# Deep learning models to predict flood events in fast-flowing watersheds

Marco Luppichini[a,b,*], Michele Barsanti[c], Roberto Giannecchini[b,d,e], Monica Bini[b,d,f]

[a]Department of Earth Sciences, University of Study of Florence, Via La Pira 4, Florence, Italy.

[b]Department of Earth Sciences, University of Pisa, Via S. Maria, 52, 56126 Pisa, Italy

[c]Department of Civil and Industrial Engineering, University of Pisa, Largo L. Lazzarino, 56122 Pisa, Italy

[d]CIRSEC Centro Interdipartimentale di Ricerca per lo Studio degli Effetti del Cambiamento Climatico dell'Università di Pisa, Via del Borghetto 80, 56124 Pisa, Italy.

[e]Institute of Geosciences and Earth Resources, IGG-CNR, via Moruzzi 1,56124 Pisa, Italy

[f]Istituto Nazionale di Geofisica e Vulcanologia (INGV), Via Vigna Murata 605, 00143 Roma, Italy.

Abstract

This study aims to explore the reliability of flood warning forecasts based on deep learning models, in particular Long-Short Term Memory (LSTM) architecture. We also wish to verify the applicability of flood event predictions for a river with flood events lasting only a few hours, with the aid of hydrometric control stations. This methodology allows for the creation of a system able to identify flood events with acceptable errors within several hours' notice. In terms of errors, the results obtained in this study can be compared to those obtained by using physics-based models for the same study area. These kinds of models use few types of data, unlike physical models that require the estimation of several parameters. However, the deep learning models are data-driven and for this reason they can influence the results obtained. Therefore, we tested the stability of the models by simulating the missing or wrong input data of the model, and this allowed us to achieve excellent results. Indeed, the models were stable even if several data were missing. This method makes it possible to lay the foundations for the future application of these techniques when there is an absence of geological-hydrogeological information preventing physical modelling of the run-off process or in cases of relatively small basins, where the complex system and the unsatisfactory modelling of the phenomenon do not allow a correct application of physical-based models. The forecast of flood events is fundamental for correct and adequate territory management, in particular when significant climatic changes occur. The study area is that of the Arno River (in Tuscany, Italy), which crosses some of the most important cities of central Italy, in terms of population, cultural heritage, and socio-economic activities.

*Keywords:* Arno River, Deep Learning, Fast Catchment Basin, Flood Forecasting, Hydraulic models, LSTM.

## 1. Introduction and goals

River flow prediction is a fundamental goal for early flood warning, water resource management, water demand assessment, irrigation, agriculture, and hydroelectric power generation. These aspects become more and more critical in the context of climate changes. The extreme weather events causing flash floods, floods, and debris flow phenomena have relevant socio-economic implications and represent a significant scientific issue producing extensive literature on the subject (e.g., Bates et al., 2008; Bryndal et al., 2017; Gaume et al., 2016, 2009; IPCC, 2014).



*Corresponding author.
Email addresses: marco.luppichini@unifi.it (Marco Luppichini), michele.barsanti@unipi.it (Michele Barsanti), roberto.giannecchini@unipi.it (Roberto Giannecchini), monica.bini@unipi.it (Monica Bini).

The method most currently used for modelling of the hydrologic scenario simplifies the hydrologic system features to predict their behaviour (Antonetti and Zappa, 2018). The physical models aim to understand and replicate the natural events by using mathematical and numeric methods based on several parameters simplifying the natural phenomena (Jaiswal et al., 2020). However, the systems in nature (including the hydrologic ones) are inherently heterogeneous (Marçais and de Dreuzy, 2017) and physically-based models may show inherent limitations in reproducing natural phenomena (Islam, 2011). Furthermore, in a physical hydraulic model, a wrong setting or a wrong estimation of the parameters cause an increase in model errors and this is the main drawback of the physical models (Luppichini et al., 2019) .

The recent developments of artificial intelligence (AI) and graphic processor units (GPU) have enabled advancements in deep learning applications, and innovative approaches based on multilayer artificial neural networks (ANN) (Goodfellow et al., 2016; LeCun et al., 2015). The application of deep learning models in various real-world cases, especially in the time series prediction (Fawaz et al., 2020; Yi et al., 2019; Zheng et al., 2019) has been successful. These procedures are the most appropriate to tackle the noisy and chaotic nature of the time series forecasting problems (Livieris et al., 2020). Several authors have applied different deep learning techniques to predict river flows with promising results (Boulmaiz et al., 2020; Chattopadhyay et al., 2020; Kratzert et al., 2018; Marçais and de Dreuzy, 2017; Sit et al., 2020; Tien Bui et al., 2020; Van et al., 2020). The long short-term memory (LSTM) is one of the most popular, efficient and deeply used learning techniques (Fawaz et al., 2020), widely applied in flood prediction studies (Boulmaiz et al., 2020; Hu et al., 2020; Kratzert et al., 2018; Le et al., 2019; Li et al., 2020; Liu et al., 2020; Nguyen and Bae, 2020).

This research aims to demonstrate that deep learning approaches can be used to predict flood events in watersheds characterized by fast flood events. The study area is represented by the Arno River basin (in Tuscany, Italy), which experienced many damaging floods in the past (Becchi I., 1986; Caporali et al., 2005) and which is characterised by very sudden events that developed in less than 24 hours. We have used very high-frequency data to reconstruct flood events in great detail. The work is designed to create deep learning models able to predict flood events within a few hours' notice. The architecture of the applied deep learning model is based on an encoder-decoder LSTM layer, by simply using rainfall and hydrometric height data. The few types of data exploited by a deep learning model, as opposed to a physical-based model, allows the user to apply the same techniques of this study to different cases. Deep learning techniques will make it possible to create flood alarm systems in catchments for which there is scarce territorial information. Many studies have applied these techniques to large river basins characterized by multi-day events. Some examples may include studies that have involved several watersheds scattered across various continents (Boulmaiz et al., 2020; Kratzert et al., 2018), large basins of North America (Chattopadhyay et al., 2020), and Chinese rivers having large flow rates (Le et al., 2019; Li et al., 2020; Liu et al., 2020; Van et al., 2020). Thus, the application of the method to smaller catchments is excellent news and of relevant importance for many purposes.

Formatted: Not Highlight

Formatted: Not Highlight

Formatted: English (United States)

## 2. Study Area

The study area is located in Tuscany, central Italy. The region includes many important elements at risk: high population and density (especially in fluvial plains); cultural heritage; historical towns rich in art and monuments and crossed by rivers (e.g., Florence and Pisa); industrial and artisanal settlements often situated in fluvial plains.

The Arno River basin (Figure 1) is approximately 8300 km$^2$ wide, the fifth in extension in Italy, and is bordered by the Apennine chain from north to east. The average and maximum elevations are about 350 and 1600 m a.s.l. respectively. The Arno is the main river in Tuscany and one of the longest (240 km) in Italy. Thus, flood forecasting is strategic in the Arno River basin, since the river crosses the

two main Tuscan cities of Florence and Pisa, as well as many production centers. Figure 1 also shows the location of the hydrometric stations used in this study, whereas Figure 2a and 2b show the digital terrain model (DTM) and the slope of the basin.

Owing to its geological, geomorphological and morphometric structure, the Arno River basin is characterized by a general relatively short run-off time. However, different sectors of the river characterized by different run-off times can be recognized: in the mountainous hilly area, the run-off times range from 4 to 6 hours, whereas in the lowland areas the run-off time is ca 20-24 hours (Autorità di Bacino del Fiume Arno, 1989), approximately. This behaviour is interesting to understand whether the applied methodology has limitations when the basin run-off times are short.

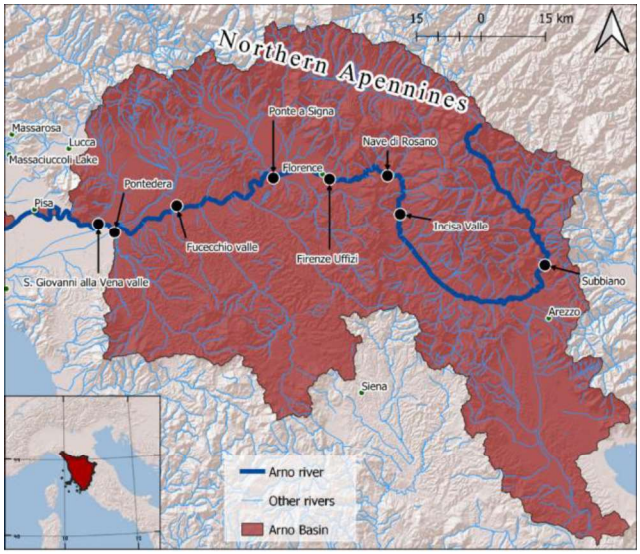
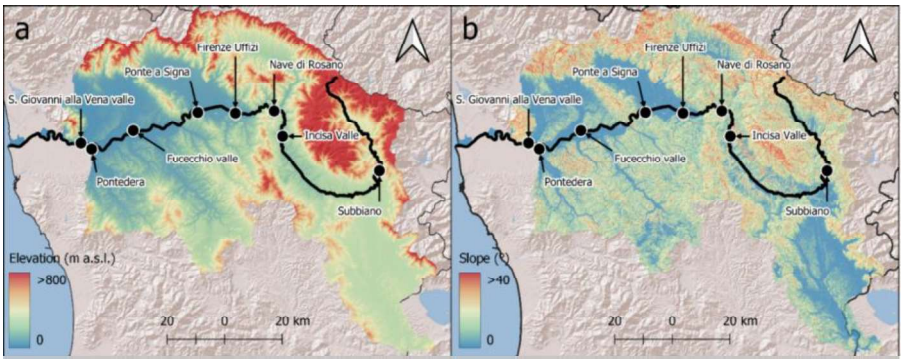



Figure 1. The Arno River basin and its main drainage network.




Formatted: Normal

Formatted: Font: 10 pt

**Figure 2. Morphological characteristics of the Arno River Basin: a) 10×10m DEM provided by the Tuscany Region (https://www.regione.toscana.it/-/geoscopio); b) grid of the slope obtained from DEM. The black dots identify the eight hydrometric stations used in this work.**

## 3. Methodology

### 3.1. Database and data input pre-processing

The dataset used in this work (provided by the Regional Hydrologic Service, hereafter SIR), comes from a very rich monitoring network that includes 487 raingauges and 193 hydrometers (Figure 4**Error! Reference source not found.**), with a sampling time of 15 minutes. For our models, we selected the most complete time series using 48 raingauges and 35 hydrometric stations for the decade 2010-2020. The 15-minute sampling time allowed us to better describe the phenomena that develop over a short time span. In fact, a low sampling frequency could lead to run-off and rainfall curves that do not adequately represent the natural phenomenon. From this point of view, deep learning models are capable of reproducing these curves, resulting to be precise but not very accurate. On the other hand, the high frequency of sampling involves an increase of the noise in the data that could create errors in the models. In our case, we have no noise signal in the data, and for this reason we have chosen to use the maximum resolution available for the study area. In our study, we want to get as close as possible to the real phenomenon by limiting simplifications in the flow or in the rainfall curves.

