Radial Basis Function Modeling of Hourly Streamflow Hydrograph

Nor Irwan Ahmat Nor[1]; Sobri Harun[2]; and Amir Hashim Mohd. Kassim[3]

Abstract: An artificial neural network is well known as a flexible mathematical tool that has the ability to generalize patterns in imprecise or noisy and ambiguous input and output data sets. The radial basis function (RBF) method is applied to model the relationship between rainfall and runoff for Sungai Bekok Catchment (Johor, Malaysia) and Sungai Ketil catchment (Kedah, Malaysia). The RBF is used to predict the streamflow hydrograph based on storm events. Evaluation on the performance of RBF is demonstrated based on errors (between predicted and actual) and comparison with the results of the Hydrologic Engineering Center hydrologic modeling system model. It is obvious that the RBF method offers an accurate modeling of streamflow hydrograph.

DOI: 10.1061/(ASCE)1084-0699(2007)12:1(113)

CE Database subject headings: Rainfall; Runoff; Streamflow; Hydrographs.

Introduction

Determining the relationship between rainfall and runoff for watersheds is one of the most important problems faced by hydrologists and engineers. Problems arise from nonlinearity of physical processes, uncertainty in parameter estimates, ungauged catchments, etc. Runoff estimation is critical to many activities such as designing flood protection works for urban areas and agricultural land and assessing how much water may be extracted from a river for planning, design, and management of water supply, irrigation, and drainage systems.

The rainfall-runoff relationship describes the time distribution of direct runoff as a function of excess rainfall. The rainfall-runoff model is required to ascertain this particular relationship. In actual fact the relationship of rainfall-runoff is known to be highly nonlinear and complex. The spatial and temporal rainfall patterns and the variability of watershed characteristics create more complex hydrologic phenomena. This is mainly in the forms of prediction and estimation of the magnitude of a hydrological variable like runoff using previous rainfall and runoff data. Various well-known currently available rainfall-runoff models such as Hydrologic Engineering Center hydrologic modeling system (HEC-HMS), MIKE-11, SWMM, etc., have been successfully applied in many problems and watersheds. However, the existing popular rainfall-runoff models were not flexible and not robust, and they require many parameters for calibration. Obviously, the models have their own weaknesses, especially in the calibration processes and the ability to adopt the nonlinearity of processes. Therefore, the present study was undertaken to develop rainfall-runoff models using an artificial neural network (ANN) method that can be used to provide reliable and accurate estimates of runoff.

Many approaches have evolved over the last few decades in hydrological modeling and forecasting. They are deterministic as well as stochastic in nature, and include conceptual and statistical methods.

MIKE-11, HEC-HMS, MIKE-SHE, and SWMM are examples of popular hydrologic models. Most of the commercial software generally apply similar methodologies for the computation of loss rates, initial losses, transfer functions, transform rates, lag times, deficits, etc. Computations are based on a limited understanding of the physics behind the hydrologic processes. Therefore, many parameters are required for calibration and good results are rarely achieved, so that empiricism plays an important role in modeling studies in the new era.

The rapid increase in the capacity of modern computers has opened up a new world of methodologies for mathematical modeling. These methodologies focus on the application of a new approach to solve problems in hydrology. The natural behavior of hydrological processes is appropriate for the application of an ANN method. An ANN can be defined as "a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain" (Broomhead and Lowe, 1988). An attractive feature of ANNs is their ability to extract the relation between the inputs and outputs of a process, without the physics being explicitly provided to them, and even if the data is noisy and contaminated with errors. ANN models have been used successfully to model complex nonlinear input-output relationships in many areas of physical sciences. It is a black-box model. The ANN is a robust tool for modeling many of the nonlinear hydrological processes such as rainfall-runoff, river flows and water stages, water quality, and

[1]Faculty of Civil and Environmental Engineering, Kolej Univ. Teknologi Tun Hussein Onn, 86400 Parit Raja, Batu Pahat, Johor, Malaysia. E-mail: nisfan2020@hotmail.com

[2]Dept. of Hydraulic and Hydrology, Faculty of Civil Engineering, Univ. Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia. E-mail: sobri@fka.utm.my

[3]Faculty of Civil and Environmental Engineering, Kolej Univ. Teknologi Tun Hussein Onn, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

Note. Discussion open until June 1, 2007. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this technical note was submitted for review and possible publication on July 25, 2003; approved on February 15, 2006. This technical note is part of the ***Journal of Hydrologic Engineering***, Vol. 12, No. 1, January 1, 2007. ©ASCE, ISSN 1084-0699/2007/1-113–123/$25.00.

groundwater management. The strengths of ANNs are that they can work well even when the data sets contain noise and measurement errors; they are able to recognize the nonlinear relationship of hydrologic processes without requiring physical information and details of case study. Further, they are able to adapt and compensate for any changing circumstances, and are relatively easy to calibrate or train. Meanwhile, the limitations of ANNs are their lack of physical concepts and relations; there are no standardized ways or methods of selecting network structure, and training algorithms. Usually, there are determined by the past experience and preference.

Due to the flexibility of ANNs, researchers continue to apply this technique to various problems. This study employs a radial basis function (RBF) method to model event-based rainfall-runoff relationship. The objectives of this study are to examine and evaluate how successfully ANNs have been utilized in rainfall-runoff modeling. RBF networks are widely used for function approximation, nonlinear modeling, pattern analysis and recognition, modeling of complex and chaotic dynamical systems, etc. (Broomhead and Lowe 1988; Chen et al. 1991; Mann and McLaughlin 2000; Lucks and Oki 1999; Yang and Tseng 1988; Poggio and Girosi 1990; Lu and Evans 1999; Dibike and Solomatine 1999; Heimes and Heuveln 1998).

Study Area

The modeling work is carried out using the previous rainfall and runoff records of Sungai Bekok (Johor, Malaysia), and Sungai Ketil (Kedah, Malaysia) Catchments, as shown in Figs. 1(a and b), respectively. Sungai Bekok is a natural catchment of size 350 km^2. It is located in the southwestern part of Johor, latitude 02° 07' 15" and longitude 103° 02' 30". Meanwhile, the Sungai Ketil Catchment is a semideveloped area and the size is 704 km^2. It is located in the south part of Kedah, the latitude 05° 38′ 20″ and the longitude 100° 48′ 45″. Figs. 1(a and b), also illustrate the location of the rain gauges and water level gauging stations. Rainfall data were obtained from two rain gauges in the Sungai Bekok Catchment, and three rain gauges in the Sungai Ketil Catchment as listed in Tables 1 and 2, respectively.

