River Flow Forecasting: a Hybrid Model of Self Organizing Maps and Least Square Support Vector Machine

S. Ismail\textsuperscript{1}, R. Samsudin\textsuperscript{2}, and A. Shabri\textsuperscript{1}

\textsuperscript{1}Department of Mathematic, Science Faculty, Universiti Technologi Malaysia, Malaysia
\textsuperscript{2}Department of Software Engineering, Faculty of Computer Science and Information System, Universiti Technologi Malaysia, Malaysia

Received: 18 May 2010 – Accepted: 14 July 2010 – Published: 18 October 2010

Correspondence to: S. Ismail (ismail.shuhaida@gmail.com)

Published by Copernicus Publications on behalf of the European Geosciences Union.
Abstract

Successful river flow time series forecasting is a major goal and an essential procedure that is necessary in water resources planning and management. This study introduced a new hybrid model based on a combination of two familiar non-linear method of mathematical modeling: Self Organizing Map (SOM) and Least Square Support Vector Machine (LSSVM) model referred as SOM-LSSVM model. The hybrid model uses the SOM algorithm to cluster the training data into several disjointed clusters and the individual LSSVM is used to forecast the river flow. The feasibility of this proposed model is evaluated to actual river flow data from Bernam River located in Selangor, Malaysia. Their results have been compared to those obtained using LSSVM and artificial neural networks (ANN) models. The experiment results show that the SOM-LSSVM model outperforms other models for forecasting river flow. It also indicates that the proposed model can forecast more precisely and provides a promising alternative technique in river flow forecasting.

1 Introduction

Bernam River is located between the Malaysian states of Perak and Selangor, demarcating the border of the two states. Bernam River flow from Mount Liang Timur in the east on the Titiwangsa Mountains to the Straits of Malacca in the west. The peak of Mount Liang Timur itself marks the point where Pahang, Perak and Selangor meet. The eastern part of the river is suitable for palm oil and rubber tree plantation while swamps fill the western areas. A percentage of the swampy areas have been reclaimed and dried up by a drainage system. Some has been converted into paddy fields. So that, the forecasting of a river flow would be such an important information needed for those working with it such as a plantation company, a farmer and etc.

Nowadays, river flow forecasting is an active research area that have been studied. The flow 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 water supply or irrigation. In the a few decades, with the development of software technology, there have been many approaches affiliated to the technique used including data-driven models. Among the various types of data-driven models, an Artificial neural network (ANN) is the most widely used for time series forecasting and has been successfully employed in modeling a wide range of hydrologic contexts (Maier and Dandy, 2000; Dibike and Solomatine, 2001; Bowden et al., 2005; Dolling and Varas, 2003; Muhamad and Hassan, 2005; Kisi, 2008; Wang et al., 2009; Keskin and Taylan, 2009; Luk et al., 2000; Hung et al., 2009; Affandi and Watanabe, 2007; Birkinshaw et al., 2008; Corzo et al., 2009). However, there are some disadvantages of the ANN. The network structure is hard to determine and it is usually determined by using a trial-and-error approach (Kisi, 2004).

Recently, the support vector machine (SVM) method, which was suggested by Vapnik (1995), has used in a range of applicants, including hydrological modeling and water resources process (Asefa et al., 2006; Yu et al., 2006) and etc. Several studies have been carried out using SVM in hydrological modelling such as stream flow forecasting (Wang et al., 2009, Asefa et al., 2006; Lin et al., 2006), rainfall runoff modeling (Dibike et al., 2001; Elshorbagy et al., 2009a, b) and flood stage forecasting (Yu et al., 2006). However, the standard SVM is solved using complicated quadratic programming methods, which are often time consuming and has higher computational burden because of the required constrained optimization programming.

As a simplification of SVM, Suykens and Vandewalle (1999) have proposed the use of the least squares support vector machines (LSSVM). LSSVM has been used successfully in various areas of pattern recognition and regression problems (Hanbay, 2009; Kang et al., 2008). LSSVM encompasses similar advantages as SVM, but its additional advantages is that it requires solving a set of only linear equations, which is much easier and computationally more simple. The method uses equality constraints instead of inequality constraints and adopts the least squares linear system as its loss function, which is computationally attractive. LSSVM also has good convergence and
high precision. Hence, this method is easier to use than quadratic programming solvers in SVM method. Extensive empirical studies (Wang and Hu, 2005) have shown that LSSVM is comparable to SVM in terms of generalization performance. The major advantage of LSSVM is that it is computationally very cheap while it still possesses some important properties of the SVM. In the water resource, the LSSVM method has received very little attention literature and only a few applications of LSSVM to modeling of environmental and ecological systems such as water quality prediction (Yunrong and Liangzhong, 2009).

