

# Neural classification under discontinuous hydrometeorological data for flash flood early warning

Classificação neural com dados hidrometeorológicos descontínuos para alerta antecipado de enxurradas

Glauston Roberto Teixeira de Lima<sup>1</sup> , Pedro Ruiz Pereira Lopes<sup>2</sup> ,  
Rochane de Oliveira Caram<sup>1</sup> , Layrson de Jesus Menezes Gonçalves<sup>3</sup> 

<sup>1</sup>Centro Nacional de Monitoramento e Alertas de Desastres Naturais – Cemaden, São José dos Campos, SP, Brasil. E-mails: glauston.lima@cemaden.gov.br; rochane.caram@cemaden.gov.br

<sup>2</sup>Escola de Engenharia de Lorena, Universidade de São Paulo – USP, Lorena, SP, Brasil.  
E-mail: pedro\_lopes@usp.br

<sup>3</sup>Universidade Federal do Pará – UFPA, Belém, PA, Brasil. E-mail: layrson@ufpa.br

**How to cite:** Lima, G. B. T., Lopes, P. R. P., Caram, R. O., & Gonçalves, L. J. M. (2026). Neural classification under discontinuous hydrometeorological data for flash flood early warning. *Revista de Gestão de Água da América Latina*, 23, e17.  
<https://doi.org/10.21168/reg.v23e17>

**Abstract:** This study presents the development of a data-driven hydrological model for one-hour-ahead water-level forecasting using a Multi-Layer Perceptron (MLP) neural network. The model was trained with water level and rainfall data from 12 monitoring stations located in a small, fast-responding watershed prone to flash floods. Training data span the period from 2020 to 2023, while an independent dataset from 2024 was used for testing. The MLP was initially implemented as a regression model, but its performance was limited by gaps in historical rainfall records. To overcome this limitation, the problem was reformulated as a classification task, in which forecasted water levels were grouped into three hydrologically meaningful classes representing non-critical, intermediate, and alert conditions. A sequence of classification experiments was conducted to refine the model, including the explicit treatment of missing rainfall data and an alternative class-definition strategy incorporating antecedent water levels and rainfall accumulation to represent pre-alert conditions. The proposed classifier achieved stable performance and improved consistency in identifying alert situations, providing a robust and operationally meaningful framework for short-term flood early-warning systems in data-limited environments.

**Keywords:** data-driven hydrological modeling; neural networks; water level forecasting; flash flood early warning.

**Resumo:** Este estudo apresenta o desenvolvimento de um modelo hidrológico orientado por dados para a previsão do nível d'água com antecedência de uma hora, utilizando uma rede neural do tipo Perceptron Multicamadas. O modelo foi treinado com dados de nível d'água e precipitação provenientes de 12 estações de monitoramento localizadas em uma pequena bacia hidrográfica de resposta rápida, suscetível a inundações repentinas. Os dados de treinamento abrangem o período de 2020 a 2023, enquanto um conjunto de dados independente de 2024 foi utilizado para teste. Inicialmente, a MLP foi implementada como um modelo de regressão; contudo, seu desempenho mostrou-se limitado pela presença de lacunas nos registros históricos de precipitação. Para contornar essa limitação, o problema foi reformulado como uma tarefa de classificação, na qual os níveis d'água previstos foram agrupados em três classes hidrológicamente significativas, representando condições não críticas, intermediárias e de alerta. Uma sequência de experimentos de classificação foi conduzida para refinar o modelo, incluindo o tratamento explícito de dados de precipitação ausentes e uma estratégia alternativa de definição de classes que incorpora níveis d'água antecedentes e acúmulo de precipitação para representar condições de pré-alerta. O classificador proposto apresentou desempenho estável e maior consistência na identificação de situações de alerta, oferecendo um arcabouço robusto e operacionalmente relevante para sistemas de alerta antecipado de inundações em curto prazo em ambientes com dados limitados.

**Palavras-chave:** modelagem hidrológica orientada por dados; redes neurais; previsão de nível d'água; alerta antecipado de enxurradas.

## Introduction

In the context of climate change, flash floods are expected to become increasingly frequent (Zhang et al., 2021; Yin et al., 2023). Fortunately, the impacts of these hydrological extremes can be mitigated by issuing early warnings to civil defense agents responsible for preparing and executing aid plans for at-risk populations (Perera et al., 2020). The key question is whether warnings are issued with sufficient lead time and accuracy. This issue is particularly relevant because nowcasting flash floods involves an inherent trade-off between forecast lead time and accuracy. The triggering phenomenon, typically heavy rainfall, poses forecasting challenges due to its occurrence over small area with imprecise location, the presence of potential biases in rainfall predictions, and its short temporal development (Vivoni et al., 2007; Hill & Verjee 2010; Poletti et al., 2019; Zanchetta & Coulibaly 2020). To improve the reliability of water level forecasting, especially under uncertain rainfall conditions, hydrological models should be fed with rainfall data from a variety of sources - including radar, satellites, numerical weather forecasts, ground-based measurements, and even bias-corrected forecasts - so as to reduce the influence of individual errors and biases.

In highly dynamic environments such as fast-responding watersheds, data-driven hydrological models based on machine learning (ML) techniques are often preferable to physically based deterministic models. ML models have a greater ability to integrate information from multiple data sources and quickly adapt to new data. Additionally, their computational efficiency is crucial for generating forecasts rapidly and issuing timely warnings.

ML already plays a critical role in disaster risk management and reduction (Deparday et al., 2019; Sun et al., 2020; WMO 2022; Linardos et al., 2022), including support for early warning systems for flash floods (Acosta-Coll et al., 2018; Al-Rawas et al., 2024). For this specific application, artificial neural networks (ANNs) have gained significant attention as a modeling approach for water level forecasting. Studies by Kanbua & Khetchaturat (2007), Sušanjan et al. (2016), Alasali et al. (2021), Lima & Scofield (2021), and Lima et al. (2023) have successfully employed Multi-Layer Perceptrons (MLPs) for this task. Further advancements have been made with Long Short-Term Memory (LSTM) neural networks, as demonstrated by Song et al. (2019), Zhao et al. (2022), and Jhong et al. (2024). Panahi et al. (2021) tested convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for spatial prediction and mapping of flash flood probability, comparing their performances. Additionally, hybrid models combining CNNs and LSTMs, such as those proposed by Oddo et al. (2024), have improved the representation of the spatiotemporal dynamics of flash flood-prone watersheds. Shahabi et al. (2021) explored Deep Belief Networks (DBNs) as an alternative for predicting flash flood susceptibility, highlighting their potential in this domain. Recent review articles (Alasali et al., 2021; Byaruhanga et al., 2024) confirm that over the past two decades, ANN-based models have become the predominant approach for water level prediction and flash flood warning issuance. These studies collectively underscore the efficacy of ANN-based methods in hydrological modeling and their critical role in developing robust early warning systems.

However, the development of reliable ML models for early warning systems may face several challenges, one of which is the discontinuity in the available training dataset, caused by equipment failures and data transmission issues at hydrometeorological stations. Such data gaps significantly reduce the amount of valid training data, potentially compromising model performance. Since many of these gaps are long, common preprocessing techniques, such as interpolation or carrying forward the last valid observation, are often inadequate. Moreover, these discontinuities pose a major obstacle to the use of deeper recurrent architectures, such as standard LSTMs (Hochreiter & Schmidhuber, 1997), which rely on continuous sequences of past data (Ghazi et al., 2019). This limitation has been documented in studies such as Che et al. (2016), who emphasize the need for RNN variants that explicitly model missingness patterns (e.g., GRU-D), and Park et al. (2023), who show that MLPs can be effectively adapted for long-gap imputation in environmental time series.

In this study, an ANN-based model was developed to forecast the water level of the Capivari Stream one hour ahead. This stream is located in the city of Campos do Jordão, southeastern Brazil, and is part of a small, urban, fast-responding watershed prone to flash floods. The study is part of an ongoing research initiative at the Brazilian National Center for Monitoring and Early Warning of Natural Disasters (Cemaden), which aims to develop hydrological models based on ML techniques to support Cemaden's situation room operators in issuing flash flood warnings.

