ISSN 1330-3651(Print), ISSN 1848-6339 (Online)
UDC/UDK 627.133:[519.233.5:004.032.26]

POTENTIAL OF SUPPORT-VECTOR REGRESSION FOR FORECASTING STREAM FLOW

Mohd Rashid Bin Mohd Radzi, Shahaboddin Shamshirband, Saeed Aghabozorgi, Sanjay Misra, Shatirah Akib, Miss Laiha Mat Kiah

Original scientific paper
Stream flow is an important input for hydrology studies because it determines the water variability and magnitude of a river. Water resources engineering always deals with historical data and tries to estimate the forecasting records in order to give a better prediction for any water resources applications, such as designing the water potential of hydroelectric dams, estimating low flow, and maintaining the water supply. This paper presents three soft-computing approaches for dealing with these issues, i.e. artificial neural networks (ANNs), adaptive-neuro-fuzzy inference systems (ANFISs), and support vector machines (SVMs). Telom River, located in the Cameron Highlands district of Pahang, Malaysia, was used in making the estimation. The Telom River's daily mean discharge records, such as rainfall and river-level data, were used for the period of March 1984 – January 2013 for training, testing, and validating the selected models. The SVM approach provided better results than ANFIS and ANNs in estimating the daily mean fluctuation of the stream's flow.

Keyword: *stream's flow, support vector machine, neuro-fuzzy, neural networks, forecast*

Potencijal support-vector regresije u prognoziranju vodotoka

Izvorni znanstveni članak
Vodotok je važan za hidrološko proučavanje zato što određuje varijabilnost vode i magnitudu rijeke. Inženjerstvo vodnih resursa uvijek se bavi povijesnim podacima i pokušava procijeniti prognostičke podatke kako bi se osiguralo bolje predviđanje za primjenu kod bilo kojeg vodnog resursa, na pr. projektiranja vodnog potencijala brane hidroelektrana, procjene niskog protoka, i održavanja zalihe vode. U radu se predstavljaju tri računalna programa za primjenu kod rješavanja ovakvih sadržaja, tj. umjetne neuronske mreže - artificial neural networks (ANNs), prilagodljivi sustavi neuro-neizrazitog zaključivanja - adaptive-neuro-fuzzy inference systems (ANFISs), i support vector machines (SVMs). Za stvaranje procjene korištena je Rijeka Telom, smještena u Cameron Highlands distriktu Pahanga, Malaysia. Podaci o dnevnom prosječnom protoku rijeke Telom, kao što su količina padavina i podaci o vodostaju, koristili su se za period od ožujka 1984. do siječnja 2013. za podučavanje, ispitivanje i ocjenjivanje izabranih modela. SVM pristup je dao bolje rezultate nego ANFIS i ANNs kod procjenjivanja dnevne prosječne fluktuacije vodotoka.

Ključne riječi: *vodotok, support vector stroj, neuro-neizraziti, neuronske mreže, prognoza*

1 Introduction

Forecasting the flow of a stream is important in water resource engineering because of its major input for river engineering applications, e.g. forecasting flooding, classifying sediments, scheduling irrigation, controlling procedures for reservoirs, and generating hydropower formation [1]. Also, variations in river levels result from many ecological factors, such as rainfall, direct or indirect discharge of adjacent watersheds, evaporation of free water bodies, temperature of the water and air, and the interaction between rivers and low-lying aquifers. Although it is possible to determine complicated patterns using the aforementioned parameters, it is desirable that the models that simulate changes in the flow of streams based on historical flow rate data be available for us in such investigations, as well as for other applications [2].

In recent times, the use of Soft Computing (SC) methods has become an accepted approach for modelling the complicated, non-linear phenomena associated with the hydrology of water resource systems. In this circumstance, the methods that have been used extensively are artificial neural networks (ANNs), adaptive neuro-fuzzy inference systems (ANFISs), and the support vector machine (SVM). In the current investigation, we used the function of ANNs for water resources modelling, including predicting water quality [3], forecasting the salinity of groundwater [4], and forecasting the daily levels of lakes [1]. Sattari et al. [5] used ANNs for estimating reservoir inflow and operations. Jain et al. [6] used ANNs for real-time forecasting of waves. Filippo et al. [7] used ANNs to improve the estimation of the fluctuations of the sea's levels. Kalteh et al. [8] used an ANN and Support Vector Regression (SVR) for forecasting river flows on a monthly basis. Thus, several approaches have been anticipated to advance the back-propagation algorithm of modification of the standard algorithms [9] and the adaptive learning rate on classification problems [10]. In this study, the Levenberg–Marquardt algorithm (a learning algorithm) based on a second-order error, back-propagation algorithm was used to prepare the neural networks.

The combination of a fuzzy inference system and an adaptive neural network is described as an adaptive neuro-fuzzy inference system (ANFIS). ANFIS approximates a real continuous role on the compact set to a degree of precision [11]. It recognizes a set of factors through a hybrid learning rule that integrates back propagation gradient descent error digestion and a least-squared error method.

