Neural Networks for Time Series Rainfall Forecasting: A Case Study in Manaus, Amazonas

Elloá B. Guedes, Patrick Magalhães de Lima, Maria Betânia Leal de Oliveira

[1]Superior School of Technology
Amazonas State University
Av. Darcy Vargas, 1200 – Manaus – Amazonas

{ebgcosta, pml.eng, mloliveira}@uea.edu.br

Abstract. *In this paper we aim at forecasting rainfall occurrence using time series and neural networks. In our approach, three meteorological variables (average high and low temperatures and relative humidity) are taken as input by neural networks with a single hidden layer in order to deliver one-step ahead rainfall occurrence predictions. We performed a case study to evaluate such approach considering* 40 *years of data from an automated weather station located in Manaus, Amazonas. From* 38 *neural networks suited to this scenario, we could identify, using Akaike's Information Criterion, that the one with window size equal to* 3 *and* 7 *neurons in the hidden layer could forecast rainfall with an accuracy of* 99.71% *for a testing set. The results obtained indicate the feasibility and efficency of time series neural networks for rainfall forecasting, suggesting a methodology that can be adopted by many other locations.*

Introduction

Forecasting rainfall is a very important task. It is a crucial activity for preventing flooding and managing water resources. The variations in timing and quantity of rainfall have potential impact on agricultural activities [Darji et al. 2015]. Given such importance, mankind has attempted to predict the weather since ancient times. However, forecasting meteorological events can be a very complex activity because of atmospheric instability, nonlinear interactions between diferent spatial scales, and even human interference [Bushara and Abraham 2013].

Although the complexity involved, forecasting rainfall has been extensive explored by different approaches. Statistical methods, for intance, consider the analysis of the rainfall distribution according to a time series and its characteristics, such as seasonality, linearity, trend, and stationarity [Palit and Popovic 2005]. Computational Intelligence has also been used to address such problem in many places around the world. Neural networks, in particular, have caught attention for being good forecasters because they are data driven, capable to generalize, computationally efficient and can detect nonlinear relationship among input and output variables [Darji et al. 2015].

Considering the role of Computational Intelligence to rainfall forecasting, in this paper we use artificial neural networks and time series to deal with the problem of rainfall occurence in a case study scenario. We considered data from Manaus, Amazonas, Brazil that has a tropical monsoon climate in which rainfall is very abundant, but not uniform. We used past lagged observations from 40 years of three atmospheric variables to train 38 different neural networks. According to our results, it was possible to see that the most adequate neural network for such scenario, according to Akaike's Information Criterion, has 7 neurons in the hidden layer and considers data from 3 previous days, delivering results for a testing set with accuracy of 99.71%. The results obtained indicate the feasibility and efficency of time series neural networks for rainfall forecasting, suggesting a methodology that can be adopted by many other locations.

To present such results, this paper is organized as follows. An overview of time series in the context of neural networks is introduced in Sect. 2. Some works in the literature that consider time series and neural network for rainfall forecasting are discussed in Sect. 3. A detailed characterization of our problem, including domain and input data, as well as the methodology adopted in order to obtain the results are shown in Sect. 4. Results obtained and the discussion are presented in Sect. 5. Lastly, final remarks and suggestion for future work can be found in Sect. 6.

Neural Networks for Time Series Forecasting

Neural networks have been widely used as time series forecasters. In 1964, Hu [Hu 1964] was the first to demonstrate – on a practical weather forecasting example – the general forecasting capability of neural networks. Since so, many successful applications on market predictions, meteorological and network traffic forecasting have been described in literature [Frank et al. 2001].

A time series is a time-ordered sequence of observation values of a physical or financial variable made at equally spaced time intervals Δt, represented as a set of discrete values $x_1, x_2, x_3, \ldots, x_{t-1}$. We are interested in forecasting a future value x_t based on these previous inputs. In a formal approach, we want to find a function $f : \mathbb{R}^n \to \mathbb{R}$ such as to obtain x_t from the n time steps back from time t:

$$x_t = f(x_{t-1}, x_{t-2}, \ldots, x_{t-w}) \tag{1}$$

The classical approach for time series predictions considers statistical methods, such as the Box-Jenkins that is a general forecasting methodology for time series generated by a stationary autoregressive moving-average process [Palit and Popovic 2005]. However, statistical approaches lack an ability to identify nonlinear patterns and irregularity in time series [Darji et al. 2015]. Several works in the literature verified that neural networks have ability to implicitly detect complex nonlinear relationships between dependent and independent variables. As a result, when compared to classical methods, it could be verified that: (i) for time series with long memory, both approaches deliver similar results; (ii) neural networks outperform the classical Box-Jenkins approach in some experiments by more than 100%; and (iii) the optimally tuned neural network topologies are of higher efficiency than the corresponding traditional algorithms [Palit and Popovic 2005].

