

# Spiking modular neural networks: A neural network modeling approach for hydrological processes

Kamban Parasuraman,<sup>1</sup> Amin Elshorbagy,<sup>1</sup> and Sean K. Carey<sup>2</sup>

Received 5 June 2005; revised 7 December 2005; accepted 19 January 2006; published 9 May 2006.

[1] Artificial Neural Networks (ANNs) have been widely used for modeling hydrological processes that are embedded with high nonlinearity in both spatial and temporal scales. The input-output functional relationship does not remain the same over the entire modeling domain, varying at different spatial and temporal scales. In this study, a novel neural network model called the spiking modular neural networks (SMNNs) is proposed. An SMNN consists of an input layer, a spiking layer, and an associator neural network layer. The modular nature of the SMNN helps in finding domain-dependent relationships. The performance of the model is evaluated using two distinct case studies. The first case study is that of streamflow modeling, and the second case study involves modeling of eddy covariance-measured evapotranspiration. Two variants of SMNNs were analyzed in this study. The first variant employs a competitive layer as the spiking layer, and the second variant employs a self-organizing map as the spiking layer. The performance of SMNNs is compared to that of a regular feed forward neural network (FFNN) model. Results from the study demonstrate that SMNNs performed better than FFNNs for both the case studies. Results from partitioning analysis reveal that, compared to FFNNs, SMNNs are effective in capturing the dynamics of high flows. In modeling evapotranspiration, it is found that net radiation and ground temperature alone can be used to model the evaporation flux effectively. The SMNNs are shown to be effective in discretizing the complex mapping space into simpler domains that can be learned with relative ease.

**Citation:** Parasuraman, K., A. Elshorbagy, and S. K. Carey (2006), Spiking modular neural networks: A neural network modeling approach for hydrological processes, *Water Resour. Res.*, 42, W05412, doi:10.1029/2005WR004317.

## 1. Introduction

### 1.1. Advances in Neural Network Modeling in Hydrology

[2] Modeling of hydrological processes is central for efficient planning and management of water resources, which is usually achieved either by conceptual models or by systems theoretic models. Artificial neural networks (ANNs), a systems theoretic method, have been shown to be a promising tool for modeling hydrological processes [*ASCE Task Committee on the Application of Neural Networks in Hydrology*, 2000a; *Maier and Dandy*, 2000]. The increasing utility of ANNs in modeling hydrological processes is attributed to their ability to capture complex nonlinear relationships between inputs and outputs with an incomplete understanding of the physics of the process involved.

[3] The presence of discontinuity in rainfall runoff mapping of a watershed and significant variations in input space motivated *Zhang and Govindaraju* [2000] to develop modular neural networks. They attributed the discontinuity to

rainfall runoff functional relationships being different for low, medium, and high magnitudes of streamflow. The modular neural network developed by *Zhang and Govindaraju* [2000] consists of a gating network and a series of neural networks. Each neural network in this series is termed as an expert, mapping the relationship in a subset of input space. The gating network helps in identifying the expert for a given input vector. The gating network outputs the probability of an input vector association with each of the experts. The output from the network is then calculated by multiplying the individual expert's response by the corresponding weights (probability) of the gating network. *Zhang and Govindaraju* [2000] showed that the performance of modular neural networks is better than that of the regular feed forward neural networks (FFNNs). *Hsu et al.* [2002] developed a self-organizing linear output map (SOLO); an artificial neural network model studying the rainfall runoff modeling problem. SOLO consists of a classification layer and a mapping layer. Classification of input space is achieved by means of a self-organizing feature map (SOFM) [*Kohonen*, 1989]. The mapping layer helps map the input to its corresponding output by means of piecewise linear regression functions. On the basis of their study, *Hsu et al.* [2002] concluded that the SOLO model resulted in rapid and precise estimation of system outputs. *Hong et al.* [2005] extended SOLO to a self-organizing nonlinear output (SONO) model for cloud patch-based rainfall estimation. Similar to SOLO, SONO made use of SOFM in its classifi-

<sup>1</sup>Centre for Advanced Numerical Simulation (CANSIM), Department of Civil and Geological Engineering, University of Saskatchewan, Saskatoon, Saskatchewan, Canada.

<sup>2</sup>Department of Geography and Environmental Studies, Carleton University, Ottawa, Ontario, Canada.

cation layer. However, in SONO, mapping input space to output is achieved by means of nonlinear regression. Recently, *Bowden et al.* [2005] used a self-organizing map (SOM) to reduce the dimensionality of the input space and obtain independent outputs, with the objective of finding the optimal combination of input parameters for neural networks modeling. The input variables are presented to the SOM, and only one input is selected from each cell on the basis of its proximity to the cluster centers. These selected inputs are then used to train the neural networks models. On the basis of the above studies it can be concluded that input-output functional relationship is quite different in different domains of the input space. Hence cluster-based mapping appears to be a promising alternative to FFNNs, particularly in cases of processes where the input-output functional relationship is fragmented or discontinuous. Although *Zhang and Govindaraju* [2000] demonstrated the importance of modular neural networks, and *Hsu et al.* [2002] demonstrated the utility of self-organizing maps in modeling hydrological processes, little effort has been made to study the usefulness of harnessing both modular learning and self-organizing networks. Moreover, to the knowledge of the authors, no work has been reported in literature to compare the different ways by which self-organization in networks can be achieved. This comparison is of particular interest as it helps in identifying the proper self-organization technique suitable for modular learning.

## 1.2. Neurohydrology: Beyond Rainfall Runoff Modeling

[4] Compared to other hydrological processes such as rainfall and runoff, evaporation (used here to describe latent heat flux from the surface) is more dynamic because it involves continuous exchange of water molecules between the land and the atmosphere. Hence the evaporation process is embedded with huge variability in both spatial and temporal scales. For this reason, evaporation is the least satisfactorily explained component of the global hydrological cycle [*Sudheer et al.*, 2002]. An improvement in the estimates of evaporation helps in partitioning the available moisture into (1) water loss back to the atmosphere and (2) the water available for runoff. Although water balance components including rainfall, infiltration, and runoff are measured directly, evaporation is most commonly estimated by energy balance, mass transfer, or water budget methods [*Sudheer et al.*, 2002]. Traditionally in engineering and application, pan evaporation is used as an index for free water surface (lakes and reservoirs) evaporation, and empirical coefficients are applied to correlate pan evaporation to reference crop evapotranspiration ( $ET_o$ ). Alternatively, lysimeters are used to directly estimate surface or crop evapotranspiration ( $ET_c$ ) by measuring changes in mass of a control volume [*Singh*, 1989]. Measurements of evaporation by pan evaporimeter and lysimeter are subject to a large set of assumptions, cumbersome and labor-intensive, and may not be appropriate for large-scale studies. In research applications, micrometeorological methods such as energy balance Bowen ratio (EBBR) and eddy covariance (EC) are typically used to measure actual evaporation (ET) [*Drexler et al.*, 2004]. However, these methods are expensive and are sufficiently complex to limit their widespread application. In order to overcome these problems, numerous studies have been carried out to estimate  $ET_o$  from climatic data.

Key examples of such studies include (1) empirical relationships between meteorological variables [*Holdridge*, 1962; *Stephens and Stewart*, 1963; *Blaney and Criddle*, 1950; *Linacre*, 1977; *Thornthwaite*, 1948; *Priestley and Taylor*, 1972; *Hargreaves and Samani*, 1982] and (2) physically based equations [*Penman*, 1948; *Monteith*, 1965]. While the former methods estimate  $ET_o$  on the basis of climatological data, the latter methods link evaporation dynamics with the supply of net radiation (NR) and aerodynamics transport characteristics of a natural surface, and hence are termed combination methods.

[5] The success of the neural network models in modeling different hydrological processes provides an impetus to test the applicability of neural networks in modeling the highly dynamic evaporation process. Key examples of such studies include *Sudheer et al.* [2002]; *Kumar et al.* [2002]; *Sudheer et al.* [2003]; *Trajkovic et al.* [2003]. Most of the above studies on modeling evaporation using neural networks estimated either the PM estimates of evaporation [*Kumar et al.*, 2002; *Trajkovic et al.*, 2003], the pan evaporation [*Sudheer et al.*, 2002], or the lysimeter-measured [*Sudheer et al.*, 2003] values. To the knowledge of the authors no work has been reported in the literature to address the application of neural networks in modeling EC measured evaporation flux.

[6] In this study, a novel neural network model is proposed: the spiking modular neural network (SMNN). The SMNN is based on the concepts of both self-organizing networks and modular networks. The performance of the model is tested on two diverse case studies. The first case study involves modeling of streamflows and the second case study involves modeling of actual evaporation measured via eddy covariance (EC). While the first case study represents a single-input single-output (SISO) process, the second case study represents a multiple-input single-output (MISO) process. Two variants of SMNNs, one employing competitive learning in the spiking layer and the other employing self-organizing maps in the spiking layer, are tested. The specific objectives of this research are as follows: (1) to evaluate the performance of regular FFNN in modeling streamflows and EC-measured evaporation flux; (2) to compare the performance of FFNNs with the proposed SMNNs on both case studies; and (3) to provide insight into the performance of the SMNNs.

[7] The remaining part of this paper is organized as follows. In section 2, an introduction to neural networks is given. Section 3 presents the architecture of the spiking modular neural networks adopted in this study. Streamflow estimation and modeling of evaporation flux are discussed in sections 4 and 5, respectively. Results and discussion are presented in section 6, and section 7 summarizes important research conclusions.

## 2. Neural Networks

[8] ANNs are a method of computation and information processing motivated by the functional units of the human brain, namely neurons. According to *Haykin* [1999], a neural network is a massively parallel distributed information processing system that is capable of storing the experiential knowledge gained by the process of learning and of making it available for future use. Mathematically, ANNs are universal approximators with an ability to solve large-

scale complex problems such as time series forecasting, pattern recognition, nonlinear modeling, classification, and control. This is achieved by identifying the relationships among given patterns.

