Optimal division of data for neural network models in water resources applications

Gavin J. Bowden, Holger R. Maier, and Graeme C. Dandy

Centre for Applied Modelling in Water Engineering, Department of Civil and Environmental Engineering, Adelaide University, Adelaide, South Australia, Australia

Received 28 July 2000; revised 7 March 2001; accepted 27 June 2001; published 3 February 2002.

[1] The way that available data are divided into training, testing, and validation subsets can have a significant influence on the performance of an artificial neural network (ANN). Despite numerous studies, no systematic approach has been developed for the optimal division of data for ANN models. This paper presents two methodologies for dividing data into representative subsets, namely, a genetic algorithm (GA) and a self-organizing map (SOM). These two methods are compared with the conventional approach commonly used in the literature, which involves an arbitrary division of the data. A case study is presented in which ANN models developed using each data division technique are used to forecast salinity in the River Murray at Murray Bridge (South Australia) 14 days in advance. When tested on a validation data set from July 1992 to March 1998, the models developed using the GA and SOM data division techniques resulted in a reduction in RMS error of 24.2% and 9.9%, respectively, over the conventional data division method. It was found that a SOM could be used to diagnose why an ANN model has performed poorly, given that the poor performance is primarily related to the data themselves and not the choice of the ANN's parameters or architecture. *INDEX TERMS:* 1871 Hydrology: Surface water quality; 1899 Hydrology: General or miscellaneous; 3210 Mathematical Geophysics: Modeling; 3299 Mathematical Geophysics: General or miscellaneous; *KEYWORDS:* artificial neural network, data division, self-organizing map, genetic algorithm, forecasting, salinity model

1. Introduction

[2] Artificial neural network (ANN) models are highly flexible function approximators that have shown their utility in a broad range of water resources applications [e.g., *American Society of Civil Engineers (ASCE) Task Committee on Application of Artificial Neural Networks in Hydrology*, 2000; *Maier and Dandy*, 2000]. While such flexibility provides a powerful tool for forecasting and prediction, there is no established methodology for the design and successful implementation of ANNs. In a review of 43 papers on the use of ANNs for the prediction or forecasting of water resources variables, *Maier and Dandy* [2000] found that in most cases, the development of ANN models was either described poorly or carried out incorrectly. They identified that future research efforts should be directed toward the development of guidelines and modeling methodologies that assist in the development of ANN models. One of the main areas that needs to be addressed is the issue of data division.

[3] When cross validation is used as a stopping criterion, three data sets are needed, namely, training, testing, and validation sets. Three data sets are also required when optimizing network architecture or internal model parameters such as the learning rate or momentum. As pointed out by *Maier and Dandy* [2000], the validation data must not be used in any capacity in the model development process. The training data are used to find an optimal set of connection weights, the test data are used to choose the best network configuration, and once an optimal network has been found, a validation set is required in order to test the true generalization ability of the model.

[4] Recent studies have found that the way the available data are divided into subsets can have a significant influence on an ANN's performance [*Maier and Dandy*, 1996; *Tokar and Johnson*, 1999]. This is because ANNs are typically unable to extrapolate beyond the range of the data used for training [*Flood and Kartam*, 1994; *Minns and Hall*, 1996; *Tokar and Johnson*, 1999]. For adequate generalization ability the training and validation sets must therefore be representative of the same population [*Masters*, 1993]. However, *Maier and Dandy* [2000] found that in most of the papers they reviewed, the data were divided on an arbitrary basis, without any consideration given to their statistical properties. As a consequence, in many cases, the optimality of the results presented was difficult to assess.

[5] *Flood and Kartam* [1994] point out that the number of training samples can significantly influence a network's performance. Increasing the number of training samples provides more information about the shape of the solution surface or surfaces and thus increases the potential level of accuracy that can be achieved by the network. However, in most practical circumstances, data availability and cost impose obvious limitations on the amount of data available and hence on the size of the training set. Thus, the proportion of samples to include in each of the subsets is an important consideration.

[6] It has been acknowledged in the past that an ANN is susceptible to becoming "...a prisoner of its training data" [*Minns and Hall*, 1996]. During prediction, the model is likely to perform poorly if faced with inputs that are far removed from the examples that it saw during training. By using the widest limits of examples during training, it is possible to prevent an ANN from the need to extrapolate [*ASCE Task Committee*, 2000]. Consequently, which samples to include in the training set is also very important.

[7] Recently, attempts have been made to ensure that the statistical properties of each subset of data are similar. *Braddock et al.* [1998] selected 3 years of data for the acceptance (validation) set so that the mean, standard deviation, and range of these years contained the top, bottom, and middle values of the available data.

0043-1397/02/2001WR000266$09.00

The training and testing sets were selected so that they were statistically representative of the entire data set. *Campolo et al.* [1999] divided daily data from January 1992 to August 1993 into training, testing, and validation sets, taking care that the mean, variance, maximum, and minimum values of each subset were most similar. The proportion of data in each subset was assigned arbitrarily, with the training, testing, and validation subsets consisting of 400, 120, and 81 samples, respectively. While this methodology determines which samples should be used in each of the subsets, the authors provide no information on how to ensure the similarity of the statistics. In addition, it does not solve the problem of choosing what proportion of data to use in each subset.

[8] *Tokar and Johnson* [1999] created rainfall-runoff ANN models using wet-, dry-, and average-year data to illustrate the impact of the content of the training data on network prediction accuracy. It was observed that the wet-year models outperformed the dry- and average-year models based on goodness-of-fit statistics. The reason given was that the wet-year models included information on both high- and low-flow conditions. Consequently, it was more likely that in order to predict the patterns in the test set, an interpolation rather than an extrapolation was required.

[9] *Ray and Klindworth* [2000] recognized that the training and testing sets must be representative of the same population. To achieve this, they made their training sets large enough to represent the full population, and data for the testing set were randomly selected and manually checked to ensure that they had characteristics similar to the training data. No information was provided detailing how this was achieved.

[10] Despite these studies, no systematic approach has been developed for the optimal division of data for ANN models. Recently, the *ASCE Task Committee* [2000] discussed future avenues of ANN research and application and noted the important issue of data division. The lack of an established procedure led the Task Committee to pose the following question: Can an optimal training set be identified? The ASCE Task Committee observed that an optimal training set would adequately represent the modeling domain but at the same time would employ the minimum number of samples for achieving this objective. Repetitive data only serve to slow down ANN training, and by keeping the training set concise, software and hardware constraints can be satisfied more easily [*Dawson and Wilby*, 1998]. The ASCE Task Committee elaborated further, stating, ''Very often we may have no alternative but to proceed with limited data. Under these circumstances can we say when generalization will fail so that we understand the range of applicability of the ANN?'' [*ASCE Task Committee*, 2000, p. 135]. Conversely, it may be equally important to ask, When a model does perform poorly, are tools available to critically dissect the model and determine if the data sets were in fact representative of the same population?

