Article

Flood Early Warning Systems using Machine Learning techniques. Case the Tomebamba catchment at the southern Andes of Ecuador

Paul Muñoz[1,2,*], Johanna Orellana-Alvear[1,2], Jörg Bendix[3], Jan Feyen[4] and Rolando Célleri[1,2]

[1] Departamento de Recursos Hídricos y Ciencias Ambientales, Universidad de Cuenca, Cuenca 010150, Ecuador; paul.munozp@ucuenca.edu.ec, johanna.orellana@ucuenca.edu.ec, and rolando.celleri@ucuenca.edu.ec
[2] Facultad de Ingeniería, Universidad de Cuenca, Cuenca 010150, Ecuador
[3] Laboratory for Climatology and Remote Sensing, Faculty of Geography, University of Marburg, 35032 Marburg, Germany; bendix@mailer.uni-marburg.de
[4] Faculty of Bioscience Engineering, Catholic University of Leuven, Leuven 3001, Belgium; jan.feyen@kuleuven.be
* Correspondence: paul.munozp@ucuenca.edu.ec

Abstract: Flood Early Warning Systems (FEWSs) using Machine Learning (ML) has gained worldwide popularity. However, determining the most efficient ML technique is still a bottleneck. We assessed FEWSs with three river states, No-alert, Pre-alert, and Alert for flooding, for lead times between 1 to 12 hours using the most common ML techniques, such as Multi-Layer Perceptron (MLP), Logistic Regression (LR), K-Nearest Neighbors (KNN), Naive Bayes (NB), and Random Forest (RF). The Tomebamba catchment in the tropical Andes of Ecuador was selected as case study. For all lead times, MLP models achieve the highest performance followed by LR, with f1-macro (log-loss) scores of 0.82 (0.09) and 0.46 (0.20) for the 1- and 12-hour cases, respectively. The ranking was highly variable for the remaining ML techniques. According to the g-mean, LR models correctly forecast and show more stability at all states, while the MLP models perform better in the Pre-alert and Alert states. Future efforts are recommended to enhance the input data representation and develop communication applications to boost the awareness of the society for floods.

Keywords: Flood Early Warning; forecasting; hydrological extremes; Machine Learning; Andes.

1. Introduction

Flooding is the most common natural hazard and produces worldwide one of the most damaging disasters [1]–[4]. Recent studies associate the increasing frequency and severity of flood events with a change in land use (e.g., deforestation and urbanization) and climate [2], [5]–[7]. This particularly holds for the tropical Andes region, where complex hydro-meteorological conditions result in the occurrence of intense and patchy rainfall events [8]–[10].

According to the flood generation mechanism can floods be classified into long- and short-rain floods [11], [12]. A key for building resilience to floods is to anticipate timely to the event, as to gain time for better preparedness. The response time between a rainfall event and its associated flood depends on the catchment properties and might vary from minutes to hours [13]. In this study special attention is given to flash-floods, which are floods that develop less than six hours after a heavy rainfall with little or no forecast lead time [14].

Flood anticipation can be achieved through the development of a Flood Early Warning System (FEWS). FEWSs have proved to be cost-efficient solutions for life preservation, damage mitigation, and resilience enhancement [15]–[17]. However, although crucial, flood forecasting remains a major challenge in mountainous regions due to the difficulty to effectively record the aerial distribution of precipitation due to the

sparse density of the monitoring network and the absence of high-tech equipment by budget constraints [8], [9].

To date, there is no report of any operational FEWS in the Andean region for scales other than continental [17]–[19]. An alternative attempt in Peru aimed to derive daily maps of potential floods based on the spatial cumulated precipitation in the past days [20]. Other endeavors in Ecuador and Bolivia focused on the monitoring of the runoff in the upper parts of the catchment to predict the likelihood of flood events in the downstream basin area [18], [21]. However, such attempts are unsatisfactory as countermeasures against floods and especially flash-floods, where it is required to have reliable and accurate forecasts with lead times shorter than the response time between the farthest precipitation station and runoff control point.

There are two paradigms that drive the modeling of the precipitation-runoff response. First, the physically-based paradigm includes knowledge of the physical processes by using physical process equations [22]. This approach requires extensive ground data and in consequence intensive computation that hinders the temporal forecast window [23]. 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 and over-parametrization of this type of models [24]. The second data-driven paradigm assumes floods as stochastic processes with an occurrence distribution probability 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 the traditional data-driven approaches, statistical modeling has proven to be unsuitable for short-term prediction due to lack of accuracy, complexity, model robustness, and even computational costs [23]. Previous encouraged the use of advanced data-driven models, e.g., machine learning (ML), to overcome the aforementioned shortcomings [7], [23], [25], [26]. Particularly during the last decade, ML approaches have gained increasing popularity among hydrologists [23].

Different ML strategies for flood forecasting are implemented, generating either a quantitative or qualitative runoff forecast [27]–[33]. Qualitative forecasting consists of classifying floods into distinct categories or river states according to their severity (i.e., runoff magnitude), and use this as a base for flood class prediction [34]. The advantage of developing a FEWS is the possibility to generate a semaphore-like warning system that is easy to understand by decision-makers and the public (non-hydrologists). The challenge of FEWSs is the selection of the most optimal ML technique to obtain reliable and accurate forecasts with sufficient lead time for decision making. To date, the problem has received scant attention in the research literature, and as far as our knowledge extends no previous work examined and compared the potential and efficacity of different ML techniques for flood forecasting.

The present study compares the performance of five ML classification techniques for short-rain flood forecasting with special attention to flash floods. ML models were developed for a medium-size mountain catchment, the Tomebamba basin located in the tropical Andes of Ecuador. The ML models were tested with respect to their capacity to forecast three flood warning stages (No-alert, Pre-alert, and Alert) for varying forecast lead times of 1, 4, and 6 hours (flash-floods), but also 8 and 12 hours to further test whether the lead time can be satisfactorily extended without losing models' operational value.

2. Materials and Methods

2.1 Study area and dataset

The study area comprises the Tomebamba catchment delineated upstream of the Matadero-Sayausí hydrological station of the Tomebamba river (Figure 1), where the river enters the city. The Tomebamba is a tropical mountain catchment located in the

 doi:10.20944/preprints202111.0510.v1

southeastern flank of the Western Andean Cordillera, draining to the Amazon River. The drainage area of the catchment is approximately 300 km2, spanning from 2800 to 4100 meters above sea level (m asl). Like many other mountain catchments of the region, it is primarily covered by a páramo ecosystem, which is known for its important water regulation function [8].