The dataset available for developing the proposed model, however, contains significant temporal discontinuities. Given the challenges posed by these data gaps, a MLP was selected as the forecasting model. While often overlooked in favor of recurrent architectures for time-series prediction, MLPs offer advantages in handling fragmented datasets where temporal continuity cannot be guaranteed.

Despite these advantages, the application of MLP-based models to flash flood forecasting under discontinuous hydrometeorological data may also involve methodological considerations related to problem formulation, class definition, and operational usability. In particular, based on the perspective adopted in this study, it is hypothesized that regression-based approaches may become particularly sensitive when forecasted water levels lie close to critical thresholds, since relatively small numerical deviations can potentially lead to substantially different warning decisions. Related discussions on operational flood forecasting, event-oriented prediction strategies, and data-driven hydrological modelling can be found in Solomatine et al. (2008) and Zhang et al. (2023).

In this context, the present study explores an alternative formulation of the forecasting problem by progressively transitioning from regression to classification. Rather than focusing exclusively on point forecasts of water level, the proposed framework groups predictions into hydrologically meaningful classes corresponding to non-critical, intermediate, and alert conditions. This formulation allows the model to emphasize decision-relevant information, which is more directly aligned with the needs of operational early warning systems.

A key contribution of this study lies in the systematic evaluation of class definitions and input representations under discontinuous data conditions. Through a sequence of controlled experiments, the robustness of the classifier to missing rainfall observations is assessed, including an explicit treatment of data availability through binary masking. The results demonstrate that the proposed classification framework is largely insensitive to rainfall data gaps, indicating that the model captures physically meaningful hydrological patterns rather than artifacts introduced by missing data.

Finally, this study introduces an expanded definition of the alert class that incorporates near-threshold hydrological states characterized by elevated antecedent water levels and significant rainfall forcing. This approach enables the identification of pre-alert conditions that may warrant attention in real-time operations, even when forecasted water levels do not strictly exceed a predefined threshold. By integrating data-driven learning with hydrologically informed class definitions, the proposed framework offers a practical and conservative decision-support tool for flash flood early warning in small, fast-responding watersheds affected by data discontinuities.

In addition to this introduction, the article is organized as follows: Section 2 describes the study area and the dataset; Section 3 details the developed neural model and the experimental design; Section 4 presents and discusses the results; and finally, Section 5 offers the concluding remarks.

## Study area and database

### Study area

The municipality of Campos do Jordão is situated within a valley in the Serra da Mantiqueira massif, located in the southeastern region of Brazil. The topography of the area is highly irregular, with 85% of the municipality consisting of undulating terrain, 10% of mountain slopes, and 5% of escarpment areas. The width of the flat areas within the municipality does not exceed 500 meters, and the surrounding altitudes vary between approximately 500 and 2,000 meters, with an average elevation of around 1,400 meters. According to the Köppen climate classification (Köppen, 1918), Campos do Jordão's climate is classified as Cwb (temperate highland climate with dry winters and mild summers). The year can be divided into two distinct periods: summer (October to March) and winter (April to September). During these periods, the average accumulated rainfall, maximum temperatures, and minimum temperatures are approximately [198.2 mm, 22.1°C, 12.6°C] for summer and [62.7 mm, 19.2°C, 6.9°C] for winter, respectively (Instituto Nacional de Meteorologia, 2024).

The main drainage system that crosses the urban core of the municipality is the Capivari Stream, which is fed by several tributaries along its course. The Capivari Stream watershed is classified as small, with an area of approximately 20 km<sup>2</sup> and a time to peak flow of around 30 minutes. Urban development in Campos do Jordão began with settlements along the Capivari Stream (IG-SP, 2014). The pattern of urbanization in the drainage areas has led to a concentration of flash floods in the most densely populated and longest-occupied parts of the municipality, particularly in the central area, as evidenced by the high number of flash floods recorded in the more central neighborhoods (Zucherato et al., 2016).

The Capivari Stream watershed is one of the areas monitored by Cemaden due to its susceptibility to flash floods. To monitor the watershed and issue warnings, Cemaden employs three water level thresholds, referred to as the attention quota, alert quota, and overflow quota, with corresponding values of 1.60 m, 2.15 m, and 2.69 m, respectively. Figure 1, on the left, shows the location of the watershed in the southeastern region of Brazil, and on the right, an elevation map of the watershed is presented, indicating the positions of Cemaden's measurement stations from which the data used in this study were obtained.

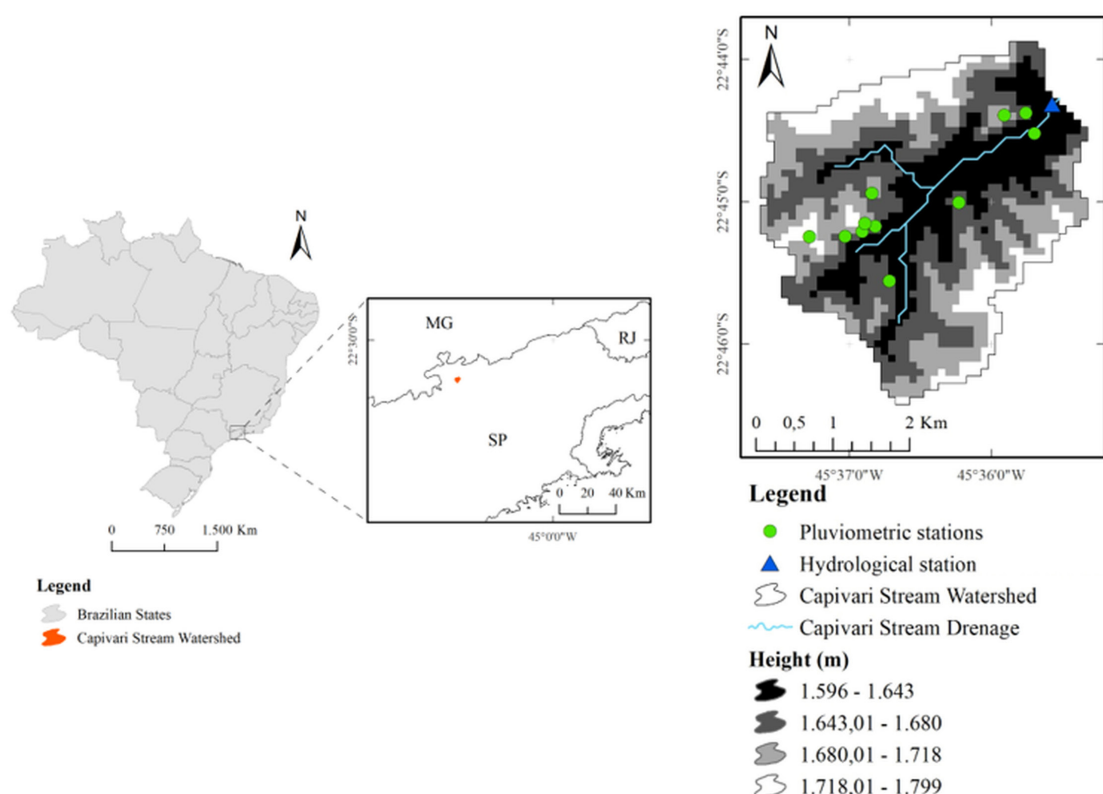


**Figure 1.** On the left: location of the Capivari Stream watershed in the southeastern region of Brazil; on the right: elevation map and location of the pluviometric stations (green circles) and the hydrological station (blue triangle).

(Source: Authors)

## Database

The dataset used in this study was obtained from 12 measurement stations within the Capivari Stream watershed, operated by Cemaden. Among these stations, 11 exclusively measure rainfall (represented as green circles in Figure 1), while one station, located at the watershed's outlet (represented as a blue triangle in Figure 1), measures both rainfall and water level. The historical series of rainfall and water level data spans from January 2020 to December 2023, with additional records from February 2024.

