular | -9.6057, 50.7620, 60.2570 |
| WI ₄ H ₃ | log logistic | -1.4920, 3.4224, 2.9183 |
| WI ₅ H ₁ | log logistic | -10.8300, 14.2550, 8.1444 |
| WI ₅ H ₂ | triangular | -50.8980, -41.3680, 8.8260 |
| WI ₅ H ₃ | logistic | 2.8222, 1.2648 |
| B1 | logistic | -5.5831, 1.9065 |
| B2 | normal | -2.0137, 3.2215 |
| B3 | logistic | 0.06969, 0.9251 |
| WH ₁ O ₁ | logistic | 24.5201, 4.2732 |
| WH ₂ O ₁ | Pearson5 | 0.8463, 0.1958 |
| WH ₃ O ₁ | inverse Gaussian | 7.8405, 9.0398 |
| B4 | beta general | 5.2362, 0.6417, -63.3200, -3.0453 |

^aThe parameters of the distribution are presented in the order of their appearance in the equation describing the distribution available in any standard text book [e.g., Haan, 2002].

conducted by putting off the individual hidden nodes (one at a time) during the validation phase. The resulting hydrograph during this experiment are presented in Figures 3a through 3c. Note that the mean value of the predicted flows from 300 networks is presented in Figure 3. It is evident from Figure 3a that the response of the first hidden node is highly significant in effective prediction of the flows. A similar pattern is observed in the predicted values of flow when the third hidden node is put off (Figure 3c). It may be noted that the value of weight connection between third hidden node and the output is relatively lesser compared with that of first hidden node (Table 2). Hence it can be concluded that the higher value of weights for first hidden node may not be resulting from saturation problem. The results presented in Figure 3b indicate that the response from second hidden node mainly affects the rising limb of the flood hydrograph.

[31] In order to assess the sensitivity of the bias parameter associated with each of the hidden node, a similar experiment was conducted by putting off each of them independently while validating the model. The results are presented in Figures 4a–4c, from which it is evident that the bias parameter at first hidden node (B1) is highly sensitive in computing the output effectively, compared with the other two. This result also reinforces the earlier inferences that first hidden node is more sensitive in the flow prediction.

[32] The probability distribution of each of the model parameters is identified, and the parameters of the corresponding distribution are presented in Table 3 for all the 22 model parameters. The distribution parameters presented in Table 3 are quantified using the Best Fit program (Palisades Corp., Ithaca, New York), which considers 28 different distributions to the data, and ranks them according to a specified criterion. The parameters of the probability

distributions are estimated using maximum-likelihood estimator [Haan, 2002]. The chi-square goodness of fit test was performed to evaluate and rank the distributions that best described the data. It can be observed from Table 3 that the weights and bias parameters follow different probability distribution, indicating that an assumption about prior probability distribution of the ANN model parameters is difficult. Hence predictions based on methods that uses Bayesian approach for uncertainty analysis (e.g., GLUE, Bayesian neural network, etc.) will lead to biased output, since these methods rely mostly on the a priori assumption about the distribution of model parameters. The distribution of optimized weights and bias (selected) are presented in Figure 5 for demonstration. Note that the distribution for other parameters is not presented here for brevity. It is noted that six of the model parameters (WI₂H₁, WI₂H₃, WI₃H₁, WI₃H₂, WI₄H₃, WI₅H₁) follow “log logistic distribution” but with varying shape and location parameters. Similarly for those parameters which follow “logistic distribution” (WI₁H₂, WI₅H₃, B₁, B₃, WH₁O₁), it is observed that the mean and variance of the distribution do not follow any specific order. These results suggest that any a priori assumption about the model parameters’ distribution will lead to additional uncertainty in the model output.

5.3. Uncertainty in Model Predictions

[33] The variations in model predictions by the 300 models are presented using box plot for two typical flood events in the validation data set in Figures 6 and 7, respectively. It can be observed that the base flow portion of the hydrograph is well forecasted by the models irrespective of the variation in training samples. This may be due to more example data present in the each training set in the low flow range. It is clear from these figures that ANN models do fail to preserve the peak flow characteristics of the hydrograph. However, the trend of variation in the flow hydrograph is well preserved by the models, which is evident from Figure 8, which depicts a series of continuous events. It is observed that the rising limb of the flow hydrograph is most of the time underestimated by the models, while the falling limb is mostly overestimated.

