Geoscience Frontiers xxx (2016) 1–11

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

China University of Geosciences (Beijing)

Geoscience Frontiers

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

Research paper

A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in Delaware River

Ehsan Olyaie [a,*], Hamid Zare Abyaneh [b], Ali Danandeh Mehr [c]

[a] Young Researchers and Elite Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran
[b] Department of Water Engineering, College of Agriculture, Bu-Ali Sina University, Hamedan, Iran
[c] Istanbul Technical University, Civil Engineering Department, Hydraulics Division, 34469 Maslak, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 13 February 2016
Received in revised form
9 April 2016
Accepted 30 April 2016
Available online xxx

Keywords:
Dissolved oxygen
SVM
LGP
ANN
Modeling

ABSTRACT

Most of the water quality models previously developed and used in dissolved oxygen (DO) prediction are complex. Moreover, reliable data available to develop/calibrate new DO models is scarce. Therefore, there is a need to study and develop models that can handle easily measurable parameters of a particular site, even with short length. In recent decades, computational intelligence techniques, as effective approaches for predicting complicated and significant indicator of the state of aquatic ecosystems such as DO, have created a great change in predictions. In this study, three different AI methods comprising: (1) two types of artificial neural networks (ANN) namely multi linear perceptron (MLP) and radial based function (RBF); (2) an advancement of genetic programming namely linear genetic programming (LGP); and (3) a support vector machine (SVM) technique were used for DO prediction in Delaware River located at Trenton, USA. For evaluating the performance of the proposed models, root mean square error (RMSE), Nash–Sutcliffe efficiency coefficient (NS), mean absolute relative error (MARE) and, correlation coefficient statistics (R) were used to choose the best predictive model. The comparison of estimation accuracies of various intelligence models illustrated that the SVM was able to develop the most accurate model in DO estimation in comparison to other models. Also, it was found that the LGP model performs better than the both ANNs models. For example, the determination coefficient was 0.99 for the best SVM model, while it was 0.96, 0.91 and 0.81 for the best LGP, MLP and RBF models, respectively. In general, the results indicated that an SVM model could be employed satisfactorily in DO estimation.

1. Introduction

Dissolved oxygen (DO) concentration reflects the equilibrium between oxygen-producing (e.g., photosynthesis) and oxygen-consuming (e.g., aerobic respiration, nitrification, and chemical oxidation) processes in aquatic ecosystems. It depends on many factors such as temperature, salinity, oxygen depletion, oxygen sources, and others (Kalff, 2002; YSI, 2009). DO level is the criterion of health (Rankovic et al., 2010), which is frequently used for water quality control at different aquatic systems such as reservoirs and wetlands (Singh et al., 2009; Ay and Kisi, 2012; Kisi et al., 2013).

The water quality modeling using either deterministic (Garcia et al., 2002; Hull et al., 2008; Shukla et al., 2008) or stochastic approaches (Boano et al., 2006) recently received great attention because of its important role in human and environment health. Owing to the dynamic feature of DO concentration, especially in rivers and wetlands, it is greatly advisable to generate DO models for aquatic ecosystems periodically, so that quality control measures can be optimized throughout a time horizons. To this end, implementation of different artificial intelligence (AI) techniques were suggested in the relevant literature.

Since 1990s, based on the understanding of the brain and nervous systems, artificial neural networks (ANNs) have been gradually used in hydrological predictions. An ANN learns to solve a problem by developing a memory capable of correlating a large number of input patterns with a resulting set of yields. They operate like a "black box" model, requiring no detailed information about the system (Ahmed et al., 2013). One of the most important

[*] Corresponding author. Tel.: +98 8134424090; fax: +98 8134424012.
E-mail address: e.olyaie@iauh.ac.ir (E. Olyaie).
Peer-review under responsibility of China University of Geosciences (Beijing).

http://dx.doi.org/10.1016/j.gsf.2016.04.007

advantages of ANNs is their ability to handle large and complex systems with many interrelated parameters (Nourani et al., 2011). An extensive review of its use in hydrological field was given by ASCE Task Committee on application of ANN in hydrology (ASCE, 2000). Different ANN algorithms were applied for water quality modeling. For example, Schmid and Koskiaho (2006) investigated the accuracy of various multi-layer perceptron (MLP) algorithms to forecast DO concentration in Finland. Singh et al. (2009) modeled DO concentration and biological oxygen demand (BOD) in the Gomti River in India using three-layer feed forward neural networks (FNN) with back propagation learning. FNN algorithm was also implemented by Rankovic et al. (2010) to predict DO in Gruza Reservoir, Serbia. Ay and Kisi (2012) compared efficiency of two different ANN algorithms in DO prediction in Foundation Creek, Colorado. Antanasijević et al. (2013) developed three different ANN architectures to improve the performance ANN modeling in DO concentration in the Danube River. More recently, back propagation neural network (BPNN) and adaptive neural-based fuzzy inference system (ANFIS) were applied by Chen and Liu (2014) to estimate the DO concentration in the Feitsui Reservoir of northern Taiwan. All of the abovementioned studies demonstrated that different ANN algorithms can be used as a satisfactory tool for DO modeling. However, explicit formulation of DO for the ecosystem of the interest remains as a problem.

Genetic Programming (GP) is another AI-based technique commonly used for hydrological predictions at nonlinear systems. The GP technique is a relatively new technique compared to ANN. The most powerful feature of GP is that the user can easily obtain an explicit program/formula of the relation between the inputs and output, which makes GP more interesting for hydrologists and practitioners (Guven and Kisi, 2013). Since the general review of GP application in water engineering is out of the scope of our study, the interested researchers can refer Ghorbani et al. (2010), Guven and Azamathulla (2012), and Traore and Guven (2013). This study specifically focuses on a new branch of GP, called Linear Genetic Programming (LGP).

Since the last decade, LGP has been pronounced as a new robust method to solve wide range of modeling problems in water engineering and has been limitedly used in estimation hydrological parameters (e.g., Guven, 2009; Guven et al., 2009; Kisi and Guven, 2010; Danandeh Mehr et al., 2013, 2014a,b and c). Guven (2009) applied LGP, a variant of GP, and two versions of neural networks for prediction of daily flow of Schuylkill River in the USA and showed that the performance of LGP was moderately better than that of ANN. Danandeh Mehr et al. (2013) applied LGP in comparison with a neuro-wavelet technique in time series modeling of stream flow on Coruh River in Turkey. Londhe and Charhate (2010) used ANN, GP and Model Trees (MT) to forecast river flow one day in advance at two stations in Narmada catchment of India. The results showed the ANNs and MT techniques performed almost equally well, but GP performed better than its counterparts. Marti et al. (2013) applied ANN and Gene Expression Programming (GEP) based models to estimate outlet DO in micro-irrigation sand filters. Also, Kisi et al. (2013) investigated the ability of GEP, ANFIS and ANN techniques in modeling DO concentration and showed that the GEP model performed better than the ANN and ANFIS models in modeling DO concentration.

Recently another mathematical tool, the Support Vector Machine (SVM), has been used in hydrology. The SVM is based on structural risk minimization (SRM) principle and is an approximation implementation of the method of SRM with a good generalization capability (Vapnik, 1998). Although SVM has been applied for a relatively short time, this learning machine has been proven to be a robust and competent algorithm for both classification and regression in many disciplines. Recently, the use of the SVM in water resources engineering has attracted much attention. Dibike et al. (2001) demonstrated its use in rainfall–runoff modeling. Liong and Sivapragasam (2002) applied SVM to flood stage forecasting in Dhaka, Bangladesh and concluded that the accuracy of SVM exceeded that of ANN in one-lead-day to seven-lead-day forecasting.

Sivapragasam and Muttil (2005) extended the rating curves developed at three gauging stations in Washington by SVM. Khan and Coulibaly (2006) applied SVM to predict future water levels in Lake Erie. Yu et al. (2006) successfully explored the usefulness of SVM based modeling technique for predicting of real-time flood stage forecasting on Lan-Yang River in Taiwan 1–6 h ahead. Cimen (2008) used SVM to predict daily suspended sediments in rivers. Wu et al. (2008) used a distributed support vector regression for river stage prediction. Wang et al. (2009) developed and compared several AI techniques include ANN, neural-based fuzzy inference system (ANFIS), GP and SVM for monthly flow forecasting using long-term observations in China. Their results indicated that the best performance can be obtained by ANFIS, GP and SVM, in terms of different evaluation criteria. To the best knowledge of the authors, there is not any published study indicating the input–output mapping capability of LGP and SVM techniques in modeling of DO concentration for rivers.

