

---

# Time series forecasting by combining the radial basis function network and the self-organizing map

Gwo-Fong Lin\* and Lu-Hsien Chen

*Department of Civil Engineering, National Taiwan University, Taipei 10617, Taiwan*

---

## Abstract:

Based on a combination of a radial basis function network (RBFN) and a self-organizing map (SOM), a time-series forecasting model is proposed. Traditionally, the positioning of the radial basis centres is a crucial problem for the RBFN. In the proposed model, an SOM is used to construct the two-dimensional feature map from which the number of clusters (i.e. the number of hidden units in the RBFN) can be figured out directly by eye, and then the radial basis centres can be determined easily. The proposed model is examined using simulated time series data. The results demonstrate that the proposed RBFN is more competent in modelling and forecasting time series than an autoregressive integrated moving average (ARIMA) model. Finally, the proposed model is applied to actual groundwater head data. It is found that the proposed model can forecast more precisely than the ARIMA model. For time series forecasting, the proposed model is recommended as an alternative to the existing method, because it has a simple structure and can produce reasonable forecasts. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS neural networks; radial basis function network; self-organizing map; time series forecasting

## INTRODUCTION

Time series modelling and forecasting continues to be an important area in hydrological research and application. Stochastic time series theory has been applied to solve hydrological problems (Bras and Rodriguez-Iturbe, 1985; Brockwell and Davis, 1987; Lin and Lee, 1992, 1994). Traditional methods, such as time series regression, exponential smoothing, and autoregressive integrated moving average (ARIMA), are available for stochastic time series analysis. In particular, the ARIMA model is representative of time series models and has achieved great popularity. Detailed discussions and applications of these models can be found in Box and Jenkins (1976). Regarding the analysis of groundwater head time series, the Box–Jenkins model has been proved to be useful for studying the behaviour of groundwater heads over time (Knotters and Van Walsum, 1997; Van Geer and Zuur, 1997; Ahn, 2000).

In recent years, artificial neural networks have been proposed as a promising alternative approach to time series forecasting. Many successful applications have shown that neural networks provide an attractive alternative tool for time series modelling and forecasting. Generally speaking, neural networks are information-processing systems devised via imitating brain activity. Neural networks let data speak for themselves and have the capability to identify the underlying functional relationship in the data. Theoretically, neural networks are universal functional approximations and can approximate any nonlinear function with arbitrary accuracy (Cybenko, 1989; Hornik *et al.*, 1989; Hornik, 1991). This is a very important advance for neural networks, because the number of possible nonlinear patterns is huge for real-world problems and a good model should be able to approximate them all well (Granger, 1993). Empirically, neural networks have been shown to be effective in modelling and forecasting nonlinear time series with or without noise (Saxen, 1996; Zhang *et al.*,

---

\* Correspondence to: Gwo-Fong Lin, Department of Civil Engineering, National Taiwan University, Taipei 10617, Taiwan.  
E-mail: gflin@ntu.edu.tw

1998). Moreover, many comparisons have been made between neural networks and traditional methods on time series forecasting performance (Kuan and Liu, 1995; Kohzadi *et al.*, 1996; Zhang *et al.*, 1998).

There are a number of reasons to use neural networks for time series analysis. First, neural networks are nonlinear models, which makes them quite flexible and powerful in modelling complex real-world phenomena. Second, neural networks are nonparametric methods. That is, they do not require any assumptions about the underlying model form. This is very important for many real-time series, because one may not know what the underlying data-generating process is. Third, theoretical results show that neural networks can approximate any complex function with arbitrary accuracy given a large enough network. If an underlying functional relationship exists between the inputs and the outputs for any forecasting model, then the accurate identification of this function is very important. All these features make neural networks very useful for time series modelling and forecasting.

Based on the structure of the neural networks and the learning algorithm, various neural network models are frequently proposed to solve time series problems. The back-propagation network is the popular one. However, the back-propagation algorithm has several serious training problems (Wasserman, 1993). First, it tends to yield local optimal solutions. Second, it may produce different results after the training process even when the same training data are used. Finally, its training rate is slow. In order to overcome these problems, we use a radial basis function network (RBFN) instead of the back-propagation algorithm. RBFNs have been widely used for nonlinear systems identification because of their simple topological structure and their ability to reveal how learning proceeds in an explicit manner. RBFNs were first introduced in solving the real multivariate interpolation problem (Powell, 1987). Broomhead and Lowe (1988) then exploited the radial basis function in the design of neural networks. RBFNs have been employed in nonlinear systems identification and time series prediction (Broomhead and Lowe, 1988; Moody and Darken, 1989). More recently, RBFNs increasingly used in hydrologic systems (such as rainfall-runoff forecasting), radar target recognition, and spatial interpolation (Mason *et al.*, 1996; Zhao and Bao, 1996; Fernando and Jayawardena, 1998; Lin and Chen, 2004a,b).

In general, there are two steps in the design of an RBFN. The first step initializes the centres using a clustering method. The second step determines the parameters and minimizes the error with respect to the connecting weights. Between the existing learning algorithms, the main difference resides in the first step. The positioning of the radial basis centres is a crucial problem for an RBFN. An easy solution is to set the radial basis centres equal to the input vectors of the training data set. However, this solution is unrealistic when the dimension of the training data set is large, and hence clustering methods are often applied to reduce the number of centres (Moody and Darken, 1989; Musavi *et al.*, 1992).

