A convolutional LSTM neural network for precipitation nowcasting based on weather radar data

Aurelienne A. S. Jorge[1], Marcos Quiles[2], Izabelly C. Costa[1], Leonardo B. L. Santos [3]

[1]Instituto Nacional de Pesquisas Espaciais (INPE)
Cachoeira Paulista, SP – Brazil

[2]Universidade Federal de São Paulo (Unifesp)
São José dos Campos, SP – Brazil

[3]Centro Nacional de Monitoramento e Alertas de Desastres Naturais (Cemaden)
São José dos Campos, SP – Brazil

aurelienne.jorge@inpe.br, quiles@gmail.com, izabelly.costa@inpe.br, santoslbl@gmail.com

Abstract. *Due to the high instability of the atmosphere, it is a challenging task to forecast precipitation events in the very short term. Artificial Neural Networks have been recently employed in this context as an alternative to solve prediction problems. This work proposes a convolutional LSTM neural network based on weather radar images (spatial data) to forecast rainfall events, whatever the precipitation intensity is. The preliminary results show better performance when comparing the network prediction error with a persistence model.*

1. Introduction

First mentioned by [Browning 1980], nowcasting is defined as the description of the weather's current state and the prediction of changes that can be expected in a few hours - from the present to 6 hours ahead. Due to the high instability of the atmosphere, it is impossible to precisely predict the location and time of convective phenomena several hours in advance. In fact, it is already a challenging task to forecast within a few hours in advance.

Artificial Neural Networks (ANNs) have been designed by researchers in many scientific areas to solve a variety of problems such as pattern recognition, optimization, and prediction, including weather forecasts [Jain et al. 1996]. Kumar et al. [Kumar et al. 2020] propose a precipitation nowcasting architecture named 'ConvCast' that uses a Convolution Long-Short Term Memory (ConvLSTM) neural network to predict short-term precipitation events based on satellite data. Caseri et al. [Caseri et al. 2022] also present a ConvLSTM solution to precipitation nowcasting using weather radar data, albeit focusing on heavy rainfall events.

This work proposes a ConvLSTM neural network based on remote sensing data for precipitation nowcasting. In this initial phase, we use weather radar data, but satellite images may also be included in future research steps. Convolutional neural networks are suitable for data grids like weather radar images as they keep the spatial relations between features [Jorge et al. 2020]. Moreover, LSTM is a recurrent neural network appropriate to perform predictions based on time series data. Therefore, ConvLSTM is an ideal tool to apply in this context of nowcasting using weather radar images.

The focus is not only on heavy rainfall events [Caseri et al. 2022], but also on events of lighter intensity. Initial results show predictions with low error to the observed data. The paper is organized as follows: Section 2 describes the dataset used, the pre-processing flow, and the implemented model architecture. Section 3 presents the preliminary results of this study, and Section 4 contains the final considerations and discusses the future perspectives of this ongoing research.

2. Materials and Methods

The dataset is composed of weather radar images from march 2019. The data was collected with the weather radar located in the city of São Roque (São Paulo - Brazil). This equipment has a spatial coverage of 250 kilometers and provides high spatial and temporal resolution data, with 1 kilometer and 10 minutes. It performs azimuth scans in 360 degrees over 15 different elevation levels composing a volumetric scan. From that, we can extract a product named the Constant Plan Position Indicator (CAPPI), which consists of a horizontal cross-section of data at a constant height. For this work, we use the CAPPI product at the height of 3 km to avoid altitude changing and ground echo problems. The product is a gray-scale image whose pixels represent reflectivity values (in dBz units), from which we can later estimate the precipitation rate. Only values above 20 dBZ were considered in this study, as it is the threshold to represent a light rain.

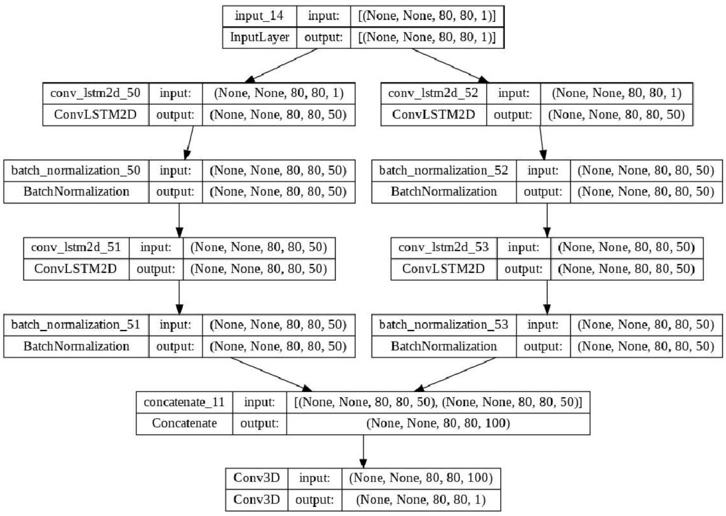

Figure 1. The ConvLSTM topology

In the pre-processing phase, we perform a spatial clipping to select only the area of interest: the Metropolitan Area of São Paulo (MASP). As a result, we have 80x80 grids with 1 kilometer of resolution. In this step, we also split the time series into samples of 10 frames each, keeping the temporal sequence with a time step of 10 minutes. There are 309 samples, which are split into train and test sets sequentially (80% and 20%, respectively). The values are normalized to fit in a range from 0 to 1.

