The use of Artificial Neural Networks with different sources of spatiotemporal information for flash flood predictions

EGU General Assembly 2019, Vienna, Austria, 7-12 April 2019; Session HS3.1: *Hydroinformatics: computational intelligence, systems analysis, optimisation, data science, high-performance computing*

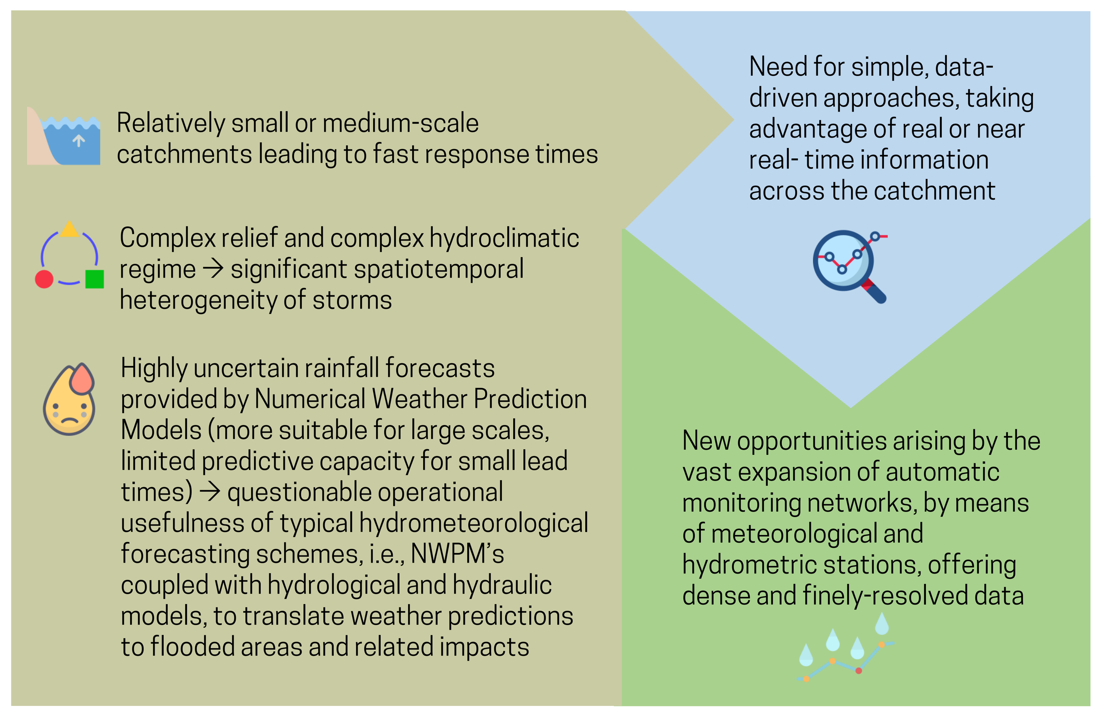
L. M. Tsiami[1], E. Zacharopoulou[1,2], D. Nikolopoulos[1], I. Tsoukalas[1], N. Mamassis[1], A. Kallioras[2], and A. Efstratiadis[1]

(1) Department of Water Resources & Environmental Engineering, National Technical University of Athens, Greece; (2) School of Mining & Metallurgical Engineering, National Technical University of Athens, Greece

Abstract

For more than two decades, the use of artificial neural networks (ANNs) in hydrology has become an effective and efficient alternative against traditional modeling approaches, i.e. physically-based or conceptual. These can take advantage of any type of available information to predict the hydrological response of complex systems, with missing data and limited knowledge about the transformation mechanisms. A promising area of application is the real-time prediction of flood propagation, which is essential element of early warning and early notification systems. In this work we focus to flash floods, considering as areas of application two medium-scale catchments in Greece with substantially different characteristics. The first one is the highly urbanized river basin of Kephissos (380 km^2), which is the main drainage channel of the Athens Metropolitan area, while the second is the rural catchment of Nedontas, SW Greece (120 km^2). Both areas have been recently equipped with automatic hydrometric stations, while online rainfall data are also available at a representative number of meteorological stations. For the two case studies we investigate several setups of ANNs, in order to predict the river stage at the catchment outlet for several lead times, using different combinations of input sets, by means of upstream stage and point rainfall data.