A number of clustering methods have been proposed to solve the clustering problem. The clustering methods consist of hierarchical methods, like Ward's minimum variance method, and nonhierarchical methods, such as the  $K$ -means method. However, each clustering method carries its own shortcomings. Ward's minimum variance method tends to be easily affected by the outliers and cannot accommodate large sample sizes. On the other hand, the  $K$ -means method cannot determine the number of clusters. The number of clusters and the starting points are selected randomly. Furthermore, when the number of clusters is too large, there are probably no training data in a cluster. These aforementioned disadvantages reveal that further improvement is still necessary. Recently, owing to increasing computer power and decreasing computer costs, the self-organizing map (SOM) has been employed to solve clustering problems. Chen *et al.* (1995) have demonstrated that an SOM is a superior clustering technique and its advantage over conventional techniques increases with increasing cluster dispersion in the data. Mangiameli *et al.* (1996) showed that the SOM performed the best when compared with the other seven hierarchical clustering methods.

The SOM was developed to simulate brain function (Kohonen, 1990). It can project a high-dimensional input space on a low-dimensional topology so as to allow one to figure out the number of clusters directly by eye. The SOM was first used as an information-processing tool in the fields of speech and image recognition. More recently, the SOM is the most widely investigated and reported method because of its close ties to biological nervous systems, its simplicity, and the wide variety of problem areas to which it might be

applied (Wang *et al.*, 1996; Orwig *et al.*, 1997; Tokutaka *et al.*, 1999; Michaelides *et al.*, 2001; Tennant and Hewitson, 2002). These advantages, coupled with the unsupervised nature of the SOM's learning algorithm, have rendered the SOM an attractive alternative for solving various problems that traditionally have been the domain of conventional statistical and operational research techniques.

In this paper, a time-series forecasting model is developed. The model is based on the combination of an RBFN and an SOM. First, the algorithms and architectures of the RBFN and the SOM are presented. Then, simulated time series with prescribed parameters are used to compare the forecasting performance of the proposed model and the traditional ARIMA model. Finally, the model is applied to actual groundwater head data.

### RADIAL BASIS FUNCTION NETWORK

The RBFN is traditionally used for strict interpolation in multi-dimensional space. An RBFN is composed of three layers: an input layer, a hidden layer and an output layer. The hidden layer of an RBFN is nonlinear, whereas the output layer is linear. The argument of the activation function of each hidden unit in an RBFN computes the Euclidean distance between the input vector and the centre of the hidden unit in the network. The basic architecture of a three-layered neural network is shown in Figure 1. The learning algorithm can be described as follows.

The input data  $Z$  is a  $P$ -dimensional vector,  $Z = [z_1, z_2, \dots, z_p]^T$ . In the structure of the RBFN, the input layer serves only as an input distributor to the hidden layer. The dimensionality of hidden units is the same as that of the input data. The response from the  $j$ th hidden unit for the  $i$ th input data  $z_i$  has the following form:

$$\phi_j(z_i) = \phi(\|z_i - c_j\|) \quad j = 1, 2, \dots, N_h \quad (1)$$

where  $\| \cdot \|$  denotes the Euclidean norm,  $c_j$  is the centre of the  $j$ th unit in the hidden layer,  $\phi(\cdot)$  is the activation function, and  $N_h$  is the number of hidden units. In the structure of the RBFN, the activation function of hidden units is symmetric in the input space, and the output of each hidden unit depends only on the Euclidean distance between the input vector and the centre of the hidden unit. The activation function has

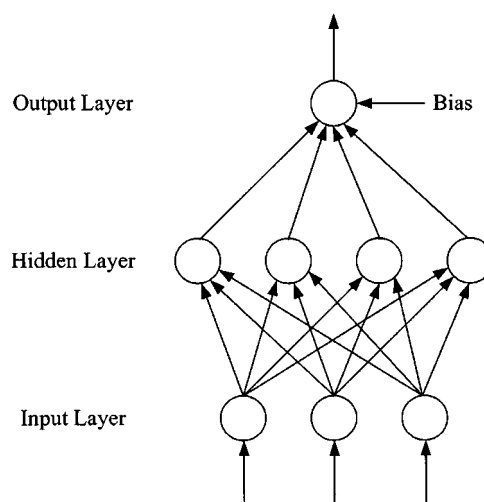


Figure 1. The structure of the RBFN

different forms. The popular form is the Gaussian function:

$$\phi(Z) = \exp\left(-\frac{\|Z - c\|^2}{2\beta^2}\right) \quad (2)$$

where  $\beta$  is the centre width, which can be obtained from (Haykin, 1994)

$$\beta = \frac{d_{\max}}{\sqrt{2N_h}} \quad (3)$$

where  $d_{\max}$  is the maximum distance between the centres of hidden units.

The activity of the  $r$ th unit in the output layer  $\hat{y}_r$  can be obtained from

$$\hat{y}_r = w_0 + \sum_{q=1}^{N_h} w_{qr} \phi_q(Z) \quad r = 1, 2, \dots, N_R \quad (4)$$

where  $\phi_q(Z)$  is the response of the  $q$ th hidden unit resulting from all input data,  $w_{qr}$  is the connecting weight between the  $q$ th hidden unit and the  $r$ th output unit,  $w_0$  is the bias term, and  $N_R$  is the number of output units.

Once the centres and widths of hidden units are determined, each weight in Equation (4) can be determined by the least-squares method. The number and centres of hidden units are determined using an SOM herein. The structure and methodology of the SOM are described in the next section.

