HUNGARIAN AGRICULTURAL ENGINEERING
N° 27/2015 38-41
Published online: http://hae-journals.org/
HU ISSN 0864-7410 (Print) / HU ISSN 2415-9751(Online)
DOI: 10.17676/HAE.2015.27.38

PERIODICAL OF THE COMITTEE OF AGRICULTURAL AND BIOSYSTEM ENGINEERING OF
THE HUNGARIAN ACADEMY OF SCIENCES
and
SZENT ISTVÁN UNIVERSITY
Faculty of Mechanical Engineering

FORECASTING OF DISSOLVED OXYGEN IN THE RIVER DANUBE USING NEURAL NETWORKS

Author(s):
A. Csábrági[1] – S. Molnár[1] – P. Tanos[1] – J. Kovács[2]

Affiliation:
[1]Institute of Mathematics and Informatics, Szent István University, Páter K. street 1., Gödöllő, H-2100, Hungary
[2]Institute of Geography and Earth Sciences, Eötvös Loránd University, Pázmány Péter walkway 1/C., Budapest, H-1117, Hungary

Email address:
csabragi.anita@gek.szie.hu, molnar.sandor@gek.szie.hu, tanos.peter@gek.szie.hu kevesolt@iris.geobio.elte.hu

Abstract

The Danube is the second-largest river in Europe and the conservation of its water quality is very important because it influences the lives of millions people. The aim of this research is to predict one of the most important water quality parameters, dissolved oxygen, with the help of water pH, runoff, water temperature and electrical conductivity data. Multivariate Linear Regression (MLR), Back-propagation Neural Networks (BPNN) and General Regression Neural Networks (GRNN) were applied and their performances compared in this study. The most accurate prediction proved to be GRNN. This paper describes the influence of single input parameters on the prediction.

Keywords

River Danube, General Regression Neural Networks, Back-propagation Neural Networks, Dissolved Oxygen

1. Introduction

Dissolved oxygen is a very significant parameter in the condition of surface waters, and so its prediction by the help of general and easily measureable parameters is an important scientific question. The concentration of dissolved oxygen (DO) reflects the equilibrium or its lack between oxygen-producing processes (e.g. photosynthesis) and oxygen-consuming processes (e.g. aerobic respiration, nitrification, and chemical oxidation) and depends on many factors such as temperature, salinity, oxygen depletion, sources of oxygen and other water quality parameters [1]. The DO level is a measure of the health of aquatic systems. A certain minimum level of DO in water is required for aquatic life to survive [2].

Various models are used for the prediction of several parameters of surface waters, but in the last decade the techniques of artificial intelligence have been successfully applied as a forecasting method. In most research, the simply prediction of the concentration of dissolved oxygen was the aim [1, 3, 4, 5, 6, 7, 8, 9], while in a number of studies the prediction of biological oxygen demand (BOD) was the purpose [2, 7, 10] and, very rarely, models were applied to the estimation of chemical oxygen demand (COD) [7, 11]. MLP was applied by Rankovic et al. [3] for the modelling of DO in a reservoir, in Serbia, and in their next study [8] an adaptive network-based fuzzy inference system (ANFIS) model was used on the same dataset, but with fewer input variables. Ahmed [1] developed two models, an MLP and a radial basis function neural network (RBFN), for the prediction of DO in the Surma River (Bangladesh) using BOD and COD and the models were compared: the RBFN predicted better. Emamgholizadeh et al. [7] used three models (MLP, RBFN and ANFIS) and the MLP was the most efficient in predicting water quality variables (DO, BOD and COD) in the Karoon River, Iran. Basant et al. [2] predicted the DO and BOD in the Gomti River, India, using two models (partial least squares regression and MLP), and the performance of the MLP was better. MLP was developed by Dogan et al. [10] to predict the BOD in the Melen River, Turkey, and the COD was found to be more effective on the BOD estimation. The MLP with the Bayesian regularization training algorithm was successfully utilized by Wen et al. [4] to simulate the DO concentrations in the Heihe River, China, where the most effective inputs were determined as pH, NO3-N and NH4-N. Two applied models (MLR and GRNN) were compared by Heddam [9] and it was found that the best fit was obtained using GRNN model in prediction of DO in the Upper Klamath River, USA. Antanasijevic et al. [5] developed three models: MLP, GRNN and Recurrent Neural Network (RNN) for the modelling of DO in the River Danube, in Serbia at a single location, Bezdan, and the obtained results showed that RNN performed much better than the other methods. Only GRNN was used by Antanasijevis et al. [6] for the prediction of DO in the River Danube, in Serbia, at 17 sample sites, and various normalization and input selection techniques were compared and applied successfully.