A neural network for time series forecasting aims at inducing a function f with the same idea as shown in Eq. (1), but with some particularities. In order to forecast a certain x_t, k parameters $p_t^1, p_t^2, \ldots, p_t^k$ collected at time t are given together with some previous inputs $u_{t-1}, u_{t-2}, \ldots, u_{t-w}$ and their respective outputs $x_{t-1}, x_{t-2}, \ldots, x_{t-w}$ of the phenomen being predicted. If the focus is on *one-step-ahead forecasting*, then only one output neuron is needed, which will return x_t ; otherwise, for *multi-step-ahead forecasting*, $\ell + 1$ neurons may be employed in the output layer, returning $x_t, x_{t+1}, \ldots, x_{t+\ell}$. This strategy to use neural network with time series is shown in Figure 1. The number w of past lagged input and output observations taken by the neural network will be called *window size* from now on.

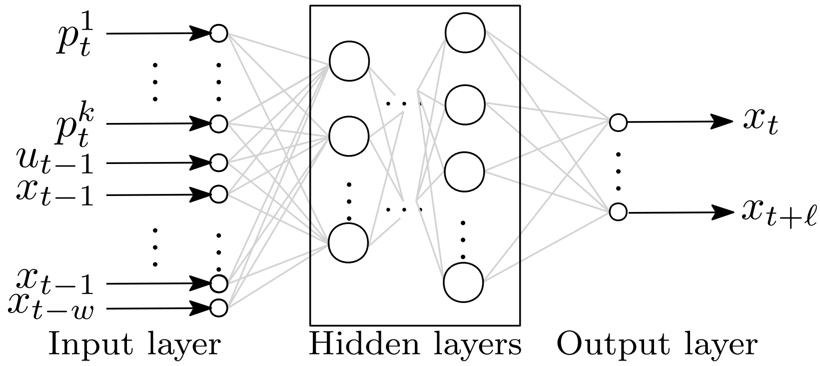


Figure 1: Architecture of a multilayer feedforward neural network to forecast data from a time series. The number of hidden layers and their neurons can vary according to the scenario considered.

Determining the neural network architecture in a time series forecasting scenario is a challenging task. One must decide: (1) what kind of forecasting will be considered (one-step or multi-step ahead) because it will define the number of neurons in the output layer; (2) how many and what past lagged observation will be taken as input (*window size estimation*); (3) the number of hidden layers and of neurons in each layer – these parameters are useful for capturing the nonlinear relationship between input and output variables; and (4) the type of neural network (Multi-Layer Perceptron, Radial Basis Function, etc.).

Some answers to these questions can be found in the literature. In the scenario of time series forecasting, feedforward neural networks are mostly used. Regarding the number of hidden layers, most forecasting applications use only one hidden layer and a small number of hidden nodes. Perhaps the most difficult part is determining the input data and the window size. Empirical results have been suggesting that the input layer is more important than the hidden layer in time series forecasting problems. After determining the neural network and its architecture, it must be trained so that the parameters of the network can be estimated from the data. The Levenberg-Marquardt optimization for back-propagation is usually a frequent choice due to its good performance, even requiring more computer memory usage [Zhang 2012].

Taking into account the practices described for a time series forecasting scenario, once the neural network has been trained, it can be used for prediction. However, an important question that needs to be discussed is the *performance evaluation* which aims at quantifying the goodness of the neural network considered. Literature suggests to divide the dataset into training examples and testing examples. The training examples will be used to figure out the window size estimation, to adjust the weights, and even to validate the neural network, among others. The testing examples are used to estimate the generalization capability of the neural network. The output of the neural network for the testing examples will be compared with the original outcome presented in the dataset. Some metrics such as *Mean Squared Error* (MSE), *Mean Absolute Percentage Error* (MAPE) are derived in order to quantify the performance of such neural network, enabling comparisons with other works, models and approachs. However, although widely used, a disadvantage of these metrics is that they rely on a specific testing scenario. In order to overcome such limitation, *Akaike's Information Criterion* (AIC) is frequently considered.

