Applied Mathematics and Computation

Article *in* Applied Mathematics and Computation · August 2015

DOI: 10.1016/j.amc.2015.08.085

7 authors, including:

Ozgur Kisi
University of Applied Sciences, Lübeck, Germany
543 PUBLICATIONS **21,992** CITATIONS
SEE PROFILE

Jalal Shiri
University of Tabriz
146 PUBLICATIONS **4,741** CITATIONS
SEE PROFILE

Sepideh Karimi
University of Tabriz
27 PUBLICATIONS **547** CITATIONS
SEE PROFILE

Roslan Hashim
134 PUBLICATIONS **3,000** CITATIONS
SEE PROFILE

Applied Mathematics and Computation 270 (2015) 731–743

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A survey of water level fluctuation predicting in Urmia Lake using support vector machine with firefly algorithm

Ozgur Kisi [a], Jalal Shiri [b], Sepideh Karimi [b], Shahaboddin Shamshirband [c,*], Shervin Motamedi [d,e], Dalibor Petković [f], Roslan Hashim [d,e]

[a] Civil Engineering Department, Architectural and Engineering Faculty, Canik Basari University, Samsun, Turkey
[b] Water Engineering Department, Faculty of Agriculture, University of Tabriz, Tabriz, Iran
[c] Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia
[d] Department of Civil Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia
[e] Institute of Ocean and Earth Sciences (IOES), University of Malaya, 50603 Kuala Lumpur, Malaysia
[f] University of Niš, Faculty of Mechanical Engineering, Department of Mechatronics and Control, Aleksandra Medvedeva 14, 18000 Niš, Serbia

ARTICLE INFO

Keywords:
Urmia Lake
Time series
Prediction
Support vector machine
Firefly algorithm

ABSTRACT

Forecasting lake level at various horizons is a critical issue in navigation, water resource planning and catchment management. In this article, multistep ahead predictive models of predicting daily lake levels for three prediction horizons were created. The models were developed using a novel method based on support vector machine (SVM) coupled with firefly algorithm (FA). The FA was applied to estimate the optimal SVM parameters. Daily water-level data from Urmia Lake in northwestern Iran were used to train, test and validate the used technique. The prediction results of the SVM–FA models were compared to the genetic programming (GP) and artificial neural networks (ANNs) models. The experimental results showed that an improvement in the predictive accuracy and capability of generalization can be achieved by the SVM–FA approach in comparison to the GP and ANN in 1 day ahead lake level forecast. Moreover, the findings indicated that the developed SVM–FA models can be used with confidence for further work on formulating a novel model of predictive strategy for lake level prediction.

1. Introduction

Lakes provide water for various domestic, industrial and agricultural applications [1]. Forecasting level of lake water using the previously recorded levels is an interesting approach in water resource planning, lake navigation, management of tidal irrigation as well as drainage canals. Level of lake water is affected by the natural water exchange between the lake and its watershed; thus, the water level reflects the climate changes within the region [2]. For many research and practical applications, it would be beneficial to have a model capable of simulating (and predicting) water level variations based solely on the previously recorded values [2]. So far, numerous studies have been carried out for predicting fluctuations of sea/lake water level using several models, including the neural networks, neuro-fuzzy and genetic programming [2–5].

* Corresponding author. Tel.: +60 146266763; fax: +60 146266763.
E-mail address: shamshirband@um.edu.my (S. Shamshirband).

http://dx.doi.org/10.1016/j.amc.2015.08.085
0096-3003/

Table 1
Statistical parameters of the used data set during the study period [2].

	X_{mean}	X_{max}	X_{min}	SD	C_V	C_{sx}
Training period	1275.5	1277	1273.5	0.93	0.001	−0.93
Testing period	1275.7	1277	1275	0.4	0.001	0.89
Validation period	1276.6	1278.4	1275	0.98	0.001	0.07
Entire dataset	1276	1278.4	1273.5	0.94	0.001	−0.13

Fig. 1. Urmia Lake map.

In this article, we introduce prediction models of daily lake levels for different prediction horizons using the data acquired from Urmia Lake in northwestern Iran. The proposed models were developed using the soft computing approach, namely the support vector machine (SVM) with firefly algorithm (FA).

The SVM is an intelligent technique, which is employed in several engineering disciplines [6,7]. The prediction accuracy of an SVM model highly depends on proper selection of parameters. Various optimization algorithms have been used for selection of these parameters [8–10], such as the grid search algorithm [11] and gradient decent algorithm [12], but the success rate has been minimal. Computational complexity seems to be the main disadvantage of the grid method, which restricts its applicability to simple cases. The grid search algorithm is also prone to local minima. Multiple local solutions exist for most of the optimization problems and evolutionary algorithms seem to be the best, because they are capable of providing global solution to such problems.

Nature-inspired metaheuristic optimization algorithms, such as the ant colony optimization (ACO), genetic algorithm (GA), particle swarm optimization (PSO) and cuckoo search (CS) have found wide applications in different fields of science for several hybrid algorithms [13–24]. The basis of these algorithms is the selection of the most appropriate in natural systems [25]. The latest algorithm among the nature-inspired metaheuristic optimization algorithms is the firefly algorithm (FA) [26]. The FA is believed to be more robust and efficient in finding both global and local optima compared to others [27–33]. The prediction accuracy of the SVM model highly depends on proper choice of model parameters. Even though prearranged approaches for parameter selection are essential, alignment of the model parameter is also important. In this study, the FA was used for determination of the SVM parameters.

A short-term predictive model of daily lake levels was developed using the SVM method, while the model parameters were obtained by the FA. The SVM–FA results were also compared to the genetic programing (GP) and artificial neural networks (ANNs) results.

The organization of the remaining parts of this paper is as follows: Section 2 explains the lake level data and prediction models for different prediction horizons. Section 3 describes the SVM–FA method, as well as GP and ANN as benchmark methods. The comparative results and discussion are put forward in Section 4. Finally, the conclusions are presented in Section 5.

2. Region and data description

Daily records of water level from Urmia Lake (North-Western Iran) were used in the present study. This lake is the world's second largest saline lake. It has started drying out, which has a dramatic impact on the environmental conditions of the neighbor regions. Table 1 presents statistics of the daily water level X, in which C_{sx}, C_V, X_{max}, SD, X_{mean}, and X_{min} represent the skewness coefficient, coefficient of variation, maximum, standard deviation, mean and, minimum, respectively. It can be seen from Table 1 that the train and test data have more skewed distribution than the validation data. Fig. 1 shows the lake map.