[11] The first objective of this paper is to present a methodology that can be used to determine the optimal division of data for ANN models. Two important considerations shall be addressed: (1) What proportion of the data should be used for each of the training, testing, and validation sets? (2) Which samples should be used in each of the sets?

[12] The second objective is to present a technique that can be used when data are limited, to diagnose why an ANN model has performed poorly.

[13] Two new methods for the optimal division of available data will be explored in this paper, and these methods will be compared with a more conventional approach. The first method (section 2.1) employs a genetic algorithm (GA) to divide the data so as to minimize the statistical difference (as measured by the mean and standard deviation) between training, testing, and validation data sets. The second data division method (section 2.2) employs a self-organizing map (SOM) [*Kohonen*, 1982], to cluster similar data records together. An equal number of data records can then be sampled from each cluster to produce training, testing, and validation sets with similar statistical properties while using a minimum number of data records. The conventional approach to be used for comparison simply involves arbitrarily dividing the data into subsets without consideration of the statistical properties. This is the approach most often employed in the literature.

[14] A case study is presented in section 3. The case study involves the development of ANN models (section 4) which are used to forecast salinity in the River Murray at Murray Bridge, South Australia. The results produced by each data division technique are presented and discussed in section 5, and the conclusions of the study are given in section 6.

2. Methods

2.1. Data Division Using a Genetic Algorithm

[15] A genetic algorithm is a powerful optimization technique inspired by the principles of natural evolution and selection [*Goldberg*, 1989]. GAs have been widely used in optimizing water resources variables [e.g., *Dandy et al.*, 1996; *Simpson et al.*, 1994]. In this study a GA has been applied to the problem of dividing the data into three statistically similar subsets. For example, if there are 60 data samples that must be divided into training, testing, and validation sets consisting of 40, 10, and 10 data samples, respectively, then there are

$$\frac{60!}{40! \times 10! \times 10!} = 7.7 \times 10^{20}$$

ways of arranging the data samples. A GA can be used to search through this large space and determine the optimal arrangement based on an objective function. In this study the aim is to arrange the available data into three statistically similar subsets of fixed size.

[16] The GA used for data division sorts the samples into training, testing, and validation sets by using a set of random numbers. The decision variable governing the arrangement of the data samples is a random number seed, chosen to be in the range [1, 100,000]. This range was selected to provide a reasonable size search space. The GA string therefore consists of a single integer between 1 and 100,000. The random number seed controls the generation of a random sequence of numbers. The random number sequence is placed alongside the data samples, and the contiguous block of data is sorted using these random numbers. In so doing, the data samples are arranged into subsets and the objective function is evaluated. Penalty constraints are added to ensure that the maximum and minimum values of each input and output variable are included in the training set, rather than in the testing or validation sets. As discussed in section 1, training the ANN model on the extreme range of values available removes the need for the network to extrapolate.

[17] To determine the ''fitness'' of each solution, an objective function is required. In this application a suitable objective function to minimize is the sum of the absolute difference in mean and standard deviation values between each pair of the three subsets.

[18] A floating point GA was used, as experiments conducted by *Michalewicz* [1994] and *Wright* [1991] have shown that floating point representation is faster and more consistent run to run, provides higher precision than binary coding, and allows greater freedom to use different mutation and crossover techniques based on the real representation.

[19] The tournament selection scheme was used, where strings are paired randomly and the string with the higher fitness in the pair progresses to the next generation [*Goldberg and Deb*, 1991]. This scheme is referred to as a binary or two-member tournament. Since only half of the strings progress, another tournament is held with another set of random pairs, and the winners make up the other half of the crossover pool for the next generation. Two copies of the best string progress, and no copies of the worst string are replicated. The tournament selection scheme was chosen, as it has comparable growth ratios with other schemes but has a better time complexity than many other selection algorithms [*Goldberg and Deb*, 1991].

[20] In this application the linear crossover operator [*Wright*, 1991] has been used, as it avoids problems associated with real crossover. From the two-parent random number seeds, p_1 and p_2, three new random seeds are generated, namely,

$$\frac{1}{2}p_1 + \frac{1}{2}p_2, \frac{3}{2}p_1 - \frac{1}{2}p_2, -\frac{1}{2}p_1 + \frac{3}{2}p_2.$$

The best two of the three random seeds are then chosen.

[21] Nonuniform mutation [*Michalewicz*, 1994], designed for use with floating point GAs, was used in this procedure. If a mutation is to occur at generation t, then a random number seed $v_k$is mutated to produce v'_k, where

$$v'_k = \begin{cases} v_k + \Delta(t, \mathrm{UB} - v_k) \\ v_k - \Delta(t, v_k - \mathrm{LB}) \end{cases} \begin{matrix} \text{if a random digit} = 0 \\ \text{if a random digit} = 1 \end{matrix} \quad (1)$$

and LB and UB are lower and upper domain bounds of the variable v_k. As described by *Michalewicz* [1994], the function $\Delta(t, y)$ returns a value in the range [0, y] such that the probability of $\Delta(t, y)$ being close to 0 increases as t increases. This causes the operator to uniformly search the space initially (i.e., when t is small), and very locally at later stages, thereby increasing the probability of generating the new number closer to its successor than a random choice. The following function is used:

$$\Delta(t, y) = y\left(1 - r^{(1-(t/T))^b}\right), \quad (2)$$

where r is a random number in the range [0,1], T is the maximum number of generations, and $b$is a system parameter determining the degree of dependency on the iteration number (b = 5 was used in this study).

[22] The data used in this procedure are scaled to the range [0,1]. The reason for this is twofold. First, the scaling prevents the inputs with much larger values from dominating the evolutionary process. Second, scaling to the interval [0,1], enables penalty constraints to be included more easily (i.e., the maximum and minimum values can be identified by the GA as the zeros and ones).

2.2. Data Division Using a Self-Organizing Map (SOM)

[23] The self-organizing map (SOM) was developed by *Kohonen* [1982] and arose from attempts to model the topographically organized maps found in the cortices of the more developed animal brains. The underlying basis behind the development of the SOM was that topologically correct maps can be formed in an n-dimensional array of processing elements (PEs) that did not have this initial ordering to begin with. In this way, input stimuli, which may have many dimensions, can come to be represented by a one- or two-dimensional vector, which preserves the order of the higher-dimensional data [*NeuralWare*, 1998].