[9] FFNNs are the most widely adopted network architecture for the prediction and forecasting of water resources variables [Maier and Dandy, 2000]. Typically, FFNNs consist of three layers: input layer, hidden layer, and output layer. The number of nodes in the input layer corresponds to the number of inputs considered for modeling the output. The input layer is connected to the hidden layer with weights that determine the strength of the connections. The number of nodes in the hidden layer indicates the complexity of the problem being modeled. The hidden layer nodes consist of the activation function, which helps in nonlinearly transforming the inputs into an alternative space where the training samples are linearly separable [Brown and Harris, 1994]. The hidden layer is connected to the output layer. An epoch is the presentation of the entire training samples to the neural networks model. Detailed review of ANNs and their application in hydrology can be found in Maier and Dandy [2000] and in *ASCE Task Committee on Application of Artificial Neural Networks in Hydrology* [2000a, 2000b].

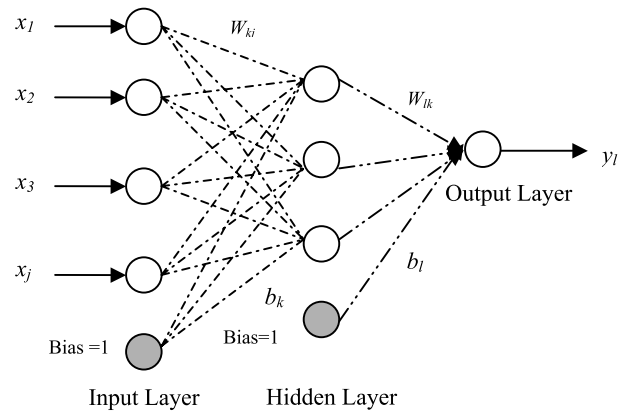
[10] The structure of the three-layered FFNN used in this study is shown in Figure 1. The neural network model consists of ‘ $j$ ’ input neurons, ‘ $k$ ’ hidden neurons, and ‘ $l$ ’ output neurons. Symbolically, the ANN architecture shown in Figure 1 can be represented as ANN( $j, k, l$ ). The FFNN adopted in this study makes use of the tan-sigmoidal activation function in the hidden layer and the linear activation function in the output layer. In Figure 1,  $W_{kj}$  represents the connection weight between the  $j$ th input neuron and  $k$ th hidden neuron. Similarly,  $W_{lk}$  represents the connection weight between the  $k$ th hidden neuron and  $l$ th output neuron. Parameters  $b_k$  and  $b_l$  represent the bias of the corresponding hidden and output layer neurons. If  $x_j$  represents the input variables and  $y_l$  represents the output variable, then the inputs are transformed to output by the following equations:

$$y_l = f_1 \left[ \sum_{k=1}^K w_{lk} f_2 \left( \sum_{j=1}^J w_{kj} x_j + b_k \right) + b_l \right] \quad (1)$$

$$f_2(p) = \frac{2}{(1 + e^{-2p})} - 1 \quad (2)$$

where  $f_1$  represents the linear activation function and  $f_2$  represents the tan-sigmoidal activation function. While the tan-sigmoidal activation function squashes the input between  $-1$  and  $1$ , the linear activation function calculates the neurons output by simply returning the value passed to it. One of the important issues in the development of neural networks model is the determination of optimal number of hidden neurons that can satisfactorily capture the nonlinear relationship existing between the input variables and the output. The number of neurons in the hidden layer is usually determined by trial and error method with the objective of minimizing the cost function [ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a].

[11] The typical cost function used in training FFNNs involves minimizing the mean sum of squares of the



**Figure 1.** Structure of the three-layered feed forward neural network (FFNN).

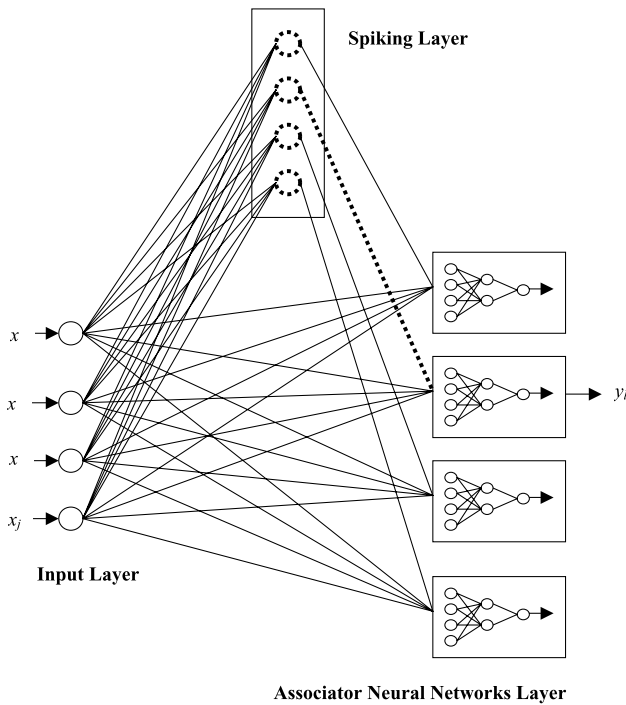
network errors (MSE). However, in this study, in order to overcome the problem of overfitting, the Bayesian regularization back-propagation algorithm [Demuth and Beale, 2001] is used for training the FFNNs. The Bayesian regularization back-propagation algorithm improves the generalization property of the ANN model by developing networks with smaller weights and biases, and thus a smoother response that is less likely to result in overfitting [Demuth and Beale, 2001]. Hence along with minimizing MSE, the cost function in the Bayesian regularization back-propagation algorithm (equation (3)) involves minimizing the mean of the sum of squares of the network weights and biases (MSW). In equation (3),  $y_i$  and  $y_i'$  represent the measured and computed counterparts;  $n$  and  $N$  represent the number of training instances and the number of network parameters. The success of the regularization depends on the choice of an appropriate value of the regularization parameter,  $\alpha$ . In this study, the method by MacKay [1992] is adopted, where the optimal  $\alpha$  is determined in a Bayesian framework using automatic relevance determination.

$$MSE\_REG = \frac{1}{n} \sum_{i=1}^n (y_i - y_i') + (1 - \alpha) \left( \frac{1}{N} \sum_{j=1}^N w_j^2 \right) \quad (3)$$

[12] A systematic search of different network configuration and user-adjustable parameters was carried out to ascertain the optimal network architecture, with the objective of minimizing the cost function. The optimal network architecture is the one which results in the least cost function. Although the FFNNs are capable of approximating a continuous well-behaved relationship between the input and output variables, they may not be suitable for mapping a fragmented or discontinuous representation of the training data that has significant variation over the input space [Zhang and Govindaraju, 2000]. In SMNNs, explained in section 3, the inability of the FFNNs in mapping fragmented data sets is overcome by decomposing the complex mapping space into simpler subdomains that can be learned with relative ease by the individual FFNN models.

### 3. Spiking Modular Neural Networks (SMNNs)

[13] The structure of the SMNNs is shown in Figure 2. The input layer consists of  $j$  input neurons ( $x_1, x_2, x_3, \dots, x_j$ )



**Figure 2.** Structure of the Spiking Modular Neural Network (SMNN).

where the number of input neurons,  $j$ , is equal to the number of input variables. The input layer neurons are connected to the spiking layer, which serves as the “memory” of the system, learning and storing different input patterns that can be used in classifying future input vectors on the basis of patterns learned during the training process. In the spiking layer, clustering of input space is achieved by unsupervised (or self-organized) learning, which is defined as the learning process that does not involve a teacher or critic to oversee the learning. Self-organized learning consists of repeatedly modifying the synaptic weights of a neural network in response to activation patterns and in accordance with prescribed rules, until a final configuration appears [Haykin, 1999]. Furthermore, self-organizing networks can learn to detect regularities and correlations in the input space, and accordingly adapt their future responses to that input.

[14] Self-organization in networks can be achieved in two ways: (1) competitive learning and (2) self-organizing maps (SOMs) [Demuth and Beale, 2001]. In competitive learning the neurons of the network compete among themselves to be active (spike), the winner of which is called a “winner-takes-all” neuron [Haykin, 1999]. The SOMs are a special case of self-organizing system as they learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors. A SOM is therefore characterized by the formation of a topographic map of the input patterns in which the spatial locations of the neurons in the lattice are indicative of intrinsic statistical features contained in the input patterns [Haykin, 1999]. Hence the main difference between competitive learning and SOMs is that, while the former learns only the distribution, the latter learns both the distribution and the topology (neighboring neurons) of the input space. SOMs are either one-dimensional or two-

dimensional and the structure of SOMs is usually represented by the form  $n_1 \times n_2$ , where  $n_1$  and  $n_2$  represent the number of rows and columns of neurons, respectively. The following paragraph outlines the mechanism involved in learning patterns by the self-organizing networks.

[15] The weights of the self-organizing networks are initialized to the center of the input ranges. Once initialized, the self-organizing network neurons are trained by the Kohonen learning rule [Kohonen, 1989] to identify the clusters in the input space and allow the connection weights of the neuron to learn an input vector. Each neuron of the self-organizing network competes to respond to an input vector. Proximity of inputs to each neuron is determined on the basis of Euclidean distance ( $d_c$ ) as given in equation (4):

$$d_{c=1..m} = \left[ \sum_{j=1}^J (x_j - w_{cj})^2 \right]^{0.5} \quad (4)$$

where  $m$  denotes the number of clusters and  $w_{cj}$  represents the connection weight linking  $j$ th input variable and  $c$ th neuron of self-organizing networks. In the case of competitive learning, the neuron whose weight vector is closest to that of the input vector is updated to be even closer. However, in self-organizing maps, along with the closest neuron, the neurons in the neighborhood of the closest neuron are also updated to be even closer. The result of such training results in a neural network model where the winning neuron is more likely to win the competition the next time a similar vector is presented and less likely to win when a very different input vector is presented. Hence for a given input vector, the neuron which represents the cluster that is closest to the input vector outputs 1 (spikes), while the remaining neurons output 0. More information on self-organizing networks can be found in Demuth and Beale [2001] and Kohonen [1989].