An application written in Python with the Keras library was developed to pre-process the dataset and implement the neural network. The ConvLSTM architecture is presented in Figure 1. The input layer is prepared to receive data with the following shape: 80x80x1 (width x height x channels). There is no fixed number of time steps to make the model flexible. The topology is mainly composed of ConvLSTM layers, each one followed by a Batch Normalization. The input is received by two branches: the first one is composed of ConvLSTM layers with 50 neurons each and a kernel size of 1x1, and the second one with the same number of layers and neurons but with a kernel size of $3x3$. At the end, the two branches are concatenated to feed a 3D convolution layer that delivers the network's output in the same shape as the input data. In the output layer (Conv3D), its size is increased to 1x3x3. The Rectified Linear Activation Function (ReLU) is the activation function of the ConvLSTM layers, and Sigmoid is the activation function of the Conv3D layer. Adam is the optimizer employed to adjust the network weights, and Mean Squared Error (MSE) is the loss function set in the model compilation.

3. Results

The training phase was executed for ten epochs, and 10% of the training set was set apart for the validation process. At the end of the training phase, the result is a validation loss of $1.2e-2$. Moreover, the loss obtained in the model evaluation, using the test subset, is even lower: $6.8e-3$. Besides, tests were made with various scenarios to analyze the model performance further.

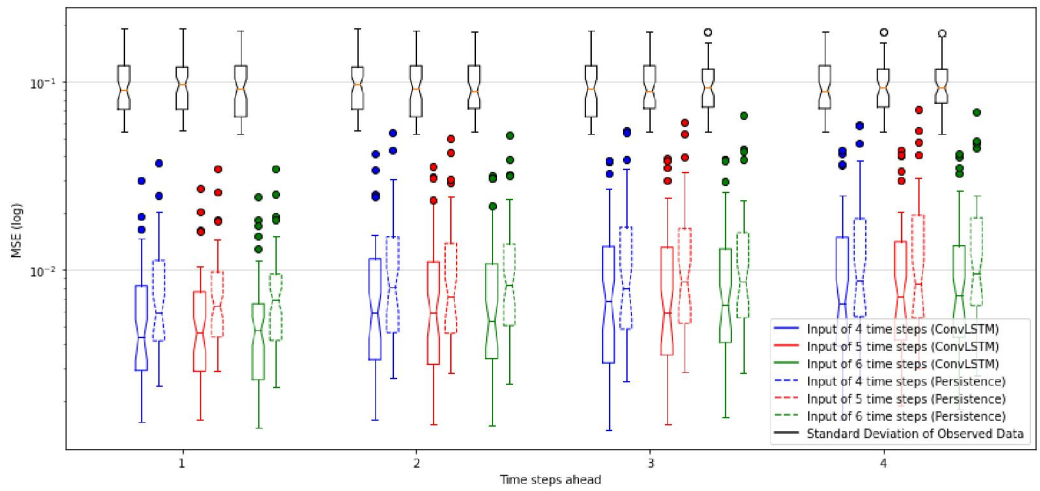

Figure 2. Boxplots of MSE based on 4 (blue), 5 (red) and 6 (green) time steps of observed data as input, and the predictions are for 1, 2, 3 and 4 time steps ahead.

Inputs with 4, 5 and 6 time steps (40, 50 and 60 minutes, respectively) were supplied to the model in order to predict the next 4 time steps (10, 20, 30 and 40 minutes). Figure 2 presents the boxplots with the MSE distribution for each scenario. The results are compared with a persistence model, which consists of keeping the last observed data. Although the range of the distributions is very similar, we notice that the error tends to slightly increase as we try to predict more time steps ahead. There is no significant difference when using more samples of observed data as input. An input of 4 time steps is enough to have a better prediction compared to the persistence model.

Figure 3 illustrates an example of the result generated from the model prediction. It is based on a sample from the test set using 4 time steps (40 minutes of observed data with 10 minutes step) as input. The predictions are made for the next 4 time steps (10 to 40 minutes ahead).

