River flow time series using least squares support vector machines

R. Samsudin\textsuperscript{1}, P. Saad\textsuperscript{1}, and A. Shabri\textsuperscript{2}

\textsuperscript{1}Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia
\textsuperscript{2}Faculty of Science, Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia

Received: 27 April 2010 – Published in Hydrol. Earth Syst. Sci. Discuss.: 23 June 2010
Revised: 25 May 2011 – Accepted: 29 May 2011 – Published: 17 June 2011

Abstract. This paper proposes a novel hybrid forecasting model known as GLSSVM, which combines the group method of data handling (GMDH) and the least squares support vector machine (LSSVM). The GMDH is used to determine the useful input variables which work as the time series forecasting for the LSSVM model. Monthly river flow data from two stations, the Selangor and Bernam rivers in Selangor state of Peninsular Malaysia were taken into consideration in the development of this hybrid model. The performance of this model was compared with the conventional artificial neural network (ANN) models, Autoregressive Integrated Moving Average (ARIMA), GMDH and LSSVM models using the long term observations of monthly river flow discharge. The root mean square error (RMSE) and coefficient of correlation ($R$) are used to evaluate the models’ performances. In both cases, the new hybrid model has been found to provide more accurate flow forecasts compared to the other models. The results of the comparison indicate that the new hybrid model is a useful tool and a promising new method for river flow forecasting.

1 Introduction

River flow forecasting is one of the most important components of hydrological processes in water resource management. Accurate estimations for both short and long term forecasts of river flow can be used in several water engineering problems such as designing flood protection works for urban areas and agricultural land and optimizing the allocation of water for different sectors such as agriculture, municipalities, hydropower generation, while ensuring that environmental flows are maintained. The identification of highly accurate and reliable river flow models for future river flow is an important precondition for successful planning and management of water resources.

Generally, river flow models can be grouped into the two main techniques: knowledge-driven modelling and data-driven modelling. The knowledge-driven modelling is known as the physically-based model approaches, which generally use a mathematical framework based on catchment characteristics such as storm characteristics (intensity and duration of rainfall events), catchment characteristics (size, shape, slope and storage characteristics of the catchment), geomorphologic characteristics of a catchment (topography, land use patterns, vegetation and soil types that affect the infiltration) and climatic characteristics (temperature, humidity and wind characteristics) (Jain and Kumar, 2007). This model requires input of initial and boundary conditions since these flow processes are described by differential equations (Rientjes, 2004). In the river flow modelling and forecasting, it is hypothesized that the forecasts could be improved if catchment characteristics variables which affect flow were to be included. It is likely that the different combinations of flow and catchment characteristics variables would improve the forecast ability of the models. Although incorporating other variables may improve the prediction accuracy, but, in practice especially in developing countries like Malaysia, such information is often either unavailable or difficult to obtain. Moreover, the influence of these variables and many of their combinations in generating streamflow is an extremely complex physical process especially due to the data collection of multiple inputs and parameters, which vary in space and time (Akhtar et al., 2009), and are not clearly understood (Zhang and Govindaraju, 2000). Owing to the complexity of this process, most conventional approaches are unable to provide sufficiently accurate and reliable results (Firat and Turan, 2010).

Published by Copernicus Publications on behalf of the European Geosciences Union.
The second approach which is the data-driven modelling is based on extracting and re-using information that is implicitly contained in the hydrological data without directly taking into account the physical laws that underlie the rainfall-runoff processes. In river flow forecasting applications, data-driven modelling using historical river flow time series data is becoming increasingly popular due to its rapid development and minimum information requirements (Adamowski and Sun, 2010; Atiya et al., 1999; Lin et al., 2006; Wang et al., 2006, 2009; Wu et al., 2009; Firat and Gungor, 2007; Firat, 2008; Kisi, 2008, 2009). Although the data-driven modelling may lack the ability to provide physical interpretation and insight of the catchment processes but it is able to provide relatively accurate flow forecasts.

Computer science and statistics have improved the data-driven modelling approaches for discovering patterns found in water resources time series data. Much effort has been devoted over the past several decades to the development and improvement of time series prediction models. One of the most important and widely used time series models is the autoregressive integrated moving average (ARIMA) model. The popularity of the ARIMA model is due to its statistical properties as well as the well-known Box-Jenkins methodology. Literature on the extensive applications and reviews of ARIMA model proposed for modeling of water resources time series are indicative of researchers’ preference (Yurekli et al., 2004; Muhamad and Hassan, 2005; Huang et al., 2004; Modarres, 2007; Fernandez and Vega, 2009; Wang et al., 2009). However, the ARIMA model provides only a reasonable level of accuracy and suffer from the assumptions of stationary and linearity.

The data-driven models such as artificial neural networks (ANN) have recently been accepted as an efficient alternative tool for modelling a complex hydrologic system compared with the conventional methods and is widely used for prediction (Karunasinghe and Liong, 2006; Rojas et al., 2008; Camasta and Colla, 1999; Han and Wang, 2009; Abraham and Nath, 2001). ANN has emerged as one of the most successful approaches in the various areas of water-related research, particularly in hydrology. A comprehensive review of the application of ANN in hydrology was presented by the ASCE Task Committee report (2000). Some specific applications of ANN to hydrology include modelling river flow forecasting (Dolling and Varas, 2003; Muhamad and Hassan, 2005; Kisi, 2008; Wang et al., 2009; Kesimal and Taylan, 2009), rainfall-runoff modeling (De Vos and Rientjes, 2005; Hsu et al., 1995; Shamseldin, 1997; Hung et al., 2009), ground water management (Affandi and Watanabe, 2007; Birkinshaw et al., 2008) and water quality management (Maier and Dandy, 2003). However, there are some disadvantages of the ANN. Its network structure is hard to determine and this is usually determined by using a trial-and-error approach (Kisi, 2004).

