Flood Early Warning Systems using Machine Learning Techniques. Application to a Catchment located in the Tropical Andes of Ecuador.

Paul Muñoz (paul.munozp@ucuenca.edu.ec)
University of Cuenca: Universidad de Cuenca https://orcid.org/0000-0002-8000-8840
Johanna Orellana-Alvear
University of Cuenca: Universidad de Cuenca
Jörg Bendix
Philipps-Universität Marburg: Philipps-Universitat Marburg
Rolando Célleri
University of Cuenca: Universidad de Cuenca

Research Article

Keywords: Early Warning, flood, forecasting, Machine Learning, Andes.

Posted Date: April 7th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-395457/v1

Flood Early Warning Systems using Machine Learning techniques. Application to a catchment located in the Tropical Andes of Ecuador.

Paul Muñoz[1,2], Johanna Orellana-Alvear[1,3], Jörg Bendix[3] and Rolando Célleri [1,2]

[1] Departamento de Recursos Hídricos y Ciencias Ambientales, Universidad de Cuenca, Cuenca 010150, Ecuador
[2] Facultad de Ingeniería, Universidad de Cuenca, Cuenca 010150, Ecuado
[3] Laboratory for Climatology and Remote Sensing, Faculty of Geography, University of Marburg, 35032 Marburg, Germany

Abstract

Short-rain floods, especially flash-floods, produce devastating impacts on society, the economy, and ecosystems. A key countermeasure is to develop Flood Early Warning Systems (FEWSs) aimed at forecasting flood warnings with sufficient lead time for decision making. Although Machine Learning (ML) techniques have gained popularity among hydrologists, the research question poorly answered is what is the best ML technique for flood forecasting? To answer this, we compare the efficiencies of FEWSs developed with the five most common ML techniques for flood forecasting, and for lead times between 1 to 12 hours. We use the Tomebamba catchment in the Ecuadorean Andes as a case study, with three warning classes to forecast *No-alert, Pre-alert,* and *Alert of floods*. For all lead times, the Multi-Layer Perceptron (MLP) technique achieves the highest model performances (f1-macro score) followed by Logistic Regression (LR), from 0.82 (1-hour) to 0.46 (12-hour). This ranking was confirmed by the log-loss scores, ranging from 0.09 (1-hour) to 0.20 (12-hour) for the above mentioned methods. Model performances decreased for the remaining ML techniques (K-Nearest Neighbors, Naive Bayes and Random Forest) but their ranking was highly variable and not conclusive. Moreover, according to the g-mean, LR models depict greater stability for correctly classifying all flood classes, whereas MLP models are specialized in the minority (*Pre-alert* and *Alert*) classes. To improve the performance and the

applicability of FEWSs, we recommend future efforts to enhance input data representation and to develop communication applications between FEWSs and the public as tools to boost the preparedness of the society against floods.

Keywords: Early Warning; flood; forecasting; Machine Learning; Andes.

1. Introduction

Flooding is the most common natural hazard and produces one of the most damaging disasters worldwide (Ávila et al., 2019; Mirza, 2011; Paprotny et al., 2018; Stefanidis and Stathis, 2013). With much concern, recent studies have associated the increasing frequency and severity of flood events worldwide with land use changes (e.g., deforestation and urbanization) and climate change (Chang et al., 2019; Min et al., 2011; Paprotny et al., 2018; Sofia et al., 2017). This particularly holds for the tropical Andes region, where complex hydro-meteorological conditions results in the occurrence of strange and patchy rainfall events (Arias et al., 2021; Célleri and Feyen, 2009; Muñoz et al., 2016).

Floods can be classified into long- and short-rain floods according to their generation mechanisms, soil saturation or infiltration excess, respectively (Hundecha et al., 2017; Turkington et al., 2016). A key for building resilience short-rain floods (simply referred as floods along the manuscript) is to sufficiently anticipate the event itself and to gain time for better preparedness. The response time between a rainfall event and its associated flood response primarily depends on catchment properties and might vary from minutes to several hours (Borga et al., 2008). Here, special attention is given to flash-floods, which are events that develop less than six hours after a heavy rainfall with little or no forecast lead time (Knocke and Kolivras, 2007).

Flood anticipation can be achieved through the development of a Flood Early Warning System (FEWS) providing forecasts. FEWSs have proved to be cost-efficient solutions for life preservation, damage mitigation and resilience enhancement (Borga et al., 2011; del Granado et al., 2016; Sottolichio et al., 2011). However, although crucial, flood forecasting is still a major challenge in mountainous regions due to the difficulty to effectively measure precipitation events. The main limitations are the extreme spatial variability of precipitation, budget constraints for a sufficiently dense monitoring network and high-tech equipment limitations (Célleri and Feyen, 2009; Muñoz et al., 2016).

To date, there is no report of any operational flood forecasting system implemented in the Andean region for scales other than continental (Boers et al., 2014; Dávila, 2016; del Granado et al., 2016). An alternative attempt in Peru was aimed to derive daily maps of potential floods based on spatial cumulated precipitation in past days (Aybar et al., 2017). Other endeavors in Ecuador and Bolivia rather focused on the monitoring of runoff in the upper parts of catchments to anticipate flood events in the downstream parts (Dávila, 2016; Fernández de Córdova Webster and Javier Rodríguez López, 2016). However, such attempts are unsatisfactory as countermeasures against flood and especially flash-floods, where it is required to have reliable and accurate forecasts with lead times at least shorter than the response time between the farthest precipitation (runoff) and the runoff control stations.

There are, in general, two paradigms that drive the modelling of the precipitation-runoff response. First, the physically based paradigm aims to include the knowledge of the physical processes that occur towards the catchment by using physical process equations (Clark et al., 2017). However, it requires extensive ground data and in consequence, intensive computation which limits the temporal forecast window (Mosavi et al., 2018). Moreover, it is argued that

physically based models are inappropriate for real-time or short-term flood forecasting due to the inherent uncertainty of river-catchment dynamics (lack of ground data) and model over-parametrization (Young, 2002). The second data-driven paradigm, assumes floods as stochastic processes with distribution probabilities of occurrence derived from historical data. Here, the idea is to exploit relevant input information (e.g., precipitation, past runoff) to find relations to the target variable (i.e., runoff) without requiring knowledge about the underlying physical processes. Among traditional data-driven approaches, statistical modeling has also proved unsuitability for short-term prediction due to lack of accuracy, complexity, model robustness, and even computational costs (Mosavi et al., 2018). This has encouraged the use of advanced data-driven models, e.g., machine learning (ML), to overcome the aforementioned deficits (Bontempi et al., 2012; Chang et al., 2019; Galelli and Castelletti, 2013; Mosavi et al., 2018). Particularly during the last decade, ML approaches have gained increasing popularity among hydrologists (Mosavi et al., 2018).

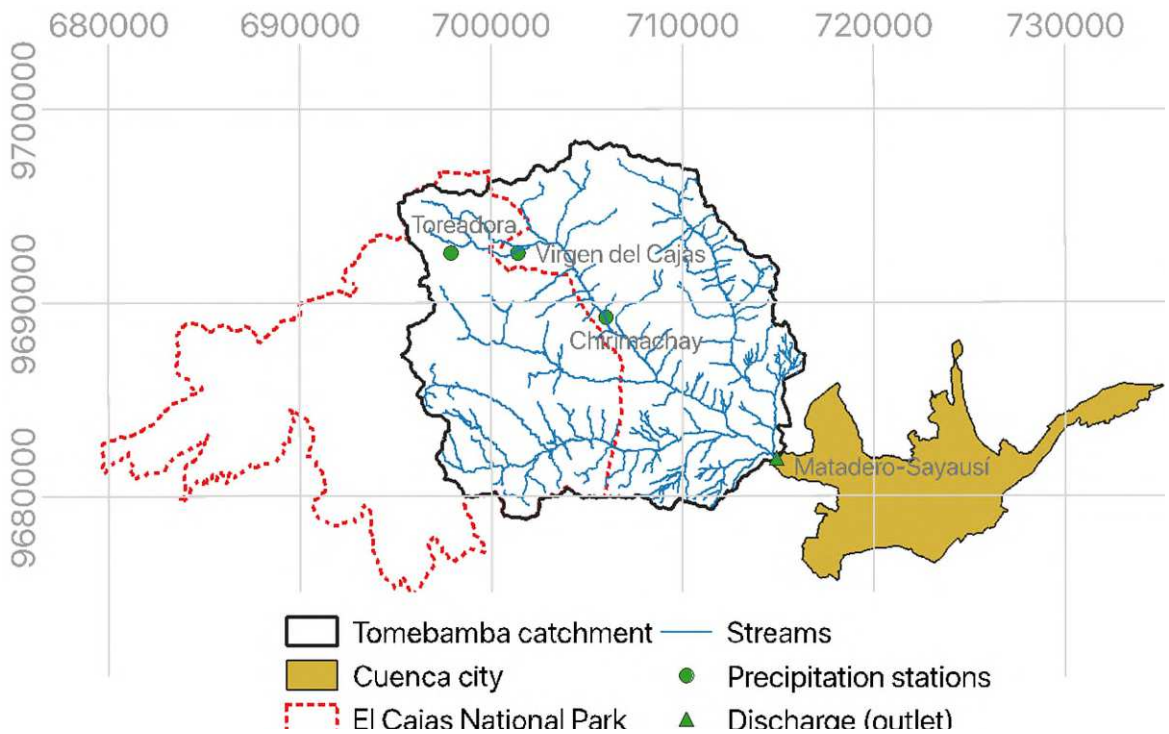
Several ML strategies for flood forecasting have been implemented worldwide by generating either, a quantitative or qualitative runoff forecast (Adamowski, 2008; Aichouri et al., 2015; Furquim et al., 2014; Khosravi et al., 2019; Muñoz et al., 2018; Solomatine and Xue, 2004; Toukourou et al., 2011). Qualitative forecasting consists on classifying floods into different categories according to its severity (i.e., runoff magnitude), and then implement flood classification prediction (Chen et al., 2013). The advantage of developing a FEWS under a qualitative forecasting approach is the possibility to generate a semaphore-like warning system which is easy to understand by decision makers and the public (non-hydrologists). The next step, however, is the selection of the optimal ML technique for flood forecasting with the objective to obtain reliable and accurate forecasts with sufficient lead time for the

89 decision making. To date, the problem has received scant attention in the research literature,
90 and no previous work has conducted such a study in mountainous regions.

91 The present study therefore aims to compare the performance of five ML classification
92 techniques for short-rain flood forecasting with special attention to flash-floods. Here, ML
93 models are developed for a medium-size mountain catchment which is representative for the
94 tropical Andes in Ecuador and comparable mountain areas. The ML models aim to forecast
95 three flood warning stages (*No-alert*, *Pre-alert* and *Alert*) for varying forecast lead times of
96 1, 4 and 6 hours (flash-floods), but also 8 and 12 hours to further test whether the lead time
97 can be satisfactory extended without losing models' operational value.

98 **2. Study area and dataset**

99 The study area comprises the Tomebamba catchment delineated upstream the Matadero-
100 Sayausí hydrological station of the Tomebamba river (Figure 1), where the river enters the
101 city. The Tomebamba is a tropical mountain catchment located in the southeastern flank of
102 the Western Andean Cordillera, draining to the Amazon river. The drainage area of the
103 catchment is approximately 300 km^2, spanning from 2800 to 4100 meters above the sea level
104 (m asl). Like many other mountain catchments of the region, it is primarily covered by a
105 páramo ecosystem, which is known by its important water regulation function (Célleri and
106 Feyen, 2009).