The main objective of this study is to predict one of the most important parameters, dissolved oxygen, with the help of some easily measured physical and chemical variables of the River Danube using MLR and two types of neural networks (GRNN and BPNN). A further aim is to evaluate the results obtained and to apply sensitivity analysis to them, in order to determine which input variable(s) played a significant role in the prediction of output.

2. Material and methods

Water quality data set

There are 12 sampling sites in the section of the River Danube in Hungary, the Mohács station (Figure 1.) was chosen as a representative location, while the studied period was from 1998

to 2003. This complete river water quality data set was divided into two subsets. The data from 2003 were used as the test data set (26 data patterns, 17% of all available data), and the data from 1998-2002 were used as the training set (128 data patterns, 83% of all available data). The output variables corresponding to the input variables belonged to the same water sample, which was measured in the same time and at the same location. The same training and testing sets were used with every single model applied.

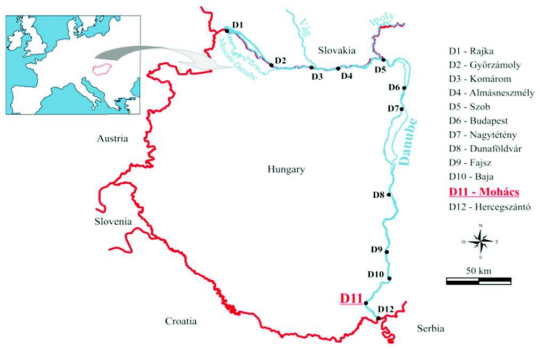

Figure 1. Hungarian section of the River Danube

Multivariate Linear Regression

Multivariate Linear Regression (MLR) is used to estimate the linear association between the dependent and one or more independent variables. MLR is based on least squares; and it expresses the value of the predicted variable as a linear function of one or more predictor variables:

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + ... + \beta_i * x_i$$

where x_i is the value of the i_{th} predictor variable, β_0 is the regression constant, and β_i is the coefficient of the i_{th} predictor variable.

Back-propagation neural network

Artificial neural networks (ANNs) are basically parallel computing systems similar to biological neural networks. Among the various types of ANNs the multilayer perceptron (MLP) neural network structure is the most commonly used and is a well-researched basic ANN architecture. The MLP has generally three layers: input, output and one or more hidden layer(s). Each layer consists of one or more basic element(s) called a neuron or a node (or a processing unit). Nodes are connected to each other by links, synapses are characterised by a weight factor, which denotes the connection strength between two nodes. Each node in the input and inner layers receives input values, processes it, and passes it to the next layer. This process is conducted by weights [10], meaning that the hidden layer sums the weighted inputs and own bias value and uses the own transfer function to create an output value. Typical transfer functions are the linear, the sigmoid or the hyperbolic tangent function [12].

Back-propagation neural networks (BPNN) are multilayer feed-forward perceptrons (MLP) trained from the input data using an error back-propagation algorithm [5]. Back-propagation was proposed by Rumelhart et al. [13], and it is the most popular algorithm for the training of an MLP network [12]. This back-propagation algorithm has two steps. The first step is a forward pass, in which the effect of the input is passed forward through the network to reach the output layer. After the error is computed, a second step starts backward through the network [7] to correct the initial assigned weights of the input layer in such a way as to minimize the error. The term "feed-forward" means that a node connection only exists from a node in the input layer to other nodes in the hidden layer or from a node in the hidden layer to nodes in the output layer; and nodes within a layer are not interconnected to each other, there are not lateral or feedback connections. MLP using a BP algorithm is sensitive to randomly assigned initial connection weights [14]. The initialization of weights and bias values for a layer is conducted using Nguyen-Widrow method in the MATLAB environment [15], and these initial values are dissimilar on every single run, so after the training process different predicted values are obtained. Since these predicted values were significantly different, the MLP was trained sixty times at the same settings (number of neuron, input and target values, transfer functions and back-propagation algorithm etc.) and the average of these predicted values were taken into account.

