

Primljen / Received: 15.2.2016.

Ispravljen / Corrected: 12.6.2017.

Prihvaćen / Accepted: 28.6.2017.

Dostupno online / Available online: 10.2.2018.

Application of artificial neural networks for hydrological modelling in karst

Authors:



Miljan Kovačević, MCE
University of Priština, Kosovska Mitrovica
Faculty of Technical Sciences
miljan.kovacevic@pr.ac.rs



Assoc.Prof. **Nenad Ivanišević**, PhD. CE
University of Belgrade
Faculty of Civil Engineering
nesa@grf.bg.ac.rs



Assoc.Prof. **Tina Dašić**, PhD. CE
University of Belgrade
Faculty for Civil Engineering
mtina@grf.bg.ac.rs



Assist.Prof. **Ljubo Marković**, PhD. CE
University of Priština, Kosovska Mitrovica
Faculty of Technical Sciences
ljubo.markovic@pr.ac.rs

Preliminary report

Miljan Kovačević, Nenad Ivanišević, Tina Dašić, Ljubo Marković

Application of artificial neural networks for hydrological modelling in karst

The possibility of short-term water flow forecasting in a karst region is presented in this paper. Four state-of-the-art machine learning algorithms are used for the one day ahead forecasting: multi-layer perceptron neural network, radial basis function neural network, support vector machines for regression (SVR), and adaptive neuro fuzzy inference system (ANFIS). The results show that the ANFIS model outperforms other algorithms when the root mean square error and mean absolute error are used as quality indicators.

Key words:

artificial neural network, SVR, ANFIS, rainfall-runoff ratio in karst areas

Prethodno priopćenje

Miljan Kovačević, Nenad Ivanišević, Tina Dašić, Ljubo Marković

Primjena neuronskih mreža za hidrološko modeliranje u krškom području

U ovom radu je razmatrana mogućnost kratkoročnog predviđanja protoka vode u krškom području. Četiri suvremena algoritma strojnog učenja su korištena za predviđanje protoka jedan dan unaprijed i to: neuronska mreža s višeslojnim perceptronom, neuronska mreža s radijalnom aktivacijskom funkcijom, metoda potpornih vektora za regresiju (SVR), adaptivni neuroneizraziti (fuzzy) sustav zaključivanja (ANFIS). Rezultati pokazuju da ANFIS model nadmašuje ostale algoritme kada se korijeni srednje kvadratne popogreške i srednja apsolutna pogreška upotrijebe kao mjere kvalitete.

Ključne riječi:

umjetna neuronska mreža, SVR, ANFIS, odnos oborine-otjecanje u krškom području

Vorherige Mitteilung

Miljan Kovačević, Nenad Ivanišević, Tina Dašić, Ljubo Marković

Anwendung von neuronalen netzen für die hydrologische modellierung im karstgebiet

In dieser Abhandlung wird die Möglichkeit einer kurzfristigen Vorhersage des Wasserflusses im Karstgebiet betrachtet. Vier moderne maschinelle Lernalgorithmen wurden angewendet, um den Durchfluss einen Tag im Voraus vorherzusagen, und dies: neurales Netz mit mehrschichtigem Perzeptron, neurales Netz mit radialer Aktivierungsfunktion, unterstützende Vektoren. Regressionsmethode (SVR), adaptives Neuro-Fuzzy-Inferenzsystem (ANFIS). Die Ergebnisse zeigen, dass das ANFIS-Modell die übrigen Algorithmen übertrifft, wenn die Wurzel des mittleren quadratischen Fehlers und des mittleren absoluten Fehlers als Qualitätsmaß herangezogen wird.

Schlüsselwörter:

künstliches neurales Netz, SVR, ANFIS, Verhältnis Niederschlag-Abfluss im Karstgebiet

1. Introduction

Water resources systems are complex systems featuring numerous interactions and complex structures. They are characterized by a number of uncertainties regarding the system input, required system output, state of the system, goals, models, etc. In such conditions, the system management is very complex and difficult. One of the key parameters in all water resources planning and management tasks is the availability of water resources, usually defined as the flow rate at the gauging station. Various hydrological models can be applied to reduce uncertainties of this parameter, including stochastic models (e.g. regression, neural networks) or process-based (deterministic) models. They usually rely on meteorological observations and catchment characteristics.

Hydrological forecasting is commonly used for the prediction of river flow. It can be divided into short-term, mid-term and long-term forecasting [1]. Based on principles adopted by the World Meteorological Organization (WMO), short-term hydrological forecasts cover a period of up to two days, mid-term forecasts apply to a period ranging from 2 to 10 days, and long-term hydrological forecasts refer to a period exceeding 10 days.

The case study described in the paper refers to a karst area in Bosnia and Herzegovina, with numerous ponors, springs, estavelles, karst poljes, and underground karst channels. Water flow in karstic zones is considered to be a complex hydrogeological phenomenon [2, 3]. Surface water flow is rarely encountered, and groundwater flow is directed depending on the existence of underground karst conduits. The function of these karstic conduits is complex and subject to extreme seasonal variations, including flow reversals. Modelling water flow in such hydrogeological conditions is highly complex and is hindered by numerous uncertainties. Therefore, a "black-box" approach was applied to predict discharge. More precisely, four machine learning algorithms were applied and their results were compared.

The application of machine learning to hydrological modelling problems constitutes a constantly growing field of research. In general terms, machine learning methods are suitable for modelling and forecasting the changes of functional characteristics of various systems, as they can be trained to find solutions, recognize model behaviour, classify data, and predict future events [4]. In hydrology, machine learning is most often applied to forecast water flow and water levels in rivers [5], although it can also be used in problems related to the transport of sediments [6], water quality modelling [7], identification of water areas prone to flooding [8], and in many other applications related to water resources.

Artificial Neural Networks (ANN) such as the Multi-Layer Perceptron (MLP), Generalized Regression Neural Networks (GRNN), and Radial Basis Function (RBF), rank among the machine learning techniques that are most widely applied in the sphere of hydrological modelling [9].