Therefore, the present study is focused on construction of different computational intelligence models, such as two different ANN models, namely, the MLP and RBF, and LGP and SVR to predict the DO concentration at a particular river water using a hydrochemical data set. The obtained results are finally compared to each other. For this purpose, based on a gauging station records, we put forward six black-box ANN structures as reference models for DO concentration prediction on Delaware River located at Trenton, NJ (USGS Station No: 01463500), USA. Then LGP and SVM were applied to model the reference scenarios. These methods offer advantages over conventional modeling, including the ability to handle large amounts of noisy data from dynamic and nonlinear systems, especially, when the underlying physical relationships are not fully understood. Ultimately, both accuracy and applicability of ANN, LGP, and SVM techniques were discussed via the comparison of their performances. It is relevant to note that the models investigated in this study are normally applied within deterministic frameworks in professional practices, which encouraged the practice of comparing the actual with predicted values. Therefore, the paper presents a comparative study on new generation computational intelligence approaches in DO modeling.

2. Methodology

2.1. Multilayer perceptron

The MLP neural network, which is a feed forward neural network with one or more layers between input and output layer, is the second most flexible mathematical structure patterned after the biological nervous system. It is a massive parallel system composed of many processing elements connected by links of variable weights (Lippman, 1987). The feed-forward MLP among many ANN paradigms is by far the most popular, which usually uses the technique of error back propagation to train the network configuration. Feed forward means that data flows in one direction from input to output layer (forward). This type of network is trained with the back propagation learning algorithm. The MLPs are widely used for pattern classification, recognition, prediction and approximation. Multilayer perceptron can solve problems which are not linearly separable. Also, the activation function consists of a sigmoid function in the hidden layer and a linear function in the output layer. It has been reported that MLP with this configuration

Please cite this article in press as: Olyaie, E., et al., A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in Delaware River, Geoscience Frontiers (2016), http://dx.doi.org/10.1016/j.gsf.2016.04.007

are the most commonly used form, as they have improved extrapolation ability (ASCE, 2000; Partal and Cigizoglu, 2008). The mathematical expression of the MLP is as follow Eq. (1):

$$y_j = f\left(\sum_{i=1}^{N} w_{ji}x_i + b_j\right), \tag{1}$$

2.2. Radial basis function (RBF)

The Radial Basis Function (RBF) neural network is a feed-forward ANN, similar in topology to the MLP network (Fernando and Jayawardena, 1998). A RBF has an input layer, a hidden layer and an output layer. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron. RBF is also known as a localized receptive field network because the basic functions in the hidden layer produce a significant nonzero response to input stimulus only when the input falls within a small localized region of the input space (Lee and Chang, 2003). The RBF only has connection weights between the hidden layer and the output layer. These weight values could be obtained by the linear least-squares method, which gives an important advantage for convergence (Danandeh Mehr et al., 2014a and b). The Gaussian activation function is widely used as a RBF. The RBF can be considered as a special case of MLR. The RBF method does not perform parameter learning as in MLP.

The input layer sends copies of the input variables to each node in the hidden layer. The nodes in the hidden layer are each specified by a transfer function f, which transforms the incoming signals. For the pth input pattern x^p, the response of the j^{th} hidden node y_j is of the form

$$y_j = f\left\{\frac{||X^p - U_j||}{2\sigma_j^2}\right\} \tag{2}$$

where $||.|| =$ Euclidian norm; $U_j =$ center of the j^{th} radial basis function f; and $\sigma =$ spread of the RBF that is indicative of the radial distance from the RBF center within which the function value is significantly different from zero. The network output is given by a linear weighted summation of the hidden node responses at each node in the output layer. In this study, different numbers of hidden layer neurons are examined for the RBF models with a simple trial-and-error method. Detailed information about the RBF method can be obtained from Haykin (1998).

2.3. Linear genetic programming (LGP)

The state of the art GP is an evolutionary technique that automatically solves problems without pre-specified form or structure of the solution in advance (Koza, 1992). In other words, GP is a systematic, domain-independent method for getting computers to solve problems automatically starting from a high-level statement of what needs to be done (Poli et al., 2008). Unlike ANN, GP is self-parameterizing that builds model's structure without any user tuning (Danandeh Mehr et al., 2014a). GP differs from conventional linear, quadratic, or polynomial regression, which merely involve finding the numeric coefficients for a function whose form has been pre-specified (Lee et al., 1997). Individual solutions in GP are computer programs represented as parse trees (Fig. 1). The population of initial generation is typically generated through a random process. However, subsequent generations are evolved through the genetic operators of selection reproduction, crossover and mutation (Babovic and Keijzer, 2002). The major inputs for a GP model are (1) patterns for learning, (2) fitness function (e.g., minimizing the squared error), (3) functional and terminal sets, and (4) parameters for the genetic operators like the crossover and mutation probabilities (Sreekanth and Datta, 2011). As it is shown in Fig. 1, in GP modeling, the functions and terminals are chosen randomly from the user defined sets to form a computer model in a tree-like structure with a root node and branches extending from each function and ending in a leaf or a terminal. In many cases in GP, leaves are the inputs to the program. More details on GP can be obtained from Koza (1992) and Babovic and Keijzer (2000).

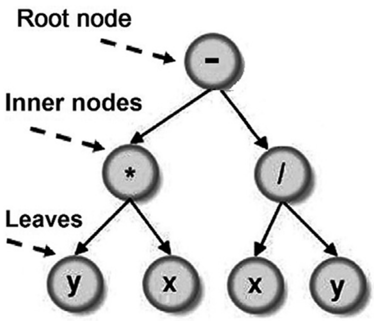

Figure 1. Tree representing of function ((x.y−x/y)).

Besides the tree-based GP, which is also referred to as the traditional Koza-style GP, there are new variants of GP such as LGP, Multi-Expression Programming (MEP), Gene Expression Programming (GEP), Cartesian Genetic Programming (CGP) and Grammatical Evolution (GE). All of these variants have a clear distinction in their genotype of a program (or an individual). Comparing to Koza-style GP, LGP is an advancement (Brameier and Banzhaf, 2001) with some main differences. The tree-based programs used in GP correspond to expressions from a functional programming language. Functions are located at root and inner nodes while the leaves hold input values or constants. In contrast, LGP denotes a GP variant that evolves sequences of instructions from an imperative programming language (like C) or machine language (Brameier and Banzhaf, 2007). The word "linear" refers to the imperative program representation. It does not mean that the method provides linear solutions (Danandeh Mehr et al., 2014a). An example of an LGP evolved program, which is the C code of the model developed for the second scenario of this study, is illustrated as follows:

```
L0 :  f [0] + =  Input000;
L1:  f [0] * =  1.21617;
L2:  f [0] * =  1.21617;
L3:  f [3] − =  f [0];
L4:  f [0] − =  −1.90761;
L5:  f [0] * =  −0.96368;
L6:  f [0] − =  f [3];
L7:  f [0] * =  f [0];
L8:  f [0] + =  f [0];
L0 :  f [0] + =  Input001;
```

Where the f[0] represents the temporary computation variable created in the program by LGP. The LGP uses such temporary computation variables to store values while performing calculations. The variable f[0] is initialized to be one in this program and the output is the value remaining in f[0] in the last line of the code.

It should be mentioned that in the abovementioned code, evolving introns has been removed. In analogy with nature introns, DNA segments in genes with information that is not expressed in proteins, an intron in LGP is defined as a program part without any influence on the calculation of the output(s) for all possible inputs. Two rather simple examples of introns are as follows:

(1) L0 : f [0] + = 0.0;
(2) L0 : f [0] + = −1.00f;
L1: f [0] + = +1.00f;

The program structure in LGP allows introns to be detected and removed much easier than in tree-based GP. Similar to pseudo-algorithm of any GP variants, LGP generally solves any problem through the six steps: (1) generation of an initial population (machine-code functions) randomly by the user defined functions and terminals; (2) selection of two functions from the population randomly, Comparison of the outputs and designation of the function that is more fit as winner_1 and less fit as loser_1; (3) selection of two other functions from the population randomly and designation of the winner_2 and loser_2; (4) application of transformation operators to winner_1 and winner_2 to create two similar; However, different evolved programs (i.e. offspring) as modified winners (5) replace the loser_1 and loser_2 in the population with modified winners and (6) repetition of steps (1)–(5) until the predefined run termination criterion.

Similar to GP, the user specified functions (or instruction set) in LGP can composed of arithmetic operations (+, −, ×, ÷), Boolean logic functions, conditionals, and any other mathematical functions such Sin, Ln, EXP, Sqrt, and others. The choice of the functional set determines the complexity of the evolved program. For example, a functional set with only addition and subtraction results in a linear model whereas a functional set which includes trigonometric functions may result in a highly nonlinear model (Danandeh Mehr et al., 2014a,b). The terminal set contains the arguments for the functions and can consist of numerical constants, logical constants, variables, etc. More details on the application of LGP in predictive modeling can be obtained from Poli et al. (2008). The LGP modeling attributes used for experimental setup in this study are explained in Table 1.

