5-20-2008

Microgenetic algorithms and artificial neural networks to assess minimum data requirements for prediction of pesticide concentrations in shallow groundwater on a regional scale

Goloka Behari Sahoo
University of California

Chittaranjan Ray
University of Hawaii, cray@nebraska.edu

Microgenetic algorithms and artificial neural networks to assess minimum data requirements for prediction of pesticide concentrations in shallow groundwater on a regional scale

Goloka Behari Sahoo[1] and Chittaranjan Ray[2]

Received 10 January 2007; revised 14 January 2008; accepted 20 February 2008; published 20 May 2008.

[1] Artificial neural networks (ANNs) have been extensively used for forecasting problems involving water quantity and quality. In most cases, the geometry and model parameters of the ANN are set using a trial-and-error approach to achieve better network generalization ability, whereby the available data are divided arbitrarily into training, testing, and validation subsets. It has been shown that using the arbitrary sample selection method to assign samples into the training subset commonly results in the inclusion of samples from densely clustered regions and omission of samples from sparsely represented regions. This paper presents a systematic approach using the self-organizing map (SOM) clustering technique that identifies which samples and determines how many samples should be included in each of the three subsets required by ANN for optimum predictive performance efficiency. In addition, this paper presents the microgenetic algorithms (μGA) that optimize ANN's geometry and model parameters in terms of the correlation coefficient (R). In the sensitivity analysis, μGA model parameters are found to be least sensitive to the optimum R value, while ANN's predictive performance is significantly affected by (1) the poor selection of its geometry and model parameters and (2) the arbitrary selection of samples for the three subsets of data used. It is demonstrated that the μGA-ANN model using the SOM technique for data division outperforms the μGA-ANN model using arbitrary data division. For the training subset, the model using the SOM technique identifies samples that are representative of the region, requiring only 20% of the total samples, whereas the arbitrary sample selection method requires 50–90%. Because resampling on a regional scale is expensive and time consuming, substantial cost and time could be saved if resampling could be done only on the 20% representative drinking water wells.

Citation: Sahoo, G. B., and C. Ray (2008), Microgenetic algorithms and artificial neural networks to assess minimum data requirements for prediction of pesticide concentrations in shallow groundwater on a regional scale, *Water Resour. Res.*, *44*, W05414, doi:10.1029/2007WR005875.

1. Introduction

[2] Groundwater constitutes 96% of the world's total available freshwater resources [*Gleick*, 1996]. Over 95% of rural residents and 50% of the total population in the United States rely on groundwater for their drinking water [*Solley et al.*, 1998; *Barbash and Resek*, 1999]. There is a potential threat of groundwater contamination across the nation, particularly from synthetic organic pesticides used for controlling weeds, insects, and other organisms in agricultural and nonagricultural settings [*U.S. Environmental Protection Agency*, 1990; *Kolpin et al.*, 1994; *Barbash and Resek*, 1999; *Barbash et al.*, 1999]. Not surprisingly, most of the chemicals found in rural domestic wells are herbicides because of the quantity used in agriculture [*Weber et al.*, 1997].

[3] A number of solute transport and nonpoint source (leaching) models are available to predict the movement of chemicals from the land surface to groundwater [e.g., *Carsel et al.*, 1984; *U.S. Department of Agriculture Agricultural Research Service*, 1992; *Knisel*, 1993; *Simunek et al.*, 1998]. For predictive calculations, these models require data on physical descriptions of the porous media, suitable initial and boundary conditions for flow and transport processes, and reactions occurring between the solid matrix and water phase chemicals in the soil. None of these models are able to predict pesticide concentrations at a well site on a regional scale for the following three reasons: (1) none includes the detailed complex interactions between soils and pesticides, heterogeneity of soil physical and chemical properties, and uncertainty in estimating regional flow and transport parameters; (2) input parameters to these models are space- and time-dependent, and at the same time they undergo complex interactions; and (3) detailed soil characterization data on a regional scale are not available. Prediction of a well's vulnerability to pesticide contamination using available

[1]Department of Civil and Environmental Engineering, University of California, Davis, California, USA.
[2]Department of Civil and Environmental Engineering and Water Resources Research Center, University of Hawaii at Manoa, Honolulu, Hawaii, USA.

0043-1397/08/2007WR005875

information is often important from a public health perspective as well as from a regulatory perspective in order to guide future monitoring efforts [*Ray and Klindworth*, 2000; *Sahoo et al.*, 2006]. Empirical models such as artificial neural networks (ANNs) are capable of predicting pesticide/pollutant occurrence in drinking water wells in complex systems [e.g., *Ray and Klindworth*, 2000; *Mishra et al.*, 2004; *Dixon*, 2005; *Sahoo et al.*, 2006; *Wang et al.*, 2006]. Examples of the use of ANN for estimating pesticide fate and transport in soils can be found in the works of *Lohninger* [1994] and *Yang et al.* [2003]. Although ANN is not intended as a substitute for conceptual process-based models, it can be used as a viable alternative to assess vulnerability of wells using readily available information about well sites.

[4] The ANN model requires three subsets of data: (1) training, (2) testing, and (3) validation. These subsets must represent the same population for the ANN to achieve adequate generalization ability [*Masters*, 1993; *Tokar and Johnson*, 1999; *ASCE Task Committee*, 2000; *Maier and Dandy*, 2000; *Bowden et al.*, 2002]. They emphasized the importance of the training subset because it represents the whole data set, providing information that extends to the edges of the modeling domain in all dimensions for the optimization of ANN's connection weights and model parameters. If this condition is not fulfilled, the predictive performance efficiency of an undertrained ANN would suffer significantly. *Flood and Kartam* [1994] pointed out that the number of training samples can significantly influence a network's performance. *Minns and Hall* [1996, p. 416] acknowledged that ANN is susceptible to becoming "... a prisoner of its training data." To ensure that their training subset included information on all dimensions of the whole population, *Ray and Klindworth* [2000] made it large enough to represent the full population. *Mishra et al.* [2004], *Sahoo and Ray* [2006a], and *Wang et al.* [2006] used approximately 85–90%, 63%, and 50%, respectively, of the total available samples in their individual training subsets. In all cases, the total samples were divided arbitrarily among the subsets, so even if the sample size was large, there was no guarantee that the training subset would include information on the entire population. Therefore, *Maier and Dandy* [2000] proposed to divide the original data set into the three subsets using a trial-and-error method for achieving optimum ANN performance. However, if 100 samples are divided into training, testing, and validation subsets consisting of 63, 19, and 18 samples, respectively, there will be $100!/(63! \times 19! \times 18!) = 1.2 \times 10^{125}$ ways of arranging the samples. It would be practically impossible to examine all the combinations.

[5] Radial basis function network (RBFN) and back propagation neural network (BPNN) are commonly used for predictive purposes in water resources systems [*ASCE Task Committee*, 2000; *Alp and Cigizoglu*, 2007]. Thus, we used these two types of ANN models in this study. *Maier and Dandy* [2000], *ASCE Task Committee* [2000], *Birikundavyi et al.* [2002], *Shi et al.* [2005], *Sahoo and Ray* [2006a], and *Alp and Cigizoglu* [2007] stressed that ANN's geometry and modeling parameters have a significant influence on its performance efficiency and should be optimized using a trial-and-error procedure. *Kingston et al.* [2005] pointed out that a significant component of prediction uncertainty can be attributed to the uncertainty in parameters that govern the model function. However, in cases where the solution space is large enough, use of an optimization technique, such as a microgenetic algorithm (μGA), saves time and computational effort. For example, if the solution space of the spread and the optimum number of neurons in an RBFN hidden layer (described in section 2.3) are in the range 1 to 80 and 1 to 90, respectively, then there will be $80 \times 90 = 7200$ ways of arranging the spread and the number of neurons. The spread is a real number and is searched up to an accuracy of 0.0001 in this study. Thus, the number of ways of arranging the spread and the number of neurons increases significantly. Searching for the optimal solution in this range using a trial-and-error approach is cumbersome and time consuming.

[6] In this study, we used a data set originally collected for the midcontinental United States during 1991 to 1994 to estimate pesticide contamination in drinking water wells derived from nonpoint sources, i.e., from pesticides applied to agricultural and nonagricultural fields [*Kolpin et al.*, 1995]. Initial assessment of contamination requires one-time sampling. Resampling is needed to ensure if the well is free from contamination. Because sampling and resampling of individual wells on a regional scale (throughout the midcontinental United States) would be expensive and time consuming, it is not practical to have a large data set for using an ANN. Thus, identification of well sites vulnerable to pesticide contamination allows us to limit monitoring (i.e., resampling) to only those wells when preparing an ANN training subset that is representative of the whole region. Therefore, the objectives of this study are (1) to identify the samples that should be included in a training subset for optimum ANN performance, (2) to develop a μGA-ANN model for searching optimal ANN's model parameters and geometry, (3) to evaluate the sensitivity of the μGA's model parameters on the basis of ANN predictive efficiency, and (4) to develop guidelines for the preparation of three subsets of data when the total available samples are limited. The first objective is achieved using a data-clustering technique, the self-organizing map (SOM). The second objective is to present a model that optimizes the ANN predictive performance efficiency for a given training subset. The third objective finds a set of optimized model parameters for the μGA. Finally, the fourth objective addresses (1) which samples and (2) what proportions of the total number of samples should be included in the training, testing, and validation subsets. Subroutines available in MATLAB version 7.1 [*The Mathworks Inc.*, 2005] were modified and used to create RBFN, BPNN, and SOM models.

2. Background

2.1. Microgenetic Algorithms

[7] Genetic algorithms (GAs) are widely used for optimization of water resources variables [e.g., *Bowden et al.*, 2002; *Jain and Srinivasulu*, 2004; *Ines and Honda*, 2005]. In this study, a binary (chromosomes are made of 0 and 1 digits, hence binary) μGA is applied to optimize the ANN's geometry and model parameters. For a simple GA, the reader is referred to *Goldberg* [1989]. For detailed information on μGA, the reader is referred to *Krishnakumar* [1989], *Carroll* [1996], *Abu-Lebdeh and Benekohal* [1999], *Ines and Honda* [2005], and D. L. Carroll (FORTRAN genetic

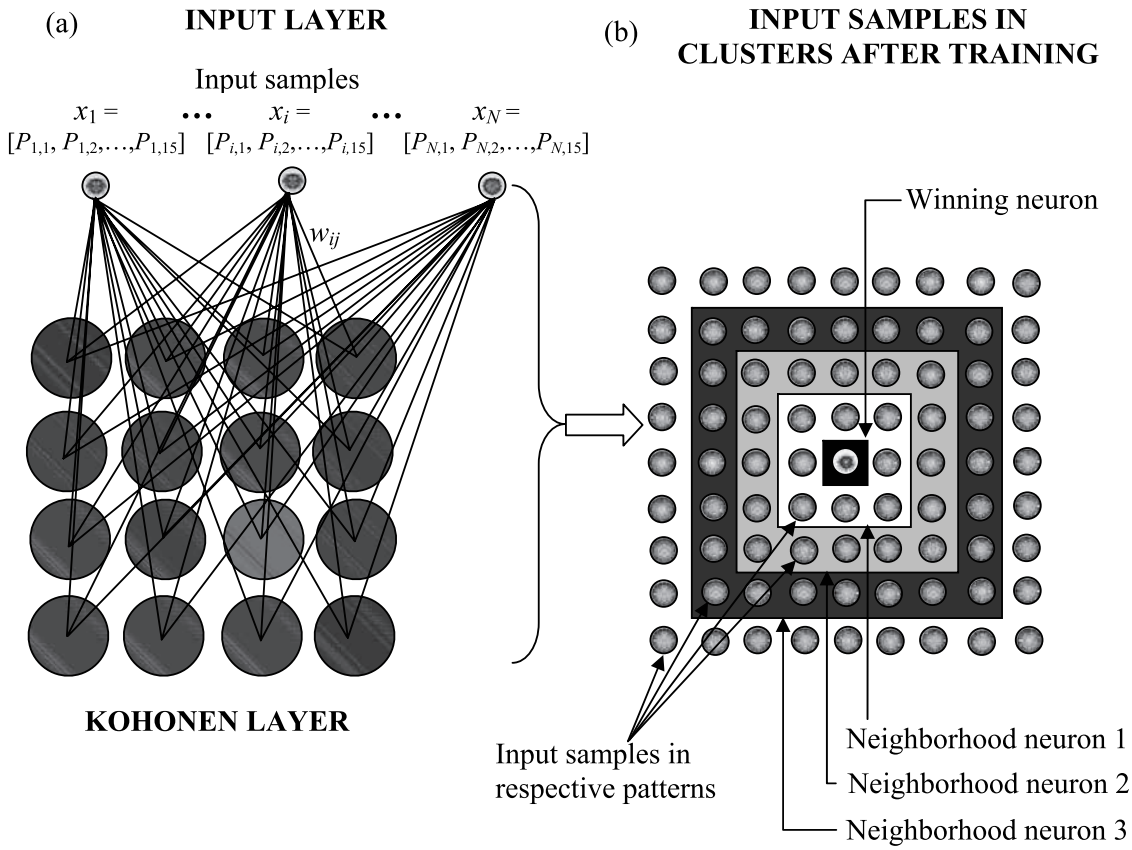

Figure 1. (a) Schematic of a two-layer Kohonen network. The input layer neurons, equal to the total number of input samples N, are connected to each neuron of the Kohonen layer (i.e., output layer) through a connection weight. Here w_{ij} represents the connection weight for ith input neuron to jth neuron of the output layer. Each input sample consists of 15 input parameters, i.e., $P_{i,1}$ to $P_{i,15}$. (b) Final configuration of the neighborhoods of Kohonen layer neurons, each including input samples of similar characteristics. A single neuron can respond to several input samples of similar characteristics, thus representing a cluster of input samples in its space. Only four clusters neighboring the winning neuron are illustrated. The winning neuron is at the center, surrounded by neighborhoods of increasing diameter.

algorithm (GA) driver version 1.7.0, 1999, available at http://www.cuaerospace.com/carroll/ga.html).