### SELF-ORGANIZING MAP

Figure 2 shows the architecture of the SOM. The SOM has two layers: one is the input layer containing the input neurons, and the other is the output layer (Kohonen layer) with numerous neurons fully connected by every input neuron. The output layer can be one- or two-dimensional. An attractive capability of SOMs is to map the high-dimensional input patterns into a lower dimensional output space and to preserve the topological relations of input patterns. There are three essential processes in the formation of an SOM: a competitive process, a cooperative process and an adaptive process.

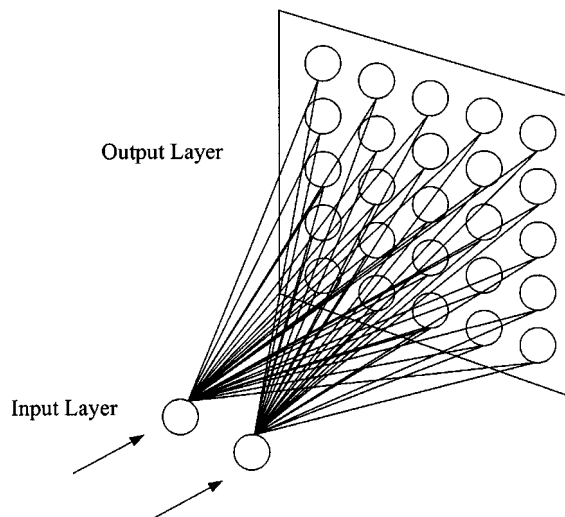


Figure 2. Architecture of the SOM

1. *Competitive process.* Let an  $M$ -dimensional input vector (pattern) be denoted by

$$X = [x_1, x_2, \dots, x_M]^T \tag{5}$$

The output layer includes the output neurons  $u_j, j = 1, 2, \dots, N$ , which are typically organized in a planar (two-dimensional) lattice. Moreover, each connecting line in Figure 2 denotes a value of weight. The weights from the input layer neurons to the output layer neurons are  $w_{ij}, i = 1, 2, \dots, M, j = 1, 2, \dots, N$ . The weight vector of each neuron has the same dimension as the input pattern. The weight vector can be written as

$$W_j = [w_{1j}, w_{2j}, \dots, w_{Mj}]^T \quad j = 1, 2, \dots, N \tag{6}$$

In the competitive process, the neurons of the network compete with each other in order to determine which one is to be activated. The neuron activated is called the winning neuron. The way to determine which neuron is the winning neuron is to measure the similarity between neurons and input patterns. The Euclidean distance  $d_j$  between the weight vector  $W_j$  and input vector  $X$  is frequently used as the similarity measure:

$$d_j = \|X - W_j\| = \sqrt{\sum_{i=1}^M (x_i - w_{ij})^2} \tag{7}$$

The output neuron whose weight vector has the smallest distance from the input vector is called the winning neuron. The weights of this winning neuron are adjusted in the direction of the input vector.

2. *Cooperative process.* In the cooperative process, not only the winning neuron but also the neurons in the topological neighbourhood of the winning neuron are affected by the competition. The topological neighbourhood implies the lateral interactions between the winning neuron and its neighbourhood. The winning neuron is the centre of the topological neighbourhood and the influence of competition decays symmetrically from the winning neuron location. A Gaussian function is a typical choice of topological neighbourhood:

$$h_j = \exp\left(-\frac{\|u_j - u_j^*\|^2}{2\sigma^2}\right) \tag{8}$$

where  $h_j$  is the topological neighborhood,  $\sigma$  is the ‘effective width’ of the topological neighborhood, and  $u_j^*$  is the winning neuron.

3. *Adaptive process.* In the adaptive process, the weights are adjusted according to the input patterns. The adjustment of weights is based on the Hebbian hypothesis (Kohonen, 1995). The change to the weight vector  $W_j$  can be obtained as

$$\Delta W_j = \eta h_j (X - W_j) \tag{9}$$

where  $\eta$  is the learning-rate parameter of the algorithm. Hence, the updating weight vector  $W_j(t + 1)$  at time  $t + 1$  is defined by (Kohonen, 1982)

$$W_j(t + 1) = W_j(t) + \eta(t)h_j(t)(X - W_j(t)) \tag{10}$$

where  $\eta(t)$  and  $h_j(t)$  are the learning-rate parameter and the topological neighbourhood at time  $t$ . The learning-rate parameter  $\eta(t)$  is time varying, as indicated in Equation (10). In particular, it starts at an initial value and then decreases gradually with increasing time.

Upon repeated presentations of the training data, the weight vectors tend to move toward the input pattern due to the neighbourhood updating. That is, the adjustment makes the weight vectors similar to the input pattern. The winning neuron shows the topological location of the input pattern. The neighborhood of the winning neuron shows the statistical distribution of the input pattern. Concerning the number of iterations, as

a general rule this must be at least 500 times the number of neurons in the network (Haykin, 1994). There is no theoretical principle for determining the optimum size of the output layer; hence, the output layer is kept large to ensure that the maximum number of clusters is formed from the training data.

