**Mathematical Modelling in Civil Engineering**
Vol. 18-No. 3: 1-17-2023
10.2478/mmce-2023-0011

# THE USE OF RECURRENT NEURAL NETWORKS (S-RNN, LSTM, GRU) FOR FLOOD FORECASTING BASED ON DATA EXTRACTED FROM CLASSICAL HYDRAULIC MODELING

**Andrei Mihai RUGINĂ** - Doctoral School, Technical University of Civil Engineering of Bucharest, S.C. AQUAPROJECT S.A., e-mail: andreimihairugina@gmail.com, andrei-mihai.rugina@phd.utcb.ro

**Abstract:** Floods are natural disasters that have a significant impact on everyday human life, both through material losses and loss of life. In the context of climate change, these events may be more frequent or more dangerous. For real-time flood forecasting, fast methods for determining flood hydrographs along watercourses are needed. Classic hydraulic modeling software provides satisfactory results, but in many cases the calculation time can be high. Another approach, different from classical hydraulic modeling is the use of neural networks for forecasting hydrographs. Thus, the present study aims to analyze three different types of recurrent neural networks, including S-RNN, RNN-LSTM, RNN-GRU. For each network type, flow hydrographs and level hydrographs resulting from hydraulic modeling were provided as input and training data. Using the deep learning environment, based on previous calibration and validation of recurrent neural networks, flood hydrographs for 2 historical events were modeled. The obtained hydrographs are extremely close to those recorded, while the running time is tens of times smaller.

**Keywords:** deep learning; neuronal networks; synthetic hydrographs; hydraulic modeling; forecasting hydrographs

**Abbreviations**

1D – one-dimensional modeling; 2D - two-dimensional modeling; AEP - annual exceedance probabilities; ANN – artificial neuronal network; DTM – digital terrain model; GRU – gated recurrent unit; LSTM – long short-term memory; NN – neuronal networks; Q – flow; RNN – recurrent neuronal network; WL – water level.

## 1. Introduction

Hydrometeorological events such as storms [1], landslides [2] or floods [3], [4] are associated with a high degree of uncertainty, thus ranking them at the top when it comes to global natural disasters [5].

Taking into account the acceleration of climate change [6] in recent decades, precipitation has begun to exceed the historical maximums recorded, having serious consequences both in terms of material losses, but more importantly loss of life [7].

Given the direct impact that high rainfall has on the environment, potentially leading to significant flooding [8], it is important to get an overview of watersheds, especially in known areas with a high degree of danger. This can be done at the moment with the help of classic hydraulic modeling, through well-known modeling software such as HEC-Ras [9], Mike Flood [10], TUFLOW [11] or Iber [12]. However, new methods to improve these calculation programs are needed.

Maximum affected area [13] is an important feature that classic modeling software offers. In addition to this feature, hydraulic models can also provide information on water velocity, water level, maximum flow, or the shape of the flood hydrograph [14]. These results help to understand the behavior of a watershed of different sizes and to identify areas with a high degree of risk.

Given that in many situations classical hydraulic models have a high computation time it is necessary to find new, much faster methods that provide results close, if not identical, to those obtained in the classical way.

In the last 20 years, but with a much higher weight in the last 5 years, many new artificial intelligence tools have been developed and successfully used in hydrology [15].

Deep learning [16] is part of the big family of artificial intelligence [17], being a class of machine learning algorithms [18]. Deep learning uses multiple layers to progressively extract higher-level features from raw inputs, a process called artificial neural network (ANN) [19]. Due to their flexible structure and ability to approximate nonlinear relationships, ANNs are a useful tool in hydrology.

The present study is concerted on the development of models based on several recurrent neural network (RNN) architectures [20] for forecasting flood waves at a certain point in a river. A section of the Teleorman River, Romania, was chosen for the study. Considering that RNNs are capable of approximating complex relationships between input-output data, the main objective of the study was to develop more RNNs capable of mimicking the classical hydraulic model.

The RNN architectures used in the present study were RNN-classical, RNN-LSTM (long short-term memory) and RNN-GRU (gated recurrent unit). Both flood hydrographs and level hydrographs were used as input-output data for these RNN architectures. After training the neural networks based on the input data, the ability of the neural networks to forecast a new flow hydrograph and the new water level hydrograph was tested, after which the results were compared with the results obtained from classical modeling with the HEC-Ras software for validation.

## 2. Materials and Methods

The diagram below shows the workflow behind the study.

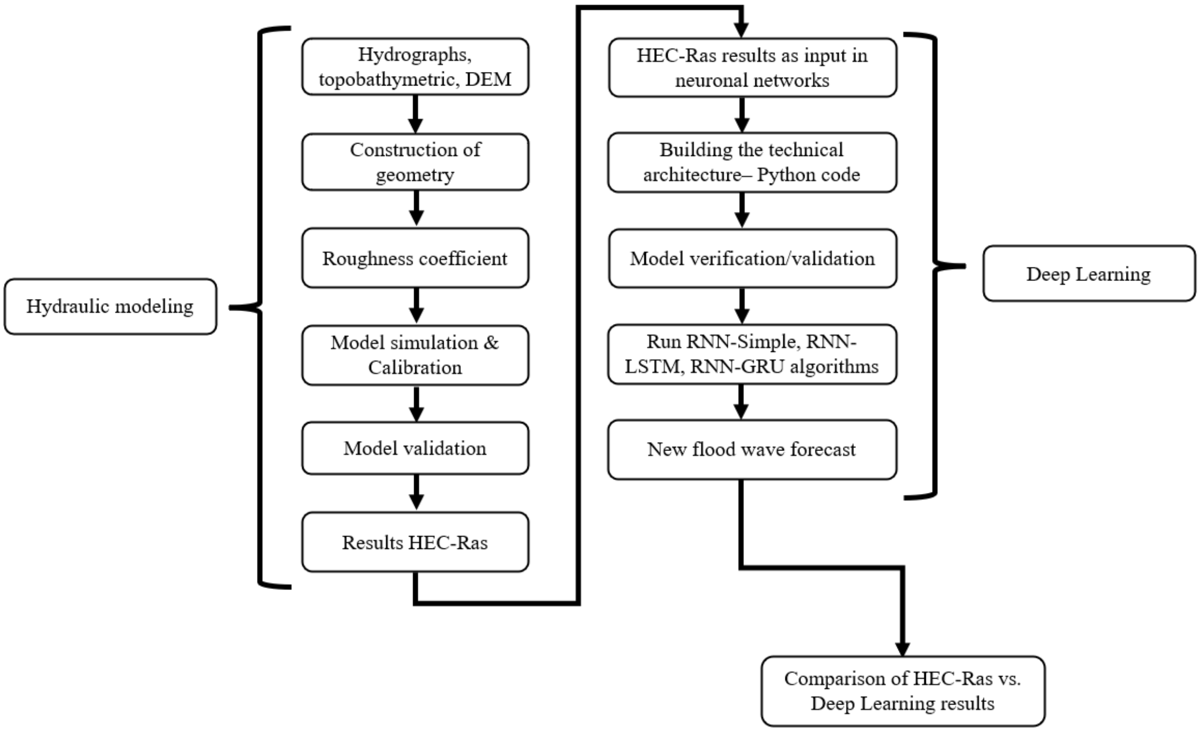


Fig. 1 - Workflow diagram, main stage achieved

### 2.1 Hydraulic model

Hydraulic calculations for the present study were conducted using HEC-Ras software (version 6.5) (Hydraulic Engineering Center's – River Analysis System) developed by the US Army Corps of Engineers. This software allows the user to perform one-dimensional (1D) steady flow, one and two-dimensional (2D) unsteady flow calculations, sediment transport/mobile bed computations, and water temperature/water quality modeling.

