



# Bayesian Inference for Training of Long Short Term Memory Models in Chaotic Time Series Forecasting

Cristian Rodríguez Rivero<sup>1</sup>, Julián Pucheta<sup>1</sup>, Daniel Patiño<sup>2</sup>,  
Jose Luis Puglisi<sup>3</sup>, Paula Otaño<sup>4</sup>, Leonardo Franco<sup>5</sup>, Gustavo Juarez<sup>6</sup>,  
Efrén Gorrostieta<sup>7</sup>, and Alvaro David Orjuela-Cañón<sup>8</sup>✉

<sup>1</sup> Universidad Nacional de Córdoba, Córdoba, Argentina

<sup>2</sup> INAUT-UNSJ, San Juan, Argentina

<sup>3</sup> California North State University, Elk Grove, CA, USA

<sup>4</sup> Universidad Tecnológica Nacional – FRC, Córdoba, Argentina

<sup>5</sup> University of Malaga, Malaga, Spain

<sup>6</sup> Universidad Nacional de Tucumán, San Miguel de Tucumán, Argentina

<sup>7</sup> Universidad Autónoma de Querétaro, Querétaro, Mexico

<sup>8</sup> Universidad del Rosario, Bogotá D.C., Colombia

alvaro.orjuela@urosario.edu.co

**Abstract.** For time series forecasting, obtaining models is based on the use of past observations from the same sequence. In those cases, when the model is learning from data, there is not an extra information that discuss about the quantity of noise inside the data available. In practice, it is necessary to deal with finite noisy datasets, which lead to uncertainty about the propriety of the model. For this problem, the employment of the Bayesian inference tools are preferable. A modified algorithm used for training a long-short term memory recurrent neural network for time series forecasting is presented. This approach was chosen to improve the forecasting of the original series, employing an implementation based on the minimization of the associated Kullback-Leibler Information Criterion. For comparison, a nonlinear autoregressive model implemented with a feedforward neural network was also presented. A simulation study was conducted to evaluate and illustrate results, comparing this approach with Bayesian neural-networks-based algorithms for artificial chaotic time-series and showing an improvement in terms of forecasting errors.

**Keywords:** Bayesian approximation · Time series forecasting · Nonlinear autoregressive models · Recurrent neural networks

## 1 Introduction

The importance of the use of Bayesian methods as a natural methodology for implementation in Time Series Forecasting (TSF) has increased rapidly over the last decade. In particular, this technique provides a formal way to incorporate the prior information from the underlying process related with data generation before of its knowing. Then, this is seen as a resource in sequential learning and decision making, where it is

possible to establish a direct relation between the exact results and the small samples. Moreover, the Bayesian paradigm takes into account all parameters and the uncertainty of the model [1], making relevant the relation between the predictive distribution and the sampling information, where the forecasting is allowed when all parameters are integrated based on a posterior distribution.

Commonly, the selection of a particular model is not specified by some theory or experience, and many adopted models can be trained with the purpose of obtaining that information [2]. Models comparison can be implemented based on a Bayesian framework through the so-called posterior odds, computed as the product of the prior odds and the Bayes factor. This Bayes factor measurement is obtained from any two models, estimated by the likelihood ratio of the marginal likelihood of two competing hypotheses represented by the models, quantifying the support of one model over another based on the available data.

Long short-term memory (LSTM) models are widely utilized for TSF because its architecture based on a special sort of recurrent neural network (RNN) [3, 4]. This kind of this artificial neural network (ANN) architecture is known due to the connections between units, which form a directed cycle. In this way, an internal state of the network is built up, which allows it to exhibit dynamic temporal behavior. In spite of the feedforward architecture of this network, their internal memory can process arbitrary sequences of inputs [5]. However, RNN are difficult to train using the stochastic gradient descent, according to the so-called “vanishing” gradient and/or “exploding” gradient phenomena. This limits the ability of simple RNN to learn sequences with relatively long dependencies [6], making that its employment was reconsidered. For this, proposals like vanilla RNN deal with the vanishing or exploding gradient problem, but remaining the long-term dependence problem, making very difficult the training [7]. Improving the mentioned problems, the LSTM introduces the gate mechanism to prevent back-propagated errors from vanishing or exploding problem, which has been shown to be more effective than conventional RNNs, preventing the overfitting and limitations with long-term sequences [8].