The SOM is an appropriate tool for cluster analysis. After the SOM training is finished, one can figure out the number of clusters directly by eye according to the two-dimensional feature map. The way to obtain the feature map is to mark all winning neurons (some specific grids) in the output space (the lattice) with the symbol (identity) of the corresponding input patterns. The location of the winning neuron in the output space shows the topological location of the corresponding input pattern in the input space, and the density of neurons shows the statistical distribution of the input patterns. If a neuron responds to a specific input pattern, then the grid representing the neuron in the output space (i.e. the feature map) is called the image of the specific input pattern. Every pattern in the input space has only one image in the feature map, but one neuron can represent the images of many input patterns. Moreover, if there are no input patterns in the input space, then the void is also shown in the feature map. This property enables the feature map obtained by labelling each grid therein to reveal the grouping of the input patterns. The number in each grid indicates how many input patterns project to this grid. Suppose that the number in each grid is the 'elevation' of the feature map. Then the grouping of input patterns can be shown by the 'plateaus' separated by 'valleys' on the feature map. The 'plateaus' and valleys imply the groupings and the void of the input patterns that are mapped into the feature map respectively. Hence, the grouping of the original input patterns can be easily recognized according to the variation of the elevation on the feature map.

#### EXPERIMENTAL DESIGN

Can neural networks approximate and forecast well the underlying structure of linear time series? What is the relative performance of neural networks when compared with traditional ARIMA methods for linear time series forecasting? To answer these questions, we conduct a simulation study with three linear time series generated from ARIMA models in this section.

To control for the differences in the characteristics of time series better, three linear time series are generated from the well-known Box–Jenkins ARIMA( $p, d, q$ ) family (Box and Jenkins, 1976). The three types of ARIMA process employed in this study are: listed below:

1. The AR(1) process

$$y_t = y_{t-1} + \varepsilon_t \quad (11)$$

2. The AR(2) process

$$y_t = y_{t-1} - 0.6y_{t-2} + \varepsilon_t \quad (12)$$

3. The ARIMA(1, 1, 1) process

$$(1 - B)y_t = 0.5(1 - B)y_{t-1} + \varepsilon_t + 0.5\varepsilon_{t-1} \quad (13)$$

where  $\varepsilon_t$  is white noise. These ARIMA processes represent the most commonly encountered nonseasonal Box–Jenkins models used in hydrological processes.

In total, there are 200 training data and 100 testing data in each time series. The training data are used to construct the RBFN and to estimate the optimal parameters of the ARIMA models. The testing data are used for testing the predictive ability of the forecasting model. The parameters of ARIMA models are estimated using the least-squares method. Regarding the RBFN structure, after the training data are clustered using an SOM, there are four, seven, and five hidden units in the RBFN for the ARIMA processes 1 to 3 respectively.

To learn the models' performance, the models' forecasts in 200 training data sets and 100 testing data sets are both determined. The mean-square-error (MSE) and the median absolute percentage error (MdAPE) are used herein as the performance measures. They are defined as

$$MSE = \frac{1}{T} \sum_{t=1}^T [\hat{y}_t - y_t]^2 \tag{14}$$

where  $\hat{y}_t$  is the forecast at time  $t$ ,  $y_t$  is the actual observation at time  $t$  and  $T$  is the number of forecasts, and

$$MdAPE = \text{Median} \left( \left| \frac{\hat{y}_t - y_t}{y_t} \right| \times 100 \right) \tag{15}$$

Gardner (1983) recommends these two measures and discusses the reasons why they should be used for forecasting comparisons. Although MSE is the most widely used measure of overall accuracy of a forecasting method, it is also the one that incurs the most criticism (Clements and Hendry, 1993). It should be noted that there is no uniformly accepted forecasting error measure. Hence, it is important to understand the advantages as well as the limitations of each error measure in evaluating forecasting models. The relative advantages of MdAPE over MSE and mean absolute percentage error (MAPE) have been discussed (Armstrong and Collopy, 1992; Fildes, 1992).

The performances of the fitted ARIMA model and the proposed RBFN are summarized in Tables I–III. The results apparently show that the proposed RBFN has better performance than the fitted ARIMA model. In addition, the results of the proposed RBFN in the training data set indicate that it can grasp the major trends in all cases. All of them indicate that the proposed RBFN has an efficient ability to learn and a high accuracy.

Table I. Comparative performance of the fitted AR(1) model and the proposed RBFN

Method	Training data		Testing data	
	MSE	MdAPE	MSE	MdAPE
AR(1) model	1.084	0.762	1.009	0.601
Proposed RBFN	1.026	0.749	0.965	0.528

Table II. Comparative performance of the fitted AR(2) model and the proposed RBFN

Method	Training data		Testing data	
	MSE	MdAPE	MSE	MdAPE
AR(2) model	0.990	0.660	1.041	0.709
Proposed RBFN	0.969	0.535	0.901	0.626

Table III. Comparative performance of the fitted ARIMA(1, 1, 1) model and the proposed RBFN

Method	Training data		Testing data	
	MSE	MdAPE	MSE	MdAPE
ARIMA(1, 1, 1) model	0.228	0.298	0.259	0.387
Proposed RBFN	0.220	0.302	0.260	0.384

## APPLICATION

*The study data*

In this paper, actual groundwater head data from Hsiu-Lin Station in southern Taiwan are used. Figure 3 shows the monthly average groundwater head for Hsiu-Lin Station. Groundwater head records are available from 1982 to 2001. In total, there are 240 sets of monthly average groundwater head. Of these 240 data sets, the first 192 data sets are selected as the training set and the remaining 48 data sets are used as the testing set.