107

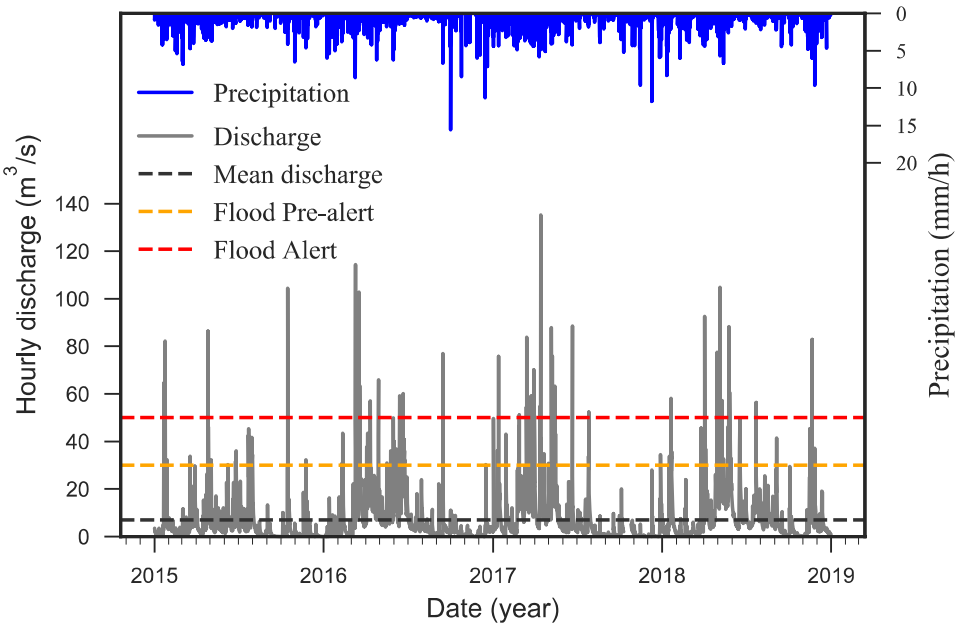
108 Figure 1. Location of the Tomebamba catchment at the Tropical Andean Cordillera of
109 Ecuador, South America (UTM coordinates).

110

111 The Tomebamba river plays a crucial role as a drinking water source for the city of Cuenca
112 (between 25 to 30 % of the demand). Other important water users are agricultural and
113 industrial activities. Cuenca, which is the third largest city of Ecuador (around 0.6 million
114 inhabitants), is crossed by 4 rivers that annually flood parts of the city, causing human and
115 significant economic losses.

116 The local water utility, the Municipal Public Company of Telecommunications, Water,
117 Sewerage and Sanitation of Cuenca (ETAPA-EP), defined three flood alert levels associated
118 to the Matadero-Sayausí station for floods originated in the Tomebamba catchment: i) *No-*
119 *alert* of flood occurs when the measured runoff is less than 30 m^3/s, ii) *Pre-alert* when runoff
120 is between 30 and 50 m^3/s, and iii) the flood *Alert* is triggered when discharge exceeds 50

121 m^3/s. With this definitions, and as shown in Figure 5, discharge label for the *No-alert* class
122 represents the majority of the data, whereas the *Pre-alert* and *Alert* classes comprises the
123 minority yet the most dangerous classes.
124 To develop and operate forecasting models, we use data of two variables: precipitation in the
125 catchment area and river discharge at a river gauge. For both variables, the available dataset
126 comprises 4 years of concurrent hourly time series, from Jan/2015 to Jan/2019 (Figure 2).
127 Precipitation information was derived from 3 tipping-bucket rain gauges: Toreadora (3955
128 m a.s.l.), Virgen (3626 m a.s.l.) and Chirimachay (3298 m a.s.l.), installed within the
129 catchment and along its altitudinal gradient. Whereas for discharge, we used data of the
130 Matadero-Sayausí station (2693 m a.s.l., Figure 1). To develop the ML modes, we split the
131 dataset into training and test subsets. The training period ran from 2015 to 2017, whereas
132 2018 was used as model testing phase.



133

134 Figure 2. Time series of precipitation (Toreadora) and discharge (Matadero-Sayausí).
135 Horizontal dashed lines indicate the mean runoff and the currently employed flood alert
136 levels for labeling the *Pre-alert* and *Alert* flood warnings classes.

3. ML methods for classification of flood alert levels

ML classification algorithms can be grouped by similarity, in terms of their functionality. According to Mosavi et al. (2018), five of the worldwide most-popular statistical method groups commonly employed for short-term flood prediction (extreme runoff) include:

i) Regression algorithms to model relationships between variables (e.g., Logistic Regression, Linear Regression, Multivariate Adaptive Regression Splines, etc.).

ii) Instance-based algorithms that rely on memory-based learning. This represents a decision problem feed with instances of data for training (e.g., K-nearest neighbor, learning vector quantification, locally weighted learning, etc.).

iii) Decision tree algorithms, which progressively divide the whole data set into subsets based on certain feature value, and until all target variables ate grouped in one category (e.g., Classification and regression tree, M5, Random Forest, etc.).

iv) Bayesian algorithms based on Bayes' theorem on conditional probability (e.g., Naive Bayes, Bayesian network, Gaussian Naïve Bayes, etc.).

v) Neural Network algorithms inspired by biological neural networks, aiming to convert input(s) to output(s) through specified transient states that enables the model to learn in a sophisticated way (e.g., Perceptron, Multi-layer perceptron, Radial Basis Function Network, etc.).

For this study, we selected five ML algorithms, one from each group. These are Logistic Regression, K-Nearest Neighbor, Random Forest, Naive Bayes, and Multi-layer Perceptron.

3.1 Logistic Regression

Logistic Regression (LR) is a discriminative model (i.e., it models the decision boundary between classes). At first, linear regressions are applied between model features to find

existent relationships. Then, probabilities (conditional) of belonging to any class are obtained through a logistic (Sigmoid) function that effectively deals with outliers (binary classification). From these probabilities, the LR classifies, with regularization, the dependent variables into any of the classes created.

However, for multiclass classification problems, all binary classification possibilities are run independently of the input (i.e., *No-alert* vs. *Pre-alert*, *No-alert* vs. *Alert* and *Pre-alert* vs. *Alert*). At the end, the solution is the classification with the maximum probability relative to others (multinomial LR). For this, the softmax function is used to find the predicted probability of each class (Bishop, 2006). The calculated probability for each class is assumed to be positive with the logistic function and then these values are normalized across all classes. The softmax function can be calculated from equation 1.

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{l=1}^{k} e^{z_l}} \quad (equation\ 1)$$

where z_i is the ith input of the softmax function, corresponding to class i from K number of classes.

3.2 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a non-parametric statistical pattern recognition algorithm. There is no theoretical or analytical background for classifying but rather intuitive statistical procedures (memory-based learning). Therefore, KNN classifies unseen data based on a similarity measure such as distance functions (e.g., Euclidean, Manhattan, Chebyshev, Hamming, etc.). The use of multiple neighbors, instead of one alone, is aimed to avoid wrong classification results occasioned by noisy features when defining boundaries for each class.

181 At the end, the majority vote of the nearest neighbors (probability, see formulation on Bishop
182 [2006]) determines the classification decision.
183 The number of nearest neighbors to be used can be optimized to reach its global minima,
184 avoiding longer computation times, and the influence of class sizes. The major advantage of
185 the KNN is its simplicity. However, the drawback is the fact that KNN is memory intensive,
186 all training data must be stored and compared when new information is to be tested.

187 ### 3.3 Random Forest

188 Random Forest (RF) is a supervised ML algorithm that ensembles a multitude of decorrelated
189 decision trees (DTs) voting for the most popular class (classification). In practice, a DT
190 (particular model) is a hierarchical analysis based on a set of conditions consecutively applied
191 to a dataset. To assure decorrelation, a bagging technique is employed by the RF algorithm
192 for growing DT from different randomly resampled training subsets obtained from the
193 original dataset. Each DT provides an independent output (class) of the phenomenon of
194 interest (i.e., runoff), contrary to numerical labels for regression applications. An extended
195 description of the RF functioning can be found in Breiman (2001) and Breiman (2017).
196 Predicted class probabilities of an input sample are calculated as the mean predicted class
197 probabilities of the trees in the forest. For a single tree, the class probability is computed as
198 the fraction of samples of the same class in a leaf. However, it is well-known that the
199 calculated training frequencies are not accurate conditional probability estimates due to the
200 high bias and variance of the frequencies (Zadrozny and Elkan, 2001). This deficiency can
201 be tackled by controlling the minimum number of samples required to be at a leaf node. This
202 was aimed to induce a smoothing effect, and therefore, to obtain probability estimates
203 statistically reliable.

3.4 Naïve Bayes

Naïve Bayes (NB) is a classification method based on Bayes' theorem with the "naive" assumption that there is no dependence between features in a class with any other feature, even if there is dependence (Zhang, 2004). Bayes' theorem can be expressed as follows:

$$P(y|X) = \frac{P(X|y)\,P(y)}{P(X)} \qquad (equation\ 2)$$

where $P(A|B)$ is the probability of y (hypothesis) happening, given the occurrence of X (features). Thus, X can be also expressed as $X = x_1, x_2, \dots, x_n$. Now, Bayes' theorem can be written as:

$$P(y|x_1, x_2, \dots, x_n) = \frac{P(x_1|y)\,P(x_2|y) \dots P(x_n|y)\,P(y)}{P(x_1)\,P(x_2) \dots P(x_n)} \qquad (equation\ 3)$$

There are different NB classifiers depending on the assumption of the distribution of $P(x_i|y)$. In this matter, the study of Zhang (2004) proved the optimality of NB under the Gaussian distribution even when the assumption of conditional independence is violated (real application cases). Additionally, for multiclass problems, the outcome of the algorithm is the class with the maximum probability. For Gaussian NB algorithm there are no parameters to be tuned.

3.5 Multi-layer Perceptron

The Multi-Layer Perceptron (MLP) is a class of feedforward artificial neural networks (ANN). A perceptron is a linear classifier that separates an input into two categories with a straight line and produces a single outcome. Input is a feature vector multiplied by specific weights and added to a bias. Contrary to a single-layer case, the MLP can approximate non-linear functions using additional so-called hidden layers. Prediction of probabilities of belonging to any class is calculated through the softmax function.

226 The MLP consists on multiple neurons in fully connected multiple layers. Determination of
227 the number of neurons in the layers with a trial-and-error approach remains widely used
228 (Maier et al., 2010). Neurons in the first layer correspond to the input data. Whereas, all other
229 nodes relate inputs to outputs by using linear combinations with certain weights and bias
230 together with an activation function. To measure the performance of the MLP, the logistic
231 loss function is defined with limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS)
232 as the optimizer for training the network. A detailed and comprehensive description of ANN
233 can be found in Maier and Dandy (2000).