[8] A μGA is similar to a standard GA. However, important distinctions of the μGA are that it uses a small population (i.e., μ population), restarts when the characteristics of the chromosomes are greater than or equal to 95% similarity in a generation, and performs no mutation since sufficient diversity is introduced after convergence of a μ population. *Krishnakumar* [1989] showed that a μGA reaches the near-optimal region faster than a simple GA for stationary and nonstationary function optimization. Also, *Carroll* [1996] demonstrated that a μGA is able to find a global maximum of the third-order deceptive function described in *Goldberg* [1989] that a simple GA failed to optimize. *Abu-Lebdeh and Benekohal* [1999] found that a μGA performed better than a simple GA in terms of best fitness value for the deceptive function. Thus, the modified μGA (http://www.cuaerospace.com/carroll/ga.html) was modified as necessary for optimization of ANN's geometry and model parameter for use in this study.

2.2. Self-Organizing Map

[9] The SOM technique is used to detect regularities and correlations among the samples of a data set. A schematic of SOM architecture is shown in Figure 1. SOM consists of the Kohonen layer (i.e., the output layer), the neurons of which are fully connected to each neuron of the input layer with a connection weight but not to the neurons of the same layer. The *Kohonen* [1982] learning algorithm, which belongs to the class of unsupervised competitive learning algorithms, is commonly used in training the SOM to group randomly sequenced input patterns into clusters [*Haykin*, 1999; *Lin and Chen*, 2006; *Chen et al.*, 2006].

[10] The SOM input layer is an array of N neurons equal to the number of samples in the data set. It can be denoted by

$$X = [x_1, x_2, \ldots, x_N]^T \tag{1}$$

where superscript T denotes matrix transposition. The output layer includes the output neurons u_j, where j = 1, 2, ..., M, which are typically organized in a two-dimensional planar lattice. Each connecting line in Figure 1 denotes a value, referred to here as connection weight. The weights from the input layer neuron to the output layer neuron are w_{ij}, where i = 1, 2, ..., N. The weight vector of

each Kohonen neuron has the same dimension as the input data set. The weight vector can be written as

$$W_j = \left[w_{1j}, w_{2j}, \ldots, w_{ij}, \ldots, w_{Nj}\right]^T \quad (2)$$

[11] The SOM training process begins with all weights initialized to small random real numbers. The SOM algorithm computes a similarity (distance) measure between the input vector X and the weight vector W_j of each neuron u_j. The Euclidean distance d_j between W_j and X is frequently used as the similarity measure [*Lin and Chen*, 2006; *Chen et al.*, 2006]. It is given as

$$d_j = \|X - W_j\| = \sqrt{\sum_{i=1}^{N} \left(x_i - w_{ij}\right)^2} \quad (3)$$

where $\|\ldots\|$ is the Euclidean distance. The neuron on the Kohonen layer whose Euclidean distance (equation (3)) is the smallest is the winner. The weights of this winning neuron are adjusted in the direction of the input vector. Not only the winning neuron but also the neurons in the topological neighborhood of the winning neuron are affected by the competition. The influence of competition decays symmetrically from the winning neuron's location. The winning neuron is at the center of the topological neighborhood. A typical choice of a topological neighborhood function that satisfies these requirements is the Gaussian function [*Lin and Chen*, 2006; *Chen et al.*, 2006; *Lin and Wang*, 2006; *Lau et al.*, 2006]:

$$h_j = \exp\left(-\frac{\|u_j - u_j^*\|^2}{2\sigma_s^2}\right) \quad (4)$$

where h_j is the topological neighborhood, σ_s is the effective width of the topological neighborhood, and u_j^* is the winning neuron.

[12] During training, the weight vector W_j changes at each iteration as

$$\Delta W_j = \eta h_j (X - W_j) \quad (5)$$

where η is the learning rate parameter of the algorithm. Hence, the updating weight vector W_j $(t + 1)$ is defined by *Kohonen* [1982] as

$$W_j(t+1) = W_j(t) + \eta(t) h_j(t) \left(X - W_j(t)\right) \quad (6)$$

where t is the iteration number, and $\eta(t)$ and $h_j(t)$ are the learning rate parameter and the topological neighborhood function at t, respectively. Equation (6) is applied to all neurons in the lattice that lie inside the topological neighborhood of the winning neuron. During SOM training, the weight vectors tend to move toward the input pattern because of the neighborhood updating, that is, the adjustment makes the weight vectors similar to the input pattern.

[13] The synaptic weights in the network are updated in two phases: (1) an ordering or self-organizing phase followed by (2) a convergence phase [*Haykin*, 1999].

[14] 1. The ordering or self-organizing phase is the first phase of the adaptive process in which the topological ordering of the weight vectors takes place. This phase takes 1000 or more iterations of the SOM algorithm [*Haykin*, 1999]. In this phase, the following two criteria must be considered in the choice of $\eta(t)$ and $h_j(t)$ [*Haykin*, 1999]: (1) $\eta(t)$ starts at an initial value (say, 0.1), and then it decreases gradually as t increases but remains above 0.01; and (2) the neighborhood function $h_j(t)$ initially includes all neurons in the network centered on the winning neuron, and then it shrinks slowly with time. Specifically, during the ordering phase, $h_j(t)$ is reduced to a small value of only a couple of neighboring neurons around a winning neuron or to the winning neuron itself. For the case of a two-dimensional lattice, the initial size σ_0 of the neighborhood function is set equal to the radius of the lattice.

[15] 2. The convergence phase of the adaptive process is needed to fine tune the feature map, thereby providing an accurate statistical quantification of the input space. As a general rule, the number of iterations constituting the convergence phase must be at least 500 times the number of neurons in the output layer of the network [*Haykin*, 1999]. Thus, the convergence phase continues for thousands or more iterations considering the following two conditions [*Haykin*, 1999]: (1) For good statistical accuracy, the learning parameter $\eta(t)$ is maintained at a small value, on the order of 0.01 during the convergence phase. However, it is not allowed to decrease to zero. (2) The neighborhood function h_j contains only the nearest neighbors of a winning neuron, which may eventually reduce to one or zero neighboring neuron.

[16] In this study, the number of iterations for the ordering phase was set at 2000. The ordering phase and convergence phase learning rates were set to 0.9 and 0.02, respectively. The number of iterations involved in the convergence phase was set at 500 times the number of neurons in the SOM output layer.

[17] The concept of neighborhoods is illustrated in Figure 1b, which shows a winning neuron in a two-dimensional grid top Kohonen layer. The winning neuron has neighborhoods of increasing diameter surrounding it. The neighborhood of diameter 1 includes the winning neuron and its immediate neighbors. The neighborhood of diameter 2 includes the diameter 1 neurons and their immediate neighbors.

[18] There is no theoretical principle for determining the optimum size of the output layer; hence, the output layer is kept large enough to ensure that the maximum number of clusters is formed from the input data set. Equation (4) indicates that the number of clusters depend on both the input data set pattern and the SOM size (i.e., number of neurons in the Kohonen layer).

2.3. Radial Basis Function Neural Network

[19] Details of RBFNs can be found in work by *Haykin* [1999], *Principe et al.* [1999], *Shi et al.* [2005], and *Alp and Cigizoglu* [2007]. This paper presents only the RBFN training process, the network geometry, processing neurons functions, and model parameters used in this study. An RBFN starts with a minimal network of one RBF neuron. The network is trained with the RBF neuron, and a mean

square error (MSE) is estimated for the training subset. MSE is the average of sum square error between ANN-predicted output and measured target values of the training subset. If the MSE is greater than the network's threshold MSE, which is set low to prevent premature cessation of the training process (10^{-20} in this study), another RBF neuron is added to the hidden layer. The MSE of the network is changed for the new geometry (i.e., a new set of weight matrices). The RBFN training continues until one of the following conditions occurs:

[20] 1. The MSE is less than or equal to the threshold MSE.

[21] 2. The total number of RBFN neurons is equal to the maximum number of RBF neurons. Since each RBF neuron must respond to at least one input sample, the maximum number of neurons cannot exceed the number of input samples [*Haykin*, 1999; *Principe et al.*, 1999]. Because the optimal number of RBF neurons is not known a priori for a specific problem, it is determined using the μGA.

[22] 3. The MSE starts to exceed the MSE of the validation subset. The modified RBFN model estimates the MSE of the validation subset presented to the network. The MSE should remain below the MSE of the validation subset to prevent overtraining.

[23] The most popular and widely used RBF is the Gaussian basis function [*Kisi*, 2004; *Shi et al.*, 2005; *Manrique et al.*, 2006]. The spread (i.e., the radius of influence, σ) of an RBF needs to be precisely fixed for optimum RBFN performance [*Shi et al.*, 2005; *Alp and Cigizoglu*, 2007] because it determines the radius of influence of the RBF neuron to which more than one input sample responds. The spread of the RBF neuron should be large so that the active input regions overlap enough for several neurons to always have fairly large outputs at any given moment. This makes the network function smoother and results in better generalization than a network having a small spread. However, the spread should not be so large that each input neuron effectively responds to one neuron of the hidden layer [*Haykin*, 1999; *Principe et al.*, 1999]. The optimum spread of RBF neurons is determined using μGA.

2.4. Back Propagation Neural Network

[24] The details of BPNNs can be found in work by *Hagan et al.* [1996] and *Haykin* [1999]. This paper only explains the geometry, processing neurons functions, and model parameters used in the network employed.

[25] Bipolar sigmoid activation functions (between -1 and 1), such as the hyperbolic tangent (see Appendix A for the equation), are most commonly used for neurons in hidden layers and produce better network performance in terms of convergence and central processing time than other activation functions [*Ray and Klindworth*, 2000; *Maier and Dandy*, 2000]. However, the output layer is normally provided with a linear activation function so that the output range is between $-\infty$ and ∞. This avoids remapping of the outputs. Therefore, the hyperbolic tangent sigmoid and the linear activation functions (see Appendix A for the equation) were used for neurons of the hidden and output layers, respectively. The Levenberg-Marquardt (LM) training algorithm (see Appendix A) was selected because (1) it has the faster convergence ability than the conventional gradient descent algorithms [*Principe et al.*, 1999; *El-Bakyr*, 2003; *Kisi*, 2004; *Cigizoglu and Kisi*, 2005; *Alp and Cigizoglu*, 2007], (2) it does not require a learning rate and momentum factor like the gradient descent algorithms [*Hagan et al.*, 1996; *Principe et al.*, 1999; *Alp and Cigizoglu*, 2007], and (3) in many cases it converges when other back propagation algorithms fail to converge [*Hagan and Menhaj*, 1994]. *Flood and Kartam* [1994], *Maier and Dandy* [1998, 2000], and *Sahoo and Ray* [2006a] reported that the use of more than one hidden layer provides greater flexibility and enables the approximation of complex functions with fewer neurons. Therefore, a two-hidden-layer BPNN is considered for this study. *Maier and Dandy* [1998, 2000] showed that optimal ANN predictive performance efficiency depends on the ratio of first-hidden-layer neurons (H_1) to second-hidden-layer neurons (H_2). Conversely, the predictive performance efficiency of a network is undermined by either overtraining (i.e., the number of epochs is higher than optimum, E_0) or undertraining (i.e., the number of epochs is lower than optimum, E_0) [*Maier and Dandy*, 2000; *Alp and Cigizoglu*, 2007]. Thus, the network geometry (i.e., H_1 and H_2) and epoch size (E_0) need to be optimized using μGA.

[26] In addition to BPNN geometry, BPNN efficiency is severely affected by (1) the selection of initial weights and (2) the training cessation criteria of the network. Little research has been conducted into finding good initial weights [*Principe et al.*, 1999; *Haykin*, 1999]. In general, the initial weight is implemented with a random number generator that provides a random value within the range of $-\alpha$ (lower boundary) and α (upper boundary) [*Principe et al.*, 1999; *Haykin*, 1999; *Maier and Dandy*, 2000], where α is a real number. *Maier and Dandy* [2000] emphasized that too large or too small an α value results in the cessation of training at suboptimal levels and the value of α is problem specific [*Alp and Cigizoglu*, 2007]. In this study, the initial weights randomly generated between -1 and 1 produced consistent results. Nevertheless, in order to overcome a set of poor initial weights, the BPNN is trained several times (50 times in this study) with different sets of initial weights. The set having the greatest ANN predictive performance efficiency in terms of R is kept for analysis. To prevent overfitting of connecting weights, the BPNN training stops when any of the following conditions occurs:

[27] 1. The maximum number of epochs (iterations) is reached.

[28] 2. The network's training MSE falls below or meets the threshold MSE.

[29] 3. The performance gradient (i.e., $\Delta\text{MSE} = \text{MSE}(t) - \text{MSE}(t-1)$) falls below the minimum gradient (10^{-10} in this study). If the performance gradient is below the minimum gradient, practically, network training does not improve the weight matrix. So, training is terminated and the network is restarted with a new set of weight matrices.

