Neural Comput & Applic (2014) 24:1115–1121
DOI 10.1007/s00521-012-1334-2

ORIGINAL ARTICLE

Predicting piezometric water level in dams via artificial neural networks

Vesna Ranković · Aleksandar Novaković · Nenad Grujović · Dejan Divac · Nikola Milivojević

Received: 31 May 2012 / Accepted: 29 December 2012 / Published online: 12 January 2013

Abstract The safety control of dams is based on measurements of parameters of interest such as seepage flows, seepage water clarity, piezometric levels, water levels, pressures, deformations or movements, temperature variations, loading conditions, etc. Interpretation of these large sets of available data is very important for dam health monitoring and it is based on mathematical models. Modelling seepage through geological formations located near the dam site or dam bodies is a challenging task in dam engineering. The objective of this study is to develop a feedforward neural network (FNN) model to predict the piezometric water level in dams. An improved resilient propagation algorithm has been used to train the FNN. The measured data have been compared with the results of FNN models and multiple linear regression (MLR) models that have been widely used in analysis of the structural dam behaviour. The FNN and MLR models have been developed and tested using experimental data collected during 9 years. The results of this study show that FNN models can be a powerful and important tool which can be used to assess dams.

Keywords Dam · Piezometric water level · Neural network · IRPROP+

V. Ranković (✉) · A. Novaković · N. Grujović
Department for Applied Mechanics and Automatic Control,
Faculty of Engineering, University of Kragujevac,
Sestre Janjić 6, 34000 Kragujevac, Serbia
e-mail: vesnar@kg.ac.rs

D. Divac · N. Milivojević
Institute for Development of Water Resources "Jaroslav Černi",
80 Jaroslava Černog St., 11226 Beli Potok, Belgrade, Serbia

1 Introduction

All dams suffer to a certain extent from seepage, as water blocked by a dam will find ways of escaping through the dam and its surroundings. Unless prevented, such seepage can undermine the dam structure and cause the dam to fail as a result of piping and sloughing [1, 2]. In order to monitor the seepage, piezometric devices are installed at certain sections of the dam. The problem of the seepage path has been resolved by developing physical and mathematical models. Examples of physical models include the one developed by Turkmen et al. [3] that consisted of drilling boreholes and using dye trace tests in order to identify the seepage paths, electrical methods used by Panthulu et al. [4] for detecting and mapping the seepage paths, as well as the study of joint systems of the rock units performed by Uromeihy and Barzegari [5] in order to evaluate the potential of water seepage.

Mathematical methods for seepage analysis can be deterministic or statistical. Deterministic methods include finite difference [6, 7], finite volume [8], finite element [9] and boundary element method [10]. The main idea of these methods is to transform the governing partial differential equations into algebraic equations and solve them over the flow domains. Deterministic methods require knowledge of spatial distribution of material properties for each node of the flow domains. However, in some cases, it may not be possible to obtain such information since most of the field measurements include only point scaled data. Although there are several methods to generate the spatial distribution of material properties from the point scaled data, some unrealistic results may be obtained due to the parameter uncertainty.

The use of the statistical models is preferred nowadays. Statistical methods are based on the results gained from experimental data. One of the most commonly used methods in statistics is regression analysis which investigates the relationship between the variables [11, 12]. The principal component analysis has been applied in the study of the groundwater evolution in different conditions [13]. Methods of artificial intelligence (artificial neural networks, fuzzy systems and neuro-fuzzy systems) have opened up new possibilities for diverse applications in civil engineering, such as sediment transport [14–16] and groundwater engineering [17–20]. During the past few years, soft computing methodologies have experienced important developments in the domain of dam engineering problems. Several studies have been conducted on the application of these techniques to forecasting of dam displacement [21, 22], dam shape optimization [23], crack detection of arch dam [24] and evaluation and prediction blast-induced ground vibration [25].