In this study, the Levenberg-Marquardt algorithm is applied for adjusting the MLP weights [16] and the number of epochs was 1000. One hidden layer and a hyperbolic tangent sigmoid transfer function were used between the input and the hidden layer and a linear transfer function was employed between the hidden and output layers. Neural Network Toolbox of MATLAB was utilized for both ANNs.

General Regression neural network

GRNN was introduced first by Specht [17] as an alternative to MLP. GRNN is a modified form of the radial basic function neural network model. GRNN is a one-pass supervised learning network, and it is a universal approximator for smooth functions. GRNN is a four-layer feed-forward neural network, which is shown in Fig. 2. The first layer is fully connected to the second. Each input unit in the first layer corresponds to an independent variable in the model and the number of pattern neurons is equal to the number of data patterns. The training between the input layer and the pattern layer is performed by defining the weights (the center of the RBF functions) with the help of a special clustering algorithm such as the k-means algorithm [14] and estimates the Euclidean distance of the ith input vector (x_i) and the weight of the i^{th} input variable and the j^{th} pattern node (w_{ij}) where N is the number of input variables.

$$D_j = \sqrt{\sum_{i=1}^{N}(w_{ij} - x_i)^2}\,. \qquad (1)$$

Using the most popular RBF function, the Gaussian Kernel Function as an activation function, where σ is the smoothing factor or spread:

$$f(D_j) = \exp\left(\frac{-D_j}{2\sigma^2}\right). \qquad (2)$$

The smoothing factor is the only "unknown" parameter in the GRNN algorithm; it represents the width of the calculated Gaussian Kernel Function, and must be given before training the model.

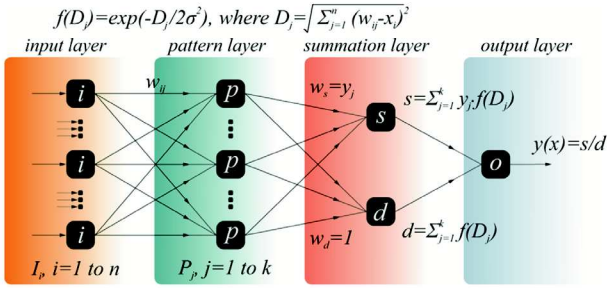

Figure 2. A schematic representation of GRNN, adopted from Antanasijevic et al. [6]

The number of neurons in the summation layer can be expressed as No+1, where No is the number of output neurons [6]. Since the model has only one output, each pattern layer unit is connected to the two neurons in the summation layer: the S-summation neuron and the D-summation neuron. The weights between the summation-neuron and output neuron are equal to the measured value of the output variable. The S-summation neuron computes the sum of the weighted outputs of the pattern layer (S) while the D-summation neuron calculates the unweighted outputs of the pattern neurons (D).

$$D = \sum_{j=1}^{K} f(D_j) \tag{3}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2} \tag{4}$$

Finally, the output layer merely divides the S-summation neuron by the D-summation neuron [9].

Statistical forecasting of the models

The performance of the applied models can be assessed by several statistical error parameters. The root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R2) were used to provide an indication of goodness of fit between the observed and predicted values. Expressions for these error parameters are given as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2} \tag{5}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|O_i - P_i| \tag{6}$$

$$R^2 = \frac{\left[\sum_{i=1}^{n}(O_i - \bar{O})(P_i - \bar{P})\right]^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2 \sum_{i=1}^{n}(P_i - \bar{P})^2} \tag{7}$$

where n is the number of input samples; and O_i and P_i are the observed and predicted output value from the ith element, respectively. $\bar{O}$ and $\bar{P}$ denote their respective averages.

3. Results and discussion

Prediction by Multivariate Linear Regression

Equation (8) was obtained by the MLR from the input water quality variables, which represent the whole dataset (training set + testing set)

$$DO = 5.9198 * pH - 0.2101 * WT - 0.0053 * EC - 0.0002 * RF - 32.6353 \tag{8}$$

where pH is the water's pH value; WT is the water temperature (°C), EC is the electrical conductivity (μS/cm), and RF is the runoff (m^3/s).

Table 1 shows the performance evaluation of the MLR model in reference to training, testing and whole set.