An Adaptive Neuro-Fuzzy Inference System (ANFIS) integrates both neural networks and fuzzy logic principles with the inference system that corresponds to a set of fuzzy if-then rules that can approximate nonlinear functions. Hence, it can potentially benefit from both the ANN and fuzzy logic within a single framework. Sanikhani et al. (2015) modelled two different ANFIS models – the ANFIS with grid partition (ANFIS-GP) and the ANFIS with subtractive clustering (ANFIS-SC) with gene expression programming (GEP) – in order to forecast one, two, and three months in advance the lake level fluctuations at Manyas and Tuz lakes in Turkey [10]. Rezaeianzadeh et al. (2014) applied the ANN, ANFIS and regression models for forecasting the maximum daily flow at the outlet of the Khosrow Shirin watershed, located at the Fars Province in Iran [11]. Tien Bui et al. (2016) proposed a neural fuzzy inference system and metaheuristic optimization for the flood susceptibility modelling. A high-frequency tropical cyclone area of the Tuong Duong district in Central Vietnam was used as a case study [12]. Chang et al. (2014) modelled water quality using two static artificial neural networks (MLP and ANFIS) and one dynamic artificial neural network (NARX) for the Dahan River basin in Taiwan [13].

Support Vector Machines (SVM) belong to a group of supervised learning models that are used for the classification and regression analyses. In addition to performing linear classification or regression, SVMs can efficiently perform a non-linear classification/regression using the so called kernel trick, by learning mapping into the high dimensional kernel induced feature space. Grenata et al. (2015) applied SVMs for rainfall-runoff modelling using the SVM regression variant called Support Vector Regression (SVR) for two experimental basins located in northern Italy [14]. Hosseini and Mahjouri (2016) presented a new rainfall-runoff model called SVR-GANN, where the SVR model is combined with a geomorphology-based ANN model in a case study of three sub-basins located in a semi-arid region in Iran [15]. Gizaw and Gun (2016) developed the Regional Flood Frequency Analysis (RFFA) model based on SVR to estimate regional flood quantiles for two study areas in Canada [16]. He et al (2014) compared ANFIS and SVM for forecasting river flow in a semiarid mountain region in north-western China [17]. Gong et al. (2015) used the ANN, SVM and ANFIS for forecasting groundwater levels near Lake Okeechobee, Florida [18].

A comparative analysis of several state-of-the-art machine learning algorithms for rainfall and runoff forecasting for the Buna Spring is presented in the paper. The MLP and RBF neural networks, and SVM and ANFIS algorithms for short-term prediction of river flow based on the precipitation and flow in the previous two days, are analysed.

2. Machine learning techniques for hydrological modelling

2.1. Multilayer perceptron neural network

The multilayer perceptron is a feed-forward neural network consisting of at least three layers: input layer L1, hidden layer L2, and output layer L3. Each layer is composed of one or more processing units called neurons, where each neuron in one layer is connected to each neuron of the next layer, as shown in Figure 1a. Multiple neuron layers with nonlinear transfer functions allow the network to learn nonlinear relationships between input and output vectors.

Most often the sigmoid activation function is used in the hidden layer. It is defined by:

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

while the linear activation function is used in the output layer. The ANN with one hidden layer whose neurons have the sigmoid activation function, and whose neurons in the output layer have the linear activation function, can approximate multidimensional function for the given set of data when there is a sufficient number of neurons in the hidden layer [19].

The number of neurons in the hidden layer is determined experimentally. It is possible to determine the upper limit, i.e. the maximum number of neurons in the hidden layer that can be used for modelling the system represented by a specific set of data. It is suggested to accept the lower value of the number of neurons in the hidden layer N_H given by (2) and (3), where N_I represents the number of inputs in the neural network, and N_S represents the number of samples for training [20, 21].

$$N_H \leq 2N_I + 1 \quad (2)$$

$$N_H \leq \frac{N_S}{N_I + 1} \quad (3)$$

The use of the MLP neural network with one hidden layer trained with Levenberg-Marquardt algorithm is proposed in this paper.

2.2. Radial basis function neural network

The RBF neural network is composed of an input layer, a hidden layer, and an output layer. Neurons in the hidden layer contain the nonlinear RBF activation function (Figure 1.b). Given input $= \{x_1, x_2, \dots, x_n\} \in R$, the output of the network φ is given by

$$\varphi(x) = \sum_{i=1}^N \omega_i e^{-\frac{\|x - c_i\|^2}{2t}} \quad (4)$$

where N is the number of neurons in the hidden layer, t is the radius (spread) of the radial basis function, c_i is the center vector of i -th neuron, and ω_i is the weight of i -th neuron.

Different types of radial basis functions could be used, but the one used in this paper is the Gaussian function $K(x, c) = e^{-\frac{\|x - c\|^2}{2t}}$. The norm is typically taken to be the Euclidean distance. The Euclidean distance is computed from the point being evaluated to the centre of each neuron, and the radial basis function is applied to the distance to compute the weight for each neuron. The parameters ω_i , c_i and t are determined to optimize the fit between φ and the data [22].

Given n training data points $\{(x_1, y_1), \dots, (x_n, y_n)\}$, the goal of the RBF network learning algorithm is to produce such set c_i and weights ω_i ($i = 1, 2, \dots, M$) so that $(x_j) \approx y_j$ ($j = 1, 2, \dots, n$).

The RBF network positions one or more RBF neurons in the space described by the inputs. The radial basis function for neuron is defined by a centre and a radius. The farther a neuron is from the point being evaluated, the less influence it has and the corresponding weight ω_i is smaller. However, with larger radius, neurons farther from a point have greater influence compared to the case when radius is smaller.

2.3. Support vector machines for regression

Suppose a set of training data $\{(x_1, y_1), \dots, (x_n, y_n)\} \in R^n \times R$, is given, where x_i denotes observations and y_i are the observed response values (targets) for these observations. The SVR technique (Figure 1.c) aims to find an approximating function $f(x)$ that has an ϵ deviation from the observed targets y_i for all training data x_i .