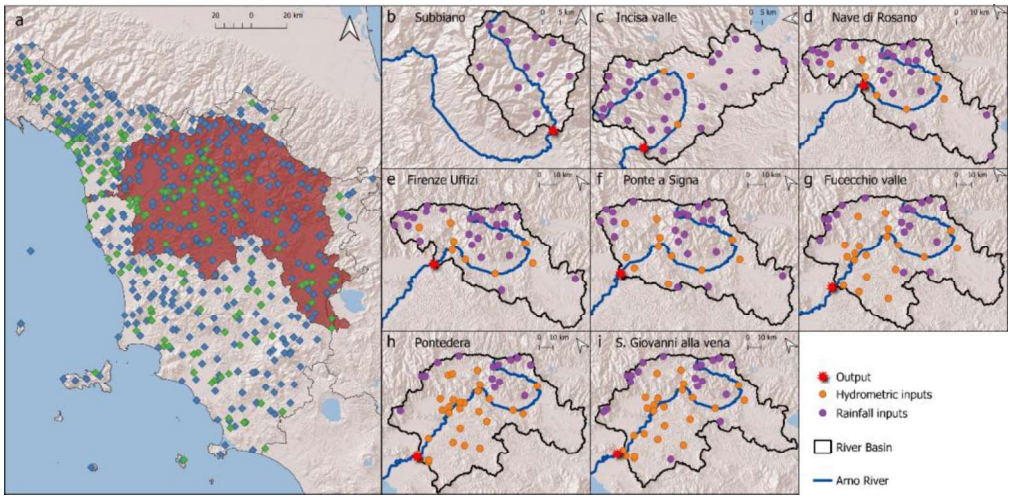



**Figure 3. a) Monitoring network of SIR (15-minute sampling frequency; source: https://www.sir.toscana.it/consistenza-rete). The brown region indicates the Arno River Basin closed at the hydrometric station of S. Giovanni alla Vena, whereas the blue rhombi represent the rainfallgauges and the green rhombi represent the hydrometers. b-i) Distribution of output hydrometers (red stars), input hydrometers (orange) and raingauges (violet) in the 8 sub-basins simulated in this work.**

For our purposes, the Arno River basin was divided into eight sub-basins closed at specific hydrometric stations, which were used as model outputs. From upstream to downstream, we selected Subbiano, Incisa valle, Nave di Rosano, Firenze Uffizi, Ponte a Signa,

Fucecchio valle, Pontedera and S. Giovanni alla Vena (see Figure 2). These stations were chosen for having almost complete time series with a small number of missing values, and for being in strategic positions for monitoring of the river. The stations of Subbiano, Incisa Valle, and Nave di Rosano are positioned upstream of the city of Florence, thus allowing to predict flood events upstream of the city of Florence. Firenze Uffizi station is a key to understanding the hydraulic behaviour of the river in the most important and vulnerable towns or cities of Tuscany. Ponte a Signa station is near a large flood reservoir, whereas the Fucecchio valle station provides information on the hydraulic behaviour of the river in a completely flat area. Pontedera station is near the Scolmatore, an artificial canal built after the 1966 flood in order to protect Pisa (the second main Tuscan city crossed by the Arno River) from flooding. The Scolmatore Canal drains the waters of the Arno River directing them towards the sea. S. Giovanni alla Vena station is the nearest one upstream to Pisa. Figure 4 shows the hydrometric and rainfall stations used for each sub-basin model. We chose the input stations according to their datasets and to their geographical positions. This choice was made to obtain a homogenous distribution of station in each sub-basin. Table 1 lists the characteristics of each modelled sub-basin and the relative stations considered. For each sub-basin, Mean Annual Precipitation (MAP) and Mean Annual Temperature (MAT) were computed by using the 1920-2020 data on the basis of the meteorological stations indicated.

**Table 1. Features of each modelled sub-basin and of the stations used (MAP: Mean Annual Precipitation; MAT: Mean Annual Temperature (MAT) and relative standard deviation - Data from 1920 to 2020).**

| Output Hydrometric Station | Sub-basin area ($km^2$) | Sub-basin average elevation (m a.s.l.) | n. input raingauges | n. input hydrometers | MAP | MAT |
|---|---|---|---|---|---|---|
| **Subbiano** | 750 | 750 | 11 | 0 | 1295 ± 276 | 13 ± 2 |
| **Incisa Valle** | 2,840 | 580 | 28 | 3 | 956 ± 193 | 18 ± 1 |
| **Nave di Rosano** | 3,840 | 460 | 26 | 8 | 1017 ± 190 | 17 ± 1 |
| **Firenze Uffizi** | 3,970 | 450 | 25 | 9 | 1014 ± 189 | 17 ± 1 |
| **Ponte a Signa** | 4,540 | 430 | 22 | 12 | 989 ± 185 | 18 ± 1 |
| **Fucecchio** | 6,600 | 370 | 19 | 20 | 997 ± 188 | 18 ± 1 |
| **Pontedera** | 7,850 | 340 | 14 | 30 | 1000 ± 188 | 19 ± 1 |
| **S. Giovanni alla Vena** | 8,030 | 320 | 19 | 25 | 996 ± 186 | 19 ± 1 |

For each output hydrometer, we built a deep learning model to predict its 15-minute measurements ($H_t$). The mathematical expression of the model, representative of all the investigated sub-basins, can be defined as follows:

$$\widehat{H} = f(X_t) = f(H_{t-1}, H_{t-2}, \dots, H_{t-n}, R_{t-1}, R_{t-2}, \dots, R_{t-m}) \quad (1)$$

where $\widehat{H}$ stands for the predicted hydrometric height at time $t$; $H_{t-1}, H_{t-2}, \dots, H_{t-n}$ are the antecedent hydrometric heights (up to $t$–1, $t$–2, …, $t$–$n$ time steps); $R_{t-1}, R_{t-2}, R_{t-m}$ are the antecedent rainfall ($t$–1, $t$–2, …, $t$–$m$ time steps).

Some tests and trials highlighted the need to create an input dataset with $t$ up to 96 steps. To decrease the noise contained by many steps and close measurements, we provided every $t$ for the first previous hour and then one every 4 steps (e.g., $t$-0, $t$-1, $t$-2, $t$-3, $t$-4, $t$-8, $t$-12, $t$-16, ..., $t$-96) up to the 24$^{th}$ hour.

The evolution of the riverbed influences the hydrometric measurements. Since a hydrometer measures the flow heights relative to a fixed point over time, sediment deposits near the hydrometer cause the measured value to be higher than the correct value (systematic error). On the contrary, the riverbed erosion causes a measurement lower than the correct one. For this reason, we could observe negative values of hydrometer measures or a progressive movement of the minimum annual level of the river. Furthermore, measurement errors could occur,

Formatted: Not Strikethrough

Formatted: Font: 9 pt

Formatted: Font: 9 pt

Formatted: Font: 9 pt

Formatted: Font: 9 pt

inducing inconsistencies and incoherence in the dataset. These problems of hydrometric measurements are summarized in Figure 4 (blue line), which shows that the past year presents the highest minimum flow of the river and that there are some errors when measurements are equal to 0. For this reason, we normalized the hydrometric measurements for each time series used by removing the incorrect data and standardizing the measurements and then comparing the minimum measured each year with the last one (Figure 4, orange lines).

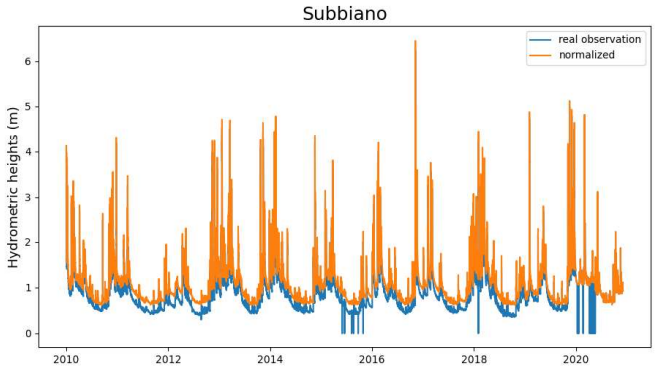



**Figure 4. Standardization of a hydrometric height time series (the example is refered to Subbiano station). We removed all fake data and calculated the minimum river level value that was stationary over time each year.**

### 3.2. Model development

To accomplish the deep learning models of this study, we mainly used the open-source framework Tensorflow (Abadi et al., 2015) and the libraries Numpy, Pandas, Scikit-Learn and Keras (Chollet, 2015) in Python language v 3.7 (van Rossum and Drake, 2009). The architecture of the developed models is based on an encoder-decoder LSTM, formed by two pairs of LSTM nodes (Figure 5). This architecture allows usage of an LSTM to read the input sequence, one step at a time, in order to obtain a fixed-size vector representation in a data structure that occupies a large amount of memory. We then introduce another LSTM to extract the output sequence from that vector (Sutskever et al., 2014). The encoder is composed of two sequence layers (LSTM) of 32 and 16 units, respectively, followed by a repeat vector node. The repeat vector layer repeats the incoming inputs for a specific number of times. The decoder is composed of two LSTM layers of 16 and 32 units respectively, followed by a time-distributed dense node as output of our model. To evaluate the discrepancy between the predicted and the measured values, we used a loss function for each observation, which allowed us to calculate the cost function. We needed to minimize the cost function by identifying the optimized values for each weight. Thanks to multiple iterations, the optimization algorithm computes the weights that minimize the cost function. In our implementation, we used the Adam optimizer (Kingma and Ba, 2014). Adam is an adaptive learning speed method, meaning that it computes individual learning rates for several parameters (Kingma and Ba, 2014). To stop the training, we used the specific API of Keras and, in particular, the early stopping method. This method allows the training procedure to stop when the monitored metric, namely the value of the cost function, has ceased to improve. Therefore, given all the possible hypotheses, we wanted to find the best one (called "optimal"). This hypothesis would make it possible for us to make more accurate estimates, still based on the data available. We split the dataset into three parts: training, validation, and test dataset (Figure 5). The training dataset includes the 2010-2017 data, and represents the input in the learning step. The validation dataset is composed of the 2017-2019 data and is used to calculate and optimize the loss function in the learning phase. Finally, we used

the test dataset (2019-2021) to evaluate the final model: the prediction step allowed us to forecast the hydrometric heights of the river because these data are unknown to the model. This partition of 60% - 20% - 20% for training, validation and test dataset is used by several studies (e.g., Hu et al., 2020; Li et al., 2020; Nguyen and Bae, 2020) and permits to have sufficient data for the training and the evaluation of the model. In detail, Li et al. (2020) use a similar subdivision of the dataset considering only one year for validation and for the test dataset.