The hydraulic calculations were carried out in a full 2D geometry which is based on solving the 2D Saint Venant equations [21], or the 2D diffusive wave equations:

$$\frac{\partial \xi}{\partial t} + \frac{\partial \rho}{\partial x} + \frac{\partial q}{\partial u} = 0 \tag{1}$$

$$\frac{\partial p}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\rho^2}{h}\right) + \frac{\partial}{\partial y}\left(\frac{\rho q}{h}\right) = -\frac{n^2 \rho g \sqrt{\rho^2+q^2}}{h^2} - gh\frac{\partial \xi}{\partial x} + pf + \frac{\partial}{\rho \partial x}(h\tau_{xx}) + \frac{\partial}{\rho \partial y}(h\tau_{xy}) \tag{2}$$

$$\frac{\partial p}{\partial t} + \frac{\partial}{\partial y}\left(\frac{q^2}{h}\right) + \frac{\partial}{\partial x}\left(\frac{\rho q}{h}\right) = -\frac{n^2 q g \sqrt{\rho^2 + q^2}}{h^2} - gh\frac{\partial \xi}{\partial y} + qf + \frac{\partial}{\rho \partial y}(h\tau_{yy}) + \frac{\partial}{\rho \partial x}(h\tau_{xy}) \tag{3}$$

where $h$ is the water depth (m), $p$ and $q$ are the specific flow in the $x$ and $y$ directions ($m^2 s^{-1}$), $\xi$ is the surface elevation (m), g is the accelaration due to gravity ($m^2/s$), $n$ is the Manning resistance, $\rho$ is the wather density ($kgm^{-3}$).

## 2.2. Recurrent neuronal networks

A recurrent neural network (RNN) is a class of artificial neural networks that deep learning can use. This type of network uses sequential or time series data [22] and is often used primarily for ordinal or temporal problems.

The main advantage lies in the ability of the recurrent neural network to work with sequential data and maintain some form of temporal memory, while artificial neural networks are designed to process fixed data without keeping a specific memory for the order of the data. However, RNNs have certain challenges, such as the gradient vanishing problem, which has led to the development of other architectures such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) to improve the ability to retain information in the long term.

### *2.2.1 Simple recurrent neuronal networks*

Simple RNN is an ANN that has three layers, input layer $(x_1, x_2, \dots, x_i, x_{i+1})$, hidden layer $(h_1, h_2, \dots, h_i, h_{i+1})$ and output layer $(h_1, h_2, \dots, h_i, h_{i+1})$ [23]. Simple RNN is the oldest and simple form of RNNs.

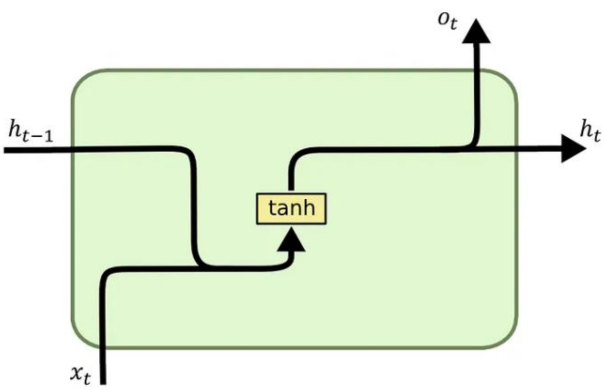


Fig. 2 - Internal architecture of a simple RNN

According to Eq. (4) at each time step, the input vector $x_t$ is combined with the previous time step's state vector.

$$h_t = \tanh(Wx_t + Uh_{t-1} + b) \tag{4}$$

Where $tanh$ is activation function, $W$ - weight at input neuron, $x_t$ - input stats, $U$ – weight at recurrent neuron, $h_{t-1}$ – previous state and $b$ – the bias associated with the recurrent layer.

### *2.2.2. Long short-term memory*

Long-Short Term Memory Networks (LSTM networks) are special RNN networks, which can learn long-term dependencies between variables. They were introduced by Hochreiter and Schmidhuber in 1997 and have been popularized by many scientific papers since then [24].

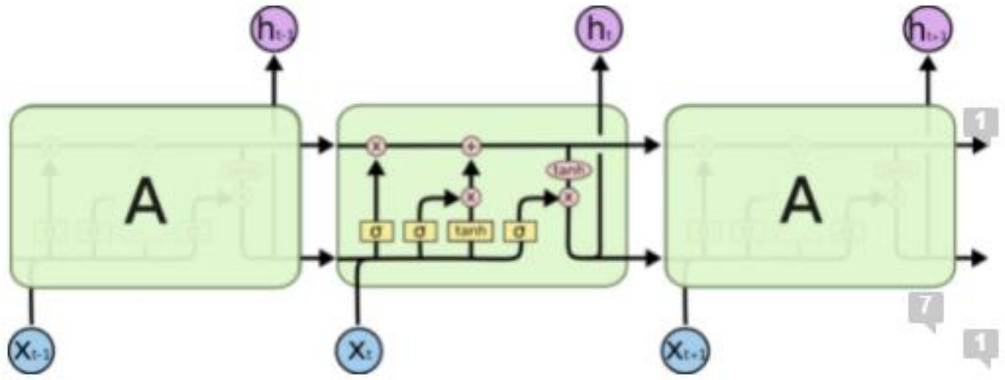


Fig. 3 - Architecture of an RNN-LSTM

The basic idea behind the LSTM architecture is the memory unit that preserves its state over time and stores useful information for a long period of time. It then uses past information and some gates containing save, write and read functions to control the flow of information in a network to select the information that enters the cell. A memory unit is composed of input, output and forget gates and memory cells. The gates of entry, exit, and oblivion are the keys to controlling the flow of information. The underlying equations of RNN-LSTM are as follows:

$$f_t = \sigma\ (W^f x_t + U^f h_{t-1} + b^f) \tag{5}$$

$$i_t = \sigma\ (W^i x_t + U^i h_{t-1} + b^i) \tag{6}$$

$$o_t = \sigma\ (W^o x_t + U^o h_{t-1} + b^o) \tag{7}$$

$$\tilde{c_t} = tanh\ (W^c x_t + U^c h_{t-1} + b^c) \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{9}$$

$$h_t = \ o_t \odot \tanh(c_t) \tag{10}$$

Where $\sigma$ is activation function, $W\ and\ U$ – weight matrices, $b$ – bias vector parameters, $x_t$ – input vector, $h_{t-1}$ – hidden state vector (also known as output vector of the LSTM unit), $i_t$ – input/update gate activation vector, $o_t$ – output gate activation vector, $\tilde{c_t}$ – cell input activation vector, $c_t$ – cell state vector, $f_t$ – forget gate activation vector and ⊙ - Hadamard Product.

### 2.2.3 Gated recurrent unit

We can say about Gated Recurrent Unit (GRU) that it is the younger brother of the more famous LSTM network, and it is a time of RNN. Like LSTMs, GRUs can efficiently retain long-term dependencies in the form of sequential data. And furthermore, they can address the "short-term memory" problem that plagues RNNs. This type of network was brought into discussion in 2014 [25]. GRU given the architecture inherited from RNN and the new approach is on its way to outshine its big brother due to its superior speed while achieving similar accuracy and effectiveness. The GRU uses the gating mechanism to control the flow of information between cells within the neural network.