In order to predict water level in piezometers, Tayfur et al. [26] developed an artificial neural network (ANN) model and used water levels on the upstream and the downstream sides of the dam as the input variables. Appropriate choice of input variables for artificial neural networks is important for accurate modelling of water levels in piezometers. Bonelli and Royet [27] demonstrated that piezometric measurements are generally affected by the water level effect and the rainfall effect. If the curtain or the core of a reservoir is watertight and the effect of rainfall on piezometers is minimal, the piezometers will react to changes in the headwater or tailwater levels depending on their location with respect to the curtain. The tailwater piezometers will not react to the changes of headwater and vice versa. However, if the curtain is not watertight at some location, the piezometer on the seepage path between the headwater and the tailwater will react to the changes of both sensors [28].

The objective of this study is to develop an ANN model to predict water levels in piezometers. In order to achieve the most accurate model, input variables have been carefully selected. An improved resilient propagation algorithm has been used to minimize the error between the neural network predictions and the desired outputs.

2 Case study: The Iron Gate 2 Dam

The Iron Gate 2 Dam (Fig. 1) was built in 1984 as a joint venture between the governments of Serbia and Romania to produce electricity for both countries. It is located on the Danube, 80 km downstream of the Iron Gate 1 Dam, and its operation is coupled with the operation of the Iron Gate 1 system. The overall Iron Gate 2 Dam system consists of two hydro power plants, two navigation locks, two overflow dams, two nonoverflow embankment dams and a road across the dam. The Iron Gate 2 reservoir is an 80 km long section of the Danube channel. At maximum water levels, the volume of the reservoir is 820 million m^3. No significant tributaries flow into it.

Fig. 1 The Iron Gate 2 Dam

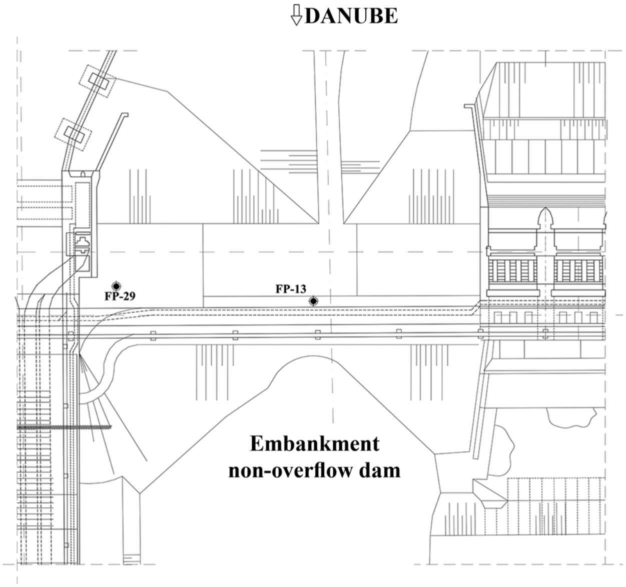

Fig. 2 The cross section of the nonoverflow embankment dam

Piezometers FP13 and FP29, installed at a section of the nonoverflow embankment dam, were used for this study (Fig. 2). Water levels in the piezometers have been measured twice a month since 1997. The data collected from January 1997 to November 2005 were used for training and testing the employed neural networks.

3 Overview of artificial neural networks and MLR

3.1 ANN model

ANNs are able to identify relationships from given patterns and therefore can solve large-scale complex problems, such as pattern recognition, nonlinear modelling, classification, etc. ANNs tend to be classified into two main categories: *recurrent* networks, in which loops occur due to feedback connections, and *feed-forward* networks (FNN), in which the network structure has no loops. Choices of network architectures are very closely related to the learning algorithm used in the training of the network. The three-layer FNN is typically used in many civil engineering applications.