For the linear SVR, this approximating function (4) can be written as

$$f(x) = \langle \omega, x \rangle + b \text{ sa } \omega \in X, b \in \mathfrak{R} \quad (5)$$

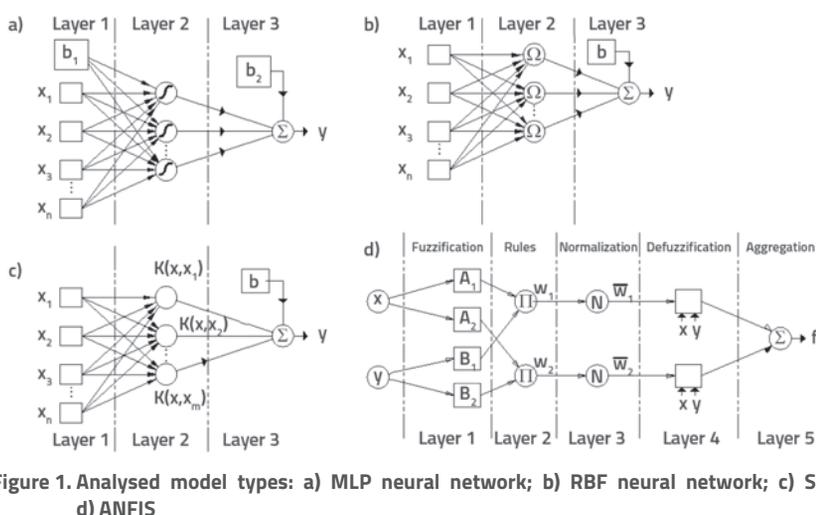


Figure 1. Analysed model types: a) MLP neural network; b) RBF neural network; c) SVR; d) ANFIS

where $\langle \omega, x \rangle$ represents the dot product. In order to get a suitable approximating functions f one should search for small ω and constant C , which will optimize an objective function given in [5]:

$$\min \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \tag{6}$$

$$\text{subject to } \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{7}$$

where ξ_i and ξ_i^* are slack variables, as shown in Figure 2. The constant $C > 0$ is the parameter chosen by the user that denotes the amount of deviation larger than ε that can be tolerated. An increase in C penalizes larger errors (larger ξ_i and ξ_i^*) and, in this way, it leads to the decrease in approximation error. Another parameter chosen by the user is the required precision ε .

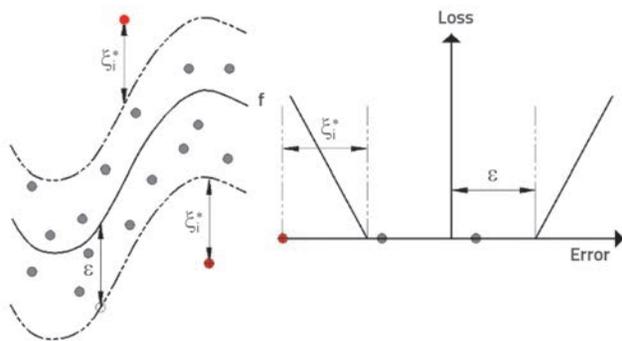


Figure 2. Nonlinear SVR with ε insensitive loss function

The solution of (6) and (7) can be found by introducing a Lagrangian function with the dual set of variables. The final approximate function can be written as

$$\omega = \sum_{i=1}^n (\alpha_i + \alpha_i^*) x_i \text{ zato } f(x) = \sum_{i=1}^n (\alpha_i + \alpha_i^*) x_i, x \tag{8}$$

and the bias term b as

$$b = y_i - \langle \omega, x_i \rangle - \varepsilon \text{ for } 0 < \alpha_i < C \tag{9}$$

$$b = y_i - \langle \omega, x_i \rangle + \varepsilon \text{ for } 0 < \alpha_i^* < C$$

When training patterns x_i are mapped to a higher dimensional space using the mapping $\Phi: x \rightarrow \Phi(x)$, the dot product is $k \langle x_i, x \rangle = \langle \Phi(x_i), \Phi(x) \rangle$. The final approximate function then becomes

$$f(x) = \sum_{i=1}^n (\alpha_i + \alpha_i^*) k \langle x_i, x \rangle + b \tag{10}$$

For linear SVR, the kernel is $k \langle x_i, x \rangle = \langle x_i, x \rangle$, while different kernel functions, e.g. polynomial, sigmoid or RBF, can be used in nonlinear case. RBF and sigmoid kernels used in this paper are defined as presented in [11, 15-18]:

$$k \langle x_i, x \rangle = \exp(-\gamma \|x_i - x\|^2), \gamma > 0 \tag{11}$$

$$k \langle x_i, x \rangle = \tanh(\gamma \langle x_i, x \rangle + r), \gamma > 0 \tag{12}$$

2.4. Adaptive neuro-fuzzy inference system

The adaptive neuro-fuzzy inference system (ANFIS) is a sort of neural network (figure 1.d) that is based on the Sugeno fuzzy inference system. Its inference system corresponds to a set of fuzzy if-then rules that have the learning capability for approximating nonlinear functions [22].

Figure 2.d shows an ANFIS system with two inputs, one output and two rules defined as:

$$\text{Rule 1: If } x = A_1 \text{ and } y = B_1, \text{ then } f_1 = p_1 x + q_1 y + r_1 \tag{13}$$

$$\text{Rule 2: If } x = A_2 \text{ and } y = B_2, \text{ then } f_2 = p_2 x + q_2 y + r_2 \tag{14}$$

where A_i and B_i are the fuzzy sets, f_i is the output within the fuzzy region specified by the fuzzy rule, p_i , q_i and r_i are the linear parameters in then-part (consequent part) of the Sugeno fuzzy model.