Artificial Neural Network

The architecture of an ANN is developed by weights between neurons, a transfer function that controls the generation of output in a neuron, and learning laws that define the relative importance of weights for input to a neuron (Caudill 1987). An ANN will process information in a way that it was previously trained on and generate results. Artificial neural networks can learn from experience, generalize from previous examples to new ones, and abstract essential characteristics from inputs containing irrelevant data (Fausett 1994). The main control parameters of an ANN model are interneuron connection strengths, also known as weights, and the biases. RBF networks were introduced by Broomhead and Lowe in 1988. RBF is a supervised, feed forward neural network that uses a linear transfer function for the output units and a nonlinear transfer function (normally Gaussian) for the hidden units. The RBF model consists of three layers: The input layer, where the data are introduced to the network; the hidden layer, where a nonlinear transformation is applied and the data are processed; and the output layer, where the results for given inputs are produced. The input layer simply consists of n units connected by weighted connections to the hidden layer and a possible smoothing factor matrix. The hidden layer of a RBF network consists of a number of nodes and a parameter vector called a "center," which can be considered as the weight vector. The number of nodes in the hidden layer was determined by using the trial and error method. The standard Euclidean distance is used to measure how far an input vector is located from the center. In RBF networks, the design of neural networks is a curve-fitting problem in a high dimensional space (Karunanithi et al. 1994). Training the RBF network implies finding the set of basis nodes and weights. Therefore, the learning process is to find the best fit for the training data.

Table 1. Rain Gauges Used in Calibration of HEC-HMS Model for the Sungai Bekok Catchment

Rain gauge	Latitude	Longitude
At Ladang Union, Yong Peng (2130068)	0.5° 07′ 50″	103° 03′ 00″
At Ladang Yong Peng, Batu Pahat (2031069)	02° 04′ 15″	103° 09′ 10″

Radial Functions

Radial functions are a special class of functions. There are several common types of functions used such as Gaussian, multiquadratic, inverse multiquadratic, and Cauchy. The Gaussian function, G, is the most popular and widely used in RBF networks as the nonlinearity of the hidden layer processing elements (Broomhead and Lowe 1988; Poggio and Girosi 1990; Lu and Evans 1999; Dibike and Solomatine 1999) which, in the case of a scalar input, is

$$G(x) = \exp\left(\frac{(x-c)^2}{r^2}\right) \qquad (1)$$

where x = its input vector, and its parameters are its center c and its radius r. The transfer functions of the nodes are governed by nonlinear functions that are assumed to be approximations of the influence that data points have at the center. Gaussian functions decrease with distance from the center. The Gaussian function responds to a small region of the input space where the Gaussian is centered. The characteristic feature is that their response decreases monotonically with distance from a central point. The center distance, the distance scale, and the precise shape of the radial function are parameters of the model. The key to a successful implementation of these networks is to find suitable centers for the Gaussian functions. The Gaussian is applied to the net input to produce a radial function of the distance between each pattern vector and each hidden unit weight vector. This can be done with supervised learning or an unsupervised approach. The

Table 2. Rain Gauges Used in Calibration of HEC-HMS Model for the Sungai Ketil Catchment

Rain gauge	Latitude	Longitude
At Pulai (5608074)	5° 39′ 25″	100° 53′ 55″
At Hospital Baling (5609072)	05° 40′ 50″	100° 55′ 00″
At Kg. Terabak (5708071)	05° 45′ 05″	100° 53′ 35″