3.1.1 FNN model and training algorithm

FNNs are composed of simple components, the so-called neurons. A neuron model is shown in Fig. 3. Neurons are structured in layers, and neurons in adjacent layers are connected by uni-directional links (*synapses*) which pass information from one neuron to another. Inputs are represented by neurons in the first layer (*input layer*); outputs are represented by neurons in the last layer (*output layer*). All layers between the input and output layers are known as *hidden layers.*

Activation functions are attached to layers. The role of an activation function is to scale data output from layers. The most commonly used activation functions in the construction of FNNs are linear and sigmoid activation functions. Linear activation functions are represented in the form of:

$$f(y) = y \tag{1}$$

The two most commonly used sigmoid activation functions are the logistic function and the hyperbolic tangent function, which can be defined as follows:

1. Logistic function

$$f(y) = \frac{1}{1 + e^{-y}} \tag{2}$$

2. Hyperbolic tangent function [29]

$$f(y) = \frac{1 - e^{-y}}{1 + e^{-y}}. \tag{3}$$

The output of a neuron in the hidden layer can be expressed as:

$$\text{output} = f(y) \tag{4}$$

where $y = \sum_{i=1}^{k} w_i x_i + b$; $x_1, x_2, \ldots, x_\text{k}$ are the input signals; $w_1, w_2, \ldots, w_\text{k}$ are the weights of neuron; b is the bias value; and $f(\cdot)$ is the activation function.

The three-layer FNN with s neurons in the input layer is shown in Fig. 4. The inputs $x = (x_1, x_2, \ldots, x_\text{s})$ are multiplied by the weights $w_{ij}^{(1)}$ and added up at each hidden node. The summed signal at the node then activates a nonlinear function (sigmoid function). The output y at the linear output node can thus be calculated from its inputs as follows:

$$y = \sum_{i=1}^{k} w_{il}^{(2)} \frac{1}{1 + e^{-\left(\sum_{j=1}^{s} x_j w_{ij}^{(1)} + b_i^{(1)}\right)}} + b_l^{(2)} \tag{5}$$

where s is the number of inputs, k is the number of hidden neurons, x_j is the jth element of input, $w_{ij}^{(1)}$ is the first layer weight between the ith hidden neuron and jth input, $w_{ij}^{(2)}$ is the second layer weight between the ith hidden neuron and output neuron, $b_i^{(1)}$ is a biased weight for the ith hidden neuron and $b_i^{(2)}$ is a biased weight for the output neuron.

The most commonly used training algorithm for updating the weights and biases of a neural network is the standard backpropagation learning algorithm introduced by Rumelhart et al. [30]. However, this scheme suffers from local minima and slow convergence. One of the many

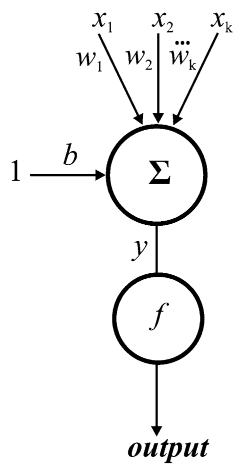

Fig. 3 A nonlinear neuron model

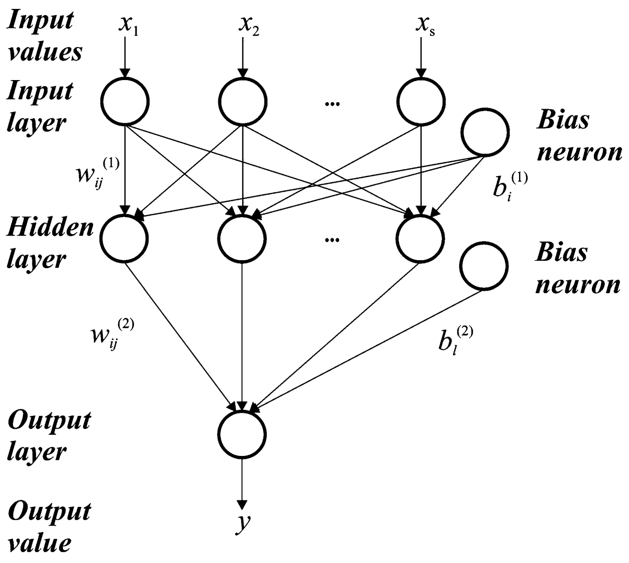

Fig. 4 The model of a feedforward neural network with one hidden layer

variant forms of the backpropagation algorithm is a resilient propagation algorithm (RPROP) [31, 32]. In contrast to the standard backpropagation algorithms, RPROP does not take into account the weakening influence of the absolute value of the partial derivative for calculation of the weight changes but only the sign of the derivative to indicate the direction of the weight update:

$$\Delta w_{ji}(s) = \begin{cases} -\Delta_{ji}(s), & \text{if } \frac{\partial E}{\partial w_{ji}}(s) > 0 \\ +\Delta_{ji}(s), & \text{if } \frac{\partial E}{\partial w_{ji}}(s) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$\Delta_{ji}(s) = \begin{cases} \alpha^{+}\Delta_{ji}(s-1), & \text{if } \frac{\partial E}{\partial w_{ji}}(s-1) * \frac{\partial E}{\partial w_{ji}}(s) > 0 \\ \alpha^{-}\Delta_{ji}(s-1), & \text{if } \frac{\partial E}{\partial w_{ji}}(s-1) * \frac{\partial E}{\partial w_{ji}}(s) < 0 \\ \Delta_{ji}(s-1), & \text{otherwise} \end{cases} \tag{7}$$

in which $0 < \mathrm{a}^{-} < \mathrm{a}^{+} < 1$, s is the step number, Δ_{ij} is the step size or update value of the weight w_{ij} and E is the local error function.

The modification of the RPROP algorithm proposed by Igel and Husken [33] and named IRPROP+ has been used in this study. As demonstrated on several benchmark problems, the IRPROP+ performance is superior compared to the original RPROP algorithm. The basic improvement is that it takes into account the change of sign in the partial derivative, which implies that the algorithm can overcome a local minimum. However, it does not indicate whether the weight update has caused an increase or a decrease. The modification of RPROP is to make a reversal step dependent on the history of the error. Those weight updates that have caused changes to the signs of the corresponding partial derivatives are reverted in case of an error increase.

3.2 Multiple linear regression

A multiple linear regression model (MLR) is one of the statistical techniques most widely used for modelling the linear relationship between a dependent variable and one or more independent variables. The briefly described standard MLR methodology has been used to validate the effectiveness of the proposed FNN model. Consider a training data set $\{(u_1, z_1), (u_2, z_2), \ldots, (u_p, z_p)\} \in \mathbb{R}^N \times \mathbb{R}$ where $u_i = \{u_{1i}\, u_{2i} \ldots u_{Ni}\}^T$ is a vector of input variables and z_i is the corresponding output value, p is the number of training data points. The multiple linear regression model is given by:

$$z_m = \beta_0 + \beta_1 u_1 + \beta_2 u_2 + \cdots + \beta_N u_N \tag{8}$$

where β_i represents unknown parameters, β_i can be estimated by which the sum of the squares of the errors:

$$\varepsilon = (z_1 - z_{m1})^2 + (z_2 - z_{m2})^2 + \cdots + (z_p - z_{mp})^2 \tag{9}$$

in which z_{mi} denotes the MLR output value from the i-th input element:

$$z_{mi} = \beta_0 + \beta_1 u_{1i} + \beta_2 u_{2i} + \cdots + \beta_N u_{Ni} \tag{10}$$

The matrix form of Eq. (9) is:

$$\varepsilon = (z - U\beta)^T (z - U\beta) \tag{11}$$

where: $U = \begin{bmatrix} 1 & u_{11} & u_{21} & \cdots & u_{N1} \\ 1 & u_{12} & u_{22} & \cdots & u_{N2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & u_{1p} & u_{2p} & \cdots & u_{Np} \end{bmatrix}$, $\beta = \{\beta_0 \beta_1 \ldots \beta_N\}^T$, $z = \{z_1\, z_2 \ldots z_p\}^T$, and the least squares estimator of $\boldsymbol{\beta}$ is given by:

$$\beta = (U^T U)^{-1} U^T z. \tag{12}$$

4 Simulation results

For the purpose of constructing and training the neural network, a program was written in Java by the authors. The program implemented classes provided by the Encog framework which offers a comprehensive range of network architectures, training algorithms and neuron activation functions. In order to predict water level in piezometers FP13 and FP29, a three-layered FNN was used (one FNN model per piezometer). Accuracy of the model depends on the appropriate choice of the input variables. The input variables of the FNN model were measurements of the tailwater levels taken on the same day (ht_1), 1 day before (ht_2) and 2 days before (ht_3) the measurements taken by piezometers. Such input variables were chosen because the core and the curtain of the reservoir were watertight and the position of piezometers with the respect to the curtain was such that they could only react to changes in the tailwater level [28]. The optimal number of neurons in the hidden layer was found using the Prune Incremental class provided by the Encog framework. This class uses the brute force technique to determine an optimal hidden layer configuration by trying out a number of different hidden layer configurations in order to find the optimal one. The water level in a piezometer is the output variable (hp).

The input and output data were normalized within a range from 0.1 to 0.9 by

$$f(zn) = \frac{(zn - D_l)(N_h - N_l)}{(D_h - D_l)} + N_l \tag{13}$$

where zn is the value to be normalized, D represents the upper and lower values of the data and the variable N represents the higher and lower bounds of the normalization range. The accuracy of approximated results was evaluated using the correlation coefficient (r), the coefficient of determination (R^2), the mean square error (MSE) and the mean absolute error (MAE):

$$\mathrm{r} = \frac{\sum_{i=1}^{N_o}(y_i - \bar{y})(m_i - \bar{m})}{\sqrt{\sum_{i=1}^{N_o}(y_i - \bar{y})^2 \sum_{i=1}^{N_o}(m_i - \bar{m})^2}} \tag{14}$$

$$R^2 = r^2 \tag{15}$$

$$\mathrm{MAE} = \frac{1}{N_o}\sum_{i=1}^{N_o}|y_i - m_i| \tag{16}$$

$$\mathrm{MSE} = \frac{1}{N_o}\sum_{i=1}^{N_o}(y_i - m_i)^2 \tag{17}$$

where y_i and m_i represent the network output and the measured value from the ith element; $\bar{y}$ and $\bar{m}$ represent their average values, respectively, and N_o denotes the number of observations. The data set included 183 data samples divided into training and test sets. The basic statistics of the measured variables are reported in (Table 1).

The data collected during the period from 29 January 1997 to 24 December 2004 were used for training the network. During the training procedure, the predicted model results and the measured water level values of the examined piezometer were compared. Subsequently, the trained model was tested by predicting the measured water level data for the period from 10 January 2005 to 21 November 2005. Prune Incremental class was used to determine the optimal number of neurons in the hidden layer and for both models the optimal number was six. In order to scale data output from layers, a log-sigmoid transfer function at the hidden layer and a linear transfer function at the output layer were used.

The architecture of optimum neural network models consisted of three neurons in the input layer, six hidden neurons and one output neuron. The number of the input and output nodes is determined by the number of variables. A detailed architecture of the optimum three-layer neural network model to predict water levels in piezometers is shown in Fig. 5.

The MLR models for prediction of water levels in the piezometers FP13 and FP29 are

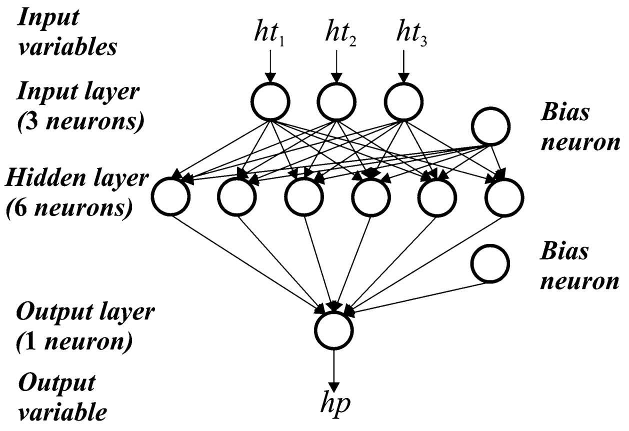

Fig. 5 A detailed architecture of the optimum neural network model to predict water levels in piezometers