234 **4. Methodology**

235 Figure 3 presents a scheme of the methodology followed in this study. As mentioned before,
236 precipitation and labeled discharge timeseries form the complete dataset for the study (see
237 Figure 2). This dataset is further split up for training and testing purposes. Then, for each
238 lead time, we compose separated training and test feature spaces (i.e., collection of input
239 features) for the tasks of model hyperparameterization and model assessment. This procedure
240 is repeated for each one of the ML techniques to be explored. Finally, we determine the
241 ranking of the performance quality of all ML methods for every lead time on the basis of
242 performance metrics and a statistical significance test.

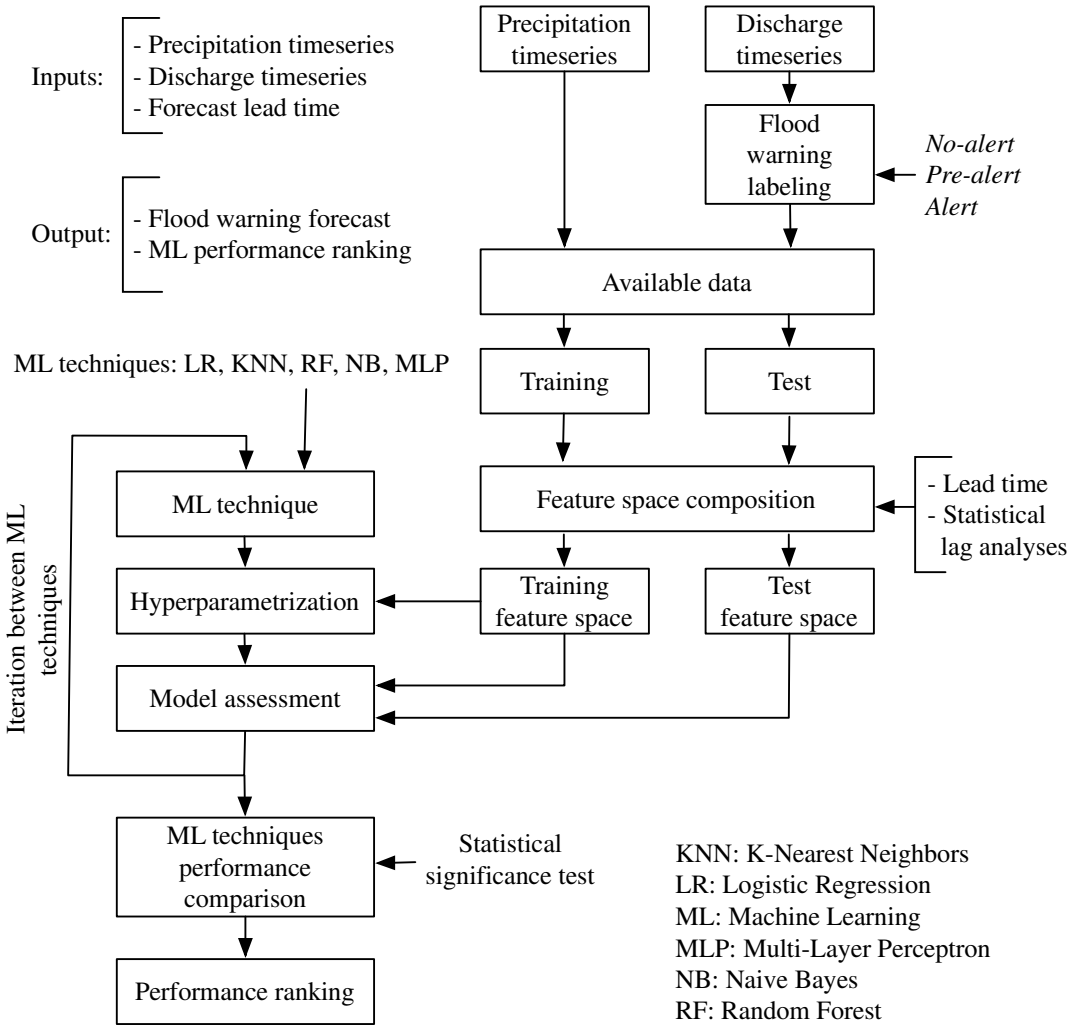


Figure 1. Workflow for developing and testing the ML flood forecasting models.

4.1 Feature space composition

For each lead time, single training and test feature spaces will be used for all ML techniques. A feature space is formed by features (predictors) coming from two variables: precipitation and discharge. The process of feature space composition starts by defining a specific number of precipitation and discharge features (present time and past hourly lags) according to statistical analyses relying on Pearson´s cross-, auto and partial-auto-correlation functions (Sudheer et al., 2002).

For precipitation, we derived one feature for each monitoring station, and as many features as number of past lags considered. As suggested by Muñoz et al. (2018), the number of lags from each station was selected by setting up a correlation threshold of 0.2. Similarly, for discharge, we used a number of features coming from past time slots of discharge selected for the analysis. It is worth noting that the number of discharge features triples since we replace each discharge feature by 3 features (one per flood warning class) in a process known as one-hot-encoding or binary encoding. Therefore, each created feature denotes 0 or 1 when the correspondent alarm stage is false or true, respectively.

Finally, we performed a feature standardization process prior to the computation stage of the KNN, LR, NB, and NN algorithms. Each feature was standardized by subtracting the mean and scaling it to unit variance. Therefore, it results in a distribution with standard deviation equal to 1 and mean equal to 0.

4.2 Model´s hyperparameterization

After the process of feature space composition, we set up optimal architectures for each ML forecasting model, and for each lead time. Optimal architectures are defined by optimal combinations of hyperparameters under the concept of balance between accuracy, and computational cost and speed. However, finding optimal architectures requires an exhaustive search of all possible combinations of hyperparameters. To overcome this issue, we relied on the Randomized Grid Search (RGS) with a 10-fold cross-validation scheme. The RGS procedure randomly explores the search space for discretized continuous hyperparameters on the basis of a cross-validation evaluation. Moreover, we selected the f1-macro score (see section 4.3.1) as the objective function.

4.2.1 Principal Component Analysis

In ML applications, analyzing high-dimension and complex data generally needs large amounts of memory and computational costs. Thus, we performed dimensionality reduction through a Principal Component Analysis (PCA) aimed to exclude correlating features that do not add additional information to the model. PCA was applied after feature scaling and normalization.

PCA is aimed to find the dimension of maximum variance and to reduce a feature space to that dimension so that model performance remains as intact as possible when compared to performance with the full feature space. But considering that each ML technique assimilates data in a different way, we did not define the number of principal components to kept on the basis of a fixed threshold of variance explanation (e.g., 80-90%) but rather performed an exploratory analysis to evaluate its influence in the models. Thus the number of PCAs was treated as an additional hyperparameter. Therefore, we optimized the number of principal components for each specific model (lead time and ML technique) with the aim to find the best possible model for each case.

All ML techniques and the RGS procedure were implemented through the scikit-learn package for ML in Python® (Pedregosa et al., 2011). Table 1 presents the relevant hyperparameters for each ML technique and their search space for tuning (Contreras et al., 2021). We employed default values for the hyperparameters that are not shown in Table 1.

Table 1. Model hyperparameters and their ranges/possibilities for tuning.

ML technique	Hyperparameters			
LR	*C* 0.001 - 1000	*penalty* {'l1','l2'}		
KNN	*n_neighbors*	*weights*	*metric*	*algorithm*

	3 - 75	{'uniform', 'distance'}	{'euclidean', 'manhattan', 'minkowski'}	{'auto','ball_tree', 'kd_tree','brute'}	
RF	*n_estimators*	*max_features*	*max_depth*	*min_samples_leaf*	*min_samples_split*
	50 -1000	{'auto', 'sqrt', 'log2'}	50 -1000	1-500	1-500
MLP	*solver*	*max_iter*	*alpha*	*hidden_layers*	
	{'lbfgs'}	10 - 5000	1 E-9 - 0.1	1 - 16	

4.3 Model performance evaluation

Forecasting hydrological extremes such as floods turns into an imbalanced classification problem. It even becomes more complex when the interest lies in the minority class of the data (flood alert). This is because most of the ML classification algorithms focus on the minimization of the overall error rate (incorrectly classifying the majority class) (Chen et al., 2004). Resampling the class distribution of the data for obtaining an equal number of samples per class is one solution. In this study, we used another approach which relies on training ML models with the assumption of imbalanced data.

The approach we used penalizes mistakes in samples belonging to the minority classes rather than under-sampling or over-sampling data. In practice, this implies that for a given efficiency metric (see section 5.1), the overall score is the result of averaging each individual performance metric (for each class) multiplied by its corresponding weight factor. The weight factors for each class are calculated according to class frequencies (inversely proportional), as follows:

$$w_i = \frac{N}{C\, n_j} \quad (equation\ 4)$$

Where w_i is the weight to class i, N is the total number of observations, C is the number of classes, and n_j is the number of observations in class i. This implies that higher weights will

311 be obtained for minority classes.

312 4.3.1 Performance metrics

313 We used measures of performance derived from the well-known confusion matrix and
314 specifically aimed for imbalanced datasets and multiclass problems. These are the f1 score,
315 the geometric mean and the logistic regression loss score (Akosa, 2017; Chen et al., 2004;
316 Gu et al., 2009; Hossin and Sulaiman, 2015; Read et al., 2011; Sun et al., 2009). The use of
317 a compendium of metrics is needed because neither of the measures is adequate by itself to
318 properly explain the performance of the model; on the contrary, they complement each other.

319 4.3.1.1 F1 score

320 F score is a metric that relies on precision and recall, which are effective metrics for
321 imbalance problems. When these metrics are integrated in the f score as an average (weighted
322 harmonic mean), we refer to the f1 score. The f1 score can be calculated from equation 5.

323 $$f1\ score = \frac{2 * Precision\ * Recall}{(Precision\ + Recall\ )} \quad (equation\ 5)$$

324 where precision and recall are defined in the following equations:

325 $$Precision = \frac{TP}{TP + FP} \quad (equation\ 6)$$

326 $$Recall = \frac{TP}{TP + FN} \quad (equation\ 7)$$

327 where TP stands for True Positives, FP for False Positives and FN for False Negatives.
328
329 The f1 score ranges from 0 to 1 (indicating perfect precision and recall). The advantage of
330 using the f1 score (compared to the arithmetic or geometric means) is that it penalizes models
331 the most when either precision or recall is low. However, by classifying a *No-Alert* flood
332 warning as *Alert* might have a different impact on the decision making than when the opposite

occurs. This limitation scales up when there is an additional state *(Pre-alert*). Thus, the interpretation of the f1 score must be taken with care. For multiclass problems, the f1 score is commonly averaged across all classes (f1-macro score) to indicate the overall model performance.

4.3.1.2 Geometric mean

The geometric-mean (g-mean) measures the balanced performance of TP and True Negative (TN) rates simultaneously. The g-mean gives equal importance to the classification task of both the majority (*No-alert*) and minority (*Pre-alert* and *Alert*) classes. The g-mean is an evaluation measure that can be used to maximize accuracy in order to balance TP and TN examples at the same time with a good trade-off (Gu et al., 2009). It can be calculated from equation 8.

$$G\text{-}mean = \sqrt{(\mathrm{TP}_{\mathrm{rate}} * \mathrm{TN}_{\mathrm{rate}})} \quad (equation\ 8)$$

Where TP_{rate} and TN_{rate} are defined by:

$$TP_{rate} = Recall \quad (equation\ 9)$$

$$TN_{rate} = \frac{TN}{TN + FP} \quad (equation\ 10)$$

The g-mean ranges from 0 to 1, where low values indicate poor performance in the classification of the majority class even if the minority classes are correctly classified as such.