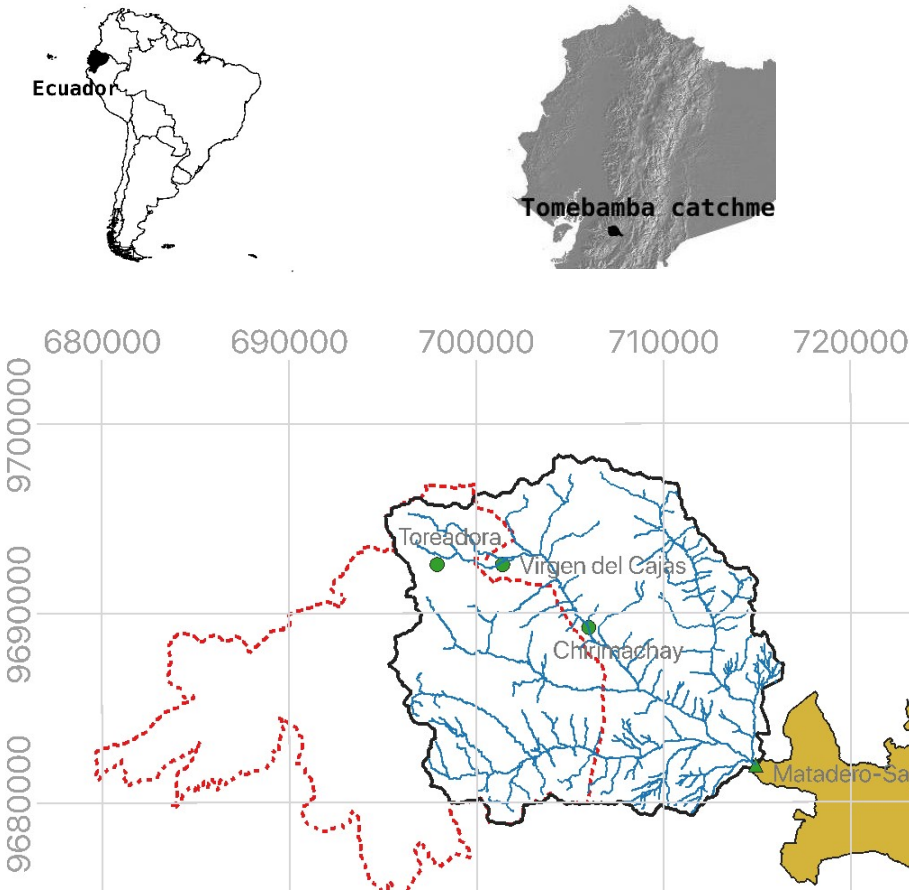


Figure 1. Location of the Tomebamba catchment at the Tropical Andean Cordillera of Ecuador, South America (UTM coordinates).

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

The local water utility, the Municipal Public Company of Telecommunications, Water, Sewerage and Sanitation of Cuenca (ETAPA-EP), defined three flood alert levels associated with the Matadero-Sayausí station for floods originated in the Tomebamba catchment: i) No-alert of flood occurs when the measured runoff is less than 30 m^3/s, ii) Pre-alert when runoff is between 30 and 50 m^3/s, and iii) the flood Alert is triggered when discharge exceeds 50 m^3/s. With these definitions, and as shown in Figure 2, the discharge label for the No-alert class represents the majority of the data, whereas the Pre-alert and Alert classes comprise the minority yet the most dangerous classes.

To develop and operate forecasting models, we use data of two variables: precipitation in the catchment area and river discharge at a river gauge. For both variables, the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 November 2021 doi:10.20944/preprints202111.0510.v1

available dataset comprises 4 years of concurrent hourly time series, from Jan/2015 to Jan/2019 (Figure 2). Precipitation information was derived from 3 tipping-bucket rain gauges: Toreadora (3955 m a.s.l.), Virgen (3626 m a.s.l.), and Chirimachay (3298 m a.s.l.) installed within the catchment and along its altitudinal gradient. Whereas for discharge, we used data of the Matadero-Sayausí station (2693 m a.s.l., Figure 1). To develop the ML modes, we split the dataset into training and test subsets. The training period ran from 2015 to 2017, whereas 2018 was used as the model testing phase.

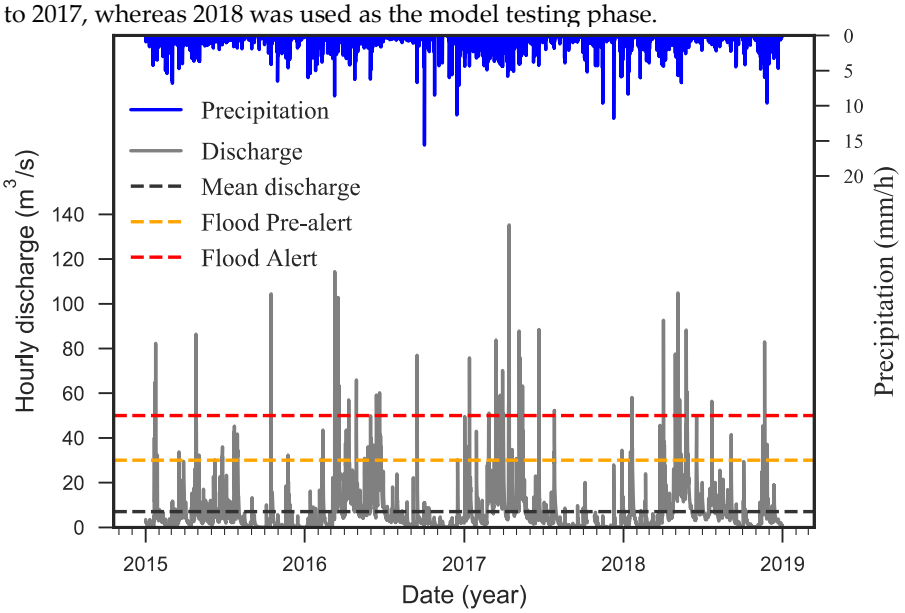


Figure 2. Time series of precipitation (Toreadora) and discharge (Matadero-Sayausí). Horizontal dashed lines indicate the mean runoff and the currently employed flood alert levels for labeling the Pre-alert and Alert flood warnings classes.