The Self Organizing Map (SOM) proposed by Kohonen (2001) is one category of ANN was first used as an information-processing tool in the fields of speech and image recognition. The SOM has been successfully applied for clustering, classification, estimation, prediction and data mining (Vesanto and Alhoniemi, 2000; Kohonen, 2001). The SOM has found increasing interest in water resources application such as classification of satellite imagery data and rainfall estimation (Murao et al., 1993), rainfall-rounoff modeling (Hsu et al., 2002), typhoon-rainfall forecasting (Lin and Wu, 2009), river flood forecasting (Chang, 2007) and water resource problems (Kalteh et al., 2008). SOM is an excellent method to cluster data according to their similarity. The superior performance of SOM compared with the other clustering methods has been discussed in the literature (Chen et al., 2005; Manigiameli et al., 1996; Lin and Chen, 2006).

Improving forecasting especially time series forecasting accuracy is an important yet often difficult task facing decision-makers in many areas. Using hybrid models has become a common practice to improve the forecasting accuracy. There have been several studies showed hybrid models can be an effective way to improving predictions achieved by either of the models used separately(Zhang, 2003; Jain and Kumar, 2007; Chen and Wang, 2007; Pai and Lin, 2005). In recent years, more hybrid forecasting model which combining cluster techniques with forecasting model has successfully solved many predictions problems such as SOM with ANN (Pal et al., 2003; Lin and Wu, 2009; Wang and Yan, 2004), SVM (Cao, 2003; Fan and Chen, 2006; Fan et al.,
2007; Huang and Tsai, 2009), Radial basis function (Lin and Chen, 2005) and other models (Chang and Liao, 2006; Chang et al., 2007, 2008).

In this paper, a new hybrid model which combines the SOM with the LSSVM (SOM-LSSVM) models is proposed in order to improve the accuracy of river flow forecasting. With the advantages of the data analysis technique developed by SOM and capability of LSSVM, the proposed hybrid model is expected to be useful for river flow forecasting. The prediction results by SOM-LSSVM model is compared with a forecasting model developed by the conventional ARIMA, ANN and LSSVM models. To verify the application of this approach, the river flow data from Bernam River located in Selangor, Malaysia is chosen as the case study.

2 Forecasting models

In this section represent the ANN, LSSVM and SOM-LSSVM models used for river flow forecasting. The choice of these models in this study was because these methods have been widely and successfully used in time series forecasting.

2.1 Artificial Neural Network

Artificial neural network is a model based on multi-layer perceptron, commonly referred to as feedforward network. The most commonly used ANN in water resources and hydrology is the feedforward multi layer perceptron (MLP) consists of three layers: an input layer, a hidden layer and an output layer. Each neuron has a number of inputs from outside the network or the previous layer and a number of outputs that leads to the subsequent layer of the network. A neuron computes the output response based on the weighted sum of all its inputs according to an activation function.
Mathematically, a three-layer MLP with $p$ input nodes, $q$ hidden nodes and one output node can be expressed as

$$y_t = g \left( \sum_{j=1}^{q} w_j f \left( \sum_{i=1}^{p} w_i x_{t-i} \right) \right)$$  \hspace{1cm} (1)$$

where $y_t$ is the output layer, $x_{t-i}$ is the input of the network, $w_i$ is the connection weights between nodes of input and hidden layers, $w_j$ is the connection weights between nodes of hidden and output layers, $g$ and $f$ are activation functions. The most common of $f$ is the sigmoid function and $g$ is the linear function are adopted here. Back-propagation is the most popular algorithm for training feed-forward MLP. For detailed reviews of ANN along with their application in water resources and hydrology can refer to Maier and Dandy (2000).

### 2.2 Least Square Support Vector Machine

Least Square Support Vector Machine (LSSVM) is a modification of the standard Support Vector Machine (SVM) was develop by Suykens and Vandewalle (Suykens, 2005). The basic LS-SVM is used for the optimal control of non-linear Karush-Kuhn-Tucker systems for classification as well as regression.

Consider a set data $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, $x_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, $x$ is the input vector, $y$ is the expected output and $n$ is the number of data. The LSSVM has been developed to find the optimally non-linear regression function

$$y(x) = w^T \phi(x) + b$$  \hspace{1cm} (2)$$

By combining the functional complexity and fitting error, the optimization problem of LSSVM is given as:

$$J(w, \xi) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{n} \xi_i^2$$  \hspace{1cm} (3)$$
Such that:
\[ y(x) = w^T \varphi(x_i) + b + \xi_i \quad i = 1, 2, 3, \ldots, n \]  \hspace{1cm} (4)