[24] The SOM employs a type of learning commonly referred to as competitive, unsupervised, or self-organizing, in which adjacent cells within the network are able to interact and develop adaptively into detectors of a specific input pattern [*Kohonen*, 1990]. The SOM can be considered to be a type of neural network because results have indicated that the adaptive processes utilized in the SOM may be similar to the processes at work within the brain [*Kohonen*, 1990].

[25] The SOM has potential extending beyond its original purpose of modeling biological phenomena. Sorting items into categories of similar objects is a challenging, yet frequent task. The SOM achieves this task by nonlinearly projecting the data onto a lower-dimensional display and by clustering these data. This attribute has been used in a wide number of applications ranging from engineering (including image and signal processing and recognition, telecommunications, process monitoring and control, and robotics) to natural sciences, medicine, humanities, economics, and mathematics [*Kaski et al.*, 1998].

2.2.1. The self-organizing map algorithm. [26] In competitive learning, neurons in the network adapt gradually to become sensitive to different input categories. The SOM network generally consists of two layers, an input layer and a Kohonen layer. The input layer is fully connected to the Kohonen layer, which in most common applications is two-dimensional. None of the PEs in the Kohonen layer is connected to another. The PEs in the Kohonen layer measure the distance of their weights to the input pattern. During the recall phase the Kohonen PE with the minimum distance is the winner and has an output of 1.0, while the other Kohonen PEs have an output of 0.0.

[27] The procedure for determining the winning PE is as follows:

[28] The first step is to determine the extent to which the weights of each PE match the corresponding input pattern. If the input data have N values and are denoted by $X = (x_i; \mathrm{i} = 1, \ldots, N) \in \Re^n$, then each of the M PEs in the Kohonen layer will also have N weight values and can be denoted by $W_{ji} = (w_{ji}; j = 1, \ldots, M; i = 1, \ldots, N) \in \Re^n$. For each of the M Kohonen PEs, the distance, such as the Euclidean distance, is calculated using

$$D_j = \|X - W_j\| = \left[\sum_{i=1}^{N} \left(x_i - w_{ji}\right)^2\right]^{1/2}, \quad j = 1, \ldots, M. \quad (3)$$

The PE with the lowest value of D_j is the winner during recall. During training, a conscience mechanism adjusts the distances to encourage PEs that are not winning with an average frequency and to negatively adjust PEs that are winning at an above average frequency. This mechanism ensures that a uniform data distribution develops in the Kohonen layer. In adjusting the distance, a bias B_j is added to the distance and forms the new adjusted distance D_j. The bias is calculated using

$$B_j = \gamma\left(M \times \left(F_j - 1\right)\right), \quad (4)$$

where γ is a learning coefficient, F_j is the frequency at which the PE j has historically won, and M is the number of PEs in the Kohonen layer. Once B_j and D_j are computed, the adjusted distance D_j can be calculated using

$$D_j' = D_j + B_j. \tag{5}$$

To ensure biological plausibility, lateral interaction with neighboring PEs is enforced by applying arbitrary network structures called neighborhood sets, N_c. Throughout the process, all PEs within the winner's neighborhood set will have their weights updated, while PEs outside of this set are left intact. The width or radius of N_c can be time variable. The updating process to implement this procedure is given by

$$W_j(t+1) = \begin{cases} W_j(t) + \alpha(t)\big(X(t) - W_j(t)\big) & j \in N_c(t) \\ W_j(t) & j \notin N_c(t), \end{cases} \tag{6}$$

where α is a scalar valued adaptation gain $0 < \alpha(t) < 1$ and N_c is the neighborhood set. After the weights have been updated, the next input is presented to the network, and the process continues until convergence has been reached. After successively presenting different inputs to the SOM, the net effect is that the weights reflect the topological relationship that exists within the input data [*Islam and Kothari*, 2000].

2.2.2. The SOM in water resources applications. [29] The use of the SOM in water resources applications has been fairly limited. Applications include the estimation of rainfall rates from infrared satellite and ground surface data [*Hsu et al.*, 1997], the identification of flow regimes in horizontal air-water flow in an experimental pipeline [*Cai et al.*, 1994], and the classification of flood data into classes defined by representative regional catchments [*Hall and Minns*, 1999].

2.2.3. Implementation of the SOM. [30] In this paper the SOM is used to cluster the input and output data into training, testing and validation sets that have similar statistical properties. The SOM is implemented using the Neusciences Neuframe software. To cluster the data, the inputs and their corresponding output are presented to the network as the SOM's inputs. The software default parameters are used for the learning rate, neighborhood size, and number of epochs. The output of the SOM is obtained using a dynamic patterns grid, which shows a dynamic representation of the nodes that are winning each pattern. Each individual cell in the grid represents a node in the Kohonen layer. There is no theoretical principle for determining the optimum size of the Kohonen layer [*Cai et al.*, 1994], and hence the Kohonen layer was kept large enough to ensure that the maximum number of clusters were formed from the training data. Once the clusters are formed, three data records from each cluster are sampled (i.e., one for each of the training, testing, and validation sets). In the instance that a cluster only contains one record, then this record is placed in the training set. If a cluster contains two records, then one record is placed in the training set and the other is placed in the testing set.

3. Case Study

[31] The case study used to demonstrate the effect of different data division techniques is that of forecasting salinity in the River Murray at Murray Bridge, South Australia, 14 days in advance. *Maier and Dandy* [1996] have previously developed ANN models for this case study, and hence it provides a good benchmark for testing the data division techniques.

[32] Adelaide is the capital of South Australia. On average, 35% of its water supply is pumped from the River Murray via two major pipelines. One of these is the Murray Bridge to Onkaparinga pipeline. Water in the River Murray is prone to high levels of salinity. From July 1975 to June 1988, the salinity in the River Murray at Murray Bridge varied between 140 and 820 mg L^{-1} with an average of 400 mg L^{-1} [*Dandy and Crawley*, 1992]. In comparison, the World Health Organisation's maximum desirable level for human consumption is 500 mg L^{-1} [*World Health Organisation*, 1984]. It is estimated that the high salinity levels cause $22 million (U.S. dollars) damage per year to domestic and industrial users [*Dwyer Leslie Pty Ltd.*, 1984].

[33] By forecasting salinity several weeks in advance, pumping policies can be developed such that more water can be pumped at times of low salinity and less water pumped at times of high salinity. *Dandy and Crawley* [1992] have developed an optimization model for obtaining an optimum pumping policy, taking salinity into account. The model shows that the average salinity of the water supplied to Adelaide consumers could be reduced by about 10% if salinity was forecast several weeks in advance and pumping policies were modified accordingly.