More advanced artificial intelligent (AI) is the support vector machine (SVM) proposed by Vapnik (1995) and his co-workers in 1995 based on the statistical learning theory, has gained the attention of many researchers. SVM has been applied to time series prediction with promising results as seen in the works of Tay and Cao (2001), Thiessen and Van Brakel (2003) and Misra et al. (2009). Several studies have also been carried out using SVM in hydrological and water resources planning (Wang et al., 2009; Asefa et al., 2006; Lin et al., 2006; Dibike et al., 2001; Liong and Sivapragasam, 2002; Yu et al., 2006). The standard SVM is solved using quadratic programming methods. However, this method is often time consuming and has a high computational burden because of the required constrained optimization programming.

Least squares support vector machines (LSSVM), as a modification of SVM was introduced by Suykens and Vandewalle (1999). LSSVM is a simplified form of SVM that uses equality constraints instead of inequality constraints and adopts the least squares linear system as its loss function, which is computationally attractive. Besides that, it 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 LS-SVM is that it is computationally very cheap besides having the important properties of the SVM. LSSVM has been successfully applied in diverse fields (Afshin et al., 2007; Lin et al., 2005; Sun and Guo, 2005; Gestel et al., 2001). However, in the water resource filed, this LSSVM method has received very little attention and there are only a few applications of LSSVM to modeling of environmental and ecological systems such as water quality prediction (Yunrong and Liangzhong, 2009).

One sub-model of ANN is a group method data handling (GMDH) algorithm which was first developed by Ivakhnenko (1971). This is a multivariate analysis method for modeling and identification of complex systems. The main idea of GMDH is to build an analytical function in a feed-forward network based on a quadratic node transfer function whose coefficients are obtained by using the regression technique. This model has been successfully used to deal with uncertainty and linear or nonlinearity systems in a wide range of disciplines such as engineering, science, economy, medical diagnostics, signal processing and control systems (Tanaka and Kondo, 1980; Ivakhnenko and Ivakhnenko, 1995; Voss and Feng, 2002). In water resource, the GMDH method has received very attention and only a few applications to modeling of environmental and ecological systems (Chang and Hwang, 1999; Onwubolu et al., 2007; Wang et al., 2005) have been carried out.

Improving forecasting especially for the accuracy of river flow is an important yet often difficult task faced by decision makers. Most of the studies as reported earlier in this paper were simple applications of using traditional time series approaches and data-driven models such as ANN, SVM, LSSVM and GMDH models. Many of the river flow series...
are extremely complex to be modeled using these simple approaches especially when a high level of accuracy is required. Different data-driven models can achieve success which is different from each other as each would capture various patterns of data sets, and numerous authors have demonstrated that a hybrid based on the predictions of several models frequently results in higher prediction accuracy than the prediction of an individual model. The hybrid model is widely used in diverse fields, such as economics, business, statistics and meteorology (Zhang, 2003; Jain and Kumar, 2006; Su et al., 1997; Wang et al., 2005; Chen and Wang, 2007; Onwubolu, 2008; Yang et al., 2006). Many studies have also developed a number of hybrid forecasting models in hydrological processes in order to improve prediction accuracy as reported in the literature. See and Openshaw (2009) proposed a hybrid model that combines fuzzy logic, neural networks and statistical-based modeling to form an integrated river level forecasting methodology. Another study by Wang et al. (2005) presented a hybrid methodology to exploit the unique strength of GMDH and ANN models for river flow forecasting. Besides that Jain and Kumar (2006) proposed a hybrid approach for time series forecasting using monthly stream flow data at Colorado river. Their study indicated that the approach of combining the strengths of the conventional and ANN techniques provided a robust modeling framework capable of capturing the nonlinear nature of the complex time series, thus producing more accurate forecasts.

In this paper, a novel hybrid approach combining GMDH model and LSSVM model is developed to forecast river flow time series data. The hybrid model combines GMDH and LSSVM into a methodology known as GLSSVM. In the first phase, GMDH is used to determine the useful input variables from the under study time series. Then, in the second phase, the LSSVM is used to model the generated data by GMDH model to forecast the future value of the time series. To verify the application of this approach, the hybrid model was compared with ARIMA, ANN, GMDH and LSSVM models using two river flow data sets: the Selangor and Bernam rivers located in Selangor, Malaysia.

2 Individual forecasting models

This section presents the ARIMA, ANN, GMDH and LSSVM models used for modeling time series. The reason for choosing these models in this study were because these methods have been widely and successfully used in forecasting time series.

2.1 The Autoregressive Integrated Moving Average (ARIMA) models

The ARIMA models introduced by Box and Jenkins (1970), has been one of the most popular approaches in the analysis of time series and prediction. The general ARIMA models are compound of a seasonal and non-seasonal part are represented as:

\[ \phi_p(B) \Phi_P(B^s) (1 - B)^d (1 - B^s)^D x_t = \theta_q(B) \Theta_q(B^s) a_t(1) \]

where \( \phi(B) \) and \( \theta(B) \) are polynomials of order \( p \) and \( q \), respectively; \( \Phi(B^s) \) and \( \Theta(B^s) \) are polynomials in \( B^s \) of degrees \( P \) and \( Q \), respectively; \( p \) is the order of non-seasonal auto regression; \( d \) is the number of regular differencing; \( q \) is the order of the non-seasonal moving average; \( P \) is the order of seasonal auto regression; \( D \) is the number of seasonal differencing; \( Q \) is the order of seasonal moving average; and \( s \) is the length of season. Random errors, \( a_t \) are assumed to be independently and identically distributed with a mean of zero and a constant variance of \( \sigma^2 \). The order of an ARIMA model is represented by ARIMA(\( p, d, q \)) and the order of an seasonal ARIMA model is represented by ARIMA(\( p, d, q \) × \( P, D, Q \)). The term \( (p, d, q) \) is the order of the non-seasonal part and \( (P, D, Q) \) is the order of the seasonal part.