2.2 Machine Learning (ML) methods for classification of flood alert levels

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

i. Regression algorithms modeling the relationships between variables (e.g., Logistic Regression, Linear Regression, Multivariate Adaptive Regression Splines, etc.).
ii. Instance-based algorithms that rely on memory-based learning, representing a decision problem fed with 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 values, 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 convert input(s) to output(s) through specified transient states that enable 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, respectively a Logistic Regression, K-Nearest Neighbor, Random Forest, Naive Bayes, and a Multi-layer Perceptron.

Logistic Regression

Logistic Regression (LR) is a discriminative model, modeling the decision boundary between classes. In a first instance, linear regressions are applied to find existent relationships between model features. Thereafter, the probability (conditional) of belonging to a class is identified using 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 created classes. However, for multiclass classification problems are all binary classification possibilities considered, it is No-alert vs. Pre-alert, No-alert vs. Alert, and Pre-alert vs. Alert. Finally, the solution is the classification with the maximum probability (multinomial LR) using the softmax function (equation 1). With this function is the predicted probability of each class defined [35]. The calculated probability for each class is positive with the logistic function and normalized across all classes.

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

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

K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a non-parametric statistical pattern recognition algorithm, for which no theoretical or analytical background exist but an intuitive statistical procedure (memory-based learning) for the classification. KNN classifies unseen data based on a similarity measure such as a distance function (e.g., Euclidean, Manhattan, Chebyshev, Hamming, etc.). The use of multiple neighbors instead of only one is recommended to avoid the wrong delineation of class boundaries caused by noisy features. In the end, the majority vote of the nearest neighbors (see the formulation in [35]) determines the classification decision. The number of nearest neighbors can be optimized to reach a global minimum avoiding longer computation times, and the influence of class size. The major advantage of the KNN is its simplicity. However, the drawback is that KNN is memory intensive, all training data must be stored and compared when added information is to be evaluated.

Random Forest

Random Forest (RF) is a supervised ML algorithm that ensembles a multitude of decorrelated decision trees (DTs) voting for the most popular class (classification). In practice, a DT (particular model) is a hierarchical analysis based on a set of conditions consecutively applied to a dataset. To assure decorrelation, the RF algorithm applies a bagging technique for a growing DT from different randomly resampled training subsets obtained from the original dataset. Each DT provides an independent output (class) of the phenomenon of interest (i.e., runoff), contrary to numerical labels for regression applications. The popularity of RF is due to the possibility to perform random subsampling and bootstrapping which minimizes biased classification [36]. An extended description of the RF functioning is available in [37], [38].

The predicted class probabilities of an input sample are calculated as the mean predicted class probabilities of the trees in the forest. For a single tree, the class probability is computed as the fraction of samples of the same class in a leaf. However, it is well-known that the calculated training frequencies are not accurate conditional probability estimates due to the high bias and variance of the frequencies [39]. This deficiency can be resolved by controlling the minimum number of samples required at a leaf node, with the objective to induce a smoothing effect, and to obtain statistically reliable probability estimates.

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, even if there is dependence [40]. Bayes' theorem can be expressed as:

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

where P(A|B) is the probability of y (hypothesis) happening, given the occurrence of X (features), and X can be defined as X=x_1,x_2,…,x_n. Based here on can Bayes' theorem be written as:

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(x_1|y)\,P(x_2|y) \ldots P(x_n|y)\,P(y)}{P(x_1)\,P(x_2) \ldots P(x_n)} \quad (3)$$

There are different NB classifiers depending on the assumption of the distribution of P(x_i |y). In this matter, the study of Zhang [40] 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 the Gaussian NB algorithm no parameters have to be tuned.

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. The MLP consists of multiple neurons in fully connected multiple layers. Determination of the number of neurons in the layers with a trial-and-error approach remains widely used [41]. Neurons in the first layer corresponding to the input data, whereas all other nodes relate inputs to outputs by using linear combinations with certain weights and biases together with an activation function. To measure the performance of the MLP, the logistic loss function is defined with the limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method as the optimizer for training the network. A detailed and comprehensive description of ANN can be found in [42].

2.3 Methodology

Figure 3 depicts schematic the methodology followed in this study. The complete dataset for the study consists, as mentioned before, of precipitation and labeled discharge time-series (see Figure 2). The dataset was split in two groups, respectively for training and testing purposes, and training and test feature spaces were composed for each lead time for the tasks of model hyperparameterization and model assessment. This procedure is repeated for each of the ML techniques studied. Finally, the ranking of the performance quality of all ML methods for every lead time, based on performance metrics and a statistical significance test, were determined.

