Statistical Analysis of a Spatially Explicit Hydrological Forecasting Model using Remote Sensing of the Atmosphere

Jaqueline A. J. P. Soares[1,2], **Michael M. Diniz**[3], **Leonardo B. L. Santos**[1,2]

[1]Center for Monitoring and Early Warning of Natural Disasters (Cemaden)
São José dos Campos – SP – Brazil

[2]National Institute for Space Research (INPE)
São José dos Campos – SP – Brazil

[3]Federal Institute of São Paulo (IFSP)
São José dos Campos – SP – Brazil

{jaqueline.soares,leonardo.santos}@cemaden.gov.br, michael.diniz@ifsp.gov.br

Abstract. *The performance analysis of a data based machine learning model depends directly on the separation of the data for training and testing, thus, a single performance evaluation can generate a biased results. Taking this into account, the aim of this paper is to propose and exemplify a statistical analysis method to get a proper understanding of a spatially explicit hydrological forecasting model performance. For this, the weather radar data used to training and testing a neural network based model was splitted randomically 100 times, and the RMSE and NASH metric was calculated for each one. The analysis of the metrics frequency distribution indicates that the neural network model had better performance than the persistence one.*

1. Introduction

According to [Mal et al. 2018], climate change has become an important factor for many environmental disasters in vulnerable communities and ecosystems. Disasters cause environmental impacts and abrupt changes in the population's daily life, causing financial and human losses.

Extreme hydrological events (EHEs), such as droughts and floods, vary spatially and temporally in nature, and their increase in the last few decades has motivated the research of the spatiotemporal variability of the future extreme precipitation and temperature [Diaz et al. 2019]. The distributed (or spatially explicit) hydrological models have increasingly contributed to these researches, as they consider the spatial variability of the study area, i.e., they use geoprocessing information to make the modeling more realistic.

During the creation, training and testing phases of a distributed hydrological model, there may be several stochastic processes involved. When dealing with stochastic processes, to provide a more adequate and reliable analysis, it is necessary to consider not just an isolated sample but, ideally, an ensemble of different samples. Otherwise, there is a risk that the analysis performed is biased and does not reflect reality.

This work presents a statistical analysis of a distributed hydrological forecasting model using remote sensing data from the atmosphere. This model was first introduced by [Freitas et al. 2022]. In this article, the performance of this model was compared with

Persistence, which will be more detailed in the Section 2.2. Furthermore, the stochastic processes present in the model were mapped and statistically analyzed. The results show that Persistence presented superior performance to the studied model for fast predictions of only 15 minutes. However, for very short-term forecasts, but with a slightly longer duration of 120 minutes, the model is already starting to show better results than Persistence (83% of the time).

2. Materials and Methods

This section presents the dataset used in this article, the sampling methodology for dividing the data between training and testing, the spatially explicit hydrological forecasting model, and the benchmark model used in the performance comparison.

2.1. Dataset

The study area covers the small watershed of the Bengalas river (Figure 1), which has approximately 190 km^2 of drainage area and is located in the mountainous region of Rio de Janeiro, Brazil. This region has a long history of disaster occurrences, mainly due to its natural characteristics and the large occupation of disaster risk areas.

The dataset used in this article is provided by the weather radar of Pico do Couto, in Petropolis, and operated by the Department of Airspace Control (DECEA). A weather radar emits high-energy electromagnetic waves to reach great distances. The analyzed data are from December 2011 to March 2013, with temporal resolution of 15 and 120 minutes. The reflectivity radar data (dBZ) was used to retrieve the accumulated volume of rainfall (mm) for 12 hours using the Marshal-Palmer relation [Marshall and Palmer 1948].