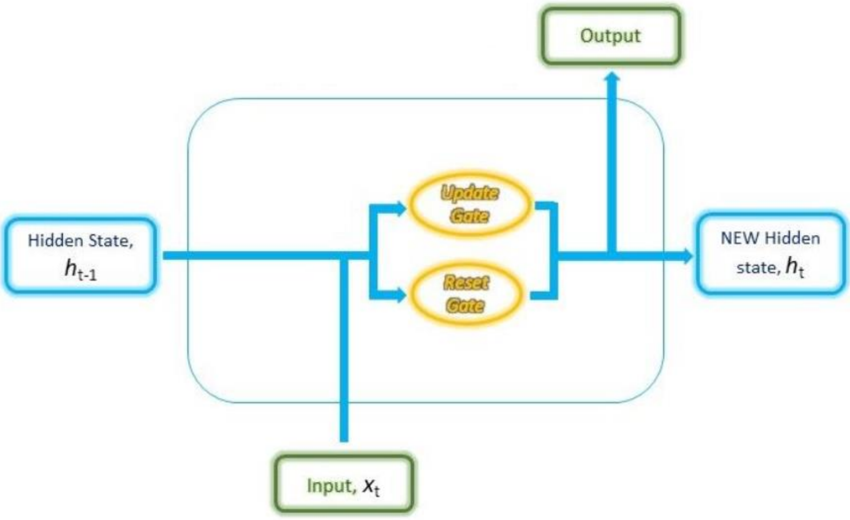


Fig. 4 - Architecture of an RNN-GRU

The underlying equations of RNN-GRU are as follows:

$$z_t = \sigma\,(W^z x_t + U^z h_{t-1} + b^z) \tag{11}$$

$$r_t = \sigma\,(W^r x_t + U^r h_{t-1} + b^r) \tag{12}$$

$$u_t = tanh(W^u x_t + U^u (r_t \odot h_{t-1}) + b) \tag{13}$$

$$h_t = z_t \odot h_{t-1} + (1 - \mathrm{z}) \odot u_t \tag{14}$$

Where $\sigma$ is activation function, $x_t$ – input vector, $h_{t-1}$ – output vector, $u_t$ – candidate activation vector, $z_t$ – update gate vector, $r_t$ – rest gate vector, $W$ and $U$ – parameter matrices, $b$ – vector which need to be learned during training and $\odot$ - Hadamard Product.

### 2.2.4 Performance indicators used by the experiments

*Mean Squared Error (MSE)*

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{15}$$

Where $n$ – total number of observation examples in the test data set; $\mathrm{y_i}$ – real value of observation i: $\widehat{y_\mathrm{i}}$ – predicted value for observation i.

*Root Mean Squared Error (RMSE)*

$$RMSE = \sqrt{MSE} \tag{16}$$

*Nash–Sutcliffe model efficiency (NSE)*

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{17}$$

Where n – total number of observation examples in the test data set; $y_i$ – is the observed value for the data point i; $\widehat{y_i}$ – predicted value for observation i; $\bar{y}$ – average of the observation values.

### 2.2.5 Neuronal networks environment

Python is a high-level programming language and one of the most popular programming languages, especially in the machine learning community [26]. Python version 3.11 was used to develop the neural networks presented in this study, along with the PyCharm Community Edition integrated development environment. Additionally, the libraries TensorFlow, Keras, Pandas, NumPy, and Matplotlib were utilized. All of these are free and open-source platforms.

TensorFlow is one of the popular libraries for machine learning [27], used for a wide range of tasks but especially focused on training and inferring deep neural networks. Complementing the TensorFlow environment is the Keras library [28], which provides the Python interface for artificial neural networks. Keras contains numerous implementations of commonly used neural network building blocks, including layers, objectives, activation functions, and optimizers. In addition to standard networks, Keras also supports convolutional and recurrent neural networks and includes other commonly used layers such as dropout, batch normalization, and pooling.,

The Pandas library was used for data manipulation [29], being a powerful, flexible, and easy-to-use tool with a high capacity for handling large volumes of data, providing fast data structures.

NumPy is considered the fundamental package for scientific computing with Python [30], being the most powerful tool that enables working with high-performance multidimensional arrays and provides the necessary tools for manipulating these arrays. Graphical visualization was accomplished using Matplotlib, a library used for creating plots and graphs.

## 3. Case Study

The Teleorman River (cadastral code: IX.1.15) is a river in the Argeș-Vedea hydrographic basin, being a left tributary of the Vedea River, having its point of confluence near the town of Beiu [31].

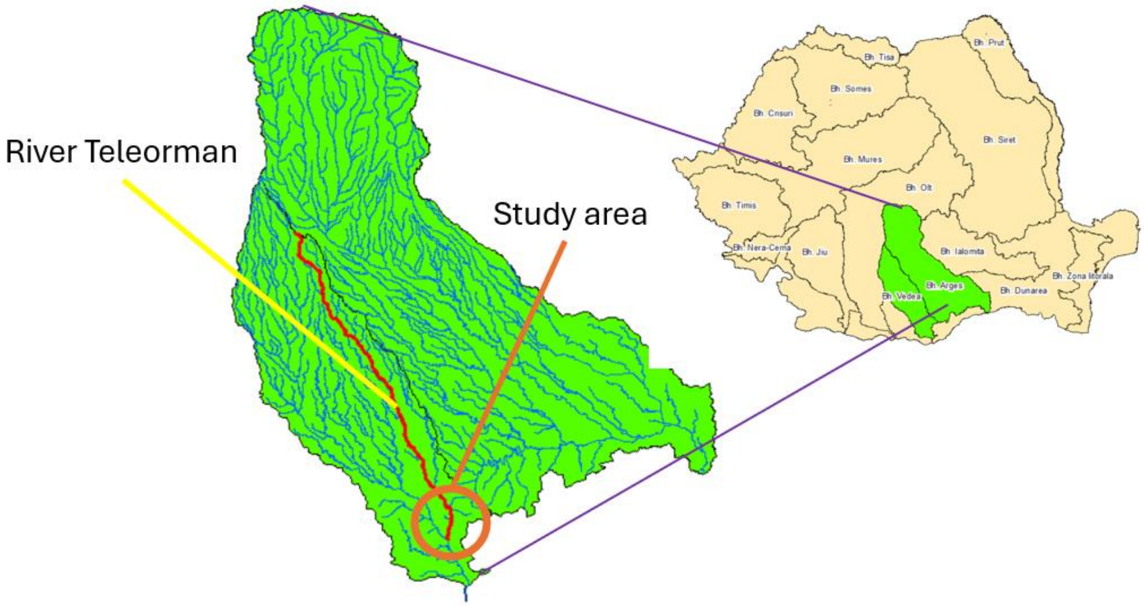


Fig. 5 - Presentation of the study area - geographical positioning of river Teleorman

The case study was conducted on the Teleorman River, over a length of approximately 9.5 km. The starting point of the case study is near the town of Valea Pârâului and extends to the confluence area of the Teleorman and Vedea Rivers.

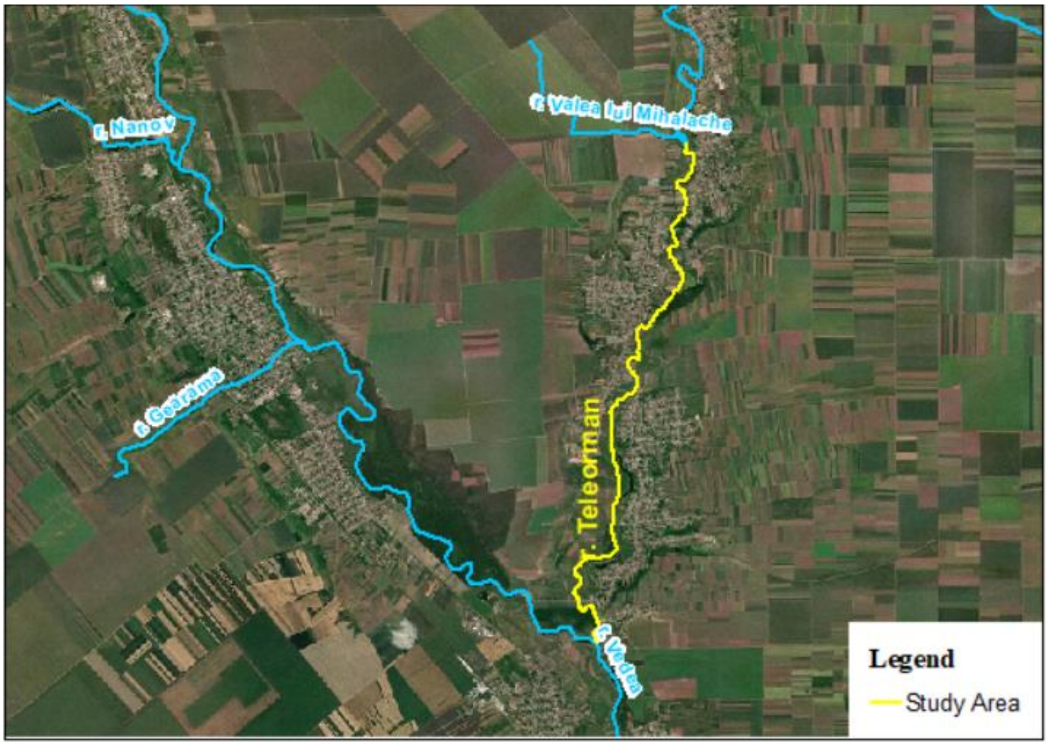


Fig. 6 - Study area – Teleorman River

## 3.1. Data available

### 3.1.1 Digital terrain model

The digital terrain model (DTM) [32] used in the present study has a high resolution of 0.5 m x 0.5 m over its entire surface.

