

CALCULATION OF THE RIVER FLOW WITH DIFFERENT PROBABILITIES OF OCCURRENCE USING ARTIFICIAL NEURAL NETWORK

Ioan Florin MOLDOVAN¹

DOI: 10.21163/GT_2020.151.15

ABSTRACT:

The aim of this paper is to calculate the river flow with different necessary probabilities of occurrence, by the Artificial Neural Network (ANN) method. The studied ANN uses a radial basis function (RBF) architecture, with an input layer, a hidden layer and an output layer. Three series of maximum annual flow were used for the input layer. The series of maximum annual flow were collected from gauge stations, situated on three tributaries of the Mures river, Romania. The number of the neurons in the input layer is variable, according to the number of the flow values in the series. It means that the architecture of the ANN is different from one series to another. Consequently, the ANN has to repeat the calculation stages for each series. The results revealed that the ANN interpolates very well in the range of the values of the series. Each time, the stop condition of the target error was met. A comparison was made between the values of the flow calculated by the ANN method and by the statistical (classical) method using the Pearson type 3 distribution. The comparison showed that there are significant differences for the flow values corresponding to the 0.01% and 0.1% probabilities, and less or not significant for the rest.

Keywords: *River flow, Probabilities of occurrence, Artificial Neural Network (ANN), Romania.*

1. INTRODUCTION

In many domains, the scientists tried to imitate nature, including to imitate the functioning of the the human brain, which is a very complex “computer”. The Artificial Neural Networks (ANNs) are inspired by the human nervous system and their architecture is based on parallel processing (Aichouri et al, 2015).

In the last decades, the Artificial Neural Networks (ANNs) were developed on a large scale, in different domains, like arts, natural sciences, social sciences, industry, agriculture, sports, entertainment. The development was due to the features that makes the ANNs very effective. The ANN is an interpolation-approximation model. An important feature is that in an ANN the information is distributed along the artificial neural network, being stored in the synaptic weights between the neurons. This property gives the possibility of the ANN to be tolerant to input data with distortions. That makes ANN very effective for distorted character recognition (Enăchescu, 1998). A well-known application of this property is used in the field of the optical character recognition (OCR).

Another important propriety is that an ANN is able to simulate non-linearity in a system. This way it can distinguish between relevant and irrelevant data characteristics (de Vos & Rientjes, 2005).

¹*Babes-Bolyai University, Faculty of Geography, 400084, Cluj-Napoca, Romania,
moio13898@studmail.ubbcluj.ro*

These features recommend the ANNs to be used in hydrology, as well. Good results were obtained for rainfall-runoff models, as mentioned in many studies (e.g. Dar, 2017; de Vos & Rientjes, 2005; Dounia, Sabri, & Yassine, 2014; Firuzi et al, 2011; Kashiwao et al, 2017; Lallahem & Mania, 2003; Sarkar & Kumar, 2012).

There are many architectures types for the ANN models. A very common neural network architecture is the Multi-Layer Perceptron (MLP). A MLP consists of three types of layers: input, hidden, and output. The input data are introduced via the input layer. The processing is done in the hidden layers, and the result is presented to the output layer. In a MLP the data flows in one direction, from input to output, and due to this property it is named feed forward neural network (Tanty & Desmukh, 2015).

For this study more appropriate is another type of architecture, the radial basis function (RBF). It is, also, a feed forward neural network. The structure of the RBF network is similar, with three layers, in which the hidden layer performs a fixed nonlinear transformation with no adjustable parameters (Leonard et al, 1992, as cited by ASCE Task Committee, 2000). The hidden layer has nodes and a "center" which can be considered the weight vector of the hidden layer. To measure the distance between the input vector and the center, it is used the standard Euclidean distance. For each node is computed the distance between the input vector and the center, and the result is transformed by the activation function, which is a nonlinear function (ASCE Task Committee, 2000).

A very important field in hydrology is the calculation of the river flow value for a certain probability of occurrence. These values are used for practical applications (e.g. hydrotechnical construction designs or bridge designs). Thus, it could be necessary to calculate the flow for the corresponding probabilities: 0,01%, 0,1%, 1%, 2%, 5% and 10% for the maximum flow, respectively 80%, 90%, 95% and 99% for the minimum flow. This study will analyze only the maximum flow series.