$$hp_{13}^{\mathrm{MLR}}(ht_1, ht_2, ht_3) = 4.9819 + 0.5693 \cdot ht_1 + 0.3138 \cdot ht_2 - 0.0308 \cdot ht_3 \tag{18}$$

$$hp_{29}^{\mathrm{MLR}}(ht_1, ht_2, ht_3) = 1.4569 + 0.2283 \cdot ht_1 - 0.1762 \cdot ht_2 + 0.9140 \cdot ht_3 \tag{19}$$

The performance of FNN and MLR models was evaluated by comparing the estimates of the models with experimental data. The performance parameters of the training and test sets are shown in Table 2. As it can be easily observed, both models are capable of predicting water levels in piezometers with reasonable accuracy (i.e. R^2 higher than 0.83). The MLR model has lower R^2 and higher MAE and MSE values than the FNN model, which shows that the FNN scheme presents a better prediction performance than the MLR model. Figures 6 and 7 show measured and FNN model computed values of water levels of the piezometers FP13 and FP29 in the training and test sets.

The results of this study can be compared with the results referenced in the literature. For instance, Tayfur et al. [26] compared the finite element method and the artificial neural network model of flow through embankment dams. They calibrated and verified both models using

Table 1 Basic statistics of the measured variables

Variable	Min	Max	Ave	SD	chp_{13}	chp_{19}
ht_1	28.6	37.16	32.1628	1.669	0.971	0.9715
ht_2	28.71	36.93	32.1157	1.6603	0.9611	0.9723
ht_3	28.87	37	32.1312	1.663	0.971	0.9634
hp_{13}—water levels in the piezometer FP13	29.96	37.26	32.392	1.5077	1	–
hp_{19}—water levels in the piezometer FP29	29.25	37.1	32.535	1.6557	–	1

Table 2 The performance parameters of the FNN and MLR models for prediction of water levels in the piezometers FP13 and FP29

Piezometer	*r*		R^2		MAE		MSE	
	Training	Test	Training	Test	Training	Test	Training	Test
P13 FNN model	0.98	0.99	0.95	0.98	0.24	0.6	0.1	0.45
P13 MLR model	0.95	0.92	0.9	0.85	0.53	1.06	0.21	0.76
P29 FNN model	0.98	0.98	0.96	0.97	0.23	0.73	0.1	0.68
P29 MLR model	0.94	0.91	0.8836	0.83	0.61	1.12	0.37	0.93

Min minimum value, *Max* maximum value, *Ave* average value, *SD* standard deviation, chp_{13} correlation with hp_{13}, chp_{29} correlation with hp_{29}

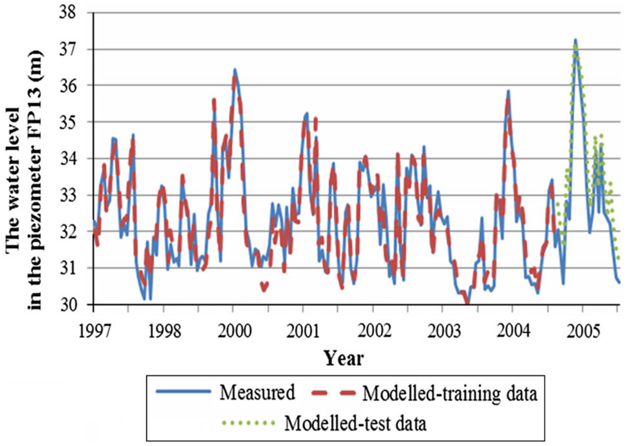

Fig. 6 Measured and FNN model computed values of the water level of the piezometer FP13 in the training and test sets