[16] Once classification of the input space is achieved, mapping of inputs to the corresponding outputs has to be carried out. Mapping inputs to outputs can be achieved by either linear regression or FFNNs. In the case of highly correlated input variables, use of the linear regression model for mapping inputs to outputs may require conversion of input variables to its principal components in order to avoid colinearity problem [Hsu et al., 2002]. In this study, mapping of inputs to outputs is achieved by neural networks, and as these networks associate input patterns to outputs, they are termed associator neural networks. The associator neural networks are similar to the neural networks detailed in section 2. SMNNs belong to a class of modular neural networks as the SMNNs works by developing ‘ $c$ ’ different associator neural networks, each specializing in ‘ $c$ ’ different subsets of the mapping domain.

[17] In this study, the performance of the proposed SMNNs employing the self-organizing networks (both competitive and SOMs) is tested. The first SMNN makes use of the competitive network as the spiking layer, and herein it will be referred as SMNN(Competitive). The second variant of the SMNNs makes use of SOM as the spiking layer. Herein, the SMNNs with SOM as the spiking layer will be referred to as SMNN(SOM). Since competitive networks learn only the distribution of the input space, and SOMs learn both the distribution and topology of the

**Table 1.** Statistical Properties of Streamflow Data Between Umfreville and Sioux Lookout

Statistics	Entire Data Set		Training Data Set		Testing Data Set	
	Umfreville	Sioux Lookout	Umfreville	Sioux Lookout	Umfreville	Sioux Lookout
Minimum, m <sup>3</sup> /s	3.2	16.2	3.2	18.5	9.7	16.2
Maximum, m <sup>3</sup> /s	372.0	634.0	372.0	634.0	274.0	511.0
Median, m <sup>3</sup> /s	41.3	90.6	39.5	88.4	47.3	93.8
Average, m <sup>3</sup> /s	57.0	121.0	55.0	120.7	62.1	121.7
SD <sup>a</sup> , m <sup>3</sup> /s	45.5	87.3	44.6	87.7	47.4	86.3
CV <sup>b</sup>	0.8	0.7	0.8	0.7	0.8	0.7

<sup>a</sup>SD is standard deviation.

<sup>b</sup>CV is coefficient of variation.

input space, comparison of SMNN(Competitive) and SMNN(SOM) would help understand the effect of topology learning in SMNNs performance. Since there is no theoretical principle to determine the optimum size of the Kohonen layer [Cai *et al.*, 1994], the number of nodes in the spiking layer of SMNN(Competitive) and SMNN(SOM) is determined by the trial-and-error method. Starting with two nodes in each of the spiking layer and hidden layer, the optimal architecture of the SMNNs is evaluated by performing a systematic search over different network configurations with the objective of minimizing the cost function (equation (3)). Symbolically, the optimal architecture of a SMNN with  $c$  spiking neurons can be represented as SMNN( $c$ , ANN( $j$ ,  $k$ ,  $l$ )), where ANN( $j$ ,  $k$ ,  $l$ ) represents the optimal associator neural networks configuration.

#### 4. Streamflow Prediction

[18] The monthly streamflow values of the English River, Ontario, Canada, between Umfreville (49° 52'N, 91° 27'W) and Sioux Lookout (50° 4'N, 91° 56'W) is considered in this study. Umfreville is located upstream from Sioux Lookout. The streamflow values are obtained from Environment Canada's Hydrometric Database (HYDAT) (Government of Canada, 2004, available at <http://www.wsc.ec.gc.ca/>). Flow values at Sioux Lookout are considered missing and are estimated on the basis of the flow values at Umfreville. Out of the available data between January 1924 and December 1981, approximately 70% of the data is used for training, and the remaining 30% of the data is used for testing the developed model. That is, monthly flow values from January 1924 to August 1965 are used for training the neural networks, and the flow values from September 1965 to December 1981 are considered for testing the models. The statistical properties of the entire data set along with the statistical properties of the data sets used for training and testing are presented in Table 1. The flow at Sioux Lookout shows slightly less variability than the flow at Umfreville. Both training and testing data sets have similar statistical properties.

##### 4.1. Predicting Streamflow Using FFNNs

[19] A three-layered FFNN is considered in this study. The input layer consists of a single input neuron representing the flow at Umfreville. The output layer has a single

output neuron corresponding to the flow at Sioux Lookout. Mapping of inputs to outputs is achieved by the hidden layer neurons. The number of hidden layer neurons is determined by the trial-and-error method as detailed in section 2, and the optimal number of hidden neurons is three. Hence the neural network architecture adopted in this study is of the form ANN(1, 3, 1). For this study, 2000 epochs is found optimal for training the networks.

##### 4.2. Predicting Streamflow Using SMNNs

[20] Similar to the FFNN, the SMNN model consists of single input and single output neurons. The performances of both SMNN(Competitive) and SMNN(SOM) in streamflow prediction are evaluated in this study. Similar to the method for determining the number of hidden nodes in FFNNs, the numbers of neurons in the spiking layers of both the SMNN(Competitive) and SMNN(SOM) models are determined by the trial-and-error method as detailed in section 3. The optimal number of neurons for both variants of SMNNs is two, indicating that there are two different clusters in the input space. As a next step, the clustered input space is mapped to the corresponding output space. This is achieved by the associator neural networks. Since there are two different clusters, the SMNNs consist of two associator neural networks. Each of these associator neural networks specializes in mapping the input-output relationship at the respective domain of the mapping space.

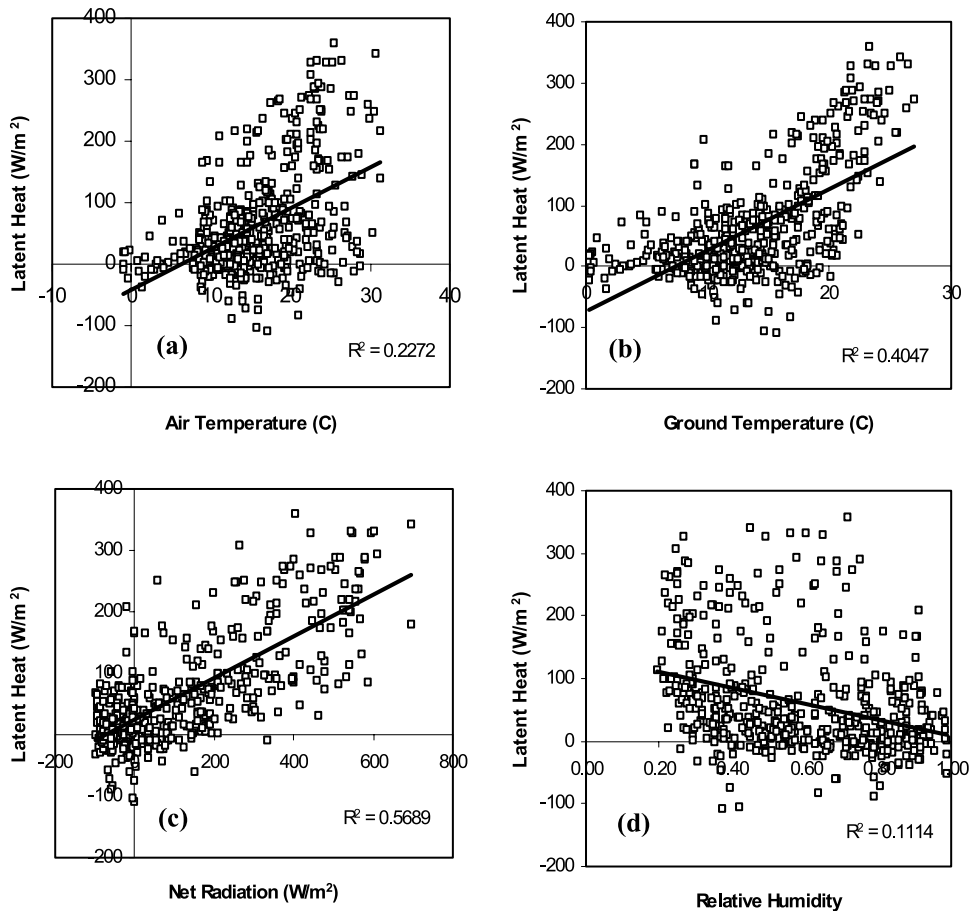
[21] The spiking layer is trained until the neurons in this layer are able to learn the classification of the input space. This is determined by finding the number of epochs beyond which there is no further improvement in classification of input vectors. For this study, the optimal number of epochs required for training the spiking layer is found to be 300. The architecture of all the associator neural networks is similar and is determined in a way analogous to the method used to determine the architecture of the regular FFNNs. For both variants of SMNNs, ANN(1, 3, 1) is the optimal architecture of the associator neural networks. Symbolically, the optimal architecture of SMNNs in modeling streamflow is given by SMNN(2, ANN(1, 3, 1)).

#### 5. Modeling Actual Evaporation Using Climatic Data

##### 5.1. Site Description and Data

[22] South Bison Hill (SBH) (57° 39'N and 111° 13'W), a waste rock overburden pile located north of Fort McMurray, Alberta, Canada, is considered in this study. SBH was constructed with waste rock material from oilsands mining in stages between 1980 and 1996. The area of SBH is 2 km<sup>2</sup>, rises 60 m above the surrounding landscape, and has a large flat top several hundred meters in diameter. To reclaim the overburden so that revegetation can occur, the underlying shale is covered by a 0.2 m layer of peat on top of a 0.8 m layer of till. The top of SBH is dominated by foxtail barley (*Hordeum jubatum*); also present are other minor species such as fireweed (*Epilobium angustifolium*). Estimation of evaporation from the reconstructed watershed is of vital importance as it plays a major role in water balance of the system, which links directly to ecosystem restoration strategies.