[34] In accordance with the ANN modeling conducted by *Maier and Dandy* [1996], a forecasting period of 14 days was chosen, as this is the minimum forecasting length required to enable short-term adjustments to be made to the pumping schedule. ANN models were considered to be a suitable technique for this application because multistep forecasts are required, nonlinear relationships are suspected, and it is difficult to prescribe the exact mathematical relationship between the variables [*Maier and Dandy*, 1996].

[35] *Maier and Dandy* [1996] used daily salinity, flow, and river level data at various locations in the river for the period 1 December 1986 to 30 June 1992. Data from this period and at the same locations were also used in this study. In addition, more recent data for the period 1 July 1992 to 1 April 1998 were used for validation purposes.

4. Model Development

[36] Back-propagation networks were developed using the commercially available software package NeuralWorks Professional II/Plus [*NeuralWare*, 1998]. Unless stated otherwise, the default software parameters were used, since the focus is on evaluating the data division techniques rather than studying the effect of varying the network's parameters. The default values were determined using the experience gained from developing back-propagation networks for a variety of applications [*NeuralWare*, 1998].

4.1. Data Division

[37] Three different data division methods were used to examine their effect on the ANN's performance. In method 1, a conventional data division technique was used, whereby the sets were divided on an arbitrary basis and the statistical properties of the respective data sets were not considered. This approach is consistent with the approach that was used in many papers on the application of ANNs to water resources variables [*Maier and Dandy*, 2000]. In the salinity case study a total of 2005 data records were available, from which 1604 records (80%) were used for calibration and 401 records

Table 1. Method 1 Statistics of the Salinity Training, Testing, and Validation Data Sets (Data Divided Using Conventional Method)

Variable and Data Set	Mean	Standard Deviation	Maximum	Minimun	Interquartile Range
Input 1: Murray Bridge salinity, EC[a] units					
Training	588.0	172.9	931.0	261.3	291.6
Testing	515.7	212.2	862.9	262.3	444.7
Validation	694.7	213.6	1,115.7	282.2	342.1
Input 2: Mannum salinity, EC units					
Training	574.5	165.7	960.4	281.1	281.7
Testing	495.2	205.0	826.6	253.2	438.4
Validation	661.4	192.4	1,075.4	269.7	265.7
Input 3: Morgan salinity, EC units					
Training	572.0	170.0	921.3	176.9	320.2
Testing	498.7	201.9	798.9	258.6	420.4
Validation	670.7	205.6	1,061.3	249.4	307.4
Input 4: Waikerie salinity, EC units					
Training	562.5	160.7	880.3	278.9	289.9
Testing	503.5	196.3	796.6	254.8	421.1
Validation	660.3	190.8	1,021.0	247.4	265.2
Input 5: Loxton salinity, EC units					
Training	491.1	107.3	698.3	263.3	181.5
Testing	449.6	155.0	697.4	244.1	328.0
Validation	583.4	144.5	906.7	224.7	162.1
Input 6: Overland Corner flow, ML d^{-1}					
Training	19,956	21,620	85,963	1796	23,995
Testing	41,318	36,690	110,618	4764	59,893
Validation	12,678	13,477	46,664	1769	15,828
Input 7: Lock 1 Lower River level, m					
Training	4.0	1.0	4.4	0.5	1.2
Testing	2.5	1.6	5.3	0.7	2.9
Validation	1.1	0.7	2.9	0.6	0.7
Output 1: Murray Bridge salinity at (t + 14) days, EC units					
Training	588.8	171.6	931.0	310.6	291.6
Testing	535.9	217.6	862.9	262.3	452.6
Validation	698.3	216.5	1,115.7	282.2	353.6

[a] Electrical conductivity.

Table 2. Method 2 Statistics of the Salinity Training, Testing, and Validation Data Sets (Data Divided Using a Genetic Algorithm)

Variable and Data Set	Mean	Standard Deviation	Maximum	Minimun	Interquartile Range
Input 1: Murray Bridge salinity, EC units					
Training	596.6	197.0	1115.7	261.3	369.1
Testing	604.3	201.8	1077.8	265.8	390.9
Validation	596.4	188.4	1045.4	265.2	347.8
Input 2: Mannum salinity, EC units					
Training	579.5	186.6	1075.4	253.2	344.2
Testing	578.9	188.2	1072.8	255.8	360.4
Validation	578.5	176.5	982.5	259.4	306.7
Input 3: Morgan salinity, EC units					
Training	581.0	190.1	1061.3	182.8	341.0
Testing	575.8	199.2	1043.9	206.1	362.6
Validation	580.4	183.5	1022.7	176.9	333.6
Input 4: Waikerie salinity, EC units					
Training	573.6	178.5	1021.0	247.4	314.4
Testing	569.7	191.7	1008.3	251.8	344.1
Validation	571.7	174.2	1005.2	257.6	295.5
Input 5: Loxton salinity, EC units					
Training	503.3	130.0	906.7	224.7	218.4
Testing	502.0	142.4	900.4	228.8	247.5
Validation	502.4	126.1	857.8	244.1	200.3
Input 6: Overland Corner flow, ML d^{-1}					
Training	21,770	24,932	110,618	1769	25,349
Testing	22,816	25,785	110,056	1820	26,417
Validation	21,685	25,134	110,496	1943	24,196
Input 7: Lock 1 Lower River level, m					
Training	1.6	1.2	5.3	0.5	1.3
Testing	1.6	1.2	5.3	0.5	1.3
Validation	1.6	1.2	5.3	0.5	1.3
Output 1: Murray Bridge salinity at (t + 14) days, EC units					
Training	603.0	195.4	1115.7	262.3	366.7
Testing	598.1	205.7	965.7	264.4	400.3
Validation	603.0	190.3	1102.5	263.4	352.8

Table 3. Method 3 Statistics of the Salinity Training, Testing, and Validation Data Sets (Data Divided Using a Self-Organizing Map)