The input variables are fuzzified in the first hidden layer. The fuzzification process transforms the values of inputs into fuzzy membership values using a fuzzy membership function. The Gaussian function (15) is used in this study.

$$\mu_{A_i}(x) = e^{-\frac{(x-c)^2}{2\delta^2}} \tag{15}$$

where $\mu_{A_i(x)}$ and $\mu_{B_i(y)}$ are the membership functions, δ and c are the width and the centre of the function, respectively. The antecedent (or premise) part of the rules is constructed in the second layer. The output is called the firing strengths (w_i) and is computed by multiplying all the incoming signals:

$$w_i = \mu_{A_i(x)} \cdot \mu_{B_i(y)}; i = 1, 2 \tag{16}$$

In the next step, the normalized firing strengths \bar{w}_i are computed in layer 3 using:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}; i = 1, 2 \tag{17}$$

The defuzzification is performed in layer 4 using:

$$\bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \tag{18}$$

The final values are calculated in layer 5 using:

$$\text{Output values} = \sum_i \bar{w}_i \cdot f_i; i = 1, 2 \quad (19)$$

2.5. Evaluation and performance assessment

The performance of the models developed in this study was assessed using various statistical performance criteria. The statistical measures considered were the root mean square error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), and the Nash-Sutcliffe efficiency coefficient (CE).

The RMSE is the measure of differences between values predicted by the model o_k and the actually observed (measured) values of the river flow d_k (20). It represents the measure of general accuracy of the model.

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (d_k - o_k)^2} \quad (20)$$

MAE is another general measure of model accuracy, and is used to represent the mean absolute error of the model according to equation (21).

$$MAE = \frac{1}{N} \sum_{k=1}^N |d_k - o_k| \quad (21)$$

The coefficient of determination R^2 is defined by equation (22).

$$R^2 = \left[\sum_{k=1}^N (d_k - \bar{d})(o_k - \bar{o}) \right]^2 \cdot \left[\sum_{k=1}^N (d_k - \bar{d})^2 \sum_{k=1}^N (o_k - \bar{o})^2 \right]^{-1} \quad (22)$$

where o_k is the mean of the values predicted by the model, and d_k is the mean of the observed values.

The Nash-Sutcliffe model efficiency coefficient CE is commonly used to assess the predictive power of hydrological discharge models. It is defined as:

$$CE = 1 - \frac{\sum_{k=1}^N (d_k - o_k)^2}{\sum_{k=1}^N (d_k - \bar{d})^2} \quad (23)$$

CE can range from $-\infty$ to 1. CE = 1 corresponds to a perfect match between the observed and predicted values. Legates and McCabe [23] suggest that at least one relative (e.g. CE or R^2) and at least one absolute (e.g. RMSE or MAE) measure of error should be used for model evaluation.

3. Case study of Buna spring

The described techniques are used to model discharge at the karst spring Buna in Eastern Herzegovina, Bosnia and Herzegovina (Figure 3). The case study area is a part of the Dinaric karst region, composed mainly of soluble carbonate rocks [2]. It is characterized by the existence of surface and underground karst phenomena, such as ponors, springs, estavelles, underground karst channels and stepwise karst poljes with elevations ranging from approximately 900 m to the sea level. In karst poljes, the surface water flows exist and are active only during precipitation period of the year. They sink into ponors, usually along the southern perimeter of poljes. They discharge points are usually situated along the northern perimeter of the lower polje. The Buna River (Figure 3) is one

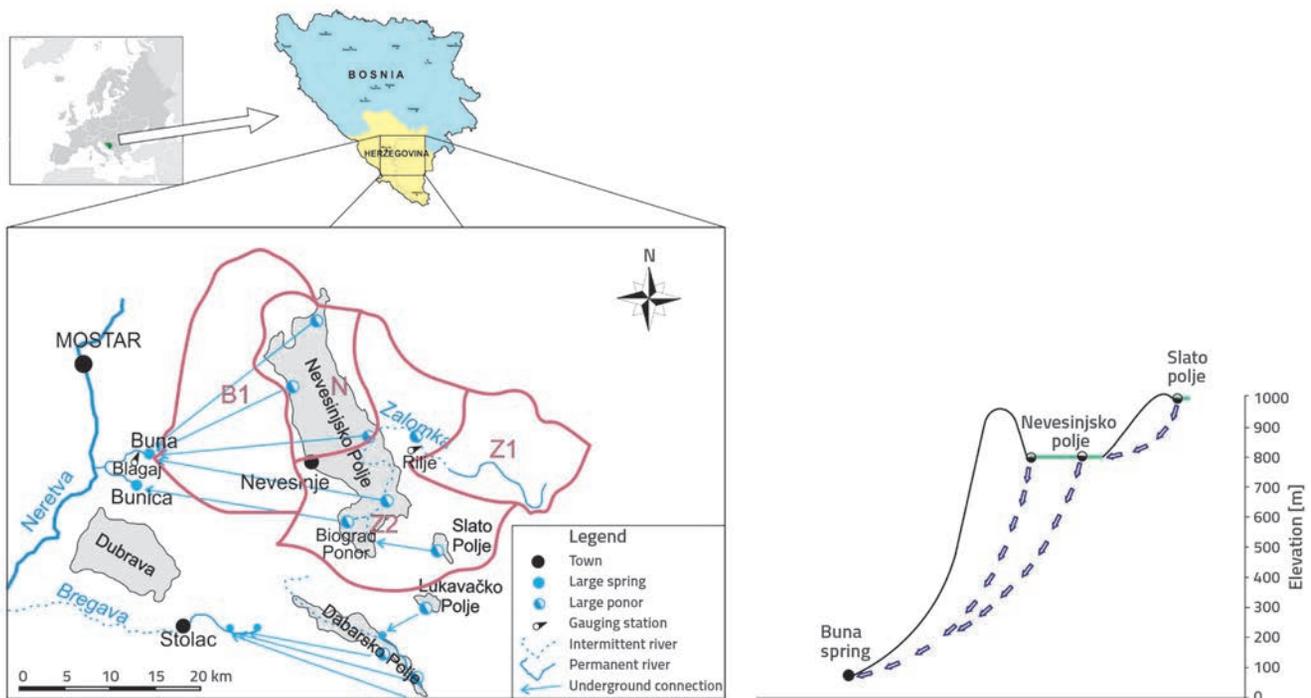


Figure 3. Overview of karst fields and their underground connections

of rare permanent surface flows in Eastern Herzegovina. The Buna River becomes tributary of the Neretva River to the downstream of the city of Mostar. The hydrological catchment area of the Buna spring includes a much wider area compared to its orographic catchment. The catchment area of the Buna Spring includes a broad area of the Zalomka River catchment, the large part of Nevesinjsko Polje, Lukavačko Polje and Slato Polje catchments, and the catchment area between Nevesinjsko Polje and a part of Velež Mountain. The most important part of the Buna Spring catchment area is the mountainous area between Nevesinjsko Polje and the spring.