2.4. Support vector machine (SVM)

The idea of SVM which is one of the classes of soft-computing techniques in recent years, has been proposed by Vapnik (1995). Originally SVM was developed for solving classification problems; then its usage has been extended to regression-type applications for function estimation (Vapnik, 1995). The SVMs used for regression modeling estimate one output variable based on a set of input variables. The formulation includes the SRM principle, which has been shown to be superior to the traditional empirical risk minimization (ERM) principle, employed by conventional neural networks (Khan and Coulibaly, 2006). SRM minimizes an upper bound on the expected risk, as opposed to ERM which minimizes the error on the training data (Wankhede and Doye, 2005). It is the difference which equips SVM with a greater ability to generalize, which is the goal in statistical learning (Gunn, 1998). As being supervised learning method, SVM uses training dataset to develop a model. Fig. 2 exhibits the basic concept of SVM. There exist uncountable decision functions, i.e. hyperplanes, which can effectively separate the negative and positive data set (denoted by '×' and 'O', respectively), that has the maximal margin. This indicates that the distance from the closest positive samples to a hyperplane and the distance from the closest negative samples to it will be maximized.

Table 1
Parameter settings for LGP.

Parameter	Value
Initial Population (programs)	100
Mutation Rate	5%
Crossover Rate	50%
Initial Program Size	80
Maximum Program Size	512
Maximum Numbers of Runs	300
Generation Without Improvement	300
Maximum Generation Since Start	1000
Terminal set	{DO, [−1,1]}
Instruction set	{+, −, ×, ÷, Sin, Cos}

As abundant papers and books provide a detailed introduction about the theory of SVM technique (Vapnik, 1998; Gao et al., 2001; Lin et al., 2006; Karamouz et al., 2009), thus a brief description of support vector regression is presented here.

Based on N training data $\{(\chi_i, d_i)\}_i^N$ (χ_i represents input vector, d_i means the desired value and N is the total number of training data), the SVM estimator on regression is expressed as follows (Huang et al., 2014):

$$y = f(x) = W_i\phi_i(\chi) + b \tag{3}$$

where ϕ_i is a nonlinear transfer function mapping the input vectors into a high dimensional feature space, and W_i represents a weight vector and b denotes a bias. The coefficients (W_i and b) can be reckoned by minimizing the following regularized risk function (Vapnik, 1995, 1998):

$$r(C) = C\frac{1}{N}\sum_{i=1}^{N} L_\varepsilon(d_i, y_i) + \frac{1}{2}\left|\left|W\right|\right|^2 \tag{4}$$

where

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon\,, & if\ |d - y| \geq \varepsilon \\ 0, & otherwise \end{cases} \tag{5}$$

In Eq. (4), the first part is the empirical risk which is measured by Eq. (5). $L_\varepsilon(d, y)$ stands for the $\varepsilon -$ insensitiveloss function. When the forecast value is within the $\varepsilon -$ tube; then the loss value is zero. The second part is used to measure the flatness of the function. C is called the regularized constant determining the degree of the empirical error in the optimization problem. Once the value of C increase, then a relative importance of the empirical risk concerning the regularization term will increase. ε is marked as the error tolerance which is equal to the approximation accuracy of the training process. ξ and ξ^* are denoted as positive slack variables penalizing the training errors by the loss function within the error tolerance ε. After that, Eq. (5) is converted to the following constrained form.

$$Minimize: \ \frac{1}{2}\left|\left|W\right|\right|^2 + C\left(\sum_{i}^{N}\left(\xi_i + \xi_i^*\right)\right) \tag{6}$$

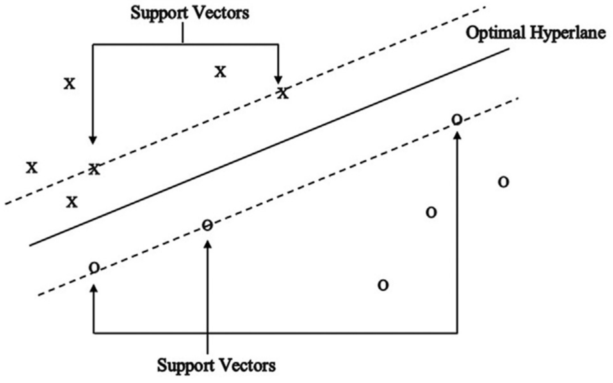

Figure 2. The basis of the support vector machines.

Please cite this article in press as: Olyaie, E., et al., A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in Delaware River, Geoscience Frontiers (2016), http://dx.doi.org/10.1016/j.gsf.2016.04.007

$$\text{Subject to}\begin{cases} W_i\phi(\chi_i)+b_i-d_i\le\varepsilon+\xi_i^* \\ d_i-W_i\phi(\chi_i)-b_i\le\varepsilon+\xi_i \\ \xi_i,\xi_i^*,\quad i\,=\,1,2,3,\;\ldots,\;N \end{cases} \tag{7}$$

For solving the constrained optimization problem above, the primal Lagrangian form is employed, whose formula is as follows:

$$\begin{aligned} L=\frac{1}{2}\left\|W\right\|^2+C\left(\sum_{i=1}^{N}\left(\xi_i+\xi_i^*\right)\right)-\sum_{i=1}^{N}\alpha_i(W_i\phi(\chi_i)+b-d_i+\varepsilon+\xi_i) \\ -\sum_{i=1}^{N}\alpha_i^*\left(d_i+W_i\phi(\chi_i)-b+\varepsilon+\xi_i^*\right)-\sum_{i=1}^{N}\left(\beta_i\xi_i+\beta_i^*\xi_i^*\right) \end{aligned} \tag{8}$$

Eq. (8) is minimized corresponding to primal variables W_i, b, ξ and ξ^*, and maximized corresponding to the positive Lagrangian multipliers α_i^* and β_i^*. Finally, Karush–Kuhn–Tucker conditions are employed to the regression, and Eq. (9) also has a dual Lagrangian form as follows:

$$\begin{aligned} \vartheta\left(\alpha_i,\alpha_i^*\right)\;=\;\sum_{i=1}^{N}d_i\left(\alpha_i-\alpha_i^*\right)-\varepsilon\sum_{i=1}^{N}\left(\alpha_i+\alpha_i^*\right) \\ +\frac{1}{2}\sum_{i=1,\,j=1}^{N}\left(\alpha_i-\alpha_i^*\right)\left(\alpha_j-\alpha_j^*\right)K\left(\chi_i,\chi_j\right) \end{aligned} \tag{9}$$

Subject to $\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)\;=\;0$ and $\alpha_i,\alpha_i^*\in[0,C]$, $i=1,2,3,\ldots,N$. In Eq. (8), the Lagrange multipliers meet the equality $\alpha_i{*}\alpha_i^*\;=\;0$. The Lagrange multipliers α_i and α_i^* are computed, and then calculating the optimal desired weight vector of the regression hyperplane as follows:

$$W^*\;=\;\sum_{i=1}^{N}\left(\alpha_i-\alpha_i^*\right)K(\chi,\chi_i) \tag{10}$$

Therefore, the regression function can be expressed as follows:

$$f\left(\chi,\alpha,\alpha^*\right)\;=\;\sum_{i=1}^{N}\left(\alpha_i-\alpha_i^*\right)K(\chi,\chi_i)+b \tag{11}$$

where $K(\chi,\chi_i)$ represents the Kernel function which is defined as follows:

$$K\left(\chi_i,\chi_j\right)\;=\;\varphi(\chi_i)^*\varphi\left(\chi_j\right) \tag{12}$$

A function meeting Mercer's condition (Vapnik, 1998) can be used as the Kernel function. In this study, RBF is employed as the kernel function:

$$K\left(\chi,\chi_j\right)\;=\;exp\left(-\left\|\chi-\chi_j\right\|^2/2\sigma^2\right) \tag{13}$$

where σ represents the Gaussian noise level of standard deviation.

2.5. Models performance criteria

Some techniques are recommended for model performance evaluation of hydrological time series forecasting according to published literature related to calibration, validation, and application of hydrological models. The considered statistical measures were coefficient of correlation (R), root mean squared error (RMSE), Nash–Sutcliffe efficiency coefficient (NS) and mean absolute relative error (MARE). The R measures the degree to which two variables are linearly related. RMSE and MARE provide different types of information about the predictive capabilities of the model. The RMSE measures the goodness-of-fit relevant to high DO values whereas the MARE not only gives the performance index in terms of predicting DO but also the distribution of the prediction errors.