This formulation consists of equality instead of inequality constraints. To solve this optimization problem, Lagrange function is constructed as
\[ L(w, b, \xi; \alpha) = J(w, b, \xi) - \sum_{i=1}^{n} \alpha_i \{ w^T \varphi(x_i) + b - y_i + \xi_i \} \]  \hspace{1cm} (5)

where \( \alpha_i \) are the Langrange multipliers, which can be positive or negative. The solution of Eq. (5) can be obtained by partially differentiating with respect to \( w, b, \xi_i \) and \( \alpha_i \)
\[
\begin{align*}
\frac{\partial L}{\partial w} &= 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i \varphi(x_i) \\
\frac{\partial L}{\partial b} &= 0 \rightarrow \sum_{i=1}^{n} \alpha_i = 0 \\
\frac{\partial L}{\partial \xi_i} &= 0 \rightarrow \alpha_i = \gamma \xi_i \\
\frac{\partial L}{\partial \alpha_i} &= 0 \rightarrow w^T \varphi(x_i) + b - y_i + \xi_i = 0
\end{align*}
\]  \hspace{1cm} (6)

After elimination of the variables \( w \) and \( \xi_i \) one obtains the following matrix solution.
\[
\begin{bmatrix}
0 & 1^T_v \\
1^T_v & \Omega + \frac{1}{\gamma} I
\end{bmatrix}
\begin{bmatrix}
b \\
\alpha
\end{bmatrix} =
\begin{bmatrix}
0 \\
y
\end{bmatrix}
\]  \hspace{1cm} (7)

with \( y = [y_1, y_2, \ldots, y_l], 1^T_v = [1, 1, \ldots, 1], \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_l] \) and Mercer’s condition is applied within the \( \Omega \) matrix;
\[ \Omega_{ij} = y_i y_j \varphi(x_i)^T \varphi(x_j) = y_i y_j K(x_i, x_j) \]  \hspace{1cm} (8)

The fitting function namely the output of LSSVM Regression is,
\[ y(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) + b \]  \hspace{1cm} (9)
For a point $x_j$ to be evaluated it is:

$$y(x_j) = \sum_{i=1}^{n} \alpha_i K(x_i, x_j) + b$$

where $\alpha_i, b$ are the solutions to the linear system and $K(x_i, x_j)$ is a kernel function. The most popular kernel function is Radial Basis Function (RBF), as shown in Eq. (10) (Liu and Wang, 2008; Gencoglu and Ulyar, 2009).

$$K(x_i, x_j) = \exp\left(-\frac{\|x - x_j\|^2}{\delta^2}\right)$$

### 2.3 Self Organizing Map and Clustering

An clustering can be considered the most important unsupervised learning problem. A cluster is therefore a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. In this paper, we present a Self Organizing Map (SOM) as a clustering technique. The SOM which is also known as Self Organizing Feature Map (SOFM) proposed by Professor Teuvo Kohonen, and is sometimes called as Kohonen Map (Kohonen, 2001) is an unsupervised and competitive learning algorithm. SOM has been used widely for data analysis in some areas such as economics, physics, chemistry as well as medical applications.

SOM can be used as clustering tools since its convert the non linear statistical relationship between high dimensional data into simple geometric relationships of their image points on a low-dimensional display. By that way, the data points which showed similar properties are placed close to each other within the output of SOM algorithm (Budayan et al., 2009).

The objective of SOM is to maximize the degree of similarity of patterns within a cluster, minimize the similarity of patterns belonging to different clusters, and then present the results in a lower-dimensional space. Basically, the SOM consists of two layer of...
artificial neurons: the input layer, which accepts the external input signals, and the output layer (also called the output map), which is usually arranged in a two-dimensional structure. Every input neuron is connected to every output neuron, and each connection has a weighting value attached to it. The architecture of SOM are shown in Fig. 1.

Output neurons will self organize to an ordered map and neurons with similar weights are placed together. They are connected to adjacent neurons by a neighborhood relation, dictating the topology of the map (Moreno, 2006). The concept of learning algorithm for SOM is unsupervised and competitive. The training process of SOM is described below:

For simplicit, we assume that the input vector \( \mathbf{X} \) of SOM by

\[
\mathbf{X} = [x_1, x_2, \ldots, x_n]
\]

where \( n \) is the dimension of the input vector. The weight vector connecting input vector to the hidden neuron \( i \) is denoted by

\[
\mathbf{W}_i = [w_{i1}, w_{i2}, \ldots, w_{in}] \quad i = 1, 2, \ldots, m
\]

The weight are initialized as small random numbers at beginning of the training process. In competitive learning networks, the neurons compete among themselves to determine a winner by calculating the distance between the input vector and the weight vectors of all neurons in the hidden layer. The winner \( I \) is defined as the one whose weight vector is the closet to the input vector \( \mathbf{X} \), i.e.

\[
I(\mathbf{X}) = \min_{\mathbf{W}_i} \| \mathbf{X} - \mathbf{W}_i \| \quad i = 1, 2, \ldots, m
\]

The Euclidean distance is often used as the similarity measure for SOM. The output neuron whose weight vector has the smallest distance from the input vector is called the winning neuron.