[30] 4. The MSE remains above the validation performances in terms of MSE continuously for a number of iterations. In a preliminary run, the MSE was found to be fluctuating around the validation MSE for a few iterations (5 to 15) before falling below the validation MSE. Therefore, to avoid undertraining, the maximum number of iterations of which the MSE can remain above the validation MSE was set at 50 (determined in section 5.3 through sensitivity analysis).

[31] 5. The scalar number, ζ, used in the LM algorithms (see Appendix A) exceeds the maximum ζ set in the model

Table 1. Actual, Range-Specific, and Descriptive Input Parameters for the ANN Model

Item	Parameter	Type of Data Collected	Actual/Range-Specific/Descriptive Data Arranged in Recognized Clusters for ANN Modeling	Model Value
1	Well depth	Actual depth, m	<7.6	4
			7.6–15.2	3
			>15.2	2
			Unknown	1
2	Depth to aquifer material	Actual depth, m	≤1.5	4
			>1.5–6.1	3
			>6.1–15.2	2
			>15.2	1
3	Age of well	Actual year of excavation	Year ≤ 1936	4
			1936 < Year ≤ 1956	3
			1956 < Year ≤ 1976	2
			<1976	1
4	Distance to cropland	Actual distance or range-specific data, m	<6.1	4
			6.1–15.2	3
			15.2–30.5	2
			>30.5	1
5	Distance to barnyard	Actual distance or range-specific data, m	<15.2	4
			15.2–30.5	3
			30.5–61	2
			>61	1
6	Distance to septic systems	Actual distance or range-specific data, m	<15.2	4
			15.2–30.5	3
			30.5–61	2
			>61	1
7	Flush windows (time between pesticide application and first significant storm over 25 mm d^{-1})	Actual days or range-specific data, d	<3	4
			3–10	3
			10–20	2
			>20	1
8	Distance to streams or other contaminant sources	Rang-specific data, m	≤30.5	2
			>30.5	1
9	Well-site topography	Descriptive data	Level land	4
			Hill top	3
			Depression	2
			Hill slope	1
10	Season of sample collection	Descriptive data	Fall	4
			Winter	3
			Spring	2
			Summer	1
11	Presence of irrigation well	Descriptive data	Yes	2
			No	1
12	Spill or disposal site	Descriptive data	Yes	2
			No	1
13	On-site pesticide storage	Descriptive data	Yes	2
			No	1
14	Presence of animals	Descriptive data	Yes	2
			No	1
15	Aquifer class	Descriptive data	Sand/gravel	2
			bedrock	1
16	Pesticide leaching	Actual concentration (μg/L)	Real index value (in the range 0.1 to 10.000)	Index value

(10^{10} in this study). When ζ is large, LM becomes the gradient descent with a small step size; contrarily, when ζ is small, LM is the same as the Gauss-Newton method. Since the Gauss-Newton method converges faster and more accurately toward the threshold MSE than does the gradient descent method, the goal is to shift toward that method as quickly as possible. Thus, ζ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step increases the performance function. In this way, the MSE diminishes in each iteration. In this study, the initial ζ was set to 0.001, and it was decreased or increased by a factor of 10 in each iteration (t).

3. Data Used

[32] The data used in this study are the same as used in *Mishra et al.* [2004]. A summary of the data used is provided in Table 1. The input data, such as well depth, depth to aquifer material, and distance from well to cropland, do not hold any linear relationship between input parameters and observed pesticide concentration; rather, the data are recognized in clusters. There is a general trend of

shallower wells (e.g., <7.6 m) being more vulnerable to pesticide contamination than deeper wells (e.g., >15.2 m). Such heterogeneity of the data set can be classified using given patterns by recognizing the cluster of responses (observed pesticide occurrence) to perturbing inputs. A generic procedure to utilize a set of categorical input parameters for predictive purposes can be found in work by *Ray and Klindworth* [2000], *Brodnjak-Vonèina et al.* [2002], *Mishra et al.* [2004], and *Sahoo et al.* [2005, 2006]. For example, shallower wells are more prone to higher pesticide contamination and vice versa. Thus a numeric value for the actual input value of a well is assigned depending on the lower and upper bounds of observed clusters (e.g., for four clusters of well depths: 4 for shallow well, 1 for deeper well, and 2 and 3 for depths in between, see Table 1). Using a similar approach, the measured values of all input parameters are categorized into different classes, as shown in Table 1. There are 15 types of input parameters in this study, all of which are primarily of three types in terms of data collected: (1) exact (parameters 1 to 3), (2) exact or range-specific (parameters 4 to 8), and (3) descriptive (parameters 9 to 15). The pesticide concentration index is used as the model target value.

4. Methodology

4.1. Data Division Using Random Selection Method

[33] The bootstrap technique [*Efron and Tibshirani*, 1998; *Schaap and Leij*, 1998; *Chrysikopoulos et al.*, 2002] was used to randomly choose samples from the original data set containing N number of samples for the training, testing, and validation subsets. In the algorithm, an array of zeros equal to N is generated. A random integer a is generated by taking the integer part of the real number generated by a random real number between 0 and 1 and multiplied by N. The zero corresponding to the ath spot in the matrix is replaced by adding 1. The procedure is carried out N times. Since the value of the ath position can be replaced several times, each sample has a chance of being replaced once or multiple times for a particular data set. The matrix containing the zeros is then placed alongside the original data set. Samples corresponding to zeros are selected for the training subset, while the other samples are randomly divided into the testing and validation subsets. Since, the ath position can be selected more than once for replacement, the ath value is equal to the number of selections/replacements. Thus, random selection for the validation and testing subsets is carried out on the basis of different numbers (e.g., 2 for testing and greater than 2 for validation). Penalty constraints are added to ensure that the maximum and minimum values are in the training subset.

4.2. Data Division Using SOM Clustering

[34] SOM is trained to cluster the whole data set. Two samples from each cluster are selected, one for the training subset and another for the validation subset, while other samples are placed in the testing subset. In the instance of a cluster containing only one record, the record is placed in the training subset. However, if a cluster contains two records, one record is placed in the training subset and the other in the validation subset.

4.3. Estimation of ANN Performance Efficiency

[35] The performance efficiency of the network is estimated by comparing the measured and ANN-estimated values. The ANN performance measures used in this study are the correlation coefficient (R), mean error (ME), root mean square error (RMSE), and MSE. The mathematical expressions of R, ME, RMSE, and MSE can be found in work by *Jain and Srinivasulu* [2004] and *Sahoo and Ray* [2006a]. Briefly, the ANN predictions are optimum if R, ME, RMSE, and MSE are found to be close to 1, 0, 0, and 0, respectively. In the present study, MSE is only used for the estimation of network training performance, whereas R, ME, and RMSE are used to measure the predictive performance of ANN on the testing subset, which is independent of ANN network training and validation.

4.4. Selection of μGA Model Parameters

[36] *Krishnakumar* [1989], *Carroll* [1996], and Carroll (1999, available at http://www.cuaerospace.com/carroll/ga.html) suggested using a population size of 5 for μGA model. *Abu-Lebdeh and Benekohal* [1999] reported that μGA performs best for a population size around or above the square root of the string length. The string length of a parameter is estimated using the equation $\psi = (XX_{m,\max} - XX_{m,\min})/(2^{\Upsilon} - 1)$ presented by *Goldberg* [1989, p. 82]. The symbol ψ represents the accuracy of the search parameter, $XX_{m,\max}$ and $XX_{m,\min}$ are the maximum and the minimum values of mth parameter, respectively; and superscript Υ is the bit size of the mth parameter. We need to optimize three parameters (i.e., H_1, H_2, and E_0) for BPNN; so, the square root of the string length (e.g., 27 bits for three parameters at 9 bits per parameter) is around 5. Note that each bit represents either 0 or 1 for the binary μGA. We used a population size equal to 10. *Abu-Lebdeh and Benekohal* [1999] reported using a binary tournament selection, 0.5 uniform probability crossover rate (*Pcross*), and no mutation. *Carroll* (1999, http://www.cuaerospace.com/carroll/ga.html) suggested using binary tournament selection with shuffling and a uniform crossover rate of 0.5, whereas *Krishnakumar* [1989] recommended using a crossover rate of 1.0 and a mutation rate equal to 0. *Ines and Honda* [2005] reported using binary tournament selection with shuffling, a uniform *Pcross* equal to 0.5, a probability of creep mutation (*Pcreep*) equal to 0.1, and 150 generations. However, *Wardlaw and Sharif* [1999] pointed out that the value of uniform *Pcross* is problem-specific. Therefore, we used binary tournament selection with shuffling and performed a sensitivity analysis for uniform *Pcross*, *Pcreep*, and number of generations for evolution of optimum R.

4.5. μGA-ANN Model Development

[37] The primary objective of μGA-ANN model development is to optimize the ANN's geometry and model parameters so that the differences between ANN-estimated output and measured target values are minimized. Since only an optimized trained network can predict output values close to measured targets and R value increases as the differences between measured and ANN-estimated values decreases, R value estimated on the testing subset is used as the fitness value in the optimization processes using μGA. The procedure for searching an optimal network's geometry and model parameter(s) using μGA involves interexchanging the information (solution set of μGA to ANN and ANN-

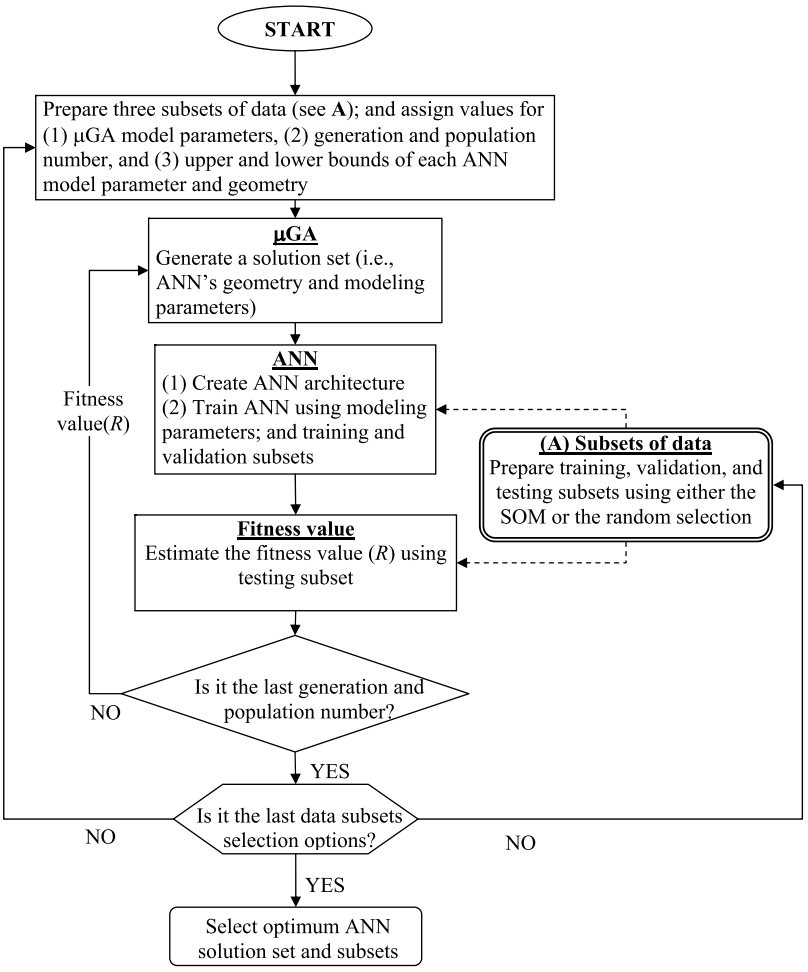

Figure 2. Flowchart for searching the three subsets of data having similar populations and for optimization of the ANN's geometry and modeling parameters.

estimated R value to μGA) between μGA and ANN (see Figure 2).

[38] Values of the lower and upper bounds of each ANN model parameters and geometry are fixed according to the number of samples included in the training subset. Thus, the data preparation techniques are the outermost loop in the flowchart. The μGA model parameters are assigned according to the ANN model to be optimized and subsets of data prepared using either the SOM technique or the bootstrap technique. The μGA generates one set of solutions (i.e., σ and N_0 for RBFN and H_1, H_2, and E_0 for BPNN). The solution set is passed on to the ANN. Using the solution set from the μGA, an ANN geometry is created and the ANN is trained using the training and validation subsets prepared using either the SOM clustering or the bootstrap technique. The performance efficiency (R) of the trained network is estimated using the testing subset that is unused during the training processes, and the R value is passed on to μGA to generate another set of solutions for the ANN that is supposed to produce better R. Thus, the interexchanging of information between μGA and ANN, referred to here as the μGA-ANN model, quickly eliminates weaker solutions and upholds/produces a solution set for the ANN that produces optimum R. The procedure continues until the last generation and population number for one set (consisting of three subsets) of data. The optimization of ANN geometry and modeling parameters for another set of data are performed as described above. The procedure continues for all sets of data prepared using either the SOM or the bootstrap technique. The three subsets, prepared from the original data set, producing the highest R value is considered to be having similar populations and the corresponding ANN model is the optimum. Thus, the objective function of the μGA-ANN combination is to maximize ANN-estimated R on the basis of the testing subset. Mathematically,

$$\max \quad R = \frac{\sum_{i=1}^{N} m_i p_i}{\sqrt{\sum_{i=1}^{N} m_i^2}\sqrt{\sum_{i=1}^{N} p_i^2}} \tag{7}$$

where $m_i = M_i - \overline{M}$, $p_i = P_i - \overline{P}$, and M_i and P_i are the measured and predicted values for $i = 1, \ldots, N$, and $\overline{M}$ and $\overline{P}$ are the mean values of the measured and predicted data sets, respectively.