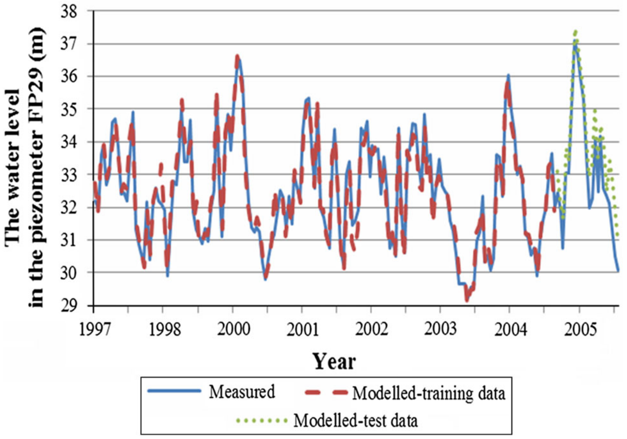

Fig. 7 Measured and FNN model computed values of the water level of the piezometer FP29 in the training and test sets

the measured data from the piezometers installed for seepage monitoring. The coefficients of determination for values modelled by the ANN were 0.96, 0.93 for the training and test data sets, respectively. Similarly, the proposed FNN model can efficiently predict water levels in piezometers.

5 Conclusions

One of the most successful artificial neural network applications is to model complex nonlinear systems. The multilayer feedforward neural networks are capable of learning and generalizing directly from training examples. This makes artificial neural networks a very powerful tool to solve many of the complicated civil engineering problems.

The seepage process through the dam system adversely influences the dam stability. Therefore, seepage modelling is very important for dam health monitoring. For this reason, the present study has been focused on the efficient prediction of water level in piezometers via advanced ANN metamodels. The improved resilient propagation algorithm was used to train the FNN. The FNN results were compared with experimental data and multiple linear regression models. The performance of the models was tested using correlation coefficients, the mean absolute error and the mean square error. The nonlinear FNN model approach has been shown to provide a better prediction of water levels in piezometers than MLR. Once developed, the neural network model can be used for further monitoring activities, as a predictive management tool.

Acknowledgments The authors would like to thank the reviewers of this paper for their interesting comments and hints, which have helped to improve the quality of the paper. The part of this research is supported by the Ministry of Science in Serbia, Grants III41007 and TR37013.

References

1. Malkawi AIH, Al-Sheriadeh M (2000) Evaluation and rehabilitation of dam seepage problems. A case study: Kafrein dam. Eng Geol 56:335–345
2. Turkmen S, Öeguler E, Taga H, Karaogullarindan T (2002) Detection and evaluation of horizontal fractures in earth dams using the self-potential method. Eng Geol 63:247–257
3. Turkmen S, Ozguler E, Taga H, Karaogullarindan T (2002) Seepage problems in the Karstic limestone foundation of the Kalecik Dam (south Turkey). Eng Geol 63(3–4):247–257