The annual maximum flow value represents the peak of the highest flood during the year, which can be very destructive, with loss of human life and huge damages. For this reason, the hydrologists are highly interested in calculating the maximum river flow with different probabilities of occurrence, in order to use the maximum flow value in the domain of flood protection. Usually, the standards in each country establishes what probabilities have to be calculated according the type of the construction that to be designed. It is the case for the Romanian legislation, too.

The most used method for calculations is the statistical (classical) method. Still, the advantage of the ANN method is that it can reach very low interpolation error, as presented in this study.

2. STUDY AREA AND DATA

The Mureș river basin has 28310 km² and a length of 761 km of the main river. It represents about 12% of the surface of Romania.

For this study data from three gauging stations were analyzed on three tributaries: Ampoi, Aries and Strei (**Fig. 1**).

Barabant gauge station is situated on the Ampoi river. The surface of the catchment area is 576 km², the length of the river is 57 km and the average altitude is 700 m. Petreni gauge station is situated on the Strei river, the surface of the catchment area is 1983 km² and the average altitude is 911 m. Finally, Turda gauge station is situated on the Aries river, the surface is 3005 km², the length of the river is 166 km and the average altitude is 818 m.

The number of the values in each series is variable as can be seen in Table 2.

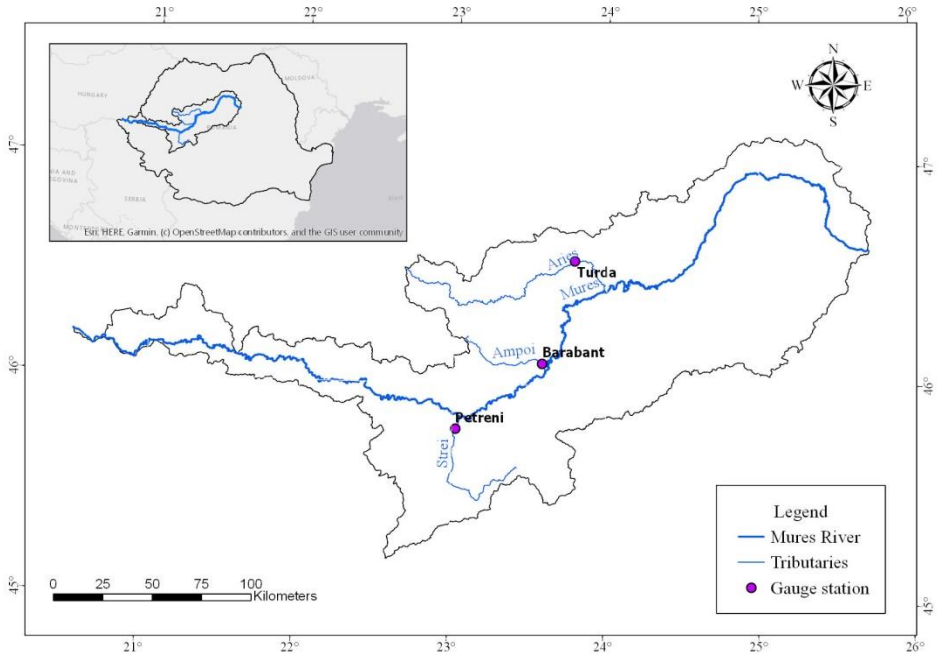


Fig. 1 – The Mureș River Basin location in Romania.

3. METHODOLOGY

3.1 Theoretical aspects of the ANNs

The general theoretical aspects of the ANNs are described in many articles and books.

In general, an ANN is composed of neurons disposed in layers. A neuron “j” in a layer (Fig. 2) computes an output, based on the weighted sum of all its inputs (S_j), and based on the activation function ($f(S_j)$).