[39] To prevent the generation of unrealistic solution sets by μGA, a few preliminary runs were made to determine realistic lower and upper boundaries of each model

Table 2a. Total 572 Samples Divided Into 91 Clusters in 100-Neuron (10 × 10 Grid) Square SOM Networks[a]

	A	B	C	D	E	F	G	H	I	J
j	9	8	9	13	4	11	3	7	6	9
i	3	13	1	3	1	13	8	6	4	11
h	12	6	5	5	4	9	6	3	4	7
g	14	5	5	4	1	15	1	11	0	5
f	0	0	0	3	4	0	2	5	3	6
e	10	7	0	8	4	7	2	0	0	7
d	7	3	13	1	2	5	4	2	3	5
c	10	7	7	4	6	4	8	10	0	4
b	3	7	6	6	9	1	7	3	3	7
a	12	10	12	6	8	10	3	10	4	8

[a]The combination of lowercase and capital letters indicates the position of the Kohonen neuron of each square SOM network.

parameter for ANN. Thus, equation (7) is subjected to the following constraints:

$$\text{For RBFN} \begin{cases} \sigma_{\min} \le \sigma \le \sigma_{\max} \\ N_{\min} \le N_0 \le N_{\max} \end{cases} \tag{8}$$

$$\text{For BPNN} \begin{cases} H_{1,\min} \le H_1 \le H_{1,\max} \\ H_{2,\min} \le H_2 \le H_{2,\max} \\ E_{\min} \le E_0 \le E_{\max} \end{cases} \tag{9}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are the minimum and the maximum spreads of an RBFN, respectively; $N_{\min}$ and $N_{\max}$ are the minimum and the maximum number of RBF neurons in the hidden layer; $H_{1,\min}$, $H_{2,\min}$, and $E_{\min}$ are the minimum neurons of the first hidden layer, the minimum neurons of the second hidden layer, and the minimum epoch number of a BPNN, respectively; and $H_{1,\max}$, $H_{2,\max}$, and $E_{\max}$ are the maximum neurons of the first hidden layer, the maximum neurons of the second hidden layer, and the maximum epoch number of a BPNN, respectively.

[40] The data set consists of 15 input parameters and 1 output parameter (i.e., pesticide concentration index). Therefore, $H_{1,\max}$, $H_{2,\max}$, and $E_{\max}$ are set to 25, 25, and 1000, respectively. The present study sets $H_{1,\min}$, $H_{2,\min}$, and $E_{\min}$ to 1, 1, and 5, respectively. Because $\sigma_{\max}$ is problem-specific, it is set to a higher value (80 in this study). The $\sigma_{\min}$ is set to 0.1. Since each RBF neuron must respond to at least one input sample, $N_{\max}$ cannot exceed the number of input samples [*Haykin*, 1999; *Principe et al.*, 1999]. Thus, $N_{\min}$ and $N_{\max}$ are set to 1 and the number of input samples of the training subset, respectively. The model parameters σ, N_0, H_1, H_2, and E_0 are optimized using a μGA.

4.6. Data Preprocessing

[41] Repetitive samples (i.e., two or more samples each having identical input values for a different target value) undermine ANN training [*ASCE Task Committee*, 2000]. Because of the categorical inputs (e.g., 1 to 4 instead of the actual value, see Table 1) for all 15 input parameters and pesticide concentration index as the target, there are few repetitive samples in the data set. All repetitive samples are replaced by one input sample and the average target value (i.e., the average pesticide concentration index) of all similar samples, thus reducing the sample size from 631 to 572.

[42] The data set is scaled to the range of 0 to 1. According to *Bowden et al.* [2002], scaling to this range has two advantages: (1) inputs with much larger values are prevented from dominating the ANN training process, and (2) penalty constraints can be included more easily (i.e., the maximum and minimum values can be identified by the μGA as 1 s and 0 s). The training, testing, and validation subsets are scaled to the range of 0 to 1 using the equation $x_{ni} = (x_i - x_{\min})/(x_{\max} - x_{\min})$, where x_i is the input value, x_{ni} is the scaled input value of the real-world input value x_i, and $x_{\max}$ and $x_{\min}$ are the respective maximum and minimum values of the unscaled data set. The network-estimated output values, which are in the range of 0 to 1, are converted to real-world values using the equation $x_i = x_{ni}\ (x_{\max} - x_{\min}) + x_{\min}$.

5. Results and Discussion

5.1. Data Clustering Using SOM

[43] A SOM of 100 neurons (10 × 10 grid) was trained using 50,000 iterations. SOM training assigned each input sample a number (1 to total Kohonen neurons, e.g., 100) on the basis of distance from the winning neuron. If nine samples clustered in one SOM neuron, then their assigned numbers or distances from the winning number would be the same. In other words, samples having the same number are considered to have similar characteristics of influence on pesticide occurrence. All 572 samples clustered in 100 Kohonen neurons are presented in Table 2a. Table 2a shows that only 91 clusters were formed for 572 samples and none of the samples was grouped into 9 Kohonen neurons. The number of samples grouped in a cluster is represented by the number indicated in that particular Kohonen neuron in the lattice (Table 2a).

[44] To examine the number of clusters for a number of SOM neurons other than 100, the SOM of 64 neurons (8 × 8 grid) and 144 neurons (12 × 12 grid) are trained using 32000 and 72000 iterations, respectively. All 572 samples were grouped into 58 clusters in the case of the 64-neuron SOM (Table 2b) and into 132 clusters in the case of the 144-neuron SOM (Table 2c). No sample was grouped into 6 and 12 Kohonen neurons of the 64-neuron SOM and the 144-neuron SOM, respectively. The reason for obtaining a different cluster number for each of the three different SOMs is that samples are different from each other on the

Table 2b. Total 572 Samples Divided Into 58 Clusters in 64-Neuron (8 × 8 Grid) Square SOM Networks[a]

	A	B	C	D	E	F	G	H
h	20	4	12	4	11	16	0	12
g	16	9	7	6	3	0	12	5
f	9	3	15	8	12	0	2	5
e	9	3	13	3	4	13	3	10
d	22	6	16	0	8	9	2	11
c	11	10	0	15	8	12	15	7
b	6	4	16	0	11	13	7	10
a	13	5	0	21	11	15	7	32

[a]The combination of lowercase and capital letters indicates the position of the Kohonen neuron of each square SOM network.

Table 2c. Total 572 Samples Divided Into 132 Clusters in 144-Neuron (12 × 12 Grid) Square SOM Networks[a]

	A	B	C	D	E	F	G	H	I	J	K	L
l	14	4	10	0	4	2	5	3	1	3	1	4
k	9	2	3	5	1	2	0	4	0	2	3	8
j	13	3	2	0	2	4	1	2	6	5	4	5
i	6	2	3	6	3	2	5	1	1	2	1	10
h	6	4	7	1	6	0	9	2	2	4	5	4
g	5	6	2	7	3	0	3	3	5	5	7	5
f	6	3	1	0	2	1	4	5	6	2	3	1
e	3	7	3	5	8	1	4	6	3	7	2	6
d	5	2	4	4	6	3	3	8	3	3	2	5
c	5	4	3	3	4	0	3	11	3	7	0	7
b	4	2	0	3	4	1	0	5	5	0	3	1
a	2	6	3	10	5	10	0	9	2	16	4	10

[a]The combination of lowercase and capital letters indicates the position of the Kohonen neuron of each square SOM network.

basis of their effects (i.e., characteristics) on pesticide concentration. Samples having similar effects on the occurrence of pesticide concentration are clustered in one neuron. Thus, a large SOM produces more clusters, narrowing down the differences in characteristics of samples in a cluster (see equation (4)) and vice versa. The training, testing, and validation subsets are prepared from SOM-produced clusters, as described in section 4.2. The training subset is checked to ensure that samples having the maximum and minimum pesticide concentrations are present as penalty constraints. The numbers of samples in the training data sets become 58, 92, and 132, for the 64-, 100-, and 144-neuron SOMs, respectively. The numbers of samples in the validation data sets become 57, 85, and 116 for the 64-, 100-, and 144-neuron SOMs, respectively. Thus, out of 572 samples, numbers of samples left for the testing data sets are 457, 394, and 324 samples for the 64-, 100-, and 144-neuron SOMs, respectively.

5.2. Sensitivity Analysis of μGA Parameters

[45] The sensitivity of *Pcross* and *Pcreep* values is demonstrated in Figure 3 using μGA-RBFN and 92 training samples of 100-neuron SOM. In total, four cases (see Figure 3) using four combinations of *Pcross* and *Pcreep* values were examined. Although the R value of μGA-RBFN was found to be optimum for *Pcross* and *Pcreep* equal to 0.5 and 0.0 (case 2 in Figure 3), respectively, the difference between the optimum R of case 2 and that of case 3 is only 0.001. The *Pcross* and *Pcreep* values of case 2 were found to be consistent with those of *Carroll* (1999, http://www.cuaerospace.com/carroll/ga.html), *Abu-Lebdeh and Benekohal* [1999], and *Ines and Honda* [2005] and are used in the rest of the analysis.

5.3. Sensitivity Analysis of ANN Geometry and Model Parameters

[46] Using subsets prepared from 10 × 10 grid SOM clusters, the μGA-ANN model generates a set of solutions (i.e., 1000 combinations of σ and N_0 in 100 generations) for RBFN and another set of solutions (i.e., 500 combinations of H_1, H_2 and E_0 in 50 generations) for BPNN. All combinations were plotted against their respective ANN R values in Figure 4. Figure 4a shows RBFN-estimated R values for respective combinations of σ and N_0. For clarity, the E_0 parameter generated alongside H_1 and H_2 for BPNN is not shown in Figure 4b. The wire mesh surface created by linear interpolation of R values shows the variation of R values for the combinations of ANN parameters. The distinguishing hill top and valley portions show high and low R values for the combination of corresponding ANN parameters. Figure 4 illustrates that the ANN-estimated R value is sensitive to ANN's geometry and model parameters and that the optimum combinations lie in a narrow range.

[47] Training should be stopped at the iteration where the MSE of the training subset is more than the MSE of the validation subset. The idea of stopping the training process is justified because the ANN begins to overtrain after that point. However, it is seen that the MSE of the training subset falls below the MSE of the validation subset after a few iterations. Thus, setting a small value (e.g., 5) for the maximum number of failed iterations allowed (MFA) may undertrain the network and vice versa. In such cases,

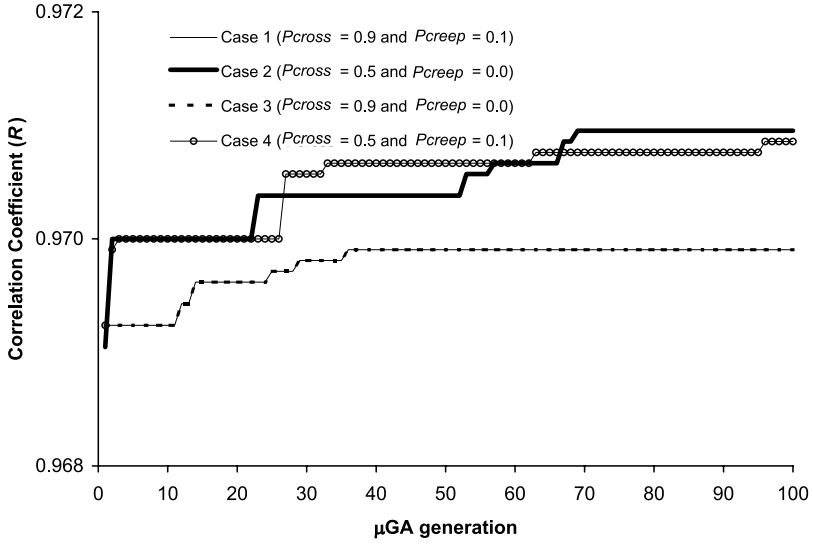

Figure 3. Effects of *Pcross* and *Pcreep* on the evolution of optimum RBFN predictive performance efficiency (R) using the 10 × 10 grid SOM data set. Lines for case 1 and case 3 overlap.