Founded on the concept of entropy from Information Theory, AIC offers a relative measure of the information lost when a given model is used to describe reality and can be said to describe the tradeoff between bias and variance in model construction. It is a basis of comparison and selection among several statistical models where the one having the lowest AIC being considered the best [Panchal et al. 2010]. In a time series forecasting scenario with neural networks, AIC will be used as a measure of the relative quality of each neural network for the dataset being used. Given a collection of different neural networks, AIC will be used to estimate the quality of each model, relative to each of the other models.

Considering the concepts and methodologies introduced for time series forecasting with neural networks, some applications described in the literature considering the problem of rainfall forecasting will be detailed in the next section.

Related Work

Neural networks have been adopted to rainfall forecasting in many works in the literature, delivering relevant results and being considered a more suitable approach than classical methods commonly used on prediction tasks [Nayak et al. 2013]. In this section we will show some works with the neural network approach to rainfall forecasting.

The work of Abbot and Marohasy considered the monthly and seasonal rainfall forecasting in Queensland, Australia [Abbot and Marohasy 2012]. The authors developed Time-Delay Recurrent Neural Networks (TDRNNs) to forecast rainfall 3-month lead. Their models considered both time-delay inputs and output recurrence to incorporate dynamics and "memory" capability. The input data were divided into four classes: (i) monthly rainfall, corresponding to 20 sites in Queensland State, lagged up to 12 months; (ii) climatic indexes, whose more relevant were Southern Oscillation Index (SOI), Pacific Decadal Oscillation (PDO), Niño 3.4 and Dipole Mode Index; (iii) atmospheric temperature, considering the minimum and maximum temperature values; (iv) solar data, in which sunspot and total solar irradiation were chosen.

The authors developed a TDRNN with two hidden layers using a trial and error approach to evaluate different networks configurations. The resulting model was used to forecast rainfall on each of 20 sites with Normalized Mean Squared Error (NMSE) and Pearson correlation coefficient as performance evaluators. Initially only the monthly rainfall values were used as inputs, other data were progressively added. The best data input combination consisted of lagged monthly rainfall, atmospheric temperature, SOI, PDO and Niño 3.4. The results of TDRNN performance were then compared with the Predictive Ocean Atmosphere Model for Australia (POAMA) and with the General Circulation Model (GCM). It could be observed that the TDRNN model achieved lower NMSE values on 17 sites of a total of 18 sites used in this comparison.

Htike and Khalifa worked in rainfall forecasting models to Malaysia using Time-Delay Neural Networks (TDNN) to perform one-step-ahead predictions [Htike and Khalifa 2010]. The data set, consisting in daily rainfall data from January of 1980 to May of 2009, were obtained from Malaysian Meteorological Department and treated to create four data sets composed by monthly, quarterly, biannual and yearly rainfall data, being their values normalized within the range from -1 to 1 afterward.

In this work, TDNN models with one hidden layer were developed to each data set and the approximated optimal number of input delays and hidden neurons for these models were obtained from systematic trial and error. The authors used 80% of the data set to train and validate the models and the remaining 20% to test them, also using the MAPE value obtained in this dataset to evaluate the models' accuracy. The yearly forecast model gave the most accurate result (94.25%), which decreased for the biannual, quarterly and monthly models, respectively.

Neural networks were used by Shukla et al. in order to forecast the Indian Summer Monsoon Rainfall Index (ISMRI) using Sea Surface Temperature (SST) anomalies indices, specifically the El Niño-Southern Oscillation (ENSO) [Shukla et al. 2011]. The authors used the ISMRI index corresponding to the period of 1951 to 2003 as well as the Niño-$1 + 2$, Niño-3, Niño-3.4 and Niño-4 indexes from 1950 to 2003 as input parameters. Principal Component Analysis technique was used to reduce the number of input parameters due to the amount of data considered. This technique generates a vector of uncorrelated values which are called Principal Components (PC), in which the first one – most relevant – was chosen to be used as input of the neural network. All the data were also normalized.