From the available time series and the list of water level fluctuations, 6 inputs were used to develop the predictive model. Due to the pronounced inertia of water level fluctuation, time-lagged water level fluctuation was created. The list of the 6 inputs is presented in Table 2.

Table 2
List and description of inputs for water level fluctuation modeling.

Input ID	Input
1	Water level fluctuation (t)
2	Water level fluctuation ($t-1$)
3	Water level fluctuation ($t-2$)
4	Water level fluctuation ($t-3$)
5	Water level fluctuation ($t-4$)
6	Water level fluctuation ($t-5$)

3. Soft computing prediction algorithms

This section describes the soft computing prediction algorithms used in this study. In this view, Section 3.1 elaborates on the support vector machine theory. Section 3.1.1 introduces the firefly searching algorithm. Section 3.2 presents artificial neural network with all used parameters. Finally Section 3.3 gives the main details of applied genetic programming method.

3.1. Support vector machine

Support vector machine (SVM) is a new soft computing learning algorithm that has been used in the field of hydrology, environmental studies and computing [34–39]. Its main applications are in regression analysis, classification, forecasting and pattern recognition, which have shown improved performance over the former methods, such as neural network [40–45]. References [46,47] describe the philosophy behind the SVM development. Structural risk minimization and statistical machine learning process are the bases of SVM. The risk minimization reduces the upper-bound generalization error instead of traditional local training error [47]. Moreover, the SVM offers an exceptional solution towing to the convex characteristic of the ideal problem and employs high-dimensional spaced set of kernel functions that subtly includes non-linear transformation. Therefore, it has no hypothesis in functional transformation, which makes it essential to have linearly divisible data.

Eqs. (1)–(4) express the SVM equations based on the theory of Vapnik. Suppose a series of data points $= \{x_i, d_i\}_i^n$, where n is the data size, d_i represents the target value and x_i is the input space vector of the sample. The SVM estimates the function as represented in Eqs. (1) and (2).

$$f(x) = w\varphi(x) + b \tag{1}$$

$$R_{\mathrm{SVMs}}(C) = \frac{1}{2}w^2 + C\frac{1}{n}\sum_{i=1}^{n} L(x_i, d_i) \tag{2}$$

Here, $\varphi(x)$ is the high-dimensional space feature that plots the x input space vector, b is a scalar, w represents a normal vector and $C\frac{1}{n}\sum_{i=1}^{n} L(x_i, d_i)$ signifies the empirical error. b and w can be assessed by Eq. (2) (minimization of regularized risk) following the ξ_i and ξ_i^* positive slack variables, which characterize upper and lower excess deviation [47].

$$\begin{aligned} &\text{Minimize } R_{\mathrm{SVMs}}(w,\ \xi^{(*)}) = 1/2\ \|\ w\|^2 + C\sum_{(i=1)}^{n}(\xi_i + \xi_i^*) \\ &\text{Subject to} \begin{cases} d_i - w\varphi(x_i) + b_i \le \varepsilon + \xi_i \\ w\varphi(x_i) + b_i - d_i \le \varepsilon + \xi_i^* \\ \xi_i,\ \xi_i^* \ge 0, \quad i = 1, \ldots, l \end{cases} \end{aligned} \tag{3}$$

where $\frac{1}{2}w^2$ represents the regularization term, l represents the quantity of features in the training dataset, C is the factor of error penalty for controlling the difference between the empirical error and regularization term and ε denotes the loss function, which associates with the approximation precision of the training dataset.

Eq. (1) is solved by the optimality constraints and Lagrange multiplier, and a generic function is obtained based on Eq. (4).

$$f(x,\ a_i a_i^*) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x, x_i) + b \tag{4}$$

where the kernel function is $K(x_i, x_j)$ and $K(x, x_i) = \varphi(x_i)\varphi(x_j)$, as product of x_i and x_j inner vectors in $\varphi(x_i)$ and $\varphi(x_j)$ the feature space, respectively.

The SVM mainly finds correlation of data through non-linear mapping. Provided that there is a method to calculate the inner product of feature space directly as a function of main input variable, a non-linear learning machine can be developed as a direct calculation technique of a kernel function. As such, the SVM is flexible in selecting the kernel functions, which indirectly develop a higher-dimensional feature space through data conversion. The results obtained from the higher-dimensional feature space are similar to the findings of the lower-dimensional input space.

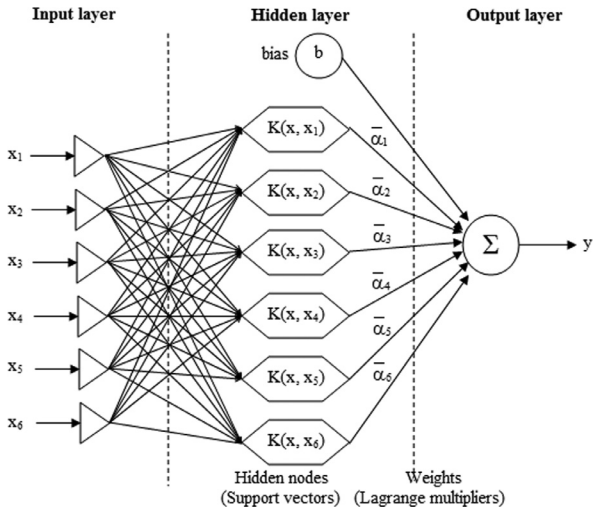

Fig. 2. The network architecture of SVM.

The SVM provides some basic kernel functions, i.e., radial basis, lineal, polynomial and sigmoid functions. It has been proved that the radial basis function (RBF) is the best kernel function owing to its computational reliability, simplicity, efficiency, simplicity of converting to optimization and its flexibility in treating more complex parameters [48–50]. The RBF kernel solves a set of linear equations rather than the computationally demanding and long quadratic programming problem for its training [51]. Thus, in this study it was adopted with σ. The kernel function of non-linear radial basis is:

$$K(x_i, x_j) = e^{-\gamma x_i - x_j{}^2} \tag{5}$$

where variables x_i and x_j are vectors in the input space, i.e. vectors of features computed from training or testing dataset.

In Fig. 2, the hidden nodes' centers are the SVM's support vectors and weights, named as the Lagrange multipliers ($\overline{\alpha_i} = \alpha_i - \alpha_i^*$) that determine the relative significance of the training datasets for the final output.

The accuracy of forecast is very much reliant on the choice of 3 parameters (C, γ, and ε). The FA was used to optimize the parameter selection.