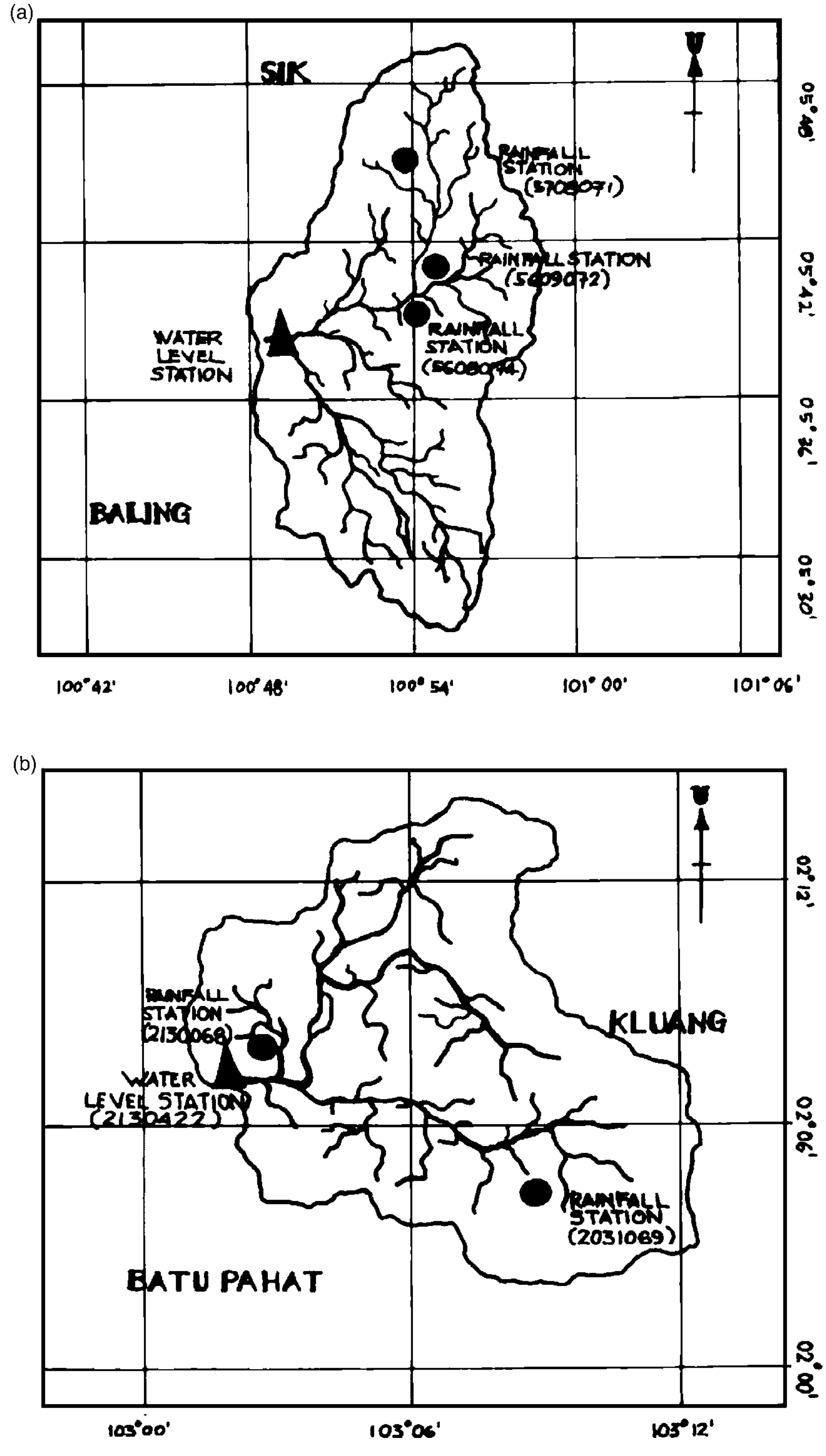

Fig. 1. Catchments area of (a) Sungai Bekok; (b) Sungai Ketil

advantage of the RBF is that it finds the input to output map using local approximations. Usually the supervised segment is simply a linear combination of the approximations. As linear combiners have few weights, these networks train extremely fast and require fewer training samples.

Radial Basis Function Networks

A well known nonlinear modeling approach is the RBF network. RBF networks have only three layer (one input, one hidden, and one output). Fig. 2 illustrates the designed architecture of the RBF. In RBF, the number of RBF "centers" (also called weights) is as good enough as data points and the performance of a RBF network depends upon the chosen centers. However, the problem is how to select the RBF centers especially for a large number of parameters. The most general formula for any RBF is

$$y(x) = \phi((x-c)^T \mathfrak{R}^{-1}(x-c)) \tag{2}$$

where ϕ=activation function used, c=center; and $\mathfrak{R}$=metric. The term $((x-c)^T\mathfrak{R}^{-1}(x-c))$ is the distance between the input x and

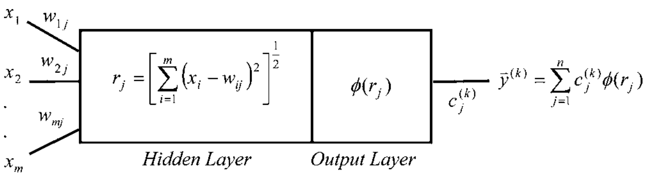

Fig. 2. Structure of a RBF model

the center c in the metric defined by $\mathfrak{R}$. Often the metric is Euclidean. In this case, $\mathfrak{R}=r^2\mathrm{I}$ for some scalar radius r and Eq. (2) simplifies to

$$y(x)=\phi\left(\frac{(x-c)^T(x-c)}{r^2}\right) \tag{3}$$

According to Fausett (1994), the Euclidean length is represented by r_j which measures the radial distance between the datum vector $\mathbf{y}=(y_1,y_2,\ldots,y_m)$, and the radial center $\mathbf{Y}^{(j)}=(w_{1j},w_{2j},\ldots,w_{mj})$, where y_i and w_{ij}=output and weights, respectively. This can be written as

$$r_j=\|\mathbf{y}-\mathbf{Y}^{(j)}\|=\left[\sum_{i=1}^{m}(y_i-w_{ij})^2\right]^{1/2} \tag{4}$$

A suitable transfer function is then applied to r_j to give

$$\phi(r_j)=\phi(\|\mathbf{y}-\mathbf{Y}^{(k)}\|) \tag{5}$$

Finally the output layer ($k=1$) receives a weighted linear combination of $\phi(r_j)$,

$$\vec{y}^{(k)}=\sum_{j=1}^{n}c_j^{(k)}\phi(r_j)=\sum_{j=1}^{n}c_j^{(k)}\phi(\|\mathbf{y}-\mathbf{Y}^{(j)}\|) \tag{6}$$

Training RBF networks

Adapting the values of the weights and centers of networks by presenting the input and output data is known as learning or training. Training in a RBF network may be done in two stages: First, calculating the parameters of the RBF, including centers and the scaling parameters and second, calculation of the weights between the hidden and output layers. Training is the actual process of adjusting weight factors based on trial and error. A supervised training requires target patterns or signals to guide the training process. The objective is to find the weights by minimizing the error between the target and actual output. To train RBF networks, the weight factors were adjusted until the calculated output pattern, based on the given input, matches the desired output.

The are several types of learning algorithms that can be used in a RBF network such as orthogonal least squares, generalized regression neural network (GRNN), K-means clustering, and probability density function. The emphasis of the paper is on adopting the GRNN routine in selecting a parsimonious RBF model. Specht (1991) has popularized "kernel regression" which he calls a GRNN. The GRNN algorithm is a kind of radial basis network that is often used for function approximation. The GRNN was introduced as a memory based neural network that would store all the independent and dependent training data available for a particular mapping (Heimes and Heuveln 1998). GRNN is particularly advantageous with sparse data in a real-time environment, because the regression surface is instantly defined everywhere, even with just one sample (Specht 1991). GRNN can be designed very quickly, has fast learning, and effectively uses historical data to estimate values for continuous dependent variables. The learning process is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data, with the criterion for the "best fit" being measured in some statistical sense. Using the features of learning and training processes, which it learned from past experience, or generalization of previous examples, RBF is capable of system modeling and forecasting.

The GRNN predicts the value of one or more dependent variables, given the value of one or more independent variables. According to Heimes and Heuveln (1998) and Specht (1991), the GRNN thus takes an input vector x of length n and generates an output vector (or scalar) y' of length m, where y'=prediction of the actual output y. The GRNN does this by comparing a new input pattern x with a set of p stored patterns x^i for which the actual output y_i is known. The predicted output y' is the weighted average of all these associated stored output y_{ij}. The following equation expresses how each predicted output component y'_j is a function of the corresponding output components y_j associated with each stored pattern x^i:

$$y'_j=\frac{N_j}{D}=\frac{\sum_{i=1}^{p}y_{ij}W(x,x^i)}{\sum_{i=1}^{p}W(x,x^i)},\quad j=1,2,\ldots,m \tag{7}$$

The weight $W(x,x^i)$ reflects the contribution of each known output y_i to the predicted output. It is a measure of the similarity of each pattern node with the input pattern. It is clear from Eq. (7) that the predicted output magnitude will always lie between the minimum and maximum magnitudes of the desired output, y_{ij} associated with the stored patterns (as $0\leqslant W\leqslant 1$). In the GRNN algorithm, the output weights are set to the desired outputs. The GRNN is best seen as an interpolator, which interpolates between the desired outputs of pattern layer nodes that are located near the input vector (or scalar) in the input space.