Rainfall and water level measurements are transmitted every 10 minutes during rainfall events and once per hour during dry periods. However, due to failures in the measurement circuits or data transmission systems, the historical dataset contains numerous gaps. In most cases, common preprocessing techniques such as interpolation or carrying forward the last valid value were not viable solutions due to the length and frequency of these discontinuities.

To construct the training dataset, hourly accumulated rainfall from each station and the last recorded water level from the previous hour were used as input variables, totaling 13 input variables. The target variable (expected output) was the water level recorded at the current hour. All training tuples were taken at hourly intervals, starting precisely at the beginning of each hour (e.g., between 1:00 PM and 1:59 PM). A schematic representation of this process is provided in Figure 2.

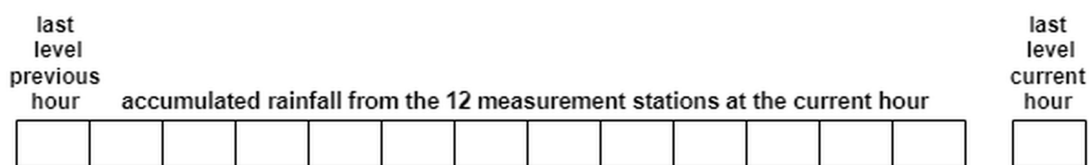


**Figure 2.** An illustration of the data tuples used in the neural network training.

(Source: Authors)

## Handling data discontinuities and selection criteria

The presence of data gaps in all measurement stations required a practical compromise between the number of available training tuples and the amount of valid information contained within each tuple. In this study, two limiting scenarios were considered to guide the selection strategy and illustrate this trade-off:

1. Strict filtering: Selecting only tuples from 1-hour intervals where all 12 stations had valid data would result in a severe reduction in the number of available training samples.
2. Relaxed filtering: Allowing tuples with very few valid rainfall measurements could introduce noise and insufficient information, potentially degrading the model's performance.

To mitigate these issues, a selection criterion was established. Only tuples from 1-hour intervals containing at least six valid rainfall measurements along with the two water level measurements were included in the training set. Missing rainfall values were set to zero, assuming no precipitation occurred. Applying this methodology, the training set (January 2020 – December 2023) comprised 22792 tuples, while the test set (February 2024) contained 572 tuples. Table 1 summarizes the main descriptive statistics of the rainfall and water level datasets from the monitoring stations, including the percentage of missing data, in order to characterize the observational dataset used in this study.

**Table 1.** Descriptive statistics for rainfall and water level data from the monitoring stations

| Station                                                    | Rainfall Data (mm/h) |         |      |      |      |       | Zero Rainfall (%) | Missing Data (%) |
|------------------------------------------------------------|----------------------|---------|------|------|------|-------|-------------------|------------------|
|                                                            | Mean                 | Std Dev | P25  | P50  | P75  | Max   |                   |                  |
| 350970012A                                                 | 5.06                 | 5.18    | 1.55 | 2.40 | 7.45 | 22.40 | 5.77              | 94.10            |
| 350970008A                                                 | 4.00                 | 3.62    | 1.60 | 2.40 | 5.35 | 17.00 | 41.15             | 58.22            |
| 350970002G                                                 | 4.54                 | 4.88    | 1.50 | 2.90 | 5.00 | 23.80 | 26.20             | 73.56            |
| 350970003G                                                 | 4.37                 | 4.90    | 1.40 | 2.50 | 5.20 | 40.40 | 83.42             | 14.45            |
| 350970004G                                                 | 4.11                 | 4.16    | 1.40 | 2.40 | 5.10 | 23.00 | 98.17             | 0.11             |
| 350970005G                                                 | 5.12                 | 5.48    | 1.30 | 3.20 | 6.80 | 22.80 | 5.80              | 94.07            |
| 350970003A                                                 | 4.63                 | 4.76    | 1.58 | 2.95 | 5.65 | 34.30 | 69.08             | 29.45            |
| 350970011A                                                 | 5.39                 | 5.84    | 1.60 | 3.00 | 6.65 | 38.20 | 47.54             | 51.42            |
| 350970001A                                                 | 4.53                 | 4.84    | 1.58 | 2.56 | 5.11 | 33.01 | 75.35             | 22.81            |
| 350970004A                                                 | 4.52                 | 4.62    | 1.57 | 2.56 | 5.90 | 24.71 | 88.09             | 11.09            |
| 350970001G                                                 | 4.47                 | 5.13    | 1.40 | 2.60 | 5.20 | 40.60 | 96.71             | 0.88             |
| 350970004H                                                 | 4.35                 | 5.01    | 1.40 | 2.40 | 5.00 | 39.60 | 96.21             | 1.01             |
| Water Level Data (m)                                       |                      |         |      |      |      |       |                   |                  |
| 350970004H                                                 | 0.49                 | 0.12    | 0.42 | 0.46 | 0.50 | 2.54  |                   |                  |
| P25 = 25th percentile; P50 = median; P75 = 75th percentile |                      |         |      |      |      |       |                   |                  |

Source: Authors

## Methodology

To forecast the water level of the Capivari stream, a fully connected feedforward MLP architecture (Haykin, 1998) with a single hidden layer was employed. The hidden layer used ELU activation functions with a relaxation parameter of 0.5. Predictions were carried out in both regression and classification modes, through four experiments detailed in the following subsections. In all experiments, water-level and rainfall values were linearly normalized to the  $[-1, 1]$  interval using predefined original ranges of  $[0, 5]$  m and  $[0, 100]$  mm, respectively. The number of hidden neurons was set equal to the number of input variables based on preliminary tests; the network weights were initialized with small random values; and training was performed using Stochastic Gradient Descent (SGD) as described by Bottou et al. (2018). The number of epochs was fixed at 1000 as the stopping criterion, and a small learning rate of 0.001 was adopted.

## Dataset Organization and Class Balancing Strategy

Before detailing the experiments, it is necessary to explain how the dataset was organized for the training phase of the neural network.

In this study, the 1.6 m threshold used in the operational monitoring context of the Capivari Stream watershed by Cemaden for flood warning activation was adopted as the reference level for class definition, consistent with previous studies conducted in the same study area (Lima & Scofield (2021); Lima et al., 2023). Accordingly, the neural network was trained to predict whether the stream level at the watershed outlet, based on a given input tuple, would exceed this threshold. This formulation reflects the model's intended role as a decision-support tool for early flash flood warnings.

Based on this threshold, the 22792 training tuples were initially divided into two groups: 22748 tuples in which the output values are below 1.6 m, and 44 tuples in which the output values exceed 1.6 m. To improve the network's ability to learn from the dominant class, the larger group (22748 tuples) was further subdivided into two subgroups: those with output values below 0.8 m and those with output values greater than or equal to 0.8 m. This process resulted in three distinct training classes:

- Class 1 ( $C_1$ ): 22349 tuples with output values  $< 0.8$  m
- Class 2 ( $C_2$ ): 399 tuples with output values  $\geq 0.8$  m and  $< 1.6$  m
- Class 3 ( $C_3$ ): 44 tuples with output values  $\geq 1.6$  m

In the test set, the distribution across  $C_1$ ,  $C_2$ , and  $C_3$  was 449, 119, and 4 tuples, respectively.

In each training iteration, batches were formed containing 15 tuples from  $C_1$ , 8 from  $C_2$ , and 4 from  $C_3$ . It is worth noting that, due to this batch composition, tuples from  $C_1$  were presented to the network only once during each training epoch, while tuples from  $C_2$  and  $C_3$  were repeated multiple times. This strategy was adopted to reduce the risk of the learning process becoming biased toward Class 1 due to its overwhelming prevalence in the dataset. The MLP was evaluated through four experiments, outlined below.

## Experiment 1

In this experiment, the network was trained to perform a regression task using the input tuples described in Section "Database".

The hidden layer consists of neurons with ELU activation functions defined as:

$$y = \begin{cases} \alpha(e^x - 1), & x < 0 \\ x, & x \geq 0 \end{cases} \quad (1)$$

where  $\alpha = 0.5$  in this study and  $x$  and  $y$  represent, respectively, the induced local field (pre-activations) and the activation of the hidden neurons.