After determining the winning neuron, the lateral intersections between the winning neuron and its neighbourhood are calculated using the topological neighbourhood
function. The neighborhood function takes the form of a Radial Basis function that is appropriate for representing the biological lateral interaction (Kohonen, 2001; Rui Xu, 2009)

\[ h_{ji}(t) = \eta(t) \exp \left( -\frac{||r_j - r_i||^2}{2\sigma^2(t)} \right) \]  

(14)

Where \( ||r_j - r_i|| \) represent Euclidean distance between the winning neuron \( i \) and the neighbouring neuron \( j \) and \( \sigma(t) \) is the bandwidth of the radial basis kernel function.

Next, the weights of this winning neuron are adjusted according to the input patterns using the algorithm defined as

\[ W_i(t + 1) = W_i(t) + h_{ji}(t)(X - W_i(t)) \]  

(15)

where \( \eta(t) \) is the learning rate at time \( t \) and \( W_i(t + 1) \) is the adjusted weight vector at time \( t + 1 \).

After the weights have been updated, the winning neuron and the neighbourhood neurons become more similar to the corresponding input pattern. The processes continues until convergence has been reach. Finally the trained SOM can be obtained. More details regarding the learning algorithm of SOM are given by Kohonen (2001).

### 2.4 Integrating the SOM-LSSVM model

In this study, a hybrid model was implemented which combines the SOM clustering algorithm with the LSSVM model, as illustrated in Fig. 2. In the first stage, the datasets are divided into several groups or clusters. In order to do this, SOM is used to cluster the training data into several disjointed clusters. Each cluster contains similar objects (Huang and Tsai, 2009). After the clustering process, an individual LSSVM model for each cluster is constructed. LSSVM can do a better forecast for each group or cluster. After running an individual LSSVM for each cluster, the result will be combined in order to get the final result. A proposed method are needed to carry out in order to discover a better results.
3 The study area and data

In this research, we examined the data gather from monthly river flow of Bernam River located in Selangor, Peninsular Malaysia. The upper Bernam River basin had been identified as the ultimate and largest source of water supply for Bernam River especially for irrigation. The study area is about 1090 km$^2$ with Bernam River monitoring station as the downstream outlet. The location of Bernam river catchment is shown in Fig. 3. The data was collected from January 1966 to December 2005. The mean monthly river flow of Bernam River were selected for this study.

Essentially, each data set is divided into two parts; training and testing data. Training data is used exclusively for model development and testing data is used to measure the performance of the model on untrained data. Solomatine et al. (2008) suggested that in splitting of data into training and testing data sets, these sets should have similar distribution of low and high flow or similar properties of the input and output variables. However, they found to make the generalization of the training and testing sets with the similar properties are not an easy task. Most studies suggested that the ratio of splitting for training and testing as 70%:30%, 80%: 20% and 90%:10%, respectively. The selection of the ratio could be based on the the particular problems (Zhang et al., 1998; Firat, 2007; Kisi, 2008, Wang et al., 2009).

In this study, 88% of the data sets consist of 456 monthly records from January 1966 to December 2003 is used for training whilst 12% of data sets containing of 60 mean monthly river flow, recorded from January 2004 to December 2008 is used for testing. The training set is used for model selection and parameter optimization, being the testing set used to compare the proposed approach with other models. Before the training process begins, data normalization is often performed. The river flow was normalized in the range [0, 1] by the following equation:

$$x_t = 0.1 + \frac{x_t'}{1.2\max(x_t')}$$ (16)
where $x'_t$ and $x_t$ represent the original data and normalized while $\max (x'_t)$ represent the maximum values among the original data.

4 Input determination

As in any data-driven models such as ANN and LSSVM models, the selection of appropriate model inputs would play an extremely important role in their successful implementations since it provides the basic information about the system being modeled. In time series forecasting, very little attention is given to the task of selecting appropriate model inputs. Many papers reviewed failed to describe the input determination methodology used, and consequently, raised doubts about optimality of the input obtained (Bowden et al., 2005). Most researchers design experiments to help select the model inputs while others adopt some empirical ideas. For example, Patil (1992) proposed model inputs based on 12 inputs for monthly data and four for quarterly data heuristically. Cheung et al. (1996) suggested the maximum entropy principles to identify the time series lag structure. Tang and Fishwick (1993) claimed that the number of model inputs is simply the number of autoregressive (AR) in the Box-Jenkins models. Refenes et al. (1997) suggested a stepwise method for determining input for ANN models. Roadknight et al. (1997) used cross correlation analysis or principal component analysis as a guide for determining the input. Aqil et al. (2006) employed two statistical methods, i.e. autocorrelation (ACF) and partial autocorrelation (PACF) to identify the appropriate input variables. Behzad et al. (2009) selected the best model inputs by trial and error according to minimum test error in the ANN and SVM modeling. Corzo et al. (2009) used the correlation and average mutual information analysis involving different sub-basins uses precipitation and river flow to determine the best input variables. Khashei and Bijari (2010) proposed ARIMA model to determining the input variables in order to yield a more accurate forecasting model than ANN. The empirical results with three well-known real data sets showed that the proposed input variables can be an effective way to improve forecasting accuracy achieved by ANN. The use of input
variables from the data values of previous time and optimum number of input variables were determined by trial and error has been reported by Firat (2007), Firat and Gun- gor (2007), Firat (2008), Sivapragasam and Liong (2005), Juhos et al. (2008), Partal and Kisi (2007) and among others.

