RAINFALL–RUNOFF MODELING USING SUPPORT VECTOR REGRESSION AND ARTIFICIAL NEURAL NETWORKS

D. BOTSIS1[1], P. LATINOPOULOS[2] and K. DIAMANTARAS[3]

[1&2] Department of Civil Engineering, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece, [3] Department of Informatics, Alexander Technological Education Institute of Thessaloniki, GR-57400 Sindos, Greece
e-mail: jimbotsis@civil.auth.gr

EXTENDED ABSTRACT

In this study a comparison is performed between support vector regression (SVR) and multilayer feed-forward neural network (MFNN) models with respect to their forecasting capabilities. The two models have been designed to estimate the relationship between rainfall and runoff, which describes one of the most complex hydrologic phenomena. This relationship is nonlinear and it is suggested that nonlinear models like SVR and MFNN may have notable advantages in estimating rainfall–runoff mapping. As a matter of fact, studies in the literature have reported that SVR and artificial neural networks are very powerful in terms of nonlinear mapping. The daily rainfall and streamflow data of a mountainous watershed were used as a case study for developing the rainfall-runoff models of this investigation. The watershed is located in northern California in a region with the same geographic latitude as Northern Greece. The dataset of rainfall and streamflow is divided into three parts; the first is used for training, the second for testing and the third for validating the models. The performance of the SVR model depends on the choice of the kernel function and the model parameters. In the present study, the polynomial kernel, the Gaussian (or RBF) kernel and the tanh kernel were used. Each kernel function has one parameter, which could influence the behavior of the SVR model. The SVR model is implemented using the SVMlight package, a very efficient tool in solving large-scale problems. The MFNN neural network employed in this investigation is a Multilayer Perceptron (MLP) with one hidden layer, which is trained with the Back-Propagation algorithm. More specifically, the Levenberg-Marquardt variation was selected as the training algorithm, because of its high speed and efficiency. The cross-validation method was used to choose the appropriate number of hidden units in the neural network. The performance and reliability of SVR and MFNN were evaluated through three different criteria (correlation coefficient, mean square error and root mean square error). Each criterion is represented by an efficiency indicator estimated from the comparison of predicted values and the measured targets that have been initially placed. SVR has been found effective in rainfall-runoff simulation by many researchers in a variety of applications. The results of the study demonstrate that the SVR model is more effective than MLP in simulating the rainfall–runoff relationship and the SVR algorithms have better generalization capability than the conventional artificial neural networks. The generalization capability of any proposed model is a critical issue. In general, making accurate predictions for events, which take place in watersheds others than the ones where the models are trained on, is still an open problem.

Keywords: support vector regression, artificial neural networks, rainfall, runoff, watershed

1. INTRODUCTION

Modern hydrology is concerned with the distribution of water on the surface of the earth and its movement over and beneath the surface, and through the atmosphere (Davie, 2008). One of the fundamental fields of hydrology is the rainfall-runoff relationship which plays an important role in water balance. This relationship is non-linear and the runoff prediction depends on many factors. The discharge prediction depends of rainfall intensity and duration and many physical characteristics of the upstream catchment. A large number of models have been developed to simulate the relationship of rainfall-runoff. These models are categorized as empirical, black-box, conceptual and physically-based models (Rajurkar et al, 2002).

This investigation describes an application of support vector regression (SVR) and multilayer feed-forward neural network (MFNN) models in rainfall-runoff relationship and shows some important issues of nonlinear models applying in runoff forecasting. Moreover in this study a comparison is performed between two models and evaluated the performance and reliability in simulation of rainfall-runoff simulation.

The support vector algorithm is a nonlinear generalization of the generalized portrait algorithm developed by Vapnik, in the early 1960s (Vapnik and Lerner, 1963, Vapnik and Chervonenkis, 1964). The foundation of Support Vector Machines (SVM) is based on the structural risk minimisation principle from statistical learning theory. The method has become very popular due to its performance and its efficiency in handling large scale and nonlinearly separable problems.

Artificial neural networks are based on the highly interconnected structure of brain cells. This approach robust in noisy environments, flexible in the range of problems it can solve, and highly adaptive to non-linear hydrologic phenomena. Mathematically, an artificial neural network may be treated as a universal approximator (ASCE Task Committee, 2000). The ability to learn and generalize ‘‘knowledge’’ from sufficient data pairs makes it possible to solve large-scale complex problems such as the estimation of rainfall-runoff relationship.