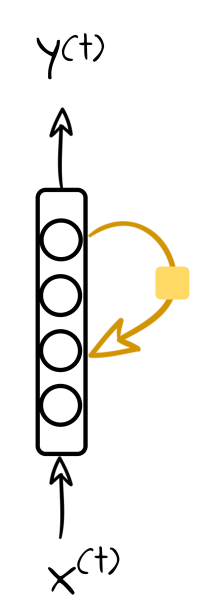
Flash flood forecasting challenges in the Mediterranean



LSTM Neural Networks

A Long-Short Term Memory (LSTM) Neural Network is a type of RNN (Recurrent Neural Network) that scans its input layer across sequences and learns how to extract information from a given event in order to make use of it at a later point in a sequence. The reason an LSTM is able to create representations of the past is that they are able to remember what they've seen previously, because they have a recurrent connection between their hidden layer and their input. The key characteristics of an LSTM are the following:

1. They learn to predict the output but they also learn how to compact all of the previous inputs.
2. The input to an LSTM is a concatenation of the original stateless input and the hidden state. This idea of a persistent hidden state that is learned from ordered inputs is what distinguishes an LSTM from linear and deep neural networks. In a DNN the hidden state is not updated during prediction. In RNN, it is.
3. They are designed for modelling long-term dependencies.



All of the above make LSTM a promising tool with multiple applications in hydrology where time series prediction with long range dependencies often occur.

Outline of Methodology

Goal: The implementation of LSTMs to predict the river stage at the basin outlet of a river using observed data across the upstream river network and rainfall data from representative meteorological stations.

For each case study:

By isolating the historical flood events from the available timeseries data, we analyze the correlations between the target variable $y(t)$ (river stage at the outlet gauge station) and alternative sets of explanatory variables, $x_i(t-\tau)$, for several lead times $\tau = 1, \ldots, N$.

An explanatory variable may refer to a known (i.e., current or past) process, i.e.:

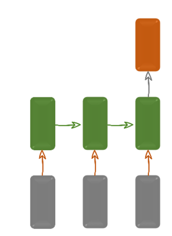
Stage Data
at the target (outlet) gauge station or at all upstream hydrometric stations

Point Rainfall Data
across areas of the river basin that are not captured by the hydrometric network

For each event and variable is calculated the sequence of empirical Pearson correlation coefficients, $\rho_{i\tau}$, and are extracted both the value of the maximun one, $\rho_i^* = \max(\rho_{i1}, \ldots \rho_{iN})$, and its corresponding lead time τ_i^*.

Based on the analysis of several flood events, we retain the *n* most informative explanatory variables. These will be the ones we will use as input on our LSTM.

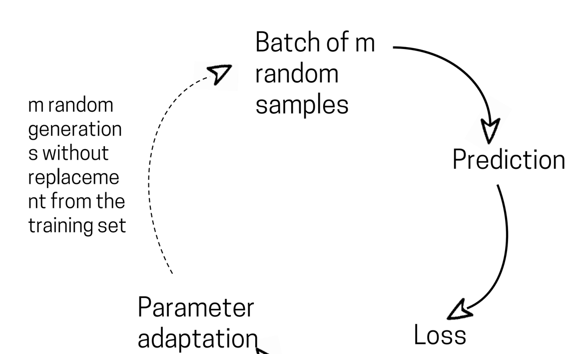
The architecture of LSTM we use is called sequence to one. In sequence to one model, a sequence (rain and stage timeseries) is passed in the LSTM and there is a non-sequence output (stage at the basin outlet).



Since the input layer accepts a sequence, the length of that sequence is important, as it must be enough to capture the dynamics of the phenomenon. Based on our analysis of the correlation coefficients and the lag time they maximize we choose the appropriate length p.

