Artificial neural network ensembles and their application in pooled flood frequency analysis

Chang Shu and Donald H. Burn

Department of Civil Engineering, University of Waterloo, Waterloo, Ontario, Canada

Received 30 October 2003; revised 27 May 2004; accepted 7 June 2004; published 4 September 2004.

[1] Recent theoretical and empirical studies show that the generalization ability of artificial neural networks can be improved by combining several artificial neural networks in redundant ensembles. In this paper, a review is given of popular ensemble methods. Six approaches for creating artificial neural network ensembles are applied in pooled flood frequency analysis for estimating the index flood and the 10-year flood quantile. The results show that artificial neural network ensembles generate improved flood estimates and are less sensitive to the choice of initial parameters when compared with a single artificial neural network. Factors that may affect the generalization of an artificial neural network ensemble are analyzed. In terms of the methods for creating ensemble members, the model diversity introduced by varying the initial conditions of the base artificial neural networks to reduce the prediction error is comparable with more sophisticated methods, such as bagging and boosting. When the same method for creating ensemble members is used, combining member networks using stacking is generally better than using simple averaging. An ensemble size of at least 10 artificial neural networks is suggested to achieve sufficient generalization ability. In comparison with parametric regression methods, properly designed artificial neural network ensembles can significantly reduce the prediction error. *INDEX TERMS:* 1821 Hydrology: Floods; 1860 Hydrology: Runoff and streamflow; 1869 Hydrology: Stochastic processes; 1894 Hydrology: Instruments and techniques; *KEYWORDS:* artificial neural network ensemble, bagging, boosting, stacking, pooled flood frequency analysis, index flood

Citation: Shu, C., and D. H. Burn (2004), Artificial neural network ensembles and their application in pooled flood frequency analysis, *Water Resour. Res.*, *40*, W09301, doi:10.1029/2003WR002816.

1. Introduction

[2] An artificial neural network (ANN), as a relatively new approach to modeling both regression and classification problems, has numerous applications in many scientific fields. ANNs have been widely used for solving a range of hydrological problems such as rainfall-runoff modeling, streamflow forecasting, water quality modeling, groundwater modeling, hydrological time series modeling, and reservoir operation [*Task Committee*, 2000]. *Govindaraju and Rao* [2000] demonstrate the diverse applications in hydrology and document a range of ANN architectures and training algorithms.

[3] Probably the greatest concern with the application of ANNs is the generalization ability as reflected in the prediction accuracy on data that were not used to train the model. Recent studies show that the generalization ability of ANNs can be improved by combining several ANNs in redundant ensembles, where the member networks are redundant in that each of them provides a solution to the same task, or task component, even though this solution might be obtained by different methods [*Sharkey*, 1999]. This approach is now formally known as an artificial neural network ensemble. An ANN ensemble is a finite number of ANNs that are trained for the identical purpose whose predictions are combined to generate a unique output. ANN ensembles offer a number of advantages over a single ANN in that they have the potential for improved generalization and increased stability [*Sharkey*, 1999]. Ensemble methods have been successfully applied in various domains, such as time series prediction, robotics, and medical diagnosis. *See and Abrahart* [2001] used ANN data fusion strategies for continuous river level forecasting where data fusion is the amalgamation of information from multiple sensors and/or different data sources. *Abrahart and See* [2002] evaluated six data fusion strategies and found that ANN data fusion provided the best solution for a stable region. *Cannon and Whitfield* [2002] used a bootstrap aggregated ANN ensemble as a downscaling model to predict streamflow and changes in streamflow conditions in British Columbia, Canada. The results showed that an ANN ensemble was better than a stepwise linear regression model and that an ensemble approach helped to mitigate some of the problems commonly encountered when applying ANNs in hydrology. There have been theoretical studies to explain the working mechanisms of ANN ensembles. *Hansen and Salamon* [1990] established that the generalization ability of a single ANN can be significantly improved using an ensemble of ANNs with a plurality consensus scheme in which the final

0043-1397/04/2003WR002816$09.00

decision is the classification reached by the majority of the networks. *Krogh and Vedelsby* [1995] established that the generalization ability of an ensemble is determined by the average generalization ability and the average ambiguity of the individual ANNs that constitute the ensemble.

[4] The success of the ensemble approach motivates further investigation of its suitability in solving problems in hydrologic modeling. In this paper, a review is provided of popular methods for creating ANN ensembles. Since most ensemble methods were originally proposed for solving classification problems, the effectiveness of ensemble approaches in solving a typical regression problem in pooled flood frequency analysis (also known as regional flood frequency analysis) is evaluated.

[5] Accurate flood quantile estimates are required for the design of hydraulic structures and for floodplain management. To obtain improved flood estimates, pooled flood frequency analysis has been introduced. Pooled flood frequency analysis is a viable approach to obtain reliable flood estimates at catchments where the flood record is either short or not available. There have been two main approaches to pooled flood frequency analysis: the index flood method [*Dalrymple*, 1960] and the quantile regression method. In the index flood method it is assumed that the distributions of flood peaks at different sites within a pooling group are the same except for a scale parameter that is the index flood for a site. Accurate estimation of the index flood is important, since uncertainty in the index flood estimate is often a major contributor to uncertainty in quantile estimates at ungaged sites. In the regression-based method, catchment characteristics of a site are related to the flood quantile of interest using a regression model. This method can also be used to estimate the index flood of an ungaged site [*Zrinji and Burn*, 1994; *Hosking and Wallis*, 1997].

[6] The ANN ensemble models implemented in this paper are applied to regression-based pooled flood frequency analysis. The use of an ANN is due to its capability of capturing complex nonlinear relationships in the underlying data that are often missed by conventional regression methods, such as multiple linear regression. Using more formal terms of regression to describe the ANN model, let x be the site characteristics of a site in a pooling group (the input variables), y is the index flood or flood quantile of the site, and then the relationship between x and y can be expressed as

$$y = f(x) + \varepsilon(x), \tag{1}$$

where $f(x)$ is the output of the physical system and $\varepsilon(x)$ is noise. The task of regression is to approximate the function $f(x)$. The implementation of the ANN model usually involves three phases: a training phase to determine the model parameters from a set of training data, a validation phase to estimate the generalization ability of the model, and a test phase to calculate the output using the optimized model.