Multilayer feed-forward neural networks with a single hidden layer were used by the authors. In addition, different regression models were constructed, where five of these models used the Niño indexes individually, one used all the input paramenters and the last model used only the PC value. These models were compared according to performance metrics such as root mean squared error, correlation coeficient and standard deviation. Although the multiple regression model being the best among this class, its prediction was inferior to the mean prediction. However, the neural network model gave significant results, better than the mean prediction, which reinforced the idea of non-linear relationship between ISMIR and Niño indexes.

The literature surveyed regarding rainfall forecasting with neural networks shows that time-delay neural networks are mostly adopted, with satisfactory accuracy results. In practical scenarios, forecasting rainfall may vary considering the time series data available in the location of interest. Instead of relying in complex metheorological variables with non-trivial measurement and significance, our case study considers attributes that are easily available in many locations, collected from typical automated weather stations.

Case Study

In this case study, we considered the one-step ahead rainfall forecasting in Manaus, Amazonas. The dataset used was obtained from an automatic weather station hosted and mantained by the National Institute of Meteorology (*Instituto Nacional de Meteorologia* – INMET) located at $3°6'13.2552''$S, $60°0'6596''$W, 61.25m above sea level, between the years of 1970 to 2010 [INMET 2015]. Such weather station is shown in Figure 2. Missing values in data were discarded, and 11719 entries carachterize the dataset considered in this work.

Our goal is to one-step ahead predict the occurrence or not of rainfall in Manaus. This is a very challenging task because the rainfall in Manaus is very abundant and not uniform [Sioli 1991]. Most part of the rainfalls in Manaus are a result of many precipitating systems, such as the Intertropical convergence zone, Subtropical Anticyclones, Bolivian High, Tropical Mesoscale Systems, South Atlantic Convergence Zone, Synoptic Scale Systems, among others [da Silva 2012]. We can see that the rainfall in Manaus is the result of a very complex natural system.

According to the Köppen climate classification system, Manaus has a tropical monsoon climate, with consistently high temperatures throughout the year and intense rainfall

Figure 2: Automated weather station in Manaus, Amazonas.

from October through to June. August is the driest month with less than 60mm of precipitation [Alavares et al. 2014]. The rainfall in Manaus is very important factor that shapes the weather in the city [Sioli 1991]. It happens because the city of Manaus and its surroundings have a naturally uncomfortable weather, with warm and humid Equatorial characteristics [da Silva 2012].

Data Overview

In our case study, we consider the following five attributes in the dataset:

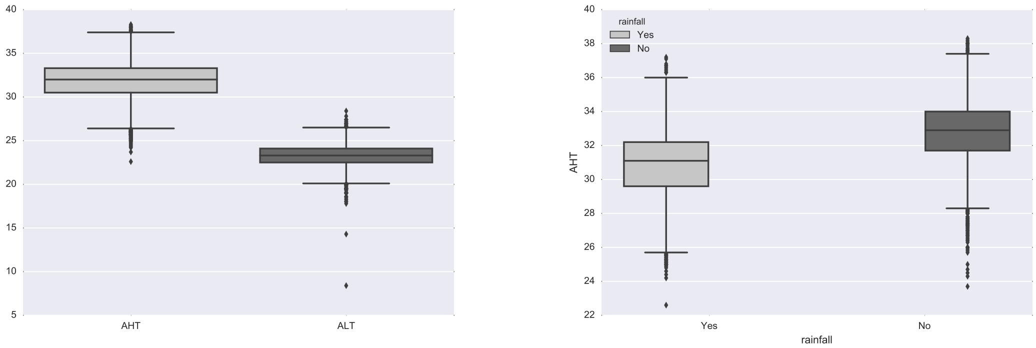
1. Average High Temperature (AHT);
2. Average Low Temperature (ALT);
3. Average Relative Humidity (RH);
4. Average Wind Speed (WS);
5. Rainfall.

The categorical rainfall attribute is the target attribute considered, indicating the occurrence or not of rainfall. The other attributes are numerical and their average and standard deviation are shown in Table 1.

Table 1: Average and standard deviation of input attributes.

	AHT	ALT	RH	WS
Average	31.81	23.31	82.97	1.64
Std. Dev.	2.17	1.20	7.84	1.60

The AHT and ALT attributes, which indicate temperatures in the Celsius scale, have their distribution showed in boxplots of Figure 3a. It is possible to notice that there are some overlapping between the values, indicating that a low temperature can be considered as a high depending on seasonal factors. It is also possible to see, in Figure 3b, that there is not a clear separation for the AHT values considering rainfall occurrence. The same was observed for ALT values.