Mamdani [12] and Sugeno [13] identified two approaches that are used for fuzzy inference systems. Sugeno's approach to the neuro-fuzzy model was used in this study to attain valuable information about an output based on the input [13]. For this purpose, ANFIS uses records of river levels and rainfall as input variables, and the flow rates of streams are the output.

Rath et al. [14] applied a hierarchical, neuro-fuzzy model for the real-time prediction of floods. ANFIS also has been used for predicting short-term operational water levels [15]. The derivation of classification and regression methods from statistical learning theory is known as the support vector machine (SVM) technique [16]. The principles of optimum division classes in the SVM technique have been used in SVM classification approaches. The method is used to select a linear classification from an unlimited number. It selects the one that reduces the generalization error and at least one upper bound of the error, derived from essential risk minimization [16]. Therefore, the currently-selected hyper-plane would be the one that leaves a maximum outlying group between two classes, where margin is distinct as the sum of the distances of the hyper-plane from the closest argument of the two classes.

If the two classes of non-detachable SVM are trying to identify a hyper-plane that maximizes margins and simultaneously reduces the quantity proportional to the number of classification mistakes, the trade-off between margins and classification of failure is controlled by the positive constants that were selected in advance. This method of designing an SVM may be extended to enable non-linear decidedness on the surface. This may be obtained by projecting the initial number of variables in a higher-dimensional feature space and developing a linear classification problem in the feature space [16]. Vapnik et al. [17] anticipated ε-support vector regression (SVR) through the alternative ε-insensitive loss function. The objective of SVR is to determine the function with the most ε-deviations from the actual target vectors for all trained information.

SVR requires that less user-defined parameters be set. In addition to selection of the kernel, SVR requires the establishment of core parameters. Furthermore, the optimal values of the regularization factor C and the size of the error in the sensitive area ε must be determined. The selection of these settings controls the complexity and forecasting. One of the main advantages of SVR is the algorithm that includes the resolution of linear-constraint, square-planning work, leading to the unique, optimum, comprehensive solution for SVM of the prediction of daily rainfall[1]. Behzad et al.[18] generalized the performance of support vector machines in modeling runoff. In this research, we used SVR to predict the flow rate of the stream. This study examined the ability of SVM, ANFIS, and ANN to predict the flow rate of a stream.

The remaining sections of this paper are structured as follows. Section 2 describes the dataset that was used and the methods that were applied on the dataset, i.e., SVM, ANFIS, and ANN. Section 3 provides the applied statistical indicators for the analysis of patterns, and section 4 presents a discussion of the results that were obtained. Our conclusions are presented in section 5.

2 Materials and methods

2.1 The data that were used

In this paper, daily records, such as mean water level and total rainfall in the Cameron Highlands in Pahang state, Malaysia, were used. Sungai Telom at Bt. 49 river is located in the high altitude area of the Cameron Highlands district. The altitude ranges from 1200 m to 1500 m above sea level. Brinchang, the closest town in the Cameron Highlands is at an altitude of 1450 m above sea level. The average annual rainfall for Cameron Highlands is about 3000 mm/m^2 per year. The data sample consisted of the records of daily stream flow for 29 years (from 1 March 1984 to 30 January 2013). For each model, 70 % of the data were used to train the samples; the other 30 % of the data were used to test the samples. Table 1 indicates the statistical parameters of the data that were used during the investigation period.

Table 1 Statistical parameters that apply to the dataset during the period of study

Model	% of patterns		Statistical parameters of recorded data							
	Rainfall	Water level	Rainfall mm/(m^2·d)				Water level mm/(m^2·a)			
ANN/ANFIS/SVM			*Xmax*	*Xmin*	*Xmean*	*Sx*	*Xmax*	*Xmin*	*Xmean*	*Sx*
	Train	75 %	89,5	0	6,98	12,3	1349	476	2037	150,8
	Test	25 %	84	0	8,59	16,6	1529	553	58301	229,6

d – day, a – year

Fig. 1 shows the location of the station where the stream's flow was measured and the station where the rainfall was measured in the Cameron Highlands District of Pahang, Malaysia. For modelling purposes, 29 years of records of the daily mean stream flow and daily total rainfall data were used.