Correlation coefficient describes the degree of collinearity between simulated and measured data, which ranges from −1 to 1 and is an index of the degree of linear relationship between observed and simulated data. If $R=0$, no linear relationship exists. If $R\,=\,1$ or −1, a perfect positive or negative linear relationship exists. Its equation is:

$$R\;=\;\frac{\sum_{i=1}^{n}\left(\mathrm{DO}_o(i)-\overline{\mathrm{DO}_o}\right)\left(\mathrm{DO}_f(i)-\overline{\mathrm{DO}_f}\right)}{\sqrt{\sum_{i=1}^{n}\left(\mathrm{DO}_o(i)-\overline{\mathrm{DO}_o}\right)^2\sum_{i=1}^{n}\left(\mathrm{DO}_f(i)-\overline{\mathrm{DO}_f}\right)^2}} \tag{14}$$

The root mean square error (RMSE) can be calculated as follows:

$$\mathrm{RMSE}\;=\;\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\mathrm{DO}_f(i)-\mathrm{DO}_o(i)\right)^2} \tag{15}$$

The Nash–Sutcliffe model efficiency coefficient (NS) is a good alternative to R or R^2 as a 'goodness-of-fit' or relative error measure in which it is sensitive to differences in the observed and forecasted means and variances (Olyaie et al., 2015):

$$\mathrm{NS}\;=\;1-\frac{\sum_{i=1}^{n}\left(\mathrm{DO}_o(i)-\mathrm{DO}_f(i)\right)^2}{\sum_{i=1}^{n}\left(\mathrm{DO}_o(i)-\overline{\mathrm{DO}_o}\right)^2} \tag{16}$$

The MARE can be calculated as follows:

$$\mathrm{MARE}\;=\;\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\mathrm{DO}_f(i)-\mathrm{DO}_o(i)}{DO_f(i)}\right|\times 100 \tag{17}$$

where $\mathrm{DO}_o(i)$ and $\mathrm{DO}_f(i)$ are, respectively, the observed and forecasted DO and $\overline{\mathrm{DO}_o}$; $\overline{\mathrm{DO}_f}$ denote their means, and n is the number data points considered.

3. Study area and statistical analysis of data

3.1. Study area

The daily data obtained from the Delaware River gauging station at Trenton City, Mercer County, NJ, Hydrologic Unit 01463500, on left bank 450 ft upstream from Calhoun Street Bridge at Trenton, 0.5 mi upstream from Assunpink Creek, 0.9 mi north of Morrisville, PA, and at river mile 134.5. This station (USGS Station No: 01463500, drainage Area (sq. mi.): 6,780, 74°46′41″W and 40°13′18″N) in New Jersey State, operated by the U.S. Geological Survey (USGS), were employed to train and test all the models developed in this paper. Fig. 3 shows the gauging station. Because the data of some variables were unrecorded, these data were removed from the dataset. Therefore, the last dataset includes 2063 samples for the station. In this context, there were two phases, the Training and testing.

In the gauging Station, the data from July 1, 2007 to June 1, 2012 (5 years; i.e. 75% of total data) and the data from June 1, 2012 to January 1, 2014 (2 year; i.e. 25% of total data) were used for training and testing sets, respectively. For this station, the daily time series of pH, electrical conductivity (EC), temperature (T) and river discharge (Q), and dissolved oxygen (DO) were downloaded from the web server of the USGS. The consideration of the first 6 years of the DO and independent parameters' time series for the training/

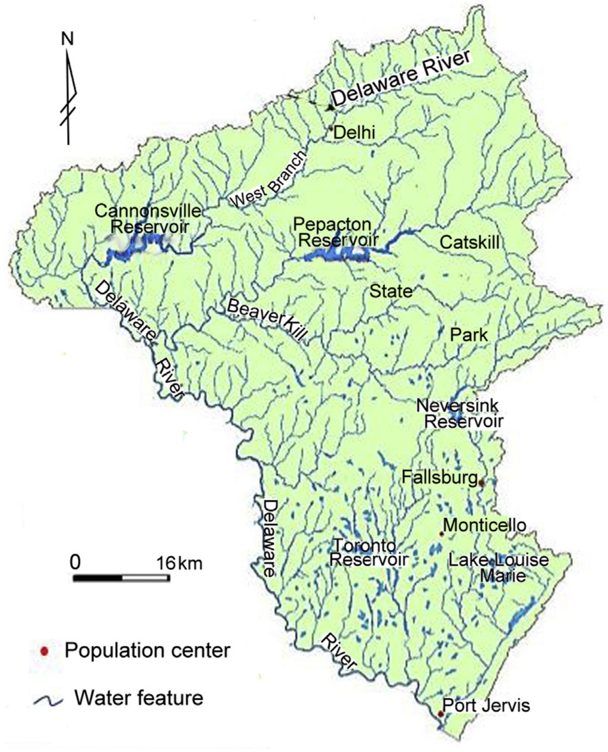

Figure 3. Delaware River basin.

calibration set has two advantages; first, the highest observed DO and used other parameters occurred during this period and second, significant variations could be possible.

3.2. *Statistical analysis of data*

Some statistical properties of all data sets for Delaware River stations are given in Table 2, which include the mean, standard deviation (S_d), skewness coefficient (C_{sx}), minimum, maximum of data. From these tables, it could be observed that the extreme values of DO and other variations were in the training set. When dividing the data set into training and testing subsets, it is essential to check that the data subsets represent the same statistical population (Masters, 1993). In general, Table 2 illustrated relatively similar statistical characteristics between training and testing sets in terms of mean, standard deviation and skewness coefficient.

Skewness coefficients were low for data sets (see the C_{sx} values). This is appropriate for modeling, because high skewness coefficient has a considerable negative effect on ANN performance (Altun et al., 2007). The correlation coefficients between observed time series are computed in order to obtain suitable input pattern for MLP, RBF, LGP and SVM models. The results for the station are shown in Table 3. The correlation coefficient (ρ) between DO and other used variations, which for n pairs are available, is defined as:

Table 2
Statistics analysis of entire hydro-chemical data sets for Delaware River station.

Statistical parameters	Q (m^3/s)	EC (μS/cm)	T (°C)	pH	DO (mg/L)
Mean	383.15	196.00	13.99	8.09	10.79
S_d	361.3638	38.59282	8.905241	0.525374	2.157795
C_{sx}	3.7890	0.042641	0.066737	0.686083	0.194058
Min	71.64	89	−0.2	6.5	6.4
Max	4728.90	372	30	9.8	16.2

Table 3
The correlation coefficients between measured DO and other input parameters.

Time series	All data	Training set	Testing set
T	0.939	0.935	0.942
Q	0.872	0.880	0.869
pH	0.795	0.798	0.767
EC	0.516	0.527	0.591

$$\rho = \frac{\sum_{i=1}^{n}\left(\mathrm{DO}_i - \overline{\mathrm{DO}}\right)(X_i - \overline{X})}{\sqrt{\sum_{i=1}^{n}\left(\mathrm{DO}_i - \overline{\mathrm{DO}}\right)^2 \sum_{i=1}^{n}(X_i - \overline{X})^2}} \tag{18}$$

where X is independent parameter and the bar denotes the mean of the variable. The higher values of correlation coefficient, which range from 0 to1, indicate better agreement between the variables. To make a suitable selection of model input variables, the autocorrelation and cross correlation between DO and independent parameters data were investigated. As can be seen from Table 3, the correlation between DO and Q, pH, EC and surely the correlation between DO and T are relatively high; therefore, DO was related to all of those mentioned parameters in the models.