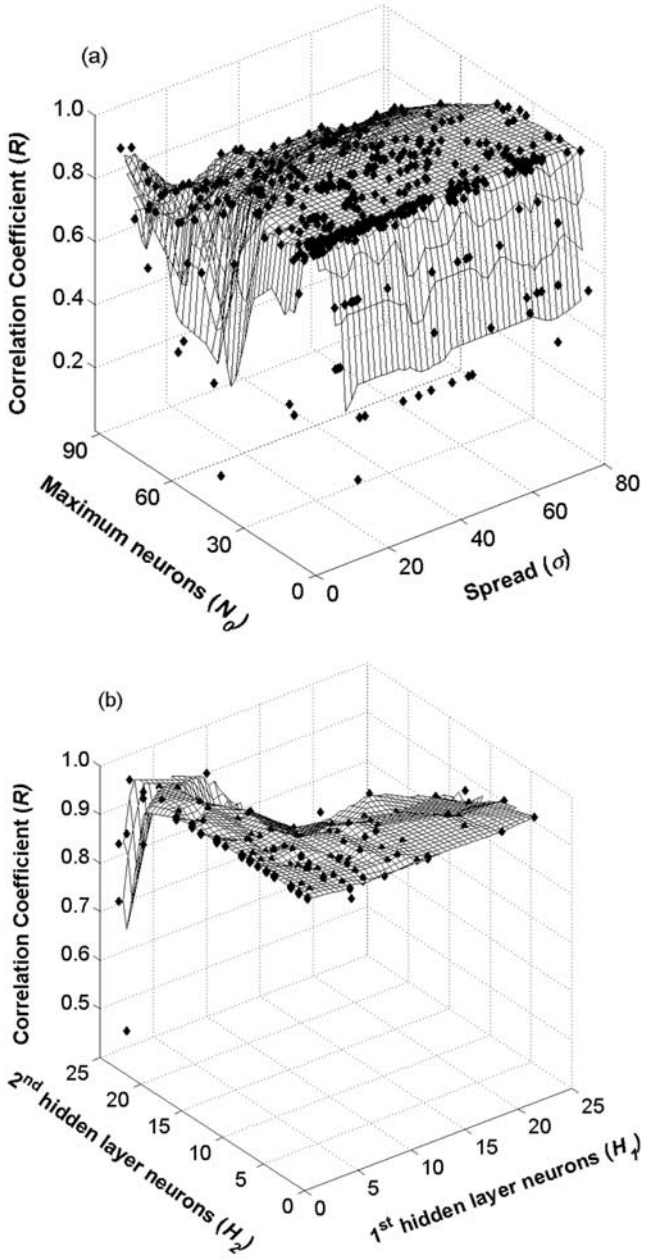

Figure 4. Linearly interpolated topography of R values of (a) 1000 sets of N_0 and σ generated by μGA in 100 generations for RBFN and (b) 500 sets of H_1 and H_2 generated by μGA in 50 generations for BPNN. The epoch numbers (E_0) generated for BPNN, along with values of H_1 and H_2, are not shown in Figure 4b for clarity. Solid diamonds represent the actual ANN-estimated R values.

different possible scenarios for setting MFA are examined. In all of the above mentioned cases, the MFA is set at 15 for the RBFN. Therefore, RBFN is examined for four cases, as shown in Figure 5. Although the evolution of highest R values (Figure 5a) of all four cases (MFA = 0, 5, 15, and 50) considered here are near to each other for the 92-sample training subset, the highest R value for the case of 58-sample training subset is obtained at MFA equal to 15. Therefore, for the rest of the study, the MFA value was fixed at 15 for the RBFN.

[48] The sensitivity of MFA for the BPNN is shown in Figure 6. The R values for all four cases examined here finally converge between 20 and 30 generations. However, the evolution of optimum R value is achieved earlier (i.e., on 6th generation) for the case of MFA = 50 than for other cases (i.e., 23rd, 30th, and 8th generation for cases where MFA equals to 5, 15, and 100, respectively). The optimum R values for all cases are almost the same (R = 0.9968, 0.9966, 0.9967, and 0.9964 for MFA equal to 5, 15, 50, and 100, respectively). Figure 6 shows that the optimum value for some cases may not be achieved in the 30 generations for MFA equal to 5 or 15. Setting a value greater than 30 for the generation number in the μGA-BPNN model takes more computer CPU time to complete a model run. Note that a Pentium 4, 1.8 GHz, and 512 MB RAM computer takes nearly 24 h to complete one run consisting of 30 generations. Thus, the MFA is set at 50 for the BPNN.

5.4. Sensitivity Analysis Using Different SOM Clusters

[49] Three sets of training, validation, and testing subsets prepared from clusters of the three SOM sizes are used to examine effects of the number samples in the training subset on RBFN and BPNN prediction performance efficiency in terms of R. Figure 7 presents the evolution of optimum R using the μGA-RBFN. Figure 7 indicates that the optimum R value increases as the number of samples in the training subset increases. Similarly, Figure 8 shows that the R values of μGA-BPNN for 92 and 132 training samples for 50 generations vary within a close range, while the R values of μGA-BPNN for 58 samples are lower than that of the other two training subsets. Note that increasing the maximum number of generations (e.g., to 150) for the μGA-RBFN does not improve the R value. On the basis of Figures 7 and 8, the maximum number of generations for μGA-RBFN and μGA-BPNN are set to 100 and 30, respectively.

[50] In Figures 7 and 8, the R values are low at the beginning because the ANN's geometry and model parameters are not optimum. As the μGA generations proceed, the solution set (i.e., the combination of ANN's geometry and model parameters) tends toward optimum. The fluctuations of R values for the BPNN result in because of different weight matrices of different neural networks. Note that initial weight matrices generated using a random number generator and μGA-generated epoch number for BPNN training are different for each case (i.e., solution set). Nevertheless, the standard deviation (SD) of R values for 50 generations is less than 0.3% (i.e., SD for 8 $\times$ 8, 10 $\times$ 10, and 12 $\times$ 12 grid SOM are 0.0024, 0.0012, and 0.0006, respectively). This indicates that the fluctuations in R values are minimized for the training subset containing a higher number of samples and vice versa. This infers that a training subset containing a larger number of samples provides more detailed information to the network.

5.5. Effects of Data Sets Prepared Using SOM and Random Selection Method on *R*

[51] Using the bootstrap technique for randomly assigning sample into the three subsets, it is found that the total samples in the training, validation, and testing subsets are 264, 97, and 211, respectively. To examine the effects of the number of samples in the training subset on the ANN-estimated R value, the number of training samples is reduced from 264 to 58, 92, and 132 by taking off the

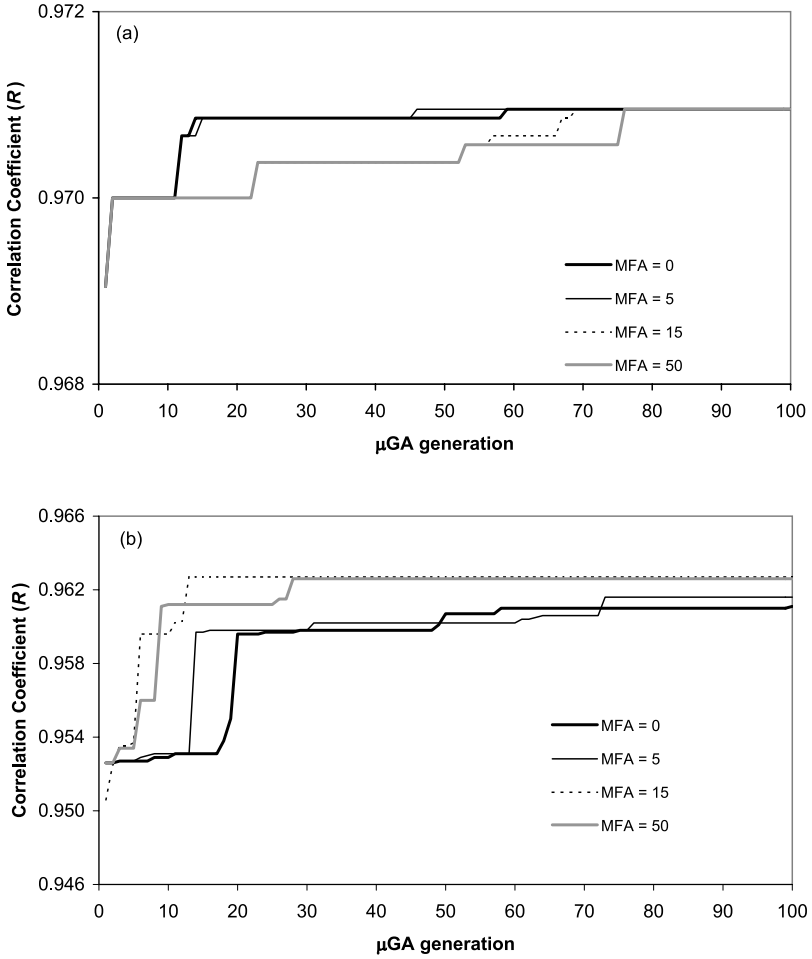

Figure 5. Effects of maximum number of iterations allowed where the MSE of the validation subset continuously remains above the MSE of the training subset (i.e., MFA is maximum failed iterations allowed) on the evolution of RBFN predictive performance efficiency (R) using (a) 92 training samples of a 10 × 10 grid SOM and (b) 58 training samples of an 8 × 8 grid SOM.

tail-end samples. The deleted tail-end samples are added into the testing subset. The purpose of preparing training subset size equal to that of three SOMs is to compare the respective ANN-estimated R values.

[52] The R values produced by the μGA-RBFN model using the SOM-clustered subsets and random selection subsets are presented in Figure 9. The R values of the μGA-RBFN model using SOM-clustered 92 and 132 samples in the training subsets (Figure 9a) are higher than those of the μGA-RBFN model using randomly selected data sets (Figure 9b). The low R values of the μGA-RBFN model using 58 SOM-clustered samples in the training subset can be attributed to a fewer number of training samples available for the RBFN to achieve adequate generalization ability.

[53] Figure 7 shows that the μGA-RBFN produced a higher R value for the training subset with a larger number of samples. Therefore, a similar trend was expected for the training subsets prepared using random sample selection method in Figure 9b. However, the R value (0.9692) was found to be highest for the training subset containing 132 samples. For the other three cases, a general trend was found, that is, a training subset consisting of a larger number of samples evolves with a higher R value. This discrepancy clearly indicates that the three subsets are not of the same population or properties. Particularly, the training and validation subsets include samples mostly from densely clustered regions. Thus, samples from clusters having one or two samples are underrepresented in the network training.

5.6. Effects of Selecting More Than One Sample From Each Cluster for Subsets on R

[54] Figures 9 and 10 indicate that using a larger number of clustered samples in the training subset increases the predictive performance efficiency of ANN. This advocates increasing the number of Kohonen neurons in the SOM. As seen in Tables 2a, 2b, and 2c, if the number of Kohonen neurons (i.e., clusters) increases, the number of samples clustering in each Kohonen neuron decreases. Kohonen neurons having only zero-, one-, and two-input samples in the 8 × 8, 10 × 10, and 12 × 12 grid SOMs are 6, 0, and 2; 9, 6, and 4; and 12, 14, and 21; respectively. Therefore, increasing the number of SOM neurons generates more clusters each having one or two samples of unique characteristics resulting in greater discrepancy among training, testing, and validation subsets. On the other hand, decreasing the number of Kohonen neurons (e.g.,

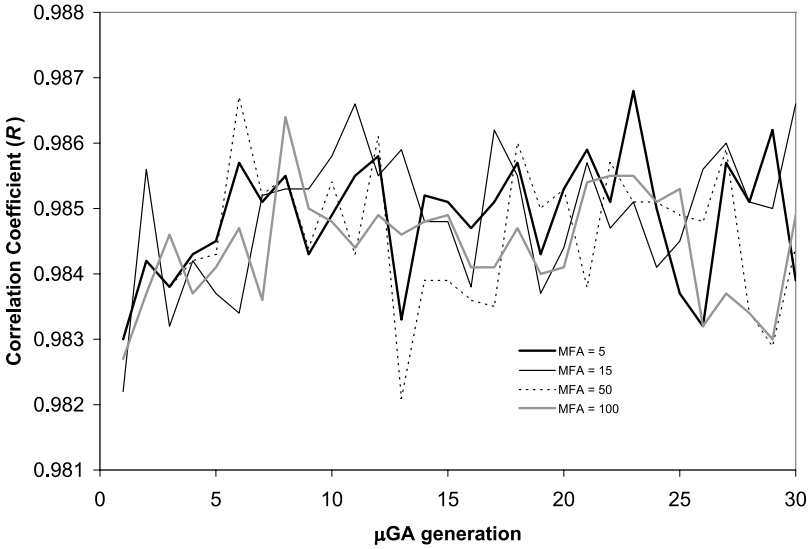

Figure 6. Effects of maximum number of iterations allowed where the MSE of the validation subset continuously remains above the MSE of the training subset (i.e., MFA is maximum failed iterations allowed) on the evolution of BPNN predictive performance efficiency (R) using 92 training samples of a 10×10 grid SOM.

8×8 grid SOM) means generating a small number of clusters each having samples of wider characteristics because σ_s in equation (4) is a fixed value. Thus, using an extremely large or small SOM is not useful for finding an optimal training subset; rather, a tradeoff is more useful. There is no straightforward solution for finding a suitably sized SOM. Thus, the case of inclusion of more than one sample per cluster in the training and validation subsets is examined below.

[55] The initial training and validation subsets include the first and second samples of each cluster, respectively. In the first increment, the third and fourth samples are placed in the training and validation subsets, respectively, and in the second increment, the fifth and sixth samples are placed in the training and validation subsets, respectively. If the number of samples available in a cluster is less than the required amounts for the training and validation subsets, first preference is given to the training subset. The sample selection process is continued for all clusters because selecting samples from only a few clusters will result in information discrepancies in subsets (particularly in the training and validation subsets). This generates two

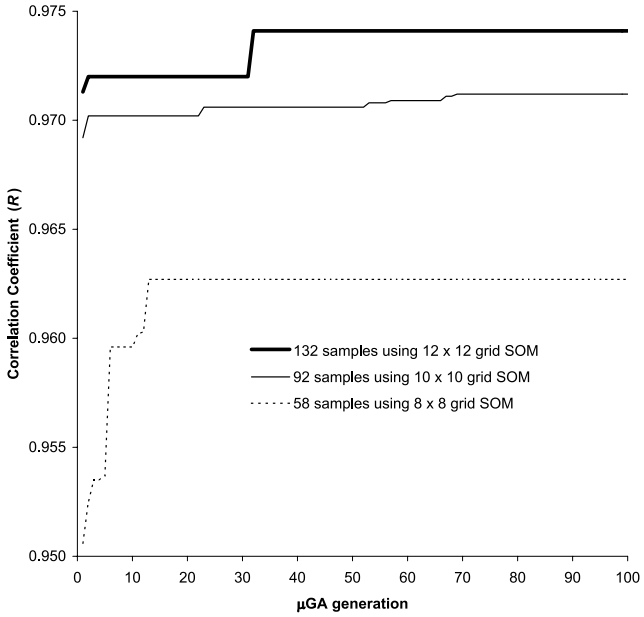

Figure 7. Effects of the number of samples included in the training subset (i.e., number of SOM clusters) on the evolution of optimum RBFN predictive performance efficiency (R).