The cost function used was the mean square error (MSE) calculated on the validation dataset. The partition of the whole dataset permitted to minimize the overfitting effect on the training set. We built a model for each hydrometric station and 25 forecasting steps for a total of 200 models considering the eight sub-basins modelled. The forecasting steps are one every hour from 0 to 24 hours.

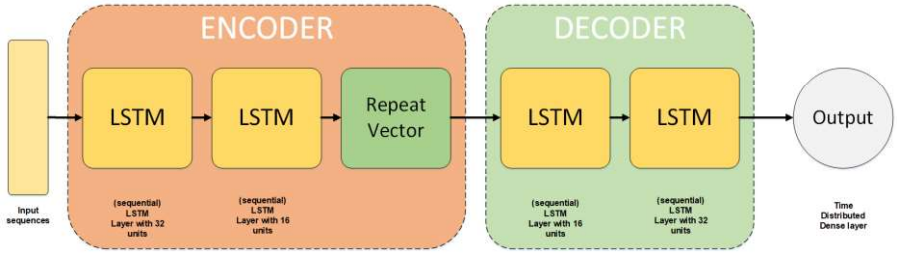



**Figure 5. Architecture of the deep learning model used in this study, based on the use of the LSTM node.**

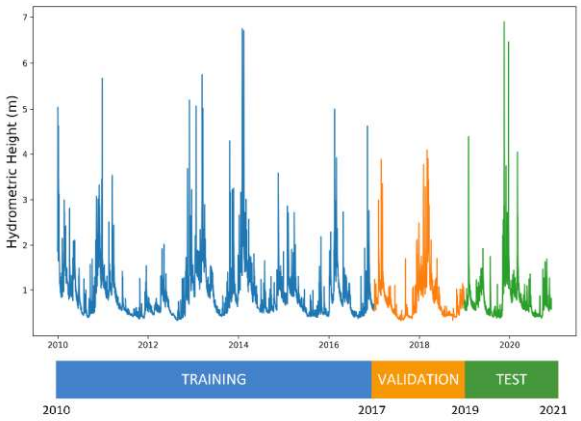



**Figure 6. Partition of the dataset in training, validation and test datasets. Training and validation datasets are the input for model training. The test datasets predict events that are unknown to the model. The example is refered to San Giovanni alla Vena station.**

## 4. Results

We evaluated the accuracy and precision of the models by analysing the model errors when predicting the maximum effects. For each station, Figure 7 shows the prediction accuracy of the 30 highest events occurred in 2019 and 2020, highlighting the absolute error, i.e.,

**Commented [ml1]:** Valutazione del modello

**Commented [ml2]:** Ripartizione / Divisione

**Formatted:** Not Highlight

the difference between the predicted value and the observed value. We also computed the relative error by dividing the absolute error by the measured value. We partitioned the results into four groups: 0-6 h, 7-12 h, 13-18 h and 19-24 h. In all cases, we can observe that the longer the forecast time, the greater were the errors. Furthermore, the percentage errors were higher for the upstream than for the downstream stations (Figure 7).

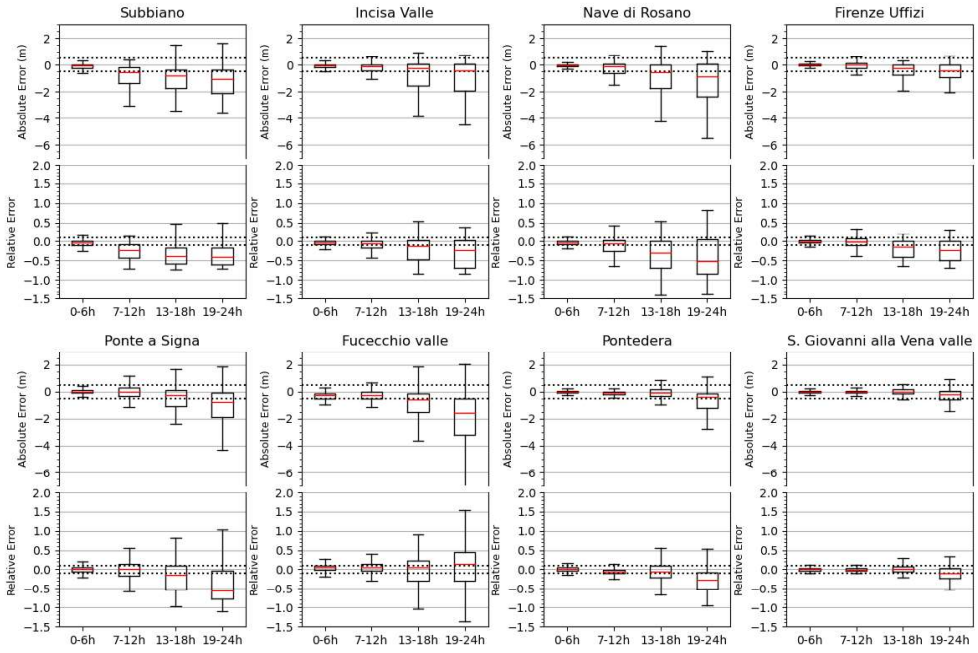



**Figure 7. Absolute and relative model errors for the 30 most severe events occurred between 2019 and 2020 (test dataset). In absolute error graphs, the dotted lines mark the range between -0.5 and +0.5m. In relative error graphs, the dotted lines mark the range between -0.1 and +0.1. The boxes represent the interval between the 25th and the 75th percentiles (Q1 and Q3). IQR is the interquartile range Q3-Q1. The upper whisker will extend to the last datum lower than Q3 + 1.5*IQR. Similarly, the lower whisker will reach the first datum higher than Q1 – 1.5*IQR. The green lines represent the medians.**

Figures 8 and 9 show two events that occurred on 3rd March 2020 and 17th November 2019, respectively and that represent examples of the behaviour of the Arno River during flooding episodes. Both events triggered a flood warning and the exceeding of the alert thresholds for the whole course of the river. Specifically, the event of 17th November 2019 is of considerable importance: in the 10-year time series used to train the model (see Figure 6), there is only one other case similar to that of 17th November 2019 (see Figure 9). The rainfall (ca 40 mm) fell from 2020-03-02 06:00 to 2020-03-04 00:00 on the basin of San Giovanni alla Vena. Instead, the event of 17th November 2019 was characterized by ca 65 mm of rainfall that fell on the same basin from 2019-11-16 12:00 to 2019-11-17 12:00. For this reason, we can consider the two cases as representative of an ordinary alert event (3rd March 2020, Figure 8) and an exceptional alert event (17th November 2019, Figure 9).

The figures show a first prediction (blue line), which started when the rainfall over the entire Arno River basin began to increase. The subsequent forecasts were temporally spaced 6 hours from each other. Each prevision lasted for 24 hours after the start. Prediction errors mainly depend on two factors: the location of the hydrometric stations, and the time interval between the start of the forecast and the instant of time when the level reaches its maximum value. The simulations carried out just before the onset of the rainfall event (blue line) or before the maximum values of cumulative rainfall on the river basin are characterized by the higher errors, with an impossibility for the model to simulate the flow event (Figures 9 and 10). The errors are lower for the simulation temporally close to the maximum flow event. When comparing Figures 9 and 10, the errors are higher in the second case, when the hydrometric heights of the river are greater.

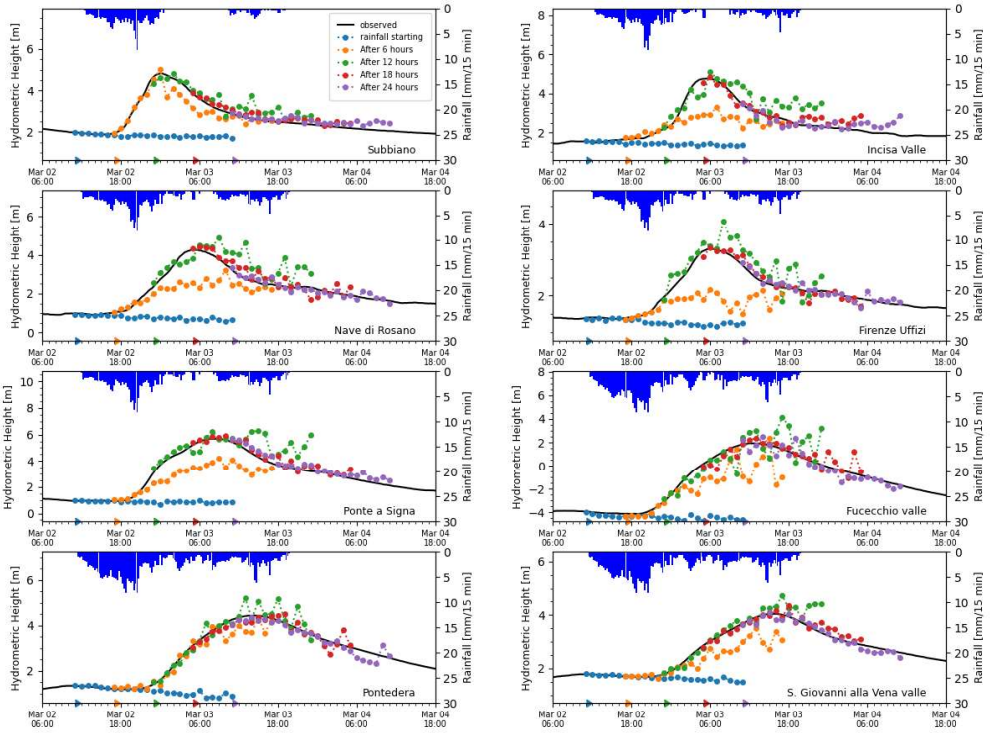



**Figure 8. Simulation of the 3rd March 2020 event, where the hydrometric level exceeded the alert threshold in all stations. The first forecast (blue line) begins when the rainfall on the Arno river basin increases. Subsequent predictions are temporally spaced 6 hours one from the other, each lasting 24 hours. The coloured triangles on the time axis indicate the start of each prediction. The blue bars indicate rainfall.**