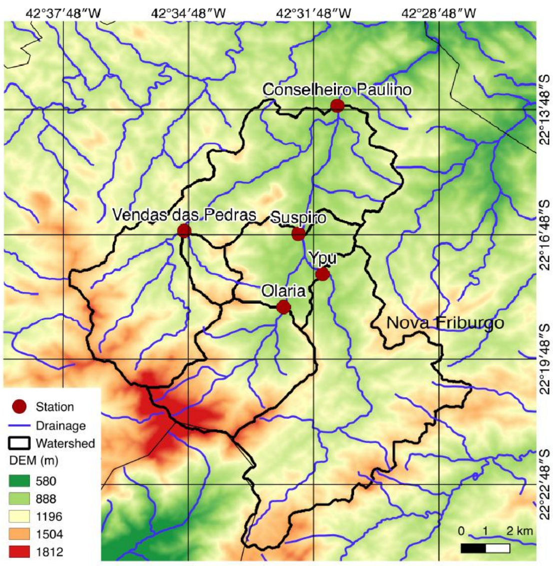


Figure 1. The area of study - Bengalas river watershed. Source: Adapted from [Freitas 2022].

In addition, the neural network training phase requires the water level data located at the lowest point of the watershed. This data was provided by the Conselheiro Paulino hydrological monitoring station of the State Environmental Institute (INEA), for the same period, with resolution of 15 minutes.

2.2. Neural Network and Persistence

The model used for the statistical tests is a neural network with Multilayer Perceptron (MLP) architecture, designed with three hidden layers and a single-neuron output layer. The neural network input is accumulated volume of rainfall for 12 hours derived from weather radar data, and the output is the water level data at the lowest point of the watershed. After training, the neural network used in this article is considered a deterministic model, i.e., it always produces the same output for the same set of inputs.

A basic performance analysis compares the neural network with a model that outputs as forecasting the replication of the last observed data (in this article, we name it "Persistence"). Consider $o(t)$ the observed level data for the time instant t and $y(t+1)$ the Persistence level prediction for the time instant $t+1$. Thus, we have that $y(t+1) = o(t)$ $\forall\ t \geq 0$. The Section 3 presents the results of this comparison.

2.3. Training and Test Samples

Several stochastic processes may be involved during a neural network's creation, training and testing phases, such as initializing the neural network weights and dividing the dataset into training and test data. In this article, seeds were defined for all these processes to make it possible to reproduce the experiments. However, after creating the neural network, according to the mapping performed, the only stochastic process consists of dividing the data between training and testing.

In order to provide a more appropriate comparison of the performance of models based on the data used during the training and testing phase, it is recommended to analyze distributions of data splits between training and testing, and not just a single split. That is because if the data division is biased, even if random, it can result in performance metrics that do not reflect reality. For example, suppose a scenario where a given seed results in an excellent performance metric, but for most seeds, the result is poor. This scenario shows the importance of a broader assessment from a statistical point of view. Thus, this article presents a statistical analysis of this stochastic process to remove any obscurity in the use of the data.

Several data divisions (samples) were performed between the training and test set varying the seeds. In order to allow the reproduction of results, an initial seed (seed = 10) was used to generate another 100 random seeds, and a new data division between training and test was composed for each. It is important to note that the greater the number of seeds analyzed, the greater the reliability of the results, but the higher the total execution time required to produce the results.

Figure 2 shows examples of data divisions between training and testing, always keeping the proportion used of 80% of the instances for training, 10% for validation, and 10% for the neural network test.

The source code used in this article was developed in Python language. Regarding the division of data between training and testing, was used a method that splits arrays or matrices into random train and test subsets (*train_test_split* method from the *sklearn.model_selection* library [Scikit-learn.ORG 2022]). However, whenever randomization is part of a Scikit-learn algorithm, a *random_state* parameter may be provided to control the random number generator used. This work used this parameter to control the shuffling applied to the data before applying the split.