[7] The remaining parts of this paper are organized as follows. An introduction to ensemble methods is given in section 2. In section 3, ANN ensemble models used for flood frequency analysis are presented. In section 4, various ensemble methods are analyzed and compared. Section 5 summarizes the important research conclusions.

2. General Ensemble Methods

[8] The creation of an ensemble can be divided into two steps. The first step is to create individual ensemble members, and the second step is the appropriate combination of output from the ensemble members to produce the ensemble output. There are many different methods that have been proposed for creating ensembles. *Opitz and Maclin* [1999] and *Sharkey* [1999] provide reviews of ensemble methods.

2.1. Methods for Creating Ensemble Members

[9] The main reason for combining ANNs in redundant ensemble is to achieve better performance. Apparently, there is no advantage to combining a set of identical nets that have similar generalization patterns. The goal is therefore to design nets with diverse generalization ability. The following approaches are the most widely used [*Sharkey*, 1999]: (1) manipulating the set of initial random weights, (2) varying the topologies, (3) varying the training algorithm, or (4) manipulating the training set. Among these approaches, the simplest way for creating diverse ensemble members is to train each network using randomly initialized weights. However, more elaborate approaches of training different ANNs on different training sets have received the most attention, with two popular methods being bagging [*Breiman*, 1996a] and boosting [*Freund and Schapire*, 1996; *Schapire*, 1990]. Both of these methods are based on resampling techniques to alter the training sets for each component network. *Dietterich* [2000] compared the performance of randomization, bagging, and boosting. The results showed that boosting often gives the best results, while bagging and randomization give similar performance.

2.1.1. Bagging

[10] Bagging [*Breiman*, 1996a] is an acronym for "bootstrap aggregation" and was originally proposed in connection with classification and regression trees. This approach is based on the bootstrap statistical resampling technique [*Efron and Tibshirani*, 1993], to generate diverse training sets that are used to train the members composing an ensemble. Suppose the training set T consists of m instances. Each instance is assigned a probability of $1/m$, and the training set of a member network, T_B, is generated by sampling with replacement m times from the original training set T, using these probabilities. Thus many cases in T may be repeated several times in T_B, while others may be left out. This process is repeated and each member network is generated with a different random sampling of the original training set.

[11] *Breiman* [1996a] concluded that bagging is effective on "unstable" learning algorithms. Predictors such as ANNs and regression trees are suitable for bagging. *Carney and Cunningham* [1999] and *Zhang* [1999] studied bagging in the context of ANNs and concluded that model generalization ability can be significantly improved.

2.1.2. Boosting

[12] Boosting algorithms [*Schapire*, 1990; *Freund and Schapire*, 1996] achieve improved performance by producing a series of predictors trained with a different distribution of the original training data. The algorithm trains the first predictor with the original training set, and the training set

of a new predictor is assembled based on the performance of the previous predictors. The learning patterns whose predicted values obtained from the previous predictor differ significantly from their observed values are adjusted with higher probability of being sampled, so they will have a greater chance of appearing in the new training set than those correctly predicted. Thus different predictors are specialized in different parts of the observation space.

[13] Since the standard version of boosting is designed for classification, an algorithm ADABoost.R2 [*Drucker*, 1997], designed to work for regression, is employed in this research. ADABoost.R2 is a variation on the ADABoost.R algorithm [*Freund and Schapire*, 1996]. When applied to ANNs, ADABoost.R2 proceeds by iteratively training networks using training sets determined by the performance of the previous network. An important feature of the ADABoost.R2 algorithm is the sampling distribution. Suppose the training set T consists of m instances $(x_1, y_1), \ldots, (x_m, y_m)$, the ith element of T has a value in the sampling distribution $D_t(i)$ at step t, and this value represents the probability of that element being included in the next training set. The following is a description of the AdaBoost.R2 algorithm [*Drucker*, 1997].

[14] Initially, each value in the distribution is assigned the same value so that each element in the initial data set has an equal chance of being included in the first training set and set the step $t = 1$:

$$D_1(i) = 1/\mathrm{m}, \text{ over all } i. \tag{2}$$

Iterate the following while the average loss $\overline{L}$, defined below, is less than 0.5 or a preset number of networks are constructed.

[15] 1. Populate a new training set NT_t from the original training data set T using the distribution D_t.

[16] 2. Construct a new network h_t, and train it using NT_t.

[17] 3. Calculate the maximum loss, $L_{\max}$, over the initial training set T where

$$L_{\max} = \sup|h_t(x_i, y) - y_i|, \text{over all } i. \tag{3}$$

[18] 4. Calculate the individual loss for each element in the initial training set as follows:

$$L_i = 1 - \exp\left[-\frac{|h_t(x_i, y) - y_i|}{L_{\max}}\right]. \tag{4}$$

[19] 5. Calculate the weighted average loss:

$$\overline{L} = \sum_{i=1}^{m} L_i D_t(i) \tag{5}$$

[20] 6. Set

$$\beta_t = \overline{L}/_{1-\overline{L}}. \tag{6}$$

[21] 7. Update the distribution D_t:

$$D_{t+1}(i) = \frac{D_t(i)\beta_t^{(1-L_i)}}{Z_t}, \tag{7}$$

where Z_t is a normalization factor chosen such that D_{t+1} is a distribution.

[22] 8. $t = t + 1$.

[23] *Drucker* [1997, 1999] showed that this algorithm was, in most cases, better than bagging in terms of prediction error when applied to ANNs.

2.2. Methods for Combining ANNs

[24] Once a set of ANNs has been created, an effective way of combining different network outputs must be found. *Ahmad and Zhang* [2002] provide a review of methods for combining member networks. The two most commonly used methods are linear combination and stacked generalization.

2.2.1. Averaging and Weighted Averaging

[25] Linear combination of the outputs of ensemble members is one of the most popular approaches for combining network outputs. A single output can be created from the outputs of a set of networks via simple averaging, or a weighted average that considers the relative performance of each network. Suppose we derived K individual networks using the bagging or boosting procedure and the ith pattern has an observed value y_i and a predicted value $\hat{y}_i^k$ obtained from the kth network. Combining using simple averaging is defined as

$$\hat{y}_i = \frac{1}{K}\sum_{k=1}^{K} \hat{y}_i^k, \qquad i = 1 \ldots m. \tag{8}$$

This method of combination is easy to implement and can lead to improved performance [*Perrone and Cooper*, 1993; *Bishop*, 1995].