The Box-Jenkins methodology is basically divided into four steps: identification, estimation, diagnostic checking and forecasting. In the identification step, transformation is often needed to make time series stationary. The behavior of the autocorrelation (ACF) and partial autocorrelation function (PACF) is used to see whether the series is stationary or not, seasonal or non-seasonal. The next step is choosing a tentative model by matching both ACF and PACF of the stationary series. Once a tentative model is identified, the parameters of the model are estimated. Then, the last step of model building is the diagnostic checking of model adequacy. Basically this is done to check if the model assumptions about the error, \( a_t \) are satisfied. If the model is not adequate, a new tentative model should be identified followed by the steps of parameter estimation and model verification. This process is repeated several times until a satisfactory model is finally selected. The forecasting model would then be used to compute the fitted values and forecasts values.

To be a reliable forecasting model, the residuals must satisfy the requirements of a white noise process i.e. independent and normally distributed around a zero mean. In order to determine whether the river flow time series are independent, two diagnostic checking statistics using the ACF of residuals of the series were carried out (Brockwell and Davis, 2002). The first one is the correlograms drawn by plotting the ACF of residuals against a lag number. If the model is adequate, the estimated ACF of the residual is independent and distributed approximately normally about zero. The second one is the Ljung-Box-Pierce statistics which are calculated for the different total numbers of successive lagged ACF of residual in order to test the adequacy of the model.

The Akaike’s Information Criterion (AIC) is also used to evaluate the goodness of fit with smaller values would indicate a better fitting and more parsimonious model than larger
values (Akaike, 1974). Mathematical formulation of AIC is defined as:

\[
AIC = \ln \left( \frac{\sum_{t=1}^{n} e_t^2}{n} \right) + \frac{2p}{n} \tag{2}
\]

where \( p \) is the number of parameters and \( n \) is the periods of data.

2.2 The Artificial Neural Network (ANN) model

The ANN models based on flexible computing have been extensively studied and used for time series forecasting in many areas of science and engineering since early 1990s. The ANN is a mathematical model which has a highly connected structure similar to brain cells. This model has the capability to execute complex mapping between input and output and could form a network that approximates non-linear functions. A single hidden layer feed forward network is the most widely used model form for time series modeling and forecasting (Zhang et al., 1998). This model usually consists of three layers: the first layer is the input layer where the data are introduced to the network followed by the hidden layer where data are processed and the final or output layer is where the results of the given input are produced. The structure of a feed-forward ANN is shown in Fig. 1.

The output of the ANN assuming a linear output neuron \( j \), a single hidden layer with \( h \) sigmoid hidden nodes and the output variable \( (x_t) \) is given by:

\[
x_t = g \left( \sum_{j=1}^{h} w_j f(s_j) + b_k \right) \tag{3}
\]

where \( g(.) \) is the linear transfer function of the output neuron \( k \) and \( b_k \) is its bias, \( w_j \) is the connection weights between hidden layers and output units, \( f(.) \) is the transfer function of the hidden layer (Coulibaly and Evora, 2007). The transfer functions can take several forms and the most widely used transfer functions are:

- Log-sigmoid: \( f(s_j) = \logsig(s_j) = \frac{1}{1 + \exp(-s_j)} \tag{4} \)
- Linear: \( f(s_j) = \text{purelin}(s_j) = s_j \)
- Hyperbolic tangent sigmoid: \( f(s_j) = \text{tansig}(s_j) = \frac{2}{1 + \exp(-2s_j)} - 1 \)

where \( s_j = \sum_{i=1}^{n} w_{ij} x_i \) is the input signal referred to as the weighted sum of incoming information.

In a univariate time series forecasting problem, the inputs of the network are the past lagged observations \((x_{t-1}, x_{t-2}, ..., x_{t-p})\) and the output is the predicted value \((x_t)\) (Zhang et al., 2001). Hence the ANN of Eq. (3) can be written as:

\[
x_t = g \left( x_{t-1}, x_{t-2}, ..., x_{t-p}, w \right) + \epsilon_t \tag{5}
\]

2.3 The Least Square Support Vector Machines (LSSVM) model

The LSSVM is a new technique for regression. In this technique, the predictor is trained by using a set of time series historic values as inputs and a single output as the target value. In the following sections, discussions on how LSSVM is used for time series forecasting is presented.

The first step would be to consider a given training set of \( n \) data points \( \{x_i, y_i\}_{i=1}^{n} \) with input data \( x_i \in \mathbb{R}^n \), \( p \) is the total number of data patterns and output \( y_i \in \mathbb{R} \). SVM approximates the function in the following form:

\[
y(x) = \mathbf{w}^T \phi(x) + b \tag{6}
\]

where \( \phi(x) \) represents the high dimensional feature spaces which is mapped in a non-linear manner from the input space \( x \). In the LSSVM for function estimation, the optimization problem is formulated (Suykens et al., 2002) as:

\[
\min J(w, e) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\gamma}{2} \sum_{i=1}^{n} e_i^2 \tag{7}
\]
Subject to the equality constraints:
\[ y(x) = \mathbf{w}^T \phi(x_i) + b + e_i \quad i = 1, 2, ..., n \]  
(8)

The solution is obtained after constructing the Lagrange:
\[ L(\mathbf{w}, b, e, \alpha) = J(\mathbf{w}, e) - \sum_{i=1}^{n} \alpha_i (\mathbf{w}^T \phi(x_i) + b + e_i - y_i) \]  
(9)

With Lagrange multipliers \( \alpha_i \). The conditions for optimality are:
\[ \frac{\partial L}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i \phi(x_i), \]
\[ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0, \]
\[ \frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, \]
\[ \frac{\partial L}{\partial \alpha_i} = 0 \rightarrow \mathbf{w}^T \phi(x_i) + b + e_i - y_i = 0, \]  
(10)

for \( i = 1, 2, ..., n \). After elimination of \( e_i \) and \( \mathbf{w} \), the solution is given by the following set of linear equations:
\[ \begin{bmatrix} 0 \\ 1 \phi(x_i)^T \phi(x_i) + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \]  
(11)

where \( y = [y_1, ..., y_n] \), \( \mathbf{1} = [1, ..., 1] \), \( \alpha = [\alpha_1, ..., \alpha_n] \). According to Mercer’s condition, the kernel function can be defined as:
\[ K(x_i, x_j) = \phi(x_i)^T \phi(x_j), \quad i, j = 1, 2, ..., n \]  
(12)