3.1.1. SVM parameters selection using firefly optimization algorithm

Firefly algorithm is grounded on a definite behavioral pattern, most especially as seen in the fireflies' flashing behavior. The fireflies attract prey or mates via the bioluminescence principle. Other fireflies can track another firefly by its luminance enables. This idea of luminance has been used to create algorithms for optimization problem solving. FA has been adjudged to be more robust and efficient in search of global and local optima compared to other nature-inspired optimization algorithms [52–54].

There are fundamental rules in the development of FA. The nature of the encoded cost function changes the brightness of every firefly, the proportion of brightness to the objective function or value of the fitness. The fireflies are presumed unisex; therefore, each of them can attract the other regardless of its gender. The attractiveness of one to another firefly is relational to the luminance quantity or intensity, which drops as their distance increases. Thus, those fireflies with less brightness are attracted to the firefly with more luminance. The variation of light strength and formulation of the objective function (attractiveness) are the main problems in the development of FA. For example, the fitness function in the optimal design problem including objective function maximization is relative to the amount of light produced. Consequently, as the fireflies go farther from each other, the light intensity decreases, which will result in intensity variation and thus the attractiveness decreases between them. The light intensity with changing distance is represented by Eq. (6).

$$I(r) = I_o e^{-\gamma r^2} \tag{6}$$

In this equation, I_o is initial light intensity, the light intensity at r (distance) from a firefly is shown by I, γ represents the coefficient of light absorption that ranges between 0.1 and 10 ($r = 0$) [55]. Since attractiveness is relative to the intensity of light seen by nearby firefly, the attractiveness β at r (distance) from the firefly is:

$$\beta(r) = \beta_o e^{-\gamma r^2} \tag{7}$$

where β_o is the attractiveness at $r = 0$. Eq. (7) is $\Gamma = 1/\sqrt{\gamma}$; obvious changes arise from β_o to $\beta_o e^{-1}$.

Eq. (8) calculates the distance (Cartesian) between any 2 i and j fireflies.

$$r_{ij} = x_i + x_j = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})} \tag{8}$$

```
\Firefly Algorithm\
start
    Define the objective function, f(x), x = (x_i,,,,,,,,,x_d)T
    Generate initial population of fireflies x_i(i = 1,2, ... ... n)
    Determine light intensity l_i at x_i from f(x_i)
    Define light absorption coefficient γ
        while t < Max Generation
                Make a copy of population for movement function
            for i = 1:n all n fireflies
                for j = 1:i all n fireflies
                        if (I_j > I_i)
                        Move fireflies i and j in d-dimension;
                        end if
                        Attractiveness varies with distance r via exp [−γr]
                        Evaluate new solution and update light intensity
                end
            end
            Rank the fireflies and find the current best
        end
        Post process results and visualization
end
```

Fig. 3. Pseudo-code for firefly algorithm.

The movement of i due to attraction to j (brighter firefly) is symbolized as:

$$\Delta x_i = \beta_o e^{-\gamma r^2}(x_j - x_i) + \alpha\varepsilon_i \tag{9}$$

Here, the 1st equation term is a result of attraction, the 2nd one signifies the randomization and coefficient of randomization (α) ranges from 0 to 1 [55]. The random number vector is ε_i, obtained from a Gaussian distribution. The subsequent movement of firefly i is:

$$x_i^{i+1} = x_i + \Delta x_i \tag{10}$$

Fig. 3 summarizes the basic phases of firefly algorithm development.

The SVM models' performance is reliant on the proper selection of parameters, which are computed by the FA. Fig. 4 depicts the flow chart for obtaining the optimal SVM parameters. The optimal SVM parameters are presented in Table 3.

3.2. Artificial neural networks

A very common neural network architecture is the multilayer feed forward network with a back propagation learning algorithm, which has been intensely studied and extensively used in numerous areas. Normally, it has 3 layers, including output, input hidden layer. $D = (X_1, X_2, , X_n)^T$ and $D \in R^n$ are the input vectors; in the hidden layer, $Z = (Z_1, Z_2, , Z_n)^T$ is the output of q neurons; and $Y = (Y_1, Y_2, , Y_n)^T$ and $Y \in R^m$ are the output layer's outputs. Supposing that and the threshold and the weight between the hidden and input layers are y_j and w_{ij}, respectively, and that the threshold and the weight between the output and hidden layers are y_k and w_{jk}, respectively, the following will be output of every neuron in a output and hidden layers:

$$Z_j = f\left(\sum_{i=1}^{n} w_{ij}X_i - \theta_j\right) \tag{11}$$

$$Y_k = f\left(\sum_{j=1}^{q} w_{kj}Z_j - \theta_k\right) \tag{12}$$

Here, transfer function represented by f, which is the mapping rule for the total input of neuron to its output, can be a good instrument to add non-linearity to the design of network. The sigmoid function is one of the most popular functions, which ranges from 0 to 1 and increases monotonically. The ANNs details can be accessed from [56–58]. Fig. 5 shows the ANN structure and the parameters used for the ANN are summarized in Table 4. MATLAB software was used for the ANN simulations.

3.3. Genetic programming

As an evolutionary algorithm, GP defines the relationship between output and input variables. The GP assumes a preliminary set of randomly produced programs, resultant from the accidental grouping of input variables, functions and numbers, which comprise of mathematical functions (log, exp, sin and cos) arithmetic operators and comparison/logical functions. Then, this set of probable solutions is exposed to an evolutionary process and the evolved programs' 'fitness' is assessed (the extent to which they solve the problem). Then, the programs with best fitness are nominated from the first population to exchange information

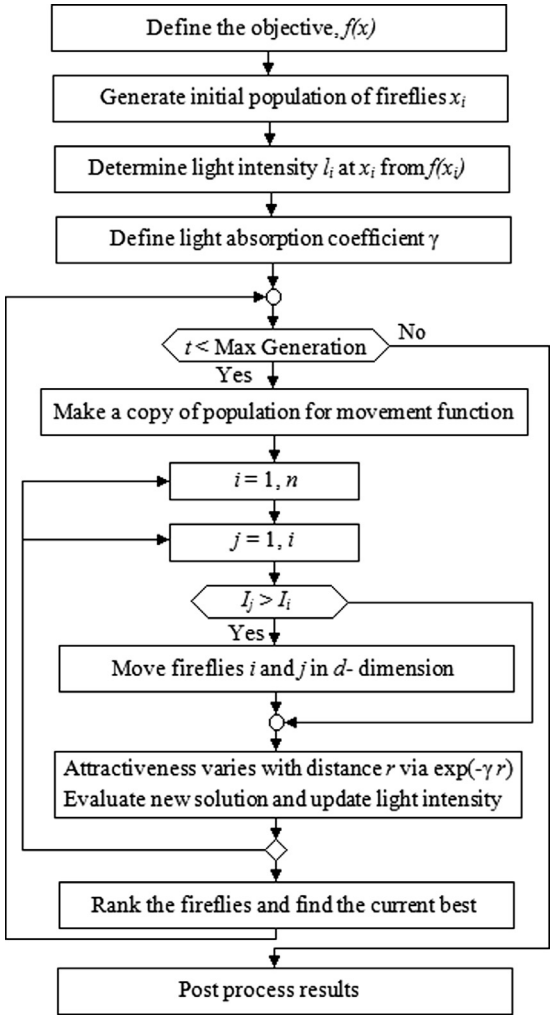

Fig. 4. Flow chart of proposed FA-based parameter determination approach for the SVM classifier.