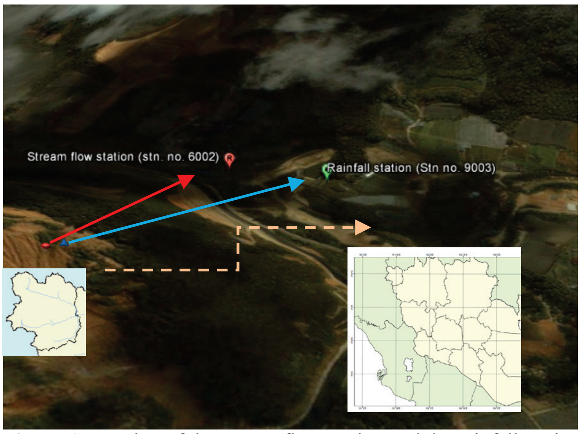

Figure 1 Location of the stream flow station and the rainfall station

2.2 Artificial neural networks

The goal of the training network was to adjust the weight and biases in the network so that specific performance measures could be optimized. This involved minimizing the sum of the squares of the differences between the target values and the network's output values. Training the network took place over and over again to feed the input-output patterns in the network. The most commonly-used formation algorithm of multi-tiered, feed-forward networks is the back-propagation method [19]. This method appraises the network's weight and bias in the orientation in which a performance feature is reduced most quickly for negative gradients. Despite the overall success of the back-propagation method of training a neural network, it has some disadvantages, such as the slow pace of the rapprochement, the sensitivity of the local minimum, and the difficulty of the regulation and training parameter. Thus, several methods have been proposed to improve the back-propagation algorithm, such as changing the standard algorithm [9] and using an adaptive learning rate on classification problems [10].

In this study, a learning algorithm (the Levenberg–Marquardt algorithm) based on a second-order error, back-propagation algorithm was used to train the neural network. The method is probably the most widely-used approach for dealing with the neural network's over-fitting problem because it is simple to implement and it does not require extra computations[20]. The Levenberg – Marquardt algorithm uses an early stop, i.e. the criterion in which the available data are separated into three sub-sets, i.e. training, validating, and testing sets. The training sets are used for calculating a gradient and updating the weight and biases of the system. The error in validating sets was monitored during the training process. This error typically declines during the first stage of the training, as is the training specified failure. As soon as the validation error starts to increase, the training is stopped to overcome the over-fitting problem, and the weights and biases at the least of the validation error are returned. The test set is for validating the weights and biases to verify the capability of the stopping criterion and to estimate the generalization ability of the trained network. This performance feature influences the network to have less weight and fewer biases, which forces the response of the network to be smooth and less likely to have over-fitting.

Neural network training could be made more effective if some pre-process steps were performed in the network's input and target. Before training, it is regularly suitable to scale off the input and output so that they still fall within the specified ranges. To determine the structure of the network, we set feature values of the Back Propagation (BP) neural network signal, rainfall, and river level as input parameters of the BP neural network. Therefore, there are two nodes in the input layer. The flow rate of the stream has one value and one layer for the output layer. During the design of the BP neural network, the important aspect that was considered was defining the number of nodes in the hidden layer. If some of hidden nodes were not well configured, the hidden nodes could cause the network to over-fit the data. The choice of the number of hidden devices has no theoretical foundation and is determined on the basis of experience and usage [19]. After completing the learning, the output of the network was processed in a manner similar to converting it to a physical quantity.

2.3 Adaptive neuro-fuzzy inference system

The primary step is to take a Nuero-Fuzzy Inference System (FIS) with two input variables, x and y, and an output variable, f. The first-order Sugeno's fuzzy model, a typical rule set with two fuzzy IF–THEN rules, can be given as:

$$\text{Rule 1: If } x \text{ is } A_1 \text{and } y \text{ is } B_1, \text{then } f_1 = P_1x + q_1y + r_1$$
$$\text{Rule 2: If } x \text{ is } A_2 \text{and } y \text{ is } B_2, \text{then } f_2 = P_2x + q_2y + r_2,$$

where A_1, A_2 and B_1, B_2 are the fuzzy value for inputs x and y, respectively, and p_1, q_1, r_1, and p_2, q_2, r_2 are the parameters of the output function. In our scheme, the output f is a weighted average of the single rule output and is, in itself, a crisp value. The node function is described in the following.

The output from the i^{th} node of level l is indicated as O_{1},i. Each node i in layer 1 is an adaptive node with node $O_1,i = A_i(x)$, for i = 1, 2, or $O_1,i = B_i - 2(y)$, for i = 3, 4, where x (or y) is input for the i^{th} node and A_i (or $B_i - 2$) is a linguistic variable (such as "low" or "high") associated with this node. The membership functions (MFs) for A and B are generally described by generalized bell functions, e.g.:

$$A_i(x) = \frac{1}{1+[(x-c_i)/a_i]^{2b_i}},$$

where $\{a_i, b_i, c_i\}$ is the parameter set. In fact, any continuous and piecewise differentiable functions, such as commonly-used, triangular-shaped, membership functions (as qualified candidates for node functions in this layer), often are chosen in practical matters. Similar to the number of hidden nodes in ANNs, there is no general rule to determine the number of MFs, so they must be defined in an iterative manner. Parameters in this layer are referred to as the principle parameters. The output of this layer is the composition value of the principle of the section. Layer 2 consists of the labelled nodes, which multiply incoming signals and send out the product. For example,