The Buna Spring is one of the most famous springs in Dinaric karst. It is formed at the contact of Cretaceous limestones and Eocene flysch, at the elevation of 36 m above sea level. The gauging station Blagaj is situated directly to the downstream of the spring. The mean annual discharge is $Q_{mean} = 23.7 \text{ m}^3/\text{s}$, while the minimum and maximum discharges are $Q_{min} = 2.95 \text{ m}^3/\text{s}$ and $Q_{max} = 380 \text{ m}^3/\text{s}$, respectively [2]. The catchment area of the Buna Spring is divided into several sub-catchment areas (Figure 3). The area marked "B1" is the direct orographic catchment area. It is a mountainous area (Velež foothills) and the water from this area flows directly to the Buna Springs. The water from the area marked "N" sinks into swallow holes (ponors) at the western part of Nevesinjsko Polje, flows through underground karst channels and discharges at the Buna Spring. A part of the water from areas Z1 and Z2 that sinks along the Zalomka River also discharges at the Buna Spring. This part of the catchment is included in calculation through discharges at the gauging station Rilje situated at the Zalomka River.

Methods used to model the Buna Spring discharge are based on the "black-box" principle. Inputs and outputs are analysed in order to obtain the mapping operators. Input values are the average daily rainfall at the catchment area, river flow at the hydrological station Rilje, and flow during previous days at the gauging station Blagaj. On site research, performed by tracing the water at swallow holes, provided information about the time when the tracer appears at the spring, which led to the conclusion that the time needed for the system to react was two days (due to karst nature of the terrain) [2]. Since there is a delay effect of the input values compared to output values, a special analysis was conducted indicating that the model accuracy is not affected by the two-day time extension. Input vectors are formed using precipitation and flow during the previous two days that is:

$$Q_i^{BL} = f \left\{ P_{i-2}^{B1}, P_{i-1}^{B1}, P_{i-2}^N, P_{i-1}^N, Q_{i-2}^r, Q_{i-1}^r, Q_{i-2}^{BL}, Q_{i-1}^{BL} \right\} \quad (24)$$

where

- P_i^{B1} - Daily value of precipitation on i -th day for zone B1,
- P_i^N - Daily value of precipitation on i -th day for zone N,
- Q_i^r - Daily flow on i -th day at Rilje Station,
- Q_i^{BL} - Daily flow on i -th day at Blagaj Station (1 day ahead).

The total number of available data is 4747. 70 % of the collected data is used for network training, 15 % of the data is used for validation, and 15 % of data is used for network testing. The data for network training covers the time span from 3 September 1971 to 7 October 1980. The data for validation covers the time span from 8 October 1980 to 19 September 1982, while the network testing data covers the time span from 20 September 1982 to 31 August 1984. Flow values which refer to the training period range from the minimum value of $2,8 \text{ m}^3/\text{s}$ to the maximum value of $115,35 \text{ m}^3/\text{s}$, while extreme values for the validation set range from $3,1 \text{ m}^3/\text{s}$ to $108,00 \text{ m}^3/\text{s}$, and extreme values for the test set range from $2,95 \text{ m}^3/\text{s}$ to $94,9 \text{ m}^3/\text{s}$. The same training, validation and test sets are used for all models.

4. Results and discussion

Rainfall and runoff forecasting results for the Buna River, as based on four machine learning techniques: MLP and RBF neural networks, SVR and ANFIS, are presented. The same set of input features is used for all models, i.e. 8 input variables that define precipitation and flow in the previous two days.

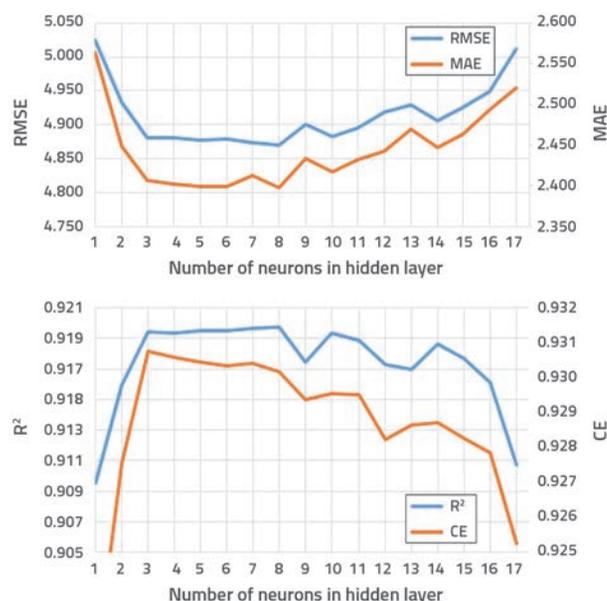


Figure 4. Comparative overview of statistical performance values for different architectures of MLP neural networks: a) root mean square error (RMSE) and mean absolute error (MAE); b) coefficient of determination (R^2) and Nash-Sutcliffe efficiency coefficient (CE)

The models were assessed using the root mean square error (RMSE), mean absolute error (MAE), coefficient of determination (R^2) and Nash-Sutcliffe efficiency coefficient (CE) as performance measures. The performance was evaluated using a test data set that was not included in the training. The same test data set was used for all the algorithms.

For the MLP neural network, the number of neurons in the input layer is determined by input variables, i.e. it consists of

8 neurons, while there is only one neuron in the output layer. The number of neurons in the hidden layer was determined experimentally. The upper limit, i.e. the maximum number of neurons in the hidden layer that can be used in modelling, was set to 17 neurons, according to equations (2) and (3).