2.2.2. Stacking

[26] Under stacking [*Wolpert*, 1992] an additional model is used to learn how to combine the networks by tuning its weights over the feature space. The outputs from a set of level 0 generalizers, which are the ensemble members, are fed to a level 1 generalizer, which is trained to produce appropriate output. This method has been reported by *Drucker* [1997] and *Hu and Tsoukalas* [2003] to be successful in improving the generalization ability of ANNs. The stacking algorithm developed by *Breiman* [1996b] suggests minimizing the following function:

$$W = \sum_{i=1}^{m}\left[y_i - \sum_{k=1}^{K} c_k \hat{y}_i^k\right]^2 \qquad c_k > 0. \tag{9}$$

This algorithm produces estimates for the coefficients $\hat{c}_1$, $\hat{c}_2$, $\ldots$, $\hat{c}_K$, which are used to construct the ensemble prediction:

$$\hat{y}_i = \sum_{k=1}^{K} c_k \hat{y}_i^k, \qquad i = 1 \ldots m. \tag{10}$$

Equation (9) minimizes squared absolute differences between observed and predicted values. This process, however, when used to determine the coefficients, may be dominated by those patterns with a large error. A better

choice, as adopted in this study, is to minimize the (squared) relative difference:

$$V = \sum_{i=1}^{m} \left[\frac{y_i - \sum_{k=1}^{K} c_k \hat{y}_i^k}{y_i} \right]^2 \qquad c_k > 0. \tag{11}$$

3. Regression-Based Methods for Pooled Flood Frequency Analysis

3.1. Parametric Regression Methods

[27] Parametric regression methods have been widely used in pooled flood frequency analysis. Unlike nonparametric regression approaches, such as ANNs, parametric regression methods require an assumption regarding the underlying model. The most commonly used model to describe the relationship between the flood quantile Q_T and the catchment characteristics (x_1, x_2, ..., x_n) is the power-form function [*Thomas and Benson*, 1970]

$$Q_T = a x_1^{\theta_1} x_2^{\theta_2} x_3^{\theta_3} \cdots x_n^{\theta_n}, \tag{12}$$

where θ_i is the ith model parameter, a is the multiplicative error term, and n is the number of catchment characteristics.

[28] Parametric estimation of equation (12) can be divided into linear and nonlinear regression. Applying a linear regression technique, such as ordinary least squares (OLS), requires linearizing the power-form model by a logarithmic transformation. The estimation of the linearized model is theoretically unbiased in the logarithmic domain, but will be biased in the real flow domain [*McCuen et al.*, 1990]. There are methods available for eliminating the bias by adjusting the intercept term; however, these methods are sensitive to the underlying normality assumption [*Koch and Smillie*, 1986]. Nonlinear regression methods directly optimize the model parameters by minimizing the estimation error in the actual flow domain. *Pandey and Nguyen* [1999] and *Grover et al.* [2002] provide reviews of commonly used linear and nonlinear regression methods and compare the relative performance of each method for the application in flood quantile and index flood estimation, respectively. Both groups concluded that nonlinear regression, with a properly selected objective function, provides more accurate estimates than linear regression. In the present paper, ordinary least squares regression (REG_OLS) and nonlinear regression (REG_NONLINEAR) are compared with the ANN models. The objective function of the nonlinear regression is selected to minimize the relative difference between the observed and predicted flow as suggested by *Grover et al.* [2002],

$$NL\_SABS = \sum_{i=1}^{m} abs\left(\frac{Q_{T,i} - \hat{Q}_{T,i}}{Q_{T,i}} \right), \tag{13}$$

where $Q_{T,i}$ and $\hat{Q}_{T,i}$ are observed and predicted flood statistics, respectively, at site i.

3.2. The Single ANN Model

[29] The multilayer perceptron (MLP) is chosen as the base ANN model forming an ensemble. A major advantage of MLPs is that given sufficient hidden units and enough data, they can learn, to any accuracy, a response relationship of arbitrary complexity between input and output variables. In other words, they are universal approximators. The introduction and statistical interpretation of the MLP can be found in major texts, such as *Bishop* [1995] and *Ripley* [1996]. *Reed and Marks* [1999] provide a general review of MLP training algorithms and practical hints for improving network generalization.

[30] An MLP having one output node, one hidden layer, and one input layer is used in this study. The system input will be a set of catchment descriptors that may explain the catchment flood generation mechanisms. The output of the system will be the flood quantile or the index flood. The neurons in the hidden layer use the tan-sigmoid transfer function, while a linear transfer function is adopted for neurons in the output layer. Choosing the linear transfer function for the output units has the benefit of a potentially unbounded output to match the characteristics of the actual flood quantile.

[31] Methods for designing and training an MLP can be easily obtained from general ANN texts. A number of points that are unique to the present implementation are explained below.

3.2.1. Data Preprocessing

[32] Appropriate data preprocessing is necessary for the input and output data to ANNs. In all cases, inputs fed to ANNs are normalized to zero mean and unit standard deviation and the output uses a logarithmic transformation. Most commonly used error functions depend solely on the difference between an observed value y and a predicted value $\hat{y}$. If a linear transformation is used for the output, the ANN will concentrate the effort on learning patterns for which $|y - \hat{y}|$ is large instead of treating each pattern to be of equivalent importance. This problem can be worse if $\hat{y}$ is very noisy. By taking logarithms, these problems can be mitigated, since a difference between two logarithmic transformed values measures the ratio of the original values. Using a logarithmic transformation of the output can have approximately the same effect as dealing with the relative difference $|(y - \hat{y})/y|$ or $|(y - \hat{y})/\hat{y}|$, rather than simple differences [*Sarle*, 2003].

3.2.2. Training Algorithm

[33] The basic back propagation algorithm for training an MLP adjusts the weights using gradient descent. A drawback of this algorithm is that it needs a long learning time and is affected by local minima. A more efficient algorithm, called the Levenberg-Marquardt (LM) algorithm [*Hagan and Menhaj*, 1994], is used in this study. This algorithm has the capability of finding optimal solutions for a variety of problems and is much faster than the gradient descent method. Implementing the algorithm requires the scalar parameter μ [*Demuth and Beale*, 2003]. The initial value of μ is set as 0.005. This value is multiplied by $\mu_d = 0.1$ whenever the performance function is reduced by a step and is multiplied by $\mu_i = 10$ whenever a step would increase the performance function. If μ becomes larger than $\mu_{max} = 1 \times 10^6$, the algorithm is stopped. Alternatively, if the training epochs exceed the maximum training epochs, 300, training will stop.