Table 1. Performance parameters of the MLR model

MLR	RMSE (mg/L)	MAE (mg/L)	R^2
training	1.13	0.79	0.52
testing	1.73	1.24	0.45
whole	1.31	0.92	0.46

Prediction by Back-propagation Neural Network

The best performance with reference to the testing set was achieved by BPNN using z-score (normalizing so the inputs and targets have zero mean and unity standard deviation) and 5 neurons in the single hidden layer. These results, which are presented in Table 2, were the errors of sixty run's average.

Table 2. Performance parameters of the BPNN model – the error of sixty run's average

BPNN	RMSE (mg/L)	MAE (mg/L)	R^2
training	0.65	0.43	0.85
testing	1.57	1.28	0.57
whole	0.88	0.57	0.77

Prediction by General Regression Neural Network

The best performance with reference to the testing set was gained by BPNN applying z-score and 0.3 as smoothing factor, Table 3 depicts the obtained values of RMSE, MAE, R2 for the training, testing and the whole sets.

Table 3. Performance parameters of the GRNN model

GRNN	RMSE (mg/L)	MAE (mg/L)	R^2
training	0.47	0.27	0.93
testing	1.42	1.14	0.72
whole	0.72	0.41	0.85

Summarizing, the comparison of the results of the MLR and BPNN models with GRNN revealed that the GRNN performed better than the MLR and BPNN models in both training and testing.

Sensitivity analysis

Sensitivity analysis was applied to determine the relative significance of each input variable, namely which parameter played the most important role in predicting the DO. The optimal network architecture (GRNN) which provided the best performance was selected as a base and the evaluation process was conducted to eliminate only one input parameter in the data set. Table 4 gives the results of five networks, and each one demonstrates the extents, to which the eliminated variable would affect the network accuracy. As the results in the Table 4 show, the pH value was the most effective parameter in predicting DO, and the runoff had the weakest effect on the accuracy the prediction of DO.

Table 4. Sensitivity analysis of input variables eliminated separately

Combination	RMSE (mg/L)		MAE (mg/L)		R^2	
	Training	Testing	Training	Testing	Training	Testing
All	0.47	1.42	0.27	1.14	0.93	0.72
Eliminate pH	1.07	2.14	0.73	1.69	0.59	0.24
Eliminate EC	0.70	1.57	0.42	1.26	0.83	0.62
Eliminate WT	0.67	1.50	0.44	1.30	0.85	0.73
Eliminate RF	0.68	1.46	0.47	1.16	0.84	0.72

Following the application of sensitivity analysis by GRNN, the pH value of the applied input variables has the most significant influence on the prediction of the DO. This result is confirmed

by correlation coefficients. The highest correlation was obtained between the DO and pH (0.33), while the lowest was between the DO and RF (-0.16). The correlation coefficients between DO-WT and DO-EC were -0.29 and 0.25.

4. Conclusion

In this study, two types of the ANNs, namely (BPNN and GRNN) and MLR were applied to predict the DO in the River Danube, at a single location, Mohács, with water pH, temperature, electrical conductivity and runoff. In order to compare the two ANNs and MLR results, RMSE, MAE, and R2 were used as evaluation criteria. Based on the results obtained by training and testing of the ANNs, it was found that the GRNN model provided better predictions of DO than the BPNN, and so the use of GRNN is justified not only due to its better performance, but also on account of it quickness, as, in contrast to BPNN, it is a one-pass training algorithm that does not necessitate an iterative training process. A comparison of the ANNs with the conventional MLR shows that the ANNs demonstrated better performance indicators than the MLR when every model was trained and tested by the same data sets and input variables. Conclusions have shown that the two ANNs, and especially the GRNN are practical methods for predicting DO concentrations in a river.

References

[1.] Ahmed A. A. M.: 2014. Prediction of dissolved oxygen in Surma River by biochemical oxygen demand and chemical oxygen demand using the artificial neural networks (ANNs), Journal of King Saud University – Engineering Sciences, http://dx.doi.org/doi:10.1016/j.jksues.2014.05.001

[2.] Basant N., Gupta S., Malik A., Singh K. P.: 2010. Linear and nonlinear modelling for simultaneous prediction of dissolved oxygen and biochemical oxygen demand of the surface water – A case study, Chemometrics and Intelligent Laboratory System, Vol. 104, pp. 172-180. http://dx.doi.org/doi:10.1016/j.chemolab.2010.08.005