4.3.1.3 Log-loss score

The Logistic regression loss (log-loss) measures the performance of a classification model when the input is a probability value between 0 and 1. It accounts for the uncertainty of the forecast based on how much it varies from the actual label. For multiclass classification, a separate log-loss is calculated for each class label (per observation), and the results are

summed up. The log-loss score for multi-class problems is defined as:

$$Log\ loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}) \quad (equation\ 11)$$

Where N is the number of samples, M is the number of classes, y_{ij} is 1 when the observation belongs to class j; else 0, and p_{ij} is the predicted probability that the observation belongs to class j.

Starting from 0 (best possible score), the log-loss magnitudes increase as the probability diverges from the actual label. It punishes worse errors more harshly to promote conservative predictions. For probabilities close to 1, the log-loss slowly decreases. However, as the predicted probability decreases, the log loss increases rapidly.

4.3.2 Statistical significance test for comparing ML algorithms

Although we can directly compare performance metrics between ML alternatives and claim to have found the best one based on the score, it is not certain whether the difference in metrics is real or the result of statistical chance. In this matter, there is a number of different statistical hypothesis frameworks that can be used to compare the performance of classification models (e.g., difference of proportions, paired comparison, binomial test, etc.). Among these, Raschka (2018) recommends the chi-squared test to quantify the likelihood of the samples of skill scores being observed under the assumption that they have the same distributions. The assumption is known as the null hypothesis and it is aimed to prove whether there is a statistically significant difference between two models (error rates). If rejected, it can be concluded that any observed difference in performance metrics is likely due to a difference in the models and not due to statistical chance. Here, the chi-squared test

is used to assess whether the difference in the observed proportions of the contingency tables of a pair of ML algorithms (for a given lead time) are significant.

For the model comparison, we proved statistical significance of improvements/degradations for all lead times (training and test subsets) under a value of 0.05 (chi-squared test). In all cases, we set the MLP as the base model to be compared against the remaining ones.

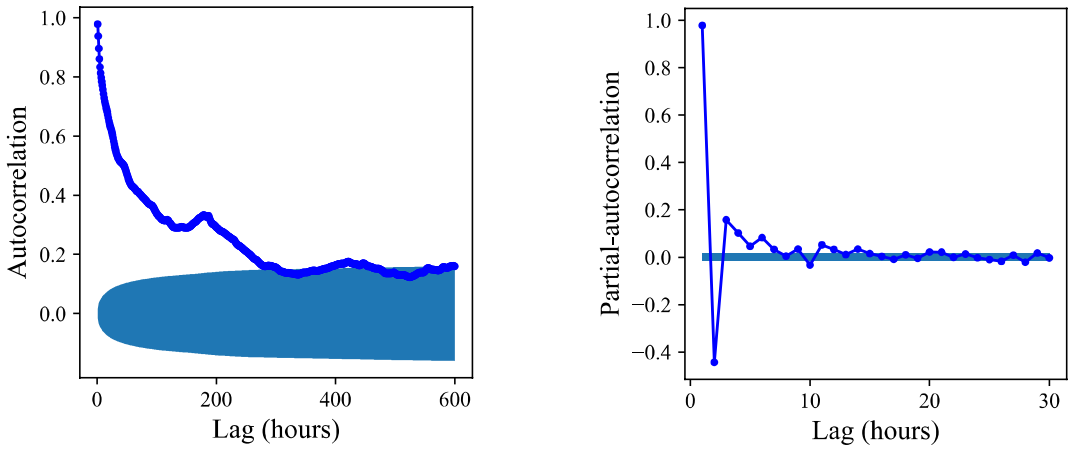
5. Results

We present in this section the results of the flood forecasting models developed with the LR, KNN, RF, NB and MLP techniques, and for lead times of 1, 4, 6, 8 and 12 hours. For each model, we addressed the forecast of three possible flood warnings (*No-alert*, *Pre-alert* and *Alert*). In the first subsection, we present the results of the feature space composition process, taking the 1-hour lead time case as an example. Then, we show the results of the hyperparameterization for all models, followed by an evaluation and ranking of ML techniques performance.

5.1 Feature space composition

Figures 4 and 5 show the results of the discharge and precipitation lag analyses for the flood forecasting model at 1-hour in advance or the rainfall. Figure 4.a plots the discharge autocorrelation function (ACF) and the corresponding 95% confidence interval from lag 1 up to 600 (hours). We found a significant correlation up to a lag of 280 h (maximum correlation at the first lag), and thereafter, the correlation fell within the confidence band. On the other hand, Figure 4.b presents the discharge partial-autocorrelation function (PACF) and its 95% confidence band from lag 1 to 30 h. We found a significant correlation up to lag 8 h (first lags outside the confidence band). As a result, based on the interpretation of the ACF

and PACF analyses (according to Muñoz et al. [2018]), we decided to include 8 discharge lags (hours) for the case of 1-hour flood forecasting in the Tomebamba catchment.



(a) (b)

Figure 4. (a) Autocorrelation function (ACF) and (b) Partial-autocorrelation function (PACF) of the Matadero-Sayausí (Tomebamba catchment) discharge series. The blue hatch indicates in each case the correspondent 95% confidence interval.

Figure 5 plots the Pearson's cross-correlation between the precipitation at each rain gauge station and the Matadero-Sayausí discharge. For all stations, we found a maximum correlation at lag 4 (maximum 0.32 for Chirimachay). With the fixed correlation threshold of 0.2, we included 11, 14 and 15 lags for Virgen, Toreadora and Toreadora stations, respectively.

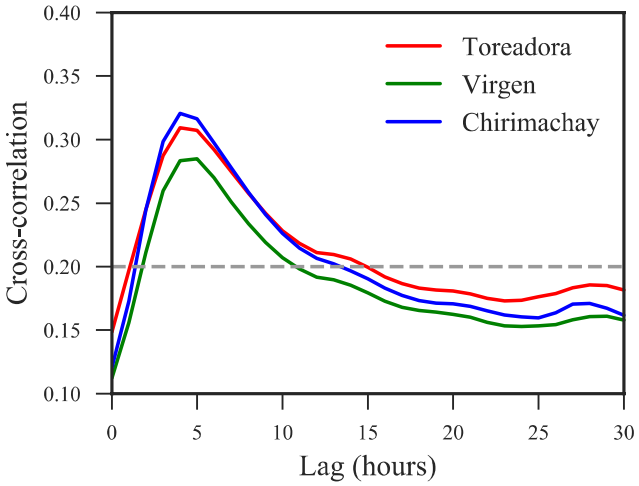


Figure 5. Pearson's cross-correlation comparison between the Toreadora (3955 m a.s.l), Virgen (3626 m a.s.l.) and Chirimachay (3298 m a.s.l.) precipitation stations and the Matadero-Sayausí discharge series. Note the blue horizontal line at a fixed correlation of 0.2 for determining past lags.

Similarly, the same procedure was applied for the remaining lead times (i.e, 4, 6, 8 and 12 hours). In Table 2, we present the input data composition and the resulting total number of features obtained from the lag analyses for each forecasting model. For instance, for the 1 hour case, the total number of features in the feature space equals 67, from which 43 are derived from precipitation (past lags and one feature from present time for each station), and 24 from discharge (one-hot-encoding).

Table 2. Input data composition (number of features) for all ML models of the Tomebamba catchment.

	Discharge lags* [hours]	Precipitation lags [hours]			
Lead time [hours]	Matadero-Sayausí	Toreadora	Chirimachay	Virgen	Number of features
1	8	15	14	11	67
4	12	18	17	14	88
6	14	20	19	16	100
8	16	22	21	18	112
12	20	26	25	22	136

* Note that each discharge feature triples (three flood warning classes) after a one-hot-encoding process.

5.2 Model hyperparameterization

In Table 3 we present the results of the hyperparameterization including the number of PCA components employed for achieving the best model efficiencies. No evident relation between the number of principal components and the ML technique nor the lead time was found. In fact for some models, we found differences in the f1-macro score lower than 0.01 for a low and high number of principal components. See for instance the case of the KNN models where the optimal number of components significantly decayed for lead times greater than 4 hours. For the 1-hour lead time, 96% of the components were used, whereas for the rest of the lead times only less than 8% of the components were used.

If we turn to the evolution of models' complexity with lead time (Table 3) more complex ML architectures were generally needed to forecast greater lead times. This is underpinned by the fact that the corresponding optimal models required for greater lead times a stronger regularization (lower values of *C*) for LR, a greater number of neighbors (*n_neighbors*) for KNN, more specific trees (lower values of *min_samples_split*) for RF and more hidden layers (*hidden_layer*s) for MLP.

Table 3. Model hyperparameters and number of principal components used for each specific model (ML technique and lead time).

ML technique	Hyperparameter	Lead time				
		1h	4h	6h	8h	12h
LR	*C*	0.01	0.00001	0.0001	0.0001	0.001
	penalty	'l2'	'l2'	'l2'	'l2'	'l2'
	*PCA_components**	58	62	78	75	51
KNN	*n_neighbors*	15	15	23	33	55
	weights	'uniform'	'uniform'	'uniform'	'uniform'	'uniform'
	metric	'minkowski'	'minkowski'	'minkowski'	'minkowski'	'minkowski'
	Algorithm	'auto'	'auto'	'auto'	'auto'	'auto'
	*PCA_components**	64	6	6	6	4
RF	*n_estimators*	700	700	700	700	800

	max_features	'sqrt'	'auto'	auto	'log2'	'auto'
	max_depth	350	350	350	350	300
	min_samples_leaf	450	450	480	480	450
	min_samples_split	10	5	5	2	4
	*PCA_components**	66	79	90	45	78
NB	*PCA_components**	63	64	87	89	15
	solver	'lbfgs'	'lbfgs'	'lbfgs'	'lbfgs'	'lbfgs'
	max_iter	2000	2000	2000	2000	2000
MLP	*alpha*	0.0001	0.0001	0.0001	0.0001	0.0001
	hidden_layers	2	3	2	2	4
	*PCA_components**	63	51	64	76	4

* From the total number of features: 1h=67, 4h=88, 6h=100, 8h=112, 12h=136 features

5.3 Model performance evaluation

As mentioned before, model performances calculated with the f1 score, g-mean and log-loss score were weighted according to class frequencies. Table 4 presents the frequency distribution for the complete dataset and the training and test subsets. Here, the dominance of the *No-alert* flood class is evident, with more than 95% of the samples in both subsets. With this information, the class weights for the training period were calculated as $w_{No-alert} = 0.01$, $w_{Pre-alert} = 0.55$ and $w_{Alert} = 0.51$.

Table 4. Number of samples and relative percentage for the entire dataset and for the training and test subsets.

Warning	Complete	Training	Test
No-alert	32596 (96.1%)	24890 (96.2%)	7706 (95.7%)
Pre-alert	720 (2.1 %)	473 (1.8 %)	247 (3.1 %)
Alert	609 (1.8 %)	509(2.0 %)	100 (1.2 %)

The results of the model performance evaluation for all ML models and lead times (test subset) are summarized in Table 5. We proved for all models that the differences in performance metrics for a given lead time were due to the difference in the ML techniques employed rather than to statistical chance.