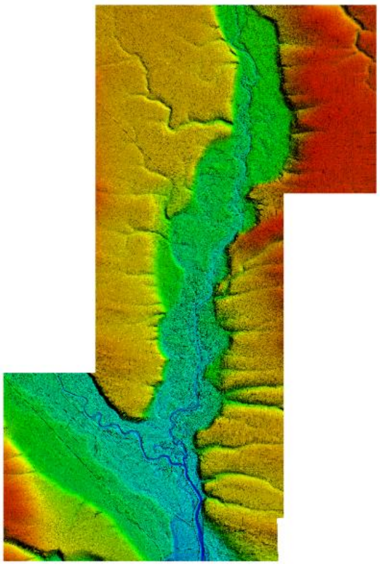

Fig. 7 - Study area – digital terrain model

### 3.1.2 Hydrological data

Hydrological and historical data were calculated by INHGA and made available by S.C. Aquaproiect S.A to carry out the study.

In the present study, 9 annual exceedance probabilities (AEP) were used. The constructional characteristics (total time ($T_t$), rise time ($T_{cr}$) and shape coefficient (γ)) of these flood waves are shown in the table below.

*Table 1*

**Characteristics of the flood wave – analyzed sector**

| Node | Q 0.1% | Q 0.2% | Q 0.5% | Q 1%CC | Q 1% | Q 2% | Q 5% | Q 10% | Q 33% | $T_t$ | $T_{cr}$ | γ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | mc/s | mc/s | mc/s | mc/s | mc/s | mc/s | mc/s | mc/s | mc/s | h | h | - |
| **Upstream** | 635 | 559 | 450 | 425.5 | 370 | 307 | 204 | 160 | 85 | 116 | 29 | 0.34 |
| **Downstream** | 635 | 559 | 450 | 425.5 | 370 | 307 | 204 | 160 | 85 | 121 | 30 | 0.34 |

The historic floods present in this work were recorded in 2014 and 2018, their form is shown in the following images.

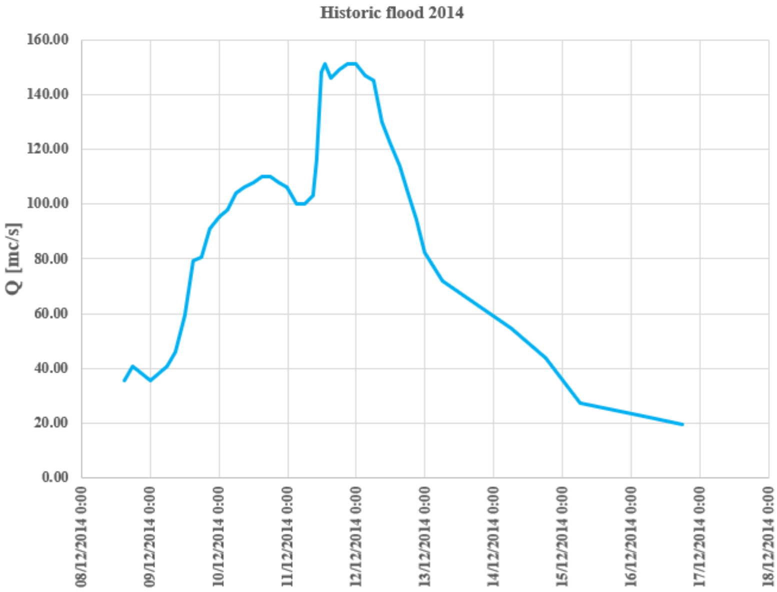


Fig. 8 - Historic flood – 2014

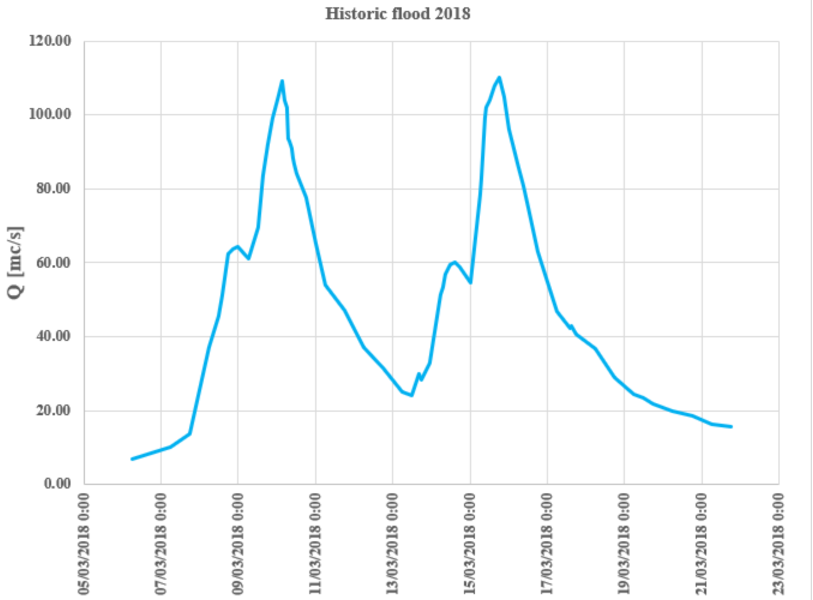


Fig. 9 - Historic flood - 2018

The hydrological hypothesis adopted was to maintain the established probability along the river, introducing a uniform lateral flow along the length of the analyzed sector, which is explained in the diagram below – topological scheme of river Teleorman – study area.

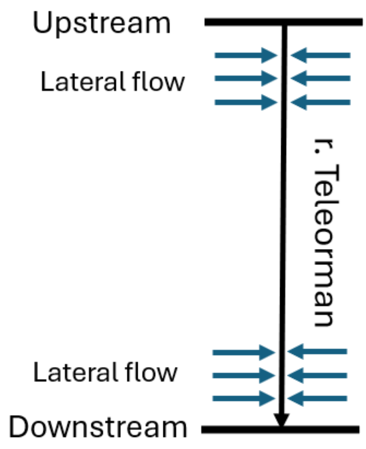


Fig. 10 - Topological scheme of Teleorman River – study area

### 3.1.3 Manning's roughness coefficient

The hydraulic model is based on the implemented roughness coefficient in function of the land cover. The coefficients were analyzed and extracted according to the specialized literature [33], [34], [35] and confirmed with of orthophotos. This is set out in detail in the table below and the accompanying figure. This was made possible thanks to the free land cover images provided by Copernicus, Corine Land Cover 2018.