A standard way to define the similar function W, is to base it on a distance function, $D(x_1,x_2)$, that gives a measure of the distance or dissimilarity between two patterns x_1 and x_2. The desired property of the weight function $W(x,x^i)$ is that its magnitude for a stored pattern x^i be inversely proportional to its distance from the

Table 3. Calibration Coefficients of the Sungai Bekok Catchment

Model parameter	Calibrated value
Constant rate (mm/h)	3
Imperviousness (%)	48
Time of concentration (h)	18.25
Storage coefficient (h)	18
Recession constant	1
Threshold flow (m^3/s)	0.99

Table 4. Calibration Coefficients of the Sungai Ketil Catchment

Model parameter	Calibrated value
Constant rate (mm/h)	34
Imperviousness (%)	56
SCS lag (min)	2,093
Recession constant	1
Threshold flow (m^3/s)	0.992

Table 5. Results of RBF Model for the Sungai Bekok Catchment

Model data set	Model structure	Number of parameter	(R^2)	RMSE (m^3/s)	RRMSE	MAPE (%)
RBF TRAINING	7 input nodes	25	0.9942	0.0372	0.0074	0.7893
RBF-TEST Set 1	7 input nodes	25	0.9995	0.0039	0.0008	0.3151
RBF-TEST Set 2	7 input nodes	25	0.9994	0.0038	0.0008	0.3430
RBF-TEST Set 3	7 input nodes	25	0.9976	0.0031	0.0006	0.2307
RBF-TEST Set 4	7 input nodes	25	0.9998	0.0011	0.0002	0.1108
RBF-TEST Set 5	7 input nodes	25	0.9997	0.0004	0.0001	0.0369

input pattern x (if the distance is zero the weight is a maximum of unity). The standard distance and weight functions are given by the following equations, respectively:

$$W(x,x^i) = e^{-D(x,x^i)} \tag{8}$$

$$D(x_1,x_2) = \sum_{k=1}^{n} \left(\frac{x_{1k} - x_{2k}}{\sigma_k} \right)^2 \tag{9}$$

In Eq. (9), each input variable has its own sigma value, σ_k, where σ_k=normalization constant that controls the width of the basis function. The procedures of GRNN algorithm can be summarized as follows:

1. Input unit stores an input vector x.
2 The pattern units computes the distances $D(x,x^i)$ between the incoming patterns x and stored patterns x^i. The pattern nodes output the quantities $W(x,x^i)$.
3 The summation units compute N_j, the sums of the products of $W(x,x^i)$ and the associated known output component y_i. This unit also has a node to compute D, the sum of all $W(x,x^i)$.
4 Finally, the output unit divides N_j by D to produce the estimated output component y_j' that is a localized average of the stored output patterns.

Selection of the Number of Input Nodes

The determination of the appropriate number of input nodes or neurons in the input layer is important for the success of the radial basis network. The network performance and efficiency is sensitive to this number. There are no fixed rules about the number of nodes in the input layer. However, if this number is small, the network may not have sufficient degrees of freedom to learn the process correctly, and if the number is too high, the network will take a long time to get trained and may sometimes over fit the data. A trial and error procedure is generally applied in selecting the number of input nodes. An appropriate number of neurons can be found by calibrating the network and evaluating its performance by increasing the number of input neurons in order to obtain high efficiency with as few neurons as possible. The performance is evaluated by using mean square error (MSE). If the input layer has too many neurons, then there are too many parameters to be estimated. Meanwhile, the hidden layers enhance the network ability to model complex functions. When the width parameter of RBF is fixed and a set of centers is provided, the RBF network structure is specified.

Application of HEC-HMS Model

A hydrologic modeling system (HEC-HMS) program was developed by a team of engineers from the Hydrologic Engineering Center, U.S. Army Corps of Engineers, led by the Director, Darryl Davis (Hydrologic Engineering Center 2000). The HEC-HMS is designed to simulate the rainfall-runoff processes of watershed systems. It is designed to be applicable in a wide range of geographic areas for solving the widest possible range of problems. It utilizes a graphical user interface to build a watershed model and to set up the rainfall and control variables for simulation. The program features a completely integrated work environment including a database, data entry utilities, computation engine, and results reporting tools. The application of the HEC-HMS model involves two steps. First, the model was calibrated using previous data sets to determine the best parameters. Second, the model was verified by using new sets of data. HEC-HMS was run with the previous hourly rainfall-runoff data in order to predict runoff entering selected catchments. For the Sungai Bekok Catchment, the models use initial-constant infiltration/loss parametrization, the Clark hydrograph transformation routine, and a recession base flow component. Meanwhile, for the Sungai Ketil Catchment, the models use initial-constant infiltration/loss parametrization, the SCS hydrograph transformation routine, and a recession base flow component. The initial loss and initial flow are treated as initial conditions and vary from simulation to simulation. The hydrographs of the catchments are determined by using a trial and error method to get the best-suite parameters that can produce the best results. Calibration parameters for the HEC-HMS model for the Sungai Bekok and Sungai Ketil Catchments are shown in Tables 3 and 4, respectively.

Rainfall-Runoff Modeling

The steps involved in the identification of a nonlinear model of a system are: Selection of input-output data suitable for calibration and verification; selection of a model structure and estimation of its parameters; and validation of the identified models (Hsu et al.

Table 6. Results of HEC-HMS Model for the Sungai Bekok Catchment

Model data set	Number of parameter	(R^2)	RMSE (m^3/s)	RRMSE	MAPE (%)
HEC TRAINING	6	0.4867	0.7105	0.1367	30.8068
HEC-TEST Set 1	6	0.0209	0.2165	0.0427	116.6770
HEC-TEST Set 2	6	0.1480	0.3832	0.0488	62.5096
HEC-TEST Set 3	6	0.0440	0.4732	0.0905	88.5970
HEC-TEST Set 4	6	0.0798	0.2328	0.1312	42.9340
HEC-TEST Set 5	6	0.7069	0.2967	0.0084	22.5021

1993). The selection of training data that represents the characteristics of a watershed and meteorological patterns is extremely important in modeling (Yapo et al. 1996). The selection of good quality input-output pairs of data sets and an adequate set of training examples is very important in order to achieve good generalization properties. The set of all available data is separated in two disjoint sets: Training set and test set. The test set is not involved in the learning or training phase of the networks and it is used to evaluate the performances of the models. In this study, 20 events of flow hydrographs were randomly selected for calibrating the RBF model. Meanwhile, another 5 different data sets were used to verify or test the performance of the model. According to Specht (1991), it found that a GRNN is effective with only a few samples and even with sparse data in a multidimensional measurement space; the algorithm provides smooth transitions from one observed value to another.