5.4. Uncertainty in High-Flow Predictions

[34] It has been reported by many researchers that ANN models fail to capture the peak flows in a hydrograph [Imrie et al., 2000; Sudheer et al., 2002]. This observation is confirmed in the current study by analyzing the high flow prediction uncertainty by the ANN models. To analyze the uncertainty in the high flow predictions (hereinafter high flows are defined as $Q \geq \mu + 2\sigma$), frequency plots of a few typical high flow values are shown in Figures 9–12. It is noted that the models fail to compute any of the high flow, except for a few. It is worth mentioning that the actual value of the high flow does not fall in the prediction confidence interval for the model. While the reason for such behavior of ANN is not clearly known, it can be attributed to a lower number of examples available in the high flow domain for capturing the nonlinear dynamics. It may be noted that only 5% of the total training examples were in the high flow domain as defined earlier. It may also be plausibly due to inaccurate representation of the state of saturation by the input vectors as envisaged by Campolo et al. [1999]. It may

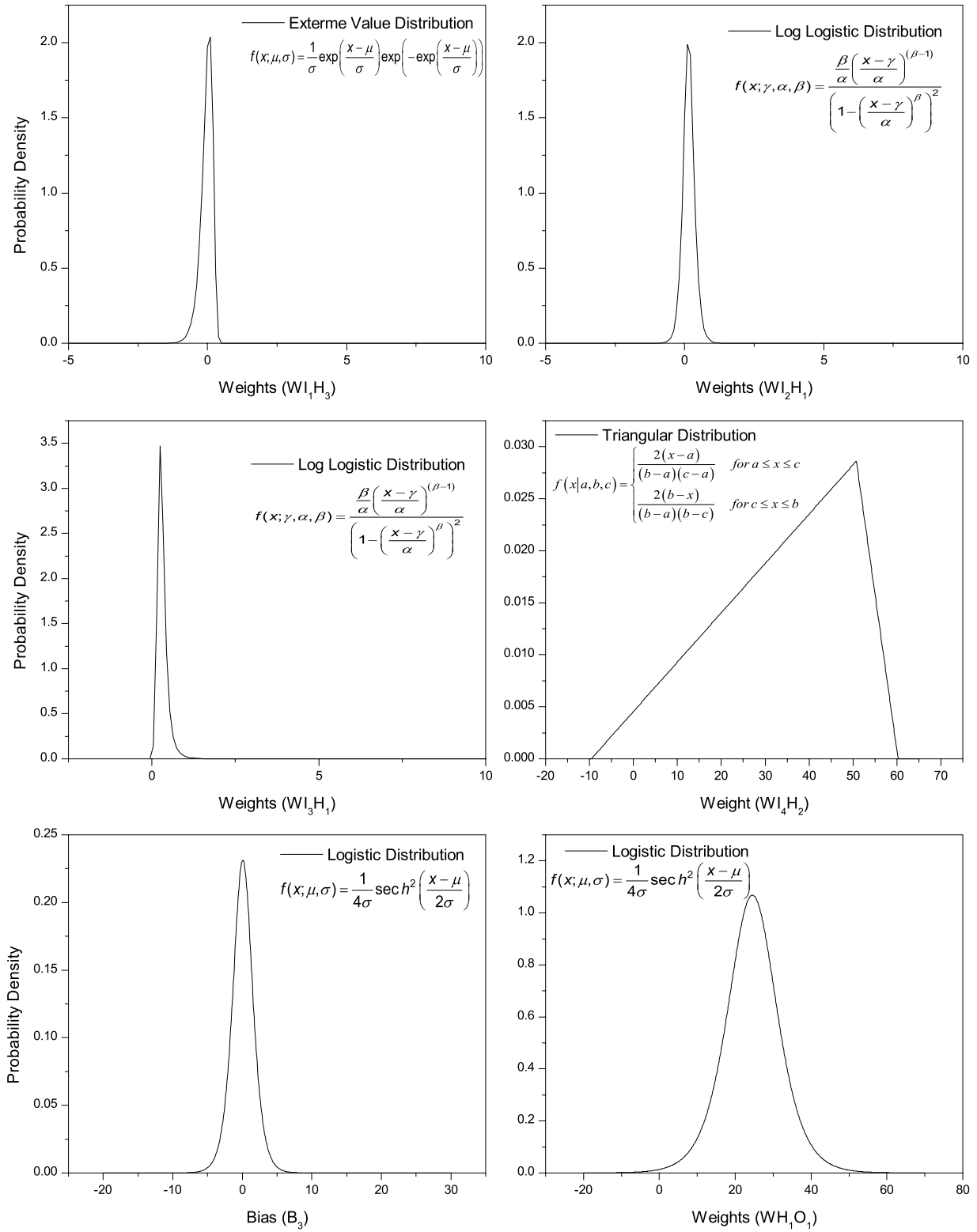