In this study, three approaches were used to determine the model inputs. The first approach, six model inputs were chosen based on the past river flow. The appropriate lags were chosen by using a trial-and-error approach ($x_{t-1}, x_{t-2}, \ldots, x_{t-p}$ where $p$ is 2, 4, \ldots, 12). The second and third approach are by setting the inputs vector nodes equal to the number of lagged variables from two statistical methods (i.e. stepwise multiple regression analysis and ARIMA model). Stepwise multiple regression analysis led to the selection of 7 input attributes ($x_{t-1}, x_{t-2}, x_{t-4}, x_{t-5}, x_{t-7}, x_{t-10}$ and $x_{t-12}$).

In the literature, extensive applications and reviews of ARIMA models proposed for modeling of water resources time series were reported (Yurekli et al., 2004; Muhamad and Hassan, 2005; Huang et al., 2004; Modarres, 2007; Fernandez and Vega, 2009; Wang et al., 2009). In the identification step, the sample autocorrelation function (ACF) and sample partial autocorrelation function (PACF) is used to see whether the series is stationary or not, and to chose a tentative model. The ACF and PACF for the series are plotted in Fig. 4. The ACFs curve for monthly streamflow data decayed with mixture of sine wave pattern and exponential curve that reflects the random periodicity of the data and indicates the need for seasonal MA terms in the model. In the PACF there were significant spikes present near lag 12 and 24, and therefore the series need for seasonal AR process. The criterions to judge for the best model based on Akaike’s Information Criterion (AIC) show that the ARIMA $(2,0,0) \times (2,0,2)_{12}$ is a relatively best model. Since the ARIMA $(2,0,0) \times (2,0,2)_{12}$ is the best model, then the model is used to identify the input structures. The model can be written as

$$(1 - 0.352B - 0.132B^2)(1 - 0.603B^{12} - 0.395B^{24})x_t = (1 - 0.477B^{12} - 0.460B^{24})a_t$$

and can be rewritten as
\[ x_t = 0.352x_{t-1} + 0.132x_{t-2} + 0.603x_{t-12} - 0.212x_{t-13} + 0.079x_{t-14}a \\
+ 0.395x_{t-24} - 0.139x_{t-25} - 0.052x_{t-26} - 0.477a_{t-12} - 0.460a_{t-24} + a_t \]

So, the function form of the model input from ARIMA model is

\[ y_t = f(x_{t-1}, x_{t-2}, x_{t-12}, x_{t-13}, x_{t-14}, x_{t-24}, x_{t-25}, x_{t-26}, a_{t-12}, a_{t-24}) \] (17)

Equation (17) represents the input lagged variables in eighth approach. The model input of this model is shown in Table 1.

5 Evaluation of performance

The performances of each model for both the training data and forecasting data are evaluated. Based on the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which are widely used for evaluating results of time series forecasting. The MAE and RMSE are defined as below:

\[ \text{MAE} = \frac{1}{n} \sum_{t=1}^{n} |y_t - \hat{y}_t| \] (18)

\[ \text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2} \] (19)

where \( y_t \) and \( \hat{y}_t \) are the observed and the forecasted at the time \( t \). The criterions to judge for the best model are relatively small of MAE and RMSE in the modeling and forecasting. Other than that, correlation coefficient sometimes also called the correlation coefficient (\( R \)) which was used as a performance measurement as well. It is a
measure of how well the predicted values from a forecast model fit with the real-life data with a perfect fit gives a coefficient of 1.0. The $R$ is defined as:

$$R = \frac{1}{n} \sum_{t=1}^{n} (y_t - \bar{y})(\hat{y}_t - \bar{\hat{y}})$$

$$\sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \bar{y})^2} \sqrt{\frac{1}{n} \sum_{t=1}^{n} (\hat{y}_t - \bar{\hat{y}})^2}$$

(20)

6 Experiment and result

6.1 Application of ANN

The key tasks in time series forecasting is the selection of the input variables and the number of neurons in the hidden layer. For the ANN models, there is no systematic approach which can be followed. The universal approximation theorem shows that a neural network with a single hidden layer with a sufficiently large number of neurons can relate any given set of inputs and a set of outputs to an arbitrary degree of accuracy. As a result, the ANN designed in this study are equipped with one single hidden layer. The determination of the number of neurons in the hidden layer is more art than science.