Applications of this kind of ANN can be seen in the work from Zhao et al. [9], which proposed a LSTM network for considering temporal–spatial correlation in traffic system. That ANN was composed of many memory units, comparing this architecture to other representative forecasting models, achieving a better performance. In addition, Kang et al. [10] employed the mentioned RNN to analyze the effect of various inputs settings on its performance. In [11] the authors used a model of ANN combined with a LSTM in a similar way that a Deep Neural Network (DNN), including the autocorrelation coefficients to improve the model accuracy and providing a better precision than traditional ones. Some recent works as in [12], where an adaptation of a LSTM to forecast sea surface temperature (SST) values was employed, including one day and three days information as past inputs. Then the RNN was compared to other models that employed information from weekly mean and monthly mean.

In this paper, a LSTM based on Bayesian Approach (LSTM-BA) method is proposed to predict time series data from well-known chaotic systems. The motivation to use this model is related to the property of this network with one full-connected layer to obtain a regression model for prediction. Also, the LSTM layer has been utilized to model the temporal relationship among time series data, using the Bayes information of

the weights updated and a heuristic approach to adjust the number of training iterations. This requires the ability to integrate a sum of terms in the log joint likelihood using the factorized distribution. In some cases, the integral operations are not in closed form, which is typically handled by using a lower bound showed by Wang et al. [13], where a new method called improved Bayesian combined model.

This work is organized as follows: Section two describes LSTM with the proposed approach (RNN-BA) using Bayesian inference-based heuristic. Section three shows details about the architectures employed for forecasting and an experimental design. The fourth section offers results conducted throughout chaotic time series. Section five provides a discussion based on the used implementation. Finally, section six concludes about this work.

## 2 Recurrent Neural Networks: Long-Short Term Memory

Classical feedforward neural networks, whose connections are directed and without cycles, only maps the current input vector to output vector [14]. This represents a disadvantage because they cannot memorize previous input data, and in addition, a determination of an optimal time lags size cannot be obtained. The mentioned problems are increased due to the input data must be truncated into specific length for developing of the model, producing prediction results not desirable.

Opposite to the classical networks, the RNN allow cyclical or recurrent connections, mapping the complete historical input data to each output. At the same time, these recurrent connections provide a special aspect to memorize information from previous inputs that persists in the network's internal state, influencing the network output. This attribute is useful when noisy signals or sequences with abrupt changes are treated, for example, chaotic series. Different applications have been developed with this architecture, which can change of complexity according to the number of units and connections. Likewise, for standard RNN architecture, the internal influence is given by the number of neurons in the hidden layer, which can decline or blowing up exponentially the value of the synaptic weights. This, according to the cycling behavior established by the recurrent connections. For this, some RNN models are trained with backpropagation through the time and real time recurrent learning for avoiding the vanishing and exploding error problem.

The main advantages of the LSTM are to model long-term dependencies and to determine the optimal time lags for TSF problems. These aspects are desirable for long-term future predictions specially, due to the lack of priori knowledge between samples. In addition, the problem of the sum of error signals increment, a proposal based on the constant error carousel (CEC) was proposed in a first version. The LSTM architecture is modified, including a pair of gates, which can allow the flow from inputs to outputs. In the enhanced version, it adds a reset gate called forget gate [8], including the notion of memory cells as shown in Fig. 1 by the yellow blocks.