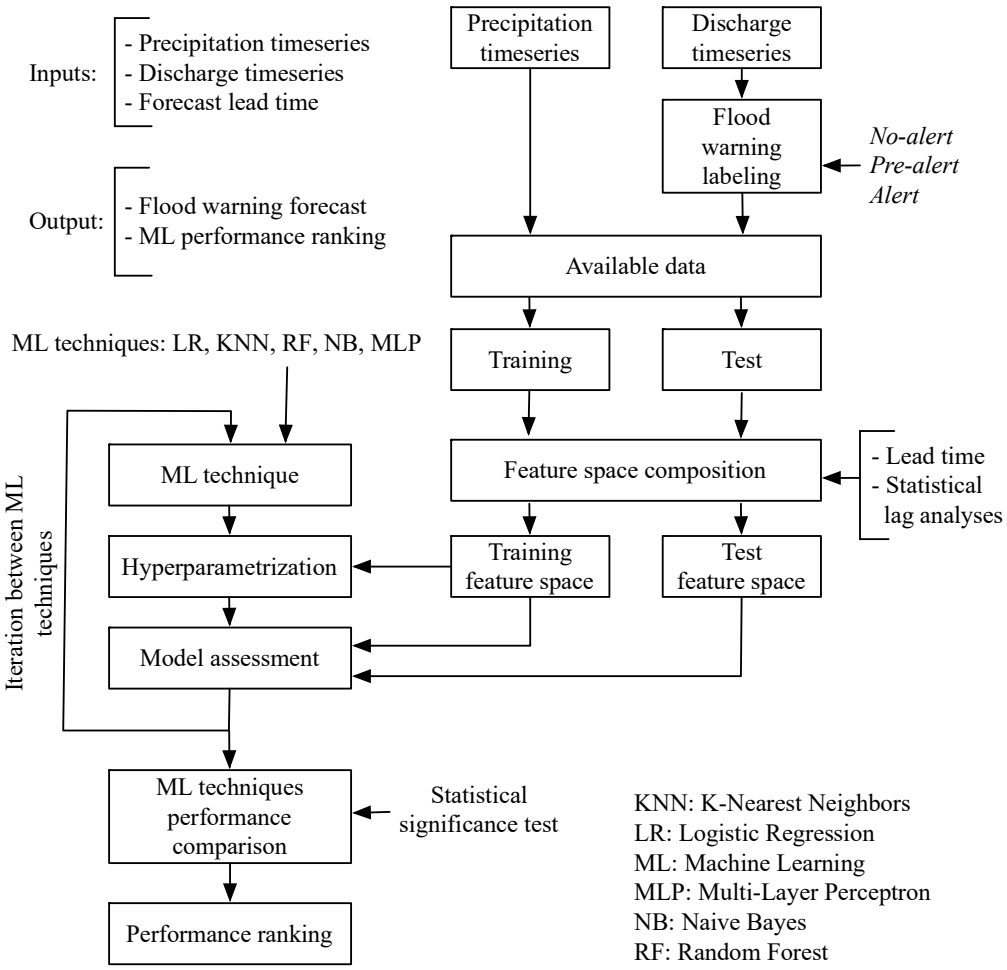


Figure 3. Work schedule for the development and testing of the ML flood forecasting models.

2.3.1 Feature space composition

For each lead time, we used single training and test feature spaces for all ML techniques. A feature space is composed 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 [43]. The number of lags from each station was selected by setting up a correlation threshold of 0.2 [27].

Similarly, for discharge, we used several 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 with 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 before the computation stage of the KNN, LR, NB, and NN algorithms. Standardization was achieved by subtracting the mean and scaling it to unit variance, resulting in a distribution with a standard deviation equal to 1 and a mean equal to 0.

2.3.2 Model hyperparameterization

After the composition of the feature space the optimal architectures for each ML forecasting model, and for each lead time was set up. The optimal architectures were de-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 November 2021 doi:10.20944/preprints202111.0510.v1

fined by the combination of hyperparameters under the concept of balance between accuracy, and computational cost, and speed. However, finding optimal architectures requires an exhaustive search of all 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 based on a cross-validation evaluation. Moreover, we selected the f1-macro score (see section 2.3.4) as objective function.

2.3.3 Principal Component Analysis

ML applications require in general the analysis of high-dimension and complex data, involving substantial amounts of memory and computational costs. Reduction of the dimensionality was realized through the application of Principal Component Analysis (PCA) enabling exclusion of correlating features that do not add information to the model. PCA was applied after feature scaling and normalization.

This method enables finding the dimension of maximum variance and the reduction of the feature space to that dimension so that the model performance remains as intact as possible when compared to performance with the full feature space. But considering that each ML technique assimilates data differently, we did not define the number of principal components to keep a fixed threshold of variance explanation (e.g., 80-90%), but performed an exploratory analysis to evaluate its influence on each model. As such, the number of PCAs was treated as an additional hyperparameter, and we optimized the number of principal components for each specific model (lead time and ML technique) with the objective 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® [44]. Table 1 presents the relevant hyperparameters for each ML technique and their search space for tuning [45]. We employed default values for the hyperparameters which are depicted in Table 1.

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

ML technique	Hyperparameters				
LR	*C*	*penalty*			
	0.001 - 1000	{'l1','l2'}			
KNN	*neighbor's*	*weights*	*metric*	*algorithm*	
	3 - 75	{'uniform', 'distance'}	{'euclidean', 'manhattan', 'minkowski'}	{'auto','ball_tree', 'kd_tree','brute'}	
RF	*estimator's*	*max_features*	*hadeeth*	*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	

2.3.4 Model performance evaluation

Forecasting hydrological extremes such as floods turns into an imbalanced classification problem, and becomes even more complex when the interest lies in the minority class of the data (flood alert). This is because most ML classification algorithms focus on the minimization of the overall error rate, it is the incorrect classification of the majority class [46]. 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 that 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 un-

 doi:10.20944/preprints202111.0510.v1

der-sampling or over-sampling data. In practice, this implies that for a given metric efficiency (see section 5.1), the overall score is the result of averaging each performance metric (for each class) multiplied by its corresponding weight factor. According to the class frequencies were the weight factors for each class calculated (inversely proportional), using equation 4.

$$w_i = \frac{N}{C\, n_j} \tag{4}$$

where w_i is the weight of class i, N is the total number of observations, C is the number of classes, and n_j the number of observations in class i. This implies that higher weights will be obtained for minority classes.

Performance metrics

The metrics for the performance assessment were derived from the well-known confusion matrix, especially suitable for imbalanced datasets and multiclass problems, and are respectively the f1 score, the geometric mean, and the logistic regression loss score [46]–[51]. Since neither of the metrics is adequate it is suggested to use a compendium of metrics to properly explain the performance of the model. In addition, those metrics complement each other.

F1 score

The f score is a metric that relies on precision and recall, which is an effective metric for imbalanced problems. When the f score as a weighted harmonic mean, we name this score f1. The latter score can be calculated with Equation 5.

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

where precision and recall are defined with the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP \mid FN} \tag{7}$$

where TP stands for True Positives, FP for False Positives, and FN for False Negatives.

The f1 score ranges from 0 to 1, indicating perfect precision and recall. The advantage of using the f1 score compared to the arithmetic or geometric mean is that it penalizes models most when either the precision or recall is low. However, classifying a *No-Alert* flood 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, e.g., *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, and is called the f1-macro score to indicate the overall model performance.

Geometric mean

The geometric-mean (g-mean) measures simultaneously the balanced performance of TP and True Negative (TN) rates. This metric 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 to balance TP and TN examples at the same time with a good trade-off [48]. It can be calculated using Equation 8.