Determining the size of the network (the number of neurons) has important consequences for its performance. A very small a network may not reach an acceptable level of accuracy. Too many neurons may result in an inability for the network to generalize, it may lean the training patterns.

Since the number of inputs varies depending on the input determination method used, it is not possible to use the same network architecture for each model. In this study, a typical three layer ANN model with the hyperbolic tangent sigmoid transfer function from the input layer to the hidden layer, and the linear function from the hidden layer to an output layer are used for forecasting monthly river flow time series. The eight models (M1–M8) having various input structures are trained and tested by ANN
models and the optimal number of neuron in the hidden layer was identified using a trail and error procedure by the number of hidden neurons are equal to $I/2$, $I$, $2I$ and $2I+1$ where $I$ is the number of input.

The network was trained for 5000 epochs using the back-propagation algorithm with a learning rate of 0.001 and a momentum coefficient of 0.9. The networks that yielded the best results for the forecasting set were selected as the best ANN for the corresponding series. The results for the training and forecasting using a ANN model are shown in Table 2.

Table 2 lists model performance evaluation results of the M1-M8 model for ANN models with different of neurons in the hidden layer. For a training and forecasting period, ANN model based on the M8 with hidden neuron 20, obtained the best result for MAE, RMSE and $R$.

### 6.2 Application of LSSVM

There is no theory that can used to guide the selection of number of input. In this study the same inputs structures of the dataset which is M1 to M8 were used. The RBF is used as the kernel function for this study. In order to better evaluate the performance of the proposed approach, we consider a grid search of $(\gamma, \sigma^2)$ with $\gamma$ in the range 10 to 1000 and $\sigma^2$ in the range 0.01 to 1.0. For each hyperparameter pair $(\gamma, \sigma^2)$ in the search space, 5-fold cross validation on the training set is performed to predict the prediction error. Table 3 shows the performance results obtained in the training and forecasting period of the LSSVM approach.

By considering these training and forecasting period, the lowest MAE and RMSE, and the largest $R$ for the series of data was calculated from M8 model.

### 6.3 Application of hybrid SOM-LSSVM Model

In this study, a SOM-LSSVM model is developed to predict 1 month ahead forecast of river flow. According to the above results, the M8 was the best input model, and then
this model was chosen as input for the SOM-LSSVM model.

Determining the appropriate map sizes of clustering is very important for cluster validation and efficiency. For a SOM of a large map sizes, input patterns will be grouped into a large number of clusters would cause each neuron to memorize one of the input patterns, although some clusters may have only one or two members. This clustering results is not suitable for the forecasting analysis. On the otherhand, if the map size is too small, then many different data groups might be lumped into the same category and the SOM fails to show the topological relationships of input patterns. Since there is no systematic or standard method for finding the optimal number of map sizes in the clustering algorithms, the optimal map size is obtained depending on the requirements of users. In this paper, three map sizes Kohonen layers of $2 \times 2$, $3 \times 3$ and $4 \times 4$ are utilized. When SOM is applied to perform cluster analysis, a SOM of a small dimension is the first choice. If the clustering result is reasonable and satisfactory, the cluster analysis is accepted. Otherwise, a SOM of a larger dimension is chosen to analyze input patterns and this continued until a satisfactory result is obtained.

In this study, only 4 cluster was considered to investigate the impacts of the number of map sizes on the performances. We used the same parameters as the LSSVM's parameter for a single LSSVM model. Table 4 showed the predicted value of SOM-LSSVM with various number of map sizes.

The results apparently show that the SOM-LSSVM with map sizes $2 \times 2$ has better performance than the the SOM-LSSVM with map sizes $3 \times 3$ and $4 \times 4$ for both training and forecasting period.

6.4 Comparison

For further analysis, the error statistic of the optimum ARIMA, ANN, LSSVM and proposed method, SOM-LSSVM model are compared to each other. Table 5 shows the comparison of training and forecasting precision among the those four approaches based on three statistical measures.

Comparing performances of ARIMA, ANN, LSSVM and SOM-SSVM models, for Bernam rivers in training and forecasting period shows that the lowest MAE and RMSE, 8195
and the largest $R$ were calculated for SOM-LSSVM model respectively. From the Table 6, it is evident that the SOM-LSSVM performed better than the ARIMA, ANN and LSSVM models in training and forecasting process.

Figure 5 shows the comparison of scatter plots between modeled results by the four models and actual data for the last sixty months during testing stage for Bernam river, respectively. All four approaches gave close approximations of the actual observations, suggesting that these approaches are applicable for modeling river flow time series data. However, the values of $R$ and fit line equation coefficients of the SOM-LSSVM is slightly superior to the other models. The results obtained in this study indicate that the SOM-LSSVM model is powerful tools to model the river flow time series and can give good prediction performance than ARIMA, ANN and LSSVM time series approach. The results indicate that the best performance can be obtained by SOM-LSSVM model followed by LSSVM, ANN and ARIMA models.