4. Data pre-processing

In order to reach affective network training, the data are needed to be normally distributed by a suitable conversion approach. Luk et al. (2000) reported that networks trained on converted data show better efficiency and faster convergence. Besides, Aqil et al. (2007) showed that the data preprocessing with log sigmoidal activation function before processing the MLP and RBF models. In this study, conversion is carried out on all time-series data independently by the below equation:

$$z = a\log_{10}(G + b) \tag{19}$$

where z is the converted value of DO, a is an arbitrary constant, and b was set to 1 to avoid the entry of zero DO in the log function. The ultimate predict results were then back converted using the following equation:

$$G = 10^{z/a} - b \tag{20}$$

4.1. *Model input selection*

As mentioned in the previous sections, for estimating DO, the river parameters considered as input parameters in this study are pH, electrical conductivity (EC), temperature (T) and river discharge (Q). The present study examines various combinations of these four parameters as inputs to the applied models so as to evaluate the degree of the effect of each of these variables on sediment load. The research aims at exposure of DO estimation by computational intelligence models which include MLP, RBF, LGP and SVM, using implicit recorded river data. Identifying the best input combination is the most important step of any modeling. As indicated in Fig. 4, the complexities of the model including the higher number of inputs, more data for training model, a model with greater parameters, may have less prediction error; however, it is not necessarily ensure fewer errors at the test phase. In this condition, there is an optimal condition in which prediction errors are minimized at the test phase (Bray and Han, 2004). Considering the statistical

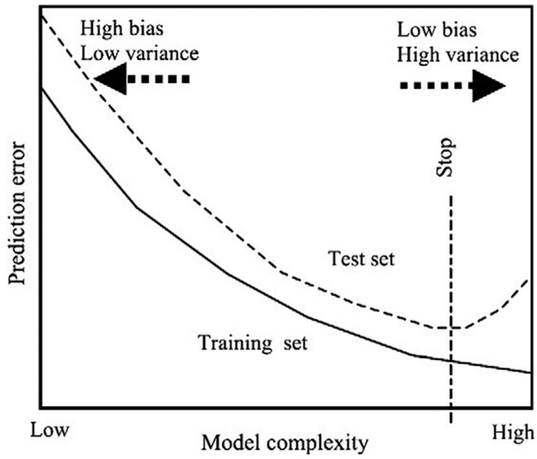

Figure 4. The impact of model complexity on accuracy of the results (adopted from Bray and Han, 2004).

analysis, several optimal input combinations were tried to estimate the DO. The input combinations evaluated in this study are: (1) T, (2) T and Q, (3) T and pH, (4) T and pH and EC, (5) T and pH and Q and (6) T and pH and EC and Q. In all cases, the output layer has only one neuron, the river DO concentration (DO).

5. Results and discussion

The MLP, RBF, LGP and SVM models with different inputs were compared based on their performance in training and testing sets. The results were summarized in Tables 4–7. It was apparent that all of the performances of these models are almost similar during training as well as validation. In order to get an effective evaluation of the MLP, RBF, LGP and SVM models performance, the best model structures, has been used to compare the models. From the best fit model, it was found that the difference between the values of the statistical indices of the training and testing set does not vary substantially. It was observed that all six models generally gave low values of the RMSE and MARE as well as high *R* and NS, the performances of the MLP, RBF, LGP and SVM models performance in the DO forecasting were satisfactory.

Table 4
The structure and the performance statistics of the MLP models during the training and testing periods.

Input	Topology	Training				Testing			
		R	RMSE	MARE	NS	*R*	RMSE	MARE	NS
T	1,2,1	0.897	1.206	7.728	0.760	0.905	1.063	7.517	0.791
T, pH	2,5,1	0.945	0.628	5.986	0.904	0.952	0.604	4.984	0.909
T, Q	2,3,1	0.888	1.462	11.432	0.683	0.893	1.234	10.120	0.733
T, pH, EC	3,2,1	0.890	1.357	10.935	0.719	0.898	1.170	9.521	0.762
T, pH, Q	3,4,1	**0.949**	0.614	**4.737**	**0.906**	**0.957**	**0.592**	**4.967**	**0.918**
T, pH, Q, EC	4,8,1	0.948	**0.611**	4.921	0.901	0.955	0.594	4.971	0.913

The bold numbers are relatively high values for *R* and CE and relatively low values for RMSE and MAPE.

Table 5
The structure and the performance statistics of the RBF models during the training and testing periods.

Input	Structure	Training				Testing			
		R	RMSE	MARE	NS	*R*	RMSE	MARE	NS
T	1,0.5,1	0.889	1.371	11.082	0.692	0.894	1.330	10.107	0.752
T, pH	2,0.3,1	0.891	1.322	10.944	0.724	0.908	1.241	8.222	0.792
T, Q	2,0.3,1	0.887	1.376	11.610	0.680	0.894	1.329	10.119	0.755
T, pH, EC	3,0.5,1	0.883	1.380	13.258	0.601	0.892	1.330	10.320	0.731
T, pH, Q	3,0.6,1	0.898	**1.250**	**10.289**	**0.765**	**0.919**	**0.994**	8.556	**0.798**
T, pH, Q, EC	4,1.2,1	**0.899**	1.253	10.290	0.764	0.911	1.033	**7.191**	0.794

The bold numbers are relatively high values for *R* and CE and relatively low values for RMSE and MAPE.

Table 6
The structure and the performance statistics of the LGP models during the training and testing periods.

Input	IB[a]	Training				Testing			
		R	RMSE	MARE	NS	*R*	RMSE	MARE	NS
T	+, −, ×, Sin, Cos	0.939	0.754	5.346	0.881	0.973	0.473	3.334	0.945
T, pH	+, −, ×	0.949	0.712	5.054	**0.936**	0.972	0.500	3.686	0.956
T, Q	+, −, ×, Sin	0.971	**0.551**	4.020	0.894	0.981	0.420	2.907	0.938
T, pH, EC	+, −, ×, Sin, Cos	0.972	0.577	4.190	0.930	0.981	0.399	2.830	0.960
T, pH, Q	+, −, ×, Sin, Cos	**0.968**	0.591	**3.967**	0.925	**0.983**	**0.374**	**2.501**	**0.965**
T, pH, Q, EC	+, −, ×, Sin, Cos	0.967	0.677	4.68	0.903	0.973	0.499	3.574	0.938

The bold numbers are relatively high values for *R* and CE and relatively low values for RMSE and MAPE.
[a] IB denotes the instructions have been used in the best program.

Table 7
The structure and the performance statistics of the SVM models during the training and testing periods.

Input	Parameter (*C*, γ)	Training				Testing			
		R	RMSE	MARE	NS	*R*	RMSE	MARE	NS
T	2,0.25	0.968	0.594	4.174	0.912	0.983	0.376	2.610	0.966
T, pH	2,0.5	0.971	0.552	4.174	0.896	0.980	0.422	3.004	0.936
T, Q	2,0.25	0.975	0.480	3.506	0.949	0.977	0.453	3.381	0.952
T, pH, EC	16,0.125	0.976	0.470	3.331	0.952	0.978	0.454	3.379	0.961
T, pH, Q	16,0.5	**0.977**	**0.462**	**3.251**	**0.955**	**0.987**	**0.330**	**2.278**	**0.973**
T, pH, Q, EC	36,0.25	0.977	0.464	3.250	0.953	0.986	0.332	2.278	0.972

The bold numbers are relatively high values for *R* and CE and relatively low values for RMSE and MAPE.

The optimal architecture of the ANN models and its parameter variation were determined based on the minimum value of the mean squared error (MSE) of the training and testing sets. For the RBF models, the optimal spread coefficients and number of hidden layers were calculated using the trial-and-error method. For MLP models, the logsig and purelin functions were respectively found to be optimal activation functions for the hidden and output layers. In the models, the number of iterations was 10,000 and the optimal number of neurons in the hidden layer was obtained as 1. With increase in number of neurons, the networks yielded several local minimum values with different MSE values for the training set. Selection of an appropriate number of nodes in the hidden layer is very important aspect as a larger number of these may result in over-fitting, while a smaller number of nodes may not capture the information adequately. Fletcher and Goss (1993) suggested that the appropriate number of nodes in a hidden layer ranges from $(2n^{1/2}+\mathrm{m})$ to $(2n+1)$, where n is the number of input nodes and m is the number of output nodes, so this range was used to determine the optimal number of hidden layer. Subsequently, six scenarios were developed using various input combinations of daily T, pH, EC, and Q.

A model can be claimed to produce a perfect estimation if the NS criterion is equal to 1. Normally, a model can be considered as accurate if the NS criterion is greater than 0.8 (Shu and Ouarda, 2008). It can be observed from Tables 4–7 that the NS values for various applied computational intelligence methods in this study are over 0.7. This indicates that they had good performance during both training and validation and these models achieved acceptable results. It also showed that the SVM model with input combination of T, pH, Q and EC, which consisted of these data in input, had the smallest value of the RMSE as well as higher value of *R* and NS in the training as well as validation period, so, it was selected as the best-fit model for predicting the DO in this study. Also, the NS values for the SVM model predict of the DO value were higher than those for the MLP, RBF and LGP models, which indicates that the overall quality of estimation of the SVM model is better than the ANNs and LGP models according to NS. Compared with the MLP, RBF, LGP and SVM models perform from the RMSE and *R* viewpoints, the SVM model performed a bit better than both the ANNs and the LGP model. Concretely, SVM model produced a lower RMSE as well as higher R, is the former being the best according to the criteria. Thus, in the Testing phase, as seen in Tables 4–7, the values with the SVM model prediction were able to produce a good, near forecast, as compared to those with other models, whilst it can be concluded that the SVM model obtained the best relative error between the observed and modeled DO. Furthermore, as can be seen from Tables 4–7 that the virtues or defect degrees of forecasting accuracy are different in terms of different evaluation measures during the training phase and the testing phase. SVM model is able to obtain the better forecasting accuracy in terms of different evaluation measures not only during the training phase but also during the validation phase.