[3.] Rankovic V., Radulovic J., Radojevic I., Ostojic A., Comic L.: 2010. Neural network modelling of dissolved oxygen in the Gruza reservoir, Serbia. Ecological Modelling, Vol. 221, pp. 1239–1244. http://dx.doi.org/doi:10.1016/j.ecolmodel.2009.12.023

[4.] Wen X., Fang, J., Diao M., Zhang C.: 2013. Artificial neural network modelling of dissolved oxygen in the Heihe River, Northwestern China, Environmental Monitoring and Assessment, Vol. 185, pp. 4361-4371. http://dx.doi.org/10.1007/s10661-012-2874-8

[5.] Antanasijevic D., Pocajt V., Povrenovic D., Peric-Grujic A., Rictic M.: 2013. Modelling of dissolved oxygen content using artificial neural networks: Danube River, North Serbia, case study, Environmental Science and Pollution Research, Vol. 20, pp. 9006-9013. http://dx.doi.org/10.1007/s11356-013-1876-6

[6.] Antanasijevic D., Pocajt V., Povrenovic D., Peric-Grujic A., Rictic M.: 2014. Modelling of dissolved oxygen in the Danube River using artificial neural networks and Monte Carlo Simulation uncertainty analysis, Journal of Hydrology Vol. 519, pp. 1895-1907. http://dx.doi.org/doi:10.1016/j.jhydrol.2014.10.009

[7.] Emamgholizadeh S., Kashi H., Marofpoor I., Zalaghi E.: 2014. Prediction of water quality parameters of Karoon River (Iran) by artificial intelligence-based models, Int. Journal of Environmental Science and Technology, Vol. 11, pp. 645-656. http://dx.doi.org/10.1007/s13762-013-0378-x

[8.] Rankovic V., Radulovic J., Radojevic I., Ostojic A., Comic L.: 2012. Prediction of dissolved oxygen in reservoirs using adaptive network-based fuzzy inference system, Journal of Hydroinformatics, Vol. 14, pp. 167-179. http://dx.doi.org/doi:10.2166/hydro.2011.084

[9.] Heddam S.: 2014. Generalized regression neural network-based approach for modelling hourly dissolved oxygen concentration in the Upper Klamath River, Oregon, USA, Environmental Technology, Vol. 35, pp. 1650-1657. http://dx.doi.org/10.1080/09593330.2013.878396

[10.] Dogan E., Sengorur B., Koklu R.: 2009. Modelling biochemical oxygen demand of the Melen River in Turkey using an artificial neural network technique, Journal of Environmental Management, Vol. 90, pp. 1229–1235. http://dx.doi.org/10.1016/j.jenvman.2008.06.004

[11.] Talib A. M. I. Amat M. I.: 2012. Prediction of chemical oxygen demand in Dondang river using artificial neural network, International Journal of Information and Education Technology, Vol. 2, pp. 259-261. http://dx.doi.org/10.7763/IJIET.2012.V2.124 IJIET.2012.V2.124 **[12.] Haykin S.** 1998. Neural Networks: A comprehensive foundation, 2nd Ed., Prentice-Hall, Upper Saddle River, NJ.

[13.] Rumelhart D. E., Hinton G. E., Williams R. J. 1986. Learning internal representation by error back propagation, In: Rumelhart DE, and JL. McClelland (eds) Parallel distributed processing. MIT Press, Cambridge, MA, pp 318-362.

[14.] Kim S., Kim H. S.: 2008. Neural networks and genetic algorithm approach for nonlinear evaporation and evapotranspiration modelling, J. Hydrol., Vol. 351. pp. 299-317. http://dx.doi.org/10.1016/j.jhydrol.2007.12.014

[15.] Pavelka A. Procházka A.: 2004. Algorithms for initialization of neural network weights, Sbornik prispevku 11. Konference MATLAB, Vol. 2. pp. 453-459.

[16.] Marquardt D.: 1963. An algorithm for least square estimation of nonlinear parameters, Journal of the Society for Industrial and Applied Mathematics, Vol. 11, pp. 431-441. http://dx.doi.org/10.1137/0111030

[17.] Specht, D. F.: 1991. A general regression neural network, IEEE Transactions on Neural Networks, Vol. 2, pp. 568-576. http://dx.doi.org/10.1109/72.97934