As a case study for developing the rainfall-runoff models we used a mountainous watershed, which is located in northern California in a region with the same geographic latitude as Northern Greece. The watershed size is about 385.9 km^2 and the larger part of the catchment surface is covered by pine forest.

2. METHODOLOGY

2.1. Support vector regression

In the real hydrological world, most physical phenomena tend to be nonlinear. In this study the basic concept of the support vector regression is to estimate the nonlinear mapping between rainfall and runoff. Support vector regression estimates the nonlinear function between an input vector x and output variable y. Using a nonlinear mapping $x \rightarrow \Phi(x)$ the relationship between x and y can be converted to linear regression problem between $\Phi(x)$ and y. The form of the function that describes the linear regression relationship between $\Phi(x)$ and y is (equation 1):

$$f(x, \omega) = \omega \cdot \Phi(x) + b \quad (1)$$

The objective of the SVR is to find optimal ω, b and the appropriate parameters in the function Φ(x) so as to construct an approximation of the f(x,ω) function. Moreover the purpose of the SVR is to find a function f that has at most ε deviation from the actually obtained targets for all the training data, and at the same time is as flat as possible (Smola and Scholkopf, 2003). Errors larger than ε are not acceptable. Flatness of function f(x,ω) means that one seeks a small ω and this is ensured by minimizing the term $1/2\|\omega\|^2$. This problem can be written as a convex optimization problem:

$$\begin{aligned} &minimize\ \frac{1}{2}\|\omega\|^2 \\ &subject\ to\ \begin{cases} y - \langle \omega, \Phi(x) \rangle - b \leq \varepsilon \\ \langle \omega, \Phi(x) \rangle + b - y \leq \varepsilon \end{cases} \end{aligned} \qquad (2)$$

The tacit assumption in (2) was that such a function f actually exists that approximates all pairs of inputs and targets (x,y) with ε precision, or in other words, that the convex optimization problem is feasible (Smola and Scholkopf, 2003). Sometimes, however, this may not be the case or the problem may admit bigger errors. In this case the initial problem of equation (2) is modified by introducing slack variables ξ_i, ξ_i' for measurements ''above'' and ''below'' an ε tube and both of them are positive values (Diamantaras, 2007). Slack variables are introduced in order to cope with the otherwise infeasible constraints of the optimization problem of equation (2). Thus we generate the formulation stated in equation (3) (Vapnik, 1995).

$$\begin{aligned} &minimize\ \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}\left(\xi_i + \xi_i'\right) \\ &subject\ to \begin{cases} y - \langle \omega, \Phi(x) \rangle - b \leq \varepsilon \\ \langle \omega, \Phi(x) \rangle + b - y \leq \varepsilon \\ \xi_i, \xi_i' \end{cases} \end{aligned} \qquad (3)$$

In equation (3) the first term $1/2\|\omega\|^2$ reflects generalization, and the second term $C\cdot\Sigma(\xi_i+\xi_i')$ stands for empirical risk. C is a positive constant that determines the degree of penalized loss when a training error occurs (Wu et al, 2008) (the architecture of regression machine is shown in figure 1). The performance of the SVR model depends on the choice of the kernel function and the hyperparameters. In this study, the linear kernel, the polynomial kernel and the Gaussian or radial basis function (RBF) kernel are simulated with the hyperparameter controlling the width of the Gaussian function. In particular for the RBF kernel used three different values for gamma parameter, g=1,2,3.

Linear: $k(x,y) = \langle x, y \rangle$, Polynomial: $k(x,y) = [\langle x, y \rangle + \theta]^p$, RBF: $k(x,y) = e^{-g\cdot\|x-y\|^2}$

2.2. Multilayer feed-forward neural network

The most common neural network used in hydrology applications is the multilayer feed-forward network trained with the back-propagation algorithm (ASCE Task Committee, 2000). The back-propagation algorithm, introduced by Rumelhart (1986), is essentially a gradient-descent technique that minimizes the network error function (Haykin, 1999, ASCE Task Committee, 2000). The neural network employed in this investigation is the multilayer feed-forward one, which is trained with the back-propagation algorithm. For the physical phenomenon of rainfall-runoff relationship, a network with one hidden layer was trained. More specifically the architecture of the network designed with four inputs, h neurons in the first hidden layer and one neuron in the output layer (the architecture of the neural network is shown in figure 1).