Multifold training of the network was done for the number of neurons in the hidden layer between 1 and 17, as shown in Figure 4. One can observe that an optimum number of neurons in the hidden layer equals 8, which is confirmed using three quality criteria, RMSE, MAE and R^2 (Figure 4).

The RBF neural network contains the same number of neurons in the input and the output layer as MLP, while an optimum number of neurons in the hidden layer was determined experimentally. It ranges from 1 to 100 neurons, as shown in Figure 5. A larger number of neurons was used as the upper limit compared to MLP because the radial basis function in the hidden layer produces a significant non-zero response only when the input falls within a small localized region of the input space [24].

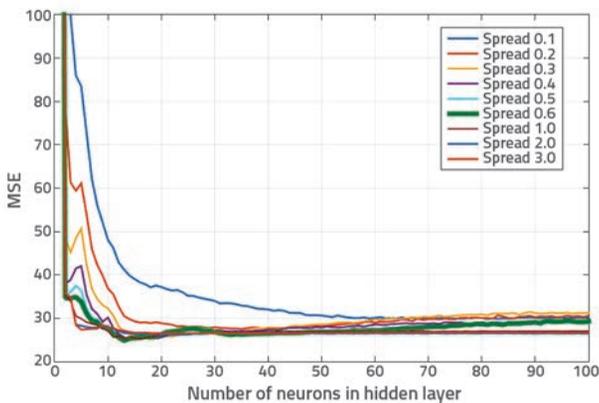


Figure 5. Performance of RBF model with different numbers of neurons in hidden layer and different spreads

The Gaussian function was used as the transfer function. The parameter spread controls the width of the Gaussian, which means that it determines how much of the input space the RBF neuron will respond to. The spreads between 0.1 and 3 with step 0.1 were examined in order to find the best value that gives the minimum MSE. It can be observed that

an optimal number of neurons in the hidden layer equals 13 with spread 0.6 (green line in Figure 5).

The SVR model was tested using linear, RBF, and sigmoid kernels. The normalization, which scales all input data into the range [0,1], was done prior to training and testing. Optimum model parameters were determined using grid search [25] for all kernels ($C = 0.038650$ and $\varepsilon = 0.019523$ for the linear kernel; $C = 0.105115$; $\varepsilon = 0.005524$; $\gamma = 0.297305$ for the RBF kernel; and $C = 0.197405$; $\varepsilon = 0.0191385$; $\gamma = 0.178585$ for the sigmoid kernel). The LIBSVM library was used for SVR implementation [26].

A hybrid-learning algorithm was applied for tuning the ANFIS parameters. The gradient method was employed to tune the premise parameters, while the least squares method was used to identify the consequent linear parameters. The input layer consists of eight input variables, while the output layer contains one output variable. The input and output variables are linearly normalized to [0,1] interval before being used by the ANFIS. The number of fuzzy membership functions of each input was set to 2. The input membership functions were Gaussian. An appropriate structure has to be established in order to begin training of the fuzzy inference system (FIS). In this study, the FIS structure was generated by means of subtractive clustering [10, 27], which is used to determine regions in the feature space with high densities of data points. The point with the maximum number of neighbours is selected as the cluster centre. The data points within a previously specified fuzzy radius are subtracted, and the algorithm looks for a new point with the highest number of neighbours. This process iterates until all data points are examined. Two clusters are used in this study. The fuzzy radius was changed from 0.15 to 0.50 with step 0.01. The value 0.22 was determined as an optimum value using the RMSE as the criterion. The number of training epochs was set to 100.

The performance of all models using RMSE, MAE, R^2 and CE as performance measures is shown in Table 1. The best performing model is highlighted in each column. When RMSE and MAE are used as performance measure, then the ANFIS model has the best performance. For R^2 and CE, all the models perform similarly, with the SVR using linear kernel slightly outperforming other models for the R^2 performance measure, and the RBF neural network being the best model for the CE performance measure. In both cases, the ANFIS was the second best model.

Table 1. Performance of MLP and RBF neural networks, SVR and ANFIS models

Model	Performance measure			
	RMSE	MAE	R^2	CE
MLP	4.870	2.399	0.920	0.930
RBF	4.993	2.544	0.927	0.947
SVR-RBF kernel	4.715	2.272	0.925	0.928
SVR-Linear kernel	4.819	2.507	0.946	0.921
SVR-Sigmoid kernel	4.815	2.459	0.922	0.921
ANFIS	4.615	2.255	0.928	0.937

Hence, based on all four criteria, we have chosen ANFIS as the most optimal model for short-term prediction of one day ahead river flow based on precipitation and flow during the previous two days, and have performed all subsequent analyses using this model. The comparison of the observed values and values modelled by the ANFIS, as well as regression plots for train and test data, are given in Figure 6.

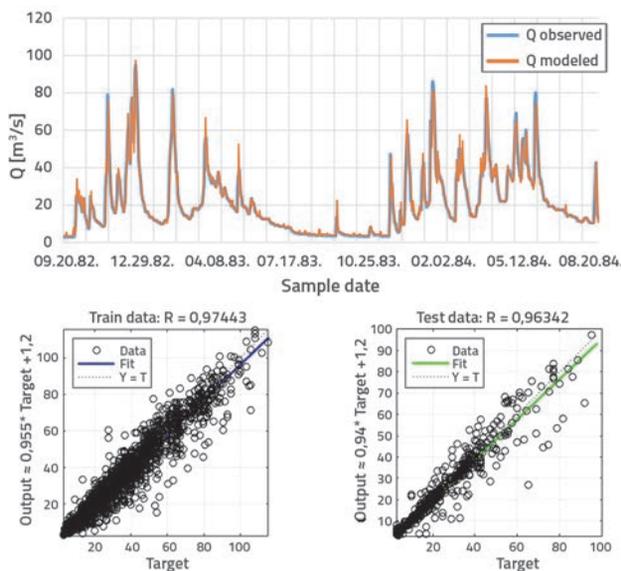


Figure 6. a) Comparison of observed values and values modelled by ANFIS, b) regression plots for train and test data for ANFIS