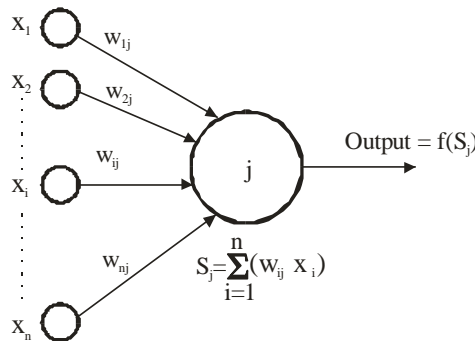


Fig. 2 - Activation of a single neuron “j” (according to Dawson & Wilby, 2001).

The activation functions and their derivative are simple to calculate. They can be identical, logistic sigmoid, linear, threshold, Gaussian or hyperbolic tangent functions and they depend on the type of network used. The most common functions are the logistic sigmoid and the hyperbolic tangent.

In the logistic sigmoid activation function (1), x represents the weighted sum of inputs to the neuron and $f(x)$ represents the neuron's output (Dawson & Wilby, 2001):

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

The ANN used in this study has a RBF architecture with three layers: the input layer, one hidden layer and the output layer. It is a feed-forward with back-propagation (FFBP) ANN. The activation of the neuron "j" is the same as already mentioned (**Fig. 2**).

Some proprieties of the RBF architecture will be presented in the following (according to Enăchescu, 1998).

The input layer has n neurons (2), based on the number of values in the series:

$$x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)}) \quad (2)$$

where: - i represents the i -th series of data

- x_i represents the input data vector of the i -th series

- n is equal to the number of values in the input data series

The Bias term (the red circle in **Fig. 3**) can be considered implicitly or explicitly in the model. In this study it was considered implicitly.

The hidden layer has the number of neurons equal to the number values N of the training set T (3):

$$T = \{(x_i, f(x_i)) | i = 1, 2, \dots, N\} \quad (3)$$

where the training set is a subset of the entire set of input data. In our case, $N=n$, because there are no subsets of the data series.

The activation functions (4) for the hidden layer are:

$$f(x) = \sum_{i=1}^N w_i G(x; x_i) \quad (4)$$

and are different for each neuron, for example the k -th neuron has the activation function centered in the point x_k , and the function is (5):

$$G(x; x_k) = G(x - x_k) \quad (5)$$

The output layer has one neuron, because the ANN will calculate one value of flow at a time.

Because of the small number of values in each data series, we will consider the following: the number of the neurons in the input layer is equal to the number of the input values in each series of annual maximum flow; the number of the neurons in the hidden layer is equal to the number of the neurons in the input layer.

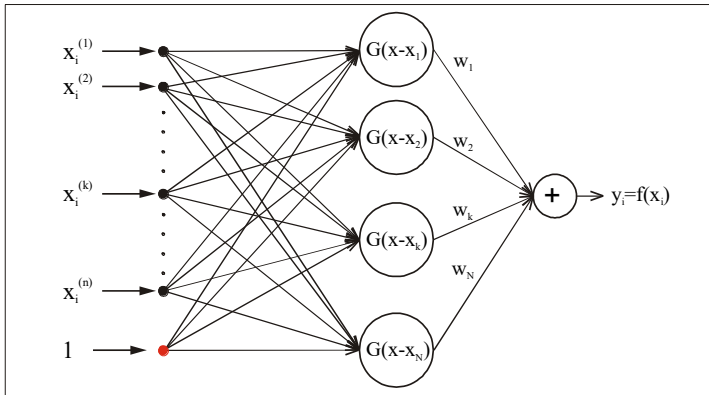


Fig. 3 - The architecture of the RBF (according to Enăchescu, 2008).

In this study, the identical function (7) was considered for the activation function of the hidden layer:

$$f(x_i^{(k)}) = x_i^{(k)} \quad (7)$$

where: $x_i^{(k)}$ represents the k -th data value in the i -th series of data.

The activation functions in the output layer are Radial Basis Functions (RBF) of Gaussian type (8):

$$G(x - x_i) = e^{-\frac{(x-x_i)^2}{\sigma_i^2}}, \quad i = 1, 2, \dots, N \quad (8)$$

where: σ_i represents the diameter of the i -th cluster.