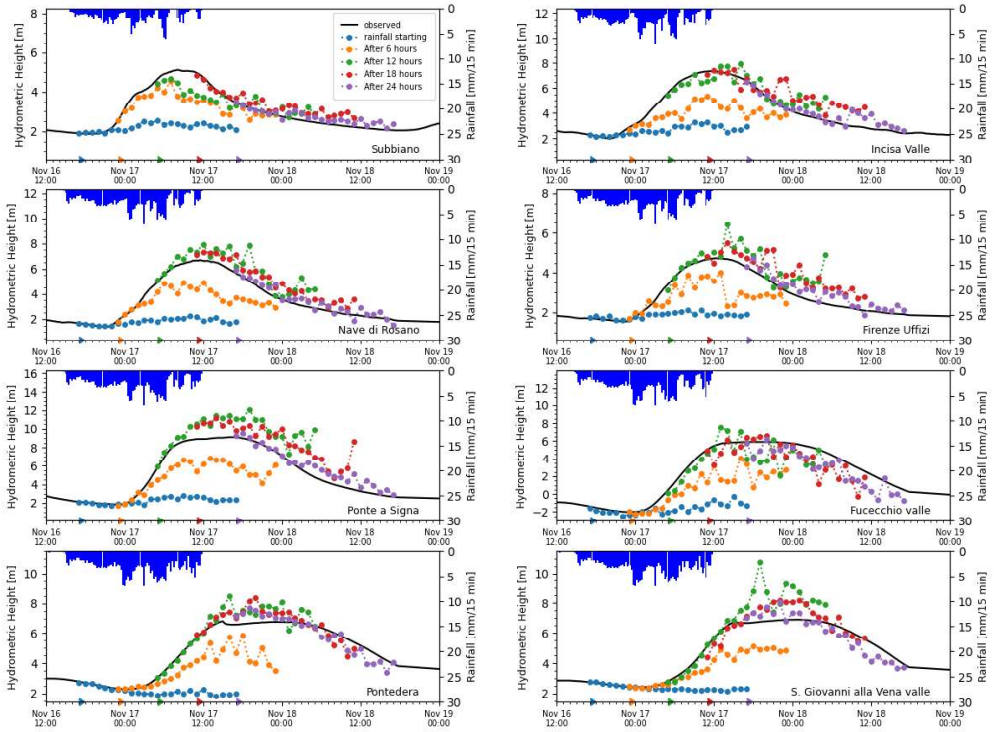



**Figure 9. Simulation of the 17th November 2019 severe event, where the hydrometric level exceeded the alert threshold in all stations. The first forecast (blue line) begins when the rainfall on the Arno river basin increases. Subsequent predictions are temporally spaced 6 hours from each other, each lasting for 24 hours. The coloured triangles on the time axis indicate the start of each prediction. The blue bars indicate rainfall.**

## 5. Discussion

As a result of the high frequency of data sampling (15 minutes), we obtained a large amount of data, which allowed for efficient model learning. Model errors are influenced by the forecasting time and by the location of the hydrometric stations. The forecasts for the hydrometric stations located at higher mean elevation exhibit the highest errors. Prediction with more than 7-12 hours for this type of basin is difficult, and is characterized by high errors. The problem is reasonably due to the shorter run-off time typical of these basins. The time interval between maximum rainfall event and maximum hydrometric height is obviously variable for the analyzed stations and it characterizes the different sub-basins. The shortest interval is associated with Subbiano (average of 13 hrs), the longest is attributable to S. Giovanni alla Vena (average of 23 hrs). Run-off times influence the capacity of the model to achieve a good prediction of the flow events.

Deep learning models are the simplest ones relatively to the data to be used and they show large flexibility at different basin scales. Their main advantage is the dynamics simplification of a run-off process by using only rainfall and hydrometric data. On the other hand, physically-based models require large amounts of different data that are sometimes very difficult to find or do not have a sufficient resolution and need specific assumptions. For example, a very low topographic resolution can cause a high error when applying a physical model (Luppichini et al., 2019) so that specific surveys are necessary. The few types of data necessary to create a deep learning model allow us to apply it in different environmental situations. These flow forecasts are valid for the entire Arno River, from its origin to the position just before its mouth. Therefore, we think these models could be applied to watercourses with different hydraulic behaviour. For these basins, it would be useful to compare our errors with those obtained by Ercolani and Castelli (2017) which used the hydrological model MOBIDIC (*MOdello di Bilancio Idrologico DIstribuito e Continuo*). MOBIDIC is a physically-based model used by the Tuscany Region authorities for the analysis of flood alarms (Ercolani and Castelli, 2017). Even in the case of the physical model, errors are higher in basins with shorter run-off times (e.g., Subbiano and Nave di Rosano stations) and they diminish for larger basins. In that specific case, the model tends to overestimate the water flow when the warning time is several hours (greater than about 12 hours from the maximum event) (Ercolani and Castelli, 2017). Instead, the deep learning models underestimate the hydrometric heights even with high errors when the simulation is performed before the rain falls on the basin. These models do not provide any specific information helping to understand when it will start raining and before this time, the rain is constantly equal to 0. Once the rainfall has started, our model can predict the hyetograph and understand how the hydraulic regime will evolve. The rainfall onset is a critical point in a flood warning system based on deep learning techniques. The forecast rainfall data can be made useful by implementing the classical techniques (physical models, analysis of satellite images, etc.), which can help these models overcome the current limits of these methods.

The deep run-off learning models need to use all the input stations for prediction, and this could be one of the main drawbacks of these techniques. In real conditions, one (or more) stations might not be working during a specific flood event, making a good prediction of the future flow height impossible. In cases like those of our study area, where the monitoring network is quite dense, this drawback is overcome by creating several models based on the use of different stations. We could apply the naivest procedure by inserting the missing values of an unworking station with 0 mm (rainfall) or 0 m (hydrometric height). However, in theory, this procedure might introduce greater errors in the flood events simulation, and we could test this by changing the input data of random groups of stations with values all equal to 0, simulating the case of the missing values for the event of S. Giovanni alla Vena station. We made a simulation as if 5, 50 and 95% of the rainfall and hydrometric stations used in the model for San Giovanni alla Vena were missing. These percentages correspond to 2 non-working stations, 22 non-working stations and 42 non-working stations respectively. For each of the three tests, we simulated the predictions from 0 to 24 hrs with a step of 1 h. We repeated these simulations 10 times changing the random group of stations each time.

We calculated these errors on the 30 highest events that occurred in 2019 and 2020 (test dataset). The 30 events used to calculate the variations had a hydrometric height between 1.40 and 6.90 m. We computed the percentage variation as follows:

$$Percentage\ Variation = \frac{\overline{y_i} - y_i}{m_i} \cdot 100 \quad (1)$$

where $y_i$ is the absolute error calculated on the difference between the measured hydrometric heights ($m_i$) and the predicted hydrometric heights estimated by using the real dataset; $\overline{y_i}$ is the absolute error computed by using the dataset with the simulation of missing values. The absolute errors ($\overline{y_i}$ and $y_i$) are calculated as follows:

$$absolute\ error = m_{sim} - m_{ob} \quad (2)$$

where $m_{sim}$ is the result of the model and $m_{ob}$ is the observed measure. By merging equation 1 and equation 2, we can write the percentage variation as follows:

$$Percentage\ Variation = \frac{\overline{m_i^s} - m_i^s}{m_i} \cdot 100 \quad (3)$$

where $\overline{m_i^s}$ is the result of the model with the dataset modified with the missing value and $m_i^s$ is the result of the model with the original dataset. A reduction in percentage variation indicates a greater underestimation of the flow level compared to models with the data available. The percentage variation is influenced by the number of non-working stations (Figure 10). With 5% of non-working stations, the model errors increase by less than 5% for each time of prediction. A 5% underestimation corresponds to a variation of the estimated hydrometric height of about 0.07 – 0.35 m. When the number of non-working stations is 95%, the model errors increase with an underestimation of the flow level that varies with the forecast time. The worst cases are for the greater forecast times with errors even higher than 20% (Figure 10). A variation of 20% corresponds to a variation of the estimated hydrometric height of about 0.28 – 1.48 m. This test allows us to observe how models can experience small increases in error when the number of non-working stations is small, but also that these errors increase greatly when the number of non-functioning stations increases significantly. In physical models, a precipitation estimate can be given by one group of stations rather than another, with results that in most cases are not significantly different. In deep learning models, which are data-driven, each station acquires greater importance and its correct management and maintenance have a greater influence on the final results of the models. For the applicability of these models in alert systems for territorial management bodies, proper maintenance and management of the monitoring stations are necessary for correct flow modeling.

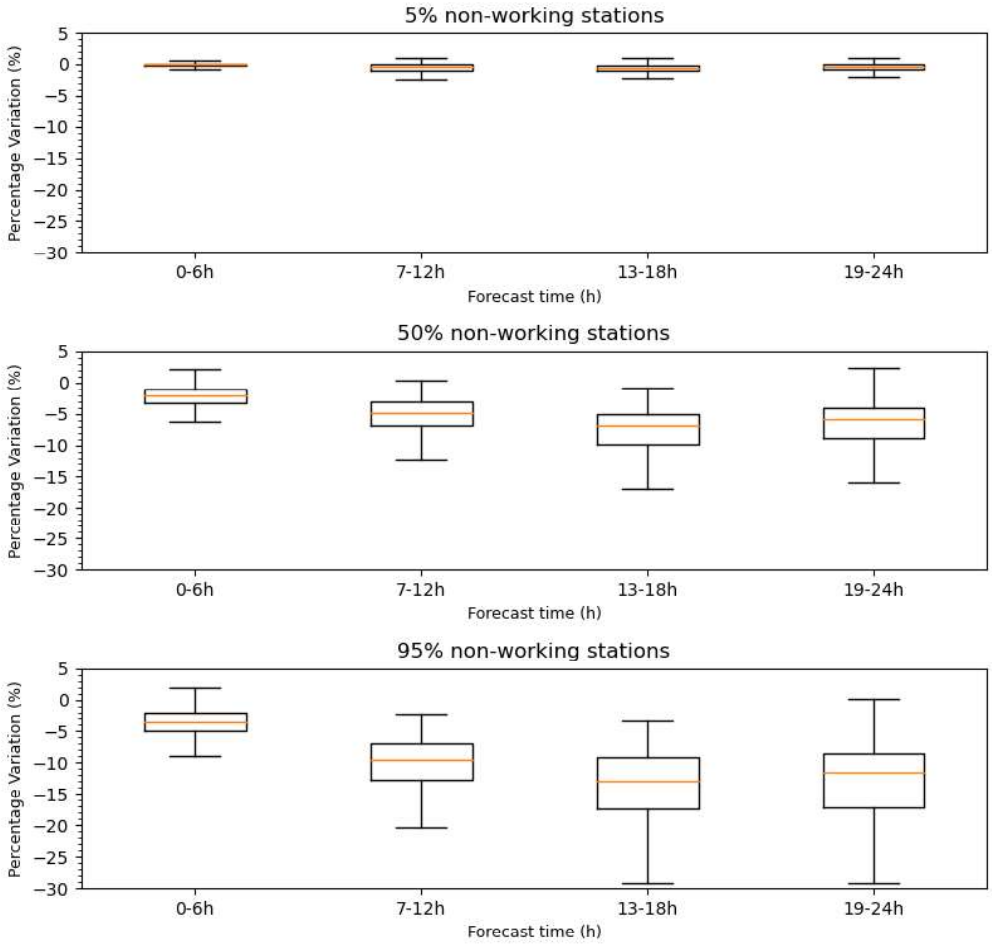



**Figure 10. Percentage variation in the absolute error estimate of the San Giovanni alla Vena station if we consider the 30 most serious events of 2019 and 2020 by simulating a variable number of non-working stations. The box indicates the range between the 25$^{th}$ and 75$^{th}$ percentiles (Q1 and Q3). IQR is the Q3-Q1 interquartile range. The upper whisker will extend to the last data lower than Q3 + 1.5 IQR. Likewise, the lower whisker will reach the first datum greater than Q1 - 1.5 IQR. The orange lines represent the medians.**