Figure 5. Empirical probability distribution of selected parameters of the network.

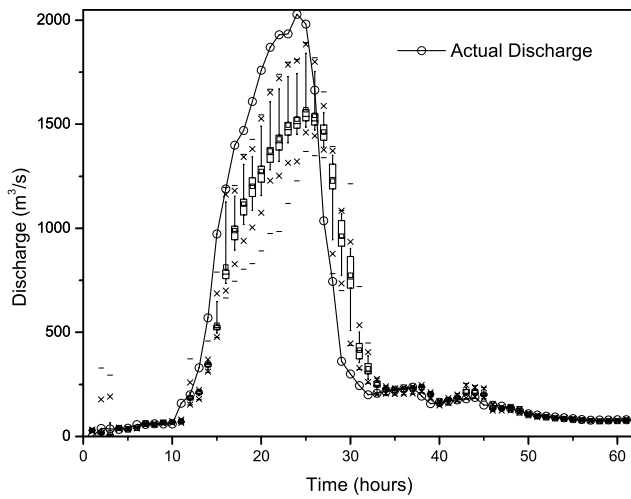


Figure 6. Box plot of predicted discharges from 300 model simulations for a typical flood event (single peak event 1) during validation.

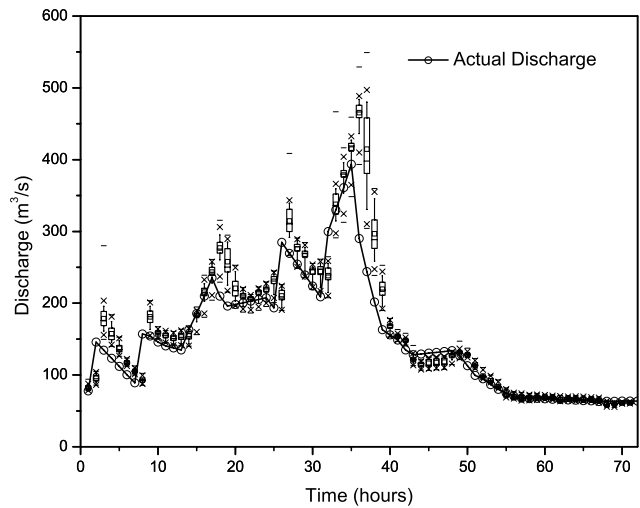


Figure 8. Box plot of predicted discharges from 300 model simulations for a typical continuous flood event (multiple peak events) during validation.

be noted that the current study did not consider the uncertainty associated with selection of inputs.

[35] It is interesting to note that the empirical distribution of peak flow predictions corresponding to 1663, 1930, and 2028 m³/s are observed to be bimodal in nature. This behavior of the predictions can be attributed to the random sampling employed during bootstrapping of training patterns; it is possible that the bootstrapped subset for some of the networks (out of 300) might not have selected these peak flows. Since ANN is not good in extrapolation [Minns and Hall, 1996], the predicted peak may be less than the actual. This observation suggests scope for further studies related to analysis of uncertainty of ANN using conditional bootstrapping.