*Table 2*

**Manning's roughness coefficient**

| Nr. | Land use | Manning's roughness coefficient |
|---|---|---|
| 1 | Discontinuous urban space and rural space | 0.07 |
| 2 | Land at the edge of the towns | 0.05 |
| 3 | Industrial or commercial units | 0.23 |
| 4 | Road network - STR&A | 0.03 |
| 5 | Road network - DC | 0.03 |
| 6 | Road network - DJ | 0.03 |
| 7 | Road network - DN | 0.03 |
| 8 | Non-irrigated arable land | 0.042 |
| 9 | Are you coming | 0.07 |
| 10 | Secondary pastures | 0.04 |
| 11 | Areas of complex cultures | 0.05 |
| 12 | Predominantly agricultural lands mixed with natural vegetation | 0.052 |
| 13 | Deciduous forests | 0.1 |
| 14 | Transition areas with shrubs (generally deforested) | 0.068 |
| 15 | Water courses | 0.033 |
| 16 | Water accumulations | 0.033 |
| 17 | Houses | 0.3 |
| 18 | Riverbed | 0.033 |

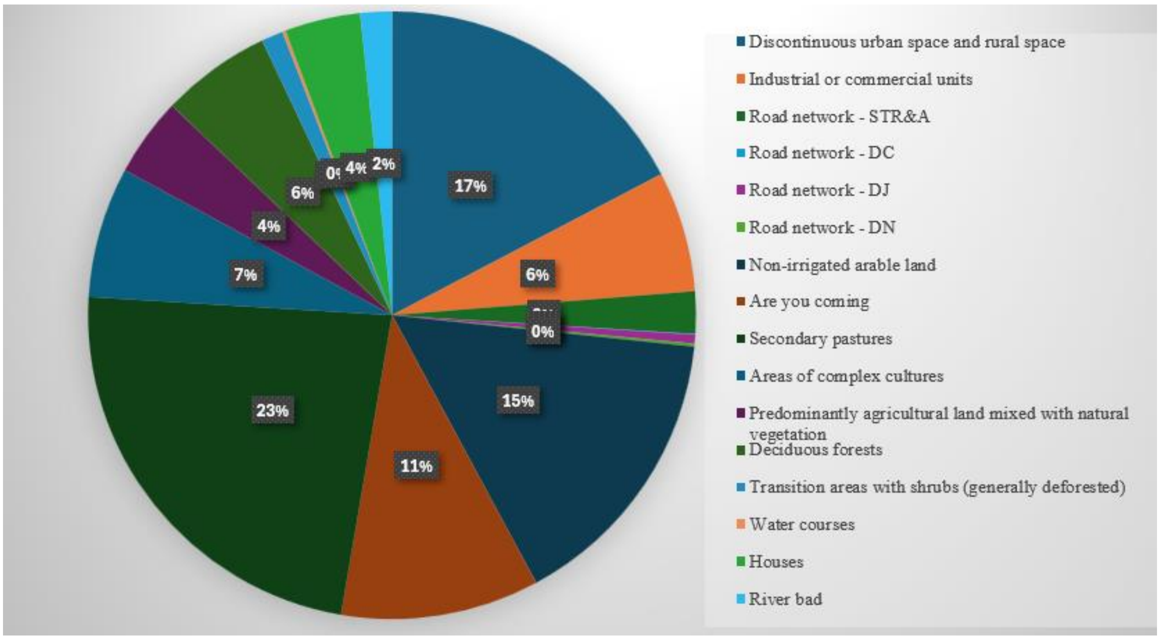


Figure 11. Land use for the study area – Teleorman river

## 3.2 Hydraulic model

For the hydraulic analysis, a mathematical model was used with the HEC-Ras software, version 6.5. The approach taken was a full 2D one. The hydraulic model has the following characteristics: 137328 calculation cells. 2D grid resolution: 20 m x 20 m in flood plain, 5 m x 5 m – riverbed, 5 m x 5 m or 10 m x 10 m - brakeline. 5 m x 5 m - dikes or other elements that influence the flow of water.

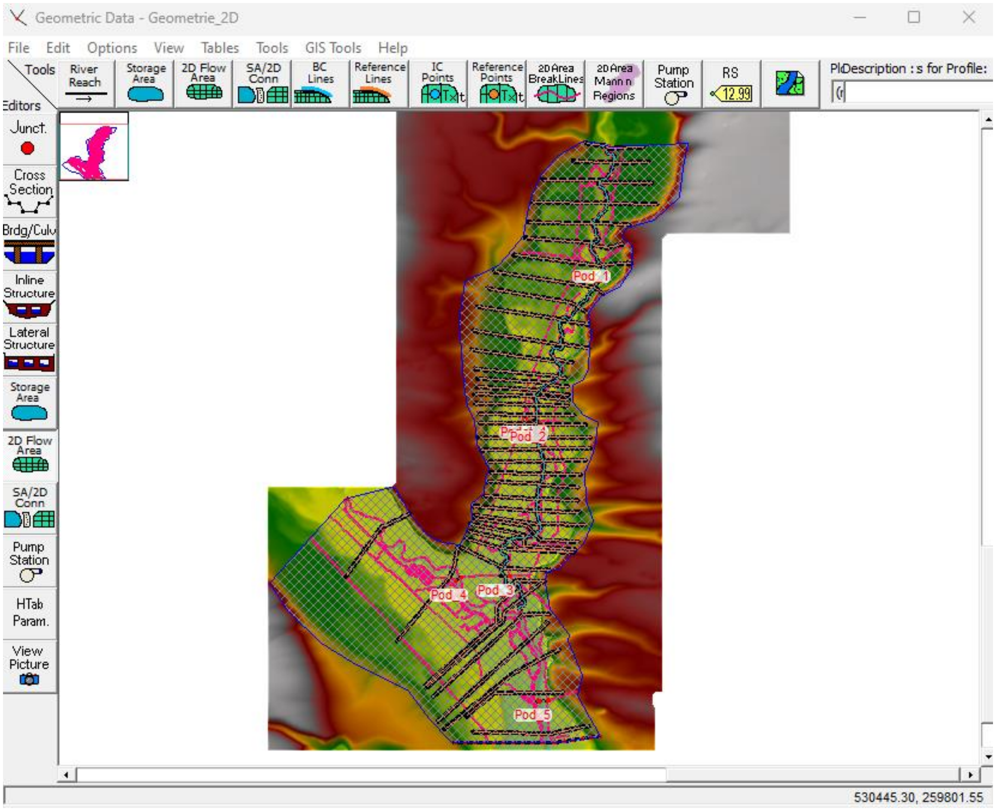


Fig.12 - Schematization of the hydraulic model and the calculation area - river Teleorman HEC-Ras – version 6.5

As boundary conditions, the flow hydrographs presented in the previous chapter were used for the upstream area, and the normal slope of the land was used in the outlet area of the model.

*Table 3*

**Centralizing table regarding the stability of the hydraulic model**

| AEPs | Volume error | Volume error | Time of simulation |
|---|---|---|---|
| | [error in 1000 m³] | [%] | [hh:mm:ss] |
| **0.10%** | 1.386 | 0.0008 | 9:02:48 |
| **0.20%** | 1.315 | 0.0008 | 6:37:24 |
| **0.50%** | 1.098 | 0.0007 | 5:43:19 |
| **1%CC** | 1.026 | 0.0007 | 5:58:54 |
| **1%** | 0.762 | 0.0005 | 5:27:01 |
| **2%** | 0.543 | 0.0004 | 5:16:15 |
| **5%** | 0.453 | 0.0004 | 4:25:05 |
| **10%** | 0.422 | 0.0004 | 4:17:39 |
| **33%** | 0.332 | 0.0003 | 5:41:24 |
| **Historic flood 2014** | 0.368 | 0.0002 | 6:25:22 |
| **Historic flood 2018** | 0.341 | 0.0001 | 14:58:08 |

### 3.3 Forecasting a new flood wave using recurrent neural networks

The present study aims to analyze the way of forecasting the new flood wave starting from a set of known data.

To achieve this, in the first step the hydraulic model was simulated for the 9 APEs and for the two historical floods. After completing the hydraulic calculations along the river, a series of results (discharge and water level for all 9 APEs and the 2 historical floods) were extracted at a point downstream of the model. The point of extracting the results was specially chosen near the city of Beiu because this is the most vulnerable area.

In the second step, the results obtained after the hydraulic modeling for the 9 AEPs were given to the neural networks as training data and the 2 historical floods were used to test and verify the forecasting ability of these networks.