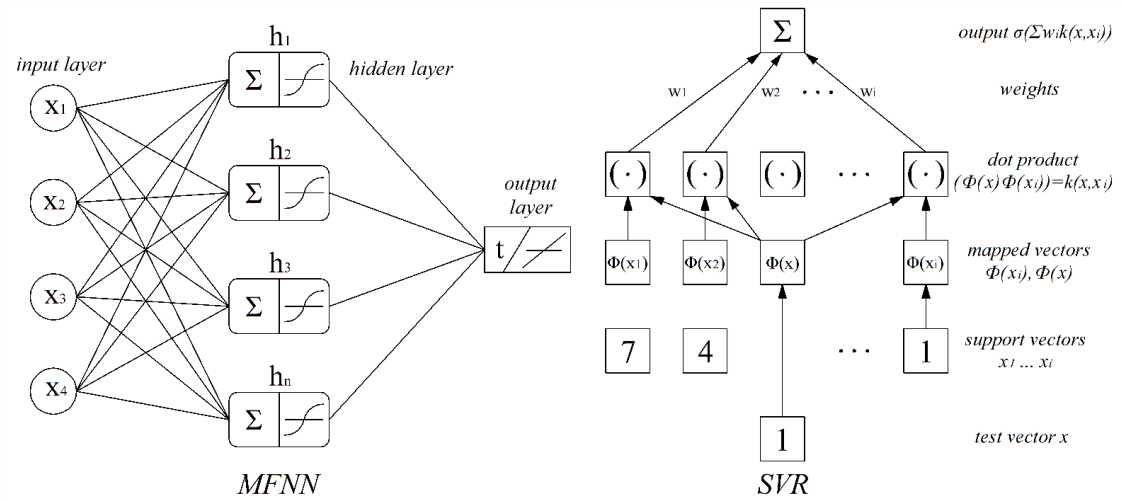

Figure 1: Architecture of neural network and regression machines.

The Levenberg-Marquardt method was selected as the training algorithm, because it is faster and more reliable than any other back-propagation technique (Jeong and Kim, 2005). The computation of the local gradient for each neuron of the neural network requires knowledge of the derivative of the transfer function associated with that neuron. For this derivative to exist it requires the function to be continuous. In basic terms, differentiability is the only requirement that an transfer function has to satisfy (Haykin, 1999). The transfer function should be differentiable, as most training algorithms in multilayer networks are based on optimization methods that use first- and second-order derivatives. In this study the hyperbolic tangent function (expression 4) is used, which is continuous, differentiable, and monotonically increasing.

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (4)$$

The output layer provides a linear activation function, so the output range is between -∞ and ∞.

For the early stopping of the training process the cross validation method was used, which sets an acceptable error level for training and stops training when the mean square error reaches a minimum in the validation phase

2.3. Data preparation

In this study the precipitation from four meteorological stations in catchment area used for inputs data in SVR and MFNN models. The precipitation data are obtained from California Data Exchange Center (CDEC – Department of Water Resources) and the streamflow data are obtained from U.S. Geological Survey (USGS).

Precipitation measurements are the aggregate of daily rainfall and daily water content of snow for 21 years for the period from 31-01-1989 to 31-01-2011. The target of the models is the daily streamflow measurements of Trinity river in northern California for the same period. From the data records we extracted 8036 input–output data pairs of the following format.

$$input: x_1, x_2, x_3, x_4 - \ target\ or\ output: y$$

A remarkable property of ANNs or SVRs is their ability to handle nonlinear, noisy, and nonstationary data (Wu et al, 2008). However with suitable data preparation beforehand, it is possible to improve the performance further (Maier and Dandy, 2000). First we interpolated the data to correct errors of measurements with exceptionally extreme values and complete missing data in the timeseries. In this investigation the inputs and targets of the MFNN were normalised in the range 0 to +1. At the end of the simulations, the model output was un-normalized and a regression analysis was carried out between the measured data and their corresponding un-normalized predicted data. For the SVR it was not realised the normalization of data. For both models SVR and MFNN the data are randomly divided into three sets: training, testing, and validation. While 70% of the data are used for training, 15% are used for validation and the rest 15% used for testing.