5.5. Impact of Initial Guess of Parameters

[36] In order to measure the effect of assumed initial values on the model’s performance an analysis has been

conducted by changing the initial guess of weights and biases and the model’s predictive uncertainty has been evaluated. During this analysis, six sets of initial values have been assigned during the training of ANN, and 300 networks have been trained using each of the initial guesses through bootstrapping of the training patterns. Note that the bootstrapped training samples were kept same for each of the initial guesses in order to have a true comparison of the models’ performance. The prediction characteristics of the models in terms of the mean of the 300 network outputs for this analysis are presented in Figure 13 (for one typical hydrograph in the validation data set). Note that the hydrograph presented in Figure 13 is the ensemble mean of 300 network outputs. It is observed that the initial values of the parameters do not significantly affect the models’ performance in low flow periods. Though the mean predictions

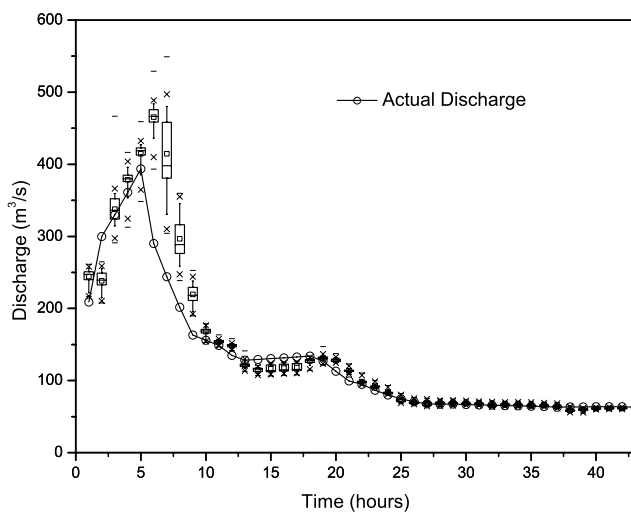


Figure 7. Box plot of predicted discharges from 300 model simulations for a typical flood event (single peak event 2) during validation.

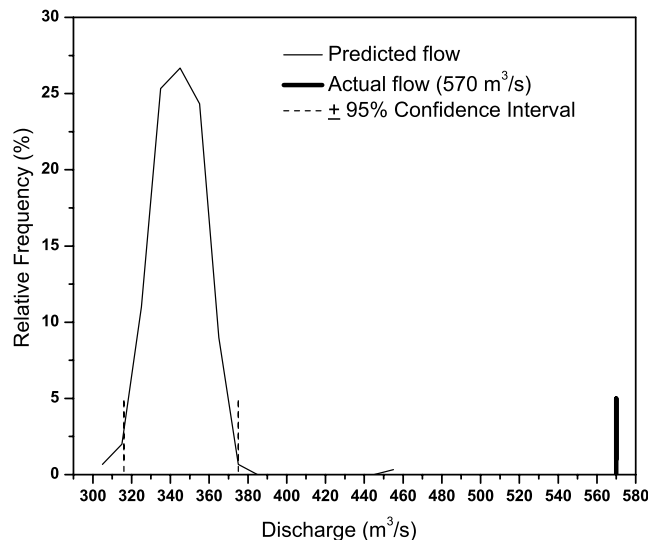


Figure 9. Uncertainty in high flow predictions for actual flow rate of 570 m³/s.

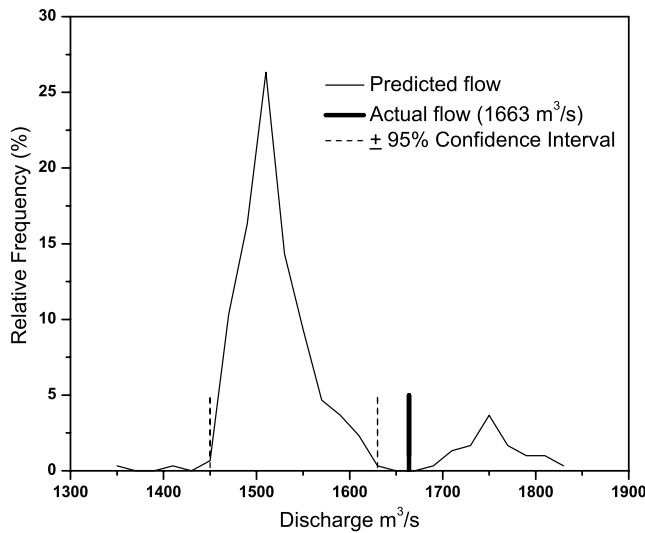


Figure 10. Uncertainty in high flow predictions for actual flow rate of 1663 m³/s.