In this particular study, the structure of an ANN model is designed based on a trial and error procedure to find the appropriate number of time-delayed input variables to the model. Dibike and Solomatine (1999) treat the rainfall as directly related to runoff at the present time t by using the following equation:

$$y(t)=f\{x(t),x(t-1),x(t-2),\ldots,x(t-n),y(t-1),\ldots,y(t-n)\} \tag{10}$$

This model treats the runoff as directly related to rainfall at the present time t. The goodness-of-fit statistics are computed for both training and testing for each ANN architecture. At the first step, the rainfall at time t was added to the model. The goodness-of-fit statistics for the present model were computed for training and testing procedures. Then rainfall at time $(t-1)$ was added as an additional input variable to the model, and the goodness-of-fit statistics were computed. This procedure is repeated by adding rainfall at previous time periods as input variable until there is no significant change in model training and testing accuracy. After the first step was completed, another input variable; the runoff at previous time periods, $(t-1)$ is added to the best-fit model obtained from the first step. Then, the goodness-of-fit statistics for the present model were computed for training and testing procedures. This procedure is repeated by adding runoff at previous time periods as input variable until there is no significant change in model training and testing accuracy. The optimal number of input nodes is important in the neural network and it is determined as follows:

1 Sungai Bekok Catchment:

$$y(t)=f\{x(t),x(t-1),x(t-2),x(t-3),x(t-4),x(t-5),y(t-1)\} \tag{11}$$

2 Sungai Ketil Catchment:

$$y(t)=f\{x(t),x(t-1),x(t-2),x(t-3),x(t-4),y(t-1)\} \tag{12}$$

The value of $y(t-1)$ in Eqs. (11) and (12) probably can repre-

Table 7. Results of RBF Model for the Sungai Ketil Catchment

Model data set	Model structure	Number of parameter	(R^2)	RMSE (m^3/s)	RRMSE	MAPE (%)
RBF TRAINING	6 input nodes	22	0.9970	0.0249	0.0008	0.1749
RBF-TEST Set 1	6 input nodes	22	0.9987	0.0112	0.0004	0.1064
RBF-TEST Set 2	6 input nodes	22	0.9982	0.0150	0.0005	0.1608
RBF-TEST Set 3	6 input nodes	22	0.9965	0.0215	0.0007	0.3010
RBF-TEST Set 4	6 input nodes	22	0.9972	0.0210	0.0007	0.2574
RBF-TEST Set 5	6 input nodes	22	0.9968	0.0387	0.0013	0.3535

Table 8. Results of HEC-HMS Model for the Sungai Ketil Catchment

Model data set	Number of parameter	(R^2)	RMSE (m^3/s)	RRMSE	MAPE (%)
HEC TRAINING	5	0.1001	0.9106	0.0300	26.6170
HEC-TEST Set 1	5	0.4194	0.2831	0.0090	60.5143
HEC-TEST Set 2	5	0.2104	0.2147	0.0122	62.5150
HEC-TEST Set 3	5	0.6183	0.1742	0.0135	49.8880
HEC-TEST Set 4	5	0.3647	0.3924	0.0129	113.1124
HEC-TEST Set 5	5	0.5767	0.2959	0.0242	77.1620

sent the conditions of the soil moisture content and water table that contributes to the current flow or runoff.

Model Performance Criteria

The RBF model is designed to simulate the rainfall-runoff processes of watersheds systems. Because there was no definitive test to evaluate the success of each model, a multicriteria assessment was carried out. Basically, the performance of model was evaluated based on the comparison between the computed output and actual data. The prediction of each model is evaluated using the correlation of coefficient (R^2), root mean square error (RMSE), relative root mean square error (RRMSE), and mean absolute percentage error (MAPE). The RMSE is one of the most commonly used performance measures in hydrological modeling. The others are to try to fill some of the gaps left by considering only RMSE, and because RRMSE and MAPE provide different types of information about model prediction capabilities. Formulas for calculating R^2, RMSE, RRMSE, and MAPE are given as follows:

$$R^2 = \frac{\sum_{t=1}^{n}[(Q_{o(t)} - \bar{Q}_{o(t)})(Q_{s(t)} - \bar{Q}_{s(t)})]}{\left[\sum_{t=1}^{n}(Q_{o(t)} - \bar{Q}_{o(t)})^2 \sum_{t=1}^{n}(Q_{s(t)} - \bar{Q}_{s(t)})^2\right]^{1/2}} \quad (13)$$

$$\text{RMSE} = \left[\frac{1}{n}\sum_{t=1}^{n}(Q_{o(t)} - Q_{s(t)})^2\right]^{1/2} \quad (14)$$

$$\text{RRMSE} = \left[\frac{1}{n}\sum_{t=1}^{n}\left[\frac{(Q_{o(t)} - Q_{s(t)})}{Q_{o(t)}}\right]^2\right]^{1/2} \quad (15)$$

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Q_{o(t)-Q_{s(t)}}}{Q_{o(t)}}\right| \times 100\% \quad (16)$$

where $Q_{o(t)}$ and $Q_{s(t)}$=observed and simulated values of output and n=number of observations or time periods over which the errors are simulated. The value of R^2 of 90% indicates a very satisfactory model performance whereas a value in the range 80–90% indicates a fairly good model. Values of R^2 in the range 60–80% would indicate an unsatisfactory model fit (Kachroo 1986). Generally, RMSE and RRMSE formulas evaluate the models based on a comparison of the estimated errors between the actual observations and the fitted model. A model with the minimum error is considered the best choice. Johnson and King (1988) stated that the MAPE around 30% is considered a reasonable prediction. Further, the analysis will be considered very accurate when the MAPE is in the range of 5–10%.