Variable and Data Set	Mean	Standard Deviation	Maximum	Minimun	Interquartile Range
Input 1: Murray Bridge salinity, EC units					
Training	587.7	202.4	981.7	267.2	346.8
Testing	586.5	202.5	1004.8	261.3	357.0
Validation	587.7	201.7	1047.0	263.9	333.9
Input 2: Mannum salinity, EC units					
Training	575.8	191.5	966.2	281.1	345.0
Testing	577.6	191.3	992.5	282.2	327.5
Validation	588.3	192.8	1075.4	286.1	320.6
Input 3: Morgan salinity, EC units					
Training	578.4	194.6	1031.8	282.8	345.0
Testing	579.8	192.0	1011.4	290.3	317.2
Validation	574.6	185.4	987.3	287.3	320.3
Input 4: Waikerie salinity, EC units					
Training	570.2	184.2	937.2	278.9	327.0
Testing	569.0	181.4	943.1	284.4	314.7
Validation	567.1	176.6	946.4	293.5	301.0
Input 5: Loxton salinity, EC units					
Training	494.2	132.1	777.1	263.3	236.3
Testing	496.5	129.8	783.5	279.8	214.3
Validation	494.5	129.2	784.0	274.9	218.6
Input 6: Overland Corner flow, ML d^{-1}					
Training	21,294	22,483	97,229	2240	22,595
Testing	21,518	22,886	99,065	2232	22,097
Validation	21,872	23,374	100,676	2069	23,606
Input 7: Lock 1 Lower River level, m					
Training	1.6	1.0	4.5	0.6	1.2
Testing	1.6	1.0	4.6	0.6	1.1
Validation	1.6	1.0	4.7	0.6	1.1
Output 1: Murray Bridge salinity at (t + 14) days, EC units					
Training	573.0	191.7	922.9	300.6	338.9
Testing	576.1	187.9	921.1	286.8	343.8
Validation	578.1	185.4	919.2	284.4	317.6

(20%) were used for validation. The time order of the data was not changed (i.e., the first 1604 records were used for calibration and the next 401 records were used for validation). The 1604 records in the calibration set were further divided into 1283 training records (80%) and 321 testing records (20%). The statistical parameters for the training, testing, and validation sets obtained using method 1 are shown in Table 1. Note that in Tables 1, 2, and 3, river salinities are given in electrical conductivity (EC) units of μS cm^{-1} at 25°C. One milligram per liter of total dissolved solids equals approximately 0.6 EC units. From Table 1 it can be seen that the statistical parameters vary widely between training, testing, and validation sets, and, in general, the statistics are not in good agreement. Hypothesis tests about the difference between the means of two samples (t test) and about the difference in the variance of two samples (F test) were performed. For each input and output variable, the testing and validation data sets were compared with the training sets, and a significance level of $\alpha = 0.05$ was chosen. In the t test it was hypothesized that there was no difference between the means of the two data sets. Likewise, in the F test it was hypothesized that there was no difference in the standard deviations of the two data sets. For the data division performed in method 1, both of these hypotheses were rejected for all of the testing and validation sets.

[38] In method 2, the GA data division technique was used. When using this technique, it is still necessary to determine the proportion of data to use for each of the subsets. For consistency, the same proportion used in method 1 was employed (1283 training records, 321 testing records, and 401 validation records). The statistical parameters for the training, testing, and validation sets obtained using method 2 are shown in Table 2. It can be seen that the statistics are in good agreement. Table 2 also shows that for each variable, the training set contains the maximum and minimum values. The only exception is the Morgan salinity variable, in which the training set contains the maximum value but not the minimum value. Hypothesis tests were also performed for the data sets obtained using method 2. The null hypotheses were accepted for all variables except that the F test null hypothesis was rejected at the 0.05 level for the Loxton salinity test set. This suggests that the variability in this test set was different to the training set. With the exception of this one test set, the GA data division technique used in method 2 was able to produce three data sets that were representatives of the same population.

[39] Method 3 employed the SOM data division technique. This technique avoids the need to arbitrarily select the proportion of data to include in each subset since only a minimum number of records are required (i.e., one record from each cluster is used for each subset) and superfluous data records are not used in the ANN. Using a SOM with a 10×10 Kohonen layer, the 2005 data samples were clustered into 49 clusters, with each cluster consisting of more than three records. Hence the training, testing, and validation sets each comprised 49 records. The statistical parameters for the training, testing, and validation sets obtained in method 3 are shown in Table 3. Once again, in Table 3 it can be seen that the statistics are in good agreement. The t test and F test null hypotheses were accepted for all input and output variables in method 3. Hence the statistics of all three data sets can be assumed to be the same.

4.2. Determination of Model Inputs

[40] *Maier and Dandy* [1996] found that the ANN models trained on the input set shown in Table 4 performed the best for this case study. Consequently, these 51 inputs were used for the

Table 4. Summary of Model Inputs

Variable	Location	Lags, days	Total Number
Salinity	Murray Bridge	1, 3,. . ., 11	6
Salinity	Mannum	1, 3,. . ., 15	8
Salinity	Morgan	1, 3,. . ., 15	8
Salinity	Waikerie	1, 2,. . ., 5	5
Salinity	Loxton	1, 2,. . ., 5	5
Flow	Overland Corner	−19, −17,. . .,7	14
Level	Lock 1 Lower	−3, −1,. . ., 5	5
Total number of inputs			51

ANN modeling. A description of how these inputs were determined is given by *Maier and Dandy* [1996].

[41] In the GA and SOM data division techniques (methods 2 and 3) performed in section 4.1, only the most recent values of each variable were used. However, it is possible to also incorporate the lags of each variable when using these two techniques.

4.3. Determination of Network Architecture

[42] The number of nodes in the input and output layers are fixed by the number of inputs and outputs, respectively. It is common practice to fix the number of hidden layers in the network and then to choose the number of nodes in each of these layers. It has been shown that only one hidden layer is required to approximate any continuous function, given that sufficient degrees of freedom (i. e., connection weights) are provided [*Cybenko*, 1989]. Hence one hidden layer was utilized in this study. *Maier and Dandy* [1998] conducted empirical trials on the salinity data set and determined that 30 hidden layer nodes provided optimal performance. Consequently, a network with 51 nodes in the input layer, 30 hidden layer nodes, and one node in the output layer was used for each of the models developed in this study. To ensure that overtraining did not occur (i.e., when the network performs well on the training data but poorly on independent test data), the SaveBest command in NeuralWorks Professional II/Plus was used. This command alternately runs train and test commands and saves the network with the best test results during the run. After 100 iterations, with no further improvement in the test set results, training is stopped.

4.4. Model Validation and Performance Measures

[43] The SaveBest command uses the RMS error (RMSE) measured on the test set to determine when to stop training the network. Hence an independent validation set was used for all three methods investigated, as discussed in section 4.1. The RMSE was used as the performance measure, as it places greater emphasis on larger forecasting errors. The average absolute percentage error (AAPE) was also calculated for comparison.