(a) Boxplots of AHT and ALT attributes.

(b) Boxplot of the AHT values considering the rainfall occurrence.

Figure 3: Boxplots obtained from the dataset considered.

Aiming at verifying which input attributes may have significant relation with rainfall occurrence, we obtained the Pearson correlation coefficient of these attributes versus the rainfall. The results are shown in Table 2. It is possible to notice that there is a significant negative correlation for AHT, ALT and RH attributes. However, the correlation coefficient for WS and rainfall has a negligible value. For that reason, it was decided not to use the attribute WS in the further steps of rainfall prediction.

Table 2: Pearson correlation coefficient of the input attributes versus target attribute.

	AHT	**ALT**	**RH**	**WS**
Rainfall	-0.45655336	-0.33254008	-0.33254008	-0.03047564

From 11719 days of observation, it was possible to see that rainfall occurred in 5830 days (49.7%), and not occurred in 5889 days (50.3%). In order to see if it is possible to detect some statistical pattern in the rainfall occurrence, we performed a χ^2 test, considering as null hypothesis that rainfall is distributed according to a random uniform distribution variable. The p-value obtained was of 0.585, indicating that the null hypothesis cannot be rejected for a significance level of $\alpha = 0.05$. It means that, for a 95% confidence level, it is not possible to distinguish a non-random pattern in rainfall occurrence. In order to illustrate such statistical difficulty in finding a non-random pattern in rainfall, we show a stem plot in Figure 4 of the last 60 days of rainfall in the dataset considered.

Methodology

In our case study we are going to follow the methodology described below to build, test and train the neural networks for the scenario considered. We are considering the problem of one-step ahead rainfall occurrence forecasting in Manaus.

1. **Window size**. Choose the window size n from the set $[1, 3, 5, 7, 10, 16, 21]$;
2. **Organize input data**. The main objective of this step is to prepare the input data.
   (a) Organize the n past lagged observations as input data: consider the n previous values of rainfall, AHT, ALT, RH;
   (b) Add to input data the values of AHT, ALT and RH of the same day that rainfall will be forecasted;

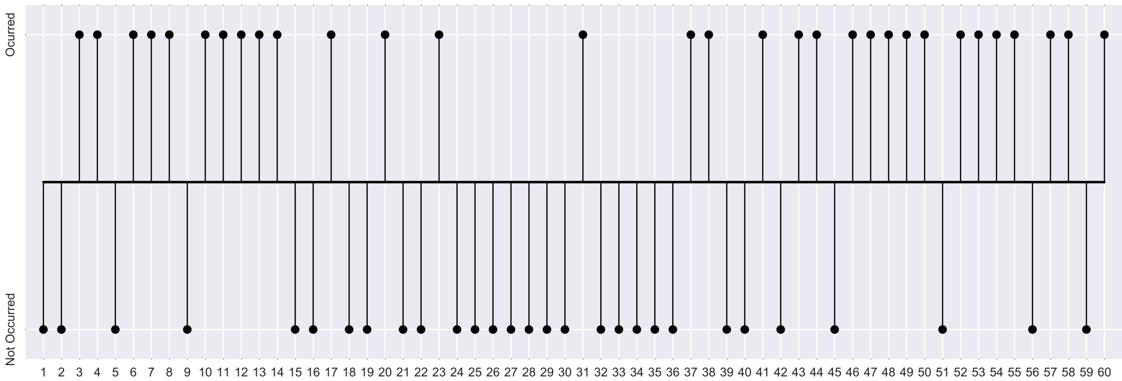


Figure 4: Stem plot of the rainfall occurence in the last 60 days of data illustrates the abscence of a non-random pattern.

   (c) Split the entire resulting data set into three subsets considering a temporal division, aiming at provide some temporal features to improve neural network training. The considered range was:
      i. Training: 1970 - 2006;
      ii. Validation: 2007 - 2009;
      iii. Test: 2010.
3. **Define several neural network architectures**. Different feedforward neural networks will be considered for this forecasting scenario.
   (a) **Input Nodes** (in). Defined according to the number of input parameters, where: $in = 4 * n + 3$, where n is the window size;
   (b) **Hidden Layer**. Only a single hidden layer will be considered;
   (c) **Neurons in the Hidden Layer** (h). Defined according to Baum and Haussler rule of thumb [Baum and Haussler 1989] adapted to this scenario:

$$h \leq \frac{N_{\text{TR}} \times E_{\text{TOL}}}{in + 1} \tag{2}$$

   where N_{TR} is the number of training examples (Step 2-c-i), E_{TOL} is the error tolerance set to 0.01, in is the number of input neurons (Step 3-a);
   (d) **Output Nodes**. Set to 1 because one-step-ahead forecasting is being considered.
4. **Train each neural network**. Using the Levenberg-Marquardt algorithm and considering the appropriate training set according to the window size defined previously;
5. **Validate each neural network**. Discard neural networks whose accuracy may indicate overfitting on the training data;
6. **Test each neural network**. Record its performance and hitting percentage on the test set (Step 2-c-iii);
7. **Performance Evaluation**. Determine the Akaike's Information Criterion (AIC), MSE and MAPE values for each model.

In the last step, after evaluating AIC, we can rank the neural networks according to it and select the model which has the lower AIC, indicating its best performance for the scenario considered. Despite that, we will also obtain MSE and MAPE because they are considered more intuitive metrics regarding the neural network accuracy, being widely used in this research domain.

Results and Discussion

By following the methodology previously described, we built and trained 38 neural networks for the scenario of rainfall forecasting in Manaus. The architecture of such networks and their MSE and MAPE results are presented in Table 3, where w is the window size, in in the number of neurons in the input layer, and h is the number of neurons in the hidden layer.

Table 3: Neural networks results of MSE and MAPE for the testing set.

w	in	h	MSE	MAPE $\times 10^{-3}$
1	7	1	0.1606	2.9911
1	7	2	0.1570	2.9090
1	7	3	0.1590	2.8663
1	7	4	0.1602	3.0250
1	7	5	0.1613	3.0356
1	7	6	0.1581	2.9637
1	7	7	0.1606	2.9600
1	7	8	0.1673	3.0587
1	7	9	0.1623	2.8997
1	7	10	0.1613	2.8997
1	7	11	0.1622	2.8874
1	7	12	0.1614	3.0567
1	7	13	0.1611	3.0332
3	15	1	0.1540	2.9149
3	15	2	0.1564	2.9151
3	15	3	0.1498	2.7936
3	15	4	0.1552	2.7894
3	15	5	0.1521	2.8525
3	15	6	0.1618	2.9722
3	15	7	0.1614	2.8826
5	23	1	0.1513	2.8893
5	23	2	0.1497	2.7620
5	23	3	0.1513	2.8893
5	23	4	0.1535	2.8135
5	23	5	0.1512	2.7526
7	31	1	0.1504	2.8796
7	31	2	0.1521	2.8591
7	31	3	0.1496	2.8069
7	31	4	0.1515	2.8455
10	43	1	0.1483	2.8711
10	43	2	0.1492	2.8558
10	43	3	0.1485	2.8356
13	55	1	0.1455	2.8531
13	55	2	0.1425	**2.7032**
16	67	1	0.1438	2.8448
16	67	2	0.1437	2.9010
21	87	1	**0.1422**	2.8297
21	87	2	**0.1422**	2.8388

The lower values for MSE in Table 3 can be seen in the neural networks which have the larger window sizes. This result indicate that such larger window size may introduce some *memory* – in a Markovian sense – in the forecasting process, reducing errors. A plot of the window size versus MSE shown in Fig. 5 illustrates such relation observed in the results.

Regarding MAPE, which can be considered as a straightforward method to obtain the accuracy of the models (accuracy $= 1 -$ MAPE), we have that all results are satisfactory on a 10^{-3} order. The best result observed for MAPE was in the neural network with windows size 13 and architecture 55-2-1 whose accuracy for the testing set was of 99.72%. All the MAPE values indicate that the time series approach for rainfall prediction in Manaus is very adequate, delivering correct results rainfall occurence with a high percentage of correct answers.