3.2.3. Improving Generalization Ability

[34] A common problem during ANN training is poor generalization that may occur due to underfitting or

overfitting. In the case of underfitting, poor generalization may be due to insufficient training time or an insufficient number of hidden units. The problem may be overcome by increasing the number of training iterations or using a different network configuration. In the case of overfitting, the prediction error on the training set has a very small value; however, when new data are presented to the network, the error is large. Two effective approaches to improve generalization are regularization and early stopping [*Bishop*, 1995]. With early stopping, a separate validation set needs to be taken from the training set, which can be problematic in the case of flood frequency analysis where the available data sets tend to be small. Furthermore, how to optimally select the validation set and when to stop training still remain as major difficulties in the application of this method. Regularization, which is less affected by these problems, is used in this study.

[35] The idea behind regularization is to modify the performance function to generate a network with smaller weights and biases, and thus a smoother response that is less likely to result in overfitting. Suppose the performance function is chosen to be the mean of the sum of squares of the network errors, mse, on the training set. By adding a term msw that consists of the mean of the sum of squares of the network weights and biases, the modified performance function can be expressed as

$$mse\_reg = \lambda \times mse + (1 - \lambda) \times msw, \tag{14}$$

where λ is the regularization parameter. The success of regularization depends on the choice of an appropriate value of λ. This study adopts the method by *MacKay* [1992], which determines the optimal λ automatically in a Bayesian framework.

3.3. The Ensemble Approaches

[36] The general methods for creating ANN ensemble were introduced in section 2. Several approaches formed by variations and/or combinations of the methods will be applied to compare their effectiveness in solving the regression problem of flood statistic estimation. Specifically, the following approaches are adopted:

[37] 1. In the basic network ensemble with simple averaging (NN_BASIC_MEAN) method, each ANN is trained using the original training set and thus only differs from the other networks in the ensemble by its random initial weights. Trained ANNs are combined using simple averaging to generate the ensemble output. This method is included because several researchers [*Agrafiotis et al.*, 2002; *Opitz and Maclin*, 1999] found that this simple method may produce results as accurate as the more complex bagging and boosting methods.

[38] 2. The basic network ensemble with stacking (NN_BASIC_STACK) method uses the same method as NN_BASIC_MEAN to generate ensemble members; however, the combination of the member networks uses the stacking algorithm.

[39] 3. In the bagging with simple averaging (NN_BAG_MEAN) method, each ANN in the ensemble is trained using a randomly sampled training set. Trained ANNs are combined using simple averaging to generate the ensemble output.

[40] 4. The bagging with stacking (NN_BAG_STACK) method uses the same method as NN_BAG_MEAN to generate ensemble members; however, the combination of the member networks uses the stacking algorithm.

[41] 5. In the Adaboost.R2 (NN_BOOST_MEDIAN) method, the Adaboost.R2 algorithm is implemented to generate a series of predictors. The cumulative output is the weighted median of the predictors, as suggested by *Drucker* [1997]. The following is the procedure for calculating the cumulative output: For a particular input x_i, each ensemble member makes a prediction h_t, $t = 1, \ldots, T$. Each h_t is associated with β_t calculated by equation (6). To get the ensemble output, the predictions are relabeled such that $h_1 < h_2 < \ldots < h_T$ and β_t are sorted to retain the association of each h_t. Then the sorted β_t are summed until the smallest $t = t_s$ is found such that the following inequality

$$\sum_{t=1}^{t_s} \log\left(\frac{1}{\beta_t}\right) \geq \frac{1}{2}\sum_{t=1}^{T} \log\left(\frac{1}{\beta_t}\right) \tag{15}$$

is satisfied, and the ensemble output for the ith input is h_{t_s}.

[42] 6. The Adaboost.R2 with stacking (NN_BOOST_STACK) method uses the same method as NN_BOOST_MEDIAN to generate ensemble members; however, the combination of the member networks uses the stacking algorithm.

4. Application

4.1. Data Sets

[43] The study region includes catchments from England, Scotland, and Wales, in the United Kingdom. The United Kingdom is part of a climate region for which flooding arises mostly from frontal cyclonic precipitation [*Hayden*, 1988].

[44] The data for flood frequency analysis used in this study were obtained from the *Flood Estimation Handbook* CD-ROM [*Institute of Hydrology*, 1999]. Catchments meeting the following six criteria were selected for the study: (1) Catchments must be essentially rural; (2) there could be no significant reservoir and lake attenuation effect for the flood events for the catchments; (3) catchment size is in the interval of 10–1000 km^2; (4) an average of four to five peaks-over-threshold (POT) events per year were required for the catchment flood record; (5) catchment records were required to have minimal gaps in the data record; and (6) catchments must have a minimum record length of 10 years. According to these criteria, 404 catchments were selected. For a detailed description of the data sets, refer to *Cunderlik and Burn* [2002]. The location of the catchment centroids in the study region can be seen in Figure 1. The drainage area statistics of the catchments are summarized in Table 1.

[45] The index flood and one representative quantile (the 10-year flood quantile) are selected for this study. The 10-year flood quantile is selected because it is often of concern from a design perspective and is also a quantile that can be estimated reliably from the length of data record available for the catchments in the database. For each site of the study region the index flood is estimated as the mean of the annual maximum flood series and the 10-year flood quantile is estimated assum-