[23] Micrometeorological techniques were used to directly measure evaporation and the surface energy balance. A mast



**Figure 3.** Correlation plot between latent heat and (a) air temperature, (b) ground temperature, (c) net radiation, and (d) relative humidity.

located in the approximate center of SBH was equipped to measure air temperature (AT) and relative humidity (RH) (HMPFC, Vaisala, 3 m) housed in a Gill radiation shield, ground temperature (GT) (TVAC, Campbell Scientific, averaged 0.01–0.05 m depth), all-wave net radiation ( $R_n$ ) (CNR-1, Kipp and Zonen, 3 m), and wind speed (WS) (015A Met One, 3.18 m). All instruments were connected to a data logger (CR23X, Campbell Scientific) and sampled at 10 s, and an average or a cumulate record was logged every half hour. The energy balance of the surface is given by:

$$R_n = LE + H + G + \varepsilon \quad (5)$$

where  $LE$  is the latent heat flux (evaporation when divided by the latent heat of vaporization),  $H$  the sensible heat flux,  $G$  the ground heat flux, and  $\varepsilon$  the residual flux density, all expressed in  $W\ m^{-2}$ .  $G$  was measured using a CM3 radiation and energy balance (REBS) ground heat flux plate placed at 0.05 m depth.  $LE$  and  $H$  were measured directly via the open path eddy covariance (EC) technique [Leuning and Judd, 1996] using a CSAT3 sonic anemometer (Campbell Scientific) and an LI-7500  $CO_2/H_2O$  gas analyzer (Li-Cor) with the midpoint of the sonic head located on a boom 2.8 m above the ground surface. Measurements of  $H$  and  $LE$  were taken at 10 Hz, and fluxes were calculated using 30-min block averages with 2-D

coordinate rotation. Sensible heat fluxes were calculated using the sonic virtual temperature [Schotanus *et al.*, 1983], and latent heat fluxes were corrected for changes in air density [Webb *et al.*, 1980]. Fluxes were removed when friction velocity was less than 0.1 m/s due to poor energy balance closure at low wind speeds [Twine *et al.*, 2000; Baker and Griffis, 2005]. Flux measurements were also removed during periods of rainfall and during periods of unexpected change in state variables. No gap filling was performed.

[24] Variation of evaporation is commonly perceived as highly dependent on climatic variables such as temperature, humidity, solar radiation, and wind speed [Brutsaert, 1982; Sudheer *et al.*, 2003]. Hence in this study, the climatic variables AT, GT,  $R_n$ , RH, and WS, which are commonly measured at weather stations, are used to estimate the evaporation flux measured by the EC system. As a common practice, a training set is used for model development, and an independent validation set is used to test the efficiency of the developed model. Hourly data between 20 May 2003 and 9 June 2003 comprise the training set, and the data between 18 June 2003 and 28 June 2003 comprise the testing set. The training set consists of 500 instances while the testing set consists of 247 instances. Plots showing the correlation of input variables AT, GT,  $R_n$ , and RH with  $LE$  are presented in Figure 3. The correlation plot between WS and  $LE$  is not shown as there is no significant correlation

between them. The correlation plots shown in Figure 3 are based on the training set alone. As expected, air temperature ( $R^2 = 0.227$ ), ground temperature ( $R^2 = 0.405$ ), and net radiation ( $R^2 = 0.569$ ) are shown to have a positive trend with LE, while relative humidity ( $R^2 = 0.114$ ) has a negative relationship with LE.

[25] Traditionally, Penman-Monteith is the most widely used method for estimating evapotranspiration due to the widespread availability of the input variables. The hourly FAO Penman-Monteith [Temesgen *et al.*, 2005] equation is given by equation (6):

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{37}{AT+273} WS(e^0 - e^a)}{\Delta + \gamma(1 + 0.34WS)} \quad (6)$$

where,  $R_n$  is net radiation at the grass surface ( $\text{MJ m}^{-2} \text{hr}^{-1}$ ),  $G$  is soil heat flux density ( $\text{MJ m}^{-2} \text{hr}^{-1}$ ),  $\Delta$  is the saturation slope vapor pressure curve at AT ( $\text{K Pa } ^\circ\text{C}^{-1}$ ),  $\gamma$  is the psychrometric constant ( $\text{K Pa } ^\circ\text{C}^{-1}$ ),  $e^0$  is saturation vapor pressure at air temperature AT ( $\text{K Pa}$ ),  $e^a$  is the average hourly actual vapor pressure ( $\text{K Pa}$ ), and  $WS$  is the average hourly wind speed ( $\text{m/s}$ ). It should be noted that the evaporation calculated by the Penman-Monteith equation is potential evaporation for a well-watered surface and not actual evaporation. Several methods of converting  $ET_0$  to actual evaporation which estimate actual evaporation based on water balance or by empirical equations have been illustrated by Saxton [1981] and Jensen [1981]. Eddy covariance (EC) offers a convenient way to directly measure actual evaporation, and hence in this study, an attempt has been made to model EC-measured evaporation flux using neural networks.

## 5.2. Estimation of Evaporation Flux Using FFNNs

[26] The FFNN model considered for modeling evaporation flux consists of five input neurons, representing AT, GT,  $R_n$ , RH, and WS. The output layer consists of a single neuron representing LE. As explained in section 2, the optimal number of hidden nodes is found by the trial-and-error method and is found to be four. Hence the neural network architecture adopted in this study is of the form ANN(5,4,1). The Bayesian regularization algorithm is used for training the networks. For this case study, 5000 epochs is found optimal for training the FFNNs.

## 5.3. Estimation of Evaporation Flux Using SMNNs

[27] The performances of both variants of SMNNs (SMNN(Competitive) and SMNN(SOM)) are tested with regard to estimating the EC-measured evaporation flux. The SMNNs considered in this application consist of five input neurons. By the trial-and-error method, as detailed in section 3, the optimal number of neurons in the spiking layer is found to be eight for both SMNN(Competitive) and SMNN(SOM). The spiking layer consists of eight neurons, representing individual clusters in the input space. Eight hundred epochs are found optimal for training the spiking layer. Corresponding to each cluster, eight different associator neural network models specializing in mapping input-output relationships at different domains of the mapping space are constructed. The associator neural network models employ Bayesian regularization algorithm for training the networks. The optimal network architecture of associa-

tor neural networks is ANN(5, 4, 1). Symbolically, the optimal architecture of SMNNs can be represented as SMNN(8, ANN(5, 4, 1)).

## 6. Results and Discussion

[28] Since root-mean-square error (RMSE) and mean relative error (MRE) give different details about the predictive ability of the models [Karunanithi *et al.*, 1994], a multicriterion performance evaluation is carried out. For both case studies the performances of the different models are evaluated on the basis of (1) RMSE, (2) MRE, and (3) the coefficient of correlation ( $R$ ). RMSE, MRE, and  $R$  are calculated using equations (7), (8), and (9), respectively, where  $n$  represents the number of instances presented to the model;  $y_i$  and  $y_i'$  represent measured and computed evaporation flux, respectively; and  $\bar{y}$  represents the mean of the corresponding variable:

$$RMSE = \left[ \frac{1}{n} \sum_{i=1}^n (y_i - y_i')^2 \right]^{0.5} \quad (7)$$

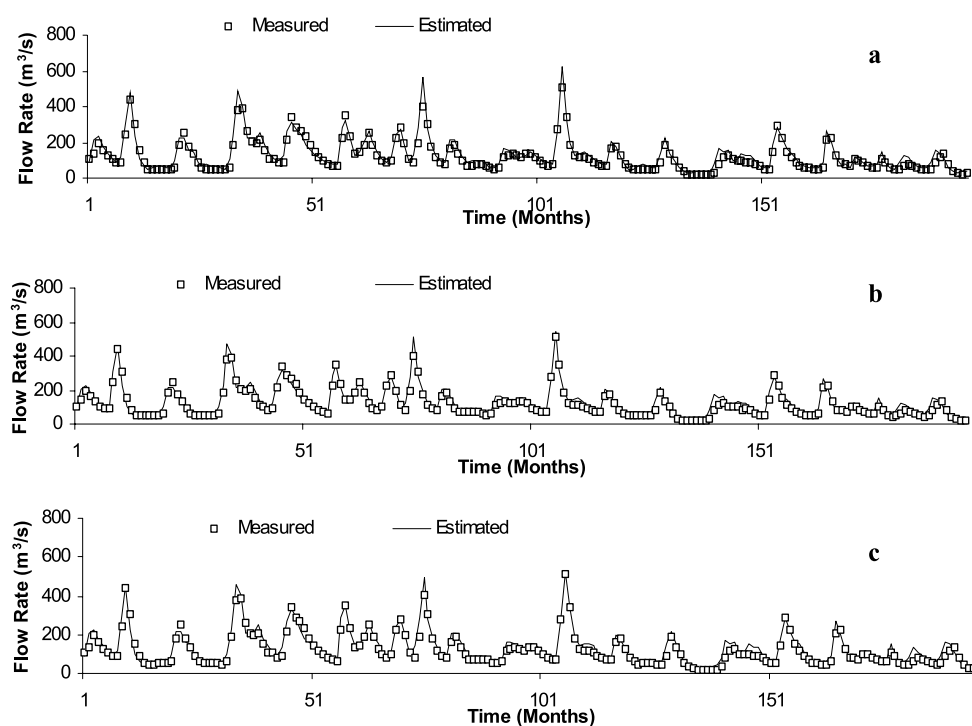
$$MRE = \frac{1}{n} \sum_{i=1}^n \left( \frac{y_i - y_i'}{y_i} \right) \quad (8)$$

$$R = \frac{\sum_{i=1}^n (y_i - \bar{y})(y_i' - \bar{y}')}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (y_i' - \bar{y}')^2}} \quad (9)$$

### 6.1. Streamflow Modeling

[29] Table 2 presents the statistical performances of different neural network models in estimating streamflow at Sioux Lookout on the basis of the flow at Umfreville. FFNNs resulted in an RMSE of  $30.4 \text{ m}^3/\text{s}$ , an MRE of 0.22, and a correlation coefficient of 0.96. Both SMNN(Competitive) and SMNN(SOM) models performed better than the FFNN model in terms of RMSE during testing, however all the models performed on par in terms of MRE and  $R$ . Figure 4 indicates that different neural network models were able to mimic the trend of measured flows at Sioux Lookout. However, as will be shown later, SMNNs resulted in better prediction accuracy in the case of high flows. Comparing SMNN(Competitive) and SMNN(SOM), the former method performed marginally better (lower RMSE) than the latter.