In the present proposal, a LSTM model is trained with focus on the exploration of solutions by employing a Bayesian heuristic method. In this way, an improvement given by the overfitting problem was searched [17]. Therefore, in order to make the topology of LSTM as simple as possible, it is important to delete unnecessary units,

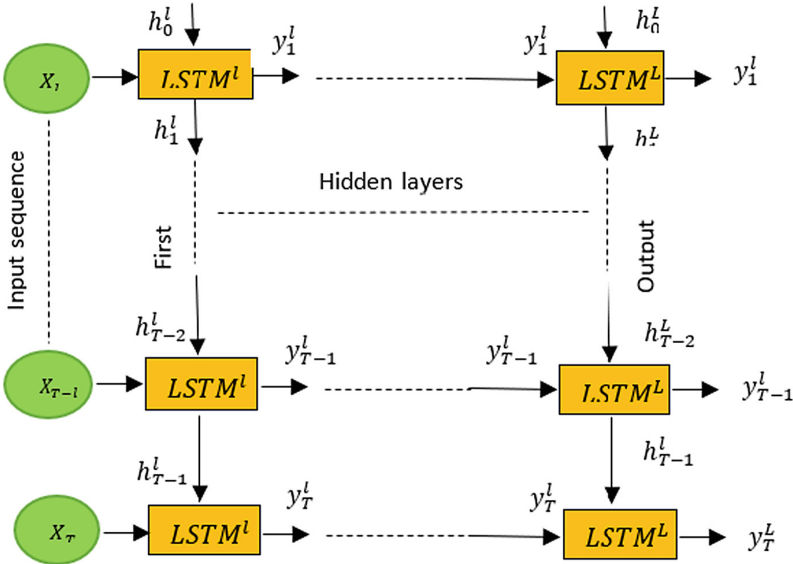


Fig. 1. Flow chart inside the LSTM RNN. (Color figure online)

layers and connections, thus, to optimize the training and topology of RNN, we proposed a heuristic adjustment, as follows:

$$1 \leq i_t \leq 2(H_0 - KL) \tag{1}$$

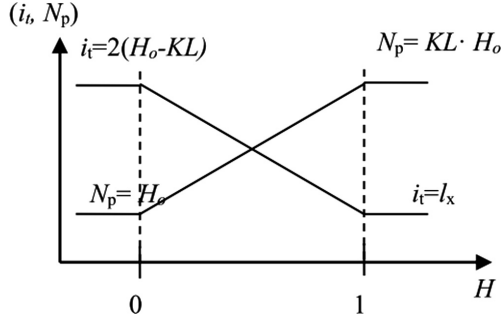
$$H_0 \leq N_p \leq KL + I_x \tag{2}$$

where  $I_x$  is the dimension of the input vector,  $N_p$  is the number patterns, and  $H_0$  is the initial value of hidden neurons. Then, a heuristic adjustment for the pair  $(i_t, N_p)$ , the number of iterations and patterns as function of the hidden units  $H_0$  and  $KL$  (Kullback-Leibler Information Criterion), according to the membership functions shown in Fig. 2. Finally, an approximation for the network weights and biases was developed, where all the model parameters were modelled as Gaussian distributions with a diagonal covariance. An exception for latent states, which was modelled as a Gaussian distribution with an almost diagonal covariance.

### 3 Experimental Design

Experiments with data generated from an artificial chaotic systems, which were performed to obtain five common benchmark series with length of 1500 points. This length was chosen intentionally, for determining whether this a limitation in order to compare the model against other much simpler models with probable less overfitting.

For chaotic series, the generation of such aspect belongs to reaching the attractor. To do this, the system was allowed to evolve after one hundred samples, achieving the



**Fig. 2.** Heuristic adjustment of  $(i_t, N_p)$  in terms of  $KL$  after each epoch.

corresponding attractor. According to this, we assumed that the number of iterations guarantees the state of the dynamic system, ending the initial transient on the attractor. In spite of this, there are no reliability that the system will revisit future events similar to those observed during training.

### 3.1 Employed Datasets

For assessment of built models, seven datasets with artificial chaotic time series were used. Each of these series is describe next:

**Mackey-Glass Series:** The dataset MG17 and MG30 are by sampling the Mackay-Glass (MG) equations, given by the expression (3), as follows:

$$\dot{x}(t) = \frac{ax(t - \tau)}{1 - x(t - \tau)^c} - bx(t) \tag{3}$$

with  $a, b, c, \tau$  setting parameters shown as follows in Table 1.