*ARIMA model*

First, the sample autocorrelation function (ACF) and partial ACF (PACF) for the original series are shown in Table IV. They appear to be annual or 12 month spikes in the ACF and PACF. The ACF in Table IV clearly exhibits the prima facie advance of seasonal nonstationarity. That is, it is evident that we need a seasonal differencing. The sample ACF and PACF for the seasonally differenced series are presented in Table V. The sample ACF shows a clear sine-cosine phenomenon and the PACF exhibits significant spike at lag 1. Since  $\bar{W} = -0.05$ ,  $S_W = 1.59$ , and  $n = 180$ , the  $t$ -value of  $\bar{W} = -0.05/(1.24/\sqrt{180}) = 0.54$ , which is significant, and thus a deterministic trend is needed. We identify the following process as a tentative model for the series:

$$(1 - \Phi B)(1 - B^{12})Z_t = \theta_0 + a_t \quad (16)$$

where  $Z_t$  denotes a measured groundwater head time series at time  $t$ ,  $B$  is a backward shift operator ( $B^k Z_t = Z_{t-k}$ ),  $\Phi$  is the autoregressive parameter,  $\theta_0$  is a constant, and  $a_t$  is a Gaussian white noise. Using a standard nonlinear estimation procedure of time series software, we obtain the following results:

$$(1 - 0.69B)(1 - B^{12})Z_t = -0.09 + a_t \quad (17)$$

However, the residual ACF for the above fitted model as shown in Table VI has a significant spike at lag 12. We modify the model to an ARIMA(1, 0, 0)  $\times$  (0, 1, 1)<sub>12</sub>:

$$(1 - \Phi B)(1 - B^{12})Z_t = \theta_0 + (1 - \Theta B^{12})a_t \quad (18)$$

where  $\Theta$  is the moving-average parameter. Parameter estimation gives

$$(1 - 0.68B)(1 - B^{12})Z_t = -0.03 + (1 - 0.91B^{12})a_t \quad (19)$$

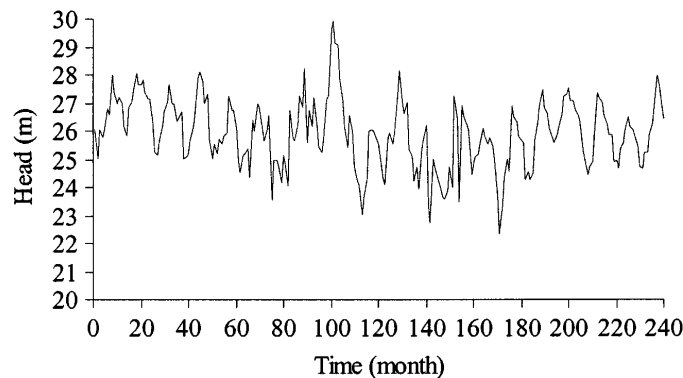


Figure 3. Monthly average groundwater head for Hsiu-Lin Station



Table IV. Sample ACFs and PACFs for monthly average groundwater head at Hsiu-Lin Station<sup>a</sup>

ACF $\hat{\rho}_k$ for $\{Z_t\}$ ; ( $\bar{Z} = 25.89, S_Z = 1.29, n = 192$ )												
$k$	1	2	3	4	5	6	7	8	9	10	11	12
$\hat{\rho}_k$	0.73	0.55	0.40	0.22	0.08	0.00	-0.01	0.00	0.09	0.12	0.18	0.25
SE	0.10	0.12	0.12	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
$k$	13	14	15	16	17	18	19	20	21	22	23	24
$\hat{\rho}_k$	0.21	0.14	0.08	0.00	-0.08	-0.16	-0.14	-0.09	-0.04	0.00	0.10	0.18
SE	0.13	0.13	0.13	0.13	0.13	0.14	0.14	0.14	0.14	0.14	0.14	0.14
$k$	25	26	27	28	29	30	31	32	33	34	35	36
$\hat{\rho}_k$	0.16	0.18	0.17	0.19	0.15	0.09	0.08	0.06	0.05	0.07	0.10	0.13
SE	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.15	0.15
PACF $\hat{\phi}_{kk}$												
$k$	1	2	3	4	5	6	7	8	9	10	11	12
$\hat{\phi}_{kk}$	0.73	0.05	-0.04	-0.14	-0.08	0.01	0.10	0.07	0.14	-0.02	0.07	0.08
SE	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
$k$	13	14	15	16	17	18	19	20	21	22	23	24
$\hat{\phi}_{kk}$	-0.09	-0.09	-0.02	-0.04	-0.03	-0.10	0.10	0.08	-0.01	-0.04	0.13	0.07
SE	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
$k$	25	26	27	28	29	30	31	32	33	34	35	36
$\hat{\phi}_{kk}$	-0.08	0.08	0.04	0.16	-0.04	-0.05	0.06	-0.06	0.00	0.08	0.00	0.00
SE	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07

Standard error.