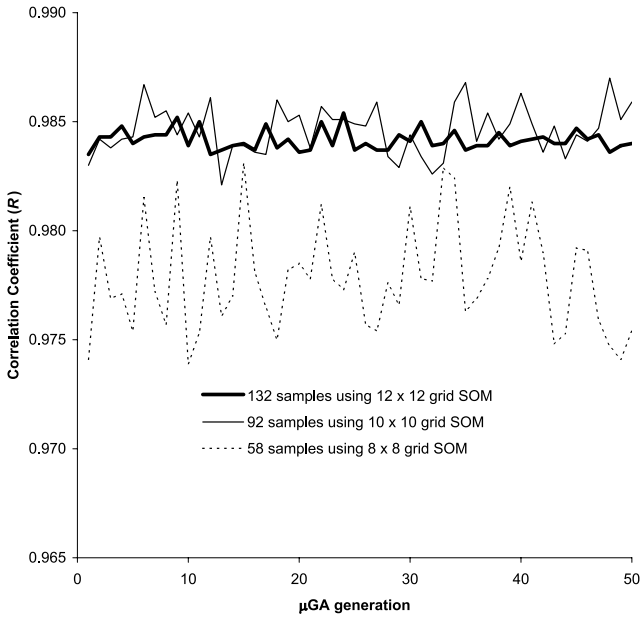

Figure 8. Effects of the number of samples included in the training subset (i.e., number of SOM cluster) on the evolution of optimum BPNN predictive performance efficiency (R).

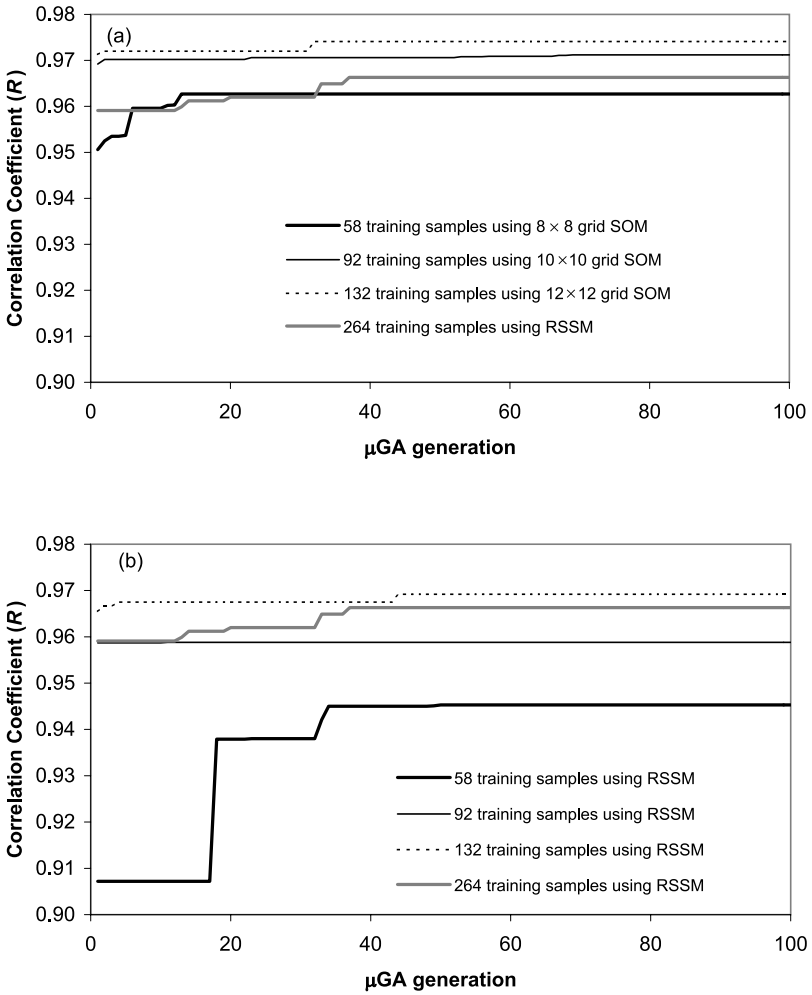

Figure 9. Evolution of RBFN predictive performance efficiency (R) for training subsets prepared using (a) SOM technique and (b) random sample selection method (RSSM).

additional data sets for the training and validation subsets. The ANN predictive performance efficiency for all data sets is examined using the μGA-ANN model.

[56] Comparing the results in Figure 9 with those in Figure 11 for the μGA-RBFN, it is clear that the training subset with two samples per cluster of 8 $\times$ 8 grid SOM outperforms all other data sets (also see Table 3). Similar results are obtained for the μGA-BPNN when comparing the results in Figure 10 with those in Figure 12. The R values of both the μGA-RBFN and μGA-BPNN for the training subset including three samples per cluster of the 12 $\times$ 12 grid SOM are found to be the worst, even though this training subset contains the highest number of samples. This is clearly attributed to the discrepancies of information in the three subsets, particularly in the training and validation subsets. Although the training subset contains samples from all clusters, the validation subset is void of samples from clusters representing one or two samples. Therefore, the validation subset, intended to prevent overfitting or undertraining, could not adequately help training the network. Contrary to this observation, the R values for the training subset including three samples per cluster of the 8 $\times$ 8 grid SOM is the second highest among all cases. It can be explained similarly that subsets consisting of two and three samples per cluster of the 8 $\times$ 8 grid SOM have less discrepancy among those of other two SOM data sets as the number of clusters consisting of one or two samples is very small. Therefore, the best way to select samples for the three subsets required by ANN is to find SOM-trained clustered data which have less number of clusters with single or two samples, and each of the subsets (particularly the training and validation subsets) should include more than one sample from each cluster.

[57] Two hypothesis tests: one to test if the standard deviations of two populations are equal (F test) and the other to test if the means of two populations are statistically different from each other (t test) were used. The greater the F test value deviates from 1, the stronger is the evidence for unequal population. In t test, the null hypotheses are examined for a significance level of $\alpha_1 = 0.05$. For each input and output variables, the validation and testing subsets were compared with the training subsets for F test and t test. Values of F test and t test for the cases (1) two samples per cluster of 8 $\times$ 8 grid SOM and (2) 264 training samples using RSSM are shown in Table 4 for comparison. The null hypothesis (acceptance range is -1.96 to $+1.96$) for t test were accepted for all cases except two cases of 264 training samples using RSSM and one case of 8 $\times$ 8 grid SOM (see

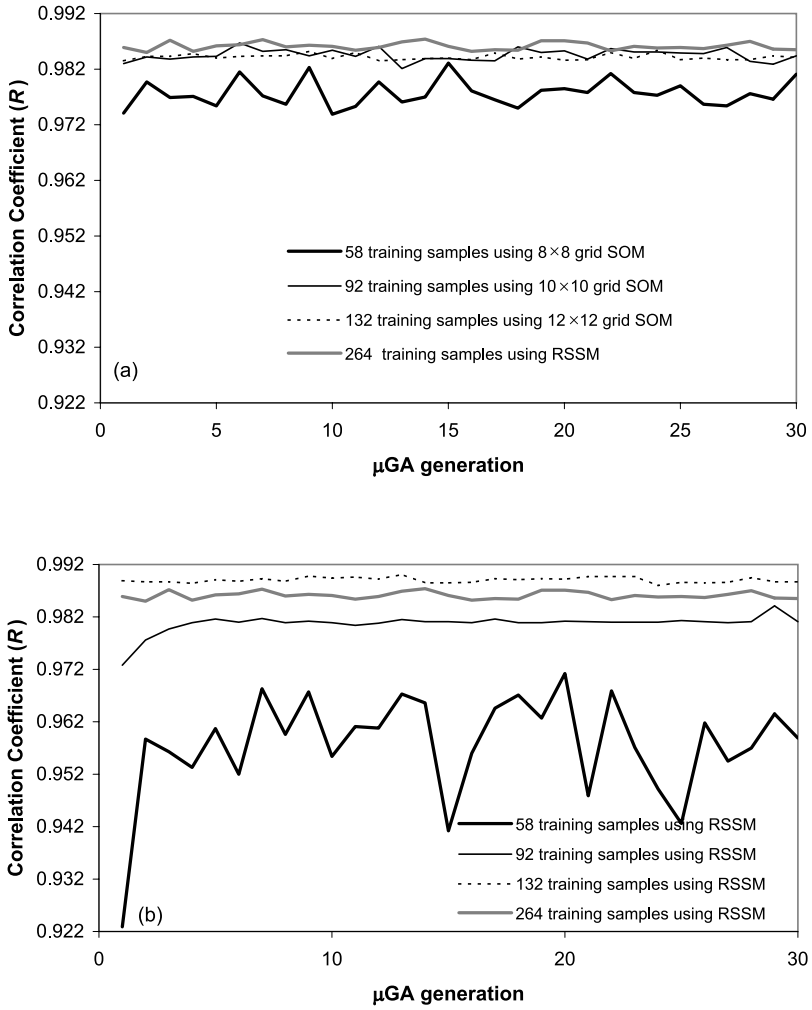

Figure 10. Evolution of BPNN predictive performance efficiency (R) for training subsets prepared using (a) SOM technique and (b) random sample selection method (RSSM).

Table 4). Also, Table 4 shows that overall, the statistics (i.e., values of F test and values of t test) of two samples per cluster of 8×8 grid SOM are in better agreement than those of 264 training samples using RSSM. Because of this, the R value of two samples per cluster of 8×8 grid SOM was found to be the highest among all other cases in Figures 11 and 12. For this reason, the R value of 264 training samples using RSSM is found to be high in Figures 11 and 12.

5.7. Comparison of Results With Previous Studies

[58] *Bowden et al.* [2002, p. 2.1] reported that "... an ANN model performs poorly, given that the poor performance is primarily related to the data themselves and not the choice of the ANN's parameters or architecture." The present study demonstrated using μGA-BPNN and μGA-RBFN that the ANN predictive performance efficiency significantly affects the choice of ANN's geometry and model parameter(s). Further, *Bowden et al.* [2002] used only one architecture of the SOM and made the training and validation data pools by selecting one sample from each cluster. It is demonstrated in this study that the number of clusters formed was dependent on the SOM architecture and the difference in characteristics of samples (or responses of samples to output) between two neighboring clusters decreases when the number of clusters increases. Also, they did not examine how many samples should be selected from each cluster for the optimum training and validation subsets.

[59] *Sahoo and Ray*'s [2006b] work is the first to optimize the ANN's geometry and model parameter(s) using a simple GA; however, they only used the training subset, but not the training and validation subsets in the ANN training process. In the absence of a validation subset, ANN training continues until the end of the assigned epoch size. This does not prevent a network from overtraining. ANN training ceases when the MSE of the validation subset starts to exceed the training MSE. To prevent premature cessation of the training processes, the present study introduces the MFA parameter which was not done previously. Also, in Sahoo and Ray's study, the whole data set was arbitrarily divided into training and testing subsets. The present study uses a μGA which is more robust than a simple GA, and performs sensitivity analysis for the μGA model parameters using the μGA-ANN model. To accommodate the validating subset in the RBFN training process, the present study modifies the RBFN subroutine. Thus, the present study not only integrates two previous independent studies but also significantly improves the methodologies in search of the three subsets of similar

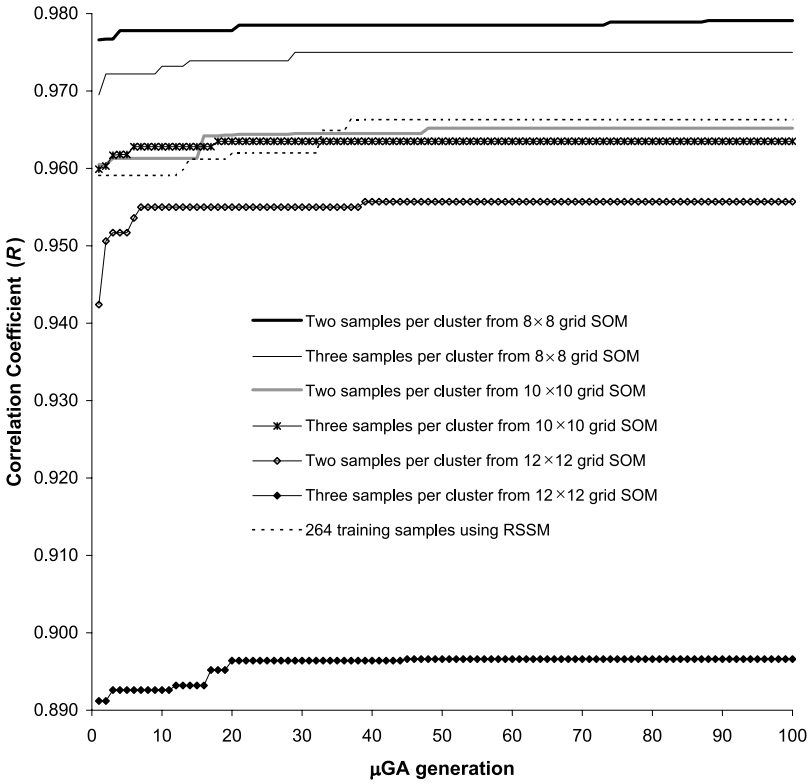

Figure 11. Evolution of RBFN predictive performance efficiency (R) using two and three samples per cluster of 8 × 8, 10 × 10, and 12 × 12 grid SOMs and random sample selection method (RSSM) in the training subsets.

populations, as well as optimum ANN's geometry and model parameters collectively.