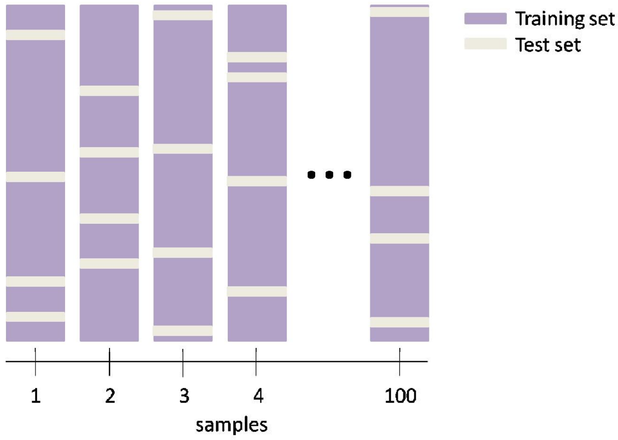
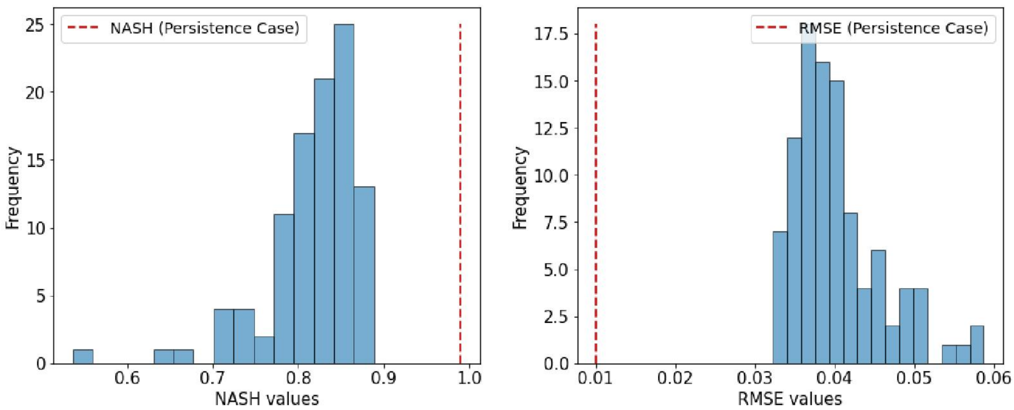


Figure 2. Examples of splitting data between the training and test set.

3. Results and Discussion

This section presents both the results obtained with the neural network against the ensemble of samples used in the stochastic process of training and testing, as well as the performance comparison between the neural network and Persistence.

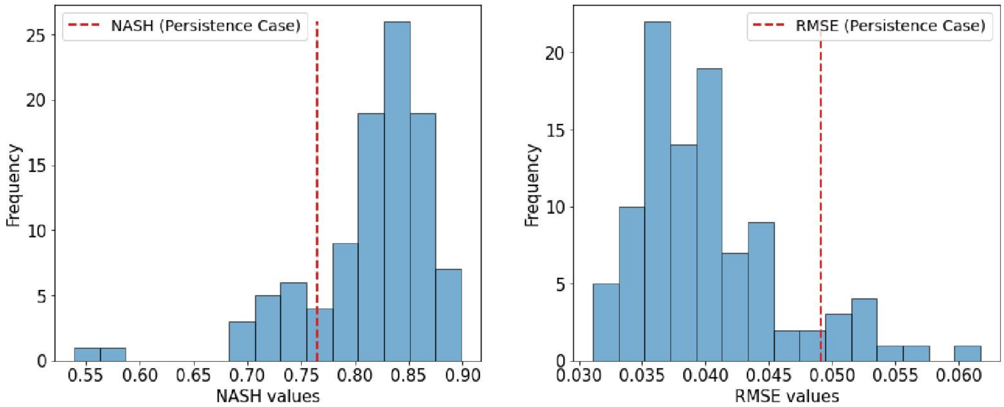
Figure 3 shows the histograms of performance metrics $NASH$ (a) and $RMSE$ (b) of the neural network for 15-minute forecast, while Figure 4 shows histograms of the same metrics for 120-minute forecast. The dashed red line in the histograms represents Persistence's performance for the same metrics.



(a) NASH **(b) RMSE**

Figure 3. Distribution of the neural network metrics for 15-minute forecast.

Note that $NASH$ and $RMSE$ distributions are similar for the 15-minute (Figure 3) and 120-minute (Figure 4) forecasts, with better results for 120 minutes. Note that in Figures 3a and 4a, there is a greater concentration of samples in the highest values, with a respective median shift for this region of the graph and that this concentration quickly drops to lower values. On the other hand, in Figures 3b and 4b, there is a higher concentration of samples at lower values, and this concentration drops a little more gradually at higher values.