The number of the clusters (K) is equal to the number of the input values ($K = N = n$). It means that each data is a cluster and the activation functions are „centered” on the data values. According to this consideration (9), $\sigma_i = 1$:

$$G(x - x_i) = e^{-(x-x_i)^2}, \quad i = 1, 2, \dots, N \quad (9)$$

The final value of the neuron in the output layer is weighted by the function (10):

$$y = \frac{\sum_{i=1}^K w_i G(x - x_i)}{\sum_{i=1}^K G(x - x_i)} \quad (10)$$

Based on the above considerations, the architecture for this study is represented in **Fig. 4**.

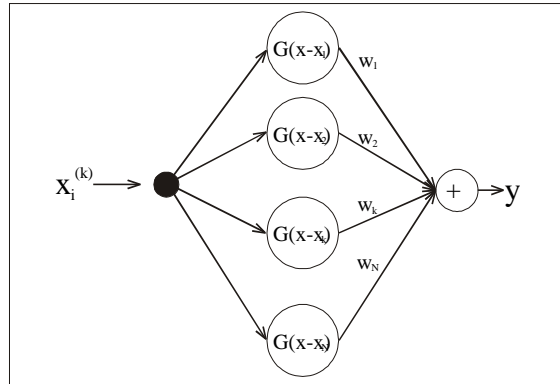


Fig. 4 - The RBF architecture for this study, corresponding to the k -th value from the i -th input series.

A descendant gradient algorithm was proposed for the model. The learning error E_l was calculated by formula (11), the root mean square error - RMSE (Sarkar & Kumar, 2012):

$$E_l = RMSE = \sqrt{\frac{\sum_{i=1}^n (z_i - y_i)^2}{n}} \tag{11}$$

where:

- z_i represents the i -th input value in the annual maximum flow series
- y_i represents the i -th corresponding value of the flow calculated by the ANN
- n represents the number of values in the annual maximum flow series

The algorithm stops when the condition for the error E_l is met (12):

$$E_l \rightarrow \min \tag{12}$$

Practically, the stop condition (12) can be written as follows (13):

$$E_l = \sqrt{\frac{\sum_{i=1}^n (z_i - y_i)^2}{n}} \leq \varepsilon \tag{13}$$

where: ε is the target error (e.g. $\varepsilon = 10^{-10}$)

After the calculation of the error E_l , the Back-Propagation algorithm, which uses the derivative of the function (9), will propagate the error back to the neurons in the hidden layer, by modifying the weights. Then a new epoch (iteration) starts, which will stop when the condition (13) is reached again.

A supplementary stop condition was imposed, by limiting the number of the learning epochs to 10.000, in case that the program cannot reach the target error. This can happen if the function E_l is blocked in a local minimum value of the Equation (13).

There are many programs for ANN models, including free versions. For this study, it was preferred to design a program in C# language (with Microsoft Visual Studio Community 2015), because it gives more flexibility to test different versions of architecture and methodology and to plot the results in a desired manner.

For instance, the results of the ANN were compared with the results given by the statistical (classical) method. A Pearson type 3 density probability function was calculated for each series of input data and plotted on the same graphics, as presented in paragraph 3.2.2 and in Chapter 4.

3.2 Practical considerations

In general, there are 2 main stages necessary to obtain the results from an ANN. The first is the “learning” stage and the second is the „generalization” stage.

3.2.1 The learning stage

In this study, the input values are the empirical probabilities. The output values are the annual maximum flows. After each series has been sorted in descending order, for each annual maximum flow the empirical probability (pe_i) is calculated, based on the Weibull formula (14), in general used to calculate the probabilities for the maximum flow:

$$pe_i = \frac{i}{n+1}, \quad (14)$$

where:

- i represents the position in the sorted series of the i -th value
- n represents the total number of the values in the series

This way, n pairs (pe_i, z_i) can be obtained, composed by the annual maximum flow value z_i and its corresponding empirical probability pe_i .