<sup>a</sup> SE:

Table V. Sample ACFs and PACFs for the seasonally differenced series

ACF $\hat{\rho}_k$ for $\{W_t = (1 - B^{12})Z_t\}$ ; ( $\bar{W} = -0.05, S_W = 1.59, n = 180$ )												
$k$	1	2	3	4	5	6	7	8	9	10	11	12
$\hat{\rho}_k$	0.70	0.55	0.42	0.30	0.17	0.10	0.01	-0.11	-0.15	-0.24	-0.34	-0.50
SE	0.10	0.12	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.14	0.14	0.15
$k$	13	14	15	16	17	18	19	20	21	22	23	24
$\hat{\rho}_k$	-0.34	-0.32	-0.29	-0.29	-0.26	-0.27	-0.23	-0.15	-0.13	-0.11	-0.03	0.02
SE	0.16	0.16	0.16	0.16	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
$k$	25	26	27	28	29	30	31	32	33	34	35	36
$\hat{\rho}_k$	0.05	0.15	0.22	0.31	0.34	0.34	0.31	0.23	0.17	0.15	0.09	0.06
SE	0.17	0.17	0.17	0.18	0.18	0.19	0.19	0.19	0.19	0.19	0.19	0.19
PACF $\hat{\phi}_{kk}$												
$k$	1	2	3	4	5	6	7	8	9	10	11	12
$\hat{\phi}_{kk}$	0.70	0.13	-0.01	-0.05	-0.10	0.01	-0.08	-0.15	0.00	-0.13	-0.18	-0.30
SE	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
$k$	13	14	15	16	17	18	19	20	21	22	23	24
$\hat{\phi}_{kk}$	0.36	-0.03	-0.07	-0.18	-0.04	-0.04	-0.06	0.02	-0.02	-0.16	-0.02	-0.13
SE	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
$k$	25	26	27	28	29	30	31	32	33	34	35	36
$\hat{\phi}_{kk}$	0.28	0.11	0.05	0.03	-0.05	-0.08	0.00	-0.03	-0.05	-0.03	-0.01	-0.04
SE	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07

Table VI. Residual ACFs  $\hat{\rho}_k$  from the fitted model  $(1 - 0.69B)(1 - B^{12})Z_t = -0.09 + a_t$

$k$	1	2	3	4	5	6	7	8	9	10	11	12
$\hat{\rho}_k$	-0.03	0.09	0.16	-0.09	-0.08	0.03	-0.09	-0.12	0.08	-0.04	-0.04	0.00
SE	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
$k$	13	14	15	16	17	18	19	20	21	22	23	24
$\hat{\rho}_k$	0.07	0.01	0.05	0.04	0.03	-0.07	-0.03	0.02	-0.10	-0.11	0.02	-0.36
SE	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.09	0.09	0.09	0.09
$k$	25	26	27	28	29	30	31	32	33	34	35	36
$\hat{\rho}_k$	-0.09	0.11	-0.07	0.10	0.17	0.04	0.06	0.07	-0.15	0.00	-0.01	0.01
SE	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10

Table VII. Residual ACFs  $\hat{\rho}_k$  from the fitted model  $(1 - 0.68B)(1 - B^{12})Z_t = -0.03 + (1 - 0.91B^{12})a_t$

$k$	1	2	3	4	5	6	7	8	9	10	11	12
$\hat{\rho}_k$	-0.03	0.09	0.10	-0.09	-0.08	0.03	-0.09	-0.11	0.08	-0.04	-0.04	0.00
SE	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
$k$	13	14	15	16	17	18	19	20	21	22	23	24
$\hat{\rho}_k$	0.07	0.01	0.05	0.04	0.03	-0.07	-0.03	0.02	-0.10	-0.11	0.02	-0.36
SE	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.09	0.09	0.09	0.09
$k$	25	26	27	28	29	30	31	32	33	34	35	36
$\hat{\rho}_k$	-0.09	0.11	-0.07	0.10	0.17	0.04	0.06	0.07	-0.15	0.00	-0.01	0.01
SE	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10

The residual ACFs for this modified model are shown in Table VII: they are all small and exhibit no patterns. Therefore, the fitted ARIMA(1, 0, 0) × (0, 1, 1)<sub>12</sub> model in Equation (18) is adequate for the time series. That is, one can use the following equation to forecast the future groundwater head time series:

$$Z_{t+l} = 0.68Z_{t+l-1} + Z_{t+l-12} - 0.68Z_{t+l-13} + a_{t+l} - 0.91a_{t+l-12} - 0.03 \tag{20}$$

*The proposed RBFN*

In this study, an RBFN model is developed to predict the 1 month ahead forecast of groundwater head. According to the above results, the past 13, 12, and 1 month average groundwater heads are used as inputs for the proposed RBFN. The functional form of the RBFN is

$$Z_{t+l} = f(Z_{t+l-13}, Z_{t+l-12}, Z_{t+l-1}) \tag{21}$$

Therefore, there are three neurons in the input layer and one neuron in the output layer. The number of neurons in the hidden layer is determined after the SOM is constructed in the development process of the proposed RBFN. After a total of 98 000 iterations, the SOM has been constructed. Figure 4 presents the two-dimensional feature map obtained on a network of 14 × 14 cells. As shown in Figure 4, the feature map can be divided into 15 regions. Therefore, the 192 training data can be grouped into 15 clusters. That is, the RBFN has 15 neurons in the hidden layer. After the 3–15–1 RBFN is constructed, it can be applied to the analysis of groundwater head time series.