**Table 1.** Parameters to generate MG time series

Series	$N$	$a$	$b$	$c$	$T$
MG17	1500	0.2	0.1	10	17
MG30	1500	0.2	0.1	10	30

**Logistic Time Series:** The dataset LOG01 and LOG02 series were mapped from logistic system and represent in (4), which is defined by:

$$x(t + 1) = ax(t)[1 - x(t)] \tag{4}$$

where  $a = 4$ , the iterations in Eq. (4) perform a chaotic time series (see Table 2).

**Table 2.** Parameters to generate LOG time series

Series	$N$	$a$	$X_0$
LOG01	1500	4	0.1
LOG02	1500	4	0.3

**Henon Chaotic Time Series:** This time series can be constructed by following Eq. (5), however, it presents many aspects of dynamical behavior of more complicated chaotic systems.

$$x(t + 1) = b + 1 - ax^2 \tag{5}$$

where  $a$  and  $b$  were fixed as shown in Table 3. These same parameters were used in both cases.

**Table 3.** Parameters to generate HEN time series

Series	$N$	$a$	$b$	$X_0$	$Y_0$
HEN01	1500	1.4	0.3	0	0
HEN02	1500	1.3	0.22	0	0

**Lorentz Time Series:** The Lorenz model is given by the Eq. (6), the data is derived from the Lorenz system, which is given by three time-delay differential systems. Table 4 specifies the values employed to generate the data samples.

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = bx - y - xz \\ \frac{dz}{dt} = xy - bz \end{cases} \tag{6}$$

**Table 4.** Parameters to generate LOR time series

Series	$N$	$a$	$b$	$c$	$X(0)$	$Y(0)$	$Z(0)$
LOR01	1500	0.2	0.2	5.7	12	9	2
LOR02	1500	0.42	0.42	0.42	0.1	0.1	2

**Rössler Chaotic Time Series:** In this example, the data was derived from the Rössler system, which is given by three time-delay differential systems represented in expression (7).

$$\begin{cases} \frac{dx}{dt} = -y(t) - z(t) \\ \frac{dy}{dt} = x(t) + ay(t) \\ \frac{dz}{dt} = b + z(t)(x(t) - c) \end{cases} \tag{7}$$

The dataset was built by using four-order Runge–Kutta method with the initial value as shown in Table 5, and the step size was chosen as 0.01.

**Table 5.** Parameters to generate ROS time series

Series	$N$	$a$	$b$	$c$	$X(0)$	$Y(0)$	$Z(0)$
ROS01	1500	0.2	0.2	5.7	12	9	2
ROS02	1500	0.42	0.42	0.42	0.1	0.1	2

**Ikeda Time Series:** The Ikeda map was given in expression (8) as follows:

$$\begin{aligned} \dot{x} &= 1 - \mu[x\cos(t) - y\sin(t)] \\ \dot{y} &= \mu[x\cos(t) - y\sin(t)] \end{aligned} \tag{8}$$

where  $t = 1/(1 + x^2 + y^2)$ . This system displays chaotic behavior over a range of values for the parameter, including the values chosen in Table 6.

**Table 6.** Parameters to generate IK time series

Series	$N$	$\mu$	$X(0)$	$Y(0)$
IK01	1500	0.9	0.5	0.5
IK02	1500	0.8	0.9	0.6

In the experiments, the datasets were splitted into two parts: the training set and the test set. In the training phase, each of the individual models was trained with optimized parameters given by each filter. This means that every model was constructed with a sequence of data different to the test set samples. Figure 3 shows examples from each generated dataset for the five cases.