6. Summary and Conclusions

[60] This paper presents a systematic method that answers the questions ''which samples'' and ''how many samples'' should be selected for the training, testing, and validation subsets required by ANN when total available samples are limited. Also, it investigates the use of μGA to develop a μGA-ANN model to search for optimal combinations of ANN's geometry and model parameters in the large solution space. Given an optimized network and model parameter(s), the ANN model using SOM as a data division technique outperformed the ANN model using random selection of samples as the data division technique. Two of the most commonly used ANN models, BPNN and RBFN, are

Table 3. Number of Samples in the Training, Validation, and Testing Subsets and the Highest Performance Efficiency Values of μGA-Optimized RBFN and BPNN[a]

Preparation of Subsets	Number of Samples				Performance Efficiency of RBFN			Performance Efficiency of BPNN		
	Training	Validation	Testing	Total	R	RMSE	ME	R	RMSE	ME
8 × 8 grid SOM	58	57	457	572	0.9627	0.2660	0.0625	0.9831	0.1700	0.0201
10 × 10 grid SOM	92	85	395	572	0.9712	0.2078	−0.0029	0.9870	0.1403	−0.0529
12 × 12 grid SOM	132	116	324	572	0.9741	0.1914	0.0165	0.9854	0.1431	−0.0018
8 × 8 grid SOM with one increment	113	107	352	572	**0.9791**	**0.1882**	**0.0266**	**0.9909**	**0.1266**	**0.0011**
8 × 8 grid SOM with two increments	159	150	263	572	0.9750	0.1980	0.0028	0.9896	0.1290	−0.0119
10 × 10 grid SOM with one increment	173	154	245	572	0.9652	0.1926	0.0164	0.9865	0.1385	0.0264
10 × 10 grid SOM with two increments	230	202	140	572	0.9635	0.1869	0.0058	0.9894	0.1289	−0.0104
12 × 12 grid SOM with one increment	228	186	158	572	0.9557	0.1525	0.0079	0.9861	0.0901	0.0049
12 × 12 grid SOM with two increments	280	220	72	572	0.8966	0.1704	0.0061	0.9786	0.0606	−0.1824
RSSM (264 samples)	264	97	211	572	0.9663	0.1752	−0.0194	0.9874	0.1811	−0.0006
RSSM (58 samples)	58	97	417	572	0.9453	0.3082	−0.1041	0.9712	0.2093	−0.0692
RSSM (92 samples)	92	97	383	572	0.9588	0.2483	−0.0412	0.9841	0.1824	−0.0861
RSSM (132 samples)	132	97	343	572	0.9692	0.2135	−0.0176	0.9901	0.1522	−0.0699

[a]RSSM represents random sample selection method for the three subsets. The bold values represent highest ANN predictive performance efficiency among all training subsets. Performance efficiency is highest in all generations.

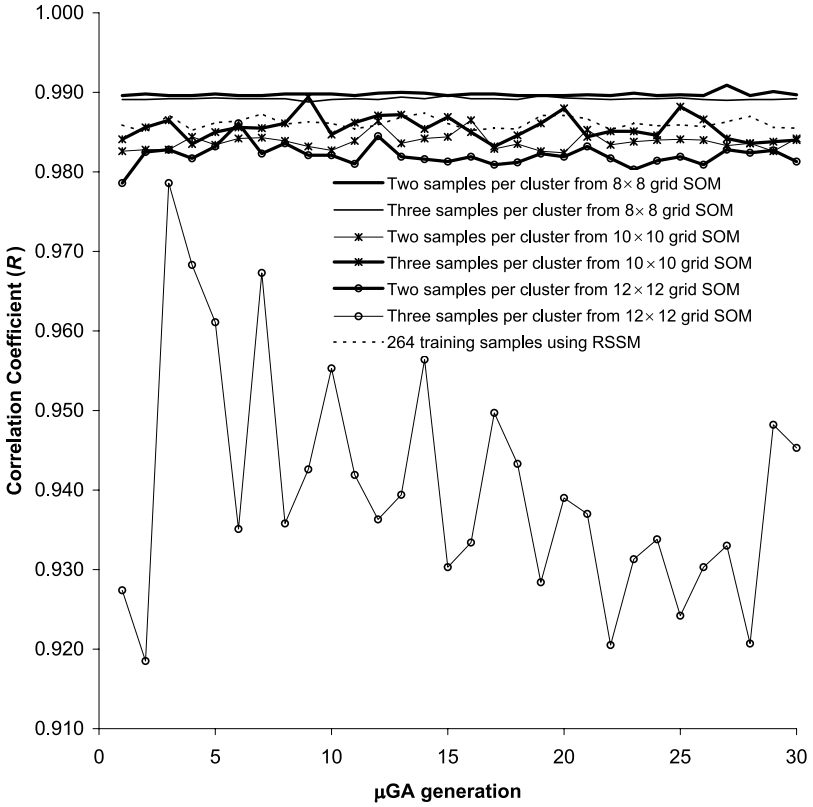

Figure 12. Evolution of BPNN predictive performance efficiency (R) using two and three samples per cluster of 8 × 8, 10 × 10, and 12 × 12 grid SOMs and random sample selection method (RSSM) in the training subsets.

employed to examine the effects of the proposed data division technique on the evolution of optimum predictive performance efficiency in terms of R.

[61] Sensitivity analysis was performed on model parameters of μGA in terms of R value before using them to optimize the ANN's geometry and model parameters. Although the optimum R value was found to be least sensitive to *Pcreep* and *Pcross* values of the parameters producing the optimum R value are used in the μGA-ANN model.

[62] The bootstrap technique is used to randomly select samples from the whole population for three subsets of data required by the ANN. However, random selection of

Table 4. Statistical Significance Tests (F Test and t Test) for the Input and Output Parameters[a]

	Two Samples per Cluster of 8 × 8 Grid Samples				264 Training Samples Using RSSM			
	Training Subset to Validation Subset		Training Subset to Testing Subset		Training Subset to Validation Subset		Training Subset to Testing Subset	
Input/Output Parameter	F Test	t Test	F Test	t Test	F Test	t Test	F Test	t Test
Well depth	0.97	−1.04	1.13	−0.89	1.91	−1.62	1.02	−0.59
Depth to aquifer material	1.22	−0.66	1.09	−0.87	0.98	−0.30	1.00	−0.50
Age of well	1.11	0.19	1.19	0.39	0.84	−1.28	1.10	0.35
Distance to cropland	1.12	0.43	0.91	0.46	0.94	−0.92	0.84	−1.09
Distance to barnyard	1.66	−0.12	1.59	−0.01	2.70	1.93	0.93	0.02
Distance to septic systems	1.16	0.85	1.35	0.65	1.01	−0.21	0.63	−1.96
Flush windows	0.91	−0.60	1.04	−0.92	0.93	0.62	1.04	−0.49
Distance to streams	1.36	0.73	1.58	1.47	0.88	0.07	0.75	−0.95
Well-site topography	0.11	0.93	0.34	0.32	3.25	−1.06	0.25	1.62
Season of sample collection	1.00	0.02	1.00	−0.60	0.99	−0.09	1.00	0.15
Presence of irrigation wall	1.02	0.10	0.77	0.09	0.93	−0.86	1.12	1.35
Spill or disposal site	1.04	0.17	1.49	0.59	0.88	−0.47	0.90	−0.54
On-site pesticide storage	0.87	−0.54	0.99	−0.34	1.02	0.10	1.16	0.81
Presence of animals	2.30	2.25	2.10	0.48	1.23	0.04	0.90	−0.18
Aquifer class	2.83	1.40	1.83	0.34	1.48	1.01	2.43	2.12
Output parameter	3.79	1.52	1.93	1.34	1.36	0.58	2.77	2.13

[a]Populations of each parameter in validation and testing subsets are compared with those of training subset.

samples results in inclusion of samples from densely clustered regions and omission of samples from sparsely represented regions, thereby undermining the ANN prediction efficiency. Thus, ANN is unable to find a generalized solution to the problem being investigated because the training and validation subsets that help train the network for adequate generalization ability are not totally representative of the entire population. To avoid this problem, the SOM clustering technique is employed to divide all the samples into clusters and then select an equal number of samples from each cluster for the three subsets. Selection of a suitable number for neurons in Kohonen layer is important because a small SOM groups samples of wider characteristics into a cluster while a large SOM is unable to generate clusters each with enough samples to contribute equally to the three subsets. It is important to note that overfitting occurs when the training and validation subsets are not representative of the entire population. Given a suitable size of data, the methodology presented herein helps in identifying samples for the training and validation subsets with information extending to the edges of the modeling domain in all dimensions.

[63] A small SOM (e.g., 8 × 8 grid; see Table 2b) generates a small number of clusters, with each comprising many samples of wider characteristics because σ_s in equation (4) is a fixed value. On the other hand, a large SOM (e.g., 12 × 12 grid; see Table 2c) generates a large number of clusters, many of which contain one or two samples of precisely similar characteristics. Each SOM cluster should contain enough samples for the three subsets, particularly for the training and validation subsets. Thus, there is a tradeoff between these two. In this study the effects of three different sizes of SOM on the R value are examined. The ANN's predictive performance efficiency in terms of R increases as the number of samples in the training subset increases (case of 12 × 12 grid SOM). However, when two samples from each cluster are included in each subset, the R value is found to be the highest for the 8 × 8 grid SOM case. The reason for the 12 × 12 grid SOM case not getting the highest R value is that most of the clusters contain only one or two samples, resulting in greater discrepancy between the training and validation subsets. Thus, it is necessary to select a suitably sized SOM which produces the maximum number of clusters with more than three or four samples each. An extremely small SOM is not recommended because each cluster will include samples having largely dissimilar characteristics. The R value for cases using the conventional data division method is found to be far below that of the case selecting two samples per cluster of 8 × 8 grid SOM.

[64] The SOM data division technique needs approximately 20% of the samples for the training subset (in the case of selecting two samples per cluster of 8 × 8 grid SOM) unlike the conventional arbitrary data division technique that needs more than 50% of the total samples. Because frequent resampling of drinking water wells on a regional scale is expensive and time consuming, substantial cost and time could be saved if resampling is done only in target areas selected using the SOM clustering technique.

[65] Of the two ANN models considered in this study, BPNN outperforms RBFN in terms of R. However, the μGA-BPNN model takes nearly 24 h for the evolution of optimum R in 30 generations, whereas, the μGA-RBFN model only takes approximately 1.5 h for the evolution of optimum R in 100 generations using a Pentium 4, 1.8 GHz, and 512 MB RAM computer. The findings reported in this paper are based on the study carried out in one region, so for other regions, the concepts need to be examined using available data sets. Nevertheless, the methodology presented in this study can serve as the basis for future advanced study.

Appendix A

A1. Levenberg-Marquardt Algorithm

[66] The Levenberg-Marquardt algorithm, an approximation to Newton's method [*Marquardt*, 1963], is

$$\Delta w = -\left[\nabla^2 F(w)\right]^{-1} \nabla F(w) \tag{A1}$$

where $\nabla^2 F(w)$ is the Hessian matrix and $\nabla F(w)$ is the gradient. $\nabla^2 F(w)$ and $\nabla F(w)$ can be shown as

$$\nabla F(w) = J^T(w) e(w) \tag{A2}$$

$$\nabla^2 F(w) = J^T(w) J(w) + S(w) \tag{A3}$$

where $J(w)$ is a Jacobian matrix and

$$S(w) = \sum_{i=1}^{N} e_i \nabla^2 e_i(w) \tag{A4}$$

[67] For the Gauss-Newton method it is assumed that $S(w) \approx 0$, and equation (A1) becomes

$$\Delta w = -\left[J^T(w) J(w)\right]^{-1} J^T(w) e(w) \tag{A5}$$

[68] The Levenberg-Marquardt modification to the Gauss-Newton method is

$$\Delta w = -\left[J^T(w) J(w) + \zeta I\right]^{-1} J^T(w) e(w) \tag{A6}$$

where I is the unit matrix and ζ is a scalar value. Equation (A6) can be written as [*Hagan et al.*, 1996; *Haykin*, 1999; *Principe et al.*, 1999]

$$w(t+1) = w(t) - \left[J^T\{w(t)\} J\{w(t)\} + \zeta I\right]^{-1} J^T\{w(t)\} e\{w(t)\} \tag{A7}$$

where $w(t)$ is the weight matrix of current iteration t and $t + 1$ is the next iteration.

[69] When the scalar ζ is zero, equation (A7) is just the Gauss-Newton's method; on the other hand when ζ is large, equation (A7) becomes the gradient descent [*Haykin*, 1999] with step size $1/\zeta$. The Gauss-Newton's method is faster and more accurate than the gradient descent near an error minimum, so the aim is to shift toward Gauss-Newton's method as quickly as possible [*Kisi*, 2004; *Cigizoglu and Kisi*, 2005]. The steepest descent method, on the other hand, has a slow asymptotic convergence rate.

A2. Activation Functions

[70] The hyperbolic tangent sigmoid activation function is

$$\varphi(z) = \frac{2}{1 + e^{-2z}} - 1 \tag{A8}$$

where z is the argument.