All eight input variables (daily precipitations and daily flows) were used for all previously tested models, without assessing the importance of each individual variable on the model prediction accuracy. The variable selection activity was performed in further analysis. Only the most relevant variables were used in the modelling, while irrelevant and redundant input variables that do not contribute to the accuracy of the predictive model, or that may in fact decrease its accuracy, were removed. This not only reduces the complexity of the model, but it also speeds up the learning task. The exhaustive search technique [28, 29] was applied for this task. $2^N - 1 = 255$ possible combinations of variables and 255 different ANFIS models were generated for a given set of $N=8$ input variables, designated as $P_{i-2}^{B1}, P_{i-1}^{B1}, P_{i-2}^N, P_{i-1}^N, Q_{i-2}^r, Q_{i-1}^r, Q_{i-2}^{BL}, Q_{i-1}^{BL}$. The 10-fold cross validation was used to determine the MSE for each model. An optimum subset of input variables is the one that gives the model with the smallest MSE.

Table 2. Models with subset of input variables

Model	Input variables								Performance measure			
	P_{i-2}^{B1}	P_{i-1}^{B1}	P_{i-2}^N	P_{i-1}^N	Q_{i-2}^r	Q_{i-1}^r	Q_{i-2}^{BL}	Q_{i-1}^{BL}	RMSE	MAE	R ²	CE
1.	1	1	0	1	1	1	1	1	4.633	2.421	0.942	0.953
2.	1	1	0	1	0	1	0	1	4.693	2.520	0.940	0.952
3.	1	1	1	1	1	1	1	1	4.615	2.255	0.928	0.937

Furthermore, the frequency of occurrence of each input variable in all generated models was assessed and, based on this frequency, the most relevant variables were determined. The results of this analysis, provided in Figure 7, show that variable 8 (Q_{i-1}^{BL}) is relevant in all models, while the frequency of variables 1, 2, 4, and 6 ($P_{i-2}^{B1}, P_{i-1}^{B1}, P_{i-1}^N, Q_{i-1}^r$) is lower by approximately 20-30 %. These variables are followed by variables 5 and 7 (Q_{i-2}^r, Q_{i-2}^{BL}) which were by 50 % less frequent. The least important input variable is variable 3 (P_{i-2}^N).

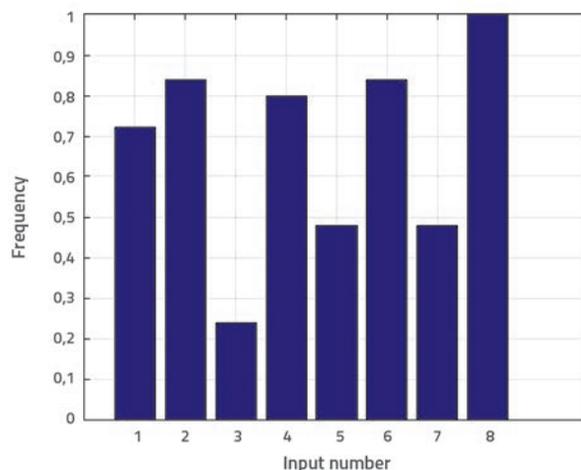


Figure 7. Frequency of occurrence of input variables in models

The results for the models that omit only variable 3 (model 1 in Table 2) and all three variables (model 2 in Table 2) are provided, as the variables 3, 5 and 7 ($P_{i-2}^N, Q_{i-2}^r, Q_{i-2}^{BL}$) are significantly less important compared to other variables. Note that the binary value (1 or 0) denotes whether a particular variable is included in the model or not. The comparison of the obtained performance with the results using all 8 input variables (model 3 in Table 2) reveals that using only the subset of 5 input variables gives comparable results using RMSE as performance measure, or even outperforms it when R^2 and CE are used as performance measures.

5. Conclusion

This paper provides a comparative analysis of a variety of state-of-the-art machine learning algorithms for the task of short-term forecasting of river flow in a karst region. Several neural network architectures were tested, i.e. MLP and RBF neural

networks, as well as the ANFIS, which incorporates both neural networks and fuzzy logic principles, and also the SVR. Different relative (*CE* or *R²*) and absolute (*RMSE* or *MAE*) measures were used for performance evaluation.

Results relating to the one day ahead flow forecast for the Buna River in Bosnia and Herzegovina, based on the precipitation and flow over two previous days, show that other tested models have been outperformed by the ANFIS model. The analysis has shown that the combination of fuzzy logic and neural networks overcomes the difficulties and inherent limitations of neural networks, enabling a better prediction model.

An in-depth analysis of the importance and influence of input variables on model performance is also provided. The

precipitation and flow data were collected at several locations (two for precipitation and two for flow). Eight input variables were used in all previously tested models. However, it was established that individual variables do not equally contribute to the performance, and that even better results can be obtained when using only a subset of five input variables, i.e. when three less relevant features are removed. At the same time, the complexity of the model has been reduced and the learning task has been accelerated.

Further research will include analysis of additional factors that could influence prediction accuracy, such as the ground water level, geological composition of river basins, morphology, and vegetation, as these factors can contribute to the universality of models.