7 Conclusions

A hybrid artificial neural network which combines the SOM with the LSSVM is proposed to forecast the river flow. One of the most important in developing a satisfactory data-driven model such as ANN and LSSVM models is the selection of the input variables. Firstly, the various input determination are performed to determined the capability and applicability of the ANN and LSSVM models to predict the river flow. By using a evaluation of performance test, the input structure based on ARIMA model is decided as the optimal input factor. Next, a cluster technique using SOM is developed to analyze this input data. The SOM algorithm cluster the training into several disjointed clusters. After decomposing the data, LSSVM is used to predict the river flow. To illustrate the capability of the SOM-LSSVM model, monthly river flow from Bernam river, located in the Selangor of Peninsular Malaysia was chosen as a case study. The results suggest that the two-stage architecture provides a promising alternative for time series forecasting. From the experimental results comparing the performance of a SOM-LSSVM, ARIMA,
LSSVM and ANN model, it indicates that SOM-LSSVM perform better than the others. We can concluded that SOM-LSSVM provides a promising alternative technique in time series forecasting.

Acknowledgements. This research is supported in part by the E-Science, Ministry of Science, Technology and Innovation (MOSTI) fundamental research grant scheme under vote number 79346.

References


Table 1. Model input for forecasting.

<table>
<thead>
<tr>
<th>Model</th>
<th>Model input</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>$y_t = f(x_{t-1}, x_{t-2})$</td>
</tr>
<tr>
<td>M2</td>
<td>$y_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4})$</td>
</tr>
<tr>
<td>M3</td>
<td>$y_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6})$</td>
</tr>
<tr>
<td>M4</td>
<td>$y_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8})$</td>
</tr>
<tr>
<td>M5</td>
<td>$y_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10})$</td>
</tr>
<tr>
<td>M6</td>
<td>$y_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12})$</td>
</tr>
<tr>
<td>M7</td>
<td>$y_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-7}, x_{t-10}, x_{t-12})$ (Stepwise)</td>
</tr>
<tr>
<td>M8</td>
<td>$y_t = f(x_{t-1}, x_{t-2}, x_{t-12}, x_{t-13}, x_{t-14}, x_{t-24}, x_{t-25}, x_{t-26}, a_{t-12}, a_{t-24})$ (ARIMA Model)</td>
</tr>
</tbody>
</table>
Table 2. The result for the training and forecasting using ANN model.

<table>
<thead>
<tr>
<th>Data</th>
<th>Hidden Neurons</th>
<th>Training</th>
<th>Forecasting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MAE</td>
<td>RMSE</td>
</tr>
<tr>
<td>M1 (I =2)</td>
<td>I/2</td>
<td>0.0967</td>
<td>0.1259</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.0961</td>
<td>0.1257</td>
</tr>
<tr>
<td></td>
<td>2I</td>
<td>0.0969</td>
<td>0.1257</td>
</tr>
<tr>
<td></td>
<td>2I+1</td>
<td>0.0971</td>
<td>0.1263</td>
</tr>
<tr>
<td></td>
<td>I/2</td>
<td>0.1150</td>
<td>0.1506</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.1135</td>
<td>0.1500</td>
</tr>
<tr>
<td></td>
<td>2I</td>
<td>0.1135</td>
<td>0.1489</td>
</tr>
<tr>
<td></td>
<td>2I+1</td>
<td>0.1126</td>
<td>0.1482</td>
</tr>
<tr>
<td>M2 (I =4)</td>
<td>I/2</td>
<td>0.1098</td>
<td>0.1411</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.0940</td>
<td>0.1258</td>
</tr>
<tr>
<td></td>
<td>2I</td>
<td>0.0909</td>
<td>0.1197</td>
</tr>
<tr>
<td></td>
<td>2I+1</td>
<td>0.0936</td>
<td>0.1232</td>
</tr>
<tr>
<td>M3 (I =6)</td>
<td>I/2</td>
<td>0.1125</td>
<td>0.1473</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.1100</td>
<td>0.1449</td>
</tr>
<tr>
<td></td>
<td>2I</td>
<td>0.1075</td>
<td>0.1404</td>
</tr>
<tr>
<td></td>
<td>2I+1</td>
<td>0.1067</td>
<td>0.1417</td>
</tr>
<tr>
<td>M4 (I =8)</td>
<td>I/2</td>
<td>0.1245</td>
<td>0.1602</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.1025</td>
<td>0.1359</td>
</tr>
<tr>
<td></td>
<td>2I</td>
<td>0.1059</td>
<td>0.1402</td>
</tr>
<tr>
<td></td>
<td>2I+1</td>
<td>0.1056</td>
<td>0.1396</td>
</tr>
<tr>
<td>M5 (I =10)</td>
<td>I/2</td>
<td>0.0868</td>
<td>0.1143</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.0870</td>
<td>0.1154</td>
</tr>
<tr>
<td></td>
<td>2I</td>
<td>0.0828</td>
<td>0.1105</td>
</tr>
<tr>
<td></td>
<td>2I+1</td>
<td>0.0838</td>
<td>0.1107</td>
</tr>
<tr>
<td>M6 (I =12)</td>
<td>I/2</td>
<td>0.0863</td>
<td>0.1150</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.0879</td>
<td>0.1155</td>
</tr>
<tr>
<td></td>
<td>2I</td>
<td>0.0852</td>
<td>0.1134</td>
</tr>
<tr>
<td></td>
<td>2I+1</td>
<td>0.0850</td>
<td>0.1127</td>
</tr>
<tr>
<td>M7 (I =7)</td>
<td>I/2</td>
<td>0.0567</td>
<td>0.0749</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>0.0595</td>
<td>0.0756</td>
</tr>
<tr>
<td></td>
<td>2I</td>
<td>0.0553</td>
<td>0.0716</td>
</tr>
<tr>
<td></td>
<td>2I+1</td>
<td>0.0653</td>
<td>0.0834</td>
</tr>
</tbody>
</table>
Table 3. The result for the training and forecasting using LSSVM model.