$$O_{2,i} = W_i = A_i(x)B_i(y), i = 1,2$$

The output of each node stands for the strength of a rule. The nodes labelled N calculate the ratio of the i^{th} rule's firing strength to the sum of all rules' firing strengths in layer 3,

$$O_{3,i} = \varpi_i = w_i/w_1 + w_2.$$

The outputs of this layer are called normalized firing strengths. The nodes of layer 4 are adaptive with node functions:

$$O_{4,i} = \varpi_i f_i = \varpi_i(P_i x + q_i y + r_i),$$

where ϖ_i is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters of this layer are referred to as resultant parameters. The sole fixed node of layer 5, labelled Σ, computes the final output as the sum of all inward signals:

$$O_{5,i} = \sum_{i=1} \varpi_i f = \frac{\sum_i w_i f_i}{\sum_i w_i}.$$

Thus, the adaptive network that is equivalent to Sugeno - the first-order, fuzzy reasoning scheme (FIS) - is created. In our case study, the FIS was generated based on subtractive clustering. Subtractive clustering [21] is a fast, one-pass algorithm for approximating the number of clusters and the centers of the clusters in a dataset. Thus, we used subtractive clustering to identify natural groupings of data from a large dataset to produce a concise representation of a system's behaviour in order to generate a Sugeno-type, fuzzy inference system that best models the data's behavior using a minimum number of rules. The proposed ANFIS has two inputs with one output. Each input performs three Gaussian membership functions (low, medium and high) in the second layers, and four rules are created based on subtractive clustering. The output from each rule is the linear combination of an input variable and a continuous period, and the final result is a weighted average output for each rule.

2.4 Support vector machines (SVM)

Cortes and Vapnik [16] stated that support vector machinery classified the regression methods that were resulting in statistical learning theories. SVM classification postulates the principle of optimal separation between categories. If a class is detachable, the method selected from the endless number of linear classifier should be the one that minimizes the generalized errors, or at least one upper bound for the error, obtained from the structural risk reduction [16]. Therefore, the selected hyper-plane would be the one that left the highest edge between two classes, where the border is defined as the sum of the distances of the hyper-plane from the nearest points on two categories [16].

If the two classes of non-detachable SVM trying to identify a hyper-plane to maximize the margins while simultaneously reducing the quantity in proportion to the number of classification mistakes, the compromise between margins and classification errors is regulated by positive constants that must be selected in advance. This method of designing an SVM may be extended to allow the non-linear decision surface. This can be obtained by projection of the initial set of variables in a higher-dimensional function space and formulation of a linear grading issue, i.e. the function room [16].

The ε-support vector regression (SVR) was introduced as an alternative to the ε-insensitive loss function [16]. The objective of the SVR is to find the function with the most ε deviations from the actual destination vector for all received training information, and it must be as flat as possible [17].

SVR requires using fewer parameters defined by the user for setting up of kernel-specific parameters. Further, the optimal values of the legalization argument C and size errors in sensitive area ε must be determined. The selection of these settings controls the complexity of the prediction. One of the main advantages of SVR is the algorithm. It includes the resolution of the quadratic programming function, a work function that leads to the unique, optimum, and comprehensive solution.

2.5 Analysis of the results

Several measures were used to confirm the validity of the proposed ANN, ANFIS, and SVM models. The root mean squared error (*RMSE*) was used to evaluate the differences between the expected and actual values. The predictive accuracy (A) was calculated to determine the correctness of the forecasting models and of the coefficient of (R). The parameter was calculated as indicated in Tab. 2.

Table 2 Performance criteria

Criteria	Calculation
Root mean squared error (*RMSE*)	$RMSE = \sqrt{\frac{1}{N_T}\sum_{i=1}^{N_t}(d_i - y_i)^2}$
Correlation coefficient (R)	$R = \frac{\sum d_i y_i - \left(\sum d_i y_i / N\right)}{\sqrt{\left(\sum d_i^2 - (\sum d_i)^2/N\right) / \left(\sum y_i^2 - (\sum y_i)^2/N\right)}}$
Accuracy (A)	$A = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{\lvert v_a - v_p \rvert}{v_a}\right) \times 100\,\%$

where n is the total number of test data, d_i is are the experimental values, and y_i are forecasted values. 30 % of the testing dataset was used to validate the proposed model.

3 The data, the experiment, and the results

The dataset used in this study was provided by Tenaga Nasional Berhad (TNB). This organization collected data at one stream-flow station and at one rainfall station using an automatic-tipping-bucket for logging data. Both stations are located in the same catchment area. The automatic rainfall recorder at Alor Masuk Sg. Telom (station no. 9003) is located at coordinates 04°32'36" N, 101°25'25" E. The available daily total data recorded from 1st March 1984 until 30th January 2013 were used for our simulation. Fig. 2 shows typical rainfall recording equipment.

Figure 2 Tipping-bucket automatic recording equipment

The stream flow data used for this study came from Sg. Telom at Batu 49 river (station no. 6002). The stream-flow station is located at coordinates 04°32'37" N, 101° 25'29" E. Fig. 3 shows the stream-flow station installed in Cameron Highlands.