REFERENCES

- [1] Peng, G., Leslie, L. M., Shao, Y.: Environmental Modelling and Prediction, Eds. Springer-Verlag, pp. 215-273, 2002.
- [2] Milanović, P.: Karst Istočne Hercegovine i dubrovačkog priobalja, Beograd, 2006.
- [3] Makropoulos, C., Koutsoyiannis, D., Stanić, M., Djordjević, S., Prodanović, D., Dašić, T., Prohaska, S., Maksimović, Č., Wheeler, H.: A multi-model approach to the simulation of large scale karst flows, *Journal of Hydrology*, 348 (2008) 3-4, pp. 412-424, <https://doi.org/10.1016/j.jhydrol.2007.10.011>
- [4] Lazarevska, M., Knežević, M., Cvetkovska, M., Ivanišević, N., Samardžioska, T., Trombeva-Gavriloska, A.: Prognošički model za određivanje požarne otpornosti AB stupova, *GRAĐEVINAR*, 64 (2012) 7, pp. 565-571.
- [5] Yaseen, Z.M., El-shafie, A., Jaafar, O., Afan, H.A., Sayl, K.M.: Artificial intelligence based models for stream-flow forecasting: 2000-2015, *Journal of Hydrology*, 530 (2015), pp. 829-844, <https://doi.org/10.1016/j.jhydrol.2015.10.038>
- [6] Afan, H.A., El-shafie, A., Mohtar, W.H.M.W., Yaseen, Z.M.: Past, present and prospect of an Artificial Intelligence (AI) based model for sediment transport prediction, *Journal of Hydrology*, 541 B (2016), pp. 902-913.
- [7] Raghavendra, S.N., Deka, P.S.: Support vector machine applications in the field of hydrology: A review, *Applied Soft Computing*, 19 (2014), pp. 372-386, <https://doi.org/10.1016/j.asoc.2014.02.002>
- [8] Ireland, G., Volpi, M.P., Petropoulos, P.G.: Examining the Capability of Supervised Machine Learning Classifiers in Extracting Flooded Areas from Landsat TM Imagery: A Case Study from a Mediterranean Flood, *Remote Sensing*, 7 (2015) 3, pp. 3372-3399.
- [9] Kostic, S., Stojkovic, M., Prohaska, S.: Hydrological flow rate estimation using artificial neural networks: Model development and potential applications, *Applied Mathematics and Computation*, 291 (2016), pp. 373-385, <https://doi.org/10.1016/j.amc.2016.07.014>
- [10] Sanikhani, H., Kisi, O., Kiafar, H., Ghavidel, S.Z.Z.: Comparison of Different Data-Driven Approaches for Modeling Lake Level Fluctuations: The Case of Manyas and Tuz Lakes (Turkey), *Water Resources Management*, 29 (2015) 5, pp. 1557-1574.
- [11] Rezaei-zadeh, M., Tabari, H., Yazdi, A.A., Isik, S., Kalin, L.: Flood flow forecasting using ANN, ANFIS and regression models, *Neural Computing and Applications*, 25 (2014) 1, pp. 25-37.
- [12] Bui, D.T., Pradhan, B., Nampak, H., Bui, Q.T., Tran, Q.A., Nguyene, Q.P.: Hybrid artificial intelligence approach based on neural fuzzy inference model and metaheuristic optimization for flood susceptibility modeling in a high-frequency tropical cyclone area using GIS, *Journal of Hydrology*, 540 (2016), pp. 317-330, <https://doi.org/10.1016/j.jhydrol.2016.06.027>
- [13] Chang, F.J., Tsai, Y.H., Chen, P.A., Coynel, G.: Modeling water quality in an urban river using hydrological factors - Data driven approaches, *Journal of Environmental Management*, 151 (2015), pp. 87-96, <https://doi.org/10.1016/j.jenvman.2014.12.014>
- [14] Granata, F., Gargano, R., de Marinis, G.: Support Vector Regression for Rainfall-Runoff Modeling in Urban Drainage: A Comparison with the EPA's Storm Water Management Model. *Water*, 8 (2016) 3.
- [15] Hosseini, S.M., Mahjouri, N.: Integrating Support Vector Regression and a geomorphologic Artificial Neural Network for daily rainfall-runoff modeling, *Journal Applied Soft Computing*, 38 (2016) C, pp. 329-345.
- [16] Gizaw, M.S., Gan, T.Y.: Regional Flood Frequency Analysis using Support Vector Regression under historical and future climate, *Journal of Hydrology*, 538 (2016), pp. 387-398, <https://doi.org/10.1016/j.jhydrol.2016.04.041>
- [17] He, Z., Wen, X., Liu, H., Du, Jun.: A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region, *Journal of Hydrology*, 509 (2014), pp. 379-386, <https://doi.org/10.1016/j.jhydrol.2013.11.054>
- [18] Gong, Y., Zhang, Y., Lan, S., Wang, H.: Comparative Study of Artificial Neural Networks, Support Vector Machines and Adaptive Neuro Fuzzy Inference System for Forecasting Groundwater Levels near Lake Okeechobee, Florida, *Water Resources Management*, 30 (2016) 1, pp. 375-391.
- [19] Beale, M.H., Hagan, M.T., Demuth, H.B.: *Neural Network Toolbox™ User's Guide*, The MathWorks, Inc, 2010.

- [20] Matić, P.: Predviđanje dotoka rijeke Cetine u akumulaciju Peruća pomoću umjetne neuronske mreže, Kvalifikacijski doktorski rad, Fakultet elektrotehnike strojarstva I brodogradnje u Splitu, Sveučilište u Splitu, 2012.
- [21] Kingston, G.B.: Bayesian Artificial Neural Networks in Water Resources Engineering, doctoral dissertation, School of Civil and Environmental Engineering, Faculty of Engineering, Computer and Mathematical Science, University of Adelaide, Australia, 2006.
- [22] Araghinejad, S.: Data-Driven Modeling: Using MATLAB® in Water Resources and Environmental Engineering, Water Science and Technology Library, 67 (2014), pp. 139-193, <https://doi.org/10.1007/978-94-007-7506-0>
- [23] Legates, D.R., McCabe, G.J. Jr.: Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation, Water Resources Research, 35 (1999) 1, pp. 233-241.
- [24] Safari, M.J.S., Aksoya, H., Mohammadi, M.: Artificial neural network and regression models for flow velocity at sediment incipient deposition, Journal of Hydrology, 541,B (2016), pp.1420-1429.
- [25] Chang, C.C., Lin, C.J., LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1-27:27, 2011.
- [26] <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [27] Sumathi, S., Paneerselvam, S.: Computational Intelligence Paradigms: Theory & Applications using MATLAB, CRC Press, pp. 351-393, 2010.
- [28] Theodoridis, S., Koutroumbas, K.: An introduction to Pattern Recognition: A MATLAB® Approach, Academic Press, pp. 122-135, 2009.
- [29] Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Academic Press, pp. 261-318, 2009, <https://doi.org/10.1016/B978-1-59749-272-0.50007-4>