This finally leads to the following LSSVM model for function estimation:
\[ y(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x_j) + b \]  
(13)

where \( \alpha_i, b \) are the solution to the linear system. Any function that satisfies Mercer’s condition can be used as the kernel function. The choice of the kernel function \( K(\cdot, \cdot) \) has several possibilities. \( K(x_i, x_j) \) is defined as the kernel function. The value of the kernel is equal to the inner product of two vectors \( \mathbf{X}_i \) and \( \mathbf{X}_j \) in the feature space \( \phi(x_i) \) and \( \phi(x_j) \), that is, \( K(x_i, x_j) = \phi(x_i) \times \phi(x_j) \). The structure of a LSSVM is shown in Fig. 2.

Typical examples of the kernel functions are:

**Linear**: \( K(x_i, x_j) = x_i^T x_j \)

**Sigmoid**: \( K(x_i, x_j) = \tanh (\gamma x_i^T x_j + r) \)

**Polynomial**: \( K(x_i, x_j) = \left(\gamma x_i^T x_j + r\right)^d, \quad \gamma > 0 \)

**Radial basis function (RBF)**: \( K(x_i, x_j) = \exp \left(-\gamma \left\| x_i - x_j \right\|^2\right), \quad \gamma > 0 \)  
(14)

Here \( \gamma, r \) and \( d \) are the kernel parameters. These parameters should be carefully chosen as they implicitly define the structure of the high dimensional feature space \( \phi(x) \) and would control the complexity of the final solution.

---

Fig. 2. Architecture of GMDH.
2.4 The Group Method of Data Handling (GMDH) model

The algorithm of GMDH was introduced by Ivakhnenko in the early 1970s as a multivariate analysis method for modeling and identification of complex systems. This method was originally formulated to solve higher order regression polynomials specially for solving modeling and classification problems. The general connection between the input and the output variables can be expressed by complicated polynomial series in the form of the Volterra series known as the Kolmogorov-Gabor polynomial (Ivakhnenko, 1971):

\[ y = a_0 + \sum_{i=1}^{M} a_i x_i + \sum_{i=1}^{M} \sum_{j=1}^{M} a_{ij} x_i x_j + \ldots \]  

(15)

where \( x \) is the input to the system, \( M \) is the number of inputs and \( a_i \) are coefficients or weights. However, many of the applications of the quadratic form are called partial descriptions (PD) where only two of the variables are used in the following form:

\[ y = a_0 + a_1 x_1 + a_2 x_j + a_3 x_1 x_j + a_4 x_1^2 + a_5 x_j^2 \]  

(16)

to predict the output. To obtain the value of the coefficients \( a_i \) for each \( m \) models, a system of Gauss normal equations is solved. The coefficient \( a_i \) of nodes in each layer are expressed in the form:

\[ A = (X^T X)^{-1} X^T Y \]  

(17)

where \( Y = [y_1 \ y_2 \ldots \ y_M]^T \), \( A = [a_0, a_1, a_2, a_3, a_4, a_5] \),

\[
X = \begin{bmatrix}
1 & x_{1p} & x_{1q} & x_{1p} x_{1q} & x_{1p}^2 & x_{1q}^2 \\
1 & x_{2p} & x_{2q} & x_{2p} x_{2q} & x_{2p}^2 & x_{2q}^2 \\
& & & & & \\
& & & & & \\
1 & x_{Mp} & x_{Mq} & x_{Mp} x_{Mq} & x_{Mp}^2 & x_{Mq}^2 \\
\end{bmatrix}
\]  

(18)

and \( M \) is the number of observations in the training set.

The main function of GMDH is based on the forward propagation of signal through nodes of the net similar to the principal used in classical neural nets. Every layer consists of simple nodes ans each one performs its own polynomial transfer function and then passes its output to the nodes in the next layer. The basic steps involved in the conventional GMDH modeling (Nariman-Zadeh et al., 2002) are:

- Step 1: Select normalized data \( X = [x_1, x_2, \ldots, x_M] \) as input variables. Divide the available data into training and testing data sets.

- Step 2: Construct \( M C_2 = M (M - 1) / 2 \) new variables in the training data set and construct the regression polynomial for the first layer by forming the quadratic expression which approximates the output \( y \) in Eq. (16).

- Step 3: Identify the contributing nodes at each of the hidden layer according to the value of mean root square error (RMSE). Eliminate the least effective variable by replacing the columns of \( X \) (old columns) with the new columns \( Z \).

- Step 4: The GMDH algorithm is carried out by repeating steps 2 and 3 of the algorithm. When the errors of the test data in each layer stop decreasing, the iterative computation is terminated.

The configuration of the conventional GMDH structure is shown in Fig. 3.

2.5 The hybrid model

In this proposed method, the combination of GMDH and LSSVM as a hybrid model to become GLSSVM is applied to enhance its capability. The input variables selected are based on the results of the GMDH and LSSVM models which would then be used as the time series forecasting. The hybrid model procedure is carried out in the following manner:

- Step 1: The normalized data are separated into the training and testing sets data.

- Step 2: All combinations of two input variables \((x_i, x_j)\) are generated in each layer. The number of input variables are \( M C_2 = \frac{M (M - 1)}{2} \). Construct the regression polynomial for this layer by forming the quadratic expression which approximates the output \( y \) in Eq. (10).
The coefficient vector of the PD is determined by the least square estimation approach.

- Step 3: Determine new input variables for the next layer. The output $x'$ variable which gives the smallest of root mean square error (RMSE) for the train data set is combined with the input variables $\{x_1, x_2, ..., x_M, x'\}$ with $M = M + 1$. The new input $\{x_1, x_2, ..., x_M, x'\}$ of the neurons in the hidden layers are used as input for the LSSVM model.