[30] To improve the insight into the performance of SMNNs, a plot showing the instances at which different spiking layer neurons spiked is shown in Figure 5. Both SMNN(Competitive) and SMNN(SOM) were successful in delineating high flows from low flows. The threshold value separating high flows and low flows differed between SMNN(Competitive) and SMNN(SOM). The solid lines in Figure 5 indicate the threshold values. For SMNN(Competitive) the threshold value was  $185.6 \text{ m}^3/\text{sec}$ , and for SMNN(SOM) the threshold value was approximately  $145.7 \text{ m}^3/\text{sec}$ . In SMNN(Competitive), 48 instances were above the threshold value, and the remaining 148 instances were below the threshold value. The MRE of the instances above the threshold is 0.14, and the MRE of the instances below the



**Figure 4.** Comparison of measured and estimated flows by (a) FFNNs, (b) SMNN(Competitive), and (c) SMNN(SOM).

threshold is 0.25. However, in the case of SMNN(SOM), 78 instances were above the threshold value, and the remaining 118 instances were below the threshold value. The MRE of the instances above the threshold is 0.19, and the MRE of the instances below threshold is 0.25.

[31] *Sajikumar and Thandaveswara* [1999] and *Tokar and Markus* [2000] experienced problems in learning patterns using a back-propagation algorithm when target flow values were in the neighborhood of zero. However, *Minns and Hall* [1996] reported that the regular FFNNs were not able to mimic larger peaks in flow data. This problem with regard to the performance of SMNNs is analyzed by partitioning analysis. Partitioning is carried out by arbitrarily choosing a certain threshold of flow, then finding the errors (RMSE and MRE) between the measured and estimated flows, both above and below the threshold value. The testing data set, which consists of 196 instances, is considered for this analysis. The mean ( $\mu$ ) and the standard deviation ( $\sigma$ ) of the data are  $121.69 \text{ m}^3/\text{s}$  and  $86.31 \text{ m}^3/\text{s}$ , respectively. A value slightly less than  $\mu + \sigma$  is considered as the threshold ( $200 \text{ m}^3/\text{s}$ ). Out of the total 196 instances, 167 instances are below the threshold value, and the remaining 29 instances are above the threshold value. The relative performance of these models above and below the threshold is presented in Table 3. From Table 3 it can be seen that the performance of SMNNs is on par with FFNNs performance below the threshold value ( $<200 \text{ m}^3/\text{s}$ ). However, SMNNs performed better than FFNNs for flows above the threshold value ( $>200 \text{ m}^3/\text{s}$ ). On the basis of the above analysis it is concluded that the proposed SMNNs are a promising alternative for modeling high flows.

## 6.2. Modeling Evaporation

[32] *Jackson et al.* [1976] and *Salvucci* [1997] showed that evaporation is a two-stage process, with a climate

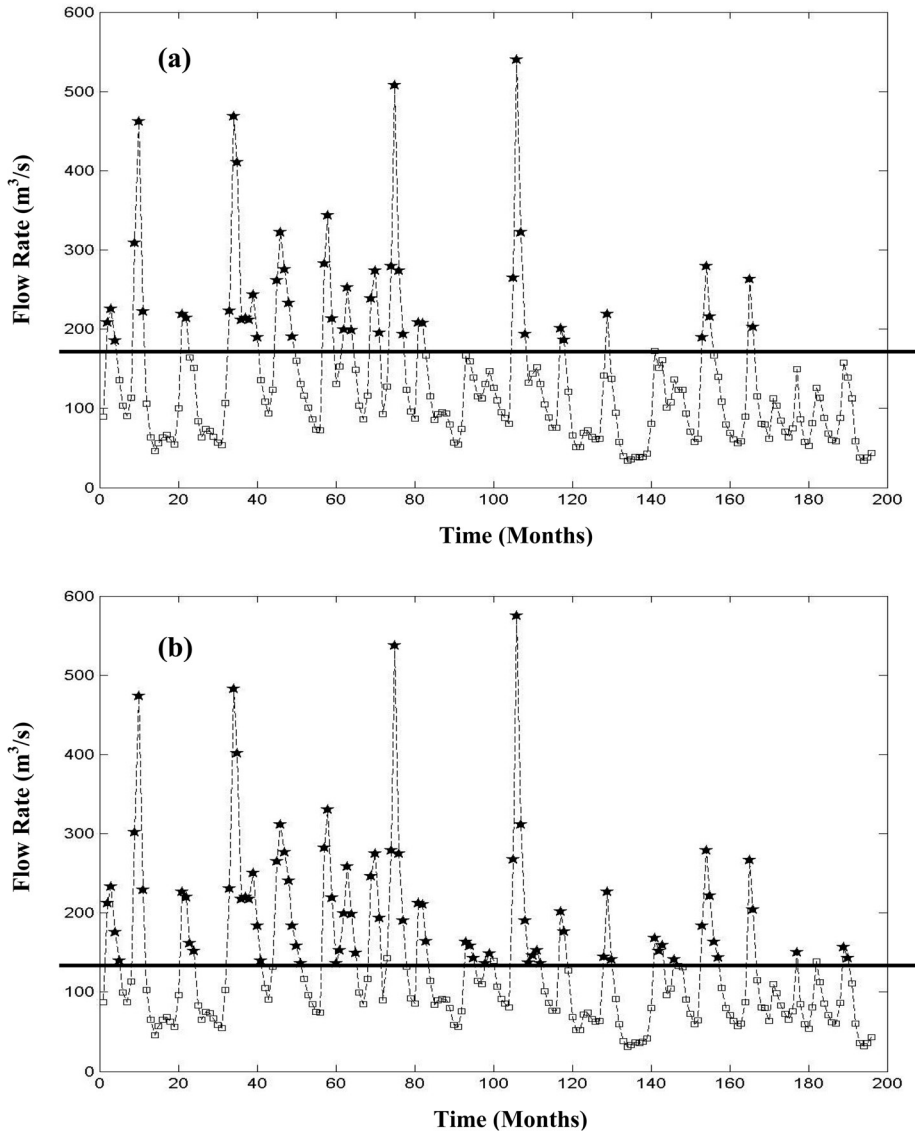
control stage followed by a soil control stage. Different dynamics of state variables govern each of these stages. Apart from these two distinct stages, it should be noted that different combinations of evaporation, ground and sensible heat flux with net radiation could satisfy the energy balance (equation (5)). Hence the performance of SMNNs in capturing such discontinuous input-output relationship would be of particular interest. Performance statistics of different neural network models for estimating actual ET during both training and testing are presented in Table 4. Modeling of evaporation flux using FFNNs resulted in an RMSE of  $73.4 \text{ W}/\text{m}^2$ , an MRE of 1.5, and an  $R$  of 0.69. Comparing the performance of FFNN with SMNNs, both SMNN (Competitive) and SMNN(SOM) outperformed FFNN in estimating evaporation flux. This reiterates the fact that the input-output relationship is discontinuous over the mapping horizon, and hence modular neural networks could offer a promising alternative to capture such discontinuous input-output mapping.

[33] It can be noted, in general, that the training RMSE of the neural network models showed significant variations

**Table 2.** Statistical Performance of Different Models in Modeling Streamflows<sup>a</sup>

Model	Training			Testing		
	RMSE, $\text{m}^3/\text{s}$	MRE	$R$	RMSE, $\text{m}^3/\text{s}$	MRE	$R$
FFNN	27.9	0.17	0.95	30.4	0.22	0.96
SMNN(Competitive)	28.4	0.17	0.95	27.5	0.22	0.96
SMNN(SOM)	28.8	0.17	0.94	28.4	0.22	0.96

<sup>a</sup>RMSE is root-mean-square error; MRE is mean relative error; FFNN is feed forward neural network; SMNN is spiking modular neural network; SOM is self-organizing map.



**Figure 5.** Plots showing the instances at which different spiking layer neurons fired: (a) SMNN(Competitive) and (b) SMNN(SOM). Solid lines indicate threshold value, stars indicate instances at which spiking layer neuron 1 fired, and open rectangles indicate instances at which spiking layer neuron 2 fired.

with the testing RMSE (Table 4). Considerable variability in RMSE statistics between training and testing is due to the following reason: the medians of LE data sets during training and testing are  $34.1 \text{ W/m}^2$  and  $61.2 \text{ W/m}^2$ , respectively. Since median gives a measure of central tendency, it

implies that, compared to the training data set, the testing data set is dominated more by higher values of evapotranspiration. Roughly, the values of evapotranspiration in testing data set are twice ( $61.2/34.1$ ) as high as the values of evapotranspiration in the training data set. *Karunanithi et al.*