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 November 2021 doi:10.20944/preprints202111.0510.v1

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

$$TP_{rate} = Recall \tag{9}$$

$$TN_{rate} = \frac{TN}{TN + FP} \tag{10}$$

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

Logistic regression loss

The metric 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:

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

where N is the number of samples, M the number of classes, y_{ij} equal to 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 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.

Statistical significance test for comparing ML algorithms

Although we can directly compare performance metrics of 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. Different statistical frameworks are available allowing to compare the performance of classification models (e.g., a difference of proportions, paired comparison, binomial test, etc.).

Among them, Raschka [52] recommends using 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 aims 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 due to a difference in the models and not due to statistical chance. In our study we used the chi-squared test to assess whether the difference in the observed proportions of the contingency tables of a pair of ML algorithms (for a given lead time) is significant.

For the model comparison, we defined the 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, the MLP model was used as the base model to which the other models were compared.

3. Results

This section presents 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 flood warnings, *No-alert*, *Pre-alert*, and *Alert*. First, 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 hyperparameteri-

 doi:10.20944/preprints202111.0510.v1

zation for all models, followed by an evaluation and ranking of the performance of the ML techniques.

3.1 Feature space composition

Figures 4 and 5 show the results of the discharge and precipitation lag analyses for the flood forecasting model 1-hour before the flood would occur. Figure 4(a) depicts 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, and according to Muñoz et al. [27] we decided to include 8 discharge lags (hours) for the case of 1-hour flood forecasting in the Tomebamba catchment.

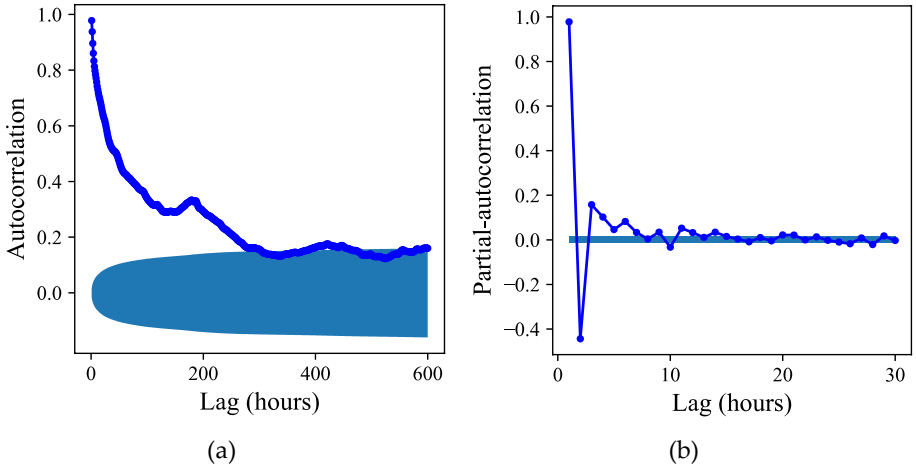
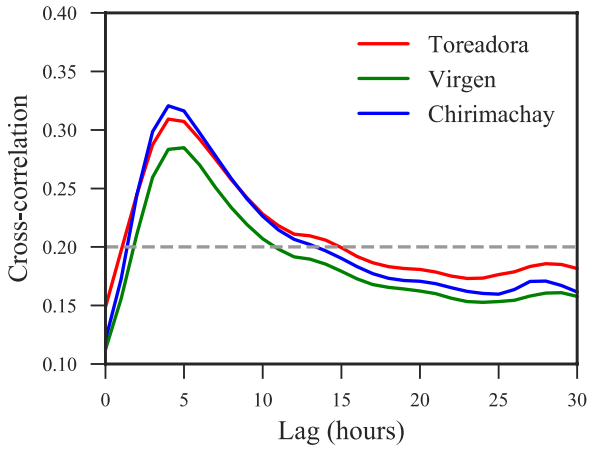


Figure 4. (a) Autocorrelation function (ACF) and (b) Partial-autocorrelation function (PACF) of the Matadero-Sayausí (Tomebamba catchment) discharge series. The blue hatch indicates in each case the correspondent 95% confidence interval.

Figure 5 plots the Pearson's cross-correlation between the precipitation at each rainfall station and the discharge at the Matadero-Sayausí stream gauging station. 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, Chirimachay, and Toreadora stations, respectively.



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 November 2021 doi:10.20944/preprints202111.0510.v1

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.

3.2 Model hyperparameterization

The results of the hyperparameterization including the number of PCA components employed for achieving the best model efficiencies are presented in Table 3. 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%.

If we turn to the evolution of models' complexity with lead time (Table 3) more complex ML architectures are needed to forecast greater lead times. This is underpinned by the fact that the corresponding optimal models require 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	'minkow-	'minkow-	'minkow-	'minkow-	'minkow-

		ski'	ski'	ski'	ski'	ski'
	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
MLP	*solver*	'lbfgs'	'lbfgs'	'lbfgs'	'lbfgs'	'lbfgs'
	max_iter	2000	2000	2000	2000	2000
	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

3.3 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 respectively for 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. The number of samples and relative percentage for the entire dataset and 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%)	(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 rather than to the 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**

Note: All improvements and degradations are statistically significant

The extensive hyperparameterization (RGS scheme) powered by 10-fold cross-validation served to assure robustness in all ML models and reduced 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 score, 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, the 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 sensitive 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. Despite 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, while 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 unveils 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 to forecast when the f1-macro score was selected as the 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 comparable results for the *Alert* class, the scores of the *Pre-alert* class were highly deteriorated, even reaching scores near zero. The most interesting aspect in 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, a lack of robustness for the *Pre-alert* warning class was found, it is major differences between the f1-scores for the training and test subsets. An explanation for

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 November 2021 doi:10.20944/preprints202111.0510.v1

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 during training leading to poor performances when assessing unseen data during the test 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 most 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.