- Step 4: The GLSSVM algorithm is carried out by repeating steps 2 to 4 until $k = 5$ iterations. The GLSSVM model with the minimum value of the RMSE is selected as the output model. The configuration of the GLSSVM structure is shown in Fig. 4.

3 Case study

In this study, monthly flow data from Selangor and Bernam rivers in Selangor, Malaysia have been selected as the study sites. The location of these rivers are shown in Fig. 5. Bernam river is located between the Malaysian states of Perak and Selangor, demarcating the border of the two states whereas Selangor river is a major river in Selangor, Malaysia. The latter runs from Kuala Kubu Bharu in the east and converges into the Straits of Malacca at Kuala Selangor in the west.

The catchment area at Selangor site ($3.24^\circ, 101.26^\circ$) is $1450 \text{ km}^2$ and the mean elevation is $8 \text{ m}$ whereas the catchment area at Bernam site ($3.48^\circ, 101.21^\circ$) is $1090 \text{ km}^2$ with the mean elevation is $19 \text{ m}$. Both these rivers basins have significant effects on the drinking water supply, irrigation and aquaculture activities such as the cultivation of fresh water fishes for human consumption.

The periods of the observed data are 47 years (564 months) with an observation period between January 1962 and December 2008 for Selangor river and 43 years (516 months) from January 1966 to December 2008 for Bernam river. The training dataset of 504 monthly records (Jan. 1962 to Dec. 2004) for Selangor river and 456 monthly records (Jan. 1966 to Dec. 2004) was used to train the network to obtain parameters model. Another dataset consisting of 60 monthly (Jan. 2005 to Dec. 2008) records was used as testing dataset for both stations (Fig. 6).

Before starting the training, the collected data were normalized within the range of 0 to 1 by using the following formula:

$$x_i = 0.1 + \frac{y_i}{1.2 \max(y_i)}$$  \hspace{1cm} (19)

where $x_i$ is the normalized value, $y_i$ is the actual value and $\max(y_i)$ is the maximum value in the collected data.

The performances of each model for both training and forecasting data are evaluated according to the root-mean-square error (RMSE) and correlation coefficient ($R$) which are widely used for evaluating results of time series forecasting. The RMSE and $R$ are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - o_i)^2}$$  \hspace{1cm} (20)

$$R = \frac{\frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})(o_i - \bar{o})}{\sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (o_i - \bar{o})^2}}$$  \hspace{1cm} (21)

where $o_i$ and $y_i$ are the observed and forecasted values at data point $i$, respectively, $\bar{y}$ is the mean of the observed values, and $n$ is the number of data points. The criterions to judge for the best model are relatively small of RMSE in the training and testing. Correlation coefficient measures how well the flows predictions correlate with the flows observations. Clearly, the $R$ value close to unity indicates a satisfactory result, while a low value or close to zero implies an inadequate result.
4 Result and discussion

4.1 Fitting the ARIMA models to the data

The sample autocorrelation function (ACF) and partial autocorrelation function (PACF) for Selangor and Bernam river series are plotted in Figs. 7 and 8 respectively. The ACFs curve of the monthly flow data of these rivers 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. For PACF, there were significant lags at spikes from lag 1 to 5, which suggest an AR process. In the PACF, there were significant spikes present near lags 12 and 24, and therefore the series would be needed for seasonal AR process. The identification of best model for river flow series is based on minimum AIC as shown in Table 1. The criteria to judge the best model based on AIC show that ARIMA \((1, 0, 0) \times (1, 0, 1)_{12}\) was selected as the best model for Selangor river and the ARIMA \((2, 0, 0) \times (2, 0, 2)_{12}\) would be relatively the best model for Bernam river.

Since the ARIMA \((1, 0, 0) \times (1, 0, 1)_{12}\) is the best model for Selangor river and ARIMA \((2, 0, 0) \times (2, 0, 2)_{12}\) for Bernam river, then the model is used to identify the input structures. The ARIMA \((2, 0, 0) \times (2, 0, 2)_{12}\) model can be written as:

\[
(1 - 0.3515 B - 0.1351 B^2) (1 - 0.7014 B^{12} - 0.2933 B^{24}) x_t = (1 - 0.5802 B_{12} - 0.3720 B_{24}) a_t
\]

\[
x_t = (1 - 0.5802 B^{12} - 0.3720 B^{24}) a_t
\]

\[
x_t = 0.3515 x_{t-1} + 0.1351 x_{t-2} + 0.7014 x_{t-12} + 0.2465 x_{t-13} - 0.0948 x_{t-14} + 0.2933 x_{t-24}
\]

and the ARIMA \((1, 0, 0) \times (1, 0, 1)_{12}\) model can be written as:

\[
(1 - 0.4013 B) (1 - 0.9956 B^{12}) x_t = (1 - 0.9460 B) a_t
\]

\[
x_t = 0.4013 x_{t-1} + 0.9956 x_{t-12} - 0.3995 x_{t-13} - 0.9460 a_{t-12} + a_t
\]

The above equation for Selangor river can be rewritten as:

\[
x_t = f (x_{t-1}, x_{t-12}, x_{t-13}, a_{t-12})
\] (22)

and for Bernam river as:

\[
x_t = f (x_{t-1}, x_{t-2}, x_{t-12}, x_{t-13}, x_{t-14}, x_{t-24}, x_{t-25}, x_{t-26}, a_{t-12}, a_{t-24})
\] (23)

Table 1. Comparison of ARIMA models’ Statistical Results for Selangor and Bernam rivers.