## 6. Conclusion

This study demonstrates that deep learning tools can be a viable alternative to physically-based models for the forecast of flood events in basins characterized by short run-off times. The study developed deep learning models based on the LSTM network for the Arno River, one of the most important and critical Italian rivers, by using eight hydrometric stations. Unlike physically-based models, these approaches offer the advantage of using only few types of data. This feature reduces model influences resulting from the accessibility of required information, which for some watersheds can be very difficult to obtain. This advantage makes it possible to create flood warning systems in situations where hydrographic and hydrogeological knowledge is very poor, making it very difficult to obtain additional information. Conversely, these methodologies can be applied in very complex geological and geomorphological situations (i.e., karst systems, steep slopes) where, despite the considerable knowledge of the territory, it is still impossible to build satisfactory physical models. Our case study falls precisely in this case. The Arno River basin was much studied in the past, and, in Italy, it is probably one of the most studied rivers together with the Po, the Tiber, and the Serchio. The alert model we have devised can be applied to different watercourses without having to deal with the study of the physics of the process. This makes possible to create a monitoring network for the simulation of the secondary channels as well.

However, this advantage can become a disadvantage if the time series is unreliable on account to the poor management of the station. Variations in the river section or a displacement of the measuring station can cause an inconsistency within the data that prevents these models from functioning correctly. This method allowed to obtain good results by exploiting an LSTM-based architecture. The errors in notification times found in this study are fully comparable with those obtained by other authors who used established physical models (Ercolani and Castelli, 2017). This comparison proved that our models are a valid tool compatible with others already used for flood forecasting. The greatest limit of our model is certainly the lack of information on the precipitation falling on the basin until the time of the forecast. If the forecast is issued before it starts raining in the basin, the model cannot predict the flood event. We think that future studies will improve these models by using precipitation forecast information extracted through artificial intelligence or physics-based techniques (physic-mathematical models, satellite image analysis). The inclusion of these techniques in the development of deep learning models can improve the prediction of a flood event.

**Supplementary Material:** Figures 1s to 10s show some simulations performed for the 10 major events that occurred in the Arno River basin in 2019 and 2020,

**Conflicts of Interest**: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript, or in the decision to publish the results.

**Acknowledgements:** The authors are grateful to the Tuscan Regional Hydrologic Service for providing the data used in this work.

## Reference

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever,

I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.

Antonetti, M., Zappa, M., 2018. How can expert knowledge increase the realism of conceptual hydrological models? A case study based on the concept of dominant runoff process in the Swiss Pre-Alps. Hydrology and Earth System Sciences 22, 4425–4447. https://doi.org/10.5194/hess-22-4425-2018

Autorità di Bacino del Fiume Arno, 1989. PIANO DI BACINO DEL FIUME ARNO.

Bates, B.C. (Bryson C.), Kundzewicz, Z., Wu, S., Palutikof, J., Intergovernmental Panel on Climate Change. Working Group II, 2008. Climate change and water.

Becchi I., 1986. Introduction to the Arno basin flooding problems, in: CNR-GNDCI pubbl. (Ed.), International Conference on the Arno Project.

Boulmaiz, T., Guermoui, M., Boutaghane, H., 2020. Impact of training data size on the LSTM performances for rainfall–runoff modeling. Modeling Earth Systems and Environment 6, 2153–2164. https://doi.org/10.1007/s40808-020-00830-w

Bryndal, T., Franczak, P., Kroczak, R., Cabaj, W., Kołodziej, A., 2017. The impact of extreme rainfall and flash floods on the flood risk management process and geomorphological changes in small Carpathian catchments: a case study of the Kasiniczanka river (Outer Carpathians, Poland). Natural Hazards 88, 95–120. https://doi.org/10.1007/s11069-017-2858-7

Caporali, E., Rinaldi, M., Casagli, N., 2005. The Arno River Floods. Giornale di Geologia Applicata 1, 177–192. https://doi.org/10.1474/GGA.2005-01.0-18.0018

Chattopadhyay, A., Nabizadeh, E., Hassanzadeh, P., 2020. Analog Forecasting of Extreme- Causing Weather Patterns Using Deep Learning. Journal of Advances in Modeling Earth Systems 12, e2019MS001958. https://doi.org/10.1029/2019MS001958

Chollet, F., 2015. Keras.

Ercolani, G., Castelli, F., 2017. Variational assimilation of streamflow data in distributed flood forecasting. Water Resources Research 53, 158–183. https://doi.org/10.1002/2016WR019208

Fawaz, H.I., Forestier, G., Weber, J., Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P., 2020. Deep learning for time series classification : a review To cite this version : HAL Id : hal-02365025 Deep learning for time series classification : a review.

Gaume, E., Bain, V., Bernardara, P., Newinger, O., Barbuc, M., Bateman, A., Blaškovičová, L., Blöschl, G., Borga, M., Dumitrescu, A., Daliakopoulos, I., Garcia, J., Irimescu, A., Kohnova, S., Koutroulis, A., Marchi, L., Matreata, S., Medina, V., Preciso, E., Sempere-Torres, D., Stancalie, G., Szolgay, J., Tsanis, I., Velasco, D., Viglione, A., 2009. A compilation of data on European flash floods. Journal of Hydrology 367, 70–78. https://doi.org/https://doi.org/10.1016/j.jhydrol.2008.12.028

Gaume, E., Borga, M., LLASSAT, M.C., Maouche, S., Lang, M., DIAKAKIS, M., 2016. Mediterranean extreme floods and flash floods , in: The Mediterranean Region under Climate Change. A Scientific Update, Coll. Synthèses. IRD Editions, pp. 133–144.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Hu, Y., Yan, L., Hang, T., Feng, J., 2020. Stream-Flow Forecasting of Small Rivers Based on LSTM.

IPCC, 2014. Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change.

Jaiswal, R.K., Ali, S., Bharti, B., 2020. Comparative evaluation of conceptual and physical rainfall–runoff models. Applied Water Science 10, 48. https://doi.org/10.1007/s13201-019-1122-6

Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018. Rainfall – runoff modelling using Long Short-Term Memory ( LSTM ) networks 6005–6022.

Le, X.H., Ho, H., Lee, G., Jung, S., 2019. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. Water 11, 1387. https://doi.org/10.3390/w11071387

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Li, W., Kiaghadi, A., Dawson, C., 2020. High temporal resolution rainfall–runoff modeling using long-short-term-memory (LSTM) networks. Neural Computing and Applications. https://doi.org/10.1007/s00521-020-05010-6

Liu, D., Jiang, W., Mu, L., Wang, S., 2020. Streamflow Prediction Using Deep Learning Neural Network: Case Study of Yangtze River. IEEE Access 8, 90069–90086. https://doi.org/10.1109/ACCESS.2020.2993874

Livieris, I.E., Pintelas, E., Pintelas, P., 2020. A CNN–LSTM model for gold price time-series forecasting. Neural Computing and Applications 32, 17351–17360. https://doi.org/10.1007/s00521-020-04867-x

Luppichini, M., Favalli, M., Isola, I., Nannipieri, L., Giannecchini, R., Bini, M., 2019. Influence of topographic resolution and accuracy on hydraulic channel flow simulations: Case study of the Versilia River (Italy). Remote Sensing 11. https://doi.org/10.3390/rs11131630

Marçais, J., de Dreuzy, J.-R., 2017. Prospective Interest of Deep Learning for Hydrological Inference. Groundwater 55, 688–692. https://doi.org/https://doi.org/10.1111/gwat.12557

Nguyen, D.H., Bae, D.-H., 2020. Correcting mean areal precipitation forecasts to improve urban flooding predictions by using long short-term memory network. Journal of Hydrology 584, 124710. https://doi.org/https://doi.org/10.1016/j.jhydrol.2020.124710

Sit, M., Demiray, B.Z., Xiang, Z., Ewing, G.J., Sermet, Y., Demir, I., 2020. A comprehensive review of deep learning applications in hydrology and water resources. Water Science and Technology. https://doi.org/10.2166/wst.2020.369

Sutskever, I., Vinyals, O., Le, Q. v., 2014. Sequence to Sequence Learning with Neural Networks.