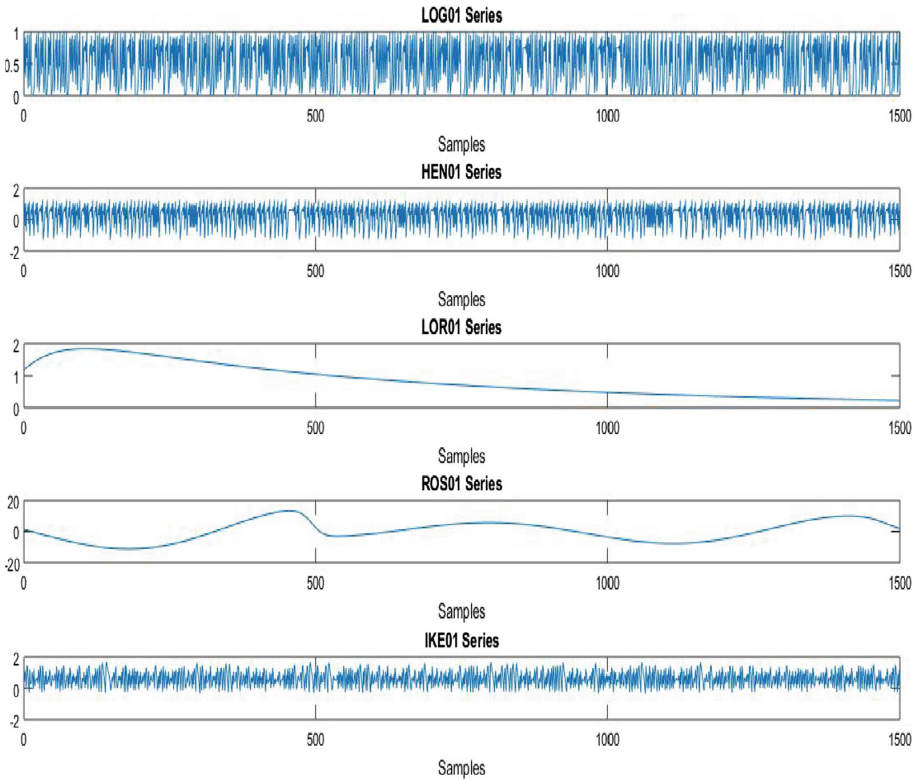
### 3.2 Neural Networks Models

The architecture of the LSTM model was composed by an input with length ( $i_x$ ) of 25 samples. The nonlinear gate was given by the sigmoid function and the nonlinearity from input to output was established by a hyperbolic tangent function. The number of epochs for training was adjusted to 50 as maximum. Learning rate was  $2e-3$  with a training percent of 0.80 and 512 units in just one hidden layer, dropout rate of 1.0 and weight decay of  $1e-8$ .

As a way to compare the results in terms of ANN models, a nonlinear autoregressive model (NAR) was employed. This is an architecture that is based on feed-forward connections as classical proposals of neural networks but with considerable results for forecasting tasks [15–17]. The main difference is determined by the recurrence that is missing in this model as previously mentioned. For developing the forecasting the expression (7) describes this model, in the way:

$$y_i = \tanh\left(\sum_{k=1}^p a_i y_{i-k} + b\right) \tag{9}$$

where  $y_i$  is the time series to be modeled,  $a_i$  are the coefficients of the model, which are called synaptic weights ( $w_{ij}$ ) in other applications models. Parameter  $b$  is a bias value used to fix the function to be found. It is possible to see nonlinearity in the hyperbolic tangent ( $\tanh$ ) in (3) known as transfer function of the units or neurons in the neural model.



**Fig. 3.** Examples of the generated datasets.

For NAR model, some equivalent parameters were fixed as LSTM model. For example, an input vector with 25 samples was used and 50 epochs for training. One hidden layer was also employed, modifying the number of neurons from two to ten and computing the results in an experimental mode. The resilient backpropagation algorithm was employed for adjusting the synaptic weights due to its fast convergence and low computational cost.



### 3.3 Prediction Error Metrics

To assess the performance of the forecasting, the symmetric mean absolute error (sMAPE) was employed as suggested in [18], and as shown in expression (10):

$$sMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|X_t - F_t|}{(|X_t| + |F_t|)/2} * 100 \tag{10}$$

where  $t$  is the observation time,  $n$  is the size of the test set,  $X_t$  is the original series and  $F_t$  is the forecasted series.

Finally, the Root Mean Square Error (RMSE) was employed to obtain the error as in [19], given by the computation in (11) as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_t - F_t)^2} \tag{11}$$

where  $t$ ,  $n$ ,  $X_t$  and  $F_t$  are the same as in (10).