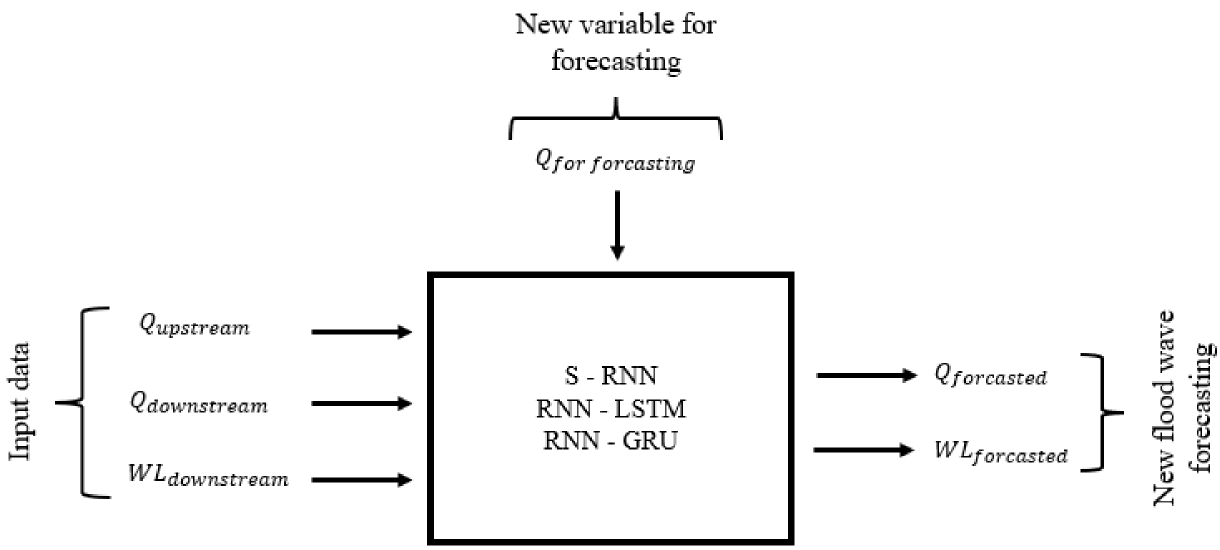


Fig. 13 - Block diagram in determining a new flood wave

where:

$Q_{upstream}$ – input flow for NN – hydraulic model upstream point;

$Q_{downstream}$ – input flow for NN – hydraulic model downstream point;

$WL_{downstream}$ – input water level for NN – hydraulic model downstream point;

$Q_{for\ forcasting}$ – new flood wave to forecast by NN;

$Q_{forcasted}$ – flood wave predicted by NN;

$WL_{forcasted}$ – Water level forecast by NN;

Three types of recurrent neural networks were chosen for the present analysis in the present study, they are S-RNN, RNN-LSTM, RNN-GRU. The present study aims to analyze the forecasting ability of the three networks and compare the results obtained. To achieve this the hidden layer of each network had 2000 neurons and 500 computation epochs were used.

The input data was split into 80% training data and 20% test data within the networks.

## 4. Results

To identify the performance of the studied neural networks, the following performance indicators RMSE (Root Mean Square Deviation), NSE (Nash-Sutcliffe Efficiency) and MSE (Mean Squared Error) were chosen. The obtained results have been sintered in the table below for each individual iteration.

*Table 4*

**Performance indicators – Loss, NSE, MSE and $R^2$**

| | RMSE | MSE | NSE |
|---|---|---|---|
| **RNN-Simple** | 0.995 | 0.99 | 0.752 |
| **RNN-LSTM** | 0.999 | 0.999 | 0.751 |
| **RNN-GRU** | 1.039 | 1.08 | 0.752 |

worst value

best value

*Table 5*

**Simulation time – HEC-Ras vs. RNN-Simple vs. RNN-LSTM vs. RNN-GRU**

| | Time of simulation [hh:mm:ss] |
|---|---|
| **RNN-Simple** | 0:10:30 |
| **RNN-LSTM** | 0:33:03 |
| **RNN-GRU** | 0:24:29 |
| **Historic flood 2014** | 6:25:22 |
| **Historic flood 2018** | 14:58:08 |

After completing the training of the recurrent neural networks and checking the performance indices, the forecasting ability of a historical flood was studied for each network.

In the first stage, starting from the upstream point of the hydraulic model, the historical floods were propagated along the river. At the downstream point, chosen as the analysis point, the flow and level results were extracted throughout the duration of the flood, these playing the role of the reference results.

In the second stage, the same flow hydrograph was provided to the deep learning algorithms, to see their ability to forecast the wave at the desired point.

For the first historic flood wave, the results are as follows:

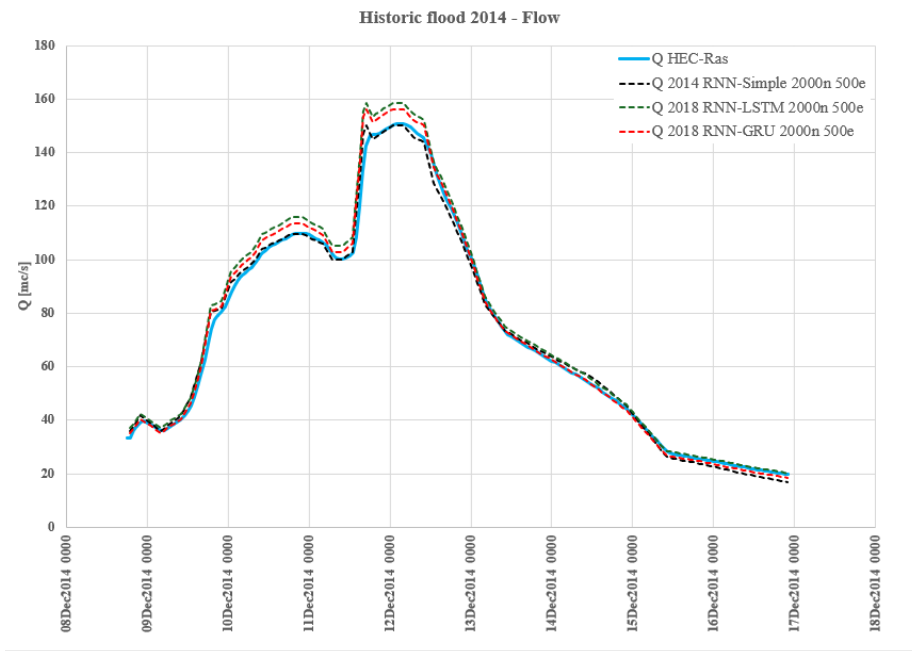


Fig. 14 - Flow hydrographs - HEC-Ras vs. RNN-Simple vs. RNN-LSTM vs. RNN-GRU – historic flood 2014

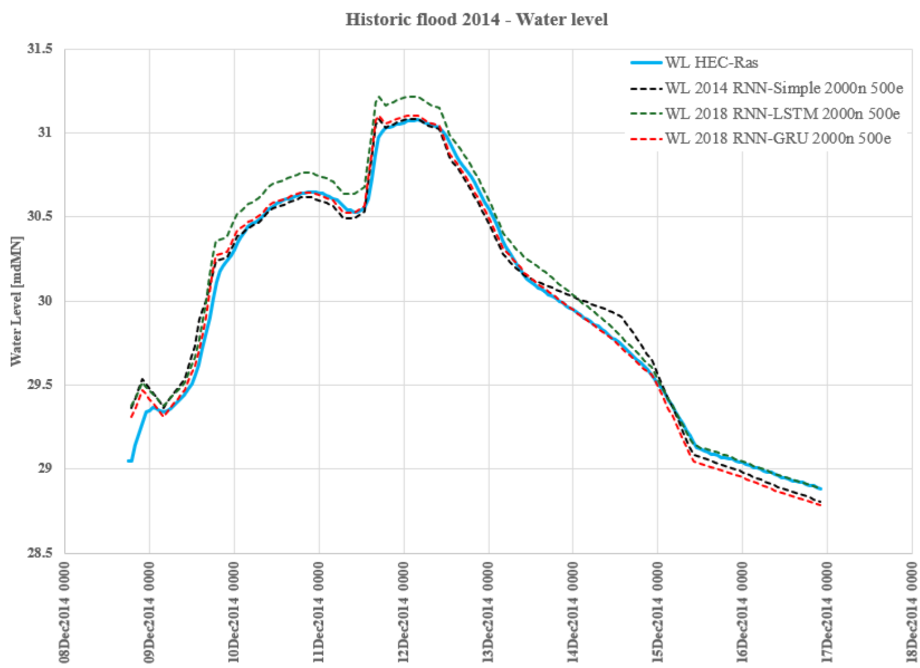


Fig. 15 - Water level hydrographs - HEC-Ras vs. RNN-Simple vs. RNN-LSTM vs. RNN-GRU – historic flood 2014