[71] The linear activation function is

$$\varphi(z) = z \tag{A9}$$

[72] **Acknowledgments.** The authors acknowledge the two anonymous reviewers and associate editor Markus Pahlow, who provided valuable suggestions and remarks for the improvement of the quality of this paper. The authors would like to thank the Water Resources Research Center (WRRC), University of Hawaii at Manoa (UHM), for assistance in the preparation and editing of this manuscript. This is WRRC contributed paper CP-2008-08.

References

Abu-Lebdeh, G., and R. F. Benekohal (1999), Convergence variability and population sizing in micro-genetic algorithms, *Comput. Aided Civ. Infrastruct. Eng.*, *14*(5), 321–334, doi:10.1111/0885-9507.00151.

Alp, M., and H. K. Cigizoglu (2007), Suspended sediment load simulation by two artificial neural network methods using hydrometeorological data, *Environ. Modell. Software*, *22*(1), 2–13, doi:10.1016/j.envsoft.2005.09.009.

ASCE Task Committee (2000), Artificial neural network in hydrology, *J. Hydrol. Eng.*, *5*(2), 124–144, doi:10.1061/(ASCE)1084-0699(2000)5:2(124).

Barbash, J. E., and E. A. Resek (1999), *Pesticides in Ground Water: Distribution, Trends, and Governing Factors*, 590 pp., Lewis, Boca Raton, Fla.

Barbash, J. E., G. P. Thelin, D. W. Kolpin, and R. J. Gillom (1999), Distribution of major herbicides in ground water of the United States, *U.S. Geol. Surv. Water Resour. Invest. Rep.*, *98-4245*, 64 pp.

Birikundavyi, S., R. Labib, H. T. Trung, and J. Rousselle (2002), Performance of neural networks in daily streamflow forecasting, *J. Hydrol. Eng.*, *7*(5), 392–398, doi:10.1061/(ASCE)1084-0699(2002)7:5(392).

Bowden, G. J., H. R. Maier, and G. C. Dandy (2002), Optimal division of data for neural network models in water resources applications, *Water Resour. Res.*, *38*(2), 1010, doi:10.1029/2001WR000266.

Brodnjak-Vončina, D., D. Dobčnik, M. Novič, and J. Zupan (2002), Chemometrics characterisation of the quality of river water, *Anal. Chim. Acta*, *462*(1), 87–100, doi:10.1016/S0003-2670(02)00298-2.

Carroll, D. L. (1996), Genetic algorithms and optimizing chemical oxygen-iodine lasers, *Dev. Theor. Appl. Mech.*, *18*, 411–424.

Carsel, R. F., C. N. Smith, L. A. Mulkey, J. D. Dean, and P. Jowsie (1984), Pesticide root zone model (PRZM), release 1, *EPA-600/3-84-109*, U.S. Environ. Prot. Agency, Washington, D. C.

Chen, C., L. P. Khoo, and W. Yan (2006), An investigation into affective design using sorting technique and Kohonen self-organizing map, *Adv. Eng. Software*, *37*, 334–349, doi:10.1016/j.advengsoft.2005.07.001.

Chrysikopoulos, C. V., P. Hsuan, and M. M. Fyrillas (2002), Bootstrap estimation of the mass transfer coefficient of a dissolving nonaqueous phase liquid pool in porous media, *Water Resour. Res.*, *38*(3), 1026, doi:10.1029/2001WR000661.

Cigizoglu, H. K., and O. Kisi (2005), Flow prediction by three back propagation techniques using k-fold partitioning of neural network training data, *Nord. Hydrol.*, *36*(1), 49–64.

Dixon, B. (2005), Applicability of neuro-fuzzy techniques in predicting ground-water vulnerability: A GIS-based sensitivity analysis, *J. Hydrol.*, *309*, 17–38, doi:10.1016/j.jhydrol.2004.11.010.

Efron, B., and R. J. Tibshirani (1998), *An Introduction to the Bootstrap*, Chapman and Hall, Boca Raton, Fla.

El-Bakyr, M. Y. (2003), Feed forward neural networks modeling for K-P interactions, *Chaos Solitons Fractals*, *18*(5), 995–1000, doi:10.1016/S0960-0779(03)00068-7.

Flood, I., and N. Kartam (1994), Neural networks in civil engineering. I: Principles and understanding, *J. Comput. Civ. Eng.*, *8*(2), 131–148, doi:10.1061/(ASCE)0887-3801(1994)8:2(131).

Gleick, P. H. (1996), Water resources, in *Encyclopedia of Climate and Weather*, vol. 2, edited by S. H. Schneider, pp. 817–823, Oxford Univ. Press, New York.

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

Hagan, M. T., and M. B. Menhaj (1994), Training feed forward techniques with the Marquardt algorithm, *IEEE Trans. Neural Networks*, *5*(6), 989–993, doi:10.1109/72.329697.

Hagan, M. T., H. P. Demuth, and M. Beale (1996), *Neural Network Design*, PWS, Boston, Mass.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, Macmillan, New York.

Ines, A. V. M., and K. Honda (2005), On quantifying agricultural and water management practices from low spatial resolution RS data using genetic algorithms: A numerical study for mixed-pixel environment, *Adv. Water Resour.*, *28*, 856–870, doi:10.1016/j.advwatres.2004.11.015.

Jain, A., and S. Srinivasulu (2004), Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques, *Water Resour. Res.*, *40*, W04302, doi:10.1029/2003WR002355.

Kingston, G. B., M. F. Lambert, and H. R. Maier (2005), Bayesian training of artificial neural networks used for water resources modeling, *Water Resour. Res.*, *41*, W12409, doi:10.1029/2005WR004152.

Kisi, Ö. (2004), Multi-layer perceptrons with Levenberg-Marquardt training algorithm for suspended sediment concentration prediction and estimation, *Hydrol. Sci. J.*, *49*(6), 1025–1040, doi:10.1623/hysj.49.6.1025.55720.

Knisel, W. G. (1993), GLEAMS: Groundwater loading effects of agricultural management systems, version 2.10, Univ. of Ga. Coastal Plain Exp. Stn., Tifton, Ga.

Kohonen, T. (1982), Self-organized formation of topologically correct feature maps, *Biol. Cybern.*, *43*, 59–69, doi:10.1007/BF00337288.

Kolpin, D. W., M. R. Burkart, and E. M. Thurman (1994), Herbicides and nitrate in near-surface aquifers in the mid-continental United States, 1991, *U.S. Geol. Surv. Water Supply Pap.*, *2413*.

Kolpin, D. W., E. M. Thurman, and D. A. Goolsby (1995), Occurrence of selected pesticides and their metabolites in near surface aquifers of the Midwestern United States, *Environ. Sci. Technol.*, *30*, 335–350, doi:10.1021/es950462q.

Krishnakumar, K. (1989), Micro-genetic algorithms for stationary and non-stationary function optimization, in *Intelligent Control and Adaptive Systems*, edited by G. Rodriguez, *Proc. SPIE Int. Soc. Opt. Eng.*, *1196*, 289–296.

Lau, K. W., H. Yin, and S. Hubbard (2006), Kernel self-organizing maps for classification, *Neurocomputing*, *69*, 2033–2040, doi:10.1016/j.neucom.2005.10.003.

Lin, G., and L. Chen (2006), Identification of homogenous regions for regional frequency analysis using the self-organizing map, *J. Hydrol.*, *324*, 1–9, doi:10.1016/j.jhydrol.2005.09.009.

Lin, G., and C. Wang (2006), Performing cluster analysis and discrimination analysis of hydrological factors in one step, *Adv. Water Resour.*, *29*, 1573–1585, doi:10.1016/j.advwatres.2005.11.008.

Lohninger, H. (1994), Estimation of soil partition coefficients of pesticides from their chemical structure, *Chemosphere*, *29*(8), 1611–1626, doi:10.1016/0045-6535(94)90309-3.

Maier, H. R., and G. C. Dandy (1998), The effect of internal parameters and geometry on the performance of back-propagation neural networks: An empirical study, *Environ. Modell. Software*, *13*(2), 193–209, doi:10.1016/S1364-8152(98)00020-6.

Maier, H. R., and G. C. Dandy (2000), Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications, *Environ. Modell. Software*, *15*(1), 101–124, doi:10.1016/S1364-8152(99)00007-9.

Manrique, D., J. Ríos, and A. Rodríguez-Patón (2006), Evolutionary system for automatically constructing and adapting radial basis function networks, *Neurocomputing*, *69*, 2268–2283, doi:10.1016/j.neucom.2005.06.018.

Marquardt, D. (1963), An algorithm for least squares estimation of nonlinear parameters, *SIAM J. Appl. Math.*, *11*(2), 431–441, doi:10.1137/0111030.

Masters, T. (1993), *Practical Neural Network Recipes in C*, Academic, San Diego, Calif.

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

Mishra, A., C. Ray, and D. W. Kolpin (2004), Use of qualitative and quantitative information in neural networks for assessing agricultural chemical contamination of domestic wells, *J. Hydrol. Eng.*, *9*(6), 502–511, doi:10.1061/(ASCE)1084-0699(2004)9:6(502).

Principe, J. C., N. R. Euliano, and W. C. Lefebvre (1999), *Neural and Adaptive Systems: Fundamentals Through Simulations*, John Wiley, New York.

Ray, C., and K. K. Klindworth (2000), Neural networks for agrichemical vulnerability assessment of rural private wells, *J. Hydrol. Eng.*, *5*(2), 162–171, doi:10.1061/(ASCE)1084-0699(2000)5:2(162).

Sahoo, G. B., and C. Ray (2006a), Flow forecasting for a Hawaii stream using rating curves and neural networks, *J. Hydrol.*, *317*, 63–80, doi:10.1016/j.jhydrol.2005.05.008.

Sahoo, G. B., and C. Ray (2006b), Predicting flux decline in cross-flow membranes using artificial neural networks and genetic algorithms, *J. Membrane Sci.*, *283*, 147–157, doi:10.1016/j.memsci.2006.06.019.

Sahoo, G. B., C. Ray, and H. F. Wade (2005), Pesticide prediction in ground water in North Carolina domestic wells using artificial neural network, *J. Ecol. Modell.*, *183*, 29–46, doi:10.1016/j.ecolmodel.2004.07.021.

Sahoo, G. B., C. Ray, E. Mehnert, and D. A. Keefer (2006), Application of artificial neural networks to assess pesticide contamination in shallow groundwater, *Sci. Total Environ.*, *367*, 234–251, doi:10.1016/j.scitotenv.2005.12.011.

Schaap, M. G., and F. J. Leij (1998), Database related accuracy and uncertainty of pedotransfer functions, *Soil Sci.*, *163*(10), 765–779, doi:10.1097/00010694-199810000-00001.

Shi, D., D. S. Yeung, and J. Gao (2005), Sensitivity analysis applied to the construction of radial basis function networks, *Neural Networks*, *18*(7), 951–957, doi:10.1016/j.neunet.2005.02.006.

Simunek, J., M. Sejna, and R. van Genuchten (1998), HYDRUS-1D manual of versions 2.0, U.S. Salinity Lab., Riverside, Calif.

Solley, W. B., R. R. Pierce, and H. A. Perlman (1998), Estimated use of water in the United States in 1995, *U.S. Geol. Surv. Circ. 1200*, 71 pp.

The MathWorks, Inc. (2005), MATLAB version 7.1, Natick, Mass.

Tokar, A. S., and P. A. Johnson (1999), Rainfall-runoff modeling using artificial neural networks, *J. Hydrol. Eng.*, *4*(3), 232–239, doi:10.1061/(ASCE)1084-0699(1999)4:3(232).

U.S. Department of Agriculture Agricultural Research Service (1992), Root zone water quality model (RZQM) v. 1.0, technical documentation, *GSPR Rep. 2*, Great Plains Syst. Res. Unit, Fort Collins, Colo.

U.S. Environmental Protection Agency (1990), National survey of pesticides in drinking water wells, phase I report, *EPA 570/9-90-015*, Off. of Water, Washington, D. C.

Wang, M. X., G. D. Liu, W. L. Wu, Y. H. Bao, and W. N. Liu (2006), Prediction of agriculture derived groundwater nitrate distribution in North China Plain with GIS-based BPNN, *Environ. Geol.*, *50*(5), 637–644, doi:10.1007/s00254-006-0237-x.

Wardlaw, R., and M. Sharif (1999), Evaluation of genetic algorithms for optimal reservoir system operation, *J. Water Resour. Plann. Manage.*, *125*(1), 25–33, doi:10.1061/(ASCE)0733-9496(1999)125:1(25).

Weber, J. B., R. A. McLaughlin, H. F. Wade, and E. Morey (1997), Finding and predicting pesticides in groundwater in North Carolina, in *Consumer Environmental Issues: Safety, Health, Chemicals, and Textiles in the Near Environment*, edited by B. M. Gatewood, A. M. Lewis, and A. C. Robinson, pp. 211–220, Kans. State Univ., Manhattan, Kans.

Yang, C. C., S. O. Prasher, R. Lacroix, and S. H. Kim (2003), A multivariate adaptive regression splines model for simulation of pesticide transport in soils, *Biosyst. Eng.*, *86*(1), 9–15, doi:10.1016/S1537-5110(03)00099-0.

C. Ray, Department of Civil and Environmental Engineering, University of Hawaii at Manoa, 2540 Dole Street, Honolulu, HI 96822, USA.

G. B. Sahoo, Department of Civil and Environmental Engineering, University of California, One Shields Avenue, Davis, CA 95616, USA. (gbsahoo@ucdavis.edu)