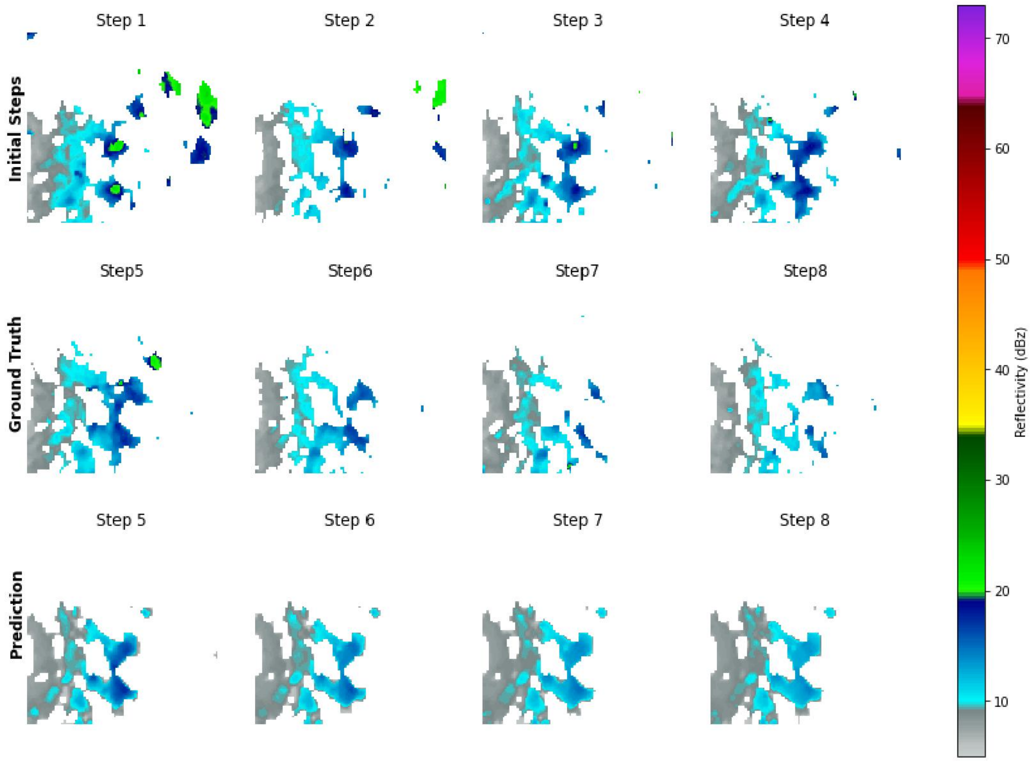

Figure 3. Example of prediction for 4 time steps ahead (steps 5 to 8 of *Prediction* row) based on 4 initial steps of observed data (steps 1 to 4). The real observed data for steps 5 to 8 are presented on the second row (*Ground Truth*).

In this sample, which shows part of a light rain event, it is possible to observe that the predictions are visually very similar to the ground truth.

4. Final Considerations

This paper presented a convolutional LSTM neural network to forecast rainfall events evolution in a very short time based on weather radar data. Tests were made with different numbers of input time steps to predict precipitation events within 40 minutes ahead. The results show better results when comparing the network prediction error with a persistence model. It could be also observed that there is no significant difference when using more samples of observed data as input. An input of only 4 time steps is enough to have a better prediction compared to the persistence model.

As a continuation of this study, there are perspectives of including more data in the model, such as other physical parameters of the atmosphere, to enable better tracking of event directions and the formation of new events.

5. Acknowledgements

This research was partially supported by grant 420338/2018-7 of the Brazilian National Council for Scientific and Technological Development (CNPq) and by grants 2018/06205-

7 and 420338/2018-7 of São Paulo Research Foundation (FAPESP) and DFG-IRTG 1740/2.

References

[Browning 1980] Browning, K. A. (1980). *Local weather forecasting*, page 179–211.

[Caseri et al. 2022] Caseri, A. N., Lima Santos, L. B., and Stephany, S. (2022). A convolutional recurrent neural network for strong convective rainfall nowcasting using weather radar data in southeastern brazil. *Artificial Intelligence in Geosciences*, 3:8–13.

[Jain et al. 1996] Jain, A., Mao, J., and Mohiuddin, K. (1996). Artificial neural networks: a tutorial. *Computer*, 29(3):31–44.

[Jorge et al. 2020] Jorge, A. A. S., Costa, I. C., and Santos, L. B. L. (2020). Geographical Complex Networks applied to describe meteorological data. In *Proceedings...*, pages 258–263, São José dos Campos, Brazil. BRAZILIAN SYMPOSIUM ON GEOINFORMATICS, INPE.

[Kumar et al. 2020] Kumar, A., Islam, T., Sekimoto, Y., Mattmann, C., and Wilson, B. (2020). Convcast: An embedded convolutional lstm based architecture for precipitation nowcasting using satellite data. *PLOS ONE*, 15(3):e0230114.