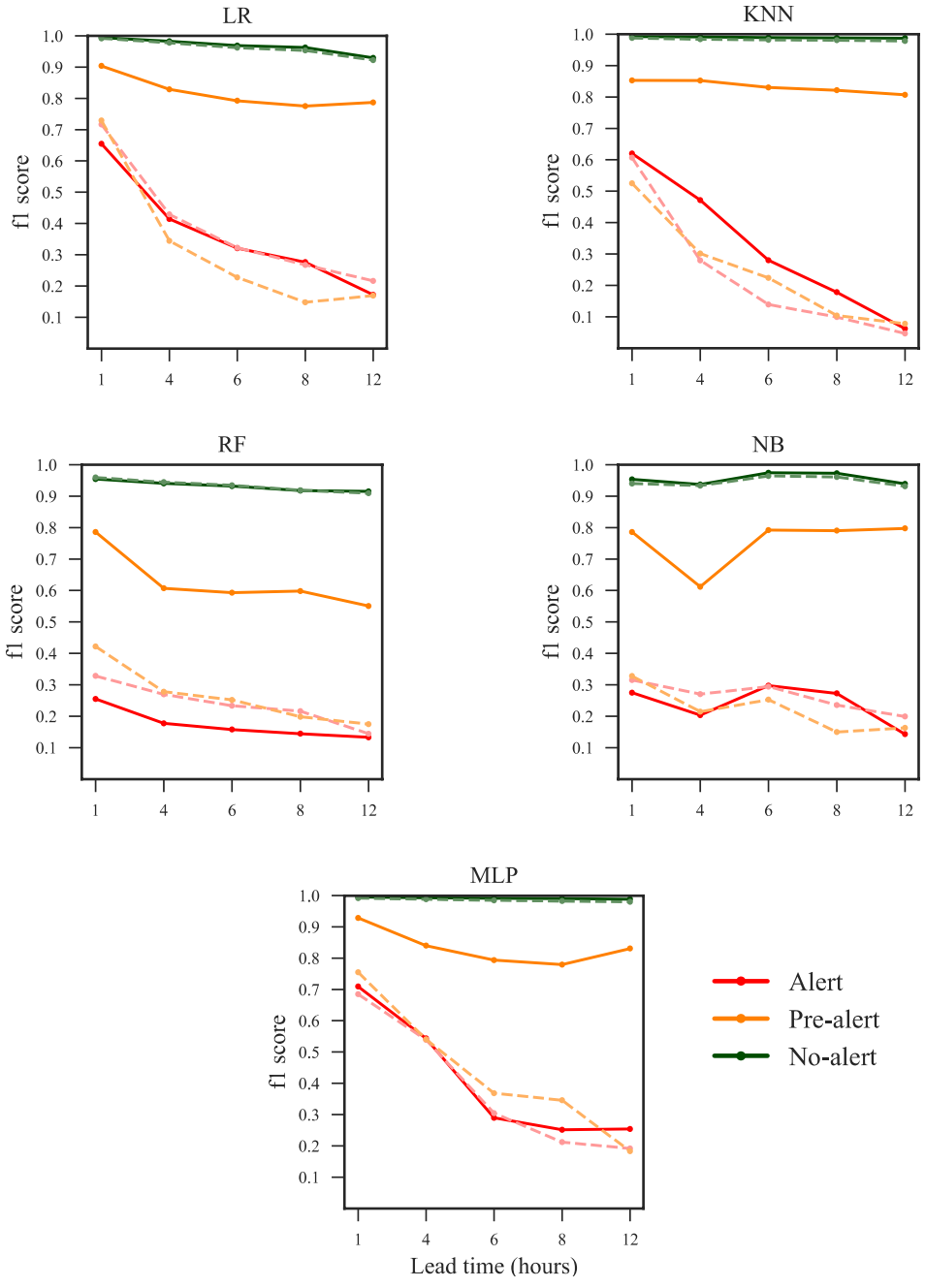


Figure 6. F1 scores per flood warning state (No-alert, Pre-alert, and Alert) for all combinations of ML techniques and lead times. The brightest and dashed lines in each case (color coding) represent the scores for the test subset.

4. 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 assess models' operational value for longer lead times. Historical runoff data were used 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 the statistical analyses of precipitation and discharge data followed by a PCA analysis embedded in the hyperparameterization.

This was aimed to better exploit the learning algorithm of each ML technique. In terms of model assessment, 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 analysis 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). 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.

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 forecasting models is limited by the models' operational value. This is because, in the absence of rainfall forecasts, the assumption of future rain is solely based on runoff measurements at past and present times. This generates forecasts that are not accurate enough for horizons greater than the concentration-time of the catchment. The concentration-time of the Tomebamba catchment was estimated between 2 and 6 hours according to the equations of Kirpich, Giandotti, Ven Te Chow, and Temez, respectively. A summary of the equations can be found in Almeida et al. [53]. This results in an additional performance decay for the 8 and 12-hour cases in addition to the error in modeling.

The study of Furquim et al. [28] is comparable. These authors analyzed the performance of different ML classification algorithms for flash-flood nowcasting (3 hours) in a river located in an urban area of Brazil. They found that models based on neural networks and decision trees outperformed the ones based on the Naive Bayes technique. However, this study only evaluated the percentage of correctly classified instances which is a simplistic evaluation. Thus, we recommend a more integral assessment of model performances, like the one in the current study, which allows for better support on decision making.

Other studies related to quantitative forecasting revealed that neural network-based models usually outperform the remaining techniques proposed in our study [29]–[31]. Nevertheless, in certain cases, the use of less expensive techniques regarding the computational costs produces comparable results as in [33]; this is also the case in our short-rain and flash-flood flood classification problem.

As a further step, we propose the development of ensemble models for improving the performance results of individual models. This can be accomplished by combining the outcomes of the ML models with weights obtained, for instance, from the log-log scores. Another alternative that is becoming popular is the construction of hybrid models as a combination of ML algorithms for more accurate and efficient models [23], [32], [33]. Moreover, as stated by Solomatine and Xue [33], inaccuracies in forecasting floods are

mainly due to data-related problems. In this regard, Muñoz et al. [9] reported a deficiency in precipitation-driven models due to rainfall heterogeneity in mountainous areas, where orographic rainfall formation occurs. In most cases, rainfall events are only partially captured by punctual measurement, and even the entire storm coverage can be missing.

In general precipitation-runoff models will reach at a certain point an effectiveness threshold that cannot be exceeded without incorporating new types of data such as soil moisture [54], [55]. In humid areas, the rainfall-runoff relation also depends on other variables such as evapotranspiration, soil moisture, and land use, which leads to significant spatial variations of water storage. However, these variables are difficult to measure or estimate.