*Results and discussions*

The comparisons of observed groundwater heads with values forecast using the seasonal ARIMA model and the proposed RBFN are given in Figures 5 and 6 respectively. According to Figures 5 and 6, the shape and the tendency of the groundwater head can be reasonably forecast using the proposed RBFN. Moreover,

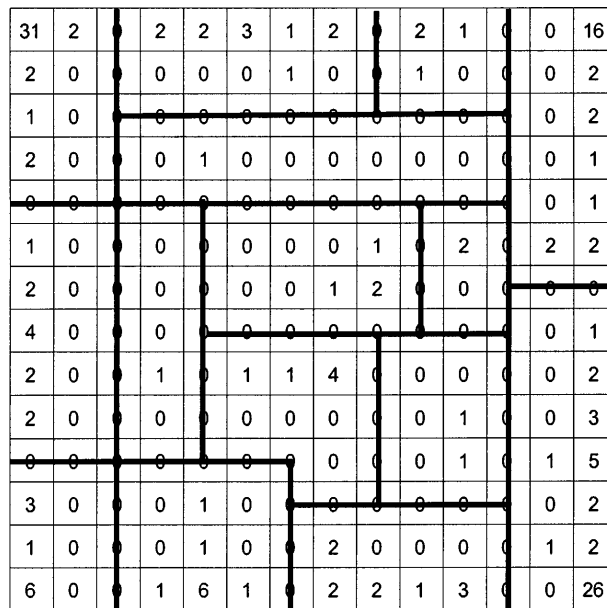


Figure 4. Two-dimensional feature map obtained on a network of 14 × 14 cells. The feature map is divided into 15 regions

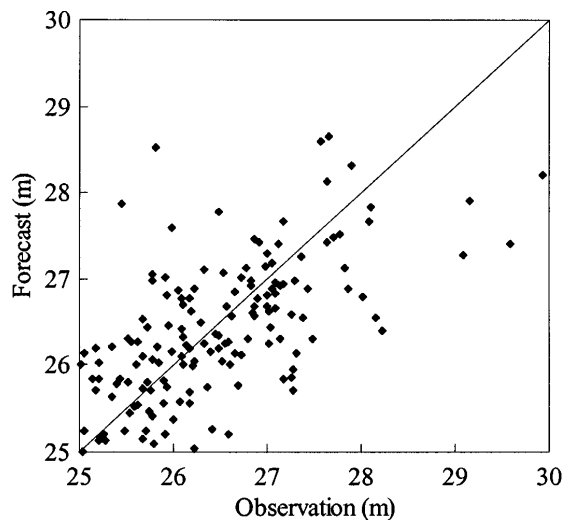


Figure 5. Comparison of observed groundwater heads with values forecast using the ARIMA model

the performances of the seasonal ARIMA model and the proposed RBFN during training and testing are summarized in Table VIII in terms of MSE and MdAPE. For training data, one can see that the values of MSE and MdAPE for the proposed RBFN are 0.06 and 0.38 respectively. Both are better than those of the seasonal ARIMA model, which are 1.15 and 2.39 respectively. The greater accuracy of the proposed RBFN is also evident in testing data, as indicated by MSE = 0.26 and MdAPE = 0.91 versus MSE = 0.42 and MdAPE = 1.46 for the seasonal ARIMA model. Therefore, the proposed RBFN has better performance than the seasonal ARIMA model based on the performance indices in this study. When the number of data is large, the SOM can reduce the number of centres and avoid too many centres for the RBFN. Another advantage of

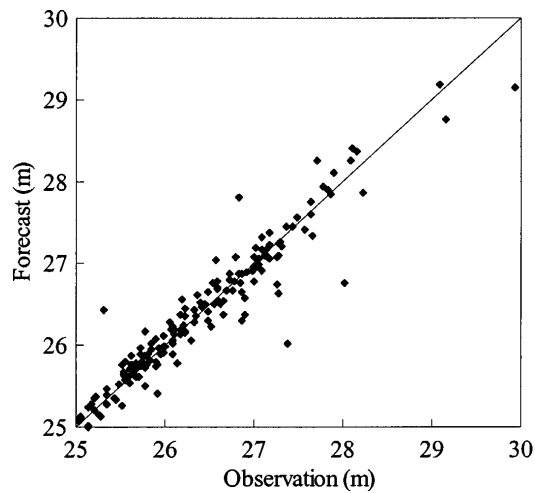


Figure 6. Comparison of observed groundwater heads with values forecast using the proposed RBFN

Table VIII. Comparative performance of the ARIMA model and the proposed RBFN for monthly average groundwater head data at Hsiu-Lin Station

Method	Training data		Testing data	
	MSE	MdAPE	MSE	AdAPE
ARIMA model	1.15	2.39	0.42	1.46
RBFN	0.06	0.38	0.26	0.91

the proposed model is that it has a clear principle and a simple structure. The proposed learning algorithm provides an alternative to the determination of the radial basis centres and it can save on computation time compared with the trial-and-error method.

### SUMMARY AND CONCLUSIONS

In this paper, a time-series forecasting model is developed. The model is based on the combination of an RBFN and an SOM. The SOM is used to construct a two-dimensional feature map from which the number of clusters (i.e. the number of hidden units in the RBFN) can be figured out directly by eye, and then the radial basis centres can be determined easily. In such a manner, the crucial problem for the RBFN, i.e. the positioning of the radial basis centres, can be solved. Time series generated from three types of Box–Jenkins model are used to test the forecasting performance of the proposed model. The results show that the proposed model is more competent in modelling and forecasting time series than the ARIMA model. Application of the proposed model to actual groundwater head data of a well in southern Taiwan indicates that the proposed model can forecast more precisely than the ARIMA model.

### REFERENCES

- Ahn H. 2000. Modeling of groundwater heads based on second-order difference time series models. *Journal of Hydrology* **234**: 82–94.  
 Armstrong JS, Collopy F. 1992. Error measures for generalizing about forecasting methods: empirical comparisons. *International Journal of Forecasting* **8**: 69–80.