Table 3
User-defined parameters for SVM model.

SVM model	Parameters
SVM–FA	$C = 1.74$, $\gamma = 0.47$, $\varepsilon = 0.27$

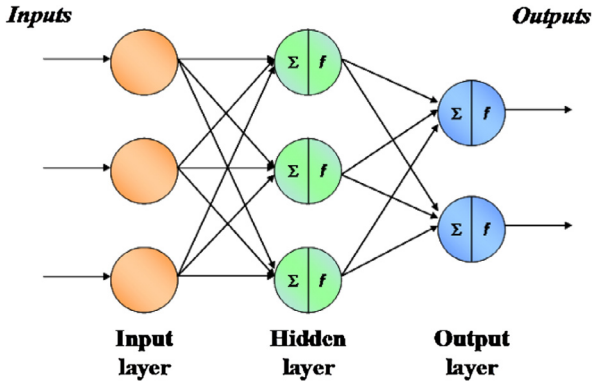

Fig. 5. ANN structure.

Table 4
User-defined parameters for ANN.

ANN parameters				Activation function
Learning rate	Momentum	Hidden node	Number of iteration	
0.2	0.1	3, 6, 10	1000	Continuous log–sigmoid function

Table 5
The list of parameters employed in GP modeling.

Population size	512
Function set	$+, -, *, /, \sqrt{}, x^2, \ln(x), e^x, a^x$
Head size	5–9
Chromosomes	20–30
Linking functions	Addition, subtraction, arithmetic, trigonometric, multiplication
Number of genes	2–3
Mutation rate	91.46
One-point recombination rate	0.2
Two-point recombination rate	0.2
Homologous crossover rate	98.46
Crossover rate	30.56
Fitness function error type	RMSE
Inversion rate	108.53
Gene transposition rate	0.1
Gene recombination rate	0.1

and develop better programs through 'mutation' and 'crossover'. When the best program parts are exchanged with each other, it is crossover, and when programs are changed randomly to generate new programs, it is termed as mutation. Those programs with lower fitness are eliminated. This process of evolution is repeated to find representative expressions labeling the data that are possible to interpret scientifically to gain knowledge. The GP has been explained in details in [59–61]. Table 5 summarizes the parameters used per run of GP. MATLAB software was used for GP simulations.

4. Results and discussion

Predictive models of lake's water levels were developed for 1 and 7 days ahead. As the inputs, we used previous water levels from one to five days in the past. When developing multistep ahead prediction models, two approaches are possible: (a) iterated prediction approach, and (b) direct prediction approach.

In the iterated prediction scheme, one step ahead prediction is used for building the subsequent predictions (i.e. for p steps ahead). Contrary to this, in the direct prediction approach, separate (direct) prediction models are developed for each prediction horizon.

Merits of each of these approaches are discussed extensively [62–64]. Here, the direct approach was adopted. Main benefit of the direct approach is that the procedure is intuitive [65] and there is no accumulation of forecast error, which is not the case for the iterated scheme. Moreover, the direct approach has been successfully used in electricity load forecasting [66–68]; a domain quite similar to the one discussed in this work.

4.1. Input variables for model building

The dataset, consisting of observations for the period of 1972–2003 for each potential predictor was divided in the training set (containing 50% of observations), test set (containing 25% of observations) and validation test (containing 25% of observations). Time series of daily lake level with linear trend for training, test and validation sets are shown in Fig. 6.

The SVM performance depends on parameters selection, which is determined by the FA. Fig. 7 depicts the flow chart for obtaining the optimal SVM parameters for prediction of water level fluctuation. The FA algorithm provides optimal SVM parameters. This is considered its main advantage. Yet, the FA is a data-driven approach; thus, the training data should be selected carefully. The average computation time for the SVM–FA modeling was around 324 s using a PC with the Intel Core Duo CPU E7600 @3.06 GHz and 2-GB RAM. The average computation time for the ANN and GP modeling using the same PC with the same performances was 434 and 456 s, respectively. For the SVM–FA modeling, MATLAB software was used.

4.2. Evaluating accuracy of proposed models

Predictive performance of the proposed models was measured for training, testing and validation datasets by the root means square error (RMSE), coefficient of determination (R^2) and Pearson correlation coefficient (r). These statistics are defined as follows:

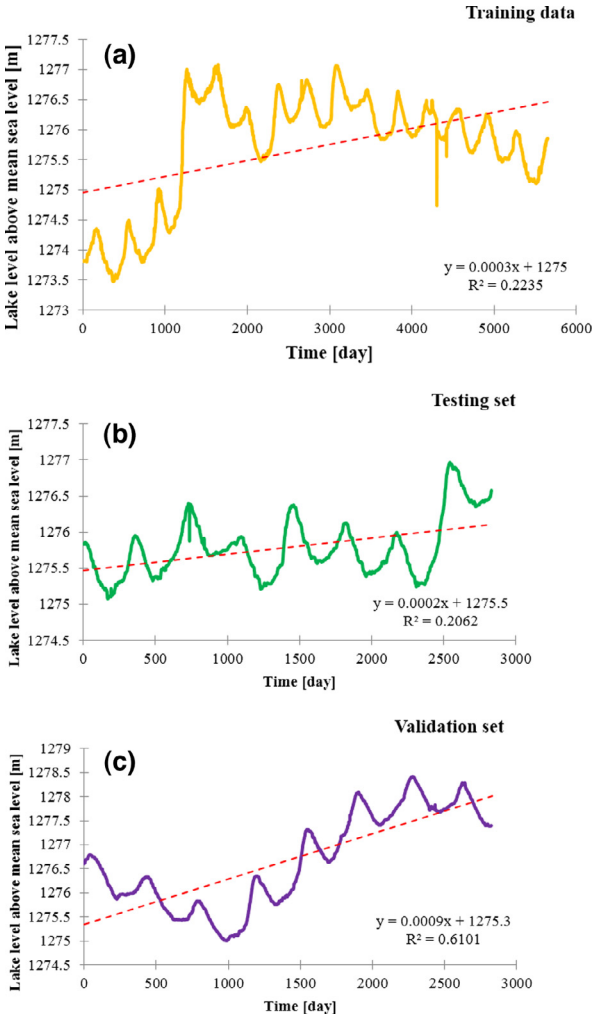

Fig. 6. Time series of the observed lake level values of for (a) training data set, (b) test data set and (c) validation set.