*Table 6*

**Tabulated results for maximum discharge, flood wave volume and maximum water level historical flood 2014**

| | HEC-Ras | 2014-RNN-Simple 2000n 500e | 2014-RNN-LSTM 2000n 500e | 2014-RNN-GRU 2000n 500e |
|---|---|---|---|---|
| **Q max [mc/s]** | 150.9 | 150.03 | 158.49 | 156.26 |
| **Vol [mil.mc]** | 51.44 | 51.37 | 53.75 | 52.36 |
| **WL max [mdMN]** | 31.08 | 31.08 | 31.21 | 31.1 |

reference value

best value

For the historical flood of 2014, analyzing the table of results as well as the figures of the flow and level hydrographs, we notice that all the networks have the ability to forecast and preserve the shape of the flood wave, but the best results are obtained by the RNN-Simple 2000n 500e network. Regardless of the analyzed parameter, whether it is the maximum flow, the flood volume or the maximum water level, the network provides the best results, which are very close to the reference results.

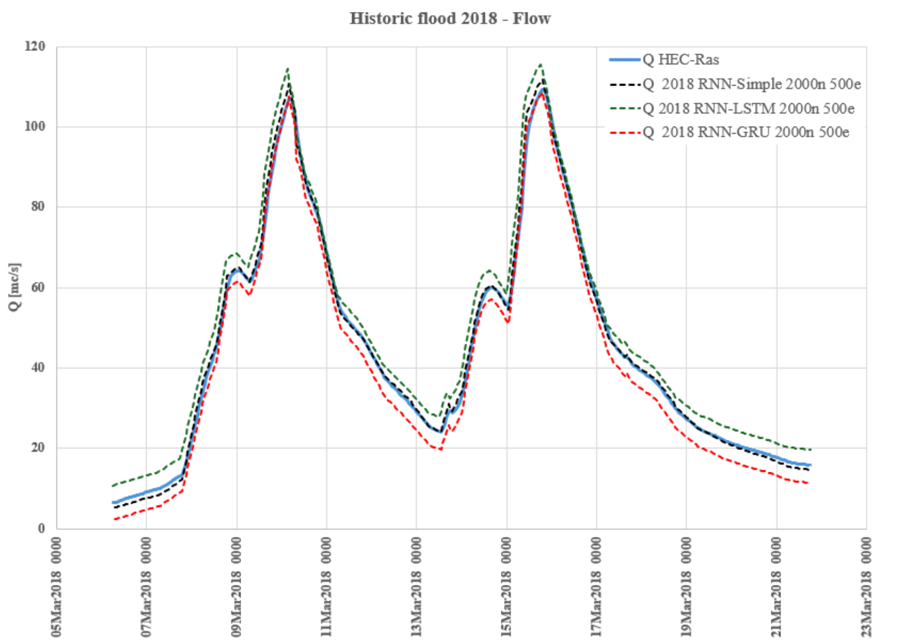


Fig. 16 - Flow hydrographs - HEC-Ras vs. RNN-Simple vs. RNN-LSTM vs. RNN-GRU – historic flood 2018

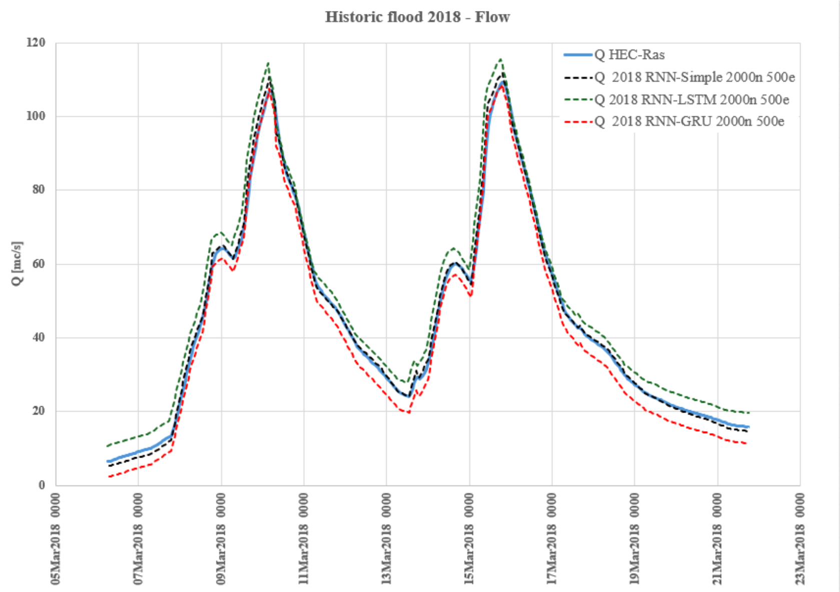


Fig. 17 - Water level hydrographs - HEC-Ras vs. RNN-Simple vs. RNN-LSTM vs. RNN-GRU – historic flood 2018

*Table 7*

**Tabulated results for maximum discharge, flood wave volume and maximum water level - historical flood 2018**

| | HEC-Ras | 2018-RNN-Simple 2000n 500e | 2018-RNN-LSTM 2000n 500e | 2018-RNN-GRU 2000n 500e |
|---|---|---|---|---|
| **Q max [mc/s]** | 109.21 | 111.73 | 115.5 | 108.51 |
| **Vol [mil.mc]** | 60.39 | 60.86 | 65.93 | 55.58 |
| **WL max [mdMN]** | 30.64 | 30.65 | 30.75 | 30.62 |

reference value

best value

The historical flood results from 2018 indicate that for this iteration the RNN-Simple 2000n 500e no longer provides the best results for all parameters, managing to keep only the flood wave volume close to the reference value. Analyzing the other parameters, we notice that in terms of maximum flow or maximum water level, the best results are given by RNN-GRU-2000n 500.

## Conclusions

Looking at the results presented in the previous chapter for the analyzed case studies, the recurrent neural networks used (S-RNN, RNN-LSTM or RNN-GRU) show good results in terms of forecasting a new discharge or level hydrograph.

In the first case study - the historical flood of 2014, the best results are provided by the RNN-Simple 2000n 500e network configuration, regardless of whether it is the maximum flow, the volume of the flood wave, the shape of the flood wave or the maximum water level.

In the second case study - the historical flood of the year 2018, looking at each parameter analyzed separately, the RNN-GRU 2000n 500e network gives the best results for the maximum flow and the maximum level, but when it comes to the volume of the flood wave, the difference is considerable compared to the value of reference (4.81 million cubic meters). The best volume value is also provided by the RNN-Simple 2000n 500e network.

Looking globally at the second iteration, considering all parameters analyzed at the same time, we notice that the network with the best result is RNN-Simple 2000n 500e, with all values close to the reference values.

In the present case, the configuration that gives the best results in both scenarios is RNN-Simple 2000n 500e.

Looking at the simulation time, for the neural network algorithms a relatively short calculation time of the order of a few dozen minutes was required, while for the classical modeling several hours were required. Thus, for example, for RNN-Simple 2000n 500e the simulation time was about 10 minutes, while the classic hydraulic simulation lasted 6 hours 25 minutes for the 2014 flood and almost 15 hours for the 2018 flood.