5. Conclusions

The current study set out to develop and perform an integral performance evaluation of five of the most common ML classification techniques for short-rain flood forecasting, with special attention to flash floods. The developed models aimed at forecasting three flood warnings, *No-alert*, *Pre-alert*, and *Alert* for the Tomebamba catchment in the tropical Andes of Ecuador.

From the results, the following conclusions can be drawn:

- 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.
- 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.
- 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.
- 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.

We recommend that future efforts should be put on applying the methodology and assessment framework hereby proposed in other tropical Andean catchments, and/or benchmarking the results obtained in this study with the outputs of physically-based forecasting models. This was not possible for this study due to lack of data.

Finally, 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 that currently threaten people's lives, possessions, and environment in Cuenca and other comparable tropical Andean cities.

Author Contributions: Conceptualization, Paul Muñoz and Rolando Célleri; Formal analysis, Paul Muñoz; Funding acquisition, Rolando Célleri; Investigation, Paul Muñoz; Methodology, Paul Muñoz and Johanna Orellana-Alvear; Project administration, Rolando Célleri; Supervision, Johanna Orellana-Alvear, Jörg Bendix, Jan Feyen and Rolando Célleri; Writing – original draft, Paul Muñoz; Writing – review & editing, Johanna Orellana-Alvear, Jörg Bendix, Jan Feyen and Rolando Célleri.

Funding: This research 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 the Empresa Pública Municipal de Telecomunicaciones, Agua Potable, Alcantarillado y Saneamiento de Cuenca (ETAPA-EP). Our thanks go to these institutions for their generous funding.

Data Availability Statement: All data, models, and code that support the findings of this study are available from the corresponding author upon reasonable request.

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

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] S. Stefanidis and D. Stathis, "Assessment of flood hazard based on natural and anthropogenic factors using analytic hierarchy process (AHP)," *Nat. Hazards*, vol. 68, no. 2, pp. 569–585, 2013, doi: 10.1007/s11069-013-0639-5.

[2] D. Paprotny, A. Sebastian, O. Morales-Nápoles, and S. N. Jonkman, "Trends in flood losses in Europe over the past 150 years," *Nat. Commun.*, vol. 9, no. 1, p. 1985, 2018.

[3] Á. Ávila, F. C. Guerrero, Y. C. Escobar, and F. Justino, "Recent Precipitation Trends and Floods in the Colombian Andes," *Water*, vol. 11, no. 2, p. 379, 2019.

[4] M. M. Q. Mirza, "Climate change, flooding in South Asia and implications," *Reg. Environ. Chang.*, vol. 11, no. 1, pp. 95–107, 2011.

[5] G. Sofia, G. Roder, G. Dalla Fontana, and P. Tarolli, "Flood dynamics in urbanised landscapes: 100 years of climate and humans' interaction," *Sci. Rep.*, vol. 7, no. July 2016, pp. 1–12, 2017, doi: 10.1038/srep40527.

[6] S. K. Min, X. Zhang, F. W. Zwiers, and G. C. Hegerl, "Human contribution to more-intense precipitation extremes," *Nature*, vol. 470, no. 7334, pp. 378–381, 2011, doi: 10.1038/nature09763.

[7] L.-C. Chang *et al.*, "Building an Intelligent Hydroinformatics Integration Platform for Regional Flood Inundation Warning Systems." Multidisciplinary Digital Publishing Institute, 2019.

[8] R. Célleri and J. Feyen, "The Hydrology of Tropical Andean Ecosystems: Importance, Knowledge Status, and Perspectives," *Mt. Res. Dev.*, vol. 29, no. 4, pp. 350–355, 2009, doi: 10.1659/mrd.00007.

[9] P. Muñoz, R. Célleri, and J. Feyen, "Effect of the Resolution of Tipping-Bucket Rain Gauge and Calculation Method on Rainfall Intensities in an Andean Mountain Gradient," *Water*, vol. 8, no. 11, p. 534, 2016.

[10] P. A. Arias *et al.*, "Hydroclimate of the Andes Part II: Hydroclimate Variability and Sub-Continental Patterns," *Front. Earth Sci.*, vol. 8, p. 666, 2021.

[11] Y. Hundecha, J. Parajka, and A. Viglione, "Flood type classification and assessment of their past changes across Europe," *Hydrol. Earth Syst. Sci. Discuss.*, pp. 1–29, 2017.

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

[13] M. Borga, E. Gaume, J. D. Creutin, and L. Marchi, "Surveying flash floods: gauging the ungauged extremes," *Hydrol. Process.*, vol. 2274, no. 4, pp. 2267–2274, 2008, doi: 10.1002/hyp.7111.

[14] E. T. Knocke and K. N. Kolivras, "Flash flood awareness in southwest Virginia," *Risk Anal. An Int. J.*, vol. 27, no. 1, pp. 155–169, 2007.

[15] A. Sottolichio, D. Hurther, N. Gratiot, and P. Bretel, "Acoustic turbulence measurements of near-bed suspended sediment dynamics in highly turbid waters of a macrotidal estuary," *Cont. Shelf Res.*, 2011, doi: 10.1016/j.csr.2011.03.016.

[16] M. Borga, E. N. Anagnostou, G. Blöschl, and J. D. Creutin, "Flash flood forecasting, warning and risk management: The HYDRATE project," *Environ. Sci. Policy*, vol. 14, no. 7, pp. 834–844, 2011, doi: 10.1016/j.envsci.2011.05.017.

[17] S. del Granado, A. Stewart, M. Borbor, C. Franco, E. Tauzer, and M. Romero, "Sistemas de Alerta Temprana para Inundaciones: Análisis Comparativo de Tres Pa{\'\i}ses Latinoamericanos," 2016.

[18] D. Dávila, "21 experiencias de sistemas de alerta temprana en América Latina." Soluciones Prácticas, 2016.

[19] N. Boers, B. Bookhagen, H. M. J. Barbosa, N. Marwan, J. Kurths, and J. A. Marengo, "Prediction of extreme floods in the eastern Central Andes based on a complex networks approach," *Nat. Commun.*, vol. 5, no. 1, pp. 1–7, 2014.