## 4 Results

Table 7 shows the results computed for sMAPE and RMSE average across all datasets when the LSTM model and its modification with the Bayesian Approach (LSTM-BA) were trained. Two horizons for this forecasting were highlighted because the best results. As a complement, Table 8 shows the results for NAR implemented models, where information about the neurons in the hidden layer and average for sMAPE and RMSE from computation in all datasets are shown.

**Table 7.** Mean forecast-error metrics computed by LSTM approaches employing all datasets.

Method	Horizon Forecast of 3 out-of-sample		Horizon Forecast of 18 out-of-sample	
	sMAPE	RMSE	sMAPE	RMSE
	LSTM-BA	<b>0.056</b>	<b>0.132</b>	<b>0.033</b>
LSTM	0.057	0.138	0.039	0.135

The BA approach shows a level improvement, indicating the necessity of information from prior distribution for an adequate model with better results in terms of prediction error. The evaluation of the results across the 10 series analyzed through its mean value evinced, with the use of sMAPE and RMSE indices, that there was an increment when the horizon is deeper for each series. Note that there is a little improvement of the forecasting given by LSTM-BA approach compared with the traditional LSTM one, which resulted from the use of a stochastic characteristic to generate a deterministic result for long-short-term prediction.

**Table 8.** Forecast-error metrics computed by NAR approaches employing all datasets.

Neurons	Measures	
	RMSE	sMAPE
2	0.465	0.099
3	0.301	0.081
4	0.336	0.085
5	<b>0.269</b>	<b>0.072</b>
6	0.299	0.077
7	0.301	0.074
8	0.311	0.076
9	0.415	0.103
10	0.329	0.078

For the NAR model, the results remained under the results of the LSTM, showing the advantages of RNN compared with feedforward networks. In spite of the comparison was not equitable due to the number of units and optimization parameters to train the models, the results exhibit advantages of the recurrent strategies.

## 5 Discussion

The assessment of the obtained results, comparing the performance of the proposed algorithm, shows a significance improvement measured either by sMAPE and RMSE index for the LSTM-BA and LSTM ones contrasted to NAR models. According to the literature, it is not properly justified or experimentally proven that LSTM networks are appropriate for modeling chaotic series. However, as consequence of using the present proposal, there was an increment of the network in terms of learning of long sequences. This approach came from the idea that only the most recent data are important, and the sliding time window methods are very useful for pattern recognition, for datasets with highly dependence on the past bulk of observations.

LSTM models are powerful enough to learn the most important past behaviors and understand whether those past samples are important features in the making prediction process. This could not be exposed by the NAR models, which, in spite of its utility in TSF, did not have better results than the RNN proposals. As mentioned, recurrent connections present capacity to memorize nonlinear relations between the samples, which allow to performance the forecasting in a better way. Other aspect for the NAR model is related with the number of neurons in the hidden layer. For both error measures, it was exhibited that with five neurons the models presented the best performance. After this number, the networks showed an overfitting incrementing the error in the test set.

In the specific case of chaotic time series, alternative models have been worked based on other Bayesian proposals. Examples of this, can be seen in [20], where a Bayesian enhanced ensemble approach, or in [21] with a Bayesian enhanced modified

proposal, and in [22, 23] with BA extended and BA basic proposals, respectively. In all these models, the results were comparable and lightly under the present ones.

The limited amount of available input and a flexible prior over a large space of possible nonlinear models produces significant posterior uncertainty on the dynamics and the global prediction, converging to the long-term mean with large variance. This is due to the poor estimation of the mean error values using so little amount of data. In order to improve the value of the results, more prior information such as the apparent periodicity and trend of the signal must be considered.

## 6 Conclusions

In this paper, a proposal that includes a Bayesian heuristic approach to optimize the training and architecture of LSTM, allowing modifying its parameters in a better way. For this, the addition of information as the number of feedback layers, past input layers and incorporation of self-adaptive heuristic, adjusts the training process. Furthermore, we have shown that our model yields neural networks with higher prediction capability for time-series data, comparing the performances of the proposed algorithm and the existing one through the numerical experiments, using well-known benchmark series. An alternative proposal based on NAR models was presented with low performance compared to the LSTM ones.