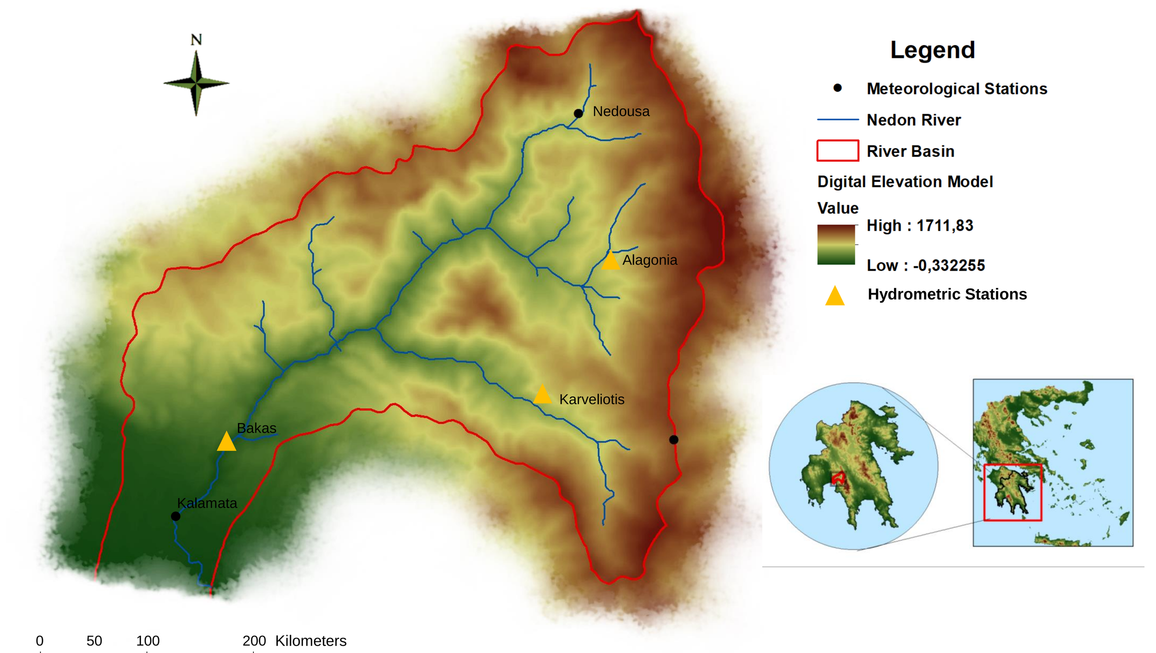
From the available timeseries for each variable we create multiple subsequences using a sliding window p using one timestep. Each instance of the sliding window corresponds to a record in the training set. So the input X corresponds to p sequential values for each input feature. If n features (explanatory variables) are used as input, that means that X is an (nxp) matrix. The output Y corresponds to the stage value at the desired timestep t+h.

A schematic illustration of one iteration step in the LSTM training/calibration is provided in the following figure. One iteration step during the training of LSTMs works with a subset (called batch or mini-batch) of the available training data. The number of samples per batch is hyperparameter of the neural network.
Note that, based on our hypothesis that each stage is a function of p steps of the explanatory variables, the samples within a batch can be random. This approach is entirely different from the procedure a traditional hydrological model would follow were data are processed in order of appearance.