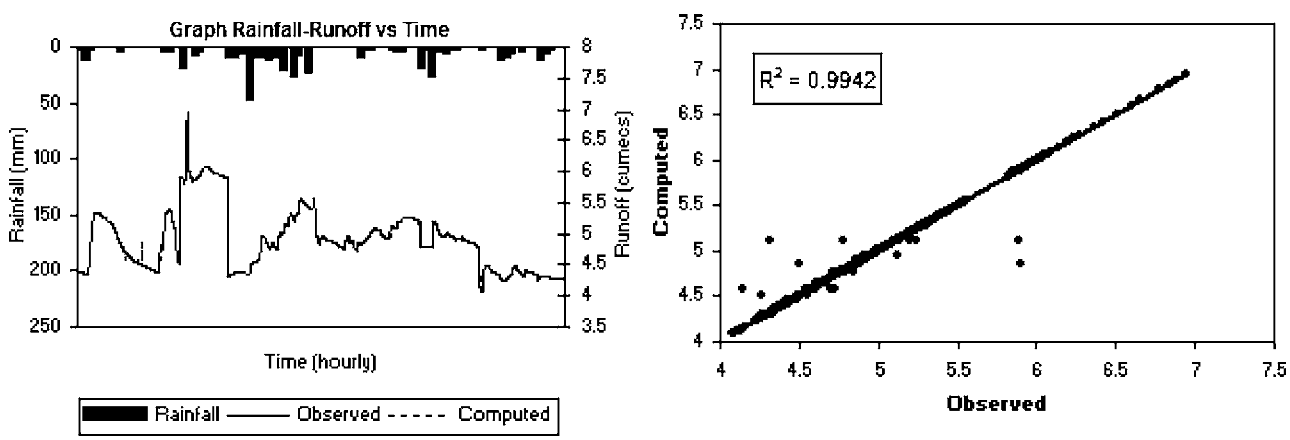

Fig. 3. Result of the Sungai Bekok Catchment area (training phase)

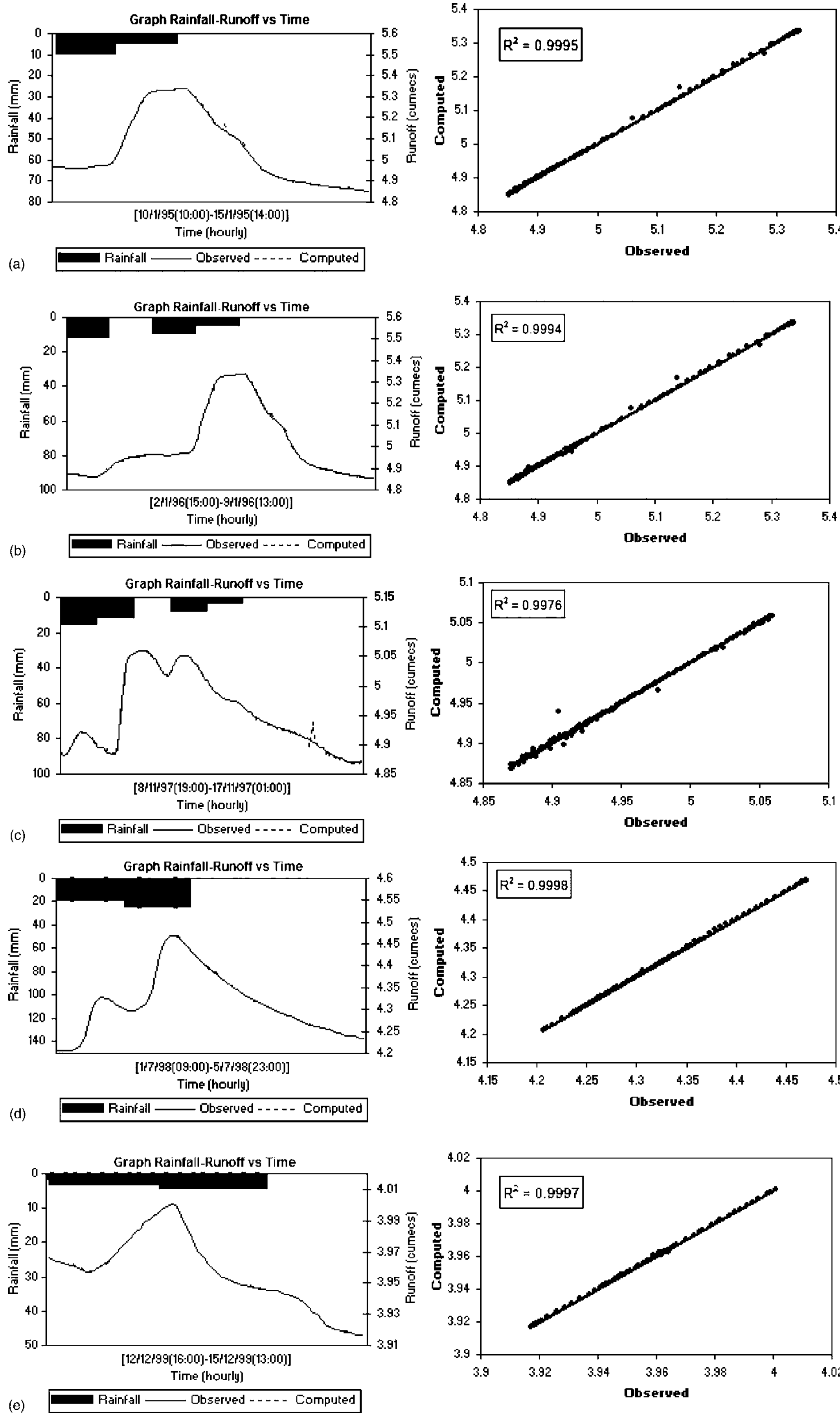

Fig. 4. (a)–(e) Result of the Sungai Bekok Catchment area (testing phase)

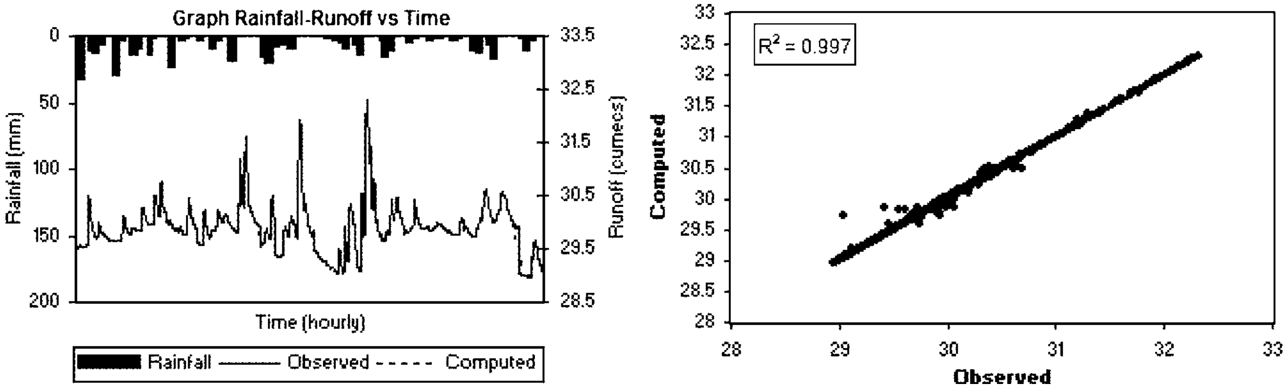

Fig. 5. Result of the Sungai Ketil Catchment area (training phase)

Results and Discussion

Results of RBF and HEC-HMS modeling for Sungai Bekok and Sungai Ketil catchments are presented in Tables 5–8, respectively. The measures of performance of each model are indicated by R^2, RMSE, RRMSE, and MAPE. Figs. 3, 4(a–e), 5, and 6(a–e), illustrate the graphical results of RBF training and testing for the Sungai Bekok and Sungai Ketil Catchments.