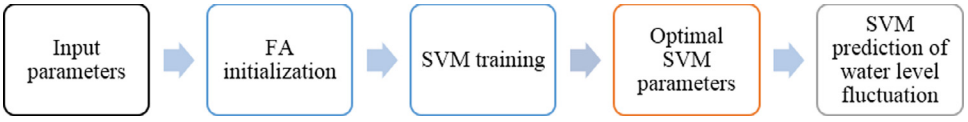

Fig. 7. Flow chart of proposed FA-based parameter determination approach for the SVM for water level fluctuation prediction.

(1) Root-mean-square error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (P_i - O_i)^2}{n}}, \tag{13}$$

(2) Pearson correlation coefficient (r)

$$r = \frac{n\left(\sum_{i=1}^{n} O_i \cdot P_i\right) - \left(\sum_{i=1}^{n} O_i\right) \cdot \left(\sum_{i=1}^{n} P_i\right)}{\sqrt{\left(n\sum_{i=1}^{n} O_i^2 - \left(\sum_{i=1}^{n} O_i\right)^2\right) \cdot \left(n\sum_{i=1}^{n} P_i^2 - \left(\sum_{i=1}^{n} P_i\right)^2\right)}} \tag{14}$$

(3) Coefficient of determination (R^2)

$$R^2 = \frac{\left[\sum_{i=1}^{n} \left(O_i - \overline{O_i}\right) \cdot \left(P_i - \overline{P_i}\right)\right]^2}{\sum_{i=1}^{n} \left(O_i - \overline{O_i}\right) \cdot \sum_{i=1}^{n} \left(P_i - \overline{P_i}\right)} \tag{15}$$

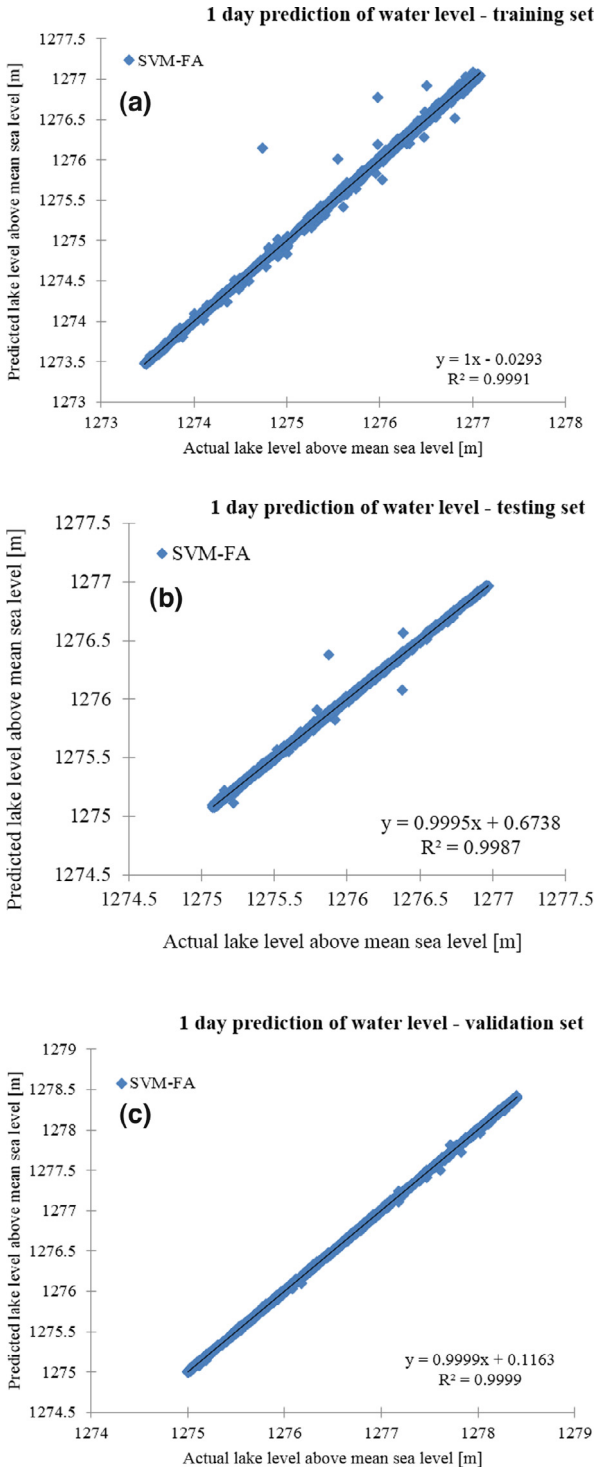

Fig. 8. Scatter plots of actual and predicted lake level values of using SVM–FA method for 1 day ahead for (a) training set, (b) testing set and (c) validation set.

where P_i and O_i are known as the experimental and forecast values of lake level, respectively, and n is the total number of test data. The main goal of the training procedure was to achieve R^2 and r coefficients higher than 0.99, because high correlation is the most important factor of the prediction accuracy.

While some controversy regarding the validity of this error metrics does exist, it pertains to comparison of forecasting methods across different datasets [69,70], which was not the case in this work.