Evaluating Akaike's Information Criterion

Recalling the importance of AIC to identify the best neural network from a set, we need some parameters which are listed below:

1. Number of input nodes (in);
2. Number of hidden nodes (h);
3. Number of output neurons (o);
4. Residual sum of squares (RSS);

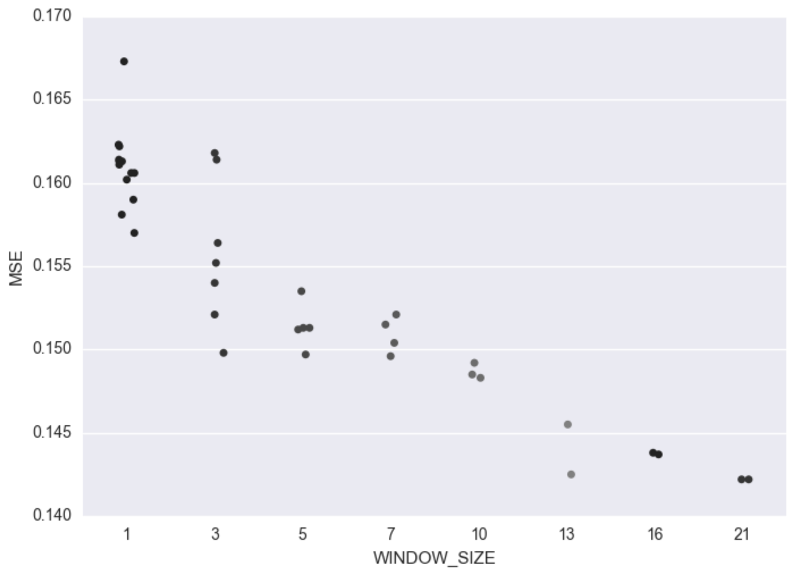


Figure 5: Window size versus MSE.

5. Number of data points per training example (n);

From these parameters we derive k, where:

$$k = in \cdot h + o \cdot h. \quad (3)$$

If the ratio $n/k < 40$, the AIC is obtained as follows:

$$AIC = n \cdot \ln\left(\frac{RSS}{n}\right) + 2 \cdot k + \frac{2 \cdot k \cdot (k+1)}{n-k-1}, \quad (4)$$

otherwise, AIC is given by:

$$AIC = n \cdot \ln\left(\frac{RSS}{n}\right) + 2 \cdot k. \quad (5)$$

Obtaining the AIC values for the neural networks previously built, we have the results shown in Table 4.

From the lowest AIC, we choose the neural network with window size 3 and architecture 15-7-1 as the most adequate to the rainfall occurence forecasting in Manaus. Although the results of MSE and MAPE may indicate neural networks with a larger window size than the former, AIC considers a deeper relation between the model and its generalization capacity for the problem under investigation, not only its performance relative to the testing examples. We recall that the accuracy of the neural network 15-7-1 is of 99.71%, only around 0.01% smaller than the best neural network with window size 21. It is important to emphasize some advantages from the result of AIC because it points out a neural network with less neurons than the one pointed by MSE and MAPE; that this neural network was trained with more data (larger n); and that it has a smaller window size.

Table 4: Results of AIC for the neural networks considered.

w	Architecture	RSS	n	AIC	w	Architecture	RSS	n	AIC
1	7-1-1	1572.70	11718	-23517.63	3	15-7-1	1468.71	11716	**-24105.06**
1	7-2-1	1548.77	11718	-23681.25	5	23-1-1	1516.32	11714	-23901.20
1	7-3-1	1579.96	11718	-23431.63	5	23-2-1	1496.50	11714	-24007.33
1	7-4-1	1549.33	11718	-23645.02	5	23-3-1	1516.32	11714	-23805.20
1	7-5-1	1537.49	11718	-23718.90	5	23-4-1	1484.46	11714	-24005.93
1	7-6-1	1550.65	11718	-23603.01	5	23-5-1	1494.74	11714	-23877.10
1	7-7-1	1529.21	11718	-23750.20	7	31-1-1	1509.98	11712	-23928.21
1	7-8-1	1569.21	11718	-23431.62	7	31-2-1	1498.47	11712	-23953.78
1	7-9-1	1506.00	11718	-23897.44	7	31-3-1	1480.49	11712	-24031.17
1	7-10-1	1520.02	11718	-23772.80	7	31-4-1	1490.30	11712	-23889.85
1	7-11-1	1523.90	11718	-23726.92	10	43-1-1	1504.87	11709	-23934.75
1	7-12-1	1543.28	11718	-23562.86	10	43-2-1	1497.52	11709	-23904.03
1	7-13-1	1532.91	11718	-23625.87	10	43-3-1	1491.47	11709	-23863.45
3	15-1-1	1530.86	11716	-23811.52	13	55-1-1	1494.82	11706	-23980.03
3	15-2-1	1521.62	11716	-23850.41	13	55-2-1	1472.25	11706	-24046.14
3	15-3-1	1505.62	11716	-23942.28	16	67-1-1	1489.54	11703	-23988.27
3	15-4-1	1497.10	11716	-23976.73	16	67-2-1	1482.00	11703	-23911.69
3	15-5-1	1504.49	11716	-23887.08	21	87-1-1	1480.64	11698	-24003.03
3	15-6-1	1492.98	11716	-23945.01	21	87-2-1	1479.98	11698	-23832.25