It appears that while assessing the performance of any model for its applicability in predicting DO, it is not only important to evaluate the average prediction error but also the distribution of prediction errors. The statistical performance evaluation criteria employed so far in this study are global statistics and do not provide any information on the distribution of errors. Therefore, in order to test the robustness of the model developed, it is important to test the model using some other performance evaluation criteria such as mean absolute relative error (MARE). The MARE index provides an indication about whether a model tends to overestimate or underestimate. The analysis based on the MARE index suggests that the SVM model performed better than the ANNs and LGP model. This indicates that the errors obtained when using the SVM model are more symmetric around zero but show more dispersion than those obtained when using the ANNs and LGP model. The

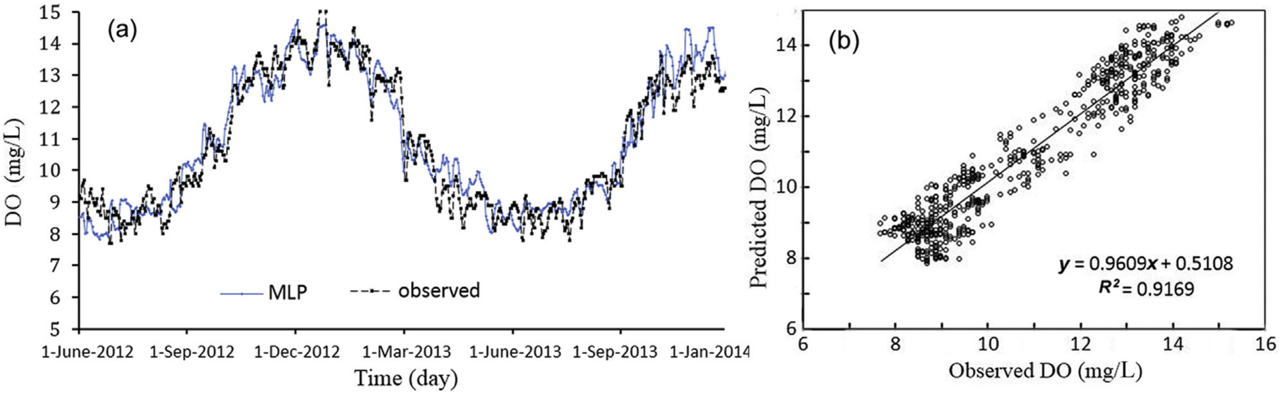

Figure 5. Observed and predicted DO values by optimal MLP in the testing period.

Please cite this article in press as: Olyaie, E., et al., A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in Delaware River, Geoscience Frontiers (2016), http://dx.doi.org/10.1016/j.gsf.2016.04.007

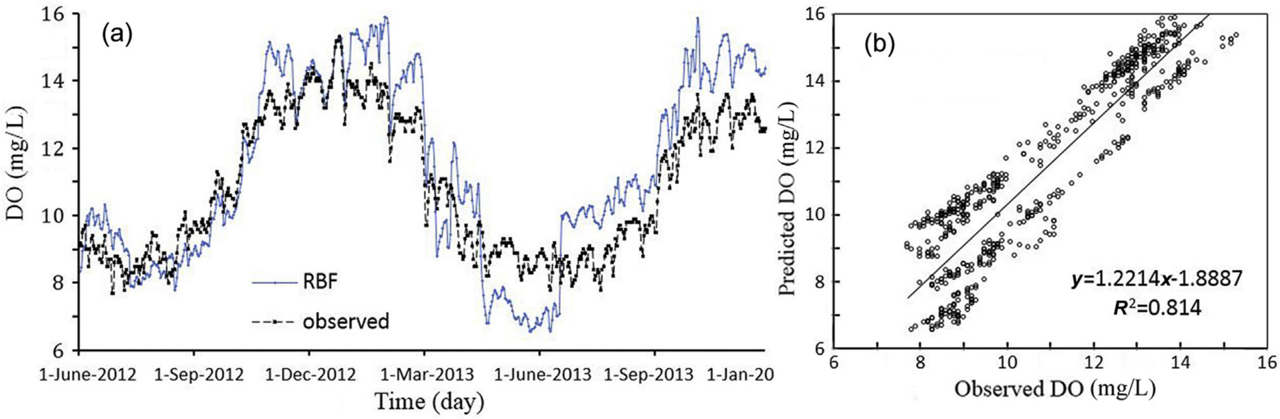

Figure 6. Observed and predicted DO values by optimal RBF in the testing period.

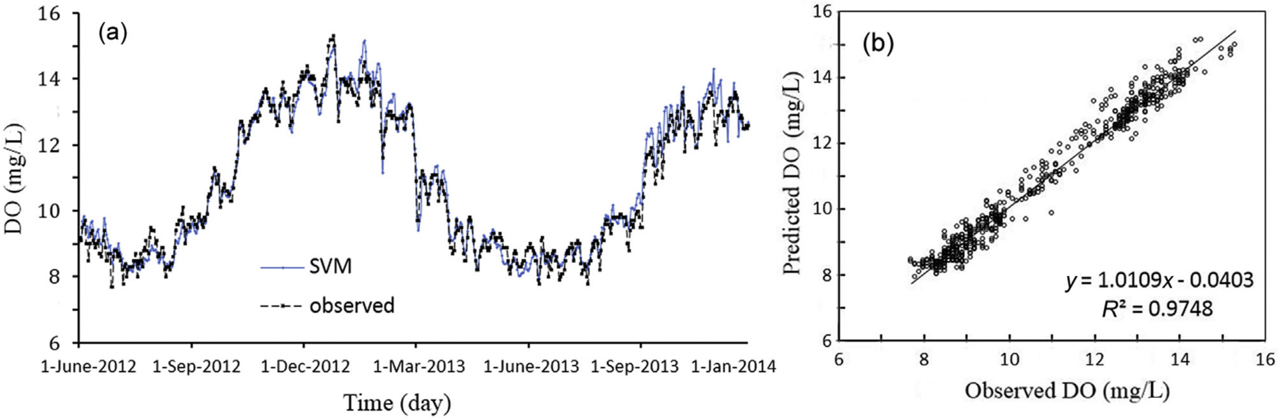

Figure 7. Observed and predicted DO values by optimal SVM in the testing period.

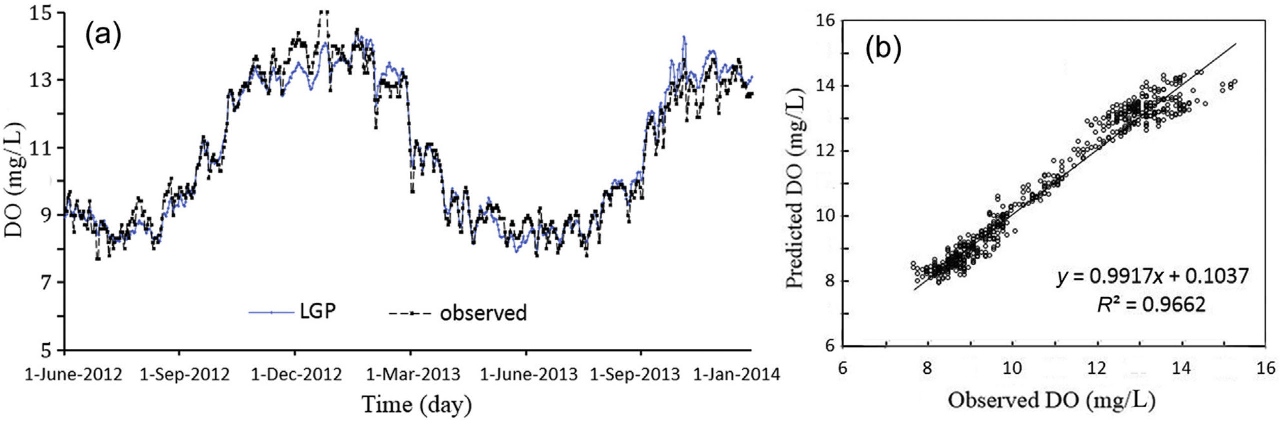

Figure 8. Observed and predicted DO values by optimal LGP in the testing period.

performances of all prediction models developed in this paper during the validation period in the study site are shown in Figs. 5–8.

It was obviously seen from the figures that the SVM estimates were closer to the corresponding observed DO values than those of the other models. As seen from the fit line equations (assume that the equation is $y = ax + b$) in the scatter plots that a and b coefficients for the SVM model are, respectively, closer to the 1 and 0 with a higher R value than ANNs and LGP models.