Due to the fact that the activation functions (9) have the results in the range of [0, 1], before starting the learning stage, the normalization (McCaffrey, 2014) of the z_i values was made by the Min-Max Normalization formula (15):

$$z_i = \frac{z_i^{ini} - z_{min}}{z_{max} - z_{min}} \quad (15)$$

where:

- z_i is the normalized value based on the initial z_i^{ini} value
- z_{max} and z_{min} are the maximum respectively the minimum values of the series

Due to the fact that the input data consists of the pairs (pe_i, z_i) where the output values are known (z_i), the proper learning method for the ANN is the supervised learning. Also, because the number of values in each series is different, the architecture of the ANN, meaning the number of neurons in the input layer, is different for each series of data. Consequently, the architecture is modified when the series is changed, and the ANN has to repeat the calculation stages for each series. The architecture is not modified during the calculation stages.

In the supervised learning stage, the input values are presented to the input layer of the ANN, the known output values are presented to the output layer, and the ANN will learn to calculate the output values with a given target error, based on the Back-Propagation algorithm.

At the beginning of the learning stage, the weights are initiated with random values comprised in the interval (0, 1). In each epoch (iteration) the learning error (11) is calculated, and the weights w_k (Fig. 4) are modified accordingly by the Back-Propagation algorithm. The learning stage stops when the condition (13) is met. Now, the ANN has “learned” to calculate the values of the annual maximum flow, based on the pairs (pe_i, z_i) . Practically, the “learning” consists in the fact that the values of the weights were fine-tuned during this stage. The final values of the weights will be used during the generalization stage.

3.2.2 The generalization stage

Based on the learning stage, in which the ANN has learned to calculate the output values, the ANN can calculate the annual maximum flow for the necessary probabilities, other than the empirical probabilities (14) that were used in the learning stage. This is done by running the program with the final weights calculated in the learning stage, for the necessary probabilities, by presenting to the input layer the series of the necessary probabilities, and obtaining at the output layer the corresponding flow values. In the generalization stage, the weights will no longer be modified.

The representation of the data is made on Normal probability paper. This representation has the propriety that it “zooms out” the distances between the extreme values of the graphics (on the left and right sides).

4. RESULTS AND DISCUSSIONS

For all three series of data the program has reached the target error in less than 2000 epochs (Fig. 5).

The processing duration was less than 1 second, for each series, on a computer with Microsoft Windows 8.1, Intel i7 processor and 4 GB RAM memory. In Fig. 5, can be seen that the input data (red “X”) are very well approximated in the learning stage (the blue dots). The results are similar for all series of data provided by the other two gauging stations (Table 1).

In the generalization stage the ANN has calculated the corresponding flow for the probabilities mentioned in the introduction. On the graphics, these probabilities were represented by the green dots.

Table 1.

The results of the ANN model.

Gauge station	River	Learning error (RMSE)	Generalization error (GE)	Number of epochs ¹	Processing duration ² (seconds)
Barabant	Ampoi	$9,95 \times 10^{-11}$	0,29	1395	0,65
Petreni	Strei	$9,86 \times 10^{-11}$	0,34	1292	0,85
Turda	Aries	$9,99 \times 10^{-11}$	0,29	1280	0,65