Tien Bui, D., Hoang, N.-D., Martínez-Álvarez, F., Ngo, P.-T.T., Hoa, P.V., Pham, T.D., Samui, P., Costache, R., 2020. A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency

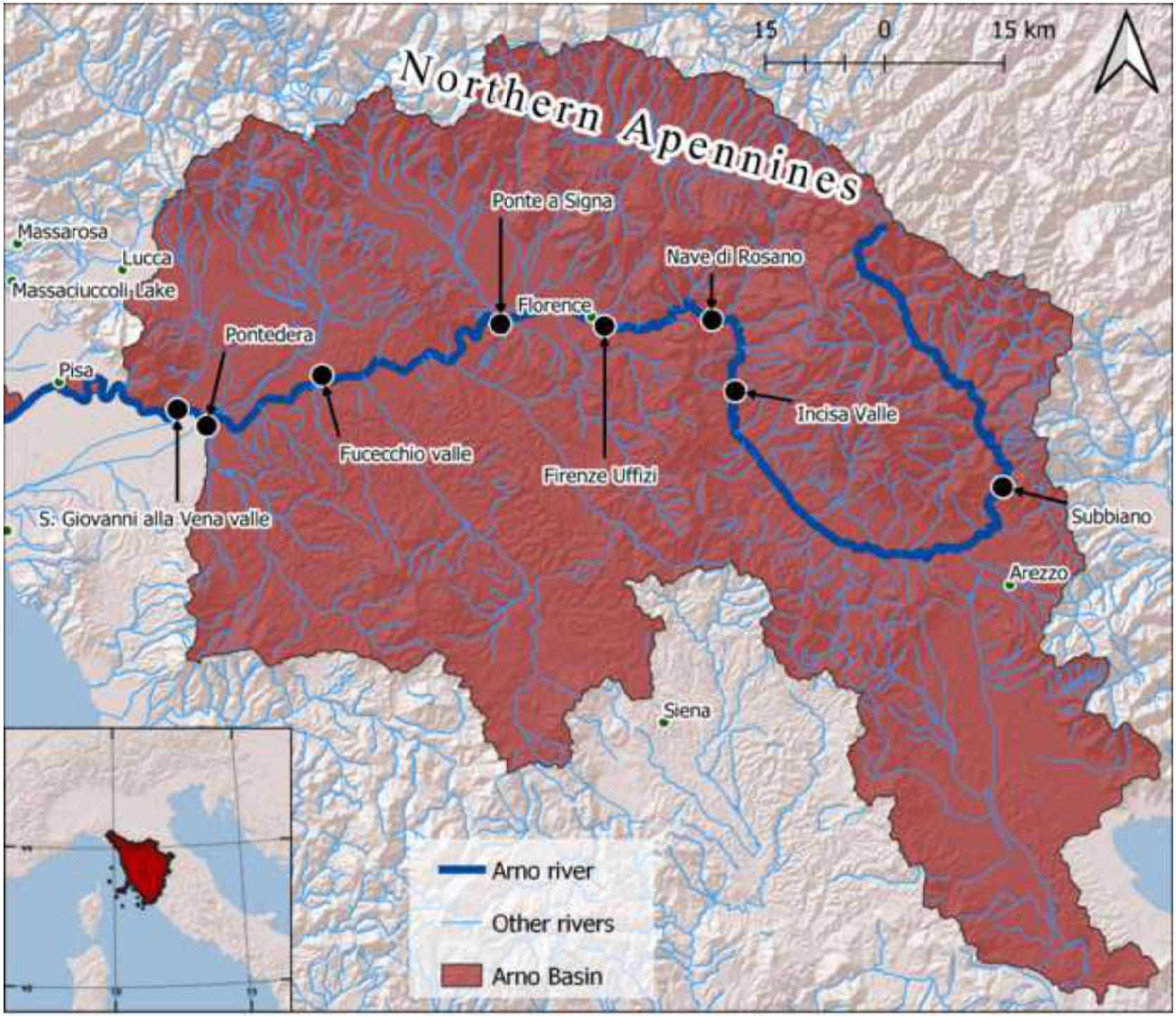
tropical storm area. Science of The Total Environment 701, 134413. https://doi.org/https://doi.org/10.1016/j.scitotenv.2019.134413

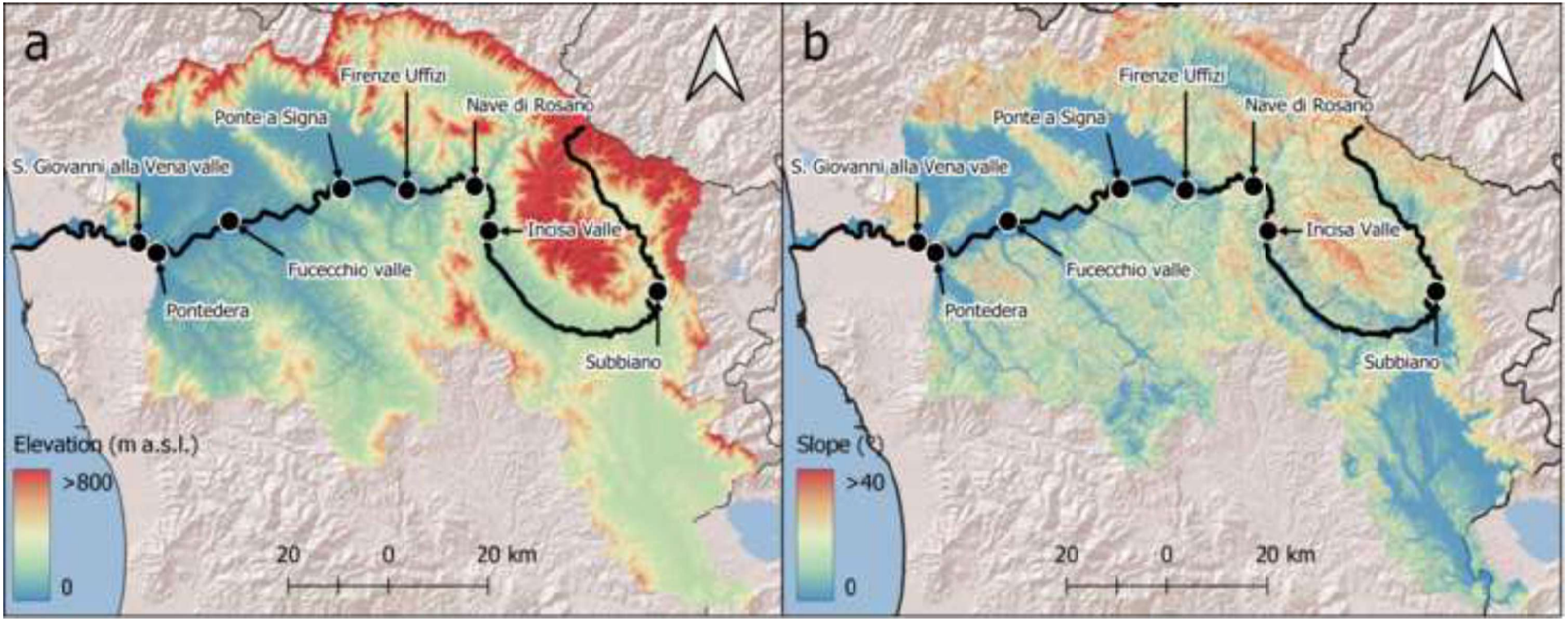
van Rossum, G., Drake, F.L., 2009. Python 3 Reference Manual. CreateSpace, Scotts Valley, CA.

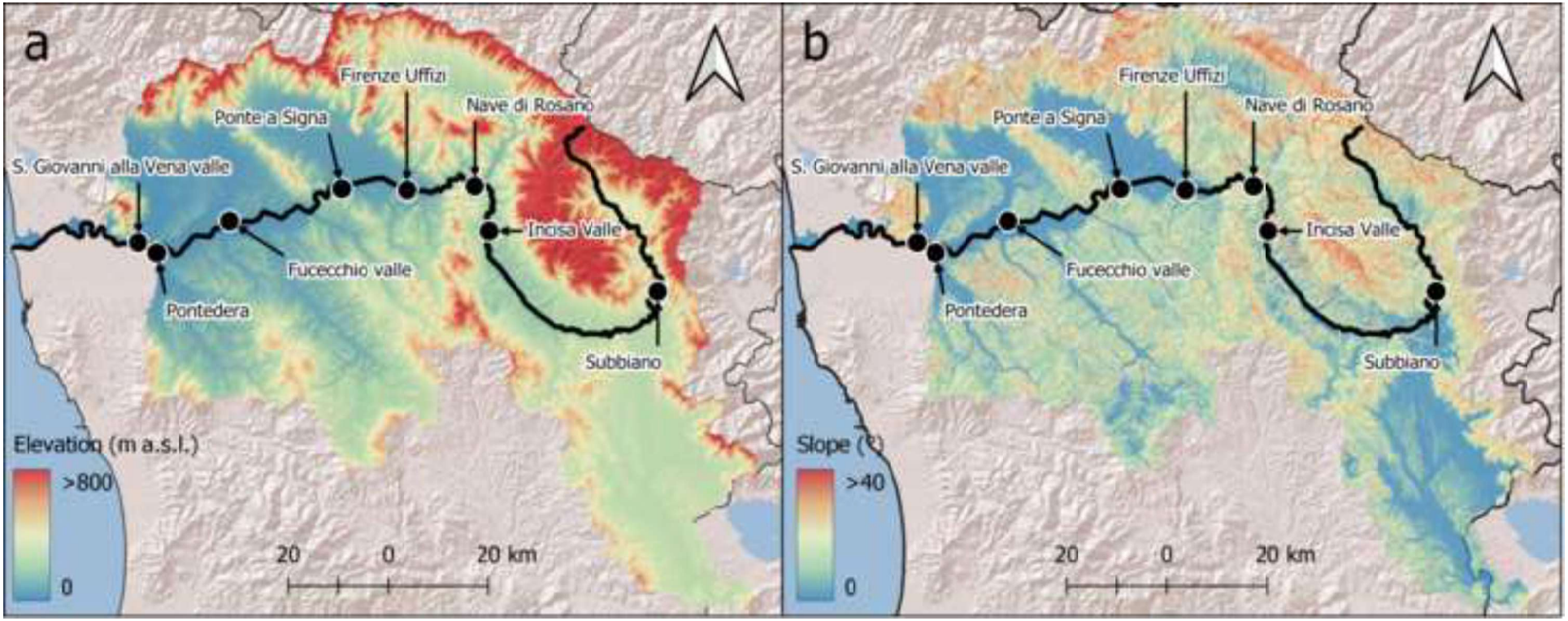
Van, S.P., Le, H.M., Thanh, D.V., Dang, T.D., Loc, H.H., Anh, D.T., 2020. Deep learning convolutional neural network in rainfall–runoff modelling. Journal of Hydroinformatics 22, 541–561. https://doi.org/10.2166/hydro.2020.095