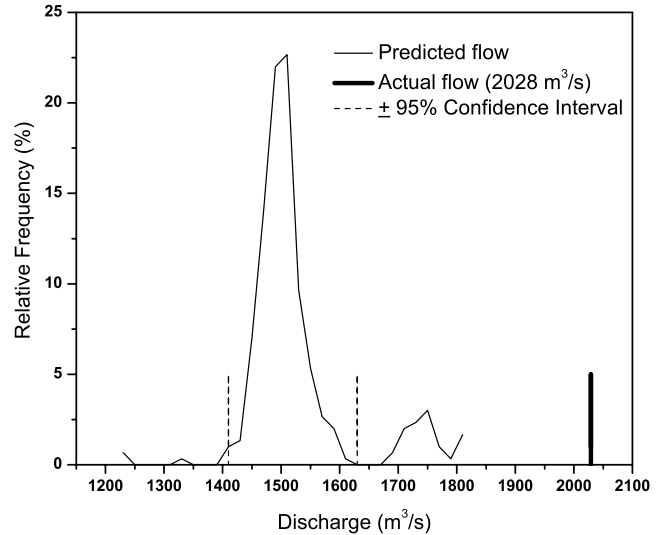


Figure 12. Uncertainty in high flow predictions for actual flow rate of 2028 m³/s.

are slightly different during the flood, it is observed that the deviation is larger only in the rising limb. It may also be noted that some of the predictions with different initial guesses coincide, indicating that the predictions are not significantly biased by the initial guesses. However, it is to be noted that the range of predictions within 300 networks (inferred from box plots) with different initial guesses are found to be the same. Note that the box plot is not presented here for brevity.

6. Summary and Conclusions

[37] This paper presents a method to conduct uncertainty analysis for ANN hydrologic models. The method, which is based on bootstrap technique, is demonstrated by a real world case study of Kolar basin in India. The analysis of results illustrate that the proposed method of uncertainty

analysis of ANN models is very effective. The results of the case study suggest that the sampling variability does not significantly affect the performance in the case of ANN model developed for the Kolar basin. It is worth mentioning that despite good generalization properties for all the 300 models developed, the models fail to capture the peak flow characteristics of the hydrograph. This observation is significant, as in a flood forecasting context accurate estimation of peak flow is very important. The results also suggest that the performance evaluation solely based on global statistical indices such as RMSE, efficiency, etc., which is being practiced currently, does not provide good confidence in model's use. The study also suggests the requirement of a good performance indicator that can be used for evaluating ANN models' applicability for practical use. Note that the proposed method evaluates the uncertainty associated with sampling variation of training patterns only. It should be noted that in the current study, the model architecture has

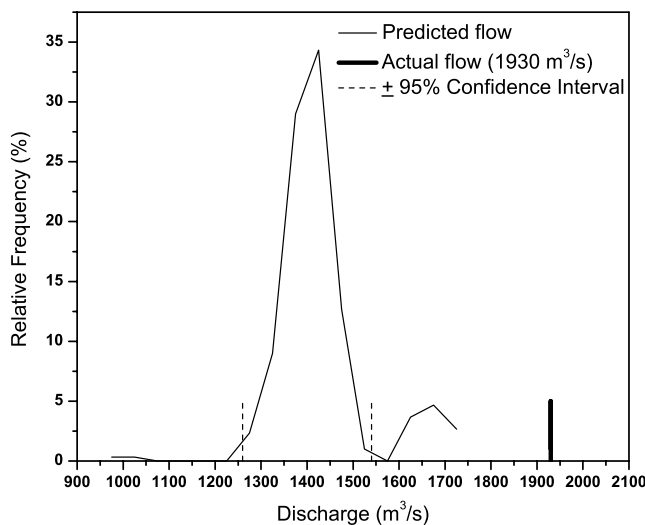


Figure 11. Uncertainty in high flow predictions for actual flow rate of 1930 m³/s.