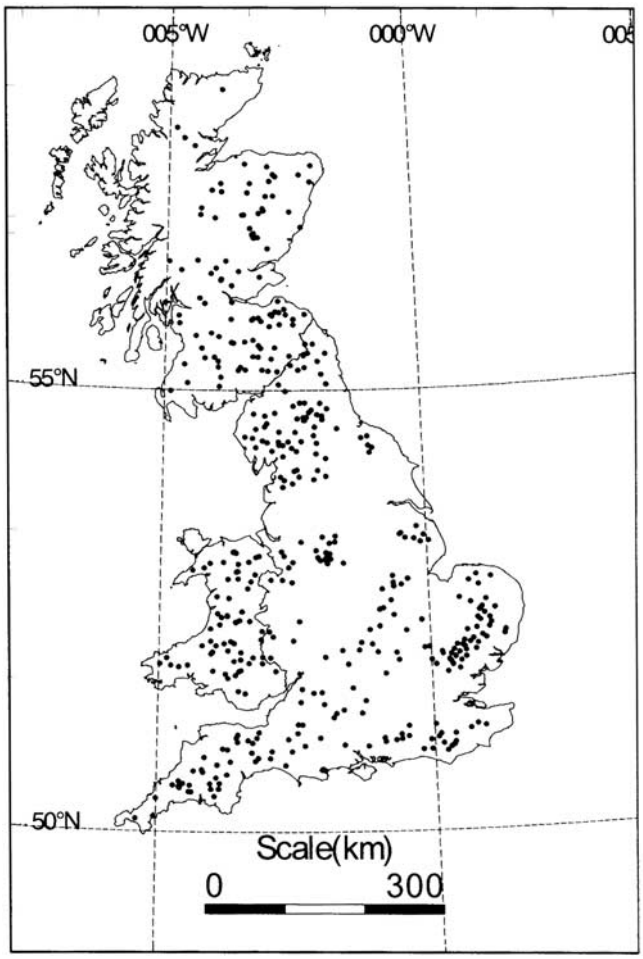

Figure 1. Location of catchments used in this study.

ing a generalized-logistic distribution according to recommendations in the *Flood Estimation Handbook* [*Institute of Hydrology*, 1999]. Variables used to estimate the index flood and the flood quantile are selected according to both hydrological criteria and statistical criteria. The objective of the selection is to get a relatively small set of variables that provides a good statistical fit and gives a sound hydrological model [*Institute of Hydrology*, 1999]. Five catchment characteristics representing catchment drainage area (AREA), standard average annual rainfall (SAAR), soil drainage type (represented by SPRHOST and BFIHOST), and reservoir/lake effects (FARL) are selected according to these criteria as reported by the *Institute of Hydrology* [1999]. AREA [km^2] is the catchment drainage area based on drainage boundaries defined by a digital terrain model. SAAR [mm] is the average annual rainfall for the standard period 1961–1990. SPRHOST [%] represents standard percentage runoff (SPR) derived from HOST (hydrology of soil types) classification. BFIHOST represents the base flow index (BFI) determined from the HOST classification. It is a nondimensional number that ranges from 0 to 1. SPRHOST and BFIHOST for each catchment are estimated using the HOST classification and an area-weighting method. FARL is the index of flood attenuation due to reservoirs or lakes. FARL ranges from 0 to 1, and a value close to 1 means there is little flood attenuation due to reservoirs and lakes. Summary statistics of these variables are shown in Table 2.

4.2. Evaluation Methodology

[46] The index flood and flood quantile estimates from the resulting models are assessed using three indices: the relative squared error (RSError), the percent relative error (PRError), and the relative bias (RBias). The equations for these indices are as follows:

$$RSError = \frac{1}{n}\sum_{1}^{n}\left(\frac{Q-\hat{Q}}{Q}\right)^2, \tag{16}$$

$$PRError = \frac{1}{n}\sum_{1}^{n} abs\left(\frac{Q-\hat{Q}}{Q}\right)\times 100, \tag{17}$$

$$RBias = \frac{1}{n}\sum_{1}^{n}\left(\frac{Q-\hat{Q}}{Q}\right), \tag{18}$$

where n is the number of catchments in the prediction set and $\hat{Q}$ and Q are the predicted and the observed flood statistic, respectively, for site i. Both RSError and PRError describe the error in the predictive ability of the model, and RBias describes whether the model tends to overpredict or underpredict. The indices are chosen to be dimensionless to eliminate the influence of the differences in the flow magnitudes for different catchments.

[47] The generalization error is estimated using tenfold cross validation. In tenfold cross validation the available data are divided into 10 subsets of approximately equal size. Each model is then trained 10 times, each time leaving out one of the subsets from the training set, and using only the omitted subset to compute the performance indices. Since ANNs are sensitive to the initial parameter selection, each method will entail a number of cross-validation runs, and the mean and standard deviation of the errors are reported to reflect the true generalization ability of each method.

[48] Since the performance indices used in this study are related to three different aspects of model performance, the overall performance of each model is evaluated using a rank score. This technique was used in previous work in regression-based pooled flood frequency analysis by *Pandey and Nguyen* [1999] and *Grover et al.* [2002]. To calculate the rank score, the models are ranked from the best to the worst according to the performance indices. A score of 1 is assigned for the best model and p is assigned for the worst, supposing there are p models for evaluation. For each model, the scores for the different performance indices are summed to obtain the overall rank score R_o for the model.

Table 1. Area Statistics of Catchments Used in This Study

Area, km^2	Number of Basins	Coefficient of Variation, %
$A < 100$	145	47.08
$100 < A < 300$	164	26.50
$300 < A < 500$	46	15.62
$500 < A < 700$	28	10.60
$A > 700$	21	7.70

Table 2. Summary Statistics for Input Variables

	Minimum	Mean	Maximum	Standard Deviation
AREA, km^2	10.55	224.54	954.90	213.71
SAAR, mm	547.00	1117.87	2846.00	458.27
FARL []	0.90	0.98	1.00	0.02
SPRHOST, %	10.70	38.67	59.90	8.76
BFIHOST []	0.23	0.48	0.90	0.12

Supposing there are q indices, then the overall rank scores are in the range $[q, pq]$. For convenience, the overall rank scores can be transformed to standardized rank scores R_s using

$$R_s = \frac{pq - R_o}{pq - q} \tag{19}$$

such that the standardized rank scores are in the range [0, 1], and an R_s close to 1 is associated with a model with good performance.

4.3. Results and Discussion

4.3.1. Experiment Results

[49] Six approaches were presented in section 3.3 to create ANN ensembles. To make a fair comparison of the performance of these approaches, each ANN composing an ensemble uses the same design and training method as described in section 3.2. A relatively large ensemble size of 20 is used for each ensemble method. The ensemble approaches are also compared with the single ANN (NN_SINGLE), which is constructed the same as the base networks forming the ensembles. The ANNs used for both ensemble approaches and NN_SINGLE are randomly initialized, and 50 runs with independent initialization are tested for each method to reflect their true generalization ability.