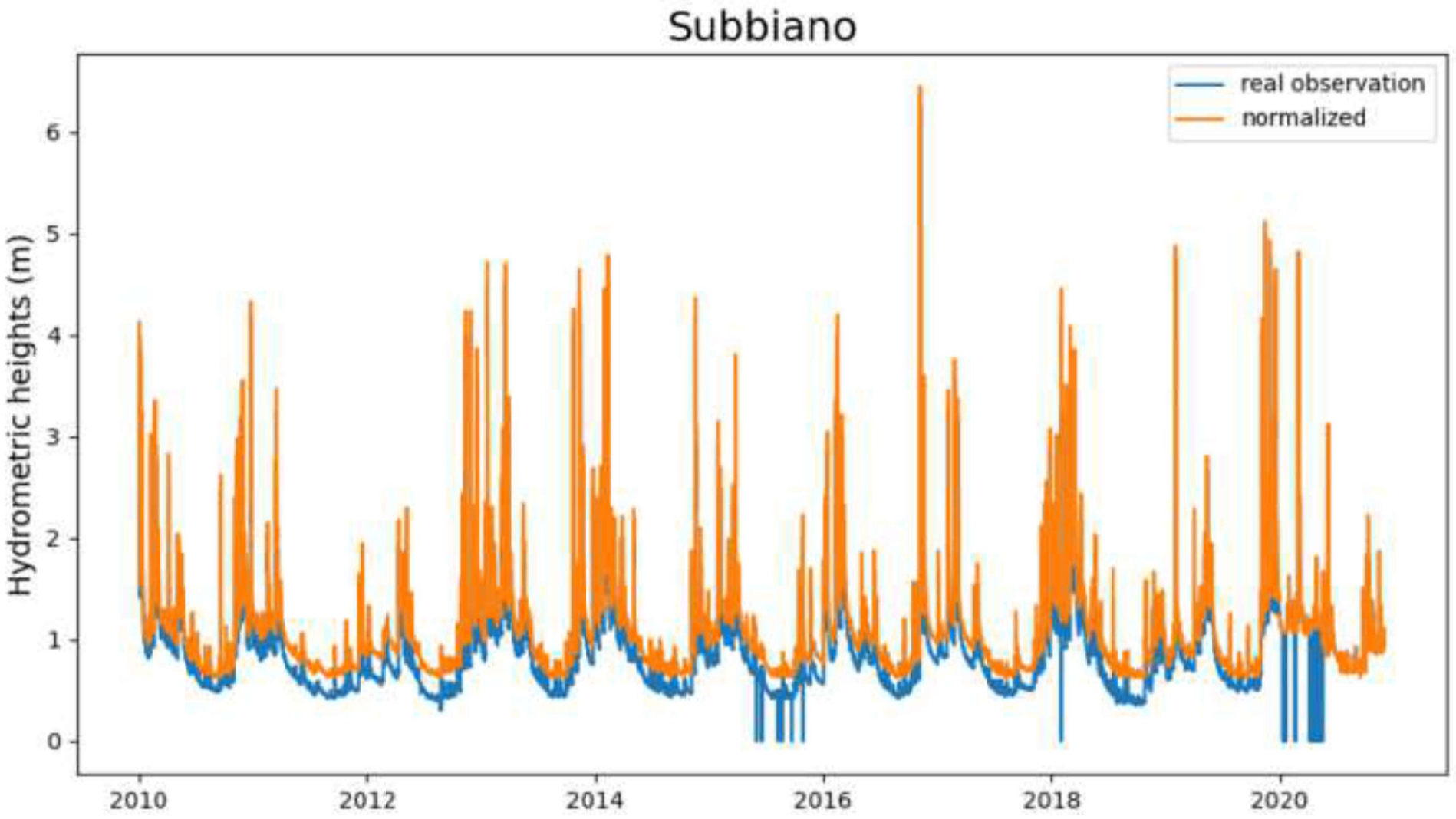
Yi, A., Li, Z., Gan, M., Zhang, Y., Yu, D., Chen, W., Ju, Y., 2019. A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system. Neural Computing and Applications 31, 1–13. https://doi.org/10.1007/s00521-018-3470-9

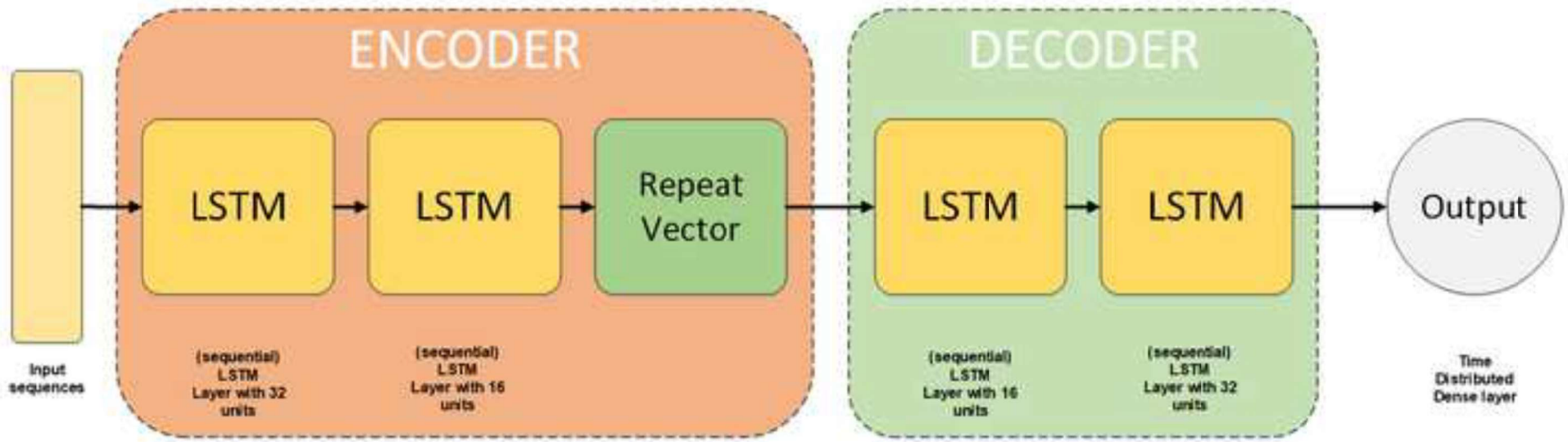
Zheng, J., Fu, X., Zhang, G., 2019. Research on Exchange Rate Forecasting Based on Deep Belief Network. Neural Comput. Appl. 31, 573–582. https://doi.org/10.1007/s00521-017-3039-z