Calibration has been carried out based on GRNN algorithm. To calibrate and evaluate the performance of the model, only a few samples of data have been used. According to Specht (1991), the main advantage of the GRNN algorithm is that it can generalize from a few examples of data (even with just one sample), as soon as they are stored. There are, two selected data sets used for training and five selected sets of data used as the prediction set. For the RBF training process, the best-input nodes are chosen based on the minimum RMSE computed for the training data. The numbers of input nodes considered for RBF are 7 nodes for Sungai Bekok, and 6 nodes for Sungai Ketil Catchments. Meanwhile, the numbers of parameters (or weights) considered for RBF networks are 25 and 22 for the Sungai Bekok and Sungai Ketil Catchments, respectively. Obviously, the RBF model learns faster than HEC-HMS model, and the model can produce results rapidly in the testing phase, even when the number of samples becomes very large.

Most values of R^2 approach 1.0. This outcome indicates that the RBF model consistently show a good performance in rainfall-runoff modelling. Kachroo (1986) reported that when R^2 is more than 90%, the model is very satisfactory. It is fairly good with R^2 in the range of 80–90%. Johnson and King (1988) decided on model accuracy based on a MAPE value. The prediction model considered reasonable with a MAPE below 30% and very accurate with a MAPE less than 10%. Results of modeling for Sungai Bekok and Sungai Ketil with a MAPE less than 10% can be considered as very accurate. Meanwhile, the RBF model gives R^2 to be more than 90% and this condition shows that the model performance is very satisfactory. The results of HEC-HMS model are presented in Tables 6 and 8 for the Sungai Bekok and Sungai Ketil Catchments, respectively. The HEC-HMS yield a R^2 below 70%, and this condition shows poor performance and is unsatisfactory. The results of MAPE for Sungai Ketil are more than 30%. It is considered an unreasonable prediction. Meanwhile, results of modeling for Sungai Bekok with MAPE are less than 30% are considered as a reasonable prediction.

During the training phase, the RMSE for Sungai Bekok is consistently less than 0.1 m^3/s and the RRMSE is also maintained below 0.02. During testing, the RMSE is less than 0.004 m^3/s and RRMSE is less than 0.001 and this comes close to zero. Meanwhile, the RMSE for Sungai Ketil is consistently less than 0.03 m^3/s and the RRMSE is also maintained below 0.001. During testing, the RMSE is less than 0.04 m^3/s and RRMSE is less than 0.015 and this comes close to zero. Results of modeling for Sungai Ketil with a MAPE less than 10% can be considered as very accurate. Obviously, the application of the RBF method to the model rainfall-runoff relationship of Sungai Bekok and Sungai Ketil is better than the HEC-HMS model in the training and testing phase.

Meanwhile, the RRMSE of RBF models for Sungai Ketil and Sungai Bekok are close to zero, respectively. The HEC-HMS model gives a higher error than the RBF models with a worse degree of efficiency; in term of RMSE and RRMSE. The RBF network shows a slightly better performance both in the training and testing periods. Apparently, the RBF model shows a better performance than the HEC-HMS model with good agreement of the RMSE and RRMSE results in the training and testing phase, revealing best fitting to the data.

Normally, for a small catchment size, the river flow is highly nonlinear and influenced by storage effect, which can affect the quality of the data. Meanwhile, the nonlinearity of river flow for a big catchment, such as the Sungai Ketil Catchment is more consistent. In addition, the effect of spatial rainfall and control structures may contribute to the complexity of the system. However, the RBF method has shown that it can easily handle the existence of nonlinearity processes within the catchment.

In general, the RBF network can be described as a universal approximated function using combinations of basis functions centered around weight vectors to provide spatial estimates. The best results were achieved for the network with a Gaussian activation function, GRNN algorithms, and an appropriated number of input nodes. If the architecture of the training algorithms is not suitable, it will affect the accuracy of predictions and a network's learning ability. The number of input nodes significantly influences the performance of a network and the time taken to train the model. It is related to the complexity of the system being modeled and to the resolution of the data fit. The number of input nodes in the input layer was determined by trial and error for each case. If this number of input nodes is small, the network can suffer from under fit of the data and may not achieve the desired level of accuracy, whereas with too many nodes it will take a long time to be adequately trained and may, sometimes, over fit the data.

Robustness test on five different data sets (refer to Tables 5 and 7) can reveal the consistency of the model. Each data set