Overall, SVM model gave good prediction performance and were successfully applied to establish the forecasting model that could provide accurate and reliable DO prediction. The results suggested that the SVM model was superior to the other in this forecasting. The reason for a better prediction accuracy of SVM model than other models primarily lied in the shortcoming of the models, e.g. slowly learning speed, over-fitting, curse of dimensionality and convergence to local minimum. Conversely, SVM model was based on the empirical risk minimization principle, which could attack the problem in theory.

Since programs evolved by LGP may be expressed by explicit formulations, mathematical relationships between input and output variables, they are preferable for practical use. In addition, the evolved formulations may be beneficial for mining the knowledge from the information contained in the field data.

Considering all the aforementioned results and the concept of "simplicity and applicability" as the main issue of hydrological modeling, the model with T and pH as input combination (Eq. (21)), has been evaluated as the best scenario for DO prediction by LGP in this study.

$$\mathrm{DO} = 2(0.053714T - 1.838331)^2 + \mathrm{pH} \tag{21}$$

Evidently, the empirical relation given in Eq. (21) represents DO process more understandable than a matrix of weights and biases generated by MLP, RBF or set of hyperplanes in a high- or infinite-dimensional space produced by SVM. However, this type of equations sometimes shows such a complexity that cannot be easily interpreted. This issue can be considered as major disadvantage of GP-based modeling that indicates a necessity for further studies to overcome such problems.

Since the feasible estimation of the peak values was usually the most important factor in any river water management program, another key point when comparing different models was the capability of the models in estimating peak values. For this purpose, peak values were sampled by considering the threshold of the top 5% of the data from the original DO time series. The performances of the various models for this modeling were evaluated using Eq. (22) and were presented in Table 8.

$$R^2_{\mathrm{Peak}} = 1 - \frac{\sum_{i=1}^{n}\left(\mathrm{DO}_f(i) - \mathrm{DO}_o(i)\right)^2}{\sum_{i=1}^{n}\left(\mathrm{DO}_o(i) - \overline{\mathrm{DO}}_o(i)\right)^2} \tag{22}$$

where R^2_{Peak} is the determination coefficient for peak values, n is number of peak values, $\mathrm{DO}_o(i)$ and $\mathrm{DO}_f(i)$ are the observed and forecasted DO and $\overline{\mathrm{DO}_o}$ denotes the mean observed data for peak values (Nourani et al., 2011), respectively.

According to Table 8, it could be concluded that the SVM was more efficient than other models (i.e., MLP, RBF and LGP) in monitoring peak values. For the other evaluation, the two by two comparisons of the models for modeling of extreme values had been presented in Table 9 percentile. For instance, the efficiency of SVM modeling was improved up to 26%, 13% and 5% compared to RBF, MLP and LGP respectively. It was obvious that the performance of the SVM model was also far superior to the other applied models in modeling peak values.

Table 8
The ability of different models in capturing peak values.

Determination coefficient for peak values ($R^2{}_{\mathrm{peak}}$)			
MLP	RBF	SVM	LG0050
0.862	0.730	0.991	0.934

Note: In this table the best result for each model has been presented.

Table 9
Comparison of different models in modeling peak values (in %).

Model	RBF	MLP	LGP	SVM
RBF	–	15.31	21.84	26.33
MLP		–	7.70	13.01
LGP			–	5.75
SVM				–

6. Conclusion

In the present study, various computational intelligence techniques (e.g., MLP, RBF, LGP and SVM), were compared to estimate DO concentration. Several input combinations comprising pH, electrical conductivity (EC), temperature (T) and river discharge (Q) were constructed. To achieve this objective, Delaware River Station in the USA was employed to develop various models investigated in this study. The methods utilized the statistical properties of the data series with certain amount of input variables. The obtained results indicated that soft computing methods were powerful tools to model the DO and could give good estimation performance. Therefore, the results of the study were highly encouraging and suggested that ANNs and SVM approaches were promising in modeling DO, and this might provide valuable reference for researchers and engineers who applied the methods for modeling long-term hydrological time series estimating. This study also addressed use of LGP for creating DO models on the basis of data, as well as in combination with empirical equation (i.e. taking advantage of knowledge about the problem domain). As next step, comparing the results of ANNs, LGP and SVM models, it was seen that the values of R and NS of SVM models were higher than those of ANNs and LGP models. Moreover, the RMSE values of SVM models were lower than those of ANNs and LGP models. Therefore, SVM model could improve the accuracy over the ANNs and LGP models. The results also demonstrated ANNs, LGP and SVM showed good prediction accuracy for low values of DO but were unable to maintain their accuracy for high values of DO. However, a significant improvement was observed for the SVM in the peak DO prediction compared to ANNs and LGP. Overall, the analysis presented in this study provides that the SVM method was superior to the ANNs and LGP in the DO forecasting. But from the standpoint of simplicity, LGP was found to be more applicable than other models for DO prediction. In general, implementation of all computational intelligence models in the present study illustrated the flexibility of DO modeling. It is hoped that future research efforts will focus in these directions, i.e. more efficient approach for training multi-layer perceptron of ANN model, improve the prediction accuracy, especially for the high values of DO, by combining or improving model parameters, the fine-tuning of algorithm for selecting more appropriate parameters of GP evolution, saving computing time or more efficient optimization algorithms in searching optimal parameters of SVM model etc., to improve the accuracy of the forecast models in terms of different evaluation measures for better planning, design, operation, and management of various engineering systems.

Acknowledgments

The data used in this study were downloaded from the U.S. Geological Survey (USGS) Web server. The authors would like to thank the staff of the USGS who are involved in the data observation, processing, and management of the USGS websites. The author also would like thank the editor and two anonymous reviewers for their constructive comments, which helped us to improve the paper.

References

Ahmed, A.A.M., Hossain, M.I., Rahman, M.T., Chowdhury, M.A.I., 2013. Application of artificial neural network models for predicting dissolved oxygen concentration

Please cite this article in press as: Olyaie, E., et al., A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in Delaware River, Geoscience Frontiers (2016), http://dx.doi.org/10.1016/j.gsf.2016.04.007

for Surma River, Bangladesh. Journal of Applied Technology in Environmental Sanitation 3, 135–140.

Altun, H., Bilgil, A., Fidan, B.C., 2007. Treatment of multi-dimensional data to enhance neural network estimators in regression problems. Expert Systems with Applications 32, 599–605.

Antanasijević, D., Pocajt, V., Povrenović, D., Perić-Grujić, A., Ristić, M., 2013. Modelling of dissolved oxygen content using artificial neural networks: Danube River, North Serbia, case study. Environmental Science and Pollution Research 20, 9006–9013.

Aqil, M., Kita, I., Yano, A., Nishiyama, S., 2007. A comparative study of artificial neural networks and neuro-fuzzy in continuous modeling of the daily and hourly behaviour of runoff. Journal of Hydrology 337, 22–34.

ASCE Task Committee on the application of ANNs in hydrology, 2000. Artificial neural networks in hydrology, I: preliminary concepts. Journal of Hydrologic Engineering 5, 115–123.

Ay, M., Kisi, O., 2012. Modeling of dissolved oxygen concentration using different neural network techniques in Foundation Creek, El Paso County, Colorado, USA. Journal of Environmental Engineering 138, 654–662.

Babovic, V., Keijzer, M., 2002. Rainfall runoff modeling based on genetic programming. Nordic Hydrology 33, 331–346.

Babovic, V., Keijzer, M., 2000. Genetic programming as a model induction engine. Journal of Hydroinformatics 2, 35–60.

Boano, F., Revelli, R., Ridolfi, L., 2006. Stochastic modelling of DO and BOD components in a stream with random inputs. Advances in Water Resources 29, 1341–1350.

Brameier, M., Banzhaf, W., 2001. A comparison of linear genetic programming and neural networks in medical data mining. IEEE Transactions on Evolutionary Computation 5, 17–26.

Brameier, M., Banzhaf, W., 2007. Linear Genetic Programming. Springer Science + Business Media, LLC, New York.

Bray, M., Han, D., 2004. Identification of support vector machines for runoff modelling. Journal of Hydroinformatics 6, 265–280.

Chen, W.B., Liu, W.C., 2014. Artificial neural network modeling of dissolved oxygen in reservoir. Environmental Monitoring and Assessment 186, 1203–1217.

Cimen, M., 2008. Estimation of daily suspended sediments using support vector machines. Hydrological Sciences Journal 53, 656–666.

Danandeh Mehr, A., Kahya, E., Olyaie, E., 2013. Streamflow prediction using linear genetic programming in comparison with a neuro-wavelet technique. Journal of Hydrology 505, 240–249.

Danandeh Mehr, A., Kahya, E., Yerdelen, C., 2014a. Linear genetic programming application for successive-station monthly streamflow prediction. Computers & Geosciences 70, 63–72.