5. Results and Discussion

[44] The RMSEs and AAPEs for each data division method are summarized in Table 5. It can be seen that the model developed in method 1 produced a much larger error on the testing and validation sets than the models developed in methods 2 and 3. However, in method 1, the model performed well on the training set. This suggests that the arbitrary data division used in method 1 produced a training set that was not representative of the entire population, which is in agreement with the statistics shown in Table 1. In Table 5 it is also important to observe that unlike method 1, methods 2 and 3 produced results that were consistent for all three data sets. Again, this is in agreement with the statistics obtained in Table 2 and 3. In method 2, the GA data set division produced results with RMSEs and AAPEs that ranged from 33.9 to 39.1 EC units and from 4.7 to 5.4%, respectively. In method 3, the SOM data set division produced results with RMSEs and AAPEs ranging from 35.5 to 39.1 EC units and from 5.6 to 6.3%, respectively. This is in direct contrast to the model developed in method 1, which produced results over a much wider range, including RMSEs and AAPEs ranging from 38.2 to 58.8 EC units and from 5.7 to 7.8%, respectively.

[45] The above results show that the GA-based approach and the SOM approach are suitable methods for ensuring that the training, testing, and validation sets are representative of the same population and hence provide an appropriate means for data division. The SOM approach has the additional advantage that a training set can be constructed using the minimum number of samples. The results also show that for good performance, the data sets need to have similar statistical properties.

[46] To understand why the ANN developed using method 1 performed poorly, it is necessary to critically dissect the model and examine the regions of poor performance. The SOM provides a useful technique for achieving this purpose. Figure 1 shows the validation results of the ANN developed using method 1, with four regions of poor performance identified. The analysis was conducted by clustering all of the input and output data from the training, testing, and validation sets using a SOM. Once the clusters had been formed, it was possible to examine each region of poor performance and determine if any data representative of that region had been included in the training set. This was done by inspecting the clusters. If the training set did not contain data representative of that region, then it is expected that the model may perform poorly, since the model had not been trained for this event.

Table 5. Root-Mean-Square Error (RMSE) and Average Absolute Percentage Error (AAPE) for the 14-Day Forecasts

Data Set	Method 1: Conventional Division		Method 2: GA[a] Division		Method 3: SOM[b] Division	
	RMSE, EC units	AAPE, %	RMSE, EC units	AAPE, %	RMSE, EC units	AAPE, %
Training	38.2	5.7	39.1	5.4	39.1	6.3
Testing	51.6	7.8	33.9	4.7	37.1	5.9
Validation	58.8	7.6	38.8	5.3	35.5	5.6

[a] Genetic algorithm.
[b] Self-organizing map.

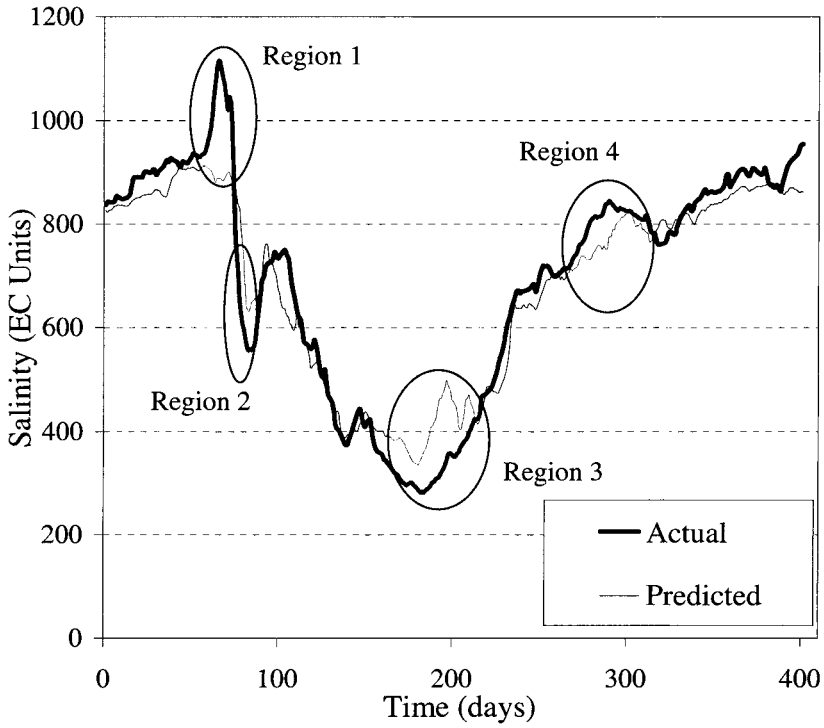

Figure 1. Validation set 14-day forecast of salinity at Murray Bridge for the model developed using method 1 (May 1991 to June 1992).

[47] After performing the analysis for region 1 (Figure 1), it was found that all of the data in this region were located in a single cluster. This single cluster only contained validation data, and hence the observed pattern was not represented in the training set. This explains why the model developed in method 1 was unable to match the peak observed in region 1 of Figure 1. The data contained in region 2 (Figure 1) were split into two clusters in the SOM. However, both of these clusters only contained validation data. Region 3 data (Figure 1) were divided into six clusters. Four of these clusters only contained validation data, and the remaining two clusters contained a small amount of training data. Region 4 data (Figure 1) were split into two clusters, both of which

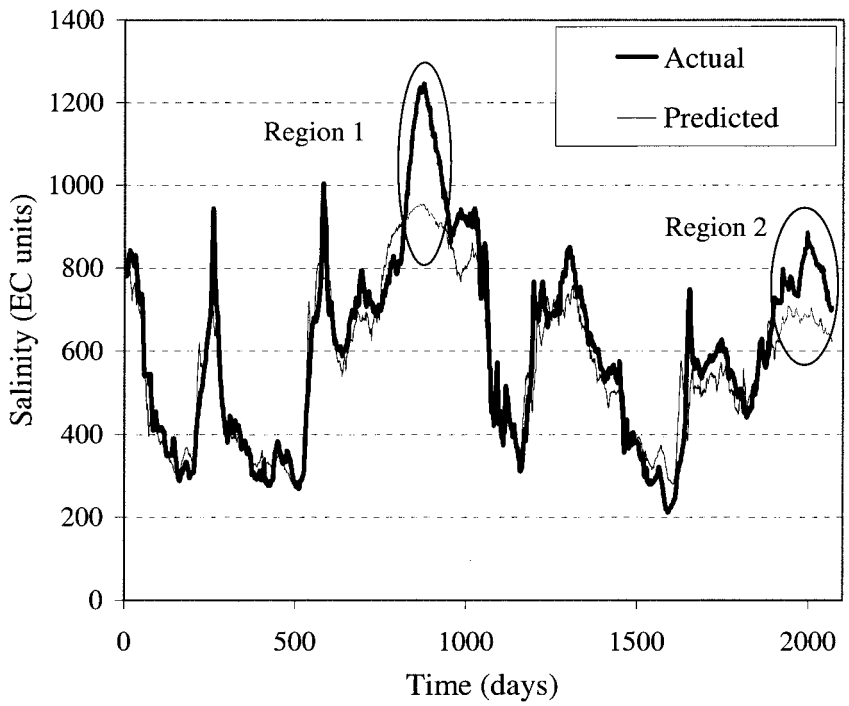

Figure 2. Second validation set 14-day forecast of salinity at Murray Bridge for the model developed using method 1 (July 1992 to March 1998).