As expected, ML models' ability to forecast floods decreased for a longer lead time. For instance, for the case of 1-hour forecasting, we found a maximum f1-macro score of 0.88 (MLP) for the training, and 0.82 (LR) for the test subset. Whereas, for the 12-hour case, the maximum f1-macro score was 0.71 (MLP) for the training and 0.46 (MLP) for the test subset.

Table 5 . Models' performance evaluation on the test subset. Bold fonts indicate the best performance for a given lead time.

Lead time (hours)	RF	KNN	LR	NB	MLP
F1-macro score					
1	0.59	0.73	**0.82**	0.57	0.78
4	0.47	0.57	0.59	0.46	**0.62**
6	0.47	0.45	0.50	0.41	**0.51**
8	0.44	0.41	0.44	0.45	**0.51**
12	0.42	0.36	0.44	0.43	**0.46**
G-mean					
1	0.86	0.77	**0.88**	0.81	0.83
4	0.75	0.63	**0.76**	0.73	0.71
6	0.70	0.56	**0.72**	0.68	0.62
8	**0.73**	0.53	0.67	0.62	0.62
12	0.69	0.50	**0.69**	0.64	0.56
Log-loss score					
1	0.28	0.38	1.09	3.14	**0.09**
4	0.38	0.46	0.74	4.10	**0.11**
6	0.45	0.58	0.47	4.71	**0.14**
8	0.50	0.65	0.53	0.59	**0.16**
12	0.59	0.70	0.57	2.17	**0.20**

All improvements and degradations are statistically significant

The extensive hyperparameterization (RGS scheme) powered by a 10-fold cross validation served to assure robustness in all ML models and therefore, to reduce overfitting. We found only a small difference between the performance values by using the training and the test subsets. For all models, maximum differences in performances were lower than 0.27 for the f1-macro score and 0.19 for the g-mean.

In general, for all lead times, the MLP technique obtained the highest f1-macro scores, followed by the LR algorithm. This performance dominance was confirmed by the ranking of the models according to the log-loss score. The ranking of the remaining models was highly variable and therefore not conclusive. For instance, results of the KNN models obtained the second highest score for the training subset, but the lowest for the test subset (especially for longer lead times). This is because the KNN is a memory-based algorithm and therefore more sensible, , to the inclusion of information different to the training subset in comparison to the remaining ML techniques. This can be noted in Table 4, where the training and test frequency distributions are different for the Pre-alert and Alert classes.

On the other hand, for the g-mean score we obtained a different ranking of the methods. We found the highest scores for the LR algorithm, followed by the RF and the MLP models. In spite of this behavior , the values of the g-mean were superior to the f1-macro scores for all lead times and subsets. This is because the f1 score relies on the harmonic mean. Therefore, the f1 score penalizes more a low precision or recall in comparison with a metric based on a geometric or arithmetic mean. Results of the g-mean served to identify that the LR is the most stable method in terms of correctly classifying both the majority (*No-alert*) and the minority (*Pre-alert* and *Alert*) flood warning classes. Whereas the MLP technique could be used to focus on the minority (flood alert) classes.

To extend the last idea, we analyzed the individual f1 scores of each flood warning class. This serves to unveil the ability of the model to forecast the main classes of interest , i.e., *Pre-alert* and *Alert*. Figure 6 presents the evolution of the f1 score of each ML algorithm at the corresponding lead time. We found that for all ML techniques, the *Alert* class is clearly the most difficult one to forecast when the f1-macro score was selected as the score metric

for the hyperparameterization task. An additional exercise consisted in choosing the individual f1 score for the *Alert* class as the target for hyperparameterization of all models. However, although we obtained similar results for the *Alert* class, the scores of the *Pre-alert* class were highly deteriorated, even reaching to scores near to zero.

The most interesting aspect of Figure 6 is that the most efficient and stable models across lead times (test subset) were the models based on MLP and LR techniques. It is also evident that for all forecasting models, we found lack of robustness for the *Pre-alert* warning class (major differences between the f1 scores for the training and test subsets). A possible explanation for this might be that the *Alert* class implies a *Pre-Alert* warning class, but not the opposite. Consequently, this might mislead the learning process causing overfitting issues during training and leading to poor performances when assessing unseen data during the testing phase.

Moreover, although we added a notion of class frequency distribution (weights) to the performance evaluation task, it can be noted that for all models, the majority class is almost perfectly classified. This is because the *No-alert* class arises from low-to-medium discharge magnitudes. This eases and simplifies the learning process of the ML techniques since these magnitudes can be related to normal conditions (present time and past lags) of precipitation and discharge.

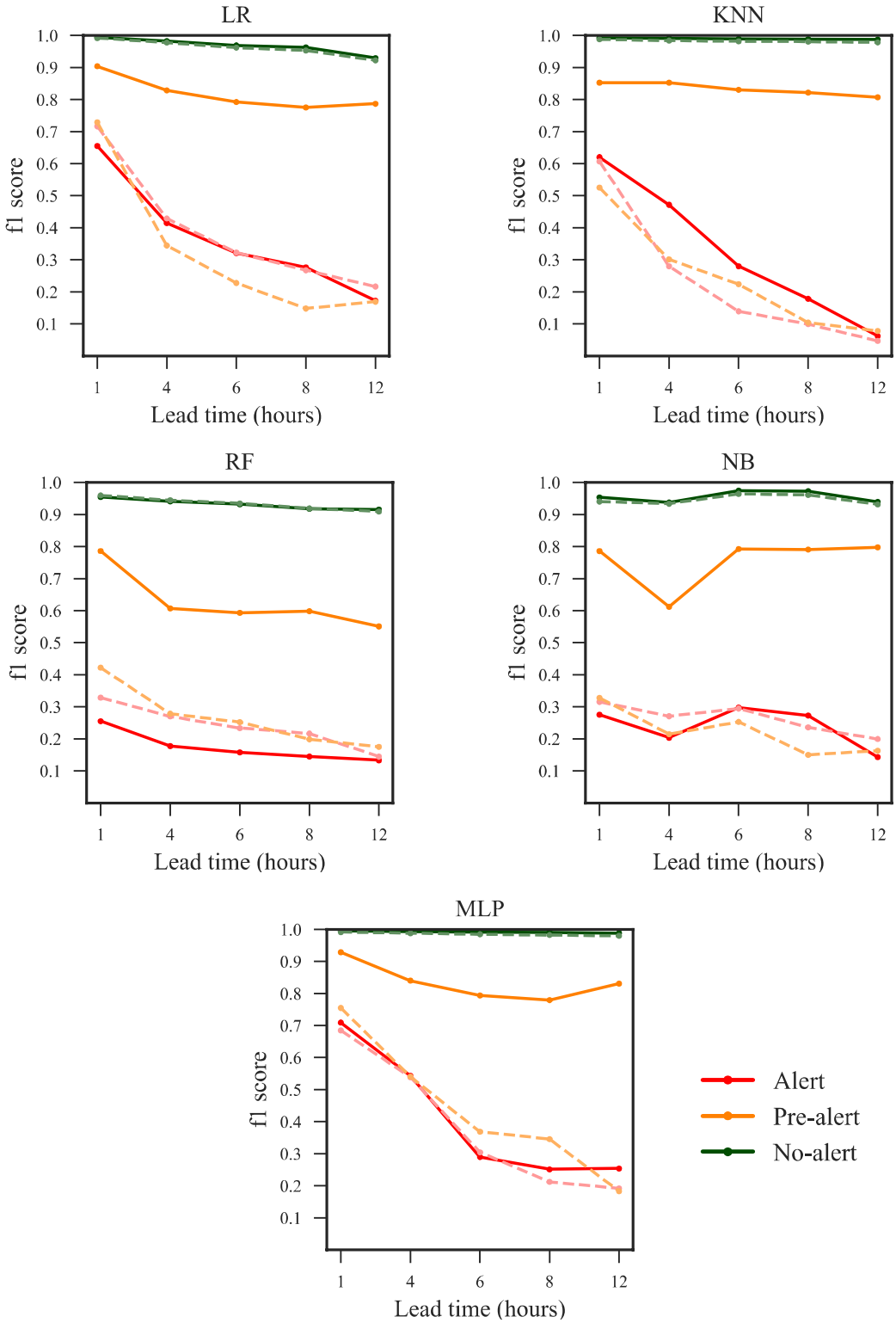


Figure 6. F1 scores per flood warning state (No-alert, Pre-alert and Alert) for all combinations of ML techniques and lead times. The brightest and dashed lines in each case (color coding) represent the scores for the test subset.

6. Discussion

In this study, we developed and evaluated five different FEWSs relying on the most common ML techniques for flood forecasting, and for short-term lead times of 1, 4, 6 hours for flash-floods, and 8 and 12 hours to prove models' operational value for longer lead times. We used historical runoff data to define and label the three flood warning scenarios to be forecasted (*No-alert*, *Pre-alert*, and *Alert*). We constructed the feature space for the models according to statistical analyses for precipitation and discharge together with a PCA analysis embedded in the hyperparameterization. This was aimed to better exploit the learning algorithm of each ML technique. In terms of model assessing, we proposed an integral scheme based on the f1 score, the geometric mean, and the log-loss score to deal with data imbalance and multiclass characteristics. Finally, the assessment was complemented with a statistical to provide a performance ranking between ML techniques.

For all lead times, we obtained the best forecasts for both, the majority and minority classes from the models based on the LR, RF and MLP techniques (G-mean). Whereas the two most suitable models for the dangerous warning classes (Pre-Alert and Alert) were the MLP and LR (f1 and log-loss scores). This finding has important implications for developing FEWSs since real-time applications must be capable to deal with both the majority and minority classes. It can therefore be suggested that the most appropriate forecasting models are based on the MLP technique.

Moreover, the results on the evolution of model performances across lead times suggest that the models are acceptable for lead times up to 6 hours, i.e., the models are suitable for flash-flood applications in the Tomebamba catchment. For lead times greater than 6 hours, we found a strong decay in model performance. In other words, the utility of the 8 and 12-hour

535 forecasting models is somewhat limited by models' operational value. This is because, in the
536 absence of rainfall forecasts, the assumption of future rain is solely based on runoff
537 measurements at past and present times. This generates forecasts that are not accurate enough
538 for horizons greater than the concentration time of the catchment. The concentration time of
539 the Tomebamba catchment was estimated between 2 and 6 hours according to the equations
540 of Kirpich, Giandotti, Ven Te Chow and Temez, respectively (a summary of the equations
541 can be found in de Almeida et al. [2014]). This results in an additional performance decay
542 for the 8 and 12-hour cases in addition to the error in modeling.
543 Comparable to our results, the study of Furquim et al. (2014) compared the performance of
544 different ML classification algorithms for flash-flood nowcasting (3 hours) in a river located
545 in an urban area of Brazil. They found that models based on neural networks and decision
546 trees outperformed the ones based on the Naïve Bayes technique. However, this study only
547 evaluated the percentage of correctly classified instances which is a very simplistic
548 evaluation. Thus, we recommend a more integral assessment of model performances, as the
549 one in the current study, that allows for a better support on decision making.

550 Other studies related to quantitative forecasting (Aichouri et al. [2015], Toukourou et al.
551 [2011] and Adamowski [2008]) have found that neural networks-based models usually
552 outperformed the remaining techniques we proposed in this study. Nevertheless, in some
553 cases, the use of less expensive techniques regarding the computational costs produces
554 comparable results as in Solomatine and Xue (2004). This is also the case in our short-rain
555 and flash-flood flood classification problem.