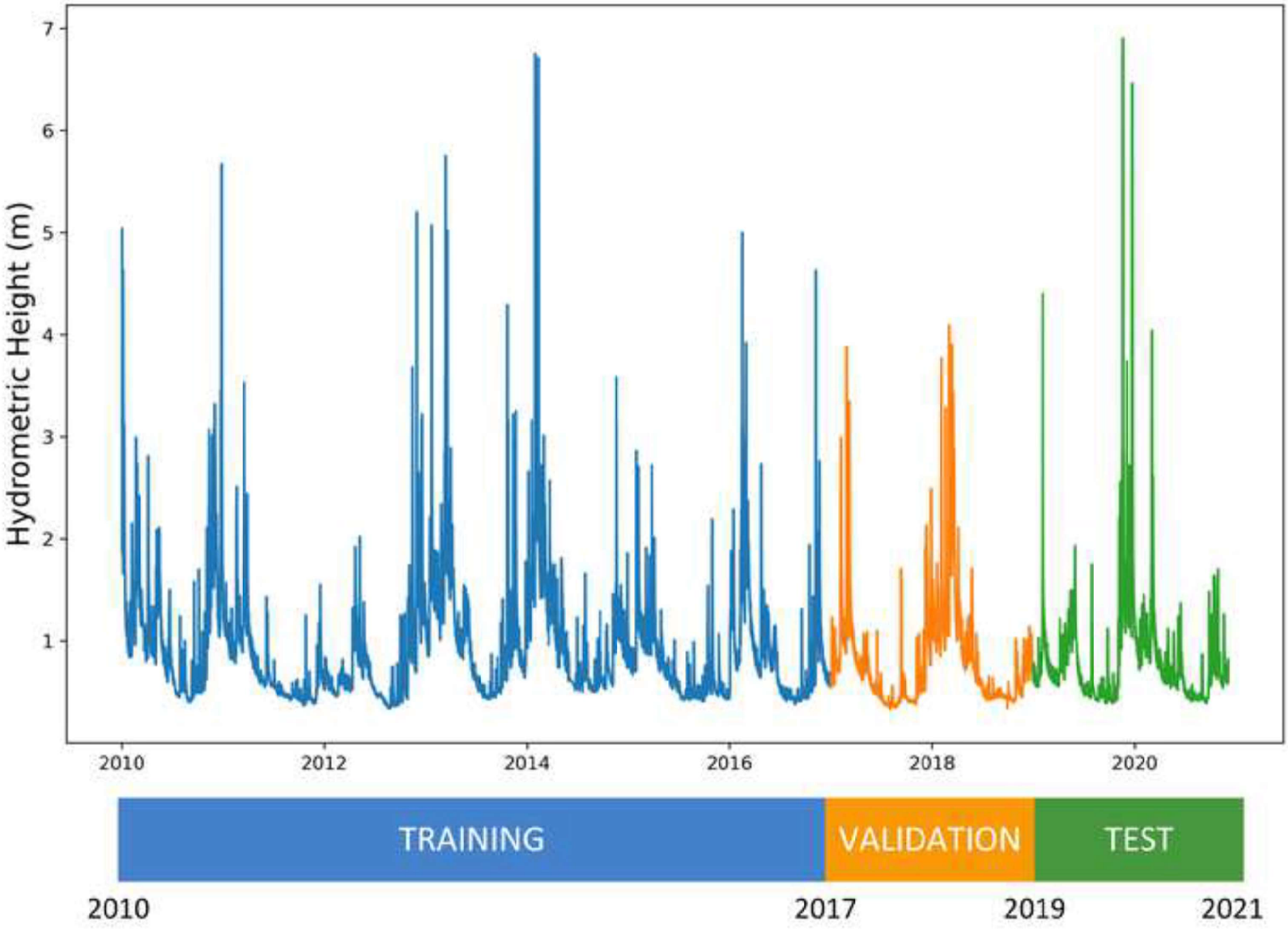


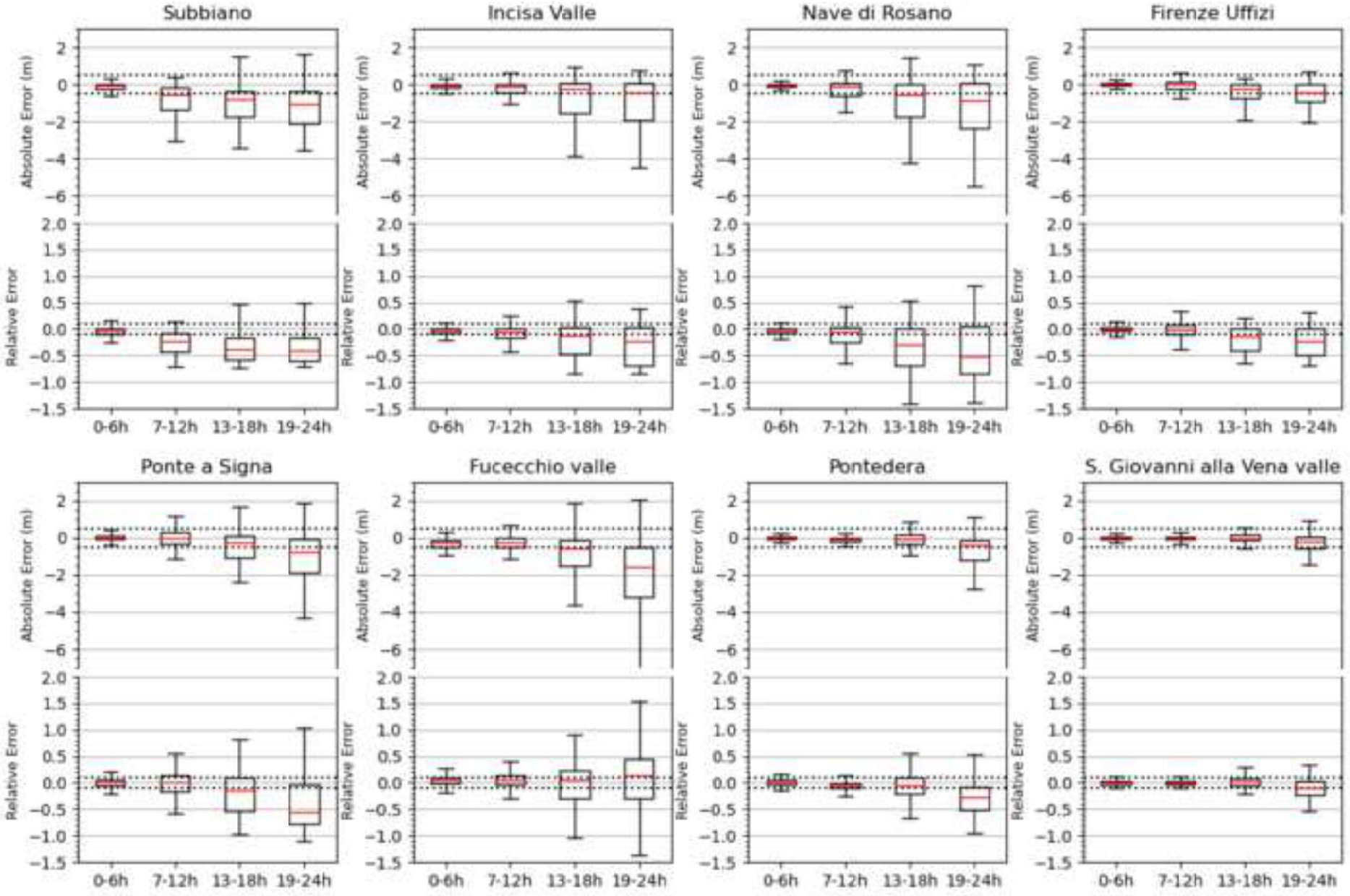


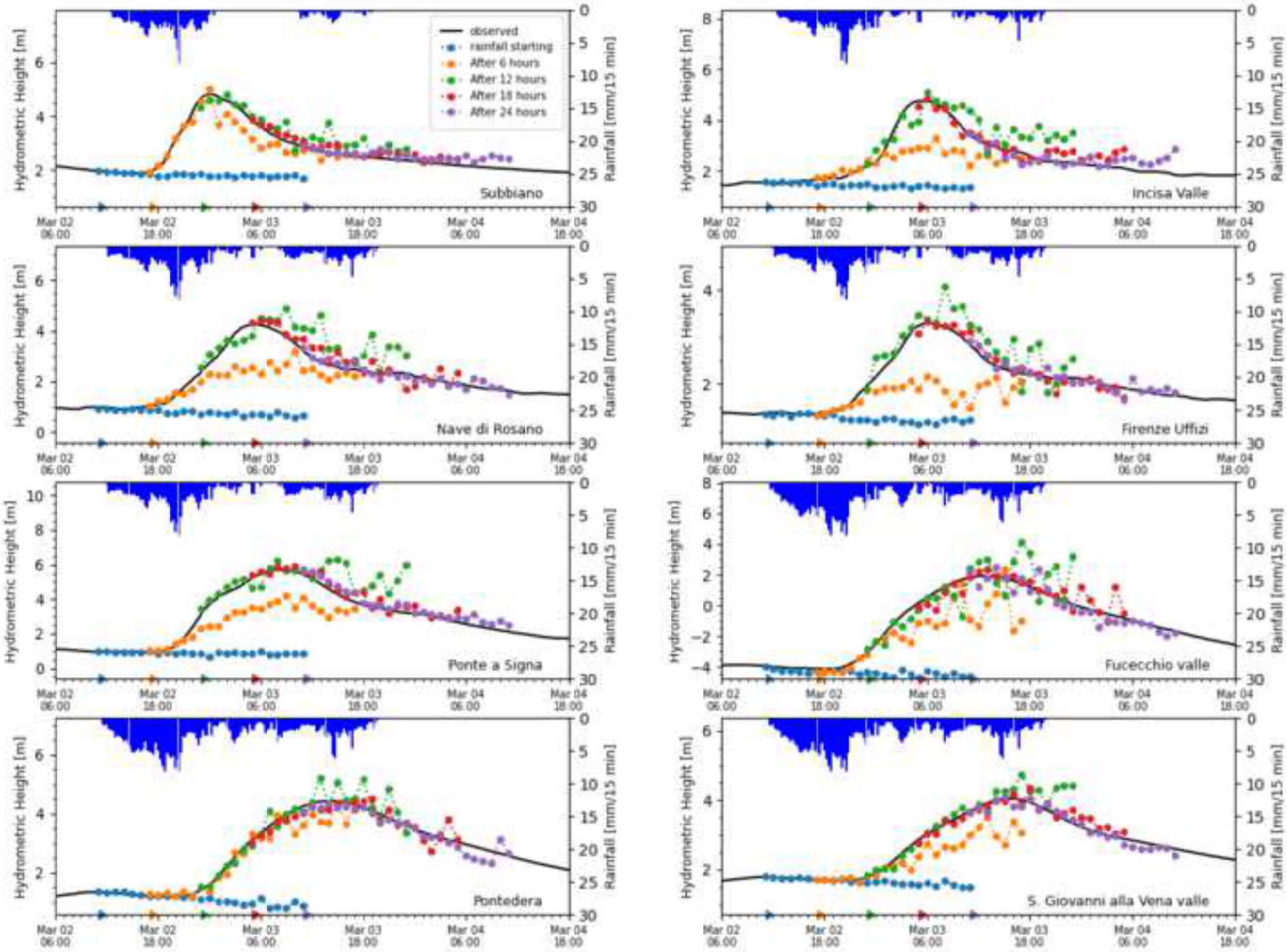


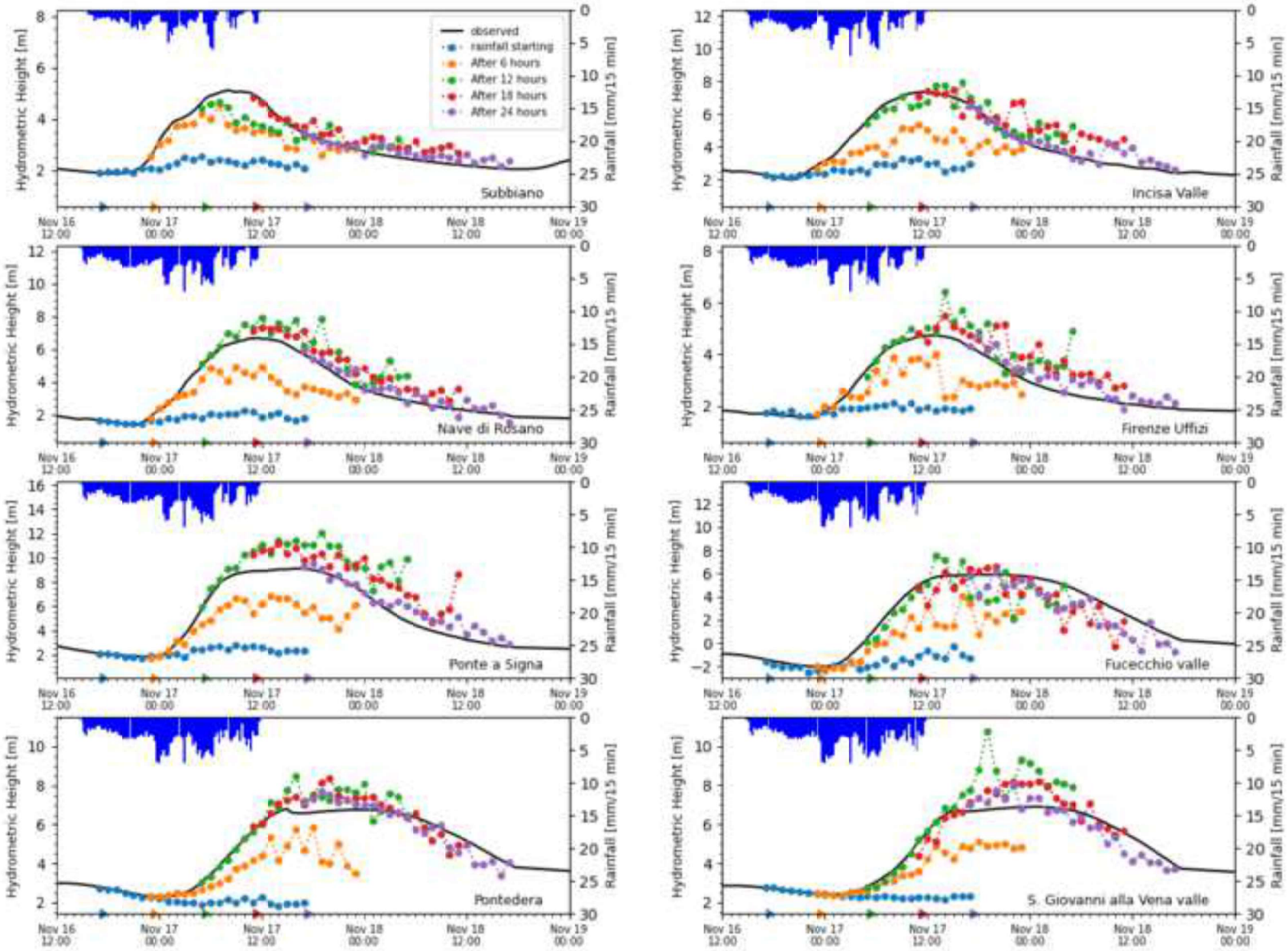












supplementary material

**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

**Conflicts of Interest**

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

**Marco Luppichini:** Conceptualization, Methodology, Software, Investigation, Data curation, Writing- Original draft preparation **Michele Barsanti** Conceptualization, Methodology , Investigation, Writing- Reviewing and Editing. **Roberto Giannecchini**: Supervision, Visualization, Writing- Reviewing and Editing. ***Monica Bini****:* Supervision, Writing- Reviewing and Editing, Conceptualization