¹ It can vary due to the random initialization values of the weights

² It can vary due to the momentary running tasks of the processor

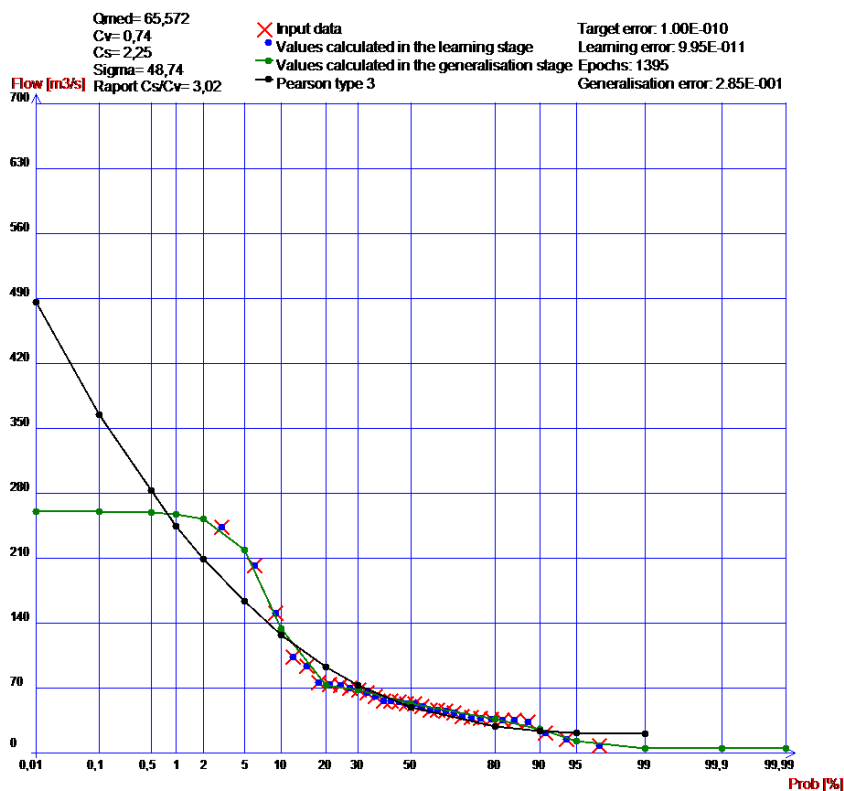


Fig. 5 – Graphical representation of the result for the Barabant gauge station.

The statistical parameters (**Table 2**): Q_{average} (average flow of the series), C_v (coefficient of variation), C_s (coefficient of asymmetry), σ (standard deviation) and C_s/C_v report were calculated in order to plot the Pearson type 3 density probability function on the same graphics (the black dots). Pearson type 3 is often used for the Romanian rivers, because in general, it approximates the river flow well. The calculations were made according to the studies presented by Drobot, R. (1997).

Table 2.

Statistical data of the annual maximum flow series.

Gauge station	River	Q_{average} (m^3/s)	C_s	C_v	C_s/C_v	σ (Sigma)	Number of values
Barabant	Ampoi	65,572	2,25	0,74	3,02	48,74	32
Petreni	Strei	211,61	0,6	0,47	1,27	99,82	40
Turda	Aries	260,724	2,09	0,77	2,73	199,54	34

The comparison between the ANN results and the Pearson type 3 values reveals that there are significant differences for the flow values corresponding to the 0.01% and 0.1% probabilities, and less or not significant for the rest.

5. CONCLUSIONS

The ANN model used in this study proved that it can interpolate very well a series of data, according to the target error. On the present generation of computers, for small number of values in the series, the duration of the calculations is very short.

In this study, the target error value of 10^{-10} was reached each time, without exceeding the limit of the number of epochs. From the physical point of view, the target error has the unit measure of a flow, consequently the value is 10^{-10} m³/s. One can say that the error of the calculation made by the ANN is very low. For the generalization error (GE) the situation is similar, because GE represents less than 1% from the Q_{average} , for all studied gauge stations.

A disadvantage of the ANN model is that for a certain gauge station, if the data series has to be added with at least one more value, the ANN model has to pass the 2 main stages again, because the addition will modify the architecture of the ANN.

The comparison between the ANN results and the Pearson type 3 values reveals that there are significant differences for the flow values corresponding to the 0.01% and 0.1% probabilities, and less or not significant for the rest. This is due to the fact that the ANN has no flow values for the learning stage in those areas of probability. Still, this is the area of interest for hydrologists, because the flow with these probabilities of occurrence are necessary to dimension the constructions (like bridges), according to the national standards. The statistical method gives the possibility to extrapolate in this area of probability.

This study is in an incipient stage. It should be repeated on the main river and on different other tributaries to observe if the conclusions have a general aspect. Another direction of research is to find a way for the ANN to be able to extrapolate in the mentioned area of interest for probabilities (possible by extending the data series with other parameters). Also, a further study should verify if there is an influence from the geographical parameters (e.g. from the elevation).

Acknowledgements

The author would like to thank Prof. Călin Enăchescu, Ph. D., University of Medicine, Pharmacy, Sciences and Technology of Târgu-Mureș, Faculty of Sciences and Letters, for his support for the preparation of the ANN model.

Also, the author would like to thank to the National Institute of Hydrology and Water Management, Bucharest, Romania, for providing the hydrological data for this study.