Limitations of the networks are given by the lack of data, or the lack of accurate data. The present study provides good results because the volume of data provided for training the neural networks was high and their quality very precise.

The results obtained in this study are encouraging for the use of recurrent neural networks in the attempt to determine the flood wave in a relatively short time.

The calculations performed here do not replace the importance of classic hydraulic calculations (using modeling software) but can provide information close to them.

***Author Contributions***

Conceptualization, A.M.R.; methodology, A.M.R..; software, A.M.R.; validation, A.M.R.; formal analysis, A.M.R.; writing - original draft preparation, A.M.R.; writing - review and editing, A.M.R.; visualization, A.M.R.; All authors have read and agreed to the published version of the manuscript.

***Funding***

The elaboration of this methodology was not financed by public funds.

***Data Availability Statement***

Data owner is the S.C. AQUAPROJECT S.A., Romania (https://www.aquaproiect.ro/) Restrictions apply to the availability of these data according to the AQUAPROJECT policy.

***Acknowledgments***

Thanks to AQUAPROJECT for the availability and provision of all the materials needed to carry out the present study. As well as thanks to the anonymous reviewers for their help and suggestions.

***Conflicts of Interest***

The authors declare no conflict of interest.

## References

[1] Peleg, N. et al., 2022. Mapping storm spatial profiles for flood impact assessments. Advances in Water Resources, 166(104258).

[2] Chen , T.-H. K., Kincey, M. E., Rosser, N. J. & Seto, K. C., 2024. Identifying recurrent and persistent landslides using satellite imagery and deep learning: A 30 year analysis of the Himalaya. Science of The Total Environment, 922(171161).

[3] Tominaga, M., 1998. Case Study of the Social Impact of Flood. IFAC Proceedings Volumes, 31(Issue 28), pp. 69-74.

[4] Guo, K., Guan, M. & Yan, H., 2023. Utilizing social media data to evaluate urban flood impact in data scarce cities. International Journal of Disaster Risk Reduction, 93(103780).
[5] Kourtit, K., Nijkamp, P. & Banica, A., 2023. An analysis of natural disasters' effects – A global commparative study of 'Belssing in Disguise'. Socio-Economic Planning Sciences, 88(101599).
[6] Tabari, H., 2021. Extreme value analysis dilemma for climate change impact assessment on global flood and extreme precipitation. Journal of Hydrology, 593(125932).
[7] Park, K. & Lee, E. H., 2024. Urban flood vulnerability analysis and prediction based on the land use using Deep Neural Network. International Journal of Disaster Risk Reduction, 101(104231).
[8] Fischer, S. & Schumann, A. H., 2024. Temporal changes in the frequency of flood types and their impact on flood statistics. Journal of Hydrology X, 22(100171).
[9] Kannapiran, U. M. & Bhaskar, A. S., 2024. "Flood inundation mapping of upstream region in the Adyar River basin: Integrating hydrologic engineering centre's river analysis system (HEC-RAS) approach with groundwater considerations". Groundwater for Sustainable Development, 24(101085).
[10] Panda, R. K., Pramanik, N. & Bala, B., 2010. Simulation of river stage using artificial neural network and MIKE 11 hydrodynamic model. Computers & Geosciences, 36(Issue 6), pp. 735-745.
[11] Banks, J. C., Camp, J. V. & Abkowitz, M. D., 2013. Adaptation planning for floods: a review of available tools. Natural Hazards, Volume 70, pp. 1327-1337.
[12] Alén, G. G., Montalvo, C., Cea, L. & Puertas, J., 2024. Iber-PEST: Automatic calibration in fully distributed hydrological models based on the 2D shallow water equations. Environmental Modelling & Software, 177(106047).
[13] Hoven, K. v. d., Kroeze, C. & Loon-Steensma, J. M. v., 2022. Characteristics of realigned dikes in coastal Europe: Overview and opportunities for nature-based flood protection. Ocean & Coastal Management, 222(106116).
[14] Fenton, J. D., 2024. Convolution, deconvolution, the unit hydrograph and flood routing. Journal of Hydrology, 634(131034).
[15] Apicella, A., Donnarumma , F., Isgrò, F. & Prevete, R., 2021. A survey on modern trainable activation functions. Neural Networksv, Volume 138, pp. 14-32.
[16] Akdemir, E. & Barışçı, N., 2024. A review on deep learning applications with semantics. Expert Systems with Applications, 251(124029).
[17] Nourani, V., Baghanam, A. H., Adamowski, J. & Kisi, O., 2014. Applications of hybrid wavelet–Artificial Intelligence models in hydrology: A review. Journal of Hydrology, Volume 514, pp. 358-377.
[18] Ajorloo, S. et al., 2024. A systematic review of machine learning methods in software testing. Applied Soft Computing, Issue 111805.
[19] Bahrami, S. et al., 2024. Application of artificial neural network to multiphase flow metering: A review. Flow Measurement and Instrumentation, 97(102601).
[20] Camero , A., Toutouh, J. & Alba, E., 2020. Random error sampling-based recurrent neural network architecture optimization. Engineering Applications of Artificial Intelligence, 96(103946).
[21] Quiroga, V. M., Kure, S., Udo, K. & Mano, A., 2016. Application of 2D numerical simulation for the analysis of the February 2014 Bolivian Amazonia flood: Aplication of the new HEC-Ras version 5. Ribagua - Revista Iberoamericana del Agua, Volume 3(1), pp. 25-33.
[22] Golshanrad, P. & Faghih, F., 2024. DeepCover: Advancing RNN test coverage and online error prediction using state machine extraction. Systems and Software, 211(112024).
[23] Gholami, H. et al., 2023. Interpretability of simple RNN and GRU deep learning models used to map land susceptibility to gully erosion. Science of The Total Environment, 904(166960).
[24] Lindemann, B. et al., 2021. A survey on long short-term memory networks for time series prediction. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, CIRP ICME 20, Volume 99, pp. 650-655.
[25] Cho, K. et al., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Doha, Qatar, Association for Computational Linguistics, pp. 1724-1734.
[26] Parisi, L., Ma, R., RaviChandran, N. & Lanzillotta, M., 2021. hyper-sinh: An accurate and reliable function from shallow to deep learning in TensorFlow and Keras. Machine Learning with Applications, 6(100112).
[27] Setiyono, T. et al., 2024. Application of TensorFlow model for identification of herbaceous mimosa (Mimosa strigillosa) from digital images. Smart Agricultural Technology, Volume 7(100400).
[28] Haghighat, E. & Juanes, R., 2021. SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. Computer Methods in Applied Mechanics and Engineering, Volume 373(113552).
[29] Elsner, J., 2023. Taming the Panda with Python: A powerful duo for seamless robotics programming and integration. SoftwareX, Volume 24(101532).
[30] Abert, C. et al., 2015. A full-fledged micromagnetic code in fewer than 70 lines of NumPy. Journal of Magnetism and Magnetic Materials, Volume Volume 387, pp. 13-18.
[31] [31] ANAR & INHGA, 2023. PLANUL DE MANAGEMENT AL RISCULUI LA INUNDAȚII A.B.A. ARGEȘ – VEDEA CICLUL II DE IMPLEMENTARE A DIRECTIVEI INUNDAȚII 2007/60/CE, București: s.n.

[32] Kraus, K. & Pfeifer, N., 2001. Advanced DTM Generation from lidar data. International Archives of Photogrammetry and Remote Sensing, Volume XXXIV-3/W4, pp. 22-24.
[33] HARUN, M. & RASHID, U., 1990. Estimation of Manning's Roughness Coefficient for Basin and Border Irrigation. Agricultural Water Management, Volume 18, pp. 29-33.
[34] Azmon, B., 1992. Manning coefficient of roughness - a case study along Soreq stream, 1971-1981. Journal of Hydrology, Volume 132, pp. 361-377.
[35] BAO, H.-j. & ZHAO, L.-n., 2011. Hydraulic model with roughness coefficient updating method based on Kalman filter for channel flood forecast. Water Science and Engineering, Volume 4, pp. 13-23.