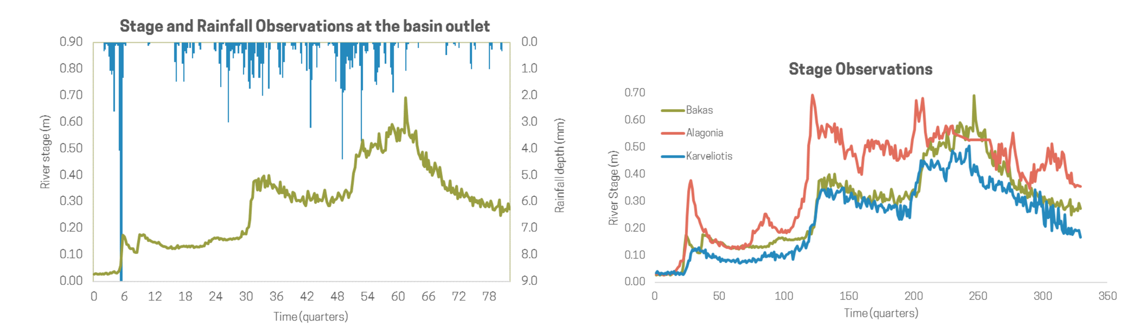


Since the goal is to predict the river stage at multiple timesteps and the model used outputs only one prediction at a specific t+h timestep, we created h training sets. Each of which has the same input X, but different output Y. By learning each training set independently, we obtain h regression models fi(i=1,2,…,h). The models are then used to predict the next h values as follows: yt+i=fi(X), i=1,2,…h.
Moreover each of these stations provided us with 15-minute interval data. We decided to make predictions for the following two hours so that led in creating h=8 different regression models.

Case study:
Prediction of Nedontas river stage (SW Greece) based on combined stage and rain data

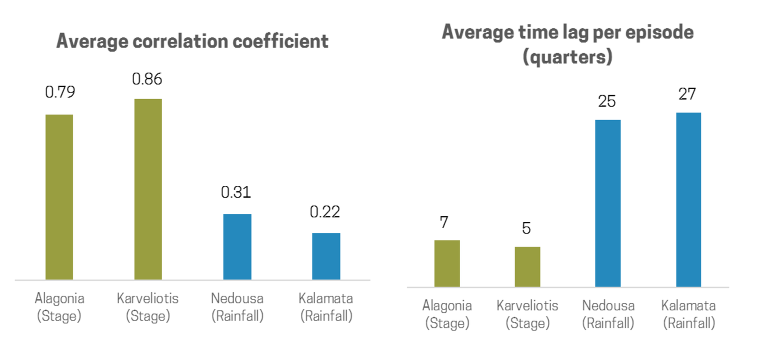


Nedontas river, located in SW Greece, flows in southwestern direction and is fed by several small tributaries. Stage and rain data are available from two of its main tributaries (Karveliotis and Alagonia) and at the basin outlet (Bakas). Further rain data that are also available from three additional meteorological stations (Poliani, Nedousa, Kalamata).



Stage and Rainfall observations during a typical flood event

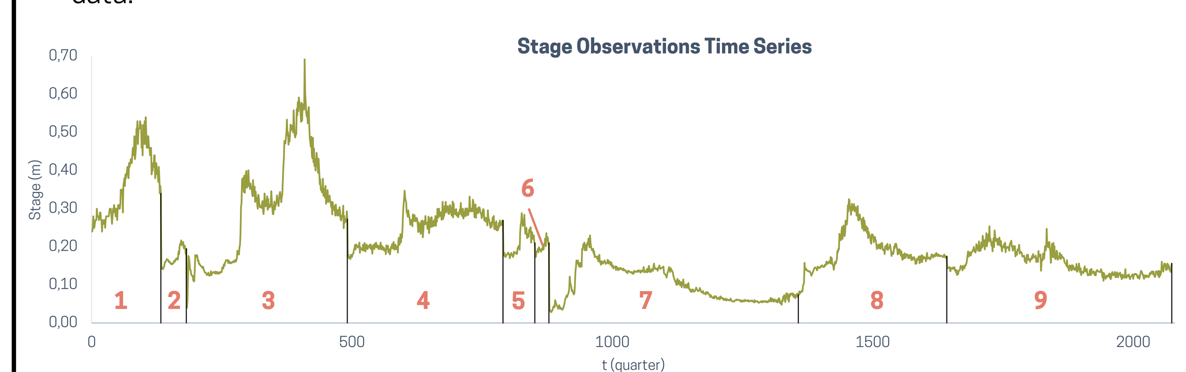
Nine flood events that occurred during the period 2012-2014 were isolated for analysis. The analysis showed that the stations with the most reliable information for explaining the river stage at the outlet station (Bakas) are the meteorological stations Nedousa and Kalamata and the hydrometric stations Alagonia and Karveliotis.