**Table 3.** RMSE and MRE Statistics of Different Models Above and Below the Threshold Streamflow Modeling<sup>a</sup>

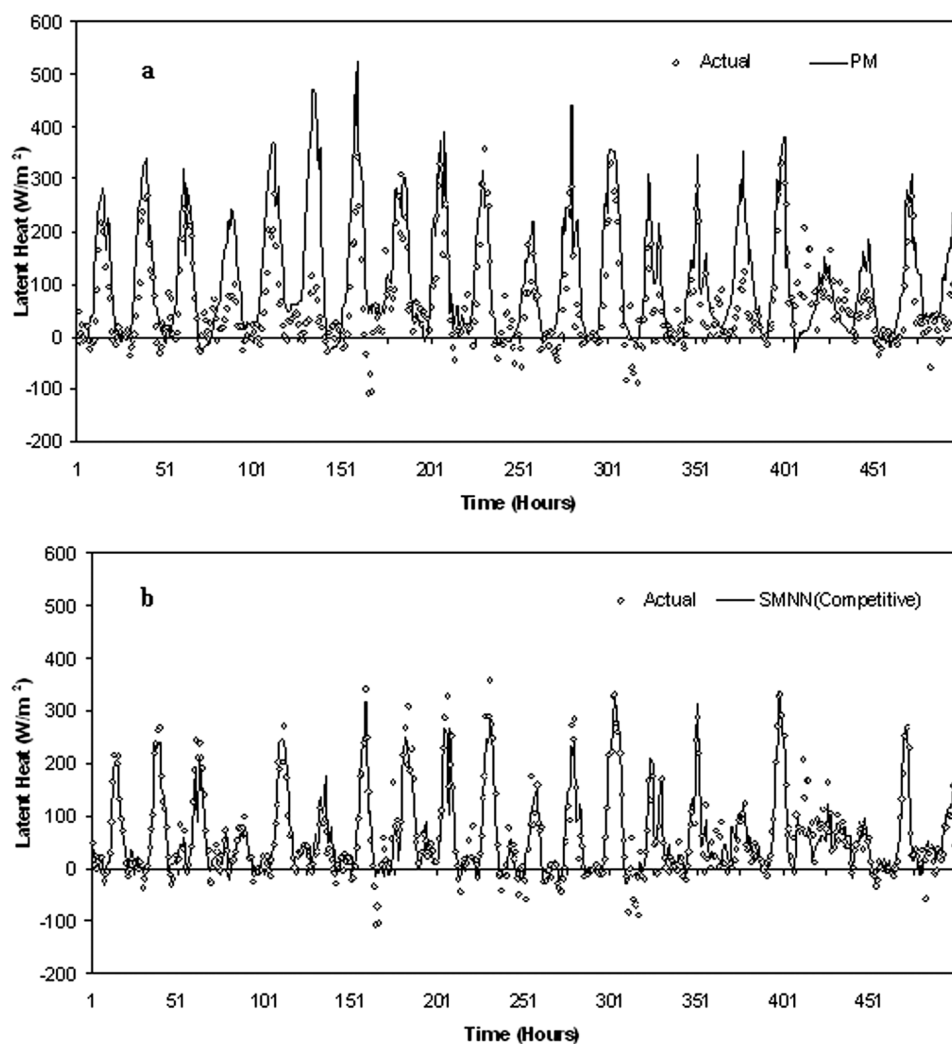
Model	Flow Rate < 200 m <sup>3</sup> /s		Flow Rate > 200 m <sup>3</sup> /s	
	RMSE, m <sup>3</sup> /s	MRE	RMSE, m <sup>3</sup> /s	MRE
FFNN	24.2	0.24	53.5	0.13
SMNN(Competitive)	24.1	0.24	41.9	0.12
SMNN(SOM)	25.4	0.25	41.4	0.12

<sup>a</sup>Abbreviations as in Table 2.

**Table 4.** Statistical Performance of Different Models in Modeling Evaporation<sup>a</sup>

Model	Training			Testing		
	RMSE, W/m <sup>2</sup>	MRE	R	RMSE, W/m <sup>2</sup>	MRE	R
FFNN	37.0	2.4	0.90	73.4	1.6	0.69
SMNN(Competitive)	32.2	2.2	0.93	70.2	1.2	0.71
SMNN(SOM)	33.4	2.0	0.92	73.0	1.3	0.67

<sup>a</sup>Abbreviations as in Table 2.



**Figure 6.** Comparison of measured evaporation flux with (a) Penman-Monteith and (b) SMNN(Competitive) estimates.

[1994] demonstrated that MSE and MRE provide different types of information about the predictive capability of the model. In their work, it has been shown that MSE is more sensitive to errors at high and low values, whereas MRE provides a more balanced perspective of the goodness of fit at moderate values. Since squared-error statistics give more weight to high values, RMSE during testing is approximately twice higher than the RMSE during training, preserving the ratio between the medians. This illustrates that the neural network models are not overtrained. SMNN(Competitive) resulted in an RMSE of  $70.2 \text{ W/m}^2$ , an MRE of 1.2, and an  $R$  of 0.71. However, SMNN(SOM) resulted in an RMSE of  $73.0 \text{ W/m}^2$ , an MRE of 1.3, and an  $R$  of 0.67. From Table 4 it can be concluded that SMNN(Competitive) provides a more generalized representation of the evaporation process. Analysis of results obtained from SMNNs reveals that different combinations of inputs may lead to the same value of evaporation flux. This was evident when similar values of evaporation flux were obtained even when different spiking layer neurons spiked (i.e., inputs from different clusters). The above effect is of particular interest as it illustrates that the SMNN as a data driven model is able to confirm that different combi-

nation of state variables can satisfy the energy balance equation.

[34] Figure 6 compares PM and SMNN(Competitive) estimates of evaporation with the EC-measured evaporation flux between 20 May 2003 and 9 June 2003. Since PM estimates potential evaporation, and in reality water is not always freely available (supply limited) to evaporate, PM overestimates evaporation during supply-limited conditions. Although it is not prudent to directly compare the PM estimates with the neural networks models predicting EC-measured LE flux, the above comparison is made because of the following reason: *Abbott et al.* [1986] stated that the PM method, which accounts for the influence of vegetation on evapotranspiration, has been used frequently to model the evapotranspiration flux. The widespread utility of PM method in characterizing evaporation is due to its ability in predicting evaporation on the basis of readily available climatic data like solar radiation, relative humidity, wind speed, and air temperature. The potential estimates of evaporation given by the PM method can be converted to actual evaporation by considering the soil moisture limitations. However, compared to climatic data, the soil moisture data is not always readily available at the temporal resolu-

**Table 5.** Statistical Performance of Different Models in Modeling Evaporation With Net Radiation and Ground Temperature Alone as Inputs<sup>a</sup>

Model	Training			Testing		
	RMSE, W/m <sup>2</sup>	MRE	R	RMSE, W/m <sup>2</sup>	MRE	R
FFNN	43.7	3.0	0.86	67.2	1.5	0.72
SMNN(Competitive)	43.9	3.3	0.86	64.4	1.1	0.74
SMNN(SOM)	43.9	3.5	0.86	65.9	0.9	0.73

<sup>a</sup>Abbreviations as in Table 2.

tion of other climatic variables. Furthermore, soil moisture shows large variability in both spatial and temporal scales [Entekhabi *et al.*, 1996]. Hence interpolation of the soil moisture data to match the temporal resolution of climatic variables is not prudent. Because of the above limitations of soil moisture data availability as well as its large temporal and spatial variability, conversion of potential evaporation to actual evaporation is cumbersome and involves large uncertainty. In this regard, the utility of neural networks model in directly predicting actual evaporation from climatic data alone is tested and compared with the PM estimates that would be used otherwise. The RMSE between the measured and the PM-estimated evaporation flux is 88.2 W/m<sup>2</sup>, which is comparatively higher than the RMSE of 32.2 W/m<sup>2</sup> obtained by SMNN(Competitive). For the period between 18 June 2003 and 28 June 2003 (testing data), the RMSE between measured and PM-estimated evaporation flux is 92.5 W/m<sup>2</sup>, which is again significantly greater than the RMSE of 70.2 W/m<sup>2</sup> obtained by SMNN(Competitive). This illustrates the better performance of ANNs against PM method in modeling EC-measured evaporation flux as a function of climatic data alone.

### 6.2.1. Identification of Optimal Combination of Input Variables

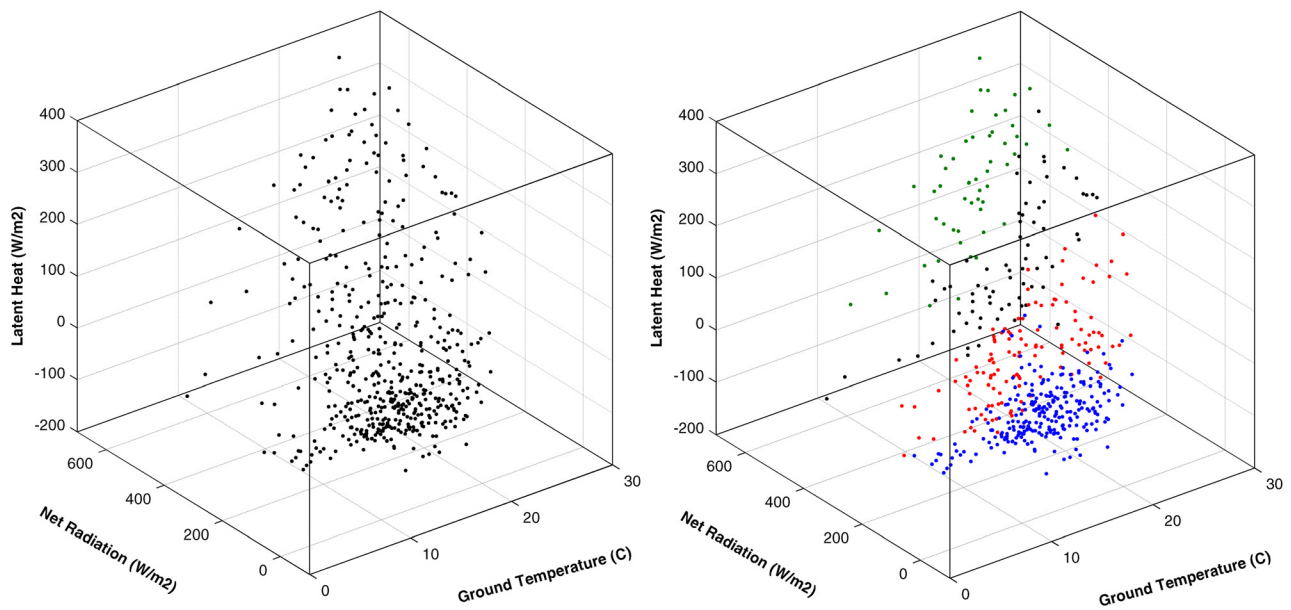
[35] Since this case study represents a MISO process, the study is further extended to find the optimal combination of inputs that can characterize the evaporation process effectively. Also, the process of evapotranspiration is controlled by different factors at different scales (vapor pressure deficit and stomatal processes at the scale of single leaf or tree) whereas radiation is the driving variable at regional scale [Jarvis and McNaughton, 1986]. Different combinations of inputs were tested with the objective of minimizing equation (3). Results indicate that the use of net radiation and ground temperature alone as inputs to neural network models can result in better prediction accuracy. Although most of the evaporation models use a water vapor pressure gradient to estimate evaporation, inclusion of RH as one of the inputs to the neural networks model does not improve the performance of the model, as RH is a somewhat redundant variable for the ANN model; the ANN model has already learned the signal of RH which is embedded in the signal of GT, because of strong land-atmosphere interaction. This reiterates the findings of Lakshmi and Susskind [2001] and Wang *et al.* [2004], where it is reported that evaporation is not sensitive to atmospheric humidity since the land surface states contain the signals of near-surface atmospheric conditions as a result of strong land-atmosphere interaction. Inclusion of WS as one of the input variables to the neural networks model does not improve the

performance of the model. Compared to neural network models using AT and NR as inputs, models using GT and NR resulted in better performance. As will be discussed later, GT acts in part as a surrogate variable to soil moisture and also has a longer memory of the feedback process inherent in the evaporation process. Also, Figure 3 reiterates that the best combination of inputs could be GT and NR as they have the greatest correlation with evaporation flux.