The output neuron uses a sigmoid activation function given by:

$$y = \frac{5}{1 + e^{-x}} \quad (2)$$

Although the maximum observed water level in the historical series is well below 5 meters, the sigmoid amplitude was set to 5 in this experiment to allow for the possibility of higher levels in future overflow events.

The adopted cost function was the logarithm of the hyperbolic cosine (log-cosh), expressed as follows:

$$\log_{\cosh} = \frac{1}{N} \sum_{i=1}^N \log(\cosh(y_{oi} - y_{fi})) \quad (3)$$

where  $N$  is the number of observations,  $y_{oi}$  is the observed water level at the  $i$ -th time step, and  $y_{fi}$  is the corresponding forecasted value.

Figure 3 illustrates the schematic MLP architecture used in Experiment 1.

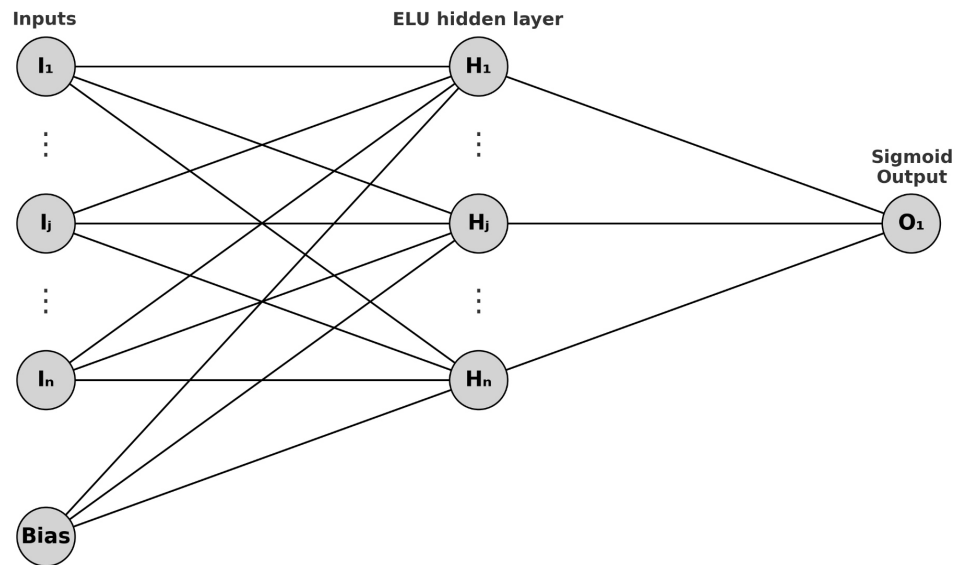


**Figure 3.** Schematic representation of the MLP architecture used in Experiment 1, with 13 input variables, a hidden layer with ELU activation ( $\alpha = 0.5$ ), a sigmoid output neuron, and a bias connection from input to hidden layer.

(Source: Authors)

## Experiment 2

In this experiment, the network was trained to classify the input tuples into three classes. The output layer consisted of three neurons with linear (identity) activation functions, and a softmax transformation (Equation 4) was applied to their outputs to convert them into class probability estimates.

$$\bar{y}_{ij} = \frac{e^{x_{ij}}}{\sum_{k=1}^3 e^{x_{ik}}} \quad (4)$$

where  $x_{ij}$  is the induced local field of the  $j$ -th output neuron for the  $i$ -th input tuple and  $\bar{y}_{ij}$  is the probability estimated by the neural network that this input belongs to the  $j$ -th class.

The cross-entropy (CE), defined in Equation 5, was used as the cost function to be minimized.

$$CE = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^3 y_{ij} \log(\bar{y}_{ij}) \quad (5)$$

where  $y_{ij}$  is the expected probability that the  $i$ -th input belongs to the  $j$ -th class and  $\bar{y}_{ij}$  is the predicted probability given by the softmax function.

Due to the characteristics of the available data, the expected probabilities  $y_{ij}$  used in the cross-entropy loss had to be defined in a binary form. The numerous temporal gaps in the data of the three classes, as well as the limited amount of data for classes  $C_2$  and  $C_3$ , made it very difficult to estimate the true probability distributions from which the datasets originated. Consequently, assigning smooth continuous membership probabilities to the input tuples was unfeasible. Therefore, binary expected probabilities were assigned instead. Specifically, input tuples from  $C_1$ ,  $C_2$ , and  $C_3$  were assigned expected probability vectors of  $[1 \ 0 \ 0]$ ,  $[0 \ 1 \ 0]$ , and  $[0 \ 0 \ 1]$ , respectively.

Prior to training, all input variables used in Experiment 2 were normalized using min-max scaling to ensure numerical stability during the optimization process. The normalization was applied independently to each input feature, including both antecedent water-level and rainfall variables, mapping their values to the  $[-1, 1]$  interval using predefined original ranges of  $[0, 5]$  m for water level and  $[0, 100]$  mm for rainfall. These upper bounds were intentionally set above the observed maxima to accommodate potentially more extreme future events. The same normalization parameters were subsequently applied to the test data.

Figure 4 illustrates the schematic MLP architecture employed in Experiment 2.

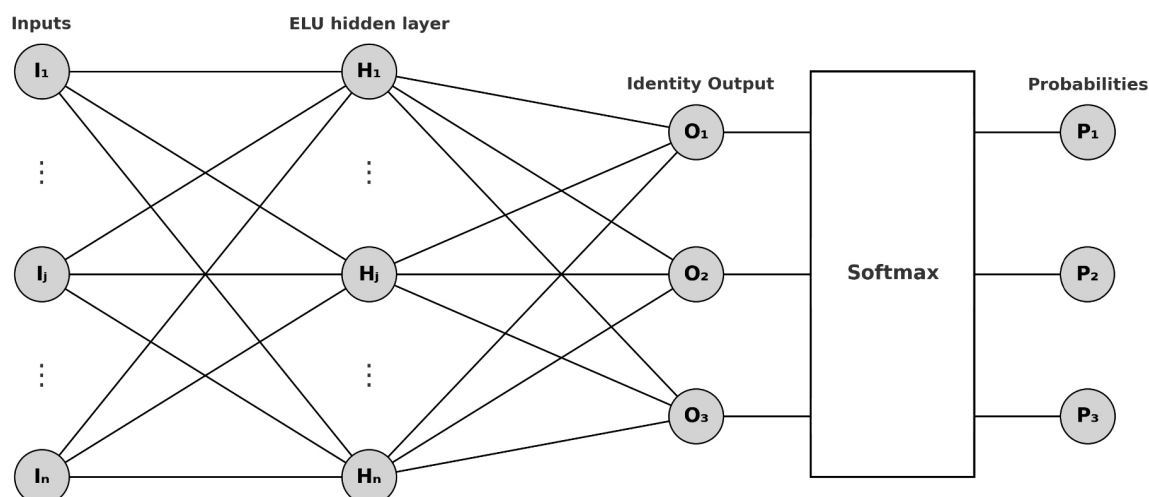


**Figure 4.** Schematic representation of the MLP architecture used in Experiment 2.  
 (Source: Authors)

### Experiment 3

In Experiment 3, an additional input representation was evaluated to investigate whether the false positives observed in Experiment 2, particularly cases in which samples from class C2 were misclassified as C3, could be attributed to the ambiguous encoding of missing rainfall observations. In the original dataset, missing rainfall measurements were represented as zero values, which are indistinguishable from true zero-rainfall observations from the perspective of the classifier.

To address this issue, a binary availability mask was appended to the input vector for each rainfall station. For each hourly rainfall input, a corresponding binary indicator was included, assuming a value of 1 when a valid observation was available and 0 when the value corresponded to a missing measurement (gap). Consequently, the dimensionality of the input vector was increased by adding one binary variable per rainfall station. The antecedent water-level input was left unchanged.

All physical input variables (antecedent water level and rainfall amounts) were normalized using the same normalization procedure adopted in Experiment 2. The binary mask variables were not normalized and were kept in their original  $\{0,1\}$  form. The same network architecture, training procedure, class definitions, and evaluation metrics used in Experiment 2 were retained to ensure a direct and fair comparison between the two experiments.