2.4. Evaluation of performance

Three different evaluation criteria were used to measure the performance and reliability of the models. Each criterion is represented by an efficiency indicator estimated from the comparison of predicted values and the measured targets that have been initially placed for the training, testing and validation data. The correlation coefficient (R-value) is a measure of the linear regression between the predicted values and the targets of models (expression 5).

$$R = \frac{\sum_{i=1}^{N}(x_i - x)(y_i - y)}{\sqrt{\sum_{i=1}^{N}(x_i - x)}\sqrt{\sum_{i=1}^{N}(y_i - y)}} \quad (5)$$

where N is the number of samples, x_i and y_i are the target and predicted values for i=1,….,n and x and y are the mean values of the target and predicted data set, respectively.

The mean square error (MSE) can be used to quantify the difference between an estimator and the true value of the quantity being estimated. In SVR and MFNN models the mean square error (MSE) is used to measure the performance of a training process. MSE is defined as the average sum of squares of the difference between measured target and predicted values (expression 6).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \quad (6)$$

The root mean square error (RMSE) is a measure of the differences between values predicted by a model or an estimator and the values actually observed (expression 7).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \quad (7)$$

A better agreement between target and predicted values is expressed by an R-value as closer to unity as possible. For normalization data of MFNN model, MSE and RMSE values close to zero show that predictions from the neural network are more precise. For prototype data of SVR as long as smaller is the value of indicator (MSE and RMSE) so much better are the results of model.

3. RESULTS

The performance of SVR and MFNN models shown in Table 1. In this table we show the mean values of performance indicators for the three types of kernels functions and for the all architectures of SVR and MFNN models respectively.

3.1. Results from SVR model

In this investigation simulated the Linear, Polynomial and Gaussian or radial basis function (RBF) kernels. The data were divided randomly into 10 subsets of equal size each time (70% training, 15% validation and 15% testing) and trained the SVR model 10 times for any kernel function. Moreover about the RBF kernel simulated with three different values of gamma parameter. From the table data it is evident that Linear kernels are superior to polynomial and RBF. More analytically the smallest training, validation and testing errors (RMSE) are 15.48, 16.33 and 14.64, respectively, and are produced from the Linear kernel. Also the values of correlation coefficient (R-value) prove that Linear kernel efficiency is the best. Last but not least is that the performance of the models showed that the RBF function is capable of generating lower RMSE results compared to the polynomial function.

The performance of SVR model is presented graphically in figure 2. The validation and training curves shown that the best results inferred by Linear kernel and the worst inferred by the Polynomial kernel. The RBF kernel is characterized by stability for the different values of gamma parameter. When the gamma value is changed from 1 to 2 and at last in 3, the machine does not react to a great degree and produces similar results. In particular the machine would ignore most of the support vectors and hence lead to a failure in the trained point prediction, known as underfitting (Han et al, 2007).

3.2. Results from MFNN model

In this study the cross-validation method for estimating generalization error and the resulting (smallest estimated generalization error) for choosing among various models, such as different network architectures, were employed. Specifically the cross-validation method was used to choose the number of hidden units in the multilayer feed-forward neural network (MFNN). In the same way as with the SVR model three data sets were employed for a rigorous analysis of neural networks performance (training, validation and a testing). The data divided randomly into 10 subsets of equal size and trained the models (SVR and MFNN) 10 times, each time leaving out one of the subsets which used for validation/testing.

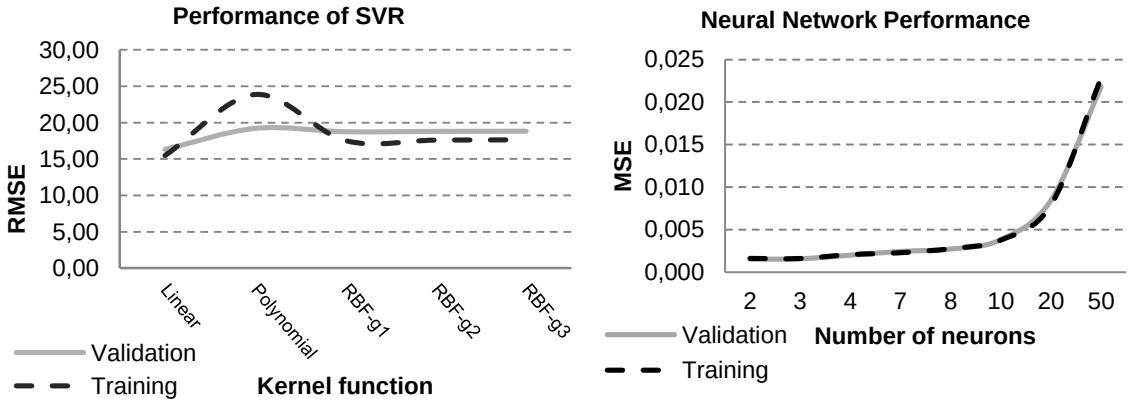

Figure 2: Performance of models (SVR & MFNN).