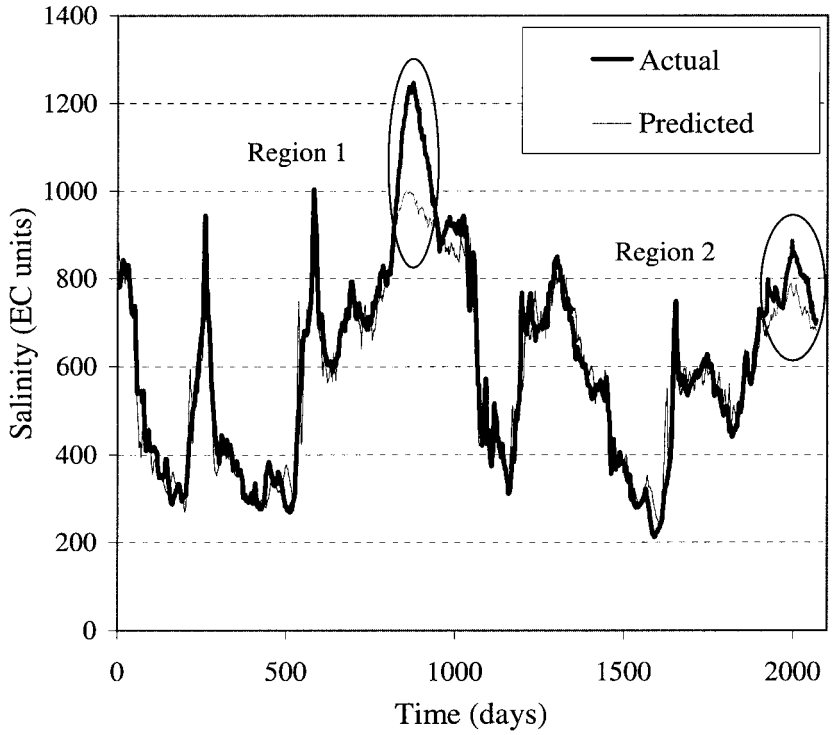

Figure 3. Second validation set 14-day forecast of salinity at Murray Bridge for the model developed using method 2 (July 1992 to March 1998).

contained only validation data. By using a SOM to analyze the regions of poor performance, it was possible to verify that the cause of the poor performance was a lack of representative data in the training set.

[48] In a real-world scenario the developed ANN model would need to produce forecasts based on new data, the statistics of which are unknown. In addition, each of the data division methods investigated produced different training, testing, and validation sets. To fairly evaluate and compare the performance of each technique, it was considered necessary to test each model on the same data set. Hence a second, independent validation data set was used, consisting of daily data from the period 15 July 1992 to 13 March 1998. The models developed in methods 1, 2, and 3 were used to obtain 14-day forecasts for this second validation set. Plots of the 14-day forecasts obtained from methods 1 and 2 are shown in Figures 2 and 3, respectively. For brevity, the plot obtained from method 3 was omitted, as it is similar to the plot obtained in method 2.

[49] In Figures 2 and 3 the most notable regions of poor performance have been identified as regions 1 and 2. The models developed using all three methods performed poorly both in regions 1 and 2 (Figures 2 and 3), and to investigate the cause of this poor performance, the SOM technique was used to analyze the data. To perform the analysis, the data from the second validation set were combined with all of the data used in the development of the models (i.e., the training, testing, and validation data). The SOM was used to cluster the data, and each of the clusters was inspected. It was discovered that all of the data from region 1 (Figures 2 and 3) were contained in six clusters; however, all of these clusters contained data only from the second validation set. Hence no data representative of region 1 had been used to develop (i.e., to train, test, or validate) any of the models. This explains why the models developed using all three methods performed poorly on this region. Likewise, the data from region 2 (Figures 2 and 3) were contained in six clusters, five of which contained data only from the second validation set. One cluster containing seven of the second validation set samples also contained training data. The models developed using methods 2 and 3 were able to perform well on these seven points.

[50] The performances of the three methods for the second validation set are summarized in Table 6. To obtain a fair representation of each model's performance, another calculation of the RMSEs and AAPEs was performed with the data from regions 1 and 2 removed, since these were unique data points and no data representative of these two regions had been used to train the models. Even after removing regions 1 and 2, the results shown

Table 6. Results of the Artificial Neural Networks Developed in Each Method on the Second Validation Set (RMSE and AAPE)

Data	Method 1: Conventional Division		Method 2: GA Division		Method 3: SOM Division	
	RMSE, EC units	AAPE, %	RMSE, EC units	AAPE, %	RMSE, EC units	AAPE, %
Second validation set	86.1	10.5	65.3	7.4	77.6	9.4
Second validation set (regions 1 and 2 removed)	67.7	9.9	51.8	7.2	62.6	9.0

in Table 6 are not as good as those obtained in Table 5. The larger RMSE and AAPE values obtained on the second validation set may be attributed to the very large size of the data set, which comprised 2068 records. This highlights the importance of periodically retraining the ANN model at regular intervals when it is to be used in a real-world situation.

[51] From Table 6 it can be seen that the GA data division (method 2) performed best out of the three techniques. The SOM data division (method 3) slightly outperformed the conventional approach (method 1). However, it is worth pointing out that the model developed using the SOM data division technique was only trained on 49 data records, whereas the models developed in methods 1 and 2 were trained using 1283 data records. The ability of the SOM technique to outperform the conventional method, while only using 49 training samples, provides further evidence of its superiority at developing a representative training set with a minimum number of samples.

[52] To investigate whether 49 training samples were able to capture enough of the variance in the available data set, an ANN model was developed using one data sample from each cluster for the testing and validation sets, and all the remaining data samples were placed in the training set. However, this only increased the noise in the forecasts, and when tested on the second validation set, no improvement was made over the model developed using 49 training samples. This suggests that the performance on the second validation set could not be improved, as it contained data records that were not represented in the original set of data. Again, in a real-world situation, this problem could be avoided by periodically retraining the ANN model.