<table>
<thead>
<tr>
<th>ARIMA Model</th>
<th>AIC</th>
<th>ARIMA Model</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1, 0, 0) \times (1, 0, 1)_{12})</td>
<td>-4.765</td>
<td>((1, 0, 0) \times (1, 0, 1)_{12})</td>
<td>-4.458</td>
</tr>
<tr>
<td>((1, 0, 0) \times (3, 0, 0)_{12})</td>
<td>-4.620</td>
<td>((5, 0, 0) \times (2, 0, 2)_{12})</td>
<td>-4.251</td>
</tr>
<tr>
<td>((1, 0, 0) \times (1, 0, 0)_{12})</td>
<td>-4.514</td>
<td>((3, 0, 0) \times (2, 0, 1)_{12})</td>
<td>-4.459</td>
</tr>
<tr>
<td>((1, 0, 1) \times (3, 0, 0)_{12})</td>
<td>-4.614</td>
<td>((2, 0, 0) \times (1, 0, 1)_{12})</td>
<td>-4.466</td>
</tr>
<tr>
<td>((1, 0, 1) \times (1, 0, 1)_{12})</td>
<td>-4.757</td>
<td>((2,0,0) \times (2,0,2)_{12})</td>
<td>-4.467</td>
</tr>
</tbody>
</table>
Table 2. The Input Structure of the Models for Forecasting of Selangor River Flow.

<table>
<thead>
<tr>
<th>Model</th>
<th>Input Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>$x_t = f(x_{t-1}, x_{t-2})$</td>
</tr>
<tr>
<td>M2</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4})$</td>
</tr>
<tr>
<td>M3</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6})$</td>
</tr>
<tr>
<td>M4</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8})$</td>
</tr>
<tr>
<td>M5</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10})$</td>
</tr>
<tr>
<td>M6</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12})$</td>
</tr>
<tr>
<td>M7</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12})$</td>
</tr>
<tr>
<td>M8</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-10}, x_{t-11}, x_{t-12})$</td>
</tr>
<tr>
<td>M9</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12})$</td>
</tr>
</tbody>
</table>

**Fig. 6.** Time series of monthly river flow of Selangor and Bernam rivers.

**4.2 Fitting ANN to the data**

One of the most important steps in developing a satisfactory forecasting model such as ANN and LSSVM models is the selection of the input variables. In this study, the nine input structures which having various input variables are trained and tested by LSSVM and ANN. Four approaches were used to identify the input structures. The first approach, six model inputs were chosen based on the past river flow. The appropriate lags were chosen by setting the input layer nodes equal to the number of the lagged variables from river flow data, $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$ where $p$ is 2, 4, 6, 8, 10 and 12. The second, third and forth approaches were identified using correlation analysis, stepwise regression analysis and ARIMA model, respectively. The model input structures of these forecasting models are shown in Tables 2 and 3.

In this study, a typical three-layer feed-forward ANN model has been constructed for forecasting the monthly river flow time series. The training and testing data were normalized within the range of zero to one. From the input layer
Table 3. The Input Structure of the Models for Forecasting of Bernam River Flow.

<table>
<thead>
<tr>
<th>Model</th>
<th>Input Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>$x_t = f(x_{t-1}, x_{t-2})$</td>
</tr>
<tr>
<td>M2</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4})$</td>
</tr>
<tr>
<td>M3</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6})$</td>
</tr>
<tr>
<td>M4</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8})$</td>
</tr>
<tr>
<td>M5</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10})$</td>
</tr>
<tr>
<td>M6</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}, x_{t-10}, x_{t-11}, x_{t-12})$</td>
</tr>
<tr>
<td>M7</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-10}, x_{t-11}, x_{t-12})$</td>
</tr>
<tr>
<td>M8</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-7}, x_{t-10}, x_{t-12})$</td>
</tr>
<tr>
<td>M9</td>
<td>$x_t = f(x_{t-1}, x_{t-2}, x_{t-12}, x_{t-13}, x_{t-14}, x_{t-24}, x_{t-25}, x_{t-26}, a_{t-12}, a_{t-24})$</td>
</tr>
</tbody>
</table>

Fig. 7. The autocorrelation and partial autocorrelation of river flow series of Selangor River.

Fig. 8. The autocorrelation and partial autocorrelation of river flow series of Bernam river.

to the hidden layer, the hyperbolic tangent sigmoid transfer function commonly used in hydrology was applied. From the hidden layer to the output layer, a linear function was employed as the transfer function because the linear function is known to be robust for a continuous output variable.

The network was trained for 5000 epochs using the conjugate gradient descent back-propagation algorithm with a learning rate of 0.001 and a momentum coefficient of 0.9. The nine models (M1–M9) having various input structures were trained and tested by these ANN models. In addition, the optimal number of neurons in the hidden layer was identified using several practical guidelines. These included the use of I/2 (Kang, 1991), I (Tang and Fishwick, 1993), 2I (Wong, 1991) and 2I + 1 (Lipmann, 1987), where I is the number of input. The effect of changing the number of hidden neurons on the RMSE and $R$ of the data set is shown in Table 4.
Table 4 shows the performance of ANN varying with the number of neurons in the hidden layer.

In the training phase for Selangor river, the M6 model with the number of hidden neurons I obtained the best RMSE and $R$ statistics of 0.0967 and 0.6677, respectively. While in testing phase, the M9 model with $2I + 1$ numbers of hidden neurons had the best RMSE and $R$ statistics of 0.1097 and 0.6163, respectively.

On the other hand, for the Bernam river, the M9 model with the number of hidden neurons was $I/2$ obtained the best RMSE and $R$ statistics, in the training and testing phase. Hence, according to these performances indices, ANN(4, 9, 1) has been selected as the most appropriate ANN model for Selangor river whereas ANN (10, 5, 1) would be best for Bernam river.
Table 5. The RMSE and $R$ statistics of GMDH, LSSVM and GLSSVM Models for Selangor and Bernam River.