(a) NASH **(b) RMSE**

Figure 4. Distribution of the neural network metrics for 120-minute forecast.

Furthermore, it is possible to verify that Persistence surpassed the neural network for all the samples used for too short time intervals (Figure 3 - time in minutes). It is worth noting that the modeled problem is a spatial-temporal process with a not-so-fast time scale compared with the data's time resolution. However, for intervals of a few hours (Figure 4), the neural network presents more satisfactory results than Persistence - 83% of the seeds used in the data division between training and testing resulted in models with better values of $NASH$ when compared to Persistence. In this case, there are significant changes in the spatial-temporal dynamics, and Persistence is not so far good as previously.

Table 1 shows a summary of the neural network performance metrics statistics for the 15 and 120-minute forecasts against the 100 samples used in the stochastic process of dividing the data between training and test set.

Table 1. Statistical indicators of neural network performance metrics for 15-minute and 120-minute forecasts.

Statistical	**15 minutes**		**120 minutes**	
Indicators	**RMSE**	**NASH**	**RMSE**	**NASH**
mean	0.0403	0.8170	0.0403	0.8122
std	0.0055	0.0554	0.0057	0.0613
min	0.0322	0.5363	0.0310	0.5389
25%	0.0362	0.7978	0.0363	0.7907
50%	0.0391	0.8326	0.0392	0.8273
75%	0.0426	0.8558	0.0430	0.8516
max	0.0587	0.8886	0.0617	0.8988

According to Table 1, statistics for metrics $RMSE$ and $NASH$ were quite similar between the 15 and 120-minute forecasts. Furthermore, for 120 minutes, the minimum $NASH$ was 0.5389, and the maximum was 0.8988, resulting in a range of 0.3599. This variation is very significant for this metric. The same is not valid for $RMSE$, whose minimum and maximum values were, respectively, 0.0310 and 0.0617, resulting in a range of 0.0307, which is not very representative for this metric.

4. Conclusions and Perspectives

In this work, we propose and exemplify a statistical analysis method to properly understand a spatially explicit hydrological forecasting neural network using weather radar data. In this method, the stochastic process of dataset separation between training and test was analyzed not as an isolated sample but as an ensemble of different samples.

The neural network's performance has been compared with Persistence, whose prediction consists of replicating the last observed data. For a 15-minute (very short-term) forecast scenario, Persistence outperformed the neural network in 100% of the samples, and for 120 minutes (short-term), the neural network outperformed Persistence in 83% of the samples. This last case shows that the choice of seed (sample) can be decisive in the performance comparison, which justifies and motivates the analysis of different samples.

Limitations of this work are the extension of the time series used in separating data between training and test and the number of analyzed seeds. Currently, the time series is limited to a few years of data. However, as future work, we intend to increase it to allow a better analysis of the stochastic process analyzed in this article. Furthermore, we intend to increase the number of seeds by an order of magnitude to increase the results' reliability further.

5. Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

Diaz, V., Corzo, G., and Pérez, J. R. (2019). 3 - Large-scale exploratory analysis of the spatiotemporal distribution of climate projections: Applying the striving toolbox. In Corzo, G. and Varouchakis, E. A., editors, *Spatiotemporal Analysis of Extreme Hydrological Events*, pages 59–76. Elsevier.

Freitas, C. P. (2022). Neural network for very short-term hydrological forecasting. Master's Dissertation in Applied Computing, National Institute for Space Research (INPE), São José dos Campos.

Freitas, C. P., Bacelar, L., Soares, J. A. J. P., Diniz, M. M., Lima, G. R. T., Stephany, S., and Santos, L. B. L. (2022). A neural network-based short-term hydrological model using weather radar data. *to be submitted*.

Mal, S., Singh, R. B., Huggel, C., and Grover, A. (2018). *Introducing Linkages Between Climate Change, Extreme Events, and Disaster Risk Reduction*, pages 1–14. Springer International Publishing, Cham.

Marshall, J. S. and Palmer, W. M. K. (1948). The distribution of raindrops with size. *Journal of the Atmospheric Sciences*, 5(4):165 – 166.

Scikit-learn.ORG (2022). API Reference. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (Accessed: 09 May 2022).