Danandeh Mehr, A., Kahya, E., Özger, M., 2014b. A gene-wavelet model for long lead-time drought forecasting. Journal of Hydrology 517, 691–699.

Danandeh Mehr, A., Kahya, E., Şahin, A., Nazemosadat, M.J., 2014c. Successive-station monthly streamflow prediction using different ANN algorithms. International Journal of Environmental Science and Technology 12, 2191–2200.

Dibike, Y.B., Velickov, S., Solomatine, D., Abbott, M.B., 2001. Model induction with support vector machines: introduction and applications. Journal of Computing in Civil Engineering 15, 208–216.

Fernando, D.A.K., Jayawardena, A.W., 1998. Runoff forecasting using RBF networks with OLS algorithm. Journal of Hydrologic Engineering 3, 203–209.

Fletcher, D., Goss, E., 1993. Forecasting with neural networks: an application using bankruptcy data. Journal of Information Management 24, 159–167.

Gao, J.B., Gunn, S.R., Harris, C.J., Brown, M., 2001. A probabilistic framework for SVM regression and error bar estimation. Machine Learning 46, 71–89.

Garcia, A., Revilla, J.A., Medina, R., Alvarez, C., Juanes, J.A., 2002. A model for predicting the temporal evolution of dissolved oxygen concentration in shallow estuaries. Hydrobiology 475–476, 205–211.

Ghorbani, M., Khatibi, R., Aytek, A., Makarynskyy, O., 2010. Sea water level forecasting using genetic programming and artificial neural networks. Computers & Geosciences 36, 620–627.

Gunn, S.R., 1998. Support Vector Machines for Classification and Regression. Technical Report. University of Southampton, England, p. 66.

Guven, A., 2009. Linear genetic programming for time-series modeling pf daily flow rate. Journal of Earth System Science 118, 137–146.

Guven, A., Azamathulla, H.M., 2012. A comparative study of predicting scour around a circular pile. Institution of Civil Engineers Journal Maritime Engineering 165, 31–40.

Guven, A., Azamathulla, H.M., Zakaria, N.A., 2009. Linear genetic programming for prediction of circular pile scour. Journal of Oceanic Engineering 36, 985–991.

Guven, A., Kisi, O., 2013. Monthly pan evaporation modeling using linear genetic programming. Journal of Hydrology 503, 178–185.

Haykin, S., 1998. Neural Networks: A Comprehensive Foundation, second ed. Prentice-Hall, Upper Saddle River, NJ http://water.usgs.gov/osw/odrm/.

Huang, S., Chang, J., Huang, Q., Chen, Y., 2014. Monthly streamflow prediction using modified EMD-based support vector machine. Journal of Hydrology 511, 764–775.

Hull, V., Parrella, L., Falcucci, M., 2008. Modelling dissolved oxygen dynamics in coastal lagoons. Ecological Modelling 2, 468–480.

Kalff, J., 2002. Limnology: Inland Water Ecosystems. Prentice-Hall, Upper Saddle River, NJ.

Karamouz, M., Ahmadi, A., Moridi, A., 2009. Probabilistic reservoir operation using Bayesian stochastic model and support vector machine. Advances in Water Resources 32, 1588–1600.

Khan, M.S., Coulibaly, P., 2006. Application of support vector machine in Lake water level prediction. Journal of Hydrologic Engineering 11, 199–205.

Kisi, O., Akbari, N., Sanatipour, M., Hashemi, A., Teimourzadeh, K., Shiri, J., 2013. Modeling of dissolved oxygen in river water using artificial intelligence techniques. Journal of Environmental Informatics 22, 92–101.

Kisi, O., Guven, A., 2010. Evapotranspiration modeling using linear genetic programming technique. Journal of Irrigation and Drainage Engineering 136, 715–723.

Koza, J.R., 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA.

Lee, D.G., Lee, B.W., Chang, S.H., 1997. Genetic programming model for long term forecasting of electric power demand. Electric Power Systems Research 40, 17–22.

Lee, G.C., Chang, S.H., 2003. Radial basis function networks applied to DNBR calculation in digital core protection systems. Annals of Nuclear Energy 30, 1516–1572.

Lin, J.Y., Cheng, C.T., Chau, K.W., 2006. Using support vector machines for long-term discharge prediction. Hydrological Sciences Journal 51, 599–612.

Liong, S.Y., Sivapragasam, C., 2002. Flood stage forecasting with support vector machines. Journal of the American Water Resources Association 38, 173–186.

Lippman, R., 1987. An introduction to computing with neural nets. IEEE ASSP Magazine 4, 4–22.

Londhe, S., Charhate, S., 2010. Comparison of data-driven modelling techniques for river flow forecasting. Hydrological Sciences Journal 55, 1163–1174.

Luk, K.C., Ball, J.E., Sharma, A., 2000. A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. Journal of Hydrology 227, 56–65.

Marti, P., Shiri, J., Duran-Ros, M., Arbat, G., Ramirez de Cartagena, F., Puig-Bargués, J., 2013. Artificial neural networks vs. Gene Expression Programming for estimating outlet dissolved oxygen in micro-irrigations and filters fed with effluents. Computers and Electronics in Agriculture 99, 176–185.

Masters, T., 1993. Practical Neural Network Recipes in C++. Academic Press, San Diego (CA).

Nourani, V., Kisi, O., Komasi, M., 2011. Two hybrid Artificial Intelligence approaches for modeling rainfall–runoff process. Journal of Hydrology 402, 41–59.

Olyaie, E., Banejad, H., Chau, K.W., Melesse, A.M., 2015. Erratum to: A comparison of various artificial intelligence approaches performance for estimating suspended sediment load of river systems: a case study in United States. Environmental Monitoring and Assessment 187 (4), 187–189.

Partal, T., Cigizoglu, H.K., 2008. Estimation and forecasting of daily suspended sediment data using wavelet-neural networks. Journal of Hydrology 358, 317–331.

Poli, R., Langdon, W.B., McPhee, N.F., 2008. A field guide to genetic programming.Lulu.com, URL http://www.gp-field-guide.org.uk (With contributions by J. R. Koza).

Rankovic, V., Radulovc, J., Radojevic, I., Ostojic, A., Comic, L., 2010. Neural network modeling of dissolved oxygen in the Gruza reservoir, Serbia. Ecological Modelling 221, 1239–1244.

Schmid, B.H., Koskiaho, J., 2006. Artificial neural network modeling of dissolved oxygen in a wetland pond: the case of Hovi, Finland. Journal of Hydrologic Engineering 11, 188–192.

Shu, C., Ouarda, T.B.M.J., 2008. Regional flood frequency analysis at ungauged sites using the adaptive neuro-fuzzy inference system. Journal of Hydrology 349, 31–43.

Shukla, J.B., Misra, A.K., Chandra, P., 2008. Mathematical modeling and analysis of the depletion of dissolved oxygen in eutrophied water bodies affected by organic pollutants. Nonlinear Analysis: Real World Applications 9, 1851–1865.

Singh, K.P., Basant, A., Malik, A., Jain, G., 2009. Artificial neural network modeling of the river water quality-A case study. Ecological Modelling 220, 888–895.

Sivapragasam, C., Muttil, N., 2005. Discharge rating curve extension: a new approach. Water Resources Management 19, 505–520.

Sreekanth, J., Datta, B., 2011. Coupled simulation-optimization model for coastal aquifer management using genetic programming-based ensemble surrogate models and multiple-realization optimization. Water Resources Management 47, 1–17.

Traore, S., Guven, A., 2013. New algebraic formulations of evapotranspiration extracted from gene-expression programming in the tropical seasonally dry regions of West Africa. Irrigation Science 31, 1–10.

U.S. Geological Survey, 2015, National Water Information System data available on the World Wide Web (USGS Water Data for the Nation), accessed [Sep 10, 2015], at URL [http://waterdata.usgs.gov/nwis/].

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, New York.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.

Wankhede, P., Doye, D., 2005. Support vector machines for fingerprint classification. Proceedings of the Eleventh National Conference on Communications 356–360.

Wang, W.C., Chau, K.W., Cheng, C.T., Qiu, L., 2009. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. Journal of Hydrology 374, 294–306.

Wu, C.L., Chau, K.W., Li, Y.S., 2008. River stage prediction based on a distributed support vector regression. Journal of Hydrology 358 (1–2), 96–111.

YSI., 2009. The dissolved Oxygen handbook. In: We Know D.O. YSI Incorporated, p. 76.

Yu, P.S., Chen, S.T., Chang, I.F., 2006. Support vector regression for real-time flood stage forecasting. Journal of Hydrology 328 (3–4), 704–716.