<table>
<thead>
<tr>
<th>Model</th>
<th>Input</th>
<th>Selangor River</th>
<th>Bernam River</th>
<th>Bernam River</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Training</td>
<td>Testing</td>
<td>Training</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>$R$</td>
<td>RMSE</td>
</tr>
<tr>
<td>GMDH</td>
<td>M1</td>
<td>0.1079</td>
<td>0.5491</td>
<td>0.1251</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>0.1253</td>
<td>0.5907</td>
<td>0.1476</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>0.1025</td>
<td>0.6114</td>
<td>0.1199</td>
</tr>
<tr>
<td></td>
<td>M4</td>
<td>0.1233</td>
<td>0.6086</td>
<td>0.1411</td>
</tr>
<tr>
<td></td>
<td>M5</td>
<td>0.1233</td>
<td>0.6100</td>
<td>0.1411</td>
</tr>
<tr>
<td></td>
<td>M6</td>
<td>0.0955</td>
<td>0.6776</td>
<td>0.1176</td>
</tr>
<tr>
<td></td>
<td>M7</td>
<td>0.0973</td>
<td>0.6621</td>
<td>0.1164</td>
</tr>
<tr>
<td></td>
<td>M8</td>
<td>0.0956</td>
<td>0.6750</td>
<td>0.1164</td>
</tr>
<tr>
<td></td>
<td>M9</td>
<td>0.1065</td>
<td>0.5729</td>
<td>0.1224</td>
</tr>
<tr>
<td>LSSVM</td>
<td>M1</td>
<td>0.1053</td>
<td>0.5792</td>
<td>0.1196</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>0.1077</td>
<td>0.7217</td>
<td>0.1456</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>0.1035</td>
<td>0.0505</td>
<td>0.1216</td>
</tr>
<tr>
<td></td>
<td>M4</td>
<td>0.1253</td>
<td>0.6056</td>
<td>0.1453</td>
</tr>
<tr>
<td></td>
<td>M5</td>
<td>0.1208</td>
<td>0.6403</td>
<td>0.1442</td>
</tr>
<tr>
<td></td>
<td>M6</td>
<td>0.1108</td>
<td>0.6809</td>
<td>0.1055</td>
</tr>
<tr>
<td></td>
<td>M7</td>
<td>0.0997</td>
<td>0.6422</td>
<td>0.1163</td>
</tr>
<tr>
<td></td>
<td>M8</td>
<td>0.0961</td>
<td>0.6747</td>
<td>0.1126</td>
</tr>
<tr>
<td></td>
<td>M9</td>
<td>0.0938</td>
<td>0.6932</td>
<td>0.1119</td>
</tr>
<tr>
<td>GLSSVM</td>
<td>M1</td>
<td>0.0908</td>
<td>0.7107</td>
<td>0.1127</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>0.1010</td>
<td>0.7622</td>
<td>0.1456</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>0.0694</td>
<td>0.8441</td>
<td>0.1187</td>
</tr>
<tr>
<td></td>
<td>M4</td>
<td>0.1187</td>
<td>0.6056</td>
<td>0.1453</td>
</tr>
<tr>
<td></td>
<td>M5</td>
<td>0.1200</td>
<td>0.6386</td>
<td>0.1425</td>
</tr>
<tr>
<td></td>
<td>M6</td>
<td>0.1006</td>
<td>0.7408</td>
<td>0.1014</td>
</tr>
<tr>
<td></td>
<td>M7</td>
<td>0.0698</td>
<td>0.8432</td>
<td>0.1511</td>
</tr>
<tr>
<td></td>
<td>M8</td>
<td>0.0853</td>
<td>0.7544</td>
<td>0.1123</td>
</tr>
<tr>
<td></td>
<td>M9</td>
<td>0.0920</td>
<td>0.7076</td>
<td>0.1138</td>
</tr>
</tbody>
</table>

4.3 Fitting LSSVM to the data

The selection of appropriate input data sets is an important consideration in the LSSVM modelling. In the training and testing of the LSSVM model, the same input structures of the data set (M1–M9) have been used. The precision and convergence of LSSVM was affected by $(\gamma, \sigma^2)$. There is no structured way to choose the optimal parameters of LSSVM. In order to obtain the optimal model parameters of the LSSVM, a grid search algorithm was employed in the parameter space. In order to evaluate the performance of the proposed approach, 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 was considered. For each hyperparameter pair $(\gamma, \sigma^2)$ in the search space, a 5-fold cross validation on the training set is performed to predict the prediction error. The best fit model structure for each model is determined according to criteria of the performance evaluation. In the study, the LSSVM model was implemented with the software package LS-SVMlab1.5 (Pelckmans et al., 2003). As the LSSVM method is employed, a kernel function has to be selected from the qualified function. Previous works on the use of LSSVM in time series modeling and forecasting have demonstrated that RBF performs favourably (Liu and Wang, 2008; Yu et al., 2006; Gencoglu and Ulyar, 2009). Therefore, the RBF, which has a parameter $\gamma$ as in Eq. (14), is adopted in this work. Table 5 shows the results of the performance obtained during the training and testing period of the LSSVM approach.

As seen in Table 5, the LSSVM models are evaluated based on their performances in the training and testing sets. For the training phase of Selangor river, the best value of the RMSE and $R$ statistics are 0.0938 and 0.6932 (in M9), respectively. However, during the testing phase, the lowest value of the RMSE was 0.1055 (in M6) and the highest value of the $R$ was 0.6269 (in M8). On the other hand, for the Bernam river, the M9 model obtained the best RMSE and $R$ statistics, in the training and testing phase.
Table 6. Forecasting performance indices of models for Selangor and Bernam River.

<table>
<thead>
<tr>
<th>Model</th>
<th>Selangor River</th>
<th>Bernam River</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
<td>Testing</td>
</tr>
<tr>
<td>ARIMA</td>
<td>0.0914 0.7055</td>
<td>0.1226 0.5487</td>
</tr>
<tr>
<td>ANN</td>
<td>0.1065 0.5727</td>
<td>0.1092 0.6219</td>
</tr>
<tr>
<td>GMDH</td>
<td>0.1101 0.6733</td>
<td>0.1034 0.5850</td>
</tr>
<tr>
<td>LSSVM</td>
<td>0.0961 0.6747</td>
<td>0.1126 0.6269</td>
</tr>
<tr>
<td>GLSSVM</td>
<td>0.0853 0.7544</td>
<td>0.1123 0.6398</td>
</tr>
</tbody>
</table>

4.4 Fitting GMDH and GLSSVM with the data

In designing the GMDH and GLSSVM models, one must determine the following variables: the number of input nodes and layers. The selection of the number of input that corresponds to the number of variables plays an important role in many successful applications of GMDH.