- Box GEP, Jenkins GM. 1976. *Time Series Analysis: Forecasting and Control*. Holden-Day: San Francisco.
- Bras RL, Rodriguez-Iturbe I. 1985. *Random Functions and Hydrology*. Addison-Wesley: Reading, MA.
- Brockwell PJ, Davis RA. 1987. *Time Series: Theory and Methods*. Springer-Verlag: New York.
- Broomhead DS, Lowe D. 1988. Multivariable functional interpolation and adaptive networks. *Complex Systems* **2**: 321–355.
- Chen SK, Mangiameli P, West D. 1995. The comparative ability of self-organizing neural networks to define cluster structure. *Omega, International Journal of Management Science* **23**(3): 271–279.
- Clements MP, Hendry DF. 1993. On the limitations of comparing mean square forecast errors. *Journal of Forecasting* **12**: 615–637.
- Cybenko G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematical Control Signals Systems* **2**: 303–314.
- Fernando DAK, Jayawardena AW. 1998. Runoff forecasting using RBF networks with OLS algorithm. *Journal of Hydrologic Engineering* **3**(3): 203–209.
- Fildes R. 1992. The evaluation of extrapolative forecasting methods. *International Journal of Forecasting* **8**: 81–98.
- Gardner ES. 1983. The trade-offs in choosing a time series method. *Journal of Forecasting* **2**: 263–267.
- Granger CWJ. 1993. Strategies for modeling nonlinear time-series relationships. *The Economic Record* **69**(206): 233–238.
- Haykin S. 1994. *Neural Networks: A Comprehensive Foundation*. IEEE Press: New York.
- Hornik K. 1991. Approximation capability of multilayer feedforward networks. *Neural Networks* **4**: 251–257.
- Hornik K, Stinchcombe M, White H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**: 359–366.
- Knotters M, Van Walsum PEV. 1997. Estimating fluctuation quantities from time series of water-table depths using models with a stochastic component. *Journal of Hydrology* **197**: 25–46.
- Kohonen T. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* **43**: 59–69.
- Kohonen T. 1990. The self-organizing map. *Proceedings of the Institute of Electrical and Electronics Engineers* **78**(9): 1464–1480.
- Kohonen T. 1995. *Self-Organizing Maps*. Springer-Verlag: Berlin.
- Kohzadi N, Boyd MS, Kermanshahi B, Kaastra I. 1996. A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing* **10**: 169–181.
- Kuan CM, Liu T. 1995. Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics* **10**: 347–364.
- Lin GF, Chen LH. 2004a. A spatial interpolation method based on radial basis function networks incorporating a semivariogram model. *Journal of Hydrology* **288**(3–4): 288–298.
- Lin GF, Chen LH. 2004b. A non-linear rainfall-runoff model using radial basis function network. *Journal of Hydrology* **289**: 1–8.
- Lin GF, Lee FC. 1992. An aggregation–disaggregation approach for hydrologic time series modelling. *Journal of Hydrology* **138**(3–4): 543–557.
- Lin GF, Lee FC. 1994. Assessment of aggregated hydrologic time series modeling. *Journal of Hydrology* **156**(1–4): 447–458.
- Mangiameli P, Chen SK, West D. 1996. A comparison of SOM neural network and hierarchical clustering methods. *European Journal of Operational Research* **93**: 402–417.
- Mason JC, Price RK, Temme A. 1996. Neural network model of rainfall-runoff using radial basis functions. *Journal of Hydraulic Research* **34**(4): 537–548.
- Michaelides SC, Pattichis CS, Kleovoulou G. 2001. Classification of rainfall variability by using artificial neural networks. *International Journal of Climatology* **21**: 1401–1414.
- Moody J, Darken C. 1989. Fast learning in networks of locally-tuned processing units. *Neural Computation* **4**: 740–747.
- Musavi MT, Ahmed W, Chan KH, Faris KB, Hummels DM. 1992. On the training of radial basis function classifiers. *Neural Networks* **5**: 595–603.
- Orwig RE, Chen H, Nunamaker JF. 1997. A graphical, self-organizing approach to classifying electronic meeting output. *Journal of the American Society for Information Science* **48**(2): 157–170.
- Powell MJD. 1987. Radial basis functions for multivariable interpolation: a review. In *Algorithms for Approximation*, Mason JC, Cox MG (eds.) Carendon Press: Oxford; 143–167.
- Saxen H. 1996. Nonlinear time series analysis by neural networks: a case study. *International Journal of Neural Systems* **7**(2): 195–201.
- Tennant WT, Hewitson BC. 2002. Intra-seasonal rainfall characteristics and their importance to the seasonal prediction problem. *International Journal of Climatology* **22**: 1033–1048.
- Tokutaka H, Yoshihara K, Fujimura K, Iwamoto K, Obu-Cann K. 1999. Application of self-organizing maps (SOM) to Auger electron spectroscopy. *Surface and Interface Analysis* **27**: 783–788.
- Van Geer FC, Zuur AF. 1997. An extension of Box–Jenkins transfer/noise models for spatial interpolation of groundwater head series. *Journal of Hydrology* **192**: 65–80.
- Wang Z, Guerriero A, DeSario M. 1996. Comparison of several approaches for the segmentation of texture images. *Pattern Recognition Letters* **17**: 509–521.
- Wasserman PD. 1993. *Advanced Methods in Neural Computing*. Van Nostrand Reinhold: New York.
- Zhang G, Patuwo BE, Hu MY. 1998. Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting* **14**: 35–62.
- Zhao Q, Bao Z. 1996. Radar target recognition using a radial basis function neural network. *Neural Networks* **9**: 709–720.