[50] All the calculations were carried out on a four-processor Silicon Graphics Origin 200 running the IRIX 6.5 64-bit operating system. Programs were written using the Matlab programming language. The base ANN models were constructed using the Matlab Neural Network Toolbox [*Demuth and Beale*, 2003]. Each ANN is designed and trained using the method described in section 3.2. On the basis of the set of inputs selected, five hidden units are used in the hidden layer. Since there is no widely accepted guideline to determine the best number of hidden units, the initial number of hidden units was based on the guidance of *Berry and Linoff* [1997], who claim the number of hidden units should never be more than twice as large as the number of inputs. The relatively small data size for model calibration tends to require a more compact network structure. Further optimization was carried out based on another tenfold cross-validation run in the training set. By increasing the hidden units from two to ten, the results indicate that five units in the hidden layer achieve sufficient generalization ability.

[51] The performance of the single ANN and the ensembles is summarized in Tables 3 and 4 for the index flood and 10-year flood quantile estimation, respectively. These tables list the mean and the standard deviation (SD) of the performance criteria obtained from 50 independent tenfold cross-validation runs. The modeling errors of each ANN ensemble method were plotted on a map of the study region (not shown), and no notable spatial patterns were observed.

4.3.2. Analysis and Comparison

[52] The following parts of this section start with a comparison between an individual ANN and ANN ensembles. This is followed by an examination of the factors that may affect the generalization ability of an ANN ensemble (i.e., the method used for creating ensemble members and the combination method). Finally, ANN ensembles are compared with parametric regression methods.

4.3.2.1. Single Versus Ensemble

[53] For index flood estimation, the average generalization ability of network ensembles is always better than that of individual predictors, regardless of the method used to construct the ensemble models. The better generalization ability is indicated (see Table 3) by a higher value in the rank score. The standard deviation for the ensembles is generally much lower than those for the single networks. The same conclusion can be reached for the quantile estimation. These results imply that the ensemble approaches are more accurate in flood estimation and less sensitive to the choice of initial parameters than is the case for a single ANN.

4.3.2.2. Randomization Versus Averaging Versus Boosting

[54] When the same combination method is used, boosting consistently outperforms bagging, and bagging is better than the basic ensemble method, as evidenced by the rank score for both index flood and quantile estimation. The improvements are mainly in the index RBias, as there is very little difference in the indices RSError and PRError when any two methods are compared. This means that the model diversity introduced by the basic ensemble is com-

Table 3. Performance Indices for Index Flood Estimation

	RSError		PRError		RBias		
Method	Mean	SD	Mean	SD	Mean	SD	Rank Score
NN_SINGLE	0.4391	0.0103	29.44	0.2962	0.0729	0.0047	0.25
NN_BASIC_MEAN	0.4247	0.0043	28.96	0.1759	0.0722	0.0018	0.50
NN_BASIC_STACK	0.3826	0.0072	27.83	0.2329	−0.0855	0.0031	0.54
NN_BAG_MEAN	0.4281	0.0070	28.80	0.2985	0.0703	0.0021	0.54
NN_BAG_STACK	0.3819	0.0050	27.72	0.2278	−0.0759	0.0021	0.67
NN_BOOST_MEDIAN	0.4103	0.0066	28.74	0.2308	0.0600	0.0022	0.71
NN_BOOST_STACK	0.3663	0.0039	27.48	0.1653	−0.0728	0.0028	0.83
REG_OLS	0.4880	...	33.70	...	0.0892	...	0.00
REG_NONLINEAR	0.4378	...	32.71	...	−0.0586	...	0.46

Table 4. Performance Indices for 10-Year Flood Quantile Estimation

Method	RSError		PRError		RBias		Rank Score
	Mean	SD	Mean	SD	Mean	SD	
NN_SINGLE	0.4819	0.0090	33.46	0.4219	0.0868	0.0051	0.25
NN_BASIC_MEAN	0.4727	0.0047	32.69	0.2565	0.0862	0.0026	0.38
NN_BASIC_STACK	0.4037	0.0061	30.72	0.2862	−0.0905	0.0032	0.54
NN_BAG_MEAN	0.4720	0.0064	32.56	0.2744	0.0853	0.0030	0.50
NN_BAG_STACK	0.3955	0.0076	30.03	0.3265	−0.0880	0.0022	0.75
NN_BOOST_MEDIAN	0.4645	0.0071	31.89	0.1849	0.0789	0.0028	0.71
NN_BOOST_STACK	0.3975	0.0054	30.28	0.2347	−0.0843	0.0026	0.83
REG_OLS	0.5359	...	37.31	...	0.1051	...	0.00
REG_NONLINEAR	0.4447	...	34.29	...	−0.0832	...	0.54

parable with the more sophisticated methods of bagging and boosting.

4.3.2.3. Averaging Versus Stacking

[55] For both index flood and quantile estimation, the combination using stacking is better than simple averaging when the same method for creating ensemble members is used. The improvements are mainly in the performance indices RSError and PRError. The flood statistics estimated using stacking are generally underestimated, while simple averaging tends to overestimate the values.

4.3.2.4. ANN Versus Parametric Regression

[56] Since parametric regression methods are the most commonly used methods in the regression-based flood frequency analysis, a comparison between ANNs and parametric regression is included. Tables 3 and 4 list the performance indices for the REG_OLS and REG_NONLINEAR methods in the last two rows. All the ANN models outperform linear regression for all error criteria. Nonlinear regression shows better generalization ability than the single ANN. However, most ANN ensembles, especially those produced by stacking, consistently outperform nonlinear regression. Improvements obtained from ANN ensembles are mainly from the reduction in the magnitude of prediction error, which, however, is generally accompanied by an increase in the bias term, RBias.