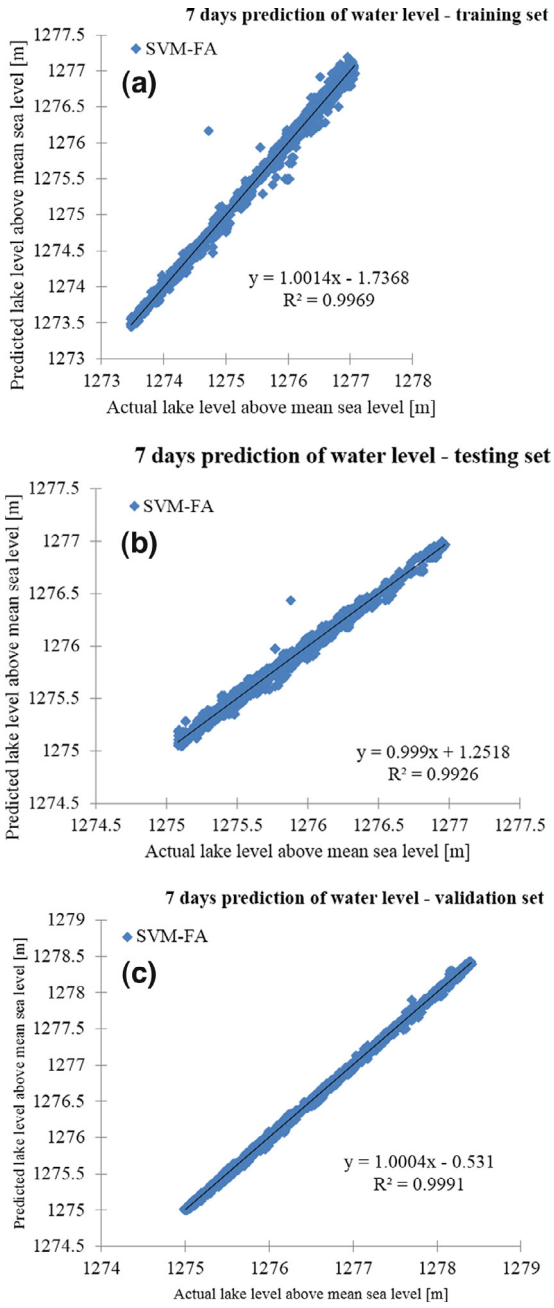

Fig. 9. Scatter plots of actual and predicted lake level values using SVM–FA method for 7 days ahead for (a) training set, (b) testing set and (c) validation set.

4.3. Performance evaluation of proposed SVM–FA models

In this section, the performance results of the SVM–FA models are reported. Fig. 8 presents the training, testing and validation accuracy of the developed SVM–FA model for the horizon of 1 day ahead. The majority of points are on the diagonal line for one day ahead prediction model. Therefore, prediction findings are in line with the values measured for the SVM–FA technique. Very high value of coefficient of determination for one day ahead prediction also confirms these results. As was expected, the prediction accuracy deteriorates with the enlarged prediction horizon. However, training error is not a credible indicator for prediction potential of a particular model.

To further prove the potential of the proposed SVM–FA method for the water level prediction, the method's suitability for 7 days ahead prediction was assessed. Scatter plot of the observed water levels against the predicted ones is illustrated in Fig. 9. The coefficient of determination in validation set ($R^2 = 0.9991$) reveals that the predicted values were quite close to the observed

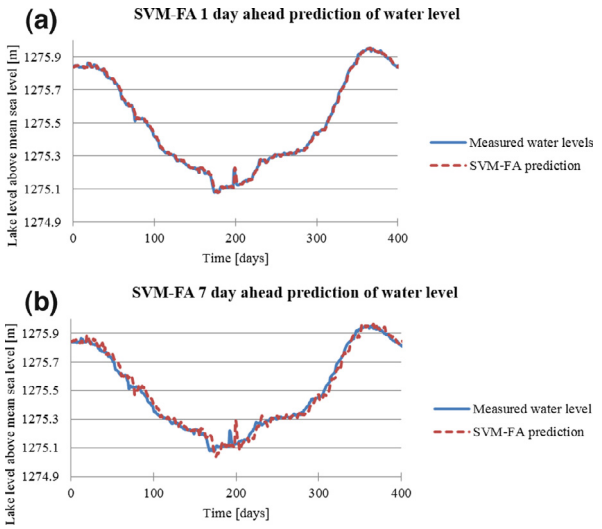

Fig. 10. Predicted lake level values against measured data using SVM–FA method for (a) 1 day ahead for (b) 7 days ahead.

Table 6
Comparative performance statistics of the SVM–FA, ANN and GP models for different lake level prediction horizons.

Prediction horizon		SVM–FA			ANN			GP		
		RMSE	R^2	r	RMSE	R^2	r	RMSE	R^2	r
1 day ahead	Training	0.028259	0.9991	0.999534	0.028244	0.9991	0.999535	0.028234	0.9991	0.999535
	Testing	0.005073	0.9987	0.999924	0.002979	0.9986	0.999974	0.002981	0.9986	0.999974
	Validation	0.000565	0.9999	1	0.00093	0.9999	1	0.001347	0.9999	0.999999
7 days ahead	Training	0.051682	0.9969	0.998442	0.053904	0.9967	0.998303	0.053484	0.9966	0.99833
	Testing	0.035187	0.9926	0.996315	0.036516	0.9921	0.996035	0.036861	0.9919	0.995945
	Validation	0.030238	0.9991	0.999534	0.029405	0.9991	0.999559	0.032507	0.9989	0.99946

values. But it is clear that 1 day ahead prediction model is better, because coefficient of determination in the validation set was $R^2 = 0.9999$). There are limited number of produced underestimated or overestimated values. Thus, high precision of the predicted values is evident.

Fig. 10 presents predictions of the lake levels for 1 day and for 7 days ahead. Clear difference can be seen between these two models. The model with 1 day ahead is more accurate than the 7 days ahead prediction model.

5. Performance comparison of SVM–FA, ANN and GP

To validate the virtues of the suggested SVM–FA approach on a more tangible and definite basis, the prediction precision of the SVM–FA model was compared to those of the GP and ANN methods. For each of the three aforementioned methods, two different direct predictive models were developed (for different prediction horizons). To compare, R^2, RMSE and r were used. The prediction accuracy of each model for training, test and validation datasets is summarized in Table 6.

For the training dataset, all the three methods yielded similar results. In case of 1 day ahead lake level forecasting, the SVM–FA model performed better than the ANN and GP models in the validation period. In case of 7 days ahead lake level forecasting, however, the SVM–FA did better forecasts than the GP. The ANN seems to be slightly better than the SVM–FA in this sense.

6. Conclusion

The ability of the SVM–FA model in prediction of daily lake levels was explored. The proposed SVM–FA models were developed by combining the FA and SVM. The SVM implements structural minimization, whereas the FA is applied to determine the optimal SVM parameters.

Observations collected from Urmia Lake (northwestern Iran) during 1972–2003 were used for model development and testing. Two predictive models for different prediction horizons (1 day and 7 days ahead) were created using the SVM–FA. Comparison of the SVM–FA method with GP and ANN indicated that the SVM–FA model is superior to the GP and ANN in 1 day ahead lake level prediction. The results suggest that the proposed SVM–FA model can be successfully used for short-term lake level forecast.