Figure 3 River discharge and water level automatic recording station

The dataset contained 29 years' worth of stream-flow and rainfall data collected by the two stations. The variables associated with the three discrete values are shown in Tab. 3.

Table 3 Input and output parameters

Factor	Variable	Range of values
Inputs	Rainfall (x_1) mm/d Water level (x_2) cm/a	0 ÷ 90; 512 ÷ 1529
Outputs	Stream flow rate (y_1) m^3/s	2,2 ÷ 16,9

Fig. 4 indicated the stream-flow discharge for Sg. Telom at Bt. 49. The fluctuating daily means of the discharge show the variation in the flows at Sg. Telom.

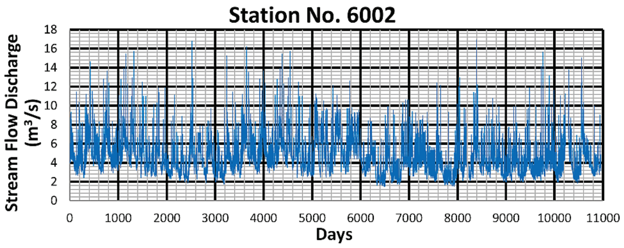

Figure 4 Daily means of the discharges from station no. 6002

3.1 ANN models

For our NN rating patterns, we considered three variables, i.e., rainfall and water (river) levels as inputs and the stream's flow rate as an output. This choice of input parameters did not increase the size of the network, which decreased the amount of data required to estimate connection weights efficiently and increased the processing speed.

In the appraisal, we standardized the original sets of dataset through normalization into training and test sets. Each training and present sample pair had approximately 70 % of the original 300 experiments in the training dataset. The remaining 30 % of the experiments were used to create a test dataset. We used the MATLAB program's random number generator to create these training and testing datasets.

The network model that we developed was based on a three-layer, feed-forward neural network. The input layer corresponded to the two selected input parameters. The output layer corresponded to the one output, i.e., the flow rate of the stream. These networks were trained using the Levenberg–Marquardt training algorithm. Here, the neural network trained with Levenberg–Marquardt algorithm was termed as Levenberg–Marquardt neural network (LMNN). The performances of these networks were measured by computing the root mean square error (*RMSE*), and the correlation coefficient (R) was used in the study of the training and test subsets of data. The LMNN model was optimized at 10 numbers of hidden nodes in the hidden layer and 100 training iterations. The best validation performance was 0,1055 at epoch 27, and the learning rate was 0,01. Tab. 4 presents the results of the LMNN models during the training.

Table 4 Structure of the ANN model

Experiments	ANN Structure (number of input, hidden, output nodes)	R^2	*RMSE* (m)
1	2, 3, 1	1,80	0,18
2	2, 6, 1	1,69	0,12
3	2, 10, 1	1,32	0,11

Tab. 4 shows that the double input ANNs, i.e. daily mean rainfall and daily mean water level data with a high number of neurons in the hidden layer (10), produced better results than the other ANNs structure combination interval. The accuracy of the experiment model decreased when the prediction intervals increased from –1 to 3. R^2 decreased from 1,80 to 1,32, and *RMSE* decreased from 0,18 to 0,11 for prediction intervals, respectively. Fig. 5 shows the error graph of the ANN model in the training epoch.

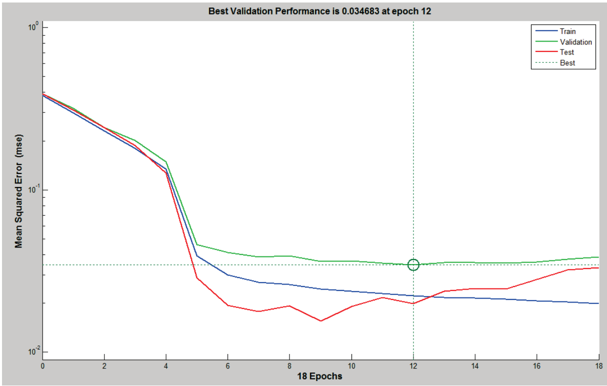

Figure 5 Performance validation of ANN

3.2 ANFIS models

For a particular input–output dataset, e.g., to estimate stream flow rate using the historical river-level values, various identification methods of the Sugeno model could be applied (i.e. subtractive-clustering, grid-partitioning, and Gaussian– kernel clustering methods). However, the identification-type methods affected the results significantly [22]. Therefore, in this paper, the normally-used, grid-partitioning classification method was used to construct the neuro-fuzzy models. The optimal structure of the model i.e. the number of inputs (2); hidden nodes (10); and output nodes, was determined in the preceding section (by the ANN template). A sensitivity analysis was conducted on the impact of the type of MFs on the results that were obtained. Several membership functions were examined, and they are shown in Tab. 5 for the estimation of the stream's flow. The second column in Tab. 5 indicates the number of MFs for each input variable. Iterative processes were used to determine the number of MFs of the ANFIS models, since there were no specific rules for doing that. In order to save time and reduce the calculations required, the number of MFs in the model should be kept as small as possible [1]. The applied statistical criteria with various MFs for ANFIS model during the test period showed that the Gaussian membership function gave better results than the other MFs that were investigated.