From the simulation results (Table 1) it is easily noticed that the architectures 4–2–1 and 4–3–1 were superior to all others. The validation and training curves of neural networks illustrated in Figure 2, which clearly demonstrate that the neural networks with big number of neurons in hidden layer are improper for simulating the rainfall-runoff relationship in the specific study area.

Table 1: Performance of SVR and MFNN

Models	*Training*			*Validation*			*Testing*		
	MSE	***RMSE***	***R***	***MSE***	***RMSE***	***R***	***MSE***	***RMSE***	***R***
SVR									
Linear	241.05	15.48	0.54726	282.07	16.33	0.57892	229.10	14.64	0.59133
P/mial	671.21	23.86	0.45132	382.89	19.21	0.54598	328.28	17.69	0.55368
RBF-g1	308.62	17.53	0.44493	367.31	18.75	0.46159	292.30	16.63	0.48830
RBF-g2	310.97	17.59	0.44155	369.32	18.79	0.45951	293.03	16.64	0.48416
RBF-g3	312.38	17.63	0.44004	369.74	18.82	0.45919	295.26	16.72	0.48470
MFNN 3-layer									
4-2-1	0.00162	0.04020	0.36643	0.00153	0.03900	0.37992	0.00172	0.04103	0.38730
4-3-1	0.00161	0.04005	0.40842	0.00158	0.03953	0.40691	0.00142	0.03759	0.42803
4-4-1	0.00207	0.04516	0.30894	0.00200	0.04453	0.32375	0.00212	0.04581	0.31512
4-7-1	0.00229	0.04745	0.29740	0.00244	0.04849	0.28130	0.00267	0.05106	0.28115
4-8-1	0.00277	0.05245	0.31709	0.00272	0.05167	0.31671	0.00253	0.04983	0.33260
4-10-1	0.00377	0.06059	0.24390	0.00382	0.06029	0.27550	0.00377	0.06045	0.24492
4-20-1	0.00787	0.08676	0.20896	0.00843	0.09048	0.17202	0.00840	0.08988	0.20049
4-50-1	0.02283	0.15017	0.11073	0.02179	0.14681	0.11469	0.02311	0.15040	0.14165

3.3. Comparison between the two models

This paper shows that SVR models and especially the linear kernels can yield superior performance against neural networks in the specific catchment. The comparison of the two models is illustrated in Figure 3 which shows the best training of any model (SVR and MFNN). In Figure 3 it is noticed that the prediction values of the SVR model are a better match to the target values.

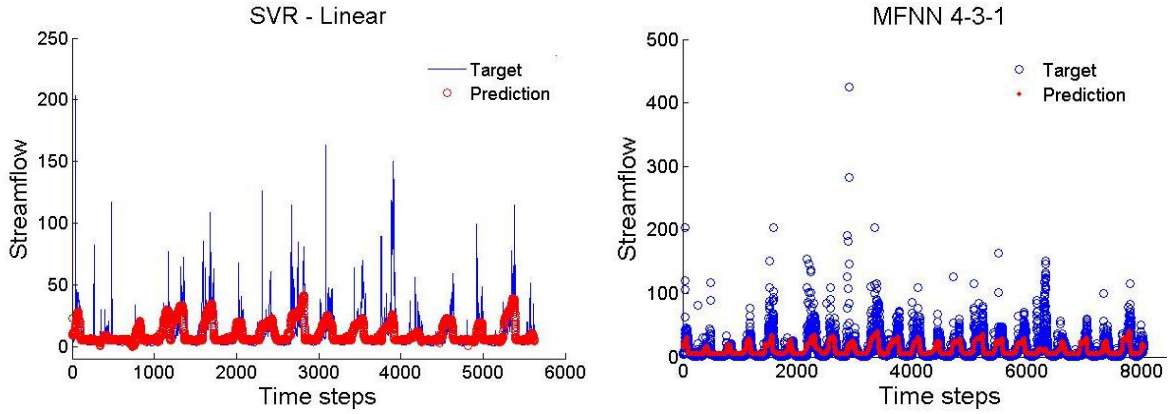

Figure 3: Predicted runoff compared with corresponding flows (targets)