Acknowledgments

We sincerely thank the editor and the reviewers for their constructive comments. The authors express their sincere thanks for the funding support they received from the HIR-MOHE University of Malaya under grant no. UM.C/HIR/MOHE/ENG/34.

References

[1] V. Vuglinskiy, Water level: water level in lakes and reservoirs, water storage, Assessment of the Status of the Development of the Standards for the Terrestrial Essential Climate Variables, Global Terrestrial Observing System (GTOS), Rome, Italy, 2010, p. 26.
[2] S. Karimi, J. Shiri, O. Kisi, O. Makarynskyy, Forecasting water level fluctuations of Urmia Lake using gene expression programming and adaptive neuro-fuzzy inference system, Int. J. Ocean Clim. Syst. 3 (2012) 109–125.
[3] S. Karimi, O. Kisi, J. Shiri, O. Makarynskyy, Neuro Fuzzy and neural network techniques for forecasting sea level in Darwin Harbor, Australia, Comp. Geosci. 52 (2013) 50–59.
[4] O. Kisi, J. Shiri, B. Nikoofar, Forecasting daily lake levels using artificial intelligence approaches, Comp. Geosci. 41 (2012) 169–180.
[5] J. Shiri, O. Makarynskyy, O. Kisi, W. Dierickx, A. FakheriFard, Prediction of short term operational water levels using an adaptive neuro-fuzzy inference system, ASCE J. Waterway Port Coast. Ocean Eng. 137 (6) (2011) 344–354.
[6] L. Ornella, E. Tapia, Supervised machine learning and heterotic classification of maize (*Zea mays* L.) using molecular marker data, Comp. Electron. Agric. 74 (2) (2010) 250–257.
[7] P. Jain, J.M. Garibaldi, J.D. Hirst, Supervised machine learning algorithms for protein structure classification, Comp. Biol. Chem. 33 (3) (2009) 216–223.
[8] Y. Bao, Z. Hu, T. Xiong, A PSO and pattern search based memetic algorithm for SVMs parameters optimization, Neurocomputing 117 (2013) 98–106.
[9] F. Friedrichs, C. Igel, Evolutionary tuning of multiple SVM parameters, Neurocomputing 64 (2005) 107–117.
[10] A.C. Lorena, A.C. De Carvalho, Evolutionary tuning of SVM parameter values in multiclass problems, Neurocomputing 71 (16) (2008) 3326–3334.
[11] C.W. Hsu, C.C. Chang, C.J. Lin, A Practical Guide to Support Vector Classification, Technical report, Department of Computer Science, National Taiwan University, 2003 16 pp.
[12] K.M. Chung, W.C. Kao, T. Sun, L.L. Wang, C.J. L, Radius margin bounds for support vector machines with the RBF kernel, in: Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'O2), 2, 2002, pp. 893–897.
[13] J.S. Angelo, H.S. Bernardino, H.J.C. Barbosa, Ant colony approaches for multiobjective structural optimization problems with a cardinality constraint, Adv. Eng. Softw. 80 (2015) 101–115.
[14] E. Assareh, M.A. Behrang, M.R. Assari, A. Ghanbarzadeh, Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran, Energy 35 (12) (2010) 5223–5229.
[15] M. Dorigo, G. Di Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artif. Life 5 (2) (1999) 137–172.
[16] M. Dorigo, T. Stützle, The ant colony optimization metaheuristic: algorithms, applications, and advances., Handbook of Metaheuristics. International Series in Operations Research & Management Science, Vol. 57, Springer, US, 2003, pp. 250–285.
[17] X.S. Yang, S. Deb, Cuckoo search: recent advances and applications, Neural Comput. Appl. 24 (1) (2014) 169–174.
[18] Y. Bao, Z. Hu, T. Xiong, A PSO and pattern search based memetic algorithm for SVMs parameters optimization, Neurocomputing 117 (2013) 98–106.
[19] M. Basu, Modified particle swarm optimization for nonconvex economic dispatch problems, Int. J. Electric. Pow. Energ. Syst. 69 (2015) 304–312.
[20] Z. Beheshti, S.M. Shamsuddin, S. Hasan, Memetic binary particle swarm optimization for discrete optimization problems, Inform. Sci. 299 (2015) 58–84.
[21] M. Kumar, T.K. Raat, Optimal design of FIR fractional order differentiator using cuckoo search algorithm, Exp. Syst. Appl. 42 (7) (2015) 3433–3449.
[22] A. Teske, R. Falcom, A. Nayak, Efficient detection of faulty nodes with cuckoo search in t-diagnosable systems, Appl. Soft Comput. 29 (2015) 52–64.
[23] P.J. García Nieto, E. García-Gonzalo, J.R. Alonso Fernández, C Díaz Muñiz, HHybrid PSO—SVM-based method for long-term forecasting of turbidity in the Nalón river basin: a case study in Northern Spain, Ecol. Eng. 73 (2014) 192–200 December.
[24] P.J. García Nieto, J.R. Alonso Fernández, F.J. de Cos Juez, F. Sánchez Lasheras, C. Díaz Muñiz, Hybrid modelling based on support vector regression with genetic algorithms in forecasting the cyanotoxins presence in the Trasona reservoir (Northern Spain), Env. Res. 122 (2013) 1–10 April.
[25] I. Fister Jr., Z.S. Yang, I. Fister, J. Brest, D. Fister, A Brief review of nature-inspired algorithms for optimization, Elektrotehniški vestnik 80 (3) (2013) 1–7.
[26] X.S. Yang, Firefly algorithms for multimodal optimization. Stochastic algorithms: foundations and applications, Lect. Notes Comput. Sci. 5792 (2009) 169–178.
[27] I. Fister, I.J. Fister, X.S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm Evolution. Comput. 13 (2013) 34–46.
[28] X.S. Yang, Multiobjective firefly algorithm for continuous optimization, Eng. Comp. 29 (2) (2013) 175–184.
[29] S. Ch., S.K. Sohani, D. Kumar, A. Malik, B.R. Chahar, A.K. Nema, B.K. Panigrahi, R.C. Dhiman, A support vector machine-firefly algorithm based forecasting model to determine malaria transmission, Neurocomputing 129 (2014) 279–288.
[30] T. Kanimozhi, K. Latha, An integrated approach to region based image retrieval using firefly algorithm and support vector machine, Neurocomputing 151 (3) (2015) 1099–1111.
[31] S.R. Massan, A.I. Wagan, M.M. Shaikh, R. Abro, Wind turbine micrositing by using the firefly algorithm, Appl. Soft Comput. 27 (2015) 450–456.
[32] I. Fister Jr., M. Perc, S.M. Kamal, I. Fister, A review of chaos-based firefly algorithms: perspectives and research challenges, Appl. Math. Comput. 252 (2015) 155–165.
[33] I. Fister, X.S. Yang, J. Brest, I. Fister Jr., Memetic self-adaptive firefly algorithm, Swarm Intelligence and Bio-Inspired Computation Theory and Applications, Elsevier, 2013, pp. 73–102.
[34] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn. 46 (2002) 131–159.
[35] S.W. Lee, A. Verri, Support Vector Machines for Computer Vision and Pattern Recognition, World Scientific, 2003.
[36] W.Z. Lu, W.J. Wang, Potential assessment of the "support vector machine" method in forecasting ambient air pollutant trends, Chemosphere 59 (5) (2005) 693–701.
[37] T. Asefa, M. Kemblowski, M. McKee, A. Khalil, Multi-time scale stream flow predictions: the support vector machines approach, J. Hydrol. 318 (1) (2006) 7–16.
[37] Y. Ji, S. Sun, Multitask multiclass support vector machines: model and experiments, Pattern Recogn. 46 (3) (2013) 914–924.
[39] S. Sun, A survey of multi-view machine learning, Neural Comput. Appl. 23 (7–8) (2013) 2031–2038.
[40] T. Joachims, Text Categorization With Support Vector Machines: Learning With Many Relevant Features, Springer, 1998.
[41] S. Mukkamala, G. Janoski, A. Sung, Intrusion detection using neural networks and support vector machines, in: International Joint Conference on Neural Networks, IJCNN'02., IEEE, 2002.
[42] A.H. Sung, S. Mukkamala, Identifying important features for intrusion detection using support vector machines and neural networks, in: 2003 Symposium on Applications and the Internet, IEEE, 2003.
[43] V. Vapnik, S.E. Golowich, A. Smola, Support vector method for function approximation, regression estimation, and signal processing, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, 9th ed., MIT Press, Cambridge, MA, 1997, pp. 281–287.
[44] C. Huang, L. Davis, J. Townshend, An assessment of support vector machines for land cover classification, Int. J. Remote Sens. 23 (4) (2002) 725–749.
[45] R. Collobert, S. Bengio, Support vector machines for large-scale regression problems, Institut Dalle Molle d'Intelligence Artificelle Perceptive (IDIAP), Martigny, Switzerland, 2000 Tech. Rep. IDIAP-RR-00-17.
[46] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 2000.
[47] V. Vapnik, Statistical Learning Theory, 2, Wiley, Ne w Y ork, 1998.
[48] S. Rajasekaran, S. Gayathri, T.L. Lee, Support vector regression methodology for storm surge predictions, Ocean Eng. 35 (16) (2008) 1578–1587.