Table 5 Optimum values of the statistical criteria for the different categories of ANFIS MFs

Type of MFs	ANFIS structure	*RMSE* (m)	R^2
Triangular	3; 3	0,073	0,077
Trapezoidal	3; 3	0,567	0,567
Gaussian	3; 3	0,157	0,157
Two Gaussian	3; 3	0,533	0,533
Generalized bell	3; 3	0,061	0,061
Pi-shaped	3; 3	0,197	0,197

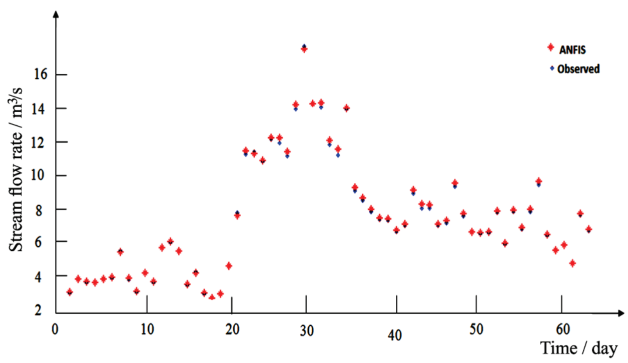

Figure 6 Observed and simulated stream flow rates of optimal ANFIS models during the test period for stream flow rate

Testing statistics for each ANFIS model (applying the two-Gaussian membership function) are given in Tab. 5. Fig. 6 shows the observed and simulated stream flow rates (by using double-input ANFIS models) during the testing period. Similar to the ANN model, the double-input structure of the ANFIS model, shown in column 2 of Tab. 5 with three membership functions, provided the best results of all of the combinations. It also was observed that increasing prediction interval decreased the model's accuracy. The comparison of Tab. 2 and Tab. 4 shows that ANNs model was slightly better than the ANFIS model. Both approaches can be considered as alternative tools for estimating the variations of stream flow because their differences were minimal.

3.3 Details of the SVM algorithm

A kernel function can be utilized to form a qualified function that uses SVM. Guo et al. [23] pointed out that SVM shows high performance in its accuracy of the prediction of the stream's flow. Asefa et al. [24] proposed different kernels in SVR for rainfall–runoff modelling and validated that the radial basis function (RBF) outperformed other kernel functions. Furthermore, researchers have demonstrated that SVM can be used in predicting hydrological information and pointed out the positive performance of the RBF [25, 26]. Hence, the RBF was used as the kernel function for prediction of discharge in this research study. Three parameters are associated with RBF kernels, i.e. C, e and r. The accuracy of an SVM model is dependent largely on the selection of the model's parameters. However, structured methods for selecting parameters are lacking. Consequently, some kind of model parameter calibration should be made.

To solve the problem of choosing parameters in SVM, the support regression machine used to choose a kernel requires setting up the kernel's specific parameters, optimum values of the regularization parameter, C, and the size of the error-insensitive zone, e. In our scheme, a default value of e, i.e. 0,1, was found to work well. To select user-defined parameters, i.e. C, d, and γ, a large number of trials were conducted using different combinations of C and d for the polynomial kernel and different combinations of C and γ for the radial-basis-function kernel. Likewise, many trials also were conducted to determine the user-defined parameters (i.e. the number of hidden layers and nodes in the hidden layer, learning rate, momentum, and the number of iterations) for a feed-forward neural network, as well as the user-defined factors (i.e. number of rules, type of membership function, and number of epochs). The results indicated that one hidden layer works well with these data. Tab. 6 provides the optimal values of the user-defined parameters for this dataset with polynomial and RBF kernel-based SVM, ANFIS parameters, and a back-propagation neural network. For reasonable appraisal of outcomes with both RBF and polynomial kernels, the same value of parameter (e) was used with SVR. In this study, SVM_poly and SVM_rbf signify polynomial and RBF kernel-based SVM, whereas NN denotes the feed-forward neural network and the ANFIS-based grid-portioning algorithm. SVM, the back-propagation neural network, and neuro-fuzzy system are implemented using MATLAB 2012.

Table 6 User-defined parameters

Support vector regression	RBF kernel			Polynomial kernel		
	C	γ	e	C	d	e
	1	0,5	0,1	0,25	2	0,1
ANFIS	Number of rules: 10; membership function: generalized bell; number of iterations: 1500; identification method: grid-partitioning					
NN	Learning rate = 0,2; momentum = 0,1; hidden nodes = 3, 6, 10; number of iterations = 1500					

4 Results

To evaluate the proposed method, the training and testing sets that were used for the models were used for our models, and two standard, quantitative, statistical measures of performance were used to assess the performances of the various models that were developed. Tab. 7 presents the results of the Cameron Highland study sites in terms of various performance statistics.