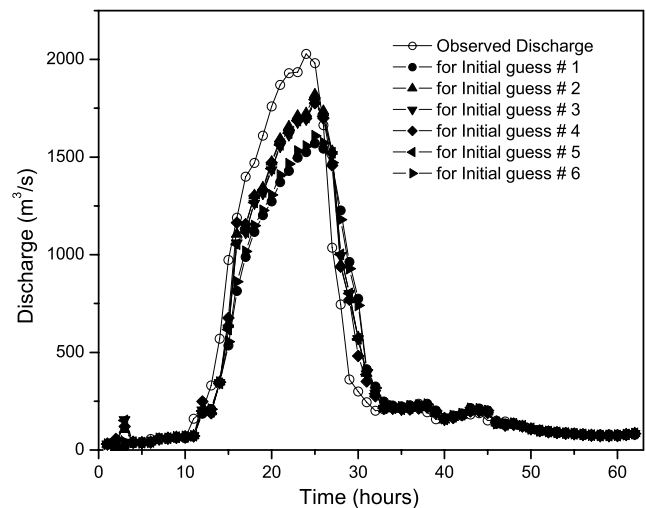


Figure 13. Predicted discharge (mean of 300 simulations) by ANN model with varying initial values of parameters during training.

been considered to be deterministic. Also, the input variables have been fixed initially before the uncertainty analysis. Hence more rigorous studies are required to accommodate the uncertainties associated with model architecture and input selection.