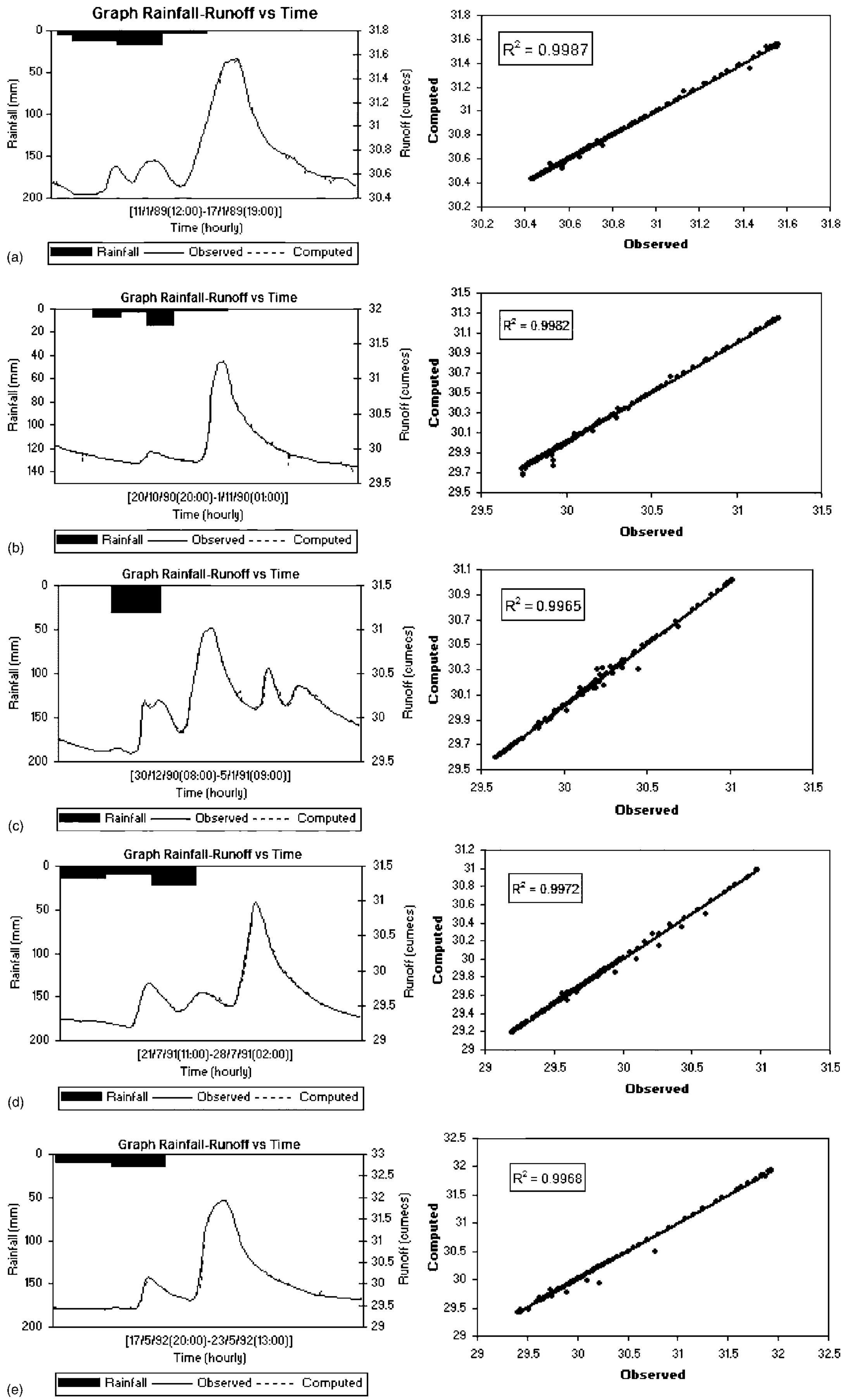

Fig. 6. (a)–(e) Result of the Sungai Ketil Catchment area (testing phase)

consists of different flow hydrographs. The robustness test is carried out to evaluate the performance of the model using different input-output data sets. Certainly, a robust model will consistently yield the lowest RMSE, RRMSE, and MAPE errors. The RBF network has been proven to be a robust model in modeling the rainfall-runoff relationship.

Conclusion

The study demonstrates that the neural network model based on RBF is suitable for modeling the rainfall-runoff relationship compared to the HEC-HMS model. By considering a good training process and suitable algorithms and nodes, the prediction is more accurate. The GRNN algorithms are "fast learners" and a RBF network could predict runoff accurately with good agreement between the observed and predicted values.

The ANN models have been identified as a robust and flexible model in modeling the rainfall-runoff relationship. Once the architecture of the network is defined, weights are calculated so as to represent the desired output through a learning process where the ANN is trained to obtain the expected results. The neural network could predict runoff accurately, with good agreement between the observed and predicted values compared to the HEC-HMS model.

A new approach such as an ANN model provides an accurate and sensible prediction that will benefit in the decision-making related to the hydrology and water resources problem. Obviously, an ANN application to model the hourly streamflow hydrograph was successful.

References

Broomhead, D. S., and Lowe, D. (1988). "Multivariate functional interpolation and adaptive networks." *Complex Syst.*, 2, 302–355.

Caudill, M. (1987). "Neural networks primer, Part I." *AI Expert*, December.

Chen, S., Cowan, C. F. N., and Grant, P. M. (1991). "Orthogonal least squares learning for radial basis function networks." *IEEE Trans. Neural Netw.*, 2(2), 302–309.

Dibike, Y. B., and Solomatine, D. P. (1999). "River flow forecasting using artificial neural networks." *J. Phys. Chem. of the Earth*, EGS1.1–0002.

Fausett, L. (1994). *Fundamentals of neural networks*, Prentice Hall, Englewood Cliffs, N.J.

Heimes, F., and Heuveln, B. V. (1998). "The normalized radial basis function neural network." *IEEE Trans. Neural Netw.*, 1, 1609–1614.

Hsu, K.-L., Gupta, H. V., and Sorooshian, S. (1993). "Artificial neural network modelling of the rainfall Runoff Process." *Water Resour. Res.*, 29(4).

Hydrologic Engineering Center. (2000). *Hydrologic modelling system HEC-HMS user's manual* U.S. Army Corps of Engineers, Vicksburg, Miss.

Johnson, D., and King, M. (1988). *Basic forecasting techniques*, Butterworth, London.

Kachroo, R. K. (1986). *HOMS Workshop on River Flow Forecasting, Nanjing, China*, Unpublished Internal Report, Dept. of Engineering Hydraulics, Univ. College Galway, Galway, Ireland.

Karunanithi, N., Grenney, W. J., Whitley, D., and Bovee, K. (1994). "Neural networks for river flow prediction." *J. Comput. Civ. Eng.*, 8(2),201–220.

Lu, B., and Evans, B. L. (1999). "Channel equalization by feedforward neural networks." *IEEE Trans. Neural Netw.*, 10, 587–590.

Lucks, M. B., and Oki, N. (1999). "A radial basis function for function approximation." *IEEE Trans. Neural Netw.*, 5, 1099–1101.

Mann, I., and McLaughlin, S. (2000). "Dynamical system modelling using radial basis function." *IEEE Trans. Neural Netw.*, 7, 461–465.

Poggio, T., and Girosi, F. (1990). "Networks for approximation and learning." *Proc. IEEE*, 78, pp. 1481–1497.

Specht, D. F. (1991). "A general regression neural network." *IEEE Trans. Neural Netw.*, 2, 568–576.

Yang, S., and Tseng, C. (1988). "An orthogonal neural network for function approximation." *IEEE Trans. Syst. Man Cybern.*, 265, 779–785.

Yapo, P. O., Gupta, V. H., and Sorooshian, S. (1996). "Automatic calibration of conceptual rainfall runoff models: Sensitivity to calibration data." *J. Hydrol.*, 181, 23–48.