GMDH works by building successive layers with complex connections that are created by using second-order polynomial functions. The first layer created is made by computing regressions of the input variables followed by the second layer that is created by computing regressions of the output value. Only the best variables are chosen from each layer and this process continues until the pre-specified selection criterion is found.

The proposed hybrid learning architecture is composed of two stages. In the first stage, GMDH is used to determine the useful inputs for LSSVM method. The estimated output values $x'$ is used as the feedback value which is combined with the input variables $\{x_1, x_2, \ldots, x_M\}$ in the next loop calculations. The second stage, the LSSVM mapping the combination inputs variables $\{x_1, x_2, \ldots, x_M, x'\}$ are used to seek optimal solutions for determining the best output for forecasting. To make the GMDH and GLSSVM models simple and reduce some of the computational burden, only nine input nodes (M1–M9) and five hidden layers ($k$) from 1 to 5 have been selected for this experiment.

In the LSSVM model, the parameter values for $\gamma$ and $\sigma^2$ need to be first specified at the beginning. Then, the parameters of the model are selected by grid searching with $\gamma$ within the range of 10 to 1000 and $\sigma^2$ within the range of 0.01 to 1.0. For each parameter pair $(\gamma, \sigma^2)$ in the search space, 5-fold cross validation of the training set is performed to predict the decision error. The performances of GMDH and GLSSVM for time series forecasting models are given in Table 5.

For Selangor river, in the training and testing phase, the best value of the RMSE and $R$ statistics for GMDH model were obtained using M6. In the training phase, GLSSVM model obtained the best RMSE and $R$ statistics of 0.0694 and 0.8441 (in M3) respectively. While in testing phase, the lowest value of the RMSE was 0.1014 (in M6) and the highest value of the $R$ was 0.6398 (in M8). However, in the training and testing phase for Bernam river, the best value of RMSE and $R$ for LSSM, GMDH and GLSSVM models were obtained by using M9.

The model that performs best during testing is chosen as the final model for forecasting the sixty monthly flows. As seen in Table 5, for Selangor river, the model input M8 gave the best performance for LSSVM and GLSSVM models, and M6 for the GMDH model. On the other hand, for Bernam river, the model input M9 gave the best performance for LSSVM, GMDH and GLSSVM models and hence, these model inputs have been chosen as the final input structures models.

4.5 Comparisons of forecasting models

To analyse these models further, the error statistics of the optimum ARIMA, ANN, GMDH, LSSVM and GLSSVM are compared. The performances of all the models for training and testing data set are in Table 6.

Comparing the performances of ARIMA, ANN, GMDH, LSSVM and GLSSVM models for in training of Selangor and Bernam rivers, the lowest RMSE and the largest $R$ were calculated for GLSSVM model respectively. For testing data, the best value of RMSE and $R$ were found for GLSSVM model. However, the lowest RMSE were observed for GMDH model for Selangor river and LSSVM model for Bernam river. From the Table 6, it is evident that the GLSSVM performed better than the ARIMA, ANN, GMDH and LSSVM models in the training and testing process.

Figures 9 and 10 show the comparison of time series and scatter plots of the results obtained from the five models and the actual data for the last sixty months during the testing stage for Selangor and Bernam rivers, respectively. All the five models gave close approximations of the actual observations, suggesting that these approaches are applicable for modeling river flow time series data. However, the tested line generated from GLSSVM is the closest to the actual value line in comparison to the tested line generated from other models. Similar to $R$ and fit line equation coefficients, the
Fig. 9. Comparison of the testing results of ARIMA, ANN, GMDH, LSSVM and GLSSVM models for Selangor river.


www.hydrol-earth-syst-sci.net/15/1835/2011/
Fig. 10. Comparison of the testing results of ARIMA, ANN, GMDH, LSSVM and GLSSVM models for Bernam river.

GLSSVM is slightly superior to the other models. The results obtained in this study indicate that the GLSSVM model is a powerful tool to model the river flow time series and can provide a better prediction performance as compared to the ARIMA, ANN, GMDH and LSSVM time series approach. The results indicate that the best performance can be obtained by the GLSSVM model and this is followed by LSSVM, GMDH, ANN and ARIMA models.

5 Conclusions

Monthly river flow estimation is vital in hydrological practices. There are plenty of models used to predict river flows. In this paper, we have demonstrated how the monthly river flow could be represented by a hybrid model combining the GMDH and LSSVM models. To illustrate the capability of the LSSVM model, Selangor and Bernam rivers, located in Selangor of Peninsular Malaysia were chosen as the case study. The river flow forecasting models having various input structures were trained and tested to investigate the applicability of GLSSVM compared with ARIMA, ANN, GMDH and LSSVM models. One of the most important issues in developing a satisfactory forecasting model such as ANN, GMDH, LSSVM and GLSSVM models is the selection of the input variables. Empirical results on the two data sets using five different models have clearly revealed the efficiency of the hybrid model. By using a evaluation of performance test, the input structure based on ARIMA model is decided as the optimal input factor. In terms of RMSE and $R$ values taken from both data sets, the hybrid model has the best in training. In testing, high correlation coefficient ($R$) was achieved by using the hybrid model for both data sets. However, the lowest value of RMSE were achieved using the GMDH for Selangor river and LSSVM for Bernam river. These results show that the hybrid model provides a robust modeling capable of capturing the nonlinear nature of the complex river flow time series and thus producing more accurate forecasts.

Acknowledgements. The authors would like to thank the Ministry of Science, Technology and Innovation (MOSTI), Malaysia for funding this research with grant number 79346 and Department of Irrigation and Drainage Malaysia for providing the data of river flow.

Edited by: E. Todini

References