Table 7 Forecasting performance indices of models for Sg. T River

Model	Result in training phase	
	R^2	*RMSE*
Observed	-	-
NN	1,320	0,110
ANFIS	0,561	0,307
SVM-rb	0,859	1,555
SVM-poly	0,995	0,302

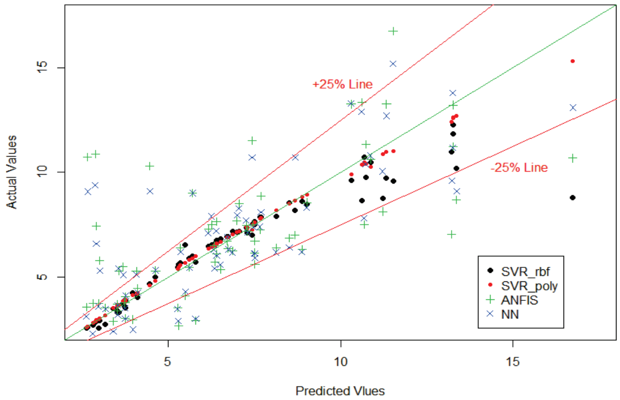

Figure 7 Forecasted and observed flow during training period by ANN, ANFIS, SVM_rbf, and SVM_poly for the Sg. T River

Tab. 7 shows that the various soft-computing methods had good performance during training, and they represented the real data that were observed for the Sungai Telom (Sg. T) river quite closely in terms of all of the standard statistical measures. For Sungai Telom (Sg. T), in the training phase, the ANFIS model acquired the best R, with a statistical value of 0,561; the SVM_poly model achieved the superlative *RMSE* statistics of 0,302.

Analyzing the outcomes throughout the forecast indicated that the SVM_poly model outperformed the other models. *RMSE* evaluates the residual between observed and forecasted flow, and *R* evaluates the linear correlation between the observed and computed flow. According to Fig. 7 and Tab. 7, the best performance of all the soft-computing methods used in this research work was different in terms of the different kinds of SVM and different statistical measures.

5 Conclusions

In this paper, we considered the potential of SVM in forecasting the stream's flow and compared its performance to those of two empirical relationships, i.e., NN and neuro-fuzzy system. The main conclusion from this research is that the polynomial and radial kernel-based SVM methods worked well in forecasting the stream's flow in appraisal to the two empirical relations. The results also indicated that the poly-based SVM performed better than the RBF-based SVM, NN, and ANFIS-based approaches. However, the RBF-based approach performed well in comparison to the NN and ANFIS-based approaches with this dataset. The performance of the SVM model was superior to all of other models that were used in that it provided explicit expressions for the studied phenomena. The difference between the SVR models (SVM_poly and SVM_rbf.) and the traditional artificial intelligence methods, i.e. neural networks and adaptive neuro-fuzzy system was significant.

Acknowledgement

The authors would like to acknowledge the Sponsoring Research Unit, Institute of Research and Consultancy Management, University of Malaya and the Faculty of Engineering, University Malaya for their financial support for this research.

6 Reference

[1] Kisi, O.; Shiri, J.; Nikoofar, B. Forecasting daily lake levels using artificial intelligence approaches. // Computers & Geosciences. 41, (2012), pp. 169-180.
[2] Ghorbani, M. A.; Kisi, O.; Aalinezhad, M. A probe into the chaotic nature of daily streamflow time series by correlation dimension and largest Lyapunov methods. // Applied Mathematical Modelling. 34, (2010), pp. 4050-4057.
[3] Ömer Faruk, D. A hybrid neural network and ARIMA model for water quality time series prediction. // Engineering Applications of Artificial Intelligence. 23, (2010), pp. 586-594.
[4] Banerjee, P.; Singh, V. S.; Chatttopadhyay, K.; Chandra, P. C.; Singh, B. Artificial neural network model as a potential alternative for groundwater salinity forecasting. // Journal of Hydrology. 398, (2011), pp. 212-220.
[5] Taghi Sattari, M.; Yurekli, K.; Pal, M. Performance evaluation of artificial neural network approaches in forecasting reservoir inflow. // Applied Mathematical Modelling. 36, (2012), pp. 2649-2657.
[6] Jain, P.; Deo, M. C.; Latha, G.; Rajendran, V. Real time wave forecasting using wind time history and numerical model. // Ocean Modelling. 36, (2011), pp. 26-39.
[7] Filippo, A.; Rebelo Torres Jr, A.; Kjerfve, B.; Monat, A. Application of Artificial Neural Network (ANN) to improve forecasting of sea level. // Ocean & Coastal Management. 55, (2012), pp. 101-110.
[8] Kalteh, A. M. Monthly river flow forecasting using artificial neural network and support vector regression models coupled with wavelet transform. // Computers & Geosciences. 54, (2013), pp. 1-8.
[9] Sperduti, A.; Starita, A. Speed up learning and network optimization with extended back propagation. // Neural networks. 6, (1993), pp. 365-383.
[10] Hamid, N. A.; Nawi, N. M.; Ghazali, R.; Salleh, M. N. M. Accelerating Learning Performance of Back Propagation Algorithm by Using Adaptive Gain Together with Adaptive Momentum and Adaptive Learning Rate on Classification Problems. // Ubiquitous Computing and Multimedia Applications, Springer, 2011, pp. 559-570.
[11] Jang, J. S. R. ANFIS: adaptive-network-based fuzzy inference system. // Systems, Man and Cybernetics, IEEE Transactions on. 23, (1993), pp. 665-685.
[12] Mamdani, E. H.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. // International Journal of Man-Machine Studies. 7, (1975), pp. 1-13.
[13] Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. // Systems, Man