[49] H. Yang, K. Huang, I. King, M.R. Lyu, Localized support vector regression for time series prediction, Neurocomputing 72 (10) (2009) 2659–2669.
[50] K.P. Wu, S.D. Wang, Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space, Pattern Recogn. 42 (5) (2009) 710–717.
[51] S. Shamshirband, D. Petkovic, H. saboohi, N.B. Anuar, I. Inayat, S. Akib, Z. Cojbasic, V. Nikolic, M.L. Mat Kiah, A. Ghani, Wind turbine power coefficient estimation by soft computing methodologies: comparative study, Energ. Convers. Manage. 81 (2014) 520–526.
[52] I. Fister, I. Fister Jr., X.S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm Evolution. Comput. 13 (2013) 34–46.
[53] S.K. Pal, C. Rai, A.P. Singh, Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems, Int. J. Intell. Syst. Appl. (IJISA) 4 (10) (2012) 50.
[54] X.S. Yang, Multiobjective firefly algorithm for continuous optimization, Eng. Comp. 29 (2) (2013) 175–184.
[55] S. Ch, S.K. Sohani, D. Kumar, A. Malik, B.R. Chahar, A.K. Nema, B.K. Panigrahi, R.C. Dhiman, A support vector machine-firefly algorithm based forecasting model to determine malaria transmission, Neurocomputing 129 (2014) 279–288.
[56] N. Karunanithi, W.J. Grenney, D. Whitley, K. Bovee, Neural networks for river flow prediction, J. Comput. Civil Eng. 8 (2) (1994) 201–220.
[57] ASCE Task Committee, Artificial neural networks in hydrology. 1. Preliminary concepts, J. Hydrol. Eng. 5 (2) (2000) 115–123.
[58] ASCE Task Committee, Artificial neural networks in hydrology. 2. Hydrologic applications, J. Hydrol.Eng. 5 (2) (2000) 124–137.
[59] J.R. Koza, Genetic Programming: On the Programming of Computers by Natural Selection, MIT Press, Cambridge , MA.
[60] V. Babovic, M. Keijzer, Rainfall runoff modeling based on genetic programming, Nord. Hydrol. 33 (2000) 331–346.
[61] C. Sivapragasam, P. Vincent, G. Vasudevan, Genetic programming model for forecast of short and noisy data, Hydrol. Proc. 21 (2) (2007) 266–272.
[62] M. Casdagli, Nonlinear prediction of chaotic time series, Physica D 35 (1989) 335–356.
[63] J. Doyne Farmer, J.J. Sidorowich, Predicting chaotic time series, Phys. Rev. Lett. 59 (8) (1987) 845–848.
[64] M. Marcellino, J. Stock, M.W. Watson, A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series, J. Econom. 135 (2006) 499–526.
[65] G. Chevillon, D.F. Hendry, Non-parametric direct multi-step estimation for forecasting economic processes, Int. J. Forecast. 21 (2) (2005) 201–218.
[66] P.E. McSharry, S. Bouwman, G. Bloemhof, Probabilistic forecasts of the magnitude and timing of peak electricity demand, IEEE Transact. Pow. Syst. 20 (2005) 1166–1172.
[67] R. Ramanathan, R.F. Engle, C.W.J. Granger, F. Vahid, C. Brace, Short-run forecasts of electricity loads and peaks, J. Forecast. 13 (1997) 161–174.
[68] S. Fan, R.J. Hyndman, Short-term load forecasting based on a semi-parametric additive model, IEEE Transac. Pow. Syst. 27 (1) (2012) 134–141.
[69] J.S. Armstrong, R. Fildes, Correspondence on the selection of error measures for comparisons among forecasting methods, J. Forecast. 14 (1) (1995) 67–71.
[70] J.S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: empirical comparisons, Int. J. Forecast. 8 (1) (1992) 69–80.