556 As a further step, we propose the development of ensemble models for improving the
557 performance results of our individual models. This can be accomplished by combining the

558 outcomes of the ML models with weights obtained, for instance, from the log-log scores.
559 Another alternative that is becoming popular is the construction of hybrid models as a
560 combination of ML algorithms for more accurate and efficient models (Khosravi et al., 2019;
561 Mosavi et al., 2018; Solomatine and Xue, 2004)
562 Moreover, as stated by Solomatine and Xue (2004), inaccuracies in forecasting floods are
563 mainly due to data-related problems. In this regard, Muñoz et al. (2016) reported a deficiency
564 in precipitation-driven models due to rainfall heterogeneity in mountainous areas, where
565 orographic rainfall formation occurs. In most cases, rainfall events are only partially captured
566 by punctual measurement or even the entire storm coverage is missing.

567 Additionally, at some point, precipitation-runoff models will reach a certain effectiveness
568 threshold that cannot be exceeded without incorporating new types of data such as soil
569 moisture (Li et al., 2016; Loumagne et al., 2001). In humid areas, the rainfall-runoff relation
570 also depends on other variables such as evapotranspiration, soil moisture, and land use, which
571 leads to significant spatial variations of water storage. However, these variables are difficult
572 to measure or estimate.

573 **7. Conclusions**

574 The current study set out to develop and perform an integral performance evaluation of five
575 of the most common ML classification techniques for short-rain flood forecasting, with
576 special attention to flash-floods. The developed models aimed at forecasting three flood
577 warnings*, No-alert*, *Pre-alert*, and *Alert for the Tomebamba catchment in the tropical Andes*
578 *of Ecuador*.
579 From the results, the following conclusions can be drawn:

(i) Results related to model comparison are statistically significant. This is important because this is not usually performed in other studies and it validates the performance comparison and ranking hereby presented.

(ii) For all lead times, the most suitable models for flood forecasting are based on the MLP followed by the LR techniques. From the integral evaluation (i.e., several performance metrics), we suggest LR models as the most efficient and stable option for classifying both the majority (*No-alert*) and the minority (*Pre-alert* and *Alert*) classes. Whereas, we recommend MLP when the interest lies in the minority classes.

(iii) The forecasting models we developed are robust. Differences in the averaged f1, g-mean and log-loss scores between training and test are consistent to all models. However, we limit the utility of the models for flash-flood applications (lead times up to 6 hours). For longer lead times, we encourage improvement in precipitation' representation, and even forecasting this variable for lead times longer than the concentration time of the catchment.

(iv) A more detailed model assessment (individual f1 scores) unveiled the difficulties to forecast the *Pre-alert* and *Alert* flood warnings. This was evidenced when the hyperparameterization was driven for the optimization of the forecast for the alert class and this, however, did not improve the model performance of this specific class.

This study can be extended with a deep exploration of the effect of input data composition, precipitation forecasting, and the feature engineering strategies for both the MLP and LR techniques. Feature engineering pursues the use of data representation strategies that could, for example, provide spatial and temporal information of the

precipitation in the study area. This can be done by spatially discretizing precipitation in the catchments with the use of remotely sensed imagery. With this additional knowledge, it would be possible to improve the performance of the models hereby developed at longer lead times. For FEWSs, the effectiveness of the models is strongly linked to the speed of communication to the public after a flood warning is triggered. Therefore, future efforts should focus on the development of a web portal and/or mobile application as a tool to boost the preparedness of the society against floods which currently threaten people's lives, possessions, and environment in Cuenca and other comparable tropical Andean cities.

Declarations

Funding The current study was funded by the project: "Desarrollo de modelos para pronóstico hidrológico a partir de datos de radar meteorológico en cuencas de montaña" funded by the Research Office of the University of Cuenca (DIUC) and Empresa Pública Municipal de Telecomunicaciones, Agua Potable, Alcantarillado y Saneamiento de Cuenca (ETAPA-EP). Our thanks go to these institutions for their generous funding.

Conflicts of interest/Competing interests The authors have no conflicts of interest to declare that are relevant to the content of this article.

Availability of data and material The authors declare that all data and materials as well as software application or custom code support their published claims and comply with field standards.

Code availability Not applicable

Authors' contributions Conceptualization: Rolando Célleri and Paul Muñoz; Methodology: Paul Muñoz and Johanna Orellana-Alvear; Formal analysis and investigation: Paul Muñoz; Writing - original draft preparation: Paul Muñoz; Writing - review and editing: Johanna Orellana-Alvear, Rolando Célleri and Jörg Bendix; Funding acquisition: Rolando Célleri; Resources: Rolando Célleri; Supervision: Rolando Célleri, Jörg Bendix.

Ethics approval Not applicable

Consent to participate Not applicable

Consent for publication Not applicable

Acknowledgments We acknowledge the Ministry of Environment of Ecuador (MAAE) for providing research permissions. We are grateful to the staff and students that contributed to the hydrometeorological monitoring.

References

Adamowski, J.F., 2008. Development of a short-term river flood forecasting method for snowmelt driven floods based on wavelet and cross-wavelet analysis. J. Hydrol. 353, 247–266.

Aichouri, I., Hani, A., Bougherira, N., Djabri, L., Chaffai, H., Lallahem, S., 2015. River flow model using artificial neural networks. Energy Procedia 74, 1007–1014.

Akosa, J., 2017. Predictive accuracy: A misleading performance measure for highly imbalanced data, in: Proceedings of the SAS Global Forum.

Arias, P.A., Garreaud, R., Poveda, G., Espinoza, J.C., Molina-Carpio, J., Masiokas, M., Viale, M., Scaff, L., van Oevelen, P.J., 2021. Hydroclimate of the Andes Part II: Hydroclimate Variability and Sub-Continental Patterns. Front. Earth Sci. 8, 666.

Ávila, Á., Guerrero, F.C., Escobar, Y.C., Justino, F., 2019. Recent Precipitation Trends and Floods in the Colombian Andes. Water 11, 379.

Aybar, C., Lavado-Casimiro, W., Huerta, A., Fernández, C., Vega, F., Sabino, E., Felipe-Obando, O., 2017. Uso del Producto Grillado "PISCO" de precipitación en Estudios, Investigaciones y Sistemas Operacionales de Monitoreo y Pronóstico Hidrometeorológico. Nota Técnica 1.

Bishop, C.M., 2006. Pattern recognition and machine learning. springer.

Boers, N., Bookhagen, B., Barbosa, H.M.J., Marwan, N., Kurths, J., Marengo, J.A., 2014. Prediction of extreme floods in the eastern Central Andes based on a complex networks approach. Nat. Commun. 5, 1–7.

Bontempi, G., Taieb, S. Ben, Le Borgne, Y.-A., 2012. Machine Learning Strategies for Time Series Forecasting., in: EBISS. pp. 62–77.

Borga, M., Anagnostou, E.N., Blöschl, G., Creutin, J.D., 2011. Flash flood forecasting, warning and risk management: The HYDRATE project. Environ. Sci. Policy 14, 834–844. https://doi.org/10.1016/j.envsci.2011.05.017

Borga, M., Gaume, E., Creutin, J.D., Marchi, L., 2008. Surveying flash floods: gauging the ungauged extremes. Hydrol. Process. 2274, 2267–2274. https://doi.org/10.1002/hyp.7111

Breiman, L., 2017. Classification and regression trees. Routledge.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. https://doi.org/10.1023/A:1010933404324

Célleri, R., Feyen, J., 2009. The Hydrology of Tropical Andean Ecosystems: Importance, Knowledge Status, and Perspectives. Mt. Res. Dev. 29, 350–355. https://doi.org/10.1659/mrd.00007

Chang, L.-C., Chang, F.-J., Yang, S.-N., Kao, I., Ku, Y.-Y., Kuo, C.-L., Amin, I., others, 2019. Building an Intelligent Hydroinformatics Integration Platform for Regional Flood Inundation Warning Systems.

Chen, C., Liaw, A., Breiman, L., 2004. Using random forest to learn imbalanced data. Univ. California, Berkeley 110, 1–12.

Chen, S., Xue, Z., Li, M., 2013. Variable Sets principle and method for flood classification. Sci. China Technol. Sci. 56, 2343–2348.

Clark, M.P., Bierkens, M.F.P., Samaniego, L., Woods, R.A., Uijlenhoet, R., Bennett, K.E., Pauwels, V.R.N., Cai, X., Wood, A.W., Peters-lidard, C.D., 2017. The evolution of process-based hydrologic models : historical challenges and the collective quest for physical realism 3427–3440.

Contreras, P., Orellana-Alvear, J., Muñoz, P., Bendix, J., Célleri, R., 2021. Influence of Random Forest Hyperparameterization on Short-Term Runoff Forecasting in an Andean Mountain Catchment. Atmosphere (Basel). 12, 238.

Dávila, D., 2016. 21 experiencias de sistemas de alerta temprana en América Latina.

de Almeida, I.K., Almeida, A.K., Anache, J.A.A., Steffen, J.L., Alves Sobrinho, T., 2014. Estimation on time of concentration of overland flow in watersheds: A review. Geociencias 33, 661–671.

del Granado, S., Stewart, A., Borbor, M., Franco, C., Tauzer, E., Romero, M., 2016. Sistemas de Alerta Temprana para Inundaciones: Análisis Comparativo de Tres Países Latinoamericanos.