4. Panthulu TV, Krishnaiah C, Shirke JM (2001) Detection of seepage paths in earth dams using self-potential and electrical resistivity methods. Eng Geol 59(3–4):281–295
5. Uromeihy A, Barzegari G (2007) Evaluation and treatment of seepage problems at Chapar-Abad Dam. Iran. Eng Geol 91(2–4): 219–228
6. Ayvaz MT, Karahan H (2007) Modeling three-dimensional free-surface flows using multiple spreadsheets. Comput Geotech 34(2):112–123
7. Bardet JP, Tobita T (2002) A practical method for solving free-surface seepage problems. Comput Geotech 29(6):451–475
8. Darbandi M, Torabi SO, Saadat M, Daghighi Y, Jarrahbashi D (2007) A moving-mesh finite-volume method to solve free-surface seepage problem in arbitrary geometries. Int J Numer Anal Met 31:1609–1629
9. Ouria A, Toufigh MM, Nakhai A (2007) An Investigation on the effect of the coupled and uncoupled formulation on transient seepage by the finite element method. Am J Appl Sci 4(12): 950–956
10. Chen JT, Hong HK, Chyuan SW (1994) Boundary element analysis and design in seepage problems using dual integral formulation. Finite Elem Anal Des 17(1):l–l20
11. Peng T-R, Wang C-H (2008) Identification of sources and causes of leakage on a zoned earth dam in northern Taiwan: hydrological and isotopic evidence. Appl Geochem 23(8):2438–2451
12. Xijian W, Haibin T, Riyun L (2011) Locating the plane concentrated seepage in dam by transient temperature field. Proc Eng 26:1749–1755
13. Mohammadi Z (2009) Assessing hydrochemical evolution of groundwater in limestone terrain via principal component analysis. Environ Earth Sci 59:429–439
14. Kabiri-Samani AR, Aghaee-Tarazjani J, Borghei SM, Jeng DS (2011) Application of neural networks and fuzzy logic models to long-shore sediment transport. Appl Soft Comput 11(2):2880–2887
15. Tayfur G (2002) Artificial neural networks for sheet sediment transport. Hydrolog Sci J 47(6):879–892
16. Yang CT, Marsooli R, Aalami MT (2009) Evaluation of total load sediment transport formulas using ANN. Int J Sediment Res 24(3):274–286
17. Adamowski J, Chan HF (2011) A wavelet neural network conjunction model for groundwater level forecasting. J Hydrol 407(1–4):28–40
18. Corsini A, Cervi F, Ronchetti F (2009) Weight of evidence and artificial neural networks for potential groundwater spring mapping: an application to the Mt. Modino area (Northern Apennines, Italy). Geomorphology 111(1–2):79–87
19. Hu ZY, Huang GH, Chan CW (2003) A fuzzy process controller for in situ groundwater bioremediation. Eng Appl Artif Intel 16(2):131–147
20. Taormina R, K-w Chau, Sethi R (2012) Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. Eng Appl Artif Intel. doi:10.1016/j.engappai.2012.02.009
21. Mata J (2011) Interpretation of concrete dam behaviour with artificial neural network and multiple linear regression models. Eng Struct 33(3):903–910
22. Ranković V, Grujović N, Divac D, Milivojević N, Novaković A (2012) Modelling of dam behaviour based on neuro-fuzzy identification. Eng Struct 35:107–113
23. Gholizadeh S, Seyedpoor SM (2011) Shape optimization of arch dams by metaheuristics and neural networks for frequency constraints. Sci Iran 18(5):1020–1027
24. Wang BS, He ZC (2007) Crack detection of arch dam using statistical neural network based on the reductions of natural frequencies. J Sound Vib 302(4–5):1037–1047
25. Monjezi M, Hasanipanah M, Khandelwal M (2012) Evaluation and prediction of blast-induced ground vibration at Shur River Dam, Iran, by artificial neural network. Neural Comput Appl. doi: 10.1007/s00521-012-0856-y
26. Tayfur G, Swiatek D, Wita A, Singh VP (2005) Case study: finite element method and artificial neural network models for flow through Jeziorsko Earthfill Dam in Poland. J Hydraul Eng 131(6):431–440
27. Bonelli S, Royet P (2001) Delayed response analysis of dam monitoring data. In: Proceedings of the ICOLD European symposium on Dams in a European Context, GEIRANGER: Norvège, pp 91–99
28. Opyrchal L (2003) Application of fuzzy sets method to identify seepage path through Dams. J Hydraul Eng-ASCE 129(7): 546–548
29. Haykin S (1999) Neural networks: a comprehensive foundation. Prentice Hall, Upper Saddle River
30. Rumelhart DE, Hinton GE, Wiliams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536
31. Riedmiller M (1994) Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms. Comput Stand Inter 16(5):265–278
32. Riedmiller M, Braun H (1993) A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Ruspini EH (ed) Proceedings of the IEEE international conference on neural networks, IEEE Press, New York, pp 586–591
33. Igel C, Husken M (2002) Empirical evaluation of the improved Rprop learning algorithms. Neurocomputing 50:105–123