5. CONCLUSIONS

This study demonstrates that SVR models have a better performance than the ANN models in rainfall-runoff simulation. At the same way Liong and Sivapragasam in 2002 compared SVR with Artificial Neural Networks, in the flood stage forecasting and concluded that SVR models inherent properties give it an edge in overcoming some of the major problems in the application of ANN. However both of them have limitations to forecasting outside values of runoff and periods with low or zero values. Han and Yang in 2001 reported these limitations in the river flow modelling field and especially in modelling

of river flows of the Bird Creek catchment in the USA with SVR. Moreover Bray and Han in 2004 illustrated the difficulties in SVR identification for flood forecasting problems.

It was also observed that kernel function selection in the case of SVR has a significant effect on the performance of the model. In particular this investigation concludes that the linear kernel function has the best performance in simulation of rainfall-runoff relationship. Bray in 2002 reaches the same conclusion, finding that the linear kernel outperformed other popular kernel functions (radial basis, polynomial, sigmoid) in runoff modelling. It is interesting to note that Han et al in 2007 come to the conclusion that the linear kernel models outperform RBF models for no-rainfall cases and this could be due to the reduced nonlinear effect from the missing rainfall.

It can be concluded that SVR can replace some of the neural network models for weather prediction applications, but it is clear that there are still many knowledge gaps in applying SVR in rainfall-runoff relationship and flood forecasting. The generalization capability of the SVR and ANN models is the biggest open question in rainfall-runoff estimation problem. It is essential to test these models in a variety of different regions in order to improve the understanding of this potentially powerful tool from the machine learning community.

REFERENCES

1. ASCE Task Committee (2000) Artificial neural network in hydrology, Journal of Hydrologic Engineering, Vol. 5 (2), 124–144.
2. Bray M. (2002) Identification of Support Vector Machines for Runoff Modelling, MEng thesis, Department of Civil Engineering, University of Bristol.
3. Bray M. and Han D. (2004) Identification of support vector machines for runoff modeling, Journal of Hydroinformatics, 6, 265–280.
4. Davie T. (2008) Fundamentals of hydrology, Routledge – Taylor & Francis Group.
5. Diamantaras K. (2007) Artificial Neural Network, Kleidarithmos, Athens.
6. Han D. and Yang Z. (2001) River flow modelling using support vector machines, In XXIX IAHR Congress, Beijing, China, 17–21 September, pp. 494–499, Qinghua University Press, China.
7. Han D., Chan L. and Zhu N. (2007) Flood forecasting using support vector machines, Journal of hydroinformatics, 267.
8. Haykin S. (1999) Neural Networks: A Comprehensive Foundation, Macmillan, New York.
9. Jeong D. and Kim Y. (2005) Rainfall-runoff models using artificial neural networks for ensemble streamflow prediction, Hydrological Processes, Vol. 19, 3819–3835.
10. Liong, S.Y. and Sivapragasam C. (2002) Flood stage forecasting with support vector machines, J. AWRA 38 (1), 173–186.
11. Maier H.R., Dandy G.C. (2000) Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications, Environ. Model Software 15 (1), 101–123.
12. Rajurkar M.P., Kothyari U.C., Chaube U.C. (2002) Artificial neural networks for daily rainfall-runoff modeling, Hydrological Sciences Journal, Vol. 47 (6), 865 – 877.
13. Rumelhart D.E., Hinton G.E., and Williams R.J. (1986) Learning internal representation by error propagation. Parallel distributed processing: Explorations in the microstructure of cognition, Rumelhart D.E. and McClelland J.L., eds., Vol. 1, MIT Press, Cambridge, Mass., 318–362.
14. Smola A.J. and Scholkopf B. (2003) A tutorial of support vector regression.
15. Vapnik V. (1995) The Nature of Statistical Learning Theory, Springer, New York.
16. Vapnik V. and Chervonenkis A. (1964) A note on one class of perceptrons, Automation and Remote Control, 25.
17. Vapnik V. and Lerner A. (1963) Pattern recognition using generalized portrait method, Automation and Remote Control, 24, 774–780.
18. Wu C.L., Chau K.W. and Li Y.S. (2008) River stage prediction based on a distributed support vector regression, Journal of hydrology, 358, 96-111.