REFERENCES

- Aichouri, I., Hani, A., Bougherira, N., Djabri, L., Chaffai, H. & Lallahem, S. (2015) River flow model using artificial neural networks, *Energy Procedia*, 74, 1007 – 1014. [Online] Available from: <https://doi.org/10.1016/j.egypro.2015.07.832>
- ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000) Artificial Neural Networks in hydrology: I: preliminary concepts, *Journal of Hydrology Engineering*, 5(2), 115-123. [Online] Available from: <https://pdfs.semanticscholar.org/439b/ca74d7b1240d218d69e6b55539b52a489f57.pdf>
- Dar, L. A. (2017) Rainfall-runoff modeling using artificial neural network technique, *International Research Journal of Engineering and Technology (IRJET)*, 04, 11. [Online] Available from: <https://www.researchgate.net/publication/321706414>
- Dawson, C. W., & Wilby, R. L., (2001) Hydrological modeling using artificial neural networks, *Progress in Physical Geography*, 25, 1, 80-108. [Online] Available from: DOI: 10.1177/030913330102500104
- de Vos, N. J. & Rientjes, T. H. M. (2005) Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation, *Hydrology and Earth System Sciences*, 9, 111–126. [Online] Available from: DOI: 10.5194/hessd-2-365-2005
- Drobot, R. (1997) *Bazele statistice ale hidrologiei*, ISBN 973-30-4832-1, Bucharest, Romania, Ed. Didactică și Pedagogică R.A.
- Dounia, M., Sabri, D., & Yassine, D. (2014) Rainfall-Rainoff Modeling Using Artificial Neural Network, *APCBEE Procedia* 10, 251 – 256. [Online] Available from: <https://doi.org/10.1016/j.apcbee.2014.10.048>
- Enăchescu, C. (1998) *Bazele teoretice ale rețelelor neuronale*, Ed. Casa Cărții de Știință, Cluj-Napoca, Romania
- Enăchescu, C. (2008) Approximation capabilities of neural networks, *JNAIAM - Journal of Numerical Analyses, Industrial and Applied Mathematics*, 3, 3-4, 221-230, [Online] Available from: <https://pdfs.semanticscholar.org/3139/611a145f3319382951fe283c810e8cc7501c.pdf>
- Firuzi, S., Sharifi, M., Borna, K. & Rajabi, M. (2011) Using Empirical Relationships and Neural Network in GIS for Developing Rainfall-Runoff Model, *TS09E - Engineering Surveying - Software*, 5264. [Online] Available from: <https://www.researchgate.net/publication/316154879>
- Kashiwao, T., Nakayama, K., Ando, S., Ikeda, K., Lee, M. & Bahadori, A. (2017) A neural network-based local rainfall prediction system using meteorological data on the Internet: A case study using data from the Japan Meteorological Agency, *Procedia Engineering* 154, 1110 – 1115. [Online] Available from: <http://dx.doi.org/10.1016/j.asoc.2017.03.015>
- Lallahem, S. & Mania, J. (2003) A Nonlinear Rainfall-Runoff Model using Neural Network Technique: Example in Fractured Porous Media, *Mathematical and Computer Modeling*, 37, 9-10, 1047-1061. [Online] Available from: [https://doi.org/10.1016/S0895-7177\(03\)00117-1](https://doi.org/10.1016/S0895-7177(03)00117-1)
- McCaffrey, J. (2014) *Neural Networks Using C# - Succinctly*, 2501 Aerial Center Parkway, Morrisville, NC 27560. [Online] Available from: www.syncfusion.com
- Sarkar, A. & Kumar, R. (2012) Artificial Neural Networks for Event Based Rainfall-Runoff Modeling, *Journal of Water Resource and Protection*, 4, 891-897. [Online] Available from: <http://dx.doi.org/10.4236/jwarp.2012.410105>
- Tanty, R. & Dr. Desmukh, T. S. (2015) Application of Artificial Neural Network in Hydrology-A Review, *International Journal of Engineering Research & Technology (IJERT)*, 2278-0181, 4, 6. [Online] Available from: DOI: 10.17577/IJERTV4IS060247