### Experiment 4

In Experiment 4, an alternative class-definition strategy was evaluated to address the persistent misclassification of samples from class  $C_2$  as  $C_3$  observed in Experiments 2 and 3. Rather than interpreting these misclassifications solely as model errors, this experiment explores the hypothesis that such samples may represent hydrologically meaningful near-critical conditions that are not adequately captured by a strict threshold-based definition of the alert class.

In the original class definition, samples were assigned to class  $C_3$  exclusively based on the predicted water level at time  $t$ , with values equal to or exceeding 1.6 m defining the alert condition. In Experiment 4, this definition was expanded by incorporating additional hydrometeorological criteria intended to capture pre-alert situations characterized by elevated antecedent water levels combined with significant rainfall forcing.

Specifically, samples originally belonging to class  $C_2$  were reassigned to class  $C_3$  when all of the following conditions were satisfied:

1. The antecedent water level at time  $t-1$  was greater than or equal to 0.8 m;
2. The predicted water level at time  $t$  satisfied  $\text{level}(t-1) < \text{level}(t) < 1.6\text{m}$ ;
3. The rainfall accumulation during the forecast interval exceeded a threshold  $T$ , defined as follows. For each sample originally assigned to class  $C_3$  in the training dataset, the average of

the three highest hourly rainfall values across the available stations was first computed. The threshold  $T$  was then defined as the median of these values across all  $C_3$  training samples, as given by Equation 6.

$$T = \text{median}\left(\frac{1}{3} \sum_{j=1}^3 R_j^i\right), i \in C_3 \quad (6)$$

where  $R_j^i$  are the three highest hourly rainfall values of sample  $i$ .

This rule-based reassignment resulted in an expanded alert class that includes both threshold-exceeding and near-threshold hydrological states with strong rainfall forcing. The reassignment was applied consistently to both training and test datasets, preserving the original temporal structure of the samples.

Following the application of the three criteria outlined above, the number of samples in class  $C_2$  was reduced to 392 and 116 in the training and test datasets, respectively, whereas class  $C_3$  comprised 51 and 7 samples.

The MLP architecture, training procedure, input normalization, and evaluation metrics adopted in Experiments 2 and 3 were retained, ensuring that differences in performance could be attributed exclusively to the modified class definition.

### Performance evaluation metrics

To evaluate the performance of the neural network in Experiments 1, the root mean square error (RMSE) and the mean absolute error (MAE) were calculated, as defined by Equations 7 and 8, respectively.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{oi} - y_{fi})^2} \quad (7)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_{oi} - y_{fi}| \quad (8)$$

The results of Experiments 2, 3, and 4 were analyzed using confusion matrices, and the classification performance of the neural network was assessed using the kappa index, as defined by Congalton & Green (2019), in Equation 9.

$$kappa = \frac{T \times \sum_{i=1}^k D_i - \sum_{i=1}^k (TC_i \times TR_i)}{T^2 - \sum_{i=1}^k (TC_i \times TR_i)} \quad (9)$$

where,  $T$  is the total number of elements in the confusion matrix,  $D_i$  is the value in the  $i$ -th diagonal position,  $TC_i$  is the sum of the  $i$ -th column,  $TR_i$  is the sum of the  $i$ -th row, and  $k$  is the number of classes.

In addition to these quantitative metrics, complementary analyses based on t-SNE visualizations and inter-class probability distance matrices were also performed to support the interpretation of the classification results obtained in Experiments 2 and 3.

## Results and discussion

### Results of Experiment 1

In this experiment, the class-wise log-cosh loss exhibited a smooth and consistent decline, stabilizing after approximately 600 epochs. The residual errors observed at convergence remained low and stable across all classes, indicating effective training and no signs of overfitting.

Figure 5 displays the water level forecasts generated by the neural network for the test sets of classes  $C_1$ ,  $C_2$ , and  $C_3$ , in comparison with the corresponding observed values.

Table 2 summarizes the RMSE and MAE metrics associated with the forecasts shown in Figure 5.

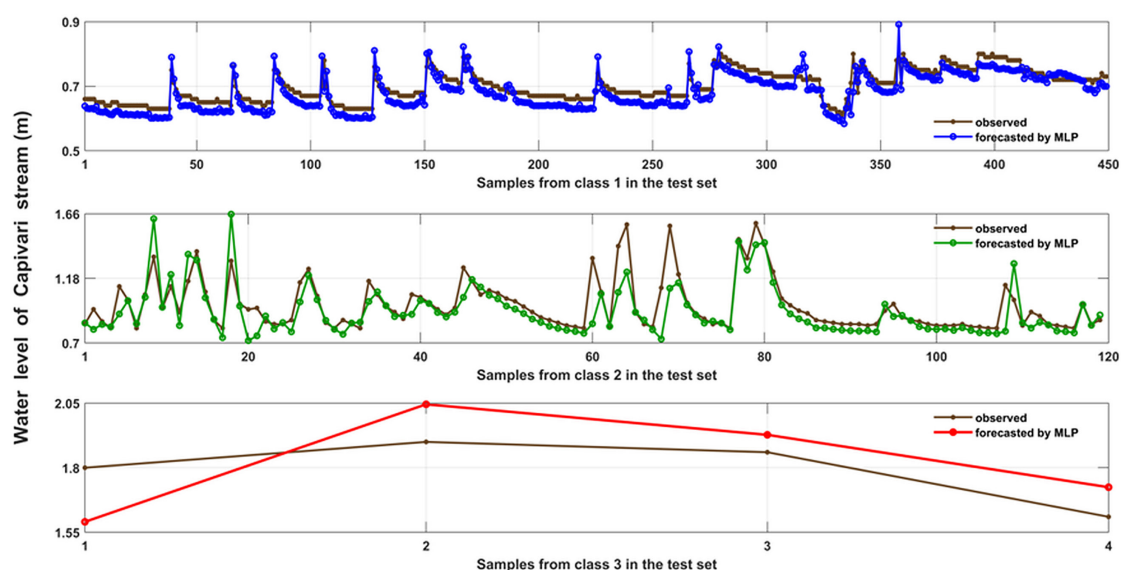


Figure 5. MLP-based water level forecasts versus observations for test samples from classes  $C_1$ ,  $C_2$ , and  $C_3$ . Each panel corresponds to one class and illustrates the model's generalization capability in the regression task.

**Table 2.** RMSE and MAE of the MLP-based regression model for the test samples of each class.

|          | $C_1$  | $C_2$  | $C_3$  |
|----------|--------|--------|--------|
| RMSE (m) | 0.0292 | 0.1216 | 0.1438 |
| MAE (m)  | 0.0265 | 0.0730 | 0.1342 |

## Results of Experiment 2

Tables 3 and 4 present the confusion matrices associated with the classification performance of the MLP in Experiment 2 for the training and test datasets, respectively. In both tables, the left matrix corresponds to the three-class classification scenario, in which tuples are categorized into classes  $C_1$ ,  $C_2$ , and  $C_3$ , while the right matrix represents the binary classification case, obtained by merging classes  $C_1$  and  $C_2$  into a single group. The primary objective of this experiment is to assess the ability of the model to discriminate between tuples with output values below 1.6 m and those equal to or exceeding this threshold, which defines the alert class ( $C_3$ ).

In the training phase, the MLP achieved a global accuracy of 0.9895 and a Kappa index of 0.7671. In the test phase, the global accuracy reached 0.9563, with a corresponding Kappa index of 0.8676.