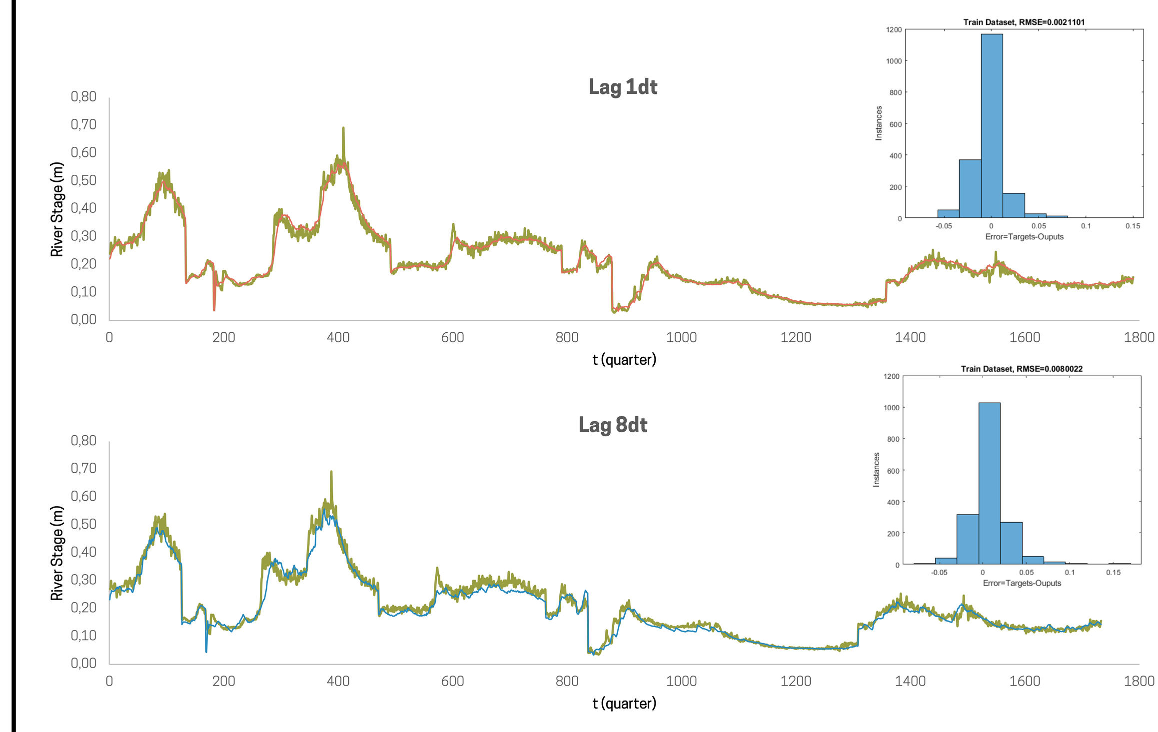
From the above it is noticeable that the hydrometric stations have a larger correlation than the meteorological ones. The exact opposite occurs when it comes to the time lag.

Based on our analysis of the correlation coefficients we decided to set the length of the input sequence constant and equal to 20 timesteps (5 hours). We trained our LSTM each time using a different episode as a test dataset and compared the results.

No matter the training set, the LSTM performed well enough on all regression models we created, meaning it was able to make predictions two hours ahead of the observed data.