686 Fernández de Córdova Webster, C., Javier Rodríguez López, Y., 2016. Primeros resultados de la red actual de
687 monitoreohidrometeorológico de Cuenca, Ecuador. Ing. Hidráulica y Ambient. 37, 44–56.
688 Furquim, G., Neto, F., Pessin, G., Ueyama, J., Joao, P., Clara, M., Mendiondo, E.M., de Souza, V.C.B., de
689 Souza, P., Dimitrova, D., others, 2014. Combining wireless sensor networks and machine learning for
690 flash flood nowcasting, in: 2014 28th International Conference on Advanced Information Networking
691 and Applications Workshops. pp. 67–72.
692 Galelli, S., Castelletti, A., 2013. Assessing the predictive capability of randomized tree-based ensembles in
693 streamflow modelling. Hydrol. Earth Syst. Sci. 17, 2669–2684.
694 Gu, Q., Zhu, L., Cai, Z., 2009. Evaluation measures of the classification performance of imbalanced data sets.
695 Commun. Comput. Inf. Sci. 51, 461–471. https://doi.org/10.1007/978-3-642-04962-0_53
696 Hossin, M., Sulaiman, M.N., 2015. A review on evaluation metrics for data classification evaluations. Int. J.
697 Data Min. Knowl. Manag. Process 5, 1.
698 Hundecha, Y., Parajka, J., Viglione, A., 2017. Flood type classification and assessment of their past changes
699 across Europe. Hydrol. Earth Syst. Sci. Discuss. 1–29.
700 Khosravi, K., Shahabi, H., Thai, B., Adamowski, J., Shirzadi, A., 2019. A comparative assessment of flood
701 susceptibility modeling using Multi- Criteria Decision-Making Analysis and Machine Learning Methods.
702 J. Hydrol. 573, 311–323. https://doi.org/10.1016/j.jhydrol.2019.03.073
703 Knocke, E.T., Kolivras, K.N., 2007. Flash flood awareness in southwest Virginia. Risk Anal. An Int. J. 27, 155–
704 169.
705 Li, Y., Grimaldi, S., Walker, J.P., Pauwels, V., 2016. Application of remote sensing data to constrain operational
706 rainfall-driven flood forecasting: a review. Remote Sens. 8, 456.
707 Loumagne, C., Normand, M., Riffard, M., Weisse, A., Quesney, A., Hegarat-Mascle, S. Le, Alem, F., 2001.
708 Integration of remote sensing data into hydrological models for reservoir management. Hydrol. Sci. J. 46,
709 89–102.
710 Maier, H.R., Dandy, G.C., 2000. Neural networks for the prediction and forecasting of water resources
711 variables: a review of modelling issues and applications. Environ. Model. Softw. 15, 101–124.
712 Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P., 2010. Methods used for the development of neural networks
713 for the prediction of water resource variables in river systems: Current status and future directions.
714 Environ. Model. Softw. 25, 891–909.
715 Min, S.K., Zhang, X., Zwiers, F.W., Hegerl, G.C., 2011. Human contribution to more-intense precipitation
716 extremes. Nature 470, 378–381. https://doi.org/10.1038/nature09763
717 Mirza, M.M.Q., 2011. Climate change, flooding in South Asia and implications. Reg. Environ. Chang. 11, 95–
718 107.
719 Mosavi, A., Ozturk, P., Chau, K.W., 2018. Flood prediction using machine learning models: Literature review.
720 Water (Switzerland) 10, 1–40. https://doi.org/10.3390/w10111536
721 Muñoz, P., Célleri, R., Feyen, J., 2016. Effect of the Resolution of Tipping-Bucket Rain Gauge and Calculation
722 Method on Rainfall Intensities in an Andean Mountain Gradient. Water 8, 534.
723 Muñoz, P., Orellana-Alvear, J., Willems, P., Célleri, R., 2018. Flash-flood forecasting in an andean mountain
724 catchment-development of a step-wise methodology based on the random forest algorithm. Water
725 (Switzerland) 10. https://doi.org/10.3390/w10111519
726 Paprotny, D., Sebastian, A., Morales-Nápoles, O., Jonkman, S.N., 2018. Trends in flood losses in Europe over
727 the past 150 years. Nat. Commun. 9, 1985.
728 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P.,
729 Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay,
730 E., 2011. Scikit-learn: Machine Learning in {P}ython. J. Mach. Learn. Res. 12, 2825–2830.
731 Raschka, S., 2018. Model evaluation, model selection, and algorithm selection in machine learning. arXiv Prepr.
732 arXiv1811.12808.
733 Read, J., Pfahringer, B., Holmes, G., Frank, E., 2011. Classifier chains for multi-label classification. Mach.
734 Learn. 85, 333.
735 Sofia, G., Roder, G., Dalla Fontana, G., Tarolli, P., 2017. Flood dynamics in urbanised landscapes: 100 years
736 of climate and humans' interaction. Sci. Rep. 7, 1–12. https://doi.org/10.1038/srep40527
737 Solomatine, D.P., Xue, Y., 2004. M5 model trees and neural networks: application to flood forecasting in the
738 upper reach of the Huai River in China. J. Hydrol. Eng. 9, 491–501.
739 Sottolichio, A., Hurther, D., Gratiot, N., Bretel, P., 2011. Acoustic turbulence measurements of near-bed
740 suspended sediment dynamics in highly turbid waters of a macrotidal estuary. Cont. Shelf Res.
741 https://doi.org/10.1016/j.csr.2011.03.016

Stefanidis, S., Stathis, D., 2013. Assessment of flood hazard based on natural and anthropogenic factors using analytic hierarchy process (AHP). Nat. Hazards 68, 569–585. https://doi.org/10.1007/s11069-013-0639-5

Sudheer, K.P., Gosain, A.K., Ramasastri, K.S., 2002. A data-driven algorithm for constructing artificial neural network rainfall-runoff models. Hydrol. Process. 16, 1325–1330. https://doi.org/10.1002/hyp.554

Sun, Y., Wong, A.K.C., Kamel, M.S., 2009. Classification of imbalanced data: A review. Int. J. Pattern Recognit. Artif. Intell. 23, 687–719.

Toukourou, M., Johannet, A., Dreyfus, G., Ayral, P.-A., 2011. Rainfall-runoff modeling of flash floods in the absence of rainfall forecasts: the case of "Cévenol flash floods." Appl. Intell. 35, 178–189.

Turkington, T., Breinl, K., Ettema, J., Alkema, D., Jetten, V., 2016. A new flood type classification method for use in climate change impact studies. Weather Clim. Extrem. 14, 1–16.

Young, P.C., 2002. Advances in real--time flood forecasting. Philos. Trans. R. Soc. London. Ser. A Math. Phys. Eng. Sci. 360, 1433–1450.

Zadrozny, B., Elkan, C., 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers, in: Icml. pp. 609–616.

Zhang, H., 2004. The optimality of naive Bayes. AA 1, 3.

Figures

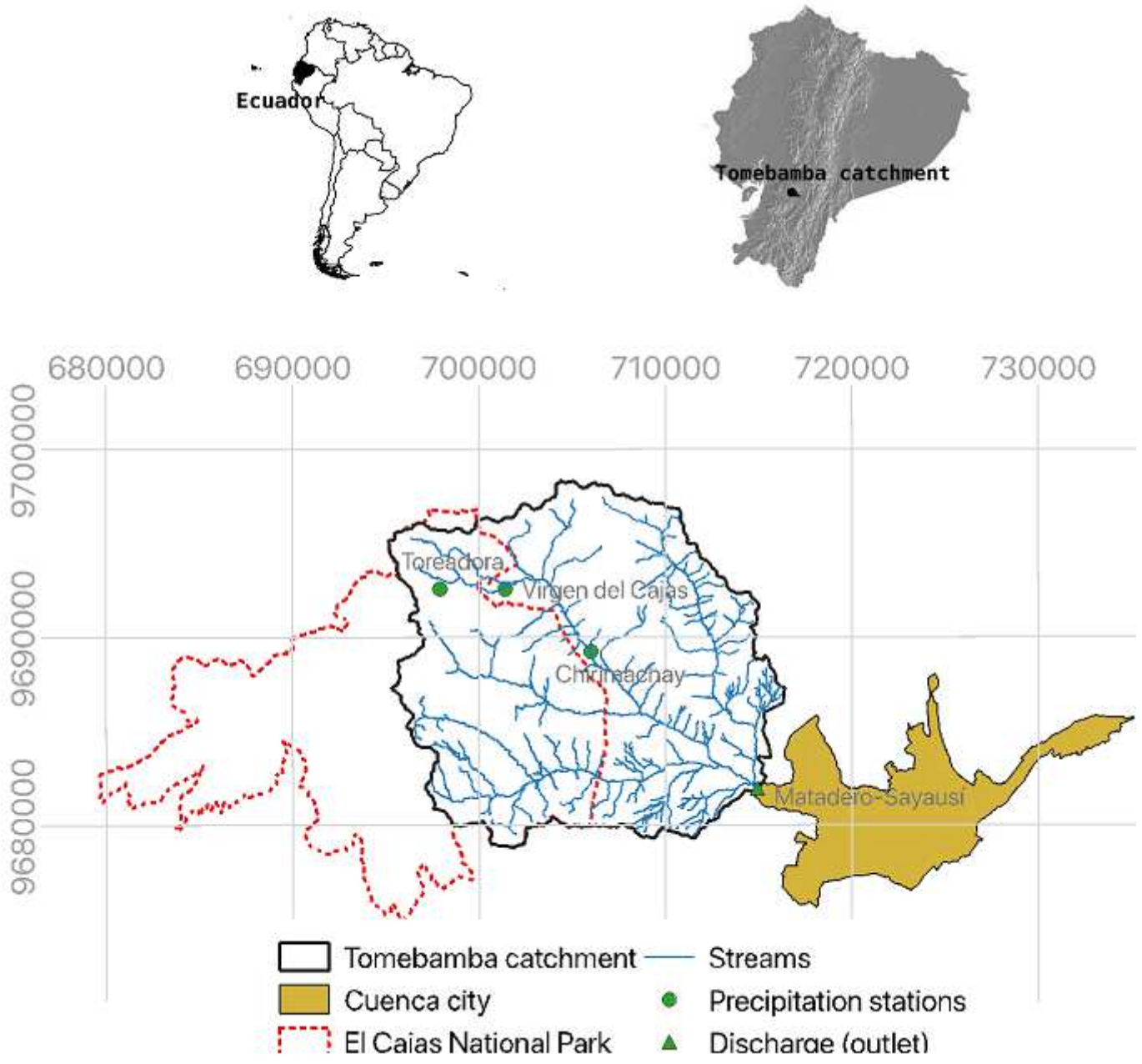


Figure 1

Location of the Tomebamba catchment at the Tropical Andean Cordillera of Ecuador, South America (UTM coordinates). Note: The designations employed and the presentation of the material on this map do not imply the expression of any opinion whatsoever on the part of Research Square concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. This map has been provided by the authors.

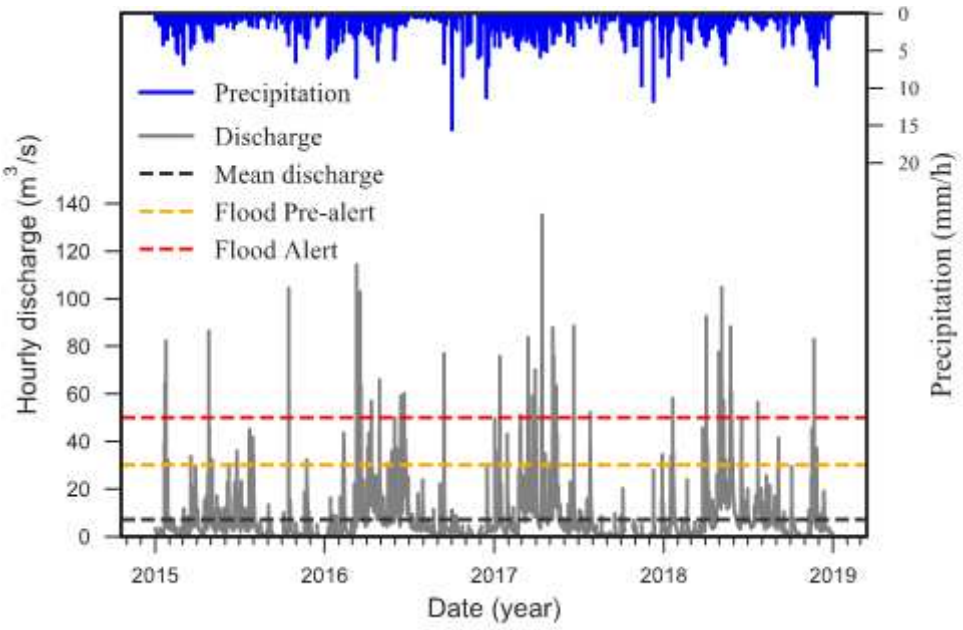


Figure 2

Time series of precipitation (Toreadora) and discharge (Matadero-Sayausí). Horizontal dashed lines indicate the mean runoff and the currently employed flood alert levels for labeling the Pre-alert and Alert flood warnings classes.