6. Conclusion

[53] This paper has reported two techniques for the optimal division of data for ANN models and compared the results with the method most commonly employed in the literature. When tested on the second validation set, the GA and SOM data division techniques resulted in a reduction in RMSE of 24.2 and 9.9%, respectively, over the conventional data division method. The GA and SOM data division techniques provide an advantage over conventional methods in that they insure that the training, testing, and validation sets are representative of the same population. Given a suitable amount of data, these techniques enable the development of a training set that extends to the edges of the modeling domain in all dimensions. In so doing, the ANN is able to find a generalized solution to the problem being investigated. It is important to note that overfitting only occurs when the training set is not totally representative of the population [*Masters*, 1993]. Hence the GA and SOM division techniques will reduce the likelihood of overfitting the data in the training set. It was found that the ANN models developed using the SOM and GA data division techniques outperformed the conventional method.

[54] When developing a training set, it is also important that the training samples are evenly distributed throughout the modeling domain; otherwise training will focus on densely clustered regions and neglect the sparsely represented regions. The SOM technique achieves an evenly distributed training set by first dividing all of the data into a number of clusters. By then sampling an equal number of records from each cluster, the training, testing, and validation set records are evenly distributed throughout the problem domain. Another advantage is that the SOM data division technique employs the minimum number of samples for achieving this objective and consequently avoids the problem of what proportion of the data should be used in the training, testing, and validation subsets. While the GA data division technique does not automatically determine the proportion of data to use for each subset, it does have the advantage of ensuring that the extreme values are contained within the training set. This can be achieved through the addition of penalty constraints.

[55] Finally, when tested on a second, independent validation set, the models developed using methods 2 and 3 outperformed the model developed in method 1. However, this performance was masked by the fact that the second validation set contained new regions of data that were not representative of the data used in the training, testing, and validation sets. This highlights the importance of periodic retraining of the model. The two data division methods presented provide very useful techniques for dividing the data into three statistically similar and representative subsets. It is important to note that these techniques provide representative data sets that best approximate the modeling domain based on the available data. There can be no guarantee that the model will perform well on new data. However, using the SOM analysis, it has been shown in this paper that it is possible to diagnose regions of poor performance resulting from uncharacteristic data (i.e., data that are unlike any of the data that have been used in developing the model).

[56] **Acknowledgments.** Financial support for this research was provided by an Australian postgraduate award at Adelaide University with assistance from an industry partner, United Utilities Australia. This support is gratefully acknowledged.

References

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

Braddock, R. D., M. L. Kremmer, and L. Sanzogni, Feed-forward artificial neural network model for forecasting rainfall run-off, *Environmetrics*, *9*, 419–432, 1998.

Cai, S., H. Toral, J. Qiu, and J. S. Archer, Neural network based objective flow regime identification in air-water two phase flow, *Can. J. Chem. Eng.*, *72*, 440–445, 1994.

Campolo, M., A. Soldati, and P. Andreussi, Forecasting river flow rate during low-flow periods using neural networks, *Water Resour. Res.*, *35*(11), 3547–3552, 1999.

Cybenko, G., Approximation by superpositions of a sigmoidal function, *Math. Control Signals Syst.*, *2*, 203–314, 1989.

Dandy, G. C., and P. D. Crawley, Optimization of multiple reservoir systems including salinity effects, *Water Resour. Res.*, *28*(4), 979–990, 1992.

Dandy, G. C., A. R. Simpson, and L. J. Murphy, An improved genetic algorithm for pipe network optimization, *Water Resour. Res.*, *32*(2), 449–458, 1996.

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

Dwyer Leslie Pty Ltd., River Murray water quality management study, *Working Pap. E and F*, Maunsell and Partners, Melbourne, Vict., Australia, 1984.

Flood, I., and N. Kartam, Neural networks in civil engineering, I, Principles and understanding, *J. Comput. Civ. Eng.*, *8*(2), 131–148, 1994.

Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, 412 pp., Addison-Wesley-Longman Reading, Mass., 1989.

Goldberg, D. E., and K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in *Foundations of Genetic Algorithms*, edited by J. E. Rawlins, pp. 69–93, Morgan-Kaufmann, Burlington, Mass., 1991.

Hall, M. J., and A. W. Minns, Classification of hydrologically homogeneous regions, *Hydrol. Sci. J.*, *44*(5), 693–704, 1999.

Hsu, K. L., X. G. Gao, S. Sorooshian, and H. V. Gupta, Precipitation estimation from remotely sensed information using artificial neural networks, *J. Appl. Meteorol.*, *36*(9), 1176–1190, 1997.

Islam, S., and R. Kothari, Artificial neural networks in remote sensing of hydrologic processes, *J. Hydrol. Eng.*, *5*(2), 138–144, 2000.

Kaski, S., J. Kangas, and T. Kohonen, Bibliography of self-organizing map (SOM) papers: 1981–1997, *Neural Comput. Surv.*, *1*, 102–350, 1998.

Kohonen, T., Self-organized formation of topologically correct feature maps, *Biol. Cybern.*, *43*, 59–69, 1982.

Kohonen, T., The self-organizing map, *Proc. IEEE*, *78*(9), 1464–1480, 1990.

Maier, H. R., and G. C. Dandy, The use of artificial neural networks for the prediction of water quality parameters, *Water Resour. Res.*, *32*(4), 1013–1022, 1996.

Maier, H. R., and G. C. Dandy, The effect of internal parameters and geometry on the performance of back-propagation neural networks: An empirical study, *Environ. Modell. Software*, *13*, 193–209, 1998.

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

Masters, T., *Practical Neural Network Recipes in C++*, 493 pp., Academic, San Diego, Calif., 1993.

Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, 340 pp., Springer-Verlag, New York, 1994.

Minns, A. W., and M. J. Hall, Artificial neural networks as rainfall-runoff models, *Hydrol. Sci. J.*, *41*(3), 399–417, 1996.

NeuralWare, *Neural Computing: A Technology Handbook for NeuralWorks Professional II/PLUS and NeuralWorks Explorer*, 324 pp., Aspen Technol., Cambridge, Mass., 1998.

Ray, C., and K. K. Klindworth, Neural networks for agrichemical vulnerability assessment of rural private wells, *J. Hydrol. Eng.*, *5*(2), 162–171, 2000.

Simpson, A. R., G. C. Dandy, and L. J. Murphy, Genetic algorithms compared to other techniques for pipeline optimisation, *J. Water Resour. Plann. Manage.*, *120*(4), 423–443, 1994.

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

World Health Organisation, *Guidelines for Drinking Water Quality*, Geneva, 1984.

Wright, A. H., Genetic algorithms for real parameter optimization, in *Foundations of Genetic Algorithms*, edited by J. E. Rawlins, pp. 205–218, Morgan Kaufmann, Burlington, Mass., 1991.

G. J. Bowden, G. C. Dandy, and H. R. Maier, Centre for Applied Modelling in Water Engineering, Department of Civil and Environmental Engineering, Adelaide University, Adelaide 5005, Australia. (gbowden@civeng.adelaide.edu.au)