References

- Abrahart, R. J. (2003), Neural network rainfall-runoff forecasting based on continuous resampling, *J. Hydroinformatics*, 5, 51–61.
- Adkison, M., and R. M. Peterman (1996), Results of Bayesian methods depend on details of implementation: An example of estimating salmon escapement goals, *Fish. Res.*, 25, 155–170.
- Beven, K., and A. Binley (1992), The future of distributed models: Model calibration and uncertainty prediction, *Hydrol. Processes*, 6, 279–298.
- Beven, K., and J. Freer (2001), Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems, *J. Hydrol.*, 249, 11–29.
- Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford Univ. Press, New York.
- Bowden, G. J., G. C. Dandy, and H. R. Maier (2004a), Input determination for neural network models in water resources applications: 1. Background and methodology, *J. Hydrol.*, 301(1–4), 75–92.
- Bowden, G. J., G. C. Dandy, and H. R. Maier (2004b), Input determination for neural network models in water resources applications: 2. Background and methodology, *J. Hydrol.*, 301(1–4), 93–107.
- Campolo, M., P. Andreussi, and A. Soldati (1999), River flood forecasting with a neural network model, *Water Resour. Res.*, 35(4), 1191–1197.
- Chetan, M., and K. P. Sudheer (2006), A hybrid linear-neural model for river flow forecasting, *Water Resour. Res.*, 42, W04402, doi:10.1029/2005WR004072.
- Christiaens, K., and J. Feyen (2002), Use of sensitivity and uncertainty measures in distributed hydrological modeling with an application to the MIKE SHE model, *Water Resour. Res.*, 38(9), 1169, doi:10.1029/2001WR000478.
- Dybowski, R., and S. J. Roberts (2000), Confidence and prediction intervals for feed-forward neural networks, in *Clinical Applications of Artificial Neural Networks*, edited by R. Dybowski and V. Gant, pp. 298–326, Cambridge Univ. Press, New York.
- Efron, B. (1979), Bootstrap methods: Another look at Jackknife, *Ann. Stat.*, 7(1), 1–26.
- Efron, B., and R. J. Tibshirani (1993), *An Introduction to the Bootstrap*, CRC Press, Boca Raton, Fla.
- Freer, J., and K. M. Beven (1996), Bayesian estimation of uncertainty in runoff prediction and the value of data: An application of the GLUE approach, *Water Resour. Res.*, 32(7), 2121–2173.
- Haan, C. T. (2002), *Statistical Method in Hydrology*, Iowa State Univ. Press, Ames.
- Han, D., T. Kwong, and S. Li (2007), Uncertainties in real-time flood forecasting with neural networks, *Hydrol. Processes*, 21(2), 223–228, doi:10.1002/hyp.6184.
- Hsu, K., V. H. Gupta, and S. Sorooshian (1995), Artificial neural network modeling of the rainfall-runoff process, *Water Resour. Res.*, 31(10), 2517–2530.
- Imrie, C. E., S. Durucan, and A. Korre (2000), River flow prediction using artificial neural networks: Generalization beyond the calibration range, *J. Hydrol.*, 233, 138–153.
- Janssen, P. H. M., P. Heuberger, and R. Sanders (1992), A software package for sensitivity and uncertainty analysis—Manual, *Rep. 959101004*, Natl. Inst. of Public Health and Environ. Prot., Bilthoven, Netherlands.
- Khan, M. S., and P. Coulibaly (2006), Bayesian neural network for rainfall-runoff modeling, *Water Resour. Res.*, 42, W07409, doi:10.1029/2005WR003971.
- Kingston, G. B., M. F. Lambert, and H. R. Maier (2005), Bayesian training of artificial neural network used for water resources modeling, *Water Resour. Res.*, 41, W12409, doi:10.1029/2005WR004152.
- LeBaron, B., and A. S. Weigend (1998), A bootstrap evaluation of the effect of data splitting on financial time series, *IEEE Trans. Neural Networks*, 9, 213–220.
- Luk, K. C., J. E. Ball, and A. Sharma (2000), A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting, *J. Hydrol.*, 227(1–4), 56–65.
- Maier, H. R., and G. C. Dandy (2000), Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications, *Environ. Modell. Software*, 15(1), 101–124.
- Minns, A. W., and M. J. Hall (1996), Artificial neural networks as rainfall-runoff models, *J. Hydrol. Sci.*, 41, 399–417.
- Morgan, M. G., and M. Henrion (1990), *Uncertainty: A Guide to Dealing With Uncertainty in Quantitative Risk and Policy Analysis*, 322 pp., Cambridge Univ. Press, New York.
- Nash, J. E., and J. V. Sutcliffe (1970), River flow forecasting through conceptual models: 1. A discussion of principles, *J. Hydrol.*, 10, 282–290.
- Nayak, P. C., K. P. Sudheer, D. M. Rangan, and K. S. Ramasastri (2005), Short-term flood forecasting with a neurofuzzy model, *Water Resour. Res.*, 41, W04004, doi:10.1029/2004WR003562.
- Rivals, I., and L. Personnaz (1998), Construction of confidence intervals in neural modeling using a linear Taylor expansion, paper presented at the International Workshop on Black-Box Techniques for Nonlinear Modeling, Katholieke Universiteit, Leuven, Belgium, 8–10 July.
- Rogers, C. C. M., K. J. Beven, E. M. Morris, and M. G. Anderson (1985), Sensitivity analysis calibration and predictive uncertainty of the Institute of Hydrology Distributed Model, *J. Hydrol.*, 81, 179–191.
- Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986), Learning representations by back-propagating errors, *Nature*, 323(6088), 533–536.
- Sajikumar, N., and B. S. Thandaveswara (1999), A nonlinear rainfall-runoff model using an artificial neural network, *J. Hydrol. Eng.*, 216(1–2), 32–55.
- Shamseldin, A. Y. (1997), Application of a neural network technique to rainfall-runoff modelling, *J. Hydrol.*, 199(3–4), 272–294.
- Shrestha, D. L., and D. P. Solomatine (2006), Machine learning approaches for estimation of prediction interval for the model output, *Neural Networks*, 19(2), 225–235.
- Silverman, D., and J. A. Dracup (2000), Artificial neural networks and long-range precipitation prediction in California, *J. Appl. Meteorol.*, 39(1), 57–66.
- Stone, M. (1974), Cross validation choice and assessment of statistical predictions, *J. R. Stat. Soc., Ser. B*, 36, 44–47.
- Sudheer, K. P., A. K. Gosain, and K. S. Ramasastri (2002), A data-driven algorithm for constructing artificial neural network rainfall-runoff models, *Hydrol. Processes*, 16(6), 1325–1330.
- Sudheer, K. P., P. C. Nayak, and K. S. Ramasastri (2003), Improving peak flow estimates in artificial neural network river flow models, *Hydrol. Processes*, 17(3), 677–686.
- Thirumalaiah, K., and M. C. Deo (2000), Hydrological forecasting using neural networks, *J. Hydrol. Eng.*, 5(2), 180–189.
- Wagener, T. (2003), Evaluation of catchment models, *Hydrol. Processes*, 17, 3375–3378.

I. Chaubey, Department of Agricultural and Biological Engineering, Purdue University, West Lafayette, IN 47907, USA. (ichaubey@purdue.edu)

R. K. Srivastav and K. P. Sudheer, Department of Civil Engineering, Indian Institute of Technology, Madras, Chennai 600036, India. (rksrivastav@rediffmail.com; sudheer@iitm.ac.in)