Final Remarks

We addressed the problem of rainfall occurence forecasting with time series neural networks. We considered the scenario of Manaus from which we used metheorological data from 40 years to train and test 38 different neural networks, considering different window sizes. It was possible to see that the results obtained considering time series delivered a significant accuracy. We were also able to identify that as the window size increases, the MSE for the testing examples decreases. By using Akaike's information criterion, it was possible to choose among the models a 15-7-1 neural network with window size 3 whose accuracy for the problem is equal to 99.71%. The results emphasize and adequation of the considered approach for the problem under consideration.

In future work we aim at using other Computational Intelligence models to deal with the problem under consideration in order to enable comparisons and to search for eventual improvements in the forecasting methodology. We also will use data from weather stations located in other areas of Manaus in order to see how the results can be generalized considering a spatial variation.

Acknowledgments

The authors acknowledge the National Institute of Meteorology (INMET) for the data used in this work. The authors EBG and MBLO acknowledge the research grant provided by the Amazonas State University that supported this work.

References

Abbot, J. and Marohasy, J. (2012). Application of artificial neural networks to rainfall forecasting in Queensland, Australia. In *Advances in Atmospheric Sciences*, volume 29, pages 717–730.

Alavares, C. A., Stape, J. L., Sentelhas, P. C., Gonçalves, J. L. M., and Sparovek, G. (2014). Köppen's climate classification map for Brazil. *Meteorologische Zeitschrift*, 22(6):711–728.

Baum, E. B. and Haussler, D. (1989). What size net gives valid generalisation? *Neural Computation*, (1):151–160.

Bushara, N. O. and Abraham, A. (2013). Computational intelligence in weather forecasting: A review. *Journal of Network and Innovative Computing*, 1:320–331.

da Silva, D. A. (2012). Função da precipitação no conforto do clima urbano da cidade de Manaus. *Revista Geonorte*, 1(5):22–40.

Darji, M. P., Dabhi, V. K., and Prajapati, H. B. (2015). Rainfall forecasting using neural network: A survey. In *International Conference on Advances in Computer Engineering and Applications*, India. IMS Engineering College.

Frank, R. J., Davey, N., and Hunt, S. P. (2001). Time series prediction and neural networks. *Journal of Intelligent and Robotic Systems*, 31:91–103.

Htike, K. K. and Khalifa, O. O. (2010). Rainfall forecasting models using focused time-delay neural networks. In *2010 International Conference on Computer and Communication Engineering*, pages 1–6. IEEE Press.

Hu, M. J. C. (1964). Application of the ADALINE system to weather forecasting. Master's thesis, Stanford El. Lab., Stanford, CA.

INMET (2015). Instituto Nacional de Meteorologia. `http://www.inmet.gov.br`.

Nayak, D. R., Mahapatra, A., and Mishra, P. (2013). A survey on rainfall prediction using artificial neural network. In *International Journal of Computer Applications*, volume 72, pages 32–40.

Palit, A. K. and Popovic, D. (2005). *Computational Inteligence in Time Series Forecasting - Theory and Engineering Applications*. Springer, 1 edition.

Panchal, G., Ganatra, A., Kosta, Y. P., and Panchal, D. (2010). Searching most efficient neural network architecture using Akaike's Information Criterion (AIC). *International Journal of Computer Applications*, 1(5):41–44.

Shukla, R. P., Tripathi, K. C., Pandey, A. C., and Das, I. M. L. (2011). Prediction of Indian summer monsoon rainfall using Niño indices: A neural network approach. In *Atmospheric Research*, volume 102, pages 99–109. Elsevier.

Sioli, H. (1991). *Amazônia: Fundamentos da ecologia da maior região de florestas tropicais*. Vozes.

Zhang, G. P. (2012). *Handbook of Natural Computing*, chapter Neural Networks for Time-Series Forecasting. Springer.