**Table 3.** Confusion matrices for the MLP based classification of the training data: three-class (left) and binary (right) scenarios. Kappa values are reported below each matrix.

|               | Three classes |       |       | Two classes   |           |       |
|---------------|---------------|-------|-------|---------------|-----------|-------|
|               | $C_1$         | $C_2$ | $C_3$ |               | $C_1/C_2$ | $C_3$ |
| $C_1$         | 22147         | 202   | 0     | $C_1/C_2$     | 22737     | 11    |
| $C_2$         | 27            | 361   | 11    | $C_3$         | 2         | 42    |
| $C_3$         | 0             | 2     | 42    |               |           |       |
| kappa: 0.7671 |               |       |       | kappa: 0.8657 |           |       |

**Table 4.** Confusion matrices for the MLP based classification of the test data: three-class (left) and binary (right) scenarios. Kappa values are reported below each matrix.

| Three classes  |                |                |                | Two classes |                                 |                |   |
|----------------|----------------|----------------|----------------|-------------|---------------------------------|----------------|---|
|                | C <sub>1</sub> | C <sub>2</sub> | C <sub>3</sub> |             | C <sub>1</sub> / C <sub>2</sub> | C <sub>3</sub> |   |
| C <sub>1</sub> | 436            | 13             | 0              |             | C <sub>1</sub> / C <sub>2</sub> | 566            | 2 |
| C <sub>2</sub> | 11             | 106            | 2              |             | C <sub>3</sub>                  | 0              | 4 |
| C <sub>3</sub> | 0              | 0              | 4              |             |                                 |                |   |
| kappa: 0.8676  |                |                |                |             | kappa: 0.7983                   |                |   |

### Results of Experiment 3

The confusion matrices obtained for Experiment 3 in the training and test datasets are presented in Tables 5 and 6, respectively. In the training phase, the MLP achieved a global accuracy of 0.9899 and a Kappa index of 0.7768. In the test phase, the global accuracy was 0.9528 and the corresponding Kappa index reached 0.8665.

**Table 5.** Confusion matrices for the MLP based classification of the training data: three-class (left) and binary (right) scenarios. Kappa values are reported below each matrix.

| Three classes  |                |                |                | Two classes                     |                                 |                |
|----------------|----------------|----------------|----------------|---------------------------------|---------------------------------|----------------|
|                | C <sub>1</sub> | C <sub>2</sub> | C <sub>3</sub> |                                 | C <sub>1</sub> / C <sub>2</sub> | C <sub>3</sub> |
| C <sub>1</sub> | 22158          | 191            | 0              | C <sub>1</sub> / C <sub>2</sub> | 22736                           | 12             |
| C <sub>2</sub> | 25             | 362            | 12             | C <sub>3</sub>                  | 2                               | 42             |
| C <sub>3</sub> | 0              | 2              | 42             |                                 |                                 |                |
| kappa: 0.7768  |                |                |                |                                 | kappa: 0.8568                   |                |

**Table 6.** Confusion matrices for the MLP based classification of the test data: three-class (left) and binary (right) scenarios. Kappa values are reported below each matrix.

| Three classes  |                |                |                | Two classes                     |                                 |                |
|----------------|----------------|----------------|----------------|---------------------------------|---------------------------------|----------------|
|                | C <sub>1</sub> | C <sub>2</sub> | C <sub>3</sub> |                                 | C <sub>1</sub> / C <sub>2</sub> | C <sub>3</sub> |
| C <sub>1</sub> | 431            | 18             | 0              | C <sub>1</sub> / C <sub>2</sub> | 565                             | 3              |
| C <sub>2</sub> | 6              | 110            | 3              | C <sub>3</sub>                  | 0                               | 4              |
| C <sub>3</sub> | 0              | 0              | 4              |                                 |                                 |                |
| kappa: 0.8665  |                |                |                |                                 | kappa: 0.7248                   |                |

### Results of Experiment 4

The confusion matrices obtained for Experiment 4 in the training and test datasets are presented in Tables 7 and 8, respectively. In the training phase, the MLP achieved a global accuracy of 0.9889 and a Kappa index of 0.7669. In the test phase, the global accuracy was 0.9528, with a corresponding Kappa index of 0.8681.

**Table 7.** Confusion matrices for the MLP based classification of the training data: three-class (left) and binary (right) scenarios. Kappa values are reported below each matrix.

| Three classes |       |       |       | Two classes   |             |       |
|---------------|-------|-------|-------|---------------|-------------|-------|
|               | $C_1$ | $C_2$ | $C_3$ |               | $C_1 / C_2$ | $C_3$ |
| $C_1$         | 22111 | 238   | 0     | $C_1 / C_2$   | 22741       | 0     |
| $C_2$         | 15    | 377   | 0     | $C_3$         | 0           | 51    |
| $C_3$         | 0     | 0     | 51    |               |             |       |
| kappa: 0.7669 |       |       |       | kappa: 1.0000 |             |       |

**Table 8.** Confusion matrices for the MLP based classification of the test data: three-class (left) and binary (right) scenarios. Kappa values are reported below each matrix.

| Three classes |       |       |       | Two classes   |             |       |
|---------------|-------|-------|-------|---------------|-------------|-------|
|               | $C_1$ | $C_2$ | $C_3$ |               | $C_1 / C_2$ | $C_3$ |
| $C_1$         | 428   | 21    | 0     | $C_1 / C_2$   | 565         | 0     |
| $C_2$         | 6     | 110   | 0     | $C_3$         | 0           | 7     |
| $C_3$         | 0     | 0     | 7     |               |             |       |
| kappa: 0.8681 |       |       |       | kappa: 1.0000 |             |       |

## Discussion

The neural network forecasts obtained in Experiment 1 were effective for class  $C_1$  but only moderately accurate for  $C_2$  and  $C_3$ , as shown by the plots in Figure 5 and the performance metrics in Table 2. Notably, for some tuples from classes  $C_2$  and  $C_3$ , the model produced forecasts close to the 1.6 m threshold, even when the observed water levels were not near this limit. These cases are important because they indicate a risk of both false positives and false negatives, which can compromise the reliability of flood warnings based on the model's outputs, a challenge commonly discussed in operational flood early-warning systems (Martina et al., 2006; Tsai et al., 2022).

Nevertheless, although the regression results were not entirely satisfactory, the network appeared to have learned the output value ranges associated with each class (i.e., output < 0.8 for  $C_1$ ,  $0.8 \leq \text{output} < 1.6$  for  $C_2$ , and output  $\geq 1.6$  for  $C_3$ ) with reasonable accuracy. In fact, the number of predicted outputs falling outside their true class ranges was low, and most of the few exceptions deviated only slightly. This behavior is typical of a classifier, suggesting that explicitly training the model to classify inputs into predefined output intervals could be a more effective strategy. Similar behavior has been reported in previous studies involving machine learning approaches for hydrological forecasting and warning systems (Muñoz et al., 2021). This finding motivated the development of the classification models evaluated in Experiment 2.

The classification results obtained in Experiment 2 showed that the MLP is able to identify the alert class ( $C_3$ ) with high reliability, particularly in the test dataset, in which all samples with output values equal to or exceeding 1.6 m were correctly classified. In the training dataset, most  $C_3$  samples were also correctly identified, although two instances were misclassified as  $C_2$ . These misclassifications occurred for samples whose output values were close to the alert threshold, indicating that the model exhibited greater uncertainty near class boundaries rather than producing random errors, a behavior frequently observed in classification problems involving near-threshold hydrological conditions.

A similar behavior was observed in Experiment 3, in which an explicit availability mask was introduced to distinguish between observed zero rainfall and missing data. While the classification performance remained high, two  $C_3$  samples in the training dataset were again misclassified as  $C_2$ , whereas all  $C_3$  samples in the test dataset were correctly identified. The persistence of this behavior, despite the explicit encoding of missing rainfall information, suggests that the ambiguity associated with data gaps was not the primary cause of misclassification near the  $C_2$ - $C_3$  boundary. Instead, these errors appear to be linked to near-threshold hydrological conditions that are intrinsically difficult to separate using a strict output-based class definition.

The consistency of these near-boundary misclassifications motivated the alternative class-definition strategy evaluated in Experiment 4. Rather than treating all  $C_2$ -to- $C_3$  or  $C_3$ -to- $C_2$  misclassifications as undesirable errors, this experiment explored the hypothesis that some samples originally assigned to  $C_2$  represent physically meaningful pre-alert conditions. By incorporating additional hydrometeorological criteria—namely, elevated antecedent water levels and significant rainfall accumulation—into the definition of the alert class, Experiment 4 expanded  $C_3$  to include near-threshold states with strong rainfall forcing. Under this revised definition, all  $C_3$  samples were correctly classified in both training and test datasets, while the overall classification performance remained stable.