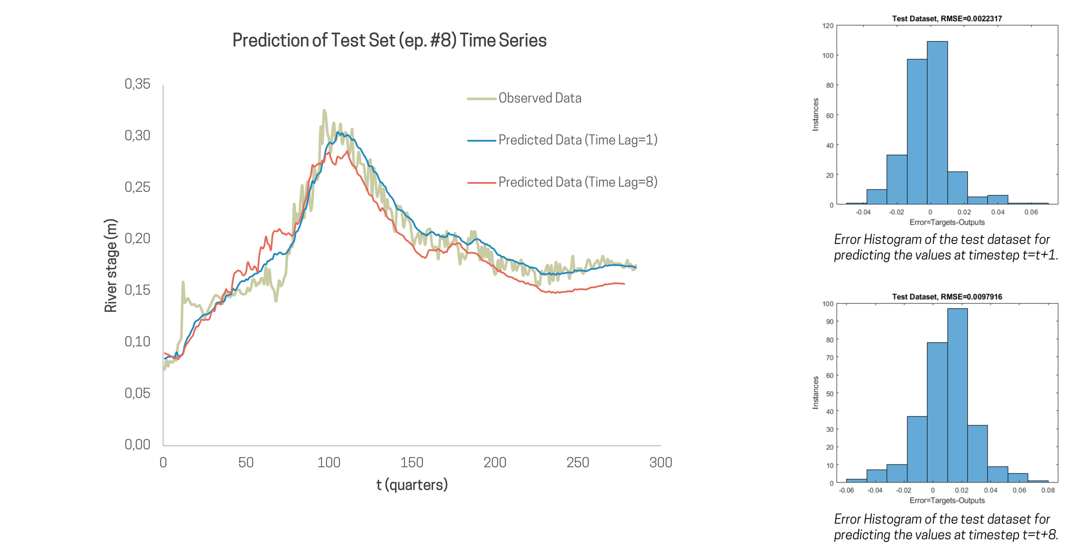


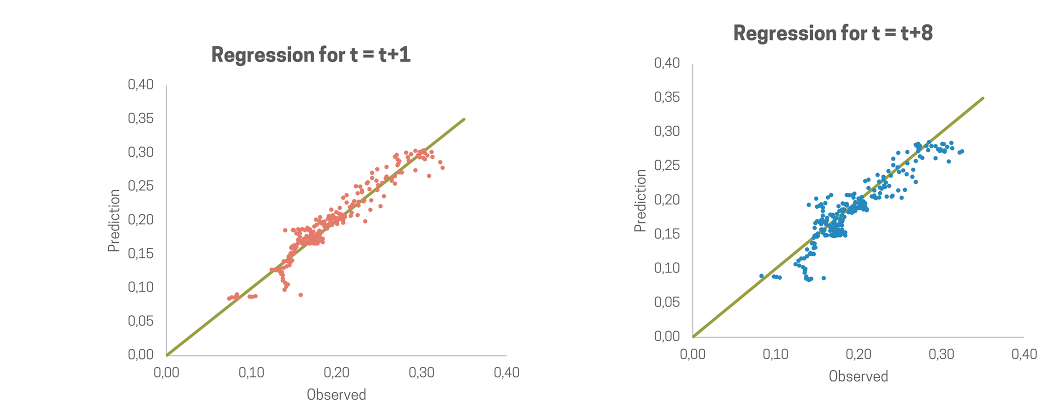
Results and Conclusions



Observed values vs Outputs from the training stage from different time lag regression models

The LSTM seems to do a pretty good job at the training stage and its outputs are pretty close to the observed values. The output values for a lag of t=t+8 are a bit off, but that is expected, due to the missing data. Note that, the purpose of this project was to explore the potential of these networks and not to find the optimum solution using them.





Regression Model with Time Lag t = t+1

	R	MSE	RMSE	# SAMPLES
TEST SET	0.9611	0.00019954	0.0022	285
TRAIN SET	0.9889	0.00028015	0.0021	1789

Regression Model with Time Lag t = t+8

	R	MSE	RMSE	# SAMPLES
TEST SET	0.9197	0.00045699	0.0098	278
TRAIN SET	0.9821	0.00053478	0.008	1733

A similar conclusion can also be drawn by looking at the results from the testing stage. Prediction for time lag t=t+8 tends to underestimate the river stage but no more than 5 cm. It is important to remind that the LSTMS were used in order to explore their potential for solving timeseries problems in the field of hydrology and ultimately becoming useful tools in flood prediction.

References

Cheng H., Tan PN., Gao J., Scripps J. (2006) Multistep-Ahead Time Series Prediction. In: Ng WK., Kitsuregawa M., Li J., Chang K. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2006. Lecture Notes in Computer Science, vol 3918. Springer, Berlin, Heidelberg

Kratzert, Frederik, et al. "Rainfall-Runoff modelling using Long-Short-Term-Memory (LSTM) networks." *Hydrol. Earth Syst. Sci. Discuss., https://doi. org/10.5194/hess-2018-247, in review*(2018).

Contact info: lydtsiam@gmail.com
Presentation available at: http://www.itia.ntua.gr/1940/