4.3.3. Ensemble Size

[57] *Hansen and Salamon* [1990] suggested that ensembles with 10 members are adequate to reduce model classification error, a result confirmed by *Agrafiotis et al.* [2002]. An empirical study by *Opitz and Maclin* [1999] showed that for both bagging and boosting, significant reduction in error occurred after 10–15 classifiers. Since most of these studies are based on classification problems, a similar experiment was conducted for our regression problem to examine the performance of each ensemble method for different ensemble sizes. As an example, the performance of NN_BAG_MEAN and NN_BAG_STACK for index flood estimation using up to 20 networks is presented in Figures 2 and 3, respectively. After the first few estimators are added, there is a substantial reduction in prediction error. However, the bias term, RBias, for the stacking combination tends to increase as the number of networks increases. The stacking combination involves a level 1 learning, which essentially tries to reduce the prediction error, which is accompanied by an increase in the bias. A better approach may be to use a multiobjective function to minimize both error and bias in the level 1 learning. The performance of the ensembles reaches the best point, according to most performance indices, when the ensemble size is 14. This suggests that the performance indices for the ANN ensembles presented in Tables 3 and 4 could be further improved by reducing the number of networks. Figures 2 and 3 also indicate that the improvement in the generalization ability of the ensemble models after size 10 is actually very small. Similar patterns were observed for the other ensemble methods. Therefore an ensemble size of at least 10 is recommended, based on the results of this study.

5. Conclusions

[58] Factors that may affect the generalization of an ANN ensemble are analyzed and compared. In terms of the methods for creating ensemble members, the model diversity introduced by varying the initial conditions of the base ANNs to reduce the prediction error is comparable with more sophisticated methods such as bagging and boosting. For both index flood and quantile estimation, combination using stacking is generally better than simple averaging when the same method for creating ensemble members is

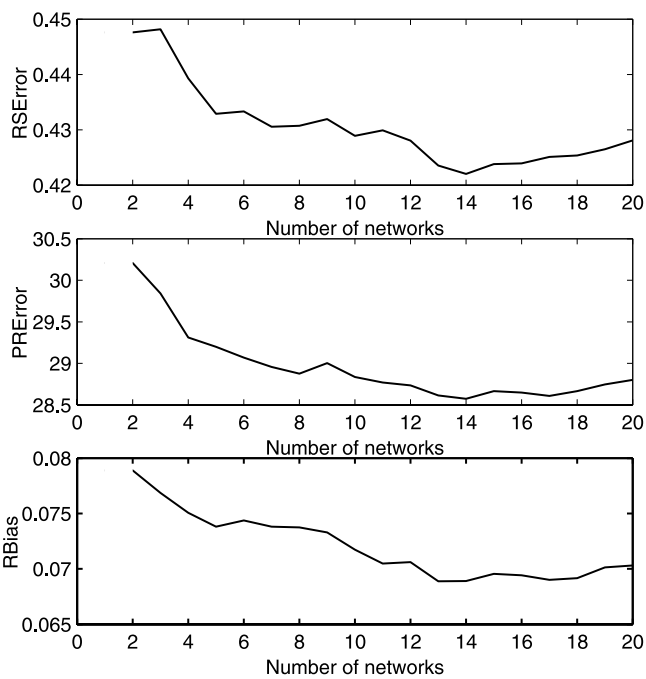

Figure 2. Performance indices versus the number of nets in the ensemble for the NN_BAG_MEAN method.

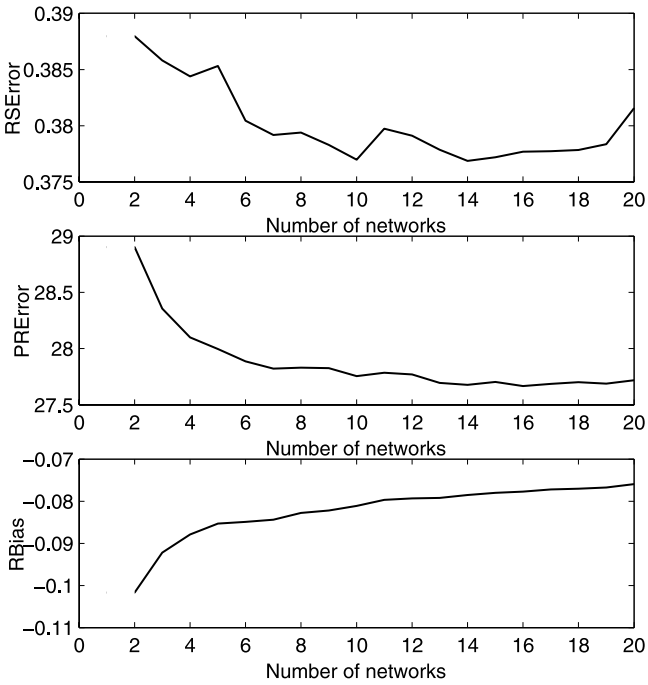

Figure 3. Performance indices versus the number of nets in the ensemble for the NN_BAG_STACK method.

used. However, flood statistics are generally underestimated when stacking is used and overestimated when simple averaging is used. At least 10 networks in an ANN ensemble are suggested to achieve sufficient generalization ability.

[59] The comparison between ANN ensembles and a single ANN shows that ANN ensembles are more accurate in flood estimation and less sensitive to the choice of initial parameters than is the case for a single ANN.

[60] The ANN models proposed in this study were also compared with two parametric regression methods. The nonlinearity introduced by the ANN models allows them to outperform the multiple linear regression method. Most ANN ensembles, especially those produced by stacking, consistently outperform nonlinear regression. Improvements obtained from ANN ensembles are mainly from the reduction in the magnitude of prediction error, which is generally accompanied by an increase in the bias term, RBias.

[61] Properly designed ensemble modeling can significantly improve the generalization ability and should be more widely adopted by the hydrological modeling community. Ensemble approaches generally result in a much heavier computational load than for single ANNs, although with the increases in computing power, the cost is not prohibitively high.

[62] **Acknowledgments.** This research was supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC). The paper has been improved by helpful comments from the associate editor and an anonymous reviewer.

References

Abrahart, R. J., and L. See (2002), Multi-model data fusion for river flow forecasting: An evaluation of six alternative methods based on two contrasting catchments, *Hydrol. Earth Syst. Sci.*, *6*(4), 655–670.

Agrafiotis, D. K., W. Cedeño, and V. S. Lobanov (2002), On the use of neural network ensembles in QSAR and QSPR, *J. Chem. Inf. Comput. Sci.*, *42*, 903–911.

Ahmad, Z., and J. Zhang (2002), A comparison of different methods for combining multiple neural networks models, in *Proceedings of the 2002 World Congress on Computational Intelligence*, pp. 12–117, IEEE Press, Piscataway, N. J.