<table>
<thead>
<tr>
<th>Data</th>
<th>MAE</th>
<th>RMSE</th>
<th>R</th>
<th>MAE</th>
<th>RMSE</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>0.0955</td>
<td>0.1248</td>
<td>0.5494</td>
<td>0.0858</td>
<td>0.1080</td>
<td>0.5191</td>
</tr>
<tr>
<td>M2</td>
<td>0.0850</td>
<td>0.1120</td>
<td>0.6757</td>
<td>0.0860</td>
<td>0.1084</td>
<td>0.5207</td>
</tr>
<tr>
<td>M3</td>
<td>0.0829</td>
<td>0.1120</td>
<td>0.6647</td>
<td>0.0797</td>
<td>0.1000</td>
<td>0.6092</td>
</tr>
<tr>
<td>M4</td>
<td>0.0853</td>
<td>0.1134</td>
<td>0.6564</td>
<td>0.0812</td>
<td>0.1021</td>
<td>0.6037</td>
</tr>
<tr>
<td>M5</td>
<td>0.0773</td>
<td>0.1035</td>
<td>0.7767</td>
<td>0.0800</td>
<td>0.1084</td>
<td>0.5232</td>
</tr>
<tr>
<td>M6</td>
<td>0.0744</td>
<td>0.1018</td>
<td>0.7308</td>
<td>0.0744</td>
<td>0.0995</td>
<td>0.6191</td>
</tr>
<tr>
<td>M7</td>
<td>0.0800</td>
<td>0.1075</td>
<td>0.7033</td>
<td>0.0775</td>
<td>0.1038</td>
<td>0.5776</td>
</tr>
<tr>
<td>M8</td>
<td>0.0438</td>
<td>0.0579</td>
<td>0.9319</td>
<td>0.0457</td>
<td>0.0621</td>
<td>0.8727</td>
</tr>
</tbody>
</table>
Table 4. The result for the training and forecasting using a hybrid model of SOM-LSSVM for different of map sizes.

<table>
<thead>
<tr>
<th>Map Sizes</th>
<th>Training</th>
<th>Forecasting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>RMSE</td>
</tr>
<tr>
<td>2 x 2</td>
<td>0.0211</td>
<td>0.0333</td>
</tr>
<tr>
<td>3 x 3</td>
<td>0.0278</td>
<td>0.0378</td>
</tr>
<tr>
<td>4 x 4</td>
<td>0.0340</td>
<td>0.0475</td>
</tr>
</tbody>
</table>
Table 5. Comparative performance between forecasting models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th>Forecasting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>RMSE</td>
</tr>
<tr>
<td>ARIMA</td>
<td>0.0797</td>
<td>0.1049</td>
</tr>
<tr>
<td>ANN</td>
<td>0.0553</td>
<td>0.0716</td>
</tr>
<tr>
<td>LSSVM</td>
<td>0.0438</td>
<td>0.0579</td>
</tr>
<tr>
<td>SOM-LSSVM</td>
<td>0.0211</td>
<td>0.0333</td>
</tr>
</tbody>
</table>
Fig. 1. The SOM architecture.
Fig. 2. The SOM-LSSVM architecture.
Fig. 3. The study area.
Fig. 4. ACF and PACF for Monthly River Flow of Bernam River.
Fig. 5. Predicted and observed streamflow during testing period by ARIMA, ANN, LSSVM and SOM-LSSVM for Bernam River.