[36] Table 5 presents the statistical performances of different neural network models using only net radiation and ground temperature as inputs. The optimal number of neurons in the spiking layers of both SMNN(Competitive) and SMNN(SOM) is four. The architecture of the associator neural networks is ANN(2, 4, 1). The SMNNs performed better than the FFNNs (Table 5), and in general, the use of net radiation and ground temperature alone as inputs resulted in an increase in the training error (Tables 4 and 5). However, during testing, better performance was obtained, indicating neural network models using net radiation and ground temperature alone as inputs have better generalization properties than do the neural network models utilizing all five inputs (air temperature, ground temperature, net radiation, relative humidity, and wind speed). The performance improvement in testing results and deterioration in training results are more apparent with regard to SMNNs, which is attributed to better generalization achieved because of the parallelization property of SMNNs.

[37] The rate of evaporation is largely controlled by the energy and moisture available for evaporation. During “energy-limited” conditions, the energy balance at the land-atmosphere boundary layer determines the direction of movement of water vapor. Nevertheless, during “supply-limited” conditions, the water balance between the land and the atmosphere determines the rate of evaporation. Net radiation is the major factor influencing evaporation during energy-limited conditions, and soil moisture is the most influential factor in determining evaporation during supply-limited conditions. Eltahir [1998] showed that an increase in soil moisture decreases the Bowen ratio, resulting in a decrease of ground temperature. By extension, ground temperature can be considered as a surrogate variable for soil moisture because of the strong link between soil thermal properties and moisture status. This provides support for the optimal combination of inputs (NR and GT) for the neural networks model. While NR accounts for energy-limited conditions, GT, as a surrogate, accounts for supply-limited conditions.

[38] In order to demonstrate the modular learning of SMNNs, instances at which different spiking layer neurons spiked the corresponding associator neural networks are presented in Figure 7. The scatterplot on the left shows the variation of latent heat with respect to ground temperature and net radiation. The scatterplot on the right shows the mapping space associated with each associator neural network module. The mapping space of each associator neural network’s module is represented by differently coloured points, which show that the SMNNs are effective in discretizing the complex mapping space into simpler domains that can be learned better. As mentioned before, clustering is carried out on the basis of unsupervised learning (i.e., on the basis of inputs alone). Points that are close to each other in the input space ideally should also be close to each



**Figure 7.** Scatterplots illustrating the performance of SMNN.

other in the output space (i.e., points in one 2-D cluster based on the inputs should be in the same 3-D cluster based on the inputs and output). Figure 7 (3-D space) shows that there are few points (points that are in one cluster in the 2-D space and are not in the same cluster in the 3-D space) in certain regions of the input-output space. Those few points demonstrate that different combinations of input variables (GT and NR) can result in similar output (LE).

### 6.2.2. Partitioning Analysis

[39] Similar to the previous case study, partitioning analysis is carried out to assess the relative strengths of different models in predicting the evaporation flux above and below a certain threshold value. The testing data set is considered for this analysis; it consists of 247 instances. The  $\mu$  and the  $\sigma$  of the data are  $87.12 \text{ W/m}^2$  and  $85.36 \text{ W/m}^2$ , respectively. A value slightly less than  $\mu + \sigma$  is considered as the threshold ( $150 \text{ W/m}^2$ ) value. Out of the total 247 instances, 200 instances are below the threshold value, and the remaining 47 instances are above the threshold value. Initially, partitioning analysis was carried out for models using AT, GT, NR, RH, and WS as inputs. The relative performance of these models above and below the threshold indicates that the SMNNs performed better than the FFNNs in modeling values below the threshold (low-evaporation flux) (Table 6). The performances of FFNNs and SMNNs are comparable for the values above the threshold, indicating that SMNNs are more robust in capturing the dynamics of low-evaporation flux. Partitioning analysis is also carried out for models using GT and NR alone as inputs to the models. Table 7 gives the performances of the models above and below the threshold value. Similar to the previous case, the SMNNs performed better than the FFNNs in modeling the low-evaporation flux and on par with the FFNNs in modeling high evaporation flux.

## 7. Summary and Conclusions

[40] In this study, a novel neural networks model called the spiking modular neural networks (SMNNs) was pro-

posed. Two variants of SMNNs were developed. The first variant, SMNN(Competitive), made use of a competitive layer as a spiking layer, and the second variant, SMNN(SOM), made use of a self-organizing map as a spiking layer. The performance of the models was tested on two different case studies. The first case study involved modeling of streamflows, and the second case study involved modeling of evaporation flux measured by an eddy covariance (EC) system. While the first case study represented a single-input single-output (SISO) process, the second case study represented a multiple-input single-output (MISO) process. The rationale behind choosing these two case studies was to evaluate the performance of SMNNs on both simple and complex hydrological processes.

[41] For the first case study (streamflow modeling), the SMNNs performed slightly better than the regular FFNNs. Comparing SMNN(Competitive) and SMNN(SOM), the performance of the former model was better than that of the latter model. For both the SMNNs the optimal number of neurons in the spiking layer was two, with the first neuron learning the dynamics of low flows and the second neuron learning the high flows. Partitioning analysis was carried out with respect to the performance of different models. It revealed that the performance of SMNNs is on par with that of FFNNs for low flows. However, SMNNs perform better than the regular FFNNs in modeling high flows.

**Table 6.** RMSE and MRE Statistics of Different Models Above and Below the Threshold When Air Temperature, Ground Temperature, Net Radiation, Relative Humidity, and Wind Speed Are Considered as Inputs<sup>a</sup>

Model	Evaporation Flux < 150 W/m <sup>2</sup>		Evaporation Flux > 150 W/m <sup>2</sup>	
	RMSE, W/m <sup>2</sup>	MRE	RMSE, W/m <sup>2</sup>	MRE
FFNN	61.3	1.8	111.1	0.4
SMNN(Competitive)	57.4	1.4	109.2	0.3
SMNN(SOM)	61.3	1.6	109.5	0.4

<sup>a</sup>Abbreviations as in Table 2.

**Table 7.** Root-Mean-Square Error and Mean Relative Error Statistics of Different Models Above and Below the Threshold When Ground Temperature and Net Radiation Are Considered as Inputs<sup>a</sup>

Model	Evaporation Flux < 150 W/m <sup>2</sup>		Evaporation Flux > 150 W/m <sup>2</sup>	
	RMSE, W/m <sup>2</sup>	MRE	RMSE, W/m <sup>2</sup>	MRE
FFNN	56.6	1.8	100.4	0.3
SMNN(Competitive)	53.5	1.2	98.2	0.3
SMNN(SOM)	55.0	1.0	99.6	0.3

<sup>a</sup>Abbreviations as in Table 2.

[42] For the second case study, initially the hourly latent heat flux was modeled as a function of air temperature (AT), ground temperature (GT), net radiation (NR), relative humidity (RH), and wind speed (WS). The optimal number of clusters in the case of SMNNs was eight. The SMNNs were found to perform better than the FFNNs in modeling evaporation flux. Results from the study revealed that different combinations of inputs may lead to similar values of evaporation flux, which indicates that the climatic variables are highly correlated with each other.

[43] Since the second case study is a MISO process, the study was extended to find the optimal combination of input variables. Although most evaporation models use water vapor pressure gradient to estimate evaporation, inclusion of RH as one of the inputs to the neural networks model, did not improve the performance of the neural networks model, reiterating the findings of *Lakshmi and Suskind* [2001] and *Wang et al.* [2004]. For modeling EC-measured evaporation, the optimal combination of inputs was GT and NR. Partitioning analysis carried out to assess the relative strengths of different models in predicting the evaporation flux above and below a certain threshold value showed that SMNNs outperformed FFNNs in modeling low-evaporation flux and on par with FFNNs in modeling high-evaporation flux. It should be noted that the performance of the neural networks model depends on the data used for training the model. A neural networks model with good generalization ability is expected to perform better on sites similar to the one used for training the model. Hence testing the robustness of the developed models on a nearby site may help in strengthening the results. Nonetheless, testing the robustness of the developed model on a completely different site may require retraining the model.

[44] In general, for both case studies, SMNNs were found to perform better than FFNNs. In the study, it is shown that SMNNs were successful in breaking down a complex mapping space into multiple relatively simpler mapping spaces that can be modeled with relative ease. The result from the study supports the findings of *Zhang and Govindaraju* [2000]. As mentioned previously, the main difference between SMNN (Competitive) and SMNN (SOM) is that the former model makes use of a competitive layer as the spiking layer, while the latter model makes use of SOM as the spiking layer. Functionally, SMNN (Competitive) learns the distribution of input space alone, and SMNN (SOM) learns both the distribution and the topology of the input space. Since SMNN(Competitive) performed better than SMNN(SOM), it can be concluded that topology learning

does not improve the performance of the SMNN model. This is due to the fact that, since individual neural network models are constructed for each cluster, the topology learned during the classification process does not influence the performances of associator neural networks.

[45] The findings reported in this study are preliminary in nature and are based on two different case studies. In order to verify and strengthen the findings of this research, the models have to be tested on further different case studies. Global optimization techniques such as genetic algorithms (GAs) are reported to be more robust than the conventional back-propagation (BP) algorithm in estimating the optimal values of weights and biases of neural networks. Hence the performance of SMNNs can further be improved by using GAs to train the associator neural networks. The proposed SMNNs are computationally intensive since they involve clustering of data and finding the optimal weights and biases of each associator neural network. However, once trained, compared to regular FFNNs, the SMNNs can be used with relative ease to accurately predict the hydrological variable of interest. The study reported in this paper is a step in the direction to develop multiple local models rather than a single global model for hydrological processes.