[20] C. Aybar *et al.*, "Uso del Producto Grillado 'PISCO' de precipitación en Estudios, Investigaciones y Sistemas Operacionales de Monitoreo y Pronóstico Hidrometeorológico," *Nota Técnica*, vol. 1, 2017.

[21] C. Fernández de Córdova Webster and Y. Javier Rodríguez López, "Primeros resultados de la red actual de monitoreohidrometeorológico de Cuenca, Ecuador," *Ing. Hidráulica y Ambient.*, vol. 37, no. 2, pp. 44–56, 2016.

[22] M. P. Clark *et al.*, "The evolution of process-based hydrologic models : historical challenges and the collective quest for physical realism," no. 1969, pp. 3427–3440, 2017.

[23] A. Mosavi, P. Ozturk, and K. W. Chau, "Flood prediction using machine learning models: Literature review," *Water (Switzerland)*, vol. 10, no. 11, pp. 1–40, 2018, doi: 10.3390/w10111536.

[24] P. C. Young, "Advances in real--time flood forecasting," *Philos. Trans. R. Soc. London. Ser. A Math. Phys. Eng. Sci.*, vol. 360, no. 1796, pp. 1433–1450, 2002.

[25] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne, "Machine Learning Strategies for Time Series Forecasting.," in *eBISS*, 2012, pp. 62–77.

[26] S. Galelli and A. Castelletti, "Assessing the predictive capability of randomized tree-based ensembles in streamflow modelling," *Hydrol. Earth Syst. Sci.*, vol. 17, no. 7, pp. 2669–2684, 2013.

[27] P. Muñoz, J. Orellana-Alvear, P. Willems, and R. Célleri, "Flash-flood forecasting in an andean mountain catchment-development of a step-wise methodology based on the random forest algorithm," *Water (Switzerland)*, vol. 10, no. 11, 2018, doi: 10.3390/w10111519.

[28] G. Furquim *et al.*, "Combining wireless sensor networks and machine learning for flash flood nowcasting," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, 2014, pp. 67–72.

[29] M. Toukourou, A. Johannet, G. Dreyfus, and P.-A. Ayral, "Rainfall-runoff modeling of flash floods in the absence of rainfall forecasts: the case of 'Cévenol flash floods,'" *Appl. Intell.*, vol. 35, no. 2, pp. 178–189, 2011.

[30] J. F. Adamowski, "Development of a short-term river flood forecasting method for snowmelt driven floods based on wavelet and cross-wavelet analysis," *J. Hydrol.*, vol. 353, no. 3–4, pp. 247–266, 2008.

[31] I. Aichouri, A. Hani, N. Bougherira, L. Djabri, H. Chaffai, and S. Lallahem, "River flow model using artificial neural networks," *Energy Procedia*, vol. 74, pp. 1007–1014, 2015.

[32] K. Khosravi, H. Shahabi, B. Thai, J. Adamowski, and A. Shirzadi, "A comparative assessment of flood susceptibility modeling using Multi- Criteria Decision-Making Analysis and Machine Learning Methods," *J. Hydrol.*, vol. 573, no. March, pp. 311–323, 2019, doi: 10.1016/j.jhydrol.2019.03.073.

[33] D. P. Solomatine and Y. Xue, "M5 model trees and neural networks: application to flood forecasting in the upper reach of the Huai River in China," *J. Hydrol. Eng.*, vol. 9, no. 6, pp. 491–501, 2004.

[34] S. Chen, Z. Xue, and M. Li, "Variable Sets principle and method for flood classification," *Sci. China Technol. Sci.*, vol. 56, no. 9, pp. 2343–2348, 2013.

[35] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[36] E. Dodangeh *et al.*, "Integrated machine learning methods with resampling algorithms for flood susceptibility prediction," *Sci. Total Environ.*, vol. 705, 2020, doi: 10.1016/j.scitotenv.2019.135983.

[37] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[38] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[39] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in *Icml*, 2001, vol. 1, pp. 609–616.

[40] H. Zhang, "The optimality of naive Bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[41] H. R. Maier, A. Jain, G. C. Dandy, and K. P. Sudheer, "Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions," *Environ. Model. Softw.*, vol. 25, no. 8, pp. 891–909, 2010.

[42] H. R. Maier and G. C. Dandy, "Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications," *Environ. Model. Softw.*, vol. 15, no. 1, pp. 101–124, 2000.

[43] K. P. Sudheer, A. K. Gosain, and K. S. Ramasastri, "A data-driven algorithm for constructing artificial neural network rainfall-runoff models," *Hydrol. Process.*, vol. 16, no. 6, pp. 1325–1330, 2002, doi: 10.1002/hyp.554.

[44] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in {P}ython," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[45] P. Contreras, J. Orellana-Alvear, P. Muñoz, J. Bendix, and R. Célleri, "Influence of Random Forest Hyperparameterization on Short-Term Runoff Forecasting in an Andean Mountain Catchment," *Atmosphere (Basel).*, vol. 12, no. 2, p. 238, 2021.

[46] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," *Univ. California, Berkeley*, vol. 110, pp. 1–12, 2004.

[47] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, p. 333, 2011.

[48] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," *Commun. Comput. Inf. Sci.*, vol. 51, pp. 461–471, 2009, doi: 10.1007/978-3-642-04962-0_53.

[49] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 04, pp. 687–719, 2009.

[50] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, p. 1, 2015.
[51] J. Akosa, "Predictive accuracy: A misleading performance measure for highly imbalanced data," in *Proceedings of the SAS Global Forum*, 2017, vol. 12.
[52] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv Prepr. arXiv1811.12808*, 2018.
[53] I. K. de Almeida, A. K. Almeida, J. A. A. Anache, J. L. Steffen, and T. Alves Sobrinho, "Estimation on time of concentration of overland flow in watersheds: A review," *Geociencias*, vol. 33, no. 4, pp. 661–671, 2014.
[54] C. Loumagne *et al.*, "Integration of remote sensing data into hydrological models for reservoir management," *Hydrol. Sci. J.*, vol. 46, no. 1, pp. 89–102, 2001.
[55] Y. Li, S. Grimaldi, J. P. Walker, and V. Pauwels, "Application of remote sensing data to constrain operational rainfall-driven flood forecasting: a review," *Remote Sens.*, vol. 8, no. 6, p. 456, 2016.