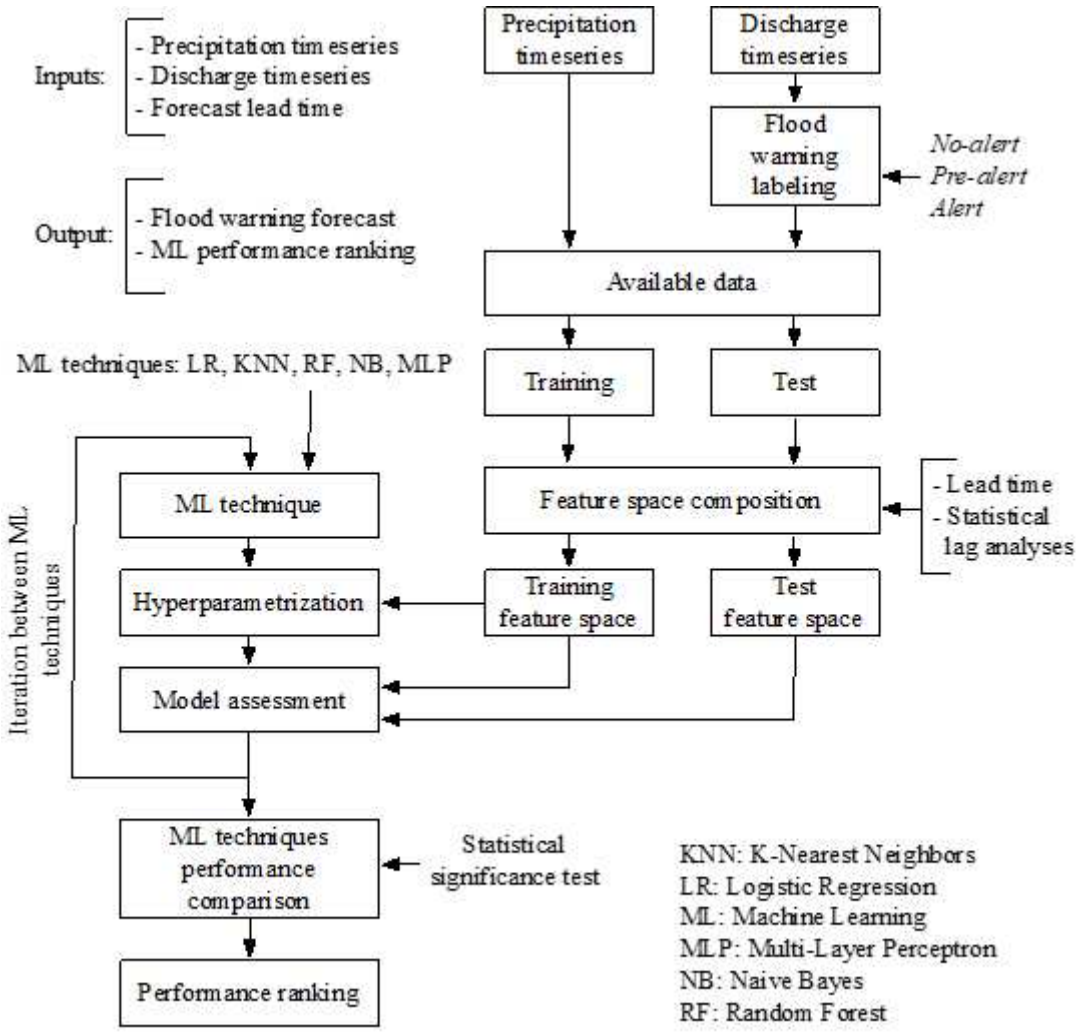


Figure 3

Workflow for developing and testing the ML flood forecasting models.

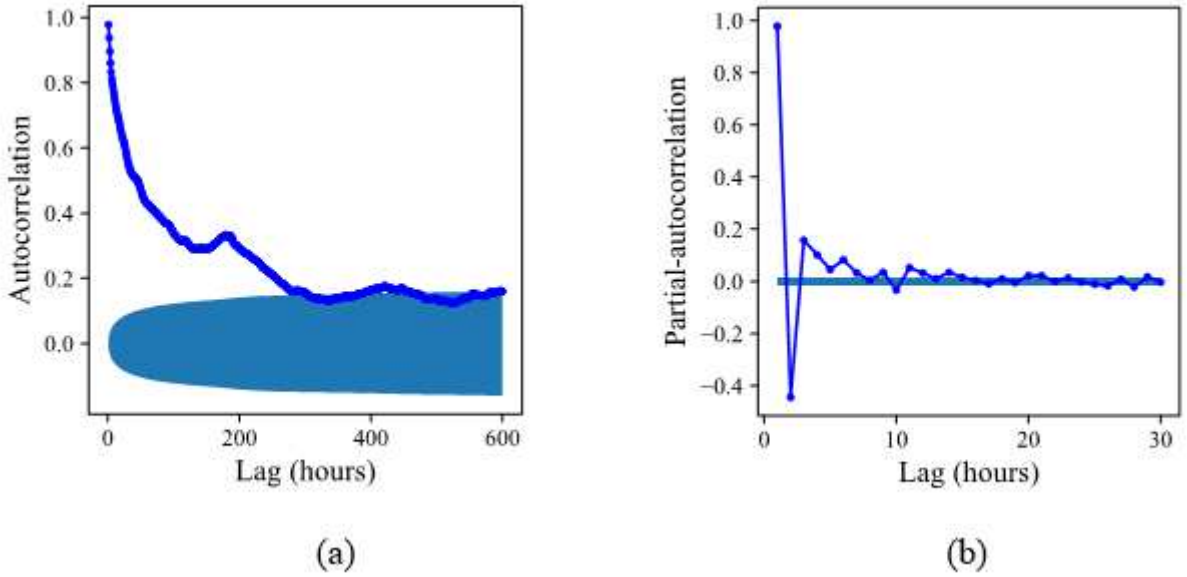


Figure 4

(a) Autocorrelation function (ACF) and (b) Partial-autocorrelation function (PACF) of the Matadero-Sayausí (Tomebamba catchment) discharge series. The blue hatch indicates in each case the correspondent 95% confidence interval.

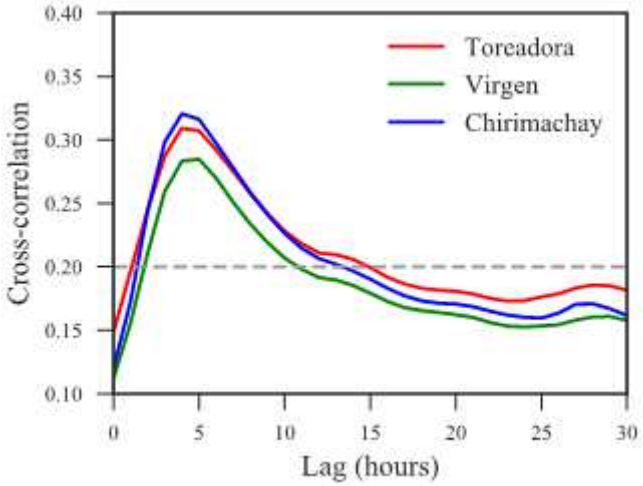


Figure 5

Pearson's cross-correlation comparison between the Toreadora (3955 m a.s.l), Virgen (3626 m a.s.l.) and Chirimachay (3298 m a.s.l.) precipitation stations and the Matadero-Sayausí discharge series. Note the blue horizontal line at a fixed correlation of 0.2 for determining past lags.

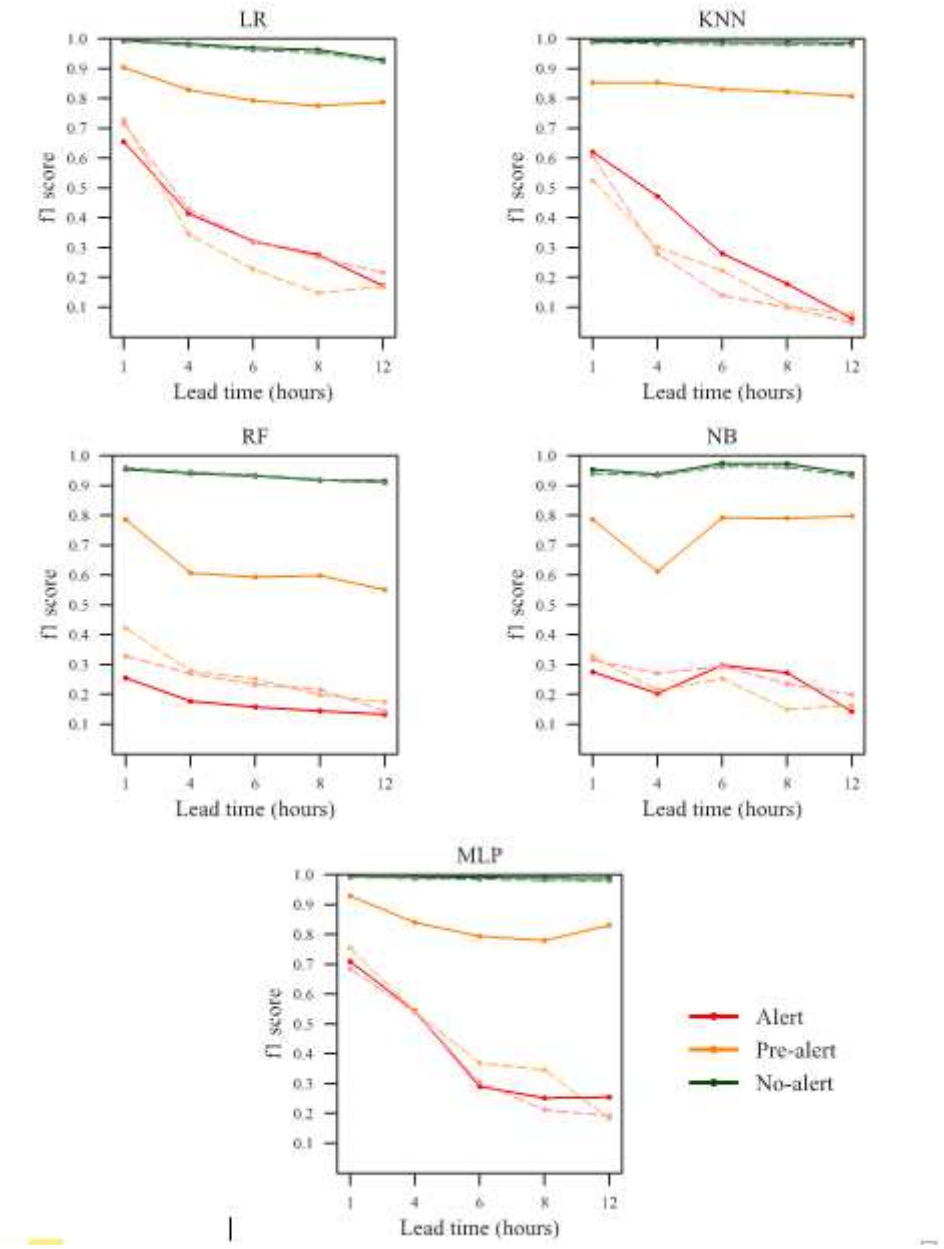


Figure 6

F1 scores per flood warning state (No-alert, Pre-alert and Alert) for all combinations of ML techniques and lead times. The brightest and dashed lines in each case (color coding) represent the scores for the test subset.