[46] **Acknowledgments.** The authors acknowledge the financial support of the Natural Sciences and Engineering Research Council (NSERC) of Canada through its Discovery Grants Program and the University of Saskatchewan through the Departmental Scholarship Program. The authors thank the Associate Editor (Steven Margulis) and three anonymous reviewers, whose comments greatly improved the quality of the paper.

## References

- Abbott, M. B., J. C. Bathurst, J. A. Cunje, P. E. O'Connell, and J. Rasmussen (1986), An introduction to the European hydrological system—System hydrologique Europeen, 'SHE', 1: History and philosophy of a physically-based, distributed modeling system, *J. Hydrol.*, *87*, 45–59.
- ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000a), Artificial neural networks in hydrology. I: Preliminary concepts, *J. Hydrol. Eng.*, *5*(2), 115–123.
- ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000b), Artificial neural networks in hydrology. II: Hydrologic applications, *J. Hydrol. Eng.*, *5*(2), 124–137.
- Baker, J. M., and T. J. Griffis (2005), Examining strategies to improve the carbon balance of corn/soybean agriculture using eddy covariance and mass balance techniques, *Agric. For. Meteorol.*, *128*, 163–177.
- Blaney, H. F., and W. D. Criddle (1950), Determining water requirements in irrigated areas from climatological irrigation data, *Tech. Pap.* 96, 48 pp., U.S. Dep. of Agric., Nat. Resour. Conserv. Serv., Washington, D. C.
- Bowden, G. J., G. C. Dandy, and H. R. Maier (2005), Input determination for neural network models in water resources applications. Part 1: Background and methodology, *J. Hydrol.*, *301*, 75–92.
- Brown, M., and C. Harris (1994), *Neurofuzzy Adaptive Modeling and Control*, Prentice-Hall, Upper Saddle River, N. J.
- Brutsaert, W. H. (1982), *Evaporation Into the Atmosphere*, Springer, New York.
- Cai, S., H. Toral, J. Qiu, and J. S. Archer (1994), Neural network based objective flow regime identification in air-water two phase flow, *Can. J. Chem. Eng.*, *72*, 440–445.
- Demuth, H., and M. Beale (2001), *Neural Network Toolbox User's Guide: Matlab User's Guide*, The Math Works Inc., Natick, Mass.
- Drexler, J. Z., R. L. Snyder, D. Spano, and K. T. Paw (2004), A review of models and micrometeorological methods used to estimate wetland evapotranspiration, *Hydrol. Processes*, *18*, 2071–2101.
- Eltahir, E. A. B. (1998), A soil moisture–rainfall feedback mechanism: 1. Theory and observations, *Water Resour. Res.*, *34*(4), 765–776.
- Entekhabi, D., I. Rodriguez-Iturbe, and F. Castelli (1996), Mutual interaction of soil moisture state and atmospheric processes, *J. Hydrol.*, *184*, 3–17.
- Hargreaves, G. H., and Z. A. Samani (1982), Estimating potential evapotranspiration, *J. Irrig. Drain. Eng.*, *108*(3), 225–230.
- Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, 2nd ed., Macmillan, New York.

- Holdridge, L. R. (1962), The determination of atmospheric water movements, *Ecology*, *43*, 1–9.
- Hong, Y., K. Hsu, S. Sorooshian, and X. Gao (2005), Self-organizing non-linear output (SONO): A neural network suitable for cloud patch-based rainfall estimation at small scales, *Water Resour. Res.*, *41*, W03008, doi:10.1029/2004WR003142.
- Hsu, K., H. V. Gupta, X. Gao, S. Sorooshian, and B. Imam (2002), Self-organizing linear output (SOLO): An artificial neural network suitable for hydrologic modeling and analysis, *Water Resour. Res.*, *38*(12), 1302, doi:10.1029/2001WR000795.
- Jackson, R. D., S. B. Idso, and R. J. Reginato (1976), Calculation of evaporation rates during the transition from energy-limiting to soil-limiting phases using albedo data, *Water Resour. Res.*, *12*(1), 23–26.
- Jarvis, P. G., and K. G. McNaughton (1986), Stomatal control of transpiration: Scaling up from leaf to region, *Adv. Ecol. Res.*, *15*, 1–49.
- Jensen, K. H. (1981), Unsaturated flow and evapotranspiration modeling as a component of the European hydrologic system (SHE), in *Modeling Components of Hydrologic Cycle*, edited by V. P. Singh, pp. 235–252, Water Resour. Publ., Highlands Ranch, Colo.
- Karunanithi, N., W. J. Grenney, D. Whitley, and K. Bovee (1994), Neural networks for river flow prediction, *J. Comput. Civ. Eng.*, *8*(2), 201–220.
- Kohonen, T. (1989), *Self-Organization and Associative Memory*, Springer, New York.
- Kumar, M., N. S. Raghuvanshi, R. Singh, W. W. Wallender, and W. O. Pruitt (2002), Estimating evapotranspiration using artificial neural network, *J. Irrig. Drain. Eng.*, *128*(4), 224–233.
- Lakshmi, V., and J. Susskind (2001), Utilization of satellite data in land-surface hydrology: Sensitivity and assimilation, *Hydrol. Processes*, *15*(5), 877–892.
- Leuning, R., and M. J. Judd (1996), The relative merits of open- and closed-path analysers for measurements of eddy fluxes, *Global Change Biol.*, *2*, 241–253.
- Linacre, E. T. (1977), A simple formula for estimating evaporation rates in various climates, using temperature data alone, *Agric. Meteorol.*, *18*, 409–424.
- MacKay, D. J. C. (1992), Bayesian methods for adaptive models, Ph.D. thesis, Calif. Inst. of Technol., Pasadena, Calif.
- Maier, H., and G. Dandy (2000), Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications, *Environ. Modell. Software*, *15*(1), 101–124.
- Minns, A. W., and M. J. Hall (1996), Artificial neural networks as rainfall runoff models, *Hydrol. Sci. J.*, *41*(3), 399–417.
- Monteith, J. L. (1965), Evaporation and environment, in *The State and Movement of Water in Living Organisms*, edited by G. E. Fogg, *Symp. Soc. Exp. Biol.*, *19*, 205–234.
- Penman, H. L. (1948), Natural evaporation from open water, bare soil and grass, *Proc. R. Soc. London*, *193*, 120–146.
- Priestley, C. H. B., and R. J. Taylor (1972), On the assessment of surface heat flux and evaporation using large scale parameters, *Mon. Weather Rev.*, *100*, 81–92.
- Sajikumar, N., and B. S. Thandaveswara (1999), A non-linear rainfall-runoff model using an artificial neural network, *J. Hydrol.*, *216*, 32–55.
- Salvucci, G. D. (1997), Soil and moisture independent estimation of stage-two evaporation from potential evaporation and albedo or surface temperature, *Water Resour. Res.*, *33*(1), 111–122.
- Saxton, K. E. (1981), Mathematical modeling of evapotranspiration on agricultural watersheds, in *Modeling Components of Hydrologic Cycle*, edited by V. P. Singh, pp. 183–204, Water Resour. Publ., Highlands Ranch, Colo.
- Schotanus, P., F. T. M. Niewstadt, and H. A. R. De Bruin (1983), Temperature measurement with a sonic anemometer and its application to heat and moisture fluxes, *Boundary Layer Meteorol.*, *26*, 81–95.
- Singh, V. P. (1989), *Hydrologic Systems: Watershed Modeling*, vol. 2, Prentice-Hall, Upper Saddle River, N. J.
- Stephens, J. C., and E. H. Stewart (1963), A comparison of procedures for computing evaporation and evapotranspiration, *IAHS Publ.*, *62*, 123–133.
- Sudheer, K. P., A. K. Gosain, D. M. Rangan, and S. M. Saheb (2002), Modelling evaporation using an artificial neural network algorithm, *Hydrol. Processes*, *16*, 3189–3202.
- Sudheer, K. P., A. K. Gosain, and K. P. Ramasastri (2003), Estimating actual evapotranspiration from limited climatic data using neural computing technique, *J. Irrig. Drain. Eng.*, *129*(3), 214–218.
- Temesgen, B., S. Eching, B. Davidoff, and K. Frame (2005), Comparison of some reference evapotranspiration equations for California, *J. Irrig. Drain. Eng.*, *131*(1), 73–84.
- Thornthwaite, C. W. (1948), An approach toward a rational classification of climate, *Geogr. Rev.*, *33*, 55–94.
- Tokar, A. S., and M. Markus (2000), Precipitation runoff modeling using artificial neural network and conceptual models, *J. Hydrol. Eng.*, *5*(2), 151–161.
- Trajkovic, S., B. Todorovic, and M. Stankovic (2003), Forecasting of reference evapotranspiration by artificial neural networks, *J. Irrig. Drain. Eng.*, *129*(6), 454–457.
- Twine, T. E., W. P. Kustas, J. M. Norman, D. R. Cook, P. R. Houser, T. P. Meyers, J. H. Prueger, P. J. Starks, and M. L. Wesely (2000), Correcting eddy-covariance flux underestimates over a grassland, *Agric. For. Meteorol.*, *103*, 279–300.
- Wang, J., G. D. Salvucci, and R. L. Bras (2004), An extremum principle of evaporation, *Water Resour. Res.*, *40*, W09303, doi:10.1029/2004WR003087.
- Webb, E. K., G. I. Pearman, and R. Leuning (1980), Correction of flux measurements for density effects due to heat and water vapour transfer, *Q. J. R. Meteorol. Soc.*, *106*, 85–100.
- Zhang, B., and S. Govindaraju (2000), Prediction of watershed runoff using Bayesian concepts and modular neural networks, *Water Resour. Res.*, *36*(3), 753–762.

S. K. Carey, Department of Geography and Environmental Studies, Carleton University, Ottawa, ON K1S 5B6, Canada.

K. Parasuraman and A. Elshorbagy, Centre for Advanced Numerical Simulation (CANSIM), Department of Civil and Geological Engineering, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada. (amin.elshorbagy@usask.ca)