From an operational standpoint, the results of Experiments 2–4 indicate that the MLP-based classifier does not merely replicate a rigid threshold-based decision rule. Instead, it learns a representation of hydrological risk that integrates both antecedent conditions and rainfall forcing, which is consistent with the broader use of machine learning techniques to represent complex hydrological responses in flood forecasting and warning systems (Muñoz et al., 2021). The misclassifications observed in Experiments 2 and 3 can thus be interpreted as conservative predictions in situations characterized by rapidly evolving or near-critical hydrological states. By explicitly accounting for these conditions in the class definition, Experiment 4 provides a classification framework that is better aligned with the requirements of flood early-warning systems, in which the cost of missed alerts often exceeds that of occasional false alarms (Martina et al., 2006).

### Concluding remarks

This study presented the development of MLP-based neural network models to forecast, with one-hour lead time, the water level in the Capivari watershed, an area monitored by Cemaden due to its susceptibility to flash floods.

The models were trained with historical rainfall and water level data recorded between 2020 and 2023, and tested with data from February 2024, obtained from 12 monitoring stations installed in the watershed. These models are expected to be used as decision-support tools by Cemaden's operational staff for monitoring the watershed and issuing early warnings.

The initial regression-based formulation highlighted important limitations associated with data gaps in the historical rainfall records, which affected the reliability of near-threshold forecasts. However, the analysis also revealed that the neural network was able to learn meaningful output ranges associated with different hydrological states. This observation motivated the reformulation of the problem as a classification task, in which forecasted water levels were grouped into hydrologically relevant classes corresponding to non-critical, intermediate, and alert conditions.

A sequence of classification experiments was conducted to progressively refine the proposed framework. The results showed that the MLP-based classifier is able to reliably identify alert situations, particularly in the independent test dataset, while maintaining stable overall performance. The explicit treatment of missing rainfall data through availability masks demonstrated that classification performance is largely insensitive to data gaps, indicating that the observed misclassifications near the alert threshold are not primarily driven by data quality issues. Instead, these cases reflect near-critical hydrological conditions characterized by elevated antecedent water levels combined with rainfall forcing.

By incorporating these conditions into an expanded definition of the alert class, the final experiment provided a classification scheme that better represents pre-alert states of practical relevance for flood early-warning applications. This approach preserves the simplicity of threshold-based decision rules while enhancing their flexibility and robustness under uncertain and rapidly evolving conditions.

Overall, the proposed MLP-based classification framework offers a practical and operationally meaningful solution for short-term water-level forecasting in small, fast-responding watersheds with incomplete data. The results support its potential integration into real-time early-warning systems, contributing to more reliable and conservative flood risk management in data-limited environments.

Although the neural models developed in this study were trained and tested exclusively using historical data and are therefore not yet operational, the results support their potential as a viable tool for early flood warning applications. Ongoing research at Cemaden is currently focused on generating rainfall estimates for the Capivari watershed through a neural-network-based downscaling scheme driven by forecasts from the WRF mesoscale model. These rainfall estimates are expected to replace the observed rainfall data used in the models proposed here, enabling the implementation and real-time evaluation of an operational version of the system.

## Acknowledgments:

P. R. P. L. acknowledges financial support from the Brazilian National Council for Scientific and Technological Development (CNPq) through a graduate scholarship (Grant No. 174682/2023-9; Process No. 800108/2022-1).

## References

- Acosta-Coll, M., Ballester-Merelo, F., Martinez-Peiró, M., & De la Hoz-Franco, E. (2018). Real-time early warning system design for pluvial flash floods: a review. *Sensors (Basel)*, 18(7), 2255. PMID:30002358. <https://doi.org/10.3390/s18072255>
- Alasali, F., Tawalbeh, R., Ghanem, Z., Mohammad, F., & Alghazzawi, M. (2021). A sustainable early warning system using rolling forecasts based on ANN and golden ratio optimization methods to accurately predict real-time water levels and flash flood. *Sensors (Basel)*, 21(13), 4598. PMID:34283144. <https://doi.org/10.3390/s21134598>
- Al-Rawas, G., Nikoo, M. R., Al-Wardy, M., & Etri, T. (2024). A critical review of emerging technologies for flash flood prediction: examining artificial intelligence, machine learning, internet of things, cloud computing, and robotics techniques. *Water (Basel)*, 16(14), 2069. <https://doi.org/10.3390/w16142069>
- Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, 60(2), 223-311. <https://doi.org/10.1137/16M1080173>
- Byaruhanga, N., Kibirige, D., Gokool, S., & Mkhonta, G. (2024). Evolution of flood prediction and forecasting models for flood early warning systems: a scoping review. *Water (Basel)*, 16(13), 1763. <https://doi.org/10.3390/w16131763>
- Che, Z., Purushotham, S., Cho, K., Sontag, D. & Liu, Y. (2016). *Recurrent Neural Networks for multivariate time series with missing values*. *Scientific Reports*, 8, 6085.
- Congalton, R. G., & Green, K. (2019). *Assessing the accuracy of remotely sensed data: principles and practices*. CRC press. <https://doi.org/10.1201/9780429052729>.
- Deparday, V., Gevaert, C. M., Molinaro, G., Soden, R., & Balog-Way, S. (2019). *Machine Learning for Disaster Risk Management*. GFDRR - World Bank. Retrieved in 2024, July 25, from <http://documents.worldbank.org/curated/en/503591547666118137/Machine-Learning-for-Disaster-Risk-Management>
- Ghazi, M. M., Nielsen, M., Pai, A., Cardoso, M. J., Modat, M., & Ourselin, S. (2019). Training recurrent neural networks robust to incomplete data: application to Alzheimer's disease progression modeling. *Medical Image Analysis*, 53, 39-46. PMID:30682584. <https://doi.org/10.1016/j.media.2019.01.004>
- Haykin, S. S. (1998). *Neural networks: a comprehensive foundation*. Upper Saddle River: Prentice Hall.
- Hill, C., & Verjee, F. (2010). Flash flood early warning system reference guide 2010. *UCAR*. Retrieved in 2024, July 25, from [https://www.meted.ucar.edu/sign\\_in.php?go\\_back\\_to=/communities/hazwarnsys/ffewsrsg/FF\\_EWS.pdf](https://www.meted.ucar.edu/sign_in.php?go_back_to=/communities/hazwarnsys/ffewsrsg/FF_EWS.pdf)
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780. PMID:9377276. <https://doi.org/10.1162/neco.1997.9.8.1735>
- IG-SP. (2014). *Mapeamento de riscos associados a escorregamentos, inundações, erosão e solapamento de margens e drenagens. Relatório Técnico Município Campos do Jordão (Vol. II)*. Retrieved in 2024, July 25, from <https://www.unisdr.org/campaign/resilientcities/uploads/city/attachments/3240-10316.pdf>
- Instituto Nacional de Meteorologia - INMET. (2024). *Normais Climatológicas do Brasil 1981-2010*. Retrieved in 2024, July 25, from <https://portal.inmet.gov.br/normais>
- Jhong, Y. D., Chen, C. S., Jhong, B. C., Tsai, C. H., & Yang, S. Y. (2024). Optimization of LSTM parameters for flash flood forecasting using genetic algorithm. *Water Resources Management*, 38(3), 1141-1164. <https://doi.org/10.1007/s11269-023-03713-8>
- Kanbua, W., & Khetchaturat, C. (2007). Decision support system for flash flood warning management using artificial neural network. In *The 11th Annual National Symposium on Computational Science and Engineering (ANSCSE11)* (pp. 28-30). Retrieved in 2024, July 25, from <http://www.marine.tmd.go.th/Presentation/TMD/TMD-paper/full-0102-back-real.pdf>
- Köppen, W. (1918). *Climatologia: con un estudio de los climas de la tierra*. Mexico-DF: Fondo de Cultura Economica.
- Lima, G. R. T., & Scofield, G. B. (2021). Feasibility study on operational use of neural networks in a flash flood early warning system. *RBRH*, 26, e7. <https://doi.org/10.1590/2318-0331.262120200152>
- Lima, G. R. T., Caram, R. O., & Filho, L. F. A. (2023). Use of neural network as a support tool in water level forecasting and issuing flash floods early warnings to three small Brazilian urban watersheds. *Earth Science Informatics*, 16(4), 4313-4326. <https://doi.org/10.1007/s12145-023-01159-5>
- Linardos, V., Drakaki, M., Tzionas, P., & Karnavas, Y. L. (2022). Machine Learning in Disaster Management: Recent Developments in Methods and Applications. *Machine Learning and Knowledge Extraction*, 4(2), 446-473. <https://doi.org/10.3390/make4020020>
- Martina, M. L. V., Todini, E., & Libralon, A. (2006). A Bayesian decision approach to rainfall thresholds based flood warning. *Hydrology and Earth System Sciences*, 10(3), 413-426. <https://doi.org/10.5194/hess-10-413-2006>
- Muñoz, P., Orellana-Alvear, J., Bendix, J., Feyen, J., & Céleri, R. (2021). Flood early warning systems using machine learning techniques: the case of the tomebamba catchment at the Southern Andes of Ecuador. *Hydrology*, 8(4), 183. <https://doi.org/10.3390/hydrology8040183>