## References

1. MacKay, D.J.C., Mac Kay, D.J.C.: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge (2003)
2. Hippert, H.S., Taylor, J.W.: An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting. *Neural Networks* **23**, 386–395 (2010)
3. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: a search space odyssey. *IEEE Trans. Neural Networks Learn. Syst.* **28**, 2222–2232 (2016)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997)
5. Karpathy, A., Johnson, J., Fei-Fei, L.: Visualizing and understanding recurrent networks. arXiv Preprint [arXiv:1506.02078](https://arxiv.org/abs/1506.02078) (2015)
6. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: *International Conference on Machine Learning*, pp. 1310–1318 (2013)
7. Williams, R.J., Zipser, D.: Gradient-based learning algorithms for recurrent. In: *Backpropagation Theory, Architectures and Applications*, p. 433 (1995)
8. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014)
9. Zhao, Z., Chen, W., Wu, X., Chen, P.C.Y., Liu, J.: LSTM network: a deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **11**, 68–75 (2017)
10. Kang, D., Lv, Y., Chen, Y.: Short-term traffic flow prediction with LSTM recurrent neural network. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6 (2017)

11. Zhuo, Q., Li, Q., Yan, H., Qi, Y.: Long short-term memory neural network for network traffic prediction. In: 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 1–6 (2017)
12. Zhang, Q., Wang, H., Dong, J., Zhong, G., Sun, X.: Prediction of sea surface temperature using long short-term memory. *IEEE Geosci. Remote Sens. Lett.* **14**, 1745–1749 (2017)
13. Wang, J., Deng, W., Zhao, J.: Short-term freeway traffic flow prediction based on improved Bayesian combined model. *J. Southeast Univ. (Nat. Sci. Ed. 1)* **42**, 162–167 (2012)
14. Haykin, S.: *Neural Networks and Learning Machines*. Prentice Hall, Upper Saddle River (2009)
15. Orjuela-Cañón, A.D., Hernández, J., Rivero, C.R.: Very short term forecasting in global solar irradiance using linear and nonlinear models. In: 2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA), pp. 1–5 (2017)
16. Almonacid, F., Pérez-Higueras, P.J., Fernández, E.F., Hontoria, L.: A methodology based on dynamic artificial neural network for short-term forecasting of the power output of a PV generator. *Energy Convers. Manage.* **85**, 389–398 (2014)
17. Men, Z., Yee, E., Lien, F.-S., Wen, D., Chen, Y.: Short-term wind speed and power forecasting using an ensemble of mixture density neural networks. *Renew. Energy* **87**, 203–211 (2016)
18. Davydenko, A., Fildes, R.: Forecast error measures: critical review and practical recommendations. In: *Business Forecasting: Practical Problems and Solutions*, p. 34. Wiley (2016)
19. Chai, T., Draxler, R.R.: Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **7**, 1247–1250 (2014)
20. Rivero, C.R., et al.: Bayesian enhanced ensemble approach (BEEA) for time series forecasting. In: 2018 IEEE Biennial Congress of Argentina (ARGENCON), pp. 1–7 (2018)
21. Rivero, C.R., Pucheta, J., Baumgartner, J., Laboret, S., Sauchelli, V.: Short-series prediction with BEMA approach: application to short rainfall series. *IEEE Lat. Am. Trans.* **14**, 3892–3899 (2016)
22. Rivero, C.R., Pucheta, J.A., Sauchelli, V.H., Patiño, H.D.: Short time series prediction: Bayesian Enhanced modified approach with application to cumulative rainfall series. *Int. J. Innov. Comput. Appl.* **7**, 153–162 (2016)
23. Rivero, C.R., Patiño, D., Pucheta, J., Sauchelli, V.: A new approach for time series forecasting: bayesian enhanced by fractional brownian motion with application to rainfall series. *Int. J. Adv. Comput. Sci. Appl.* **7** (2016)