Berry, M. J. A., and G. Linoff (1997), *Data Mining Techniques*, John Wiley, Hoboken, N. J.

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford Univ. Press, New York.

Breiman, L. (1996a), Bagging predictors, *Mach. Learning*, *24*, 123–140.

Breiman, L. (1996b), Stacked regressions, *Mach. Learning*, *24*, 49–64.

Cannon, A. J., and P. H. Whitfield (2002), Downscaling recent streamflow conditions in British Columbia, Canada, using ensemble neural network models, *J. Hydrol.*, *259*(1–4), 136–151.

Carney, J. G., and P. Cunningham (1999), The NeuralBAG algorithm: Optimizing generalization performance in bagged neural networks, in *Proceedings of the 7th European Symposium on Artificial Neural Networks*, pp. 35–40, D-Facto Publ., Brussels, Belgium.

Cunderlik, J. M., and D. H. Burn (2002), The use of flood regime information in regional flood frequency analysis, *Hydrol. Sci. J.*, *47*, 77–92.

Dalrymple, T. (1960), Flood-frequency analyses, *U.S. Geol. Surv. Water Supply Pap.*, *1543-A*, 75 pp.

Demuth, H., and M. Beale (2003), *Matlab Neural Network Toolbox*, *Version 4*, Math Works, Natick, Mass.

Dietterich, T. G. (2000), An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Mach. Learning*, *40*(2), 139–157.

Drucker, H. (1997), Improving regressors using boosting techniques, in *Machine Learning: Proceedings of the Fourteenth International Conference*, pp. 107–115, Morgan Kaufmann, Burlington, Mass.

Drucker, H. (1999), Boosting using neural networks, in *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, edited by A. J. C. Sharkey, pp. 51–78, Springer-Verlag, New York.

Efron, B., and T. J. Tibshirani (1993), *An Introduction to the Bootstrap*, Chapman and Hall, New York.

Freund, Y., and R. E. Schapire (1996), Experiments with a new boosting algorithm, in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148–156, Morgan Kaufmann, Burlington, Mass.

Govindaraju, R. S., and A. R. Rao (Eds.) (2000), *Artificial Neural Networks in Hydrology*, 329 pp., Kluwer Acad., Norwell, Mass.

Grover, P. L., D. H. Burn, and J. M. Cunderlik (2002), A comparison of index flood estimation procedures for ungauged catchments, *Can. J. Civ. Eng.*, *29*(5), 734–741.

Hagan, M. T., and M. Menhaj (1994), Training feedforward networks with the Marquardt algorithm, *IEEE Trans. Neural Networks*, *5*(6), 989–993.

Hansen, L., and P. Salamon (1990), Neural network ensembles, *IEEE Trans. Pattern Anal. Mach. Intell.*, *12*, 993–1001.

Hayden, B. P. (1988), Flood climates, in *Flood Geomorphology*, edited by V. R. Baker et al., pp. 13–27, John Wiley, Hoboken, N. J.

Hosking, J. R. M., and J. R. Wallis (1997), *Regional Frequency Analysis: An Approach Based on L-Moments*, Cambridge Univ. Press, New York.

Hu, M. Y., and C. Tsoukalas (2003), Explaining consumer choice through neural networks: The stacked generalization approach, *Eur. J. Oper. Res.*, *146*(3), 650–660.

Institute of Hydrology (1999), *Flood Estimation Handbook*, Wallingford, UK.

Koch, R. W., and G. M. Smillie (1986), Bias in hydrologic prediction using log-transformed regression models, *Water Resour. Bull.*, *22*(5), 717–723.

Krogh, A., and J. Vedelsby (1995), Neural network ensembles, cross validation and active learning, in *Advances in Neural Information Processing Systems*, edited by D. S. Touretzky, G. Tesauro, and T. K. Leen, vol. 7, pp. 231–238, MIT Press, Cambridge, Mass.

MacKay, D. J. C. (1992), Bayesian methods for adaptive models, Ph.D. thesis, Calif. Inst. of Technol., Pasadena.

McCuen, R. H., R. B. Leahy, and P. A. Johnson (1990), Problems with logarithmic transformations in regression, *J. Hydraul. Eng.*, *116*(3), 414–428.

Opitz, D., and R. Maclin (1999), Popular ensemble methods: An empirical study, *J. Artif. Intell. Res.*, *11*, 169–198.

Pandey, G. R., and V.-T.-V. Nguyen (1999), A comparative study of regression based methods in regional flood frequency analysis, *J. Hydrol.*, *225*, 92–101.

Perrone, M. P., and L. N. Cooper (1993), When networks disagree: Ensemble methods for hybrid neural networks, in *Artificial Neural Networks for Speech and Vision*, edited by R. J. Mammone, pp. 126–142, Chapman and Hall, New York.

Reed, R. D., and R. J. Marks (1999), *Neural Smithing*, MIT Press, Cambridge, Mass.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge Univ. Press, New York.

Sarle, W. S. (Ed.) (2003), Neural Network FAQ, Part 2, Monthly posting to the Usenet newsgroup comp.ai.neural-nets. (Available at URL: ftp://ftp.sas.com/pub/neural/FAQ.html.)

Schapire, R. E. (1990), The strength of weak learnability, *Mach. Learning*, *5*, 197–227.

See, L., and R. J. Abrahart (2001), Multi-model data fusion for hydrological forecasting, *Comput. Geosci.*, *27*, 987–994.

Sharkey, A. J. C. (Ed.) (1999), *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, Springer-Verlag, New York.

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

Thomas, D. M., and M. A. Benson (1970), Generalization of streamflow characteristics from drainage-basin characteristics, *U.S. Geol. Surv. Water Supply Pap.*, *1975*, 55 pp.

Wolpert, D. H. (1992), Stacked generalization, *Neural Networks*, *5*, 241–259.

Zhang, J. (1999), Developing robust non-linear models through bootstrap aggregated neural networks, *Neurocomputing*, *25*(1–3), 93–113.

Zrinji, Z., and D. H. Burn (1994), Flood frequency analysis for ungauged sites using a region of influence approach, *J. Hydrol.*, *153*, 1–21.

D. H. Burn and C. Shu, Department of Civil Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. (dhburn@civmail.uwaterloo.ca)