- Oddo, P. C., Bolten, J. D., Kumar, S. V., & Cleary, B. (2024). Deep Convolutional LSTM for improved flash flood prediction. *Frontiers in Water*, 6, 1346104. <https://doi.org/10.3389/frwa.2024.1346104>
- Panahi, M., Jaafari, A., Shirzadi, A., Shahabi, H., Rahmati, O., Omidvar, E., & Bui, D. T. (2021). Deep learning neural networks for spatially explicit prediction of flash flood probability. *Geoscience Frontiers*, 12(3), 101076. <https://doi.org/10.1016/j.gsf.2020.09.007>
- Park, J., Muller, J., Arora, B., Faybishenko, B., Pastorello, G., Varadharajan, C., Sahu, R., & Agarwal, D. (2023). Longterm missing value imputation for time series data using deep neural networks. *Neural Computing and Applications*, 35, 9071-9091. <https://doi.org/10.1007/s00521-022-08165-6>
- Perera, D., Seidou, O., Agnihotri, J., Mahmood, H., & Rasmy, M. (2020). *Challenges and Technical Advances in Flood Early Warning Systems (FEWSs)*. IntechOpen. <https://doi.org/10.5772/intechopen.93069>
- Poletti, M. L., Silvestro, F., Davolio, S., Pignone, F., & Rebora, N. (2019). Using nowcasting technique and data assimilation in a meteorological model to improve very short range hydrological forecasts. *Hydrology and Earth System Sciences*, 23(9), 3823-3841. <https://doi.org/10.5194/hess-23-3823-2019>
- Shahabi, H., Shirzadi, A., Ronoud, S., Asadi, S., Pham, B. T., Mansouripour, F., & Bui, D. T. (2021). Flash flood susceptibility mapping using a novel deep learning model based on deep belief network, back propagation and genetic algorithm. *Geoscience Frontiers*, 12(3), 101100. <https://doi.org/10.1016/j.gsf.2020.10.007>
- Solomatine, D. P., Maskey, M., & Shrestha, D. L. (2008). Instance-based learning compared to other data-driven methods in hydrological forecasting. *Hydrological Processes*, 22(2), 275-287. <https://doi.org/10.1002/hyp.6592>
- Song, T., Ding, W., Wu, J., Liu, H., Zhou, H., & Chu, J. (2019). Flash flood forecasting based on long short-term memory networks. *Water (Basel)*, 12(1), 109. <https://doi.org/10.3390/w12010109>
- Sun, W., Bocchini, P., & Davison, B. D. (2020). Applications of artificial intelligence for disaster management. *Natural Hazards*, 103(3), 2631-2689. <https://doi.org/10.1007/s11069-020-04124-3>
- Sušanj, I., Ožanić, N., & Marović, I. (2016). Methodology for developing hydrological models based on an artificial neural network to establish an early warning system in small catchments. *Advances in Meteorology*, 1, 9125219. <https://doi.org/10.1155/2016/9125219>
- Tsai, Y. L., Irvin, J., Chundi, S., Ng, A. Y., Field, C. B., & Kitanidis, P. K. (2022). Improving debris flow evacuation alerts in Taiwan using machine learning. *Research Square*. arXiv:2208.13027. arXiv preprint. <https://doi.org/10.21203/rs.3.rs-2235385/v1/>
- Vivoni, E. R., Entekhabi, D., & Hoffman, R. N. (2007). Error Propagation of Radar Rainfall Nowcasting Fields through a Fully Distributed Flood Forecasting Model. *Journal of Applied Meteorology and Climatology*, 46(6), 932-940. <https://doi.org/10.1175/JAM2506.1>
- World Meteorological Organization - WMO. (2022). *Bulletin*, 71(1). Retrieved in 2024, July 25, from <https://library.wmo.int/records/item/46869-vol-71-1-2022?offset=6> accessed on 07/17/2024
- Yin, J., Gao, Y., Chen, R., Yu, D., Wilby, R., Wright, N., & Guan, M. (2023). Flash floods: why are more of them devastating the world's driest regions? *Nature*, 615(7951), 212-215. PMID:36882611. <https://doi.org/10.1038/d41586-023-00626-9>
- Zanchetta, A. D. L., & Coulibaly, P. (2020). Recent Advances in Real-Time Pluvial Flash Flood Forecasting. *Water (Basel)*, 12(2), 570. <https://doi.org/10.3390/w12020570>
- Zhang, Y., Pan, D., Van Griensven, J., Yang, S. X., & Gharabaghi, B. (2023). Intelligent flood forecasting and warning: a survey. *Intell. Robot.*, 3(2), 190-212. <https://doi.org/10.20517/ir.2023.12>
- Zhang, Y., Wang, Y., Chen, Y., Xu, Y., Zhang, G., Lin, Q., & Luo, R. (2021). Projection of changes in flash flood occurrence under climate change at tourist attractions. *Journal of Hydrology (Amsterdam)*, 595, 126039. <https://doi.org/10.1016/j.jhydrol.2021.126039>
- Zhao, G., Liu, R., Yang, M., Tu, T., Ma, M., Hong, Y., & Wang, X. (2022). Large-scale flash flood warning in China using deep learning. *Journal of Hydrology (Amsterdam)*, 604, 127222. <https://doi.org/10.1016/j.jhydrol.2021.127222>
- Zucherato, B., Cunha, L., & Freitas, M. I. C. (2016). Análise das áreas de Inundação de Campos do Jordão SP: caracterização por meio de bancos de dados. *Territórios de Água*, 162-170. Retrieved in 2024, July 25, from [https://www.researchgate.net/publication/309282372\\_Analise\\_das\\_areas\\_de\\_inundacao\\_de\\_campos\\_do\\_Jordao\\_-\\_SP\\_Characterizacao\\_atraves\\_de\\_banco\\_de\\_dados](https://www.researchgate.net/publication/309282372_Analise_das_areas_de_inundacao_de_campos_do_Jordao_-_SP_Characterizacao_atraves_de_banco_de_dados)

#### Author contributions:

Glauston Roberto Teixeira de Lima: contributed to data acquisition (download) and organization of the datasets; proposed neural modeling approaches and supervised their implementation; conducted the literature review and wrote most of the manuscript; reviewed the final version of the manuscript.

Pedro Ruiz Pereira Lopes: carried out the computational implementation of the algorithms and the simulations of the different neural network approaches; contributed to result analysis and algorithm refinement.

Rochane de Oliveira Caram: contributed to the preparation of figures and to manuscript review.

Layrson de Jesus Menezes Gonçalves: provided technical support during the retrieval of part of the dataset and its storage in formats suitable for neural network applications.