and Cybernetics, IEEE Transactions on. SMC-15, (1985), pp. 116-132.
[14] Rath, S.; Nayak, P. C.; Chatterjee, C. Hierarchical neurofuzzy model for real-time flood forecasting. // International Journal of River Basin Management. (2013), pp. 1-16.
[15] Shiri, J.; Makarynskyy, O.; Kisi, O.; Dierickx, W.; Fard, A. Prediction of Short-Term Operational Water Levels Using an Adaptive Neuro-Fuzzy Inference System. // Journal of Waterway, Port, Coastal, and Ocean Engineering. 137, (2011), pp. 344-354.
[16] Cortes, C.; Vapnik, V. Support-vector networks. // Machine Learning. 20, (1995), pp. 273-297.
[17] Smola, A.; Schölkopf, B. A tutorial on support vector regression. // Statistics and Computing. 14, (2004), pp. 199-222.
[18] Behzad, M.; Asghari, K.; Eazi, M.; Palhang, M. Generalization performance of support vector machines and neural networks in runoff modeling. // Expert Systems with Applications. 36, (2009), pp. 7624-7629.
[19] Wang, P.; Zhu, L.; Zhu, Q.; Ji, X.; Wang, H.; Tian, G.; Yao, E. An application of back propagation neural network for the steel stress detection based on Barkhausen noise theory. // NDT & E International. 55, (2013), pp. 9-14.
[20] Hagan, M. T.; Menhaj, M. B. Training feedforward networks with the Marquardt algorithm. // Neural Networks, IEEE Transactions on. 5, (1994), pp. 989-993.
[21] Chiu, S. L. Fuzzy model identification based on cluster estimation. // Journal of intelligent and Fuzzy systems. 2, (1994), pp. 267-278.
[22] Sanikhani, H.; Kisi, O. River Flow Estimation and Forecasting by Using Two Different Adaptive Neuro-Fuzzy Approaches. // Water Resour Manage. 26, (2012), pp. 1715-1729.
[23] Guo, J.; Zhou, J.; Qin, H.; Zou, Q.; Li, Q. Monthly streamflow forecasting based on improved support vector machine model. // Expert Systems with Applications. 38, (2011), pp. 13073-13081.
[24] Asefa, T.; Kemblowski, M.; McKee, M.; Khalil, A. Multi-time scale stream flow predictions: The support vector machines approach. // Journal of Hydrology. 318, (2006), pp. 7-16.
[25] Wang, W.-C.; Chau, K.-W.; Cheng, C.-T.; Qiu, L. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. // Journal of Hydrology. 374, (2009), pp. 294-306.
[26] Ch, S.; Anand, N.; Panigrahi, B. K.; Mathur, S. Streamflow forecasting by SVM with quantum behaved particle swarm optimization. // Neurocomputing. 101, (2013), pp. 18-23.
[27] Pal, M.; Singh, N. K.; Tiwari, N. K. Support vector regression based modeling of pier scour using field data. // Engineering Applications of Artificial Intelligence. 24, (2011), pp. 911-916.

Authors' addresses

Mohd Rashid Bin Mohd Radzi
Department of Civil Engineering, Faculty of Engineering,
University of Malaya, Kuala Lumpur, Malaysia

Shahaboddin Shamshirband
Department of Computer Science, Chalous Branch, Islamic Azad University (IAU), 46615-397 Chalous, Mazandaran, Iran
E-mail: shahab1396@gmail.com

Saeed Aghabozorgi, Dr.
Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

Sanjay Misra, Prof.
Department of Computer Engineering,
Atilim University, 06836-Incek, Ankara Turkey

Shatirah Akib, Dr.
Department of Civil Engineering, Faculty of Engineering,
University of Malaya, Kuala Lumpur, Malaysia
E-mail: shatirah@um.edu.my

Miss Laiha Mat Kiah, Associate Prof.
Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia