Efficiency of recurrent neural networks for seasonal trended time series modelling

Rida El Abassi[1], Jaafar Idrais[1], Abderrahim Sabour[1,2]
[1]IMI Laboratory, Department of Mathematics, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
[2]Department of Computer Science, High School of Technology, Ibn Zohr University, Agadir, Morocco

Article Info

Article history:

Received May 4, 2022
Revised Feb 9, 2023
Accepted Mar 9, 2023

Keywords:

Automatic learning
Long short-term memory
Machine learning
Recurrent neural network
Time series

ABSTRACT

Seasonal time series with trends are the most common data sets used in forecasting. This work focuses on the automatic processing of a non-pre-processed time series by studying the efficiency of recurrent neural networks (RNN), in particular both long short-term memory (LSTM), and bidirectional long short-term memory (Bi-LSTM) extensions, for modelling seasonal time series with trend. For this purpose, we are interested in the learning stability of the established systems using the mean average percentage error (MAPE) as a measure. Both simulated and real data were examined, and we have found a positive correlation between the signal period and the system input vector length for stable and relatively efficient learning. We also examined the white noise impact on the learning performance.

Corresponding Author:

Rida El Abassi
Laboratory of Mathematical and Computer Engineering, Faculty of Sciences, Ibn Zohr University
Agadir, Morocco
Email: rida.elabassi@edu.uiz.ac.ma

1. INTRODUCTION

The analysis of time series represents a source of knowledge and information given the amount of data generated through technical and technological development, which multiplies the fields of application for this discipline. In the field of time series, researchers tend to propose models describing the underlying relationship of the generator process and to forecast time series [1]. The seasonal and trend components are characteristics of several time series resulting from economic phenomena. Seasonality is considered as a periodic and recurring pattern, while the trend component characterizes the long-term evolution of the time series studied. The importance of accurate forecasting of seasonal time series trends is crucial for areas such as marketing, inventory control and many other business sectors.

The traditional methods of time series analysis proceed with two main steps: decomposition, then reconstitution of series to carry out the forecast [2]. This approach assumes that the structure of time series can be decomposed into modellable elements [3]. There are three main components: the trend T_t, which describes the long-term evolution and the phenomenon's pattern, the seasonal component S_t, which characterizes repetition over time, and the residual component R_t, which represent the noise [4].

In the 1970s, Box and Jenkins introduced another perspective on time series modelling [5], named Box and Jenkins methodology, it is based on the Wold's representation theorem [6]–[9]; in fact, once a process is (weakly) stationary, it can be written as the weighted sum of past shocks. This is how the notion of stationary becomes fundamental to the analysis process [10]. However, a seasonal and trend time series is considered to be non-stationary and often needs to be made stationary, using a certain seasonal adjustment method [11], before most modelling and forecasting processes take place.

Moreover, neural networks (NN) offer new perspectives [12]–[14] for modelling time series than traditional seasonal autoregressive integrated moving average (SARIMA) models [15], [16]. The learning mechanism allows to establish a neural architecture based on parameters such as the size of the input vector, and the number of hidden layers. Indeed, NN have been widely applied to many fields through their flexibility to design a network structure [17]. The fully connected NN (FNN) is a basic structure of neural networks, Qi and Zhang [18] implemented this structure to seasonal time series with trends, indeed, they conducted experiments by comparing the two models, autoregressive integrated moving average (ARIMA) and FFN, which report that an FNN cannot directly model seasonality, however, a preprocessing step is needed involving seasonal and trend adjustments for proper modelling. Liu *et al.* [19] also compares FFN and ARIMA using the same type of simulated time series, this study concludes that by choosing rectified linear unit (ReLU) or the linear activation function and Adam optimizer, the FFN model performs well.

The motivation for this works was inspired by Qi and Liu studies [18], [19], in which the authors compare the performance of SARIMA to an FNN and a convolutional NN. In this paper, we plan to use a recurrent neural network (RNN), in particular, long short-term memory (LSTM) and bidirectional long short-term memory (Bi-LSTM) extensions. However, their experiments and conclusions are inadequate for our purposes.

The aim of our study is to find a modelling method such that users do not have to worry about preprocessing time series. Thus, the initial motivation of this paper is to develop a machine learning tool to predict time series data without manual intervention, using recurrent neural networks. The main problem is to find a general-purpose modelling method or algorithm that can handle seasonality, trends and auto-correlations in time series data. It is important to note that the initial question was about the choice of these parameters, in particular the size of the input vector and the number of hidden layers for additive and multiplicative signals.

2. METHOD

2.1. Principle of LSTM and Bi-LSTM structures

Extensions of recurrent neural networks (RNNs) such as LSTMs are the most feasible solutions since, they are directed to the problem of the gradient disappearance by managing short-and long-term memory. They anticipate future predictions based on various highlighted characteristics present in the dataset. LSTMs can remember or forget things precisely. Data collected on progressive timescales is presented as time series, and let to make predictions, while LSTMs are proposed as a stable methodology. In this type of design, the model passes the past protection state to the next stage of the layout. Since RNNs can only store a limited amount of information for long-term memory storage, LSTMs cells are used with RNNs [20]. They overcome the difficulties of leakage gradient and explosion gradient and have the ability to support long-term dependencies by replacing the hidden layers of RNN with memory cells. The LSTM block contains three gates [21] and each gate corresponds to a processing step. Standard recurrent neural architectures, like LSTM, treat the inputs in one direction only and ignore the possessed information about the future. The bi-directional LSTM (Bi-LSTM) model responds to this issue in its operating process [22].

For the Bi-LSTM topology [23]–[26], the information flows in two directions as illustrated in Figure 1, taking into account the temporal dynamics of the vectors of past and future inputs. Standard RNN's hidden neurons are split forward and backward. The basic structure of Bi-LSTM [27] is unfolded in three-time steps: forward pass, backward pass, and weight update.

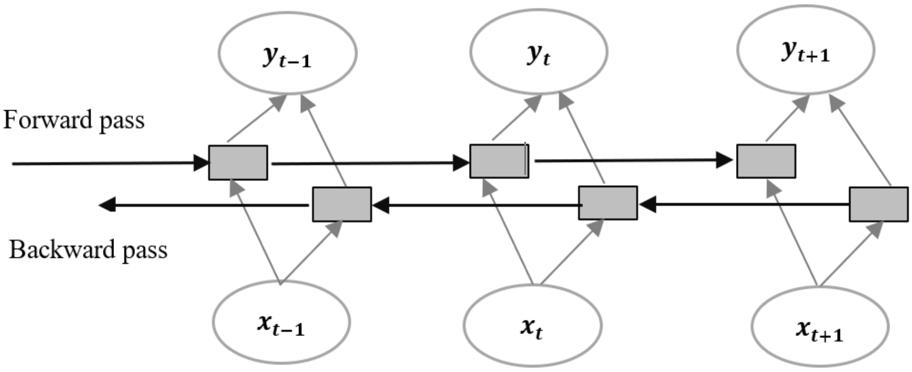


Figure 1. Example of Bi-LSTM structure

2.2. Processing strategy

The first questions at the origin of this study were mainly related to the capacity of RNNs to model the regularities of a signal, specifically the seasonality and the trend. Then by developing a neural model, we realized that several parameters are put into the equation, namely, the size of the input vector, and the number of hidden layers. This work empirically highlights a correlation between the period of the time series and the size of the input vector for stable and relatively successful learning. This study is conducted on twenty-six time series derived from a real phenomenon, the first characterizes the evolution of the number of passengers in an international airport, which is a classic signal in the literature [4], it was the first signal for which the two researchers George Box and Gwilym Jenkins established their methodology, then the second characterizes the evolution of CO_2 concentrations in the air measured during 1974 through 1987 [28]. In another sense, it is important to note that the time series belong to two basic classes of models, namely the additive and the multiplicative models, in order to analyze the robustness of the RNNs not only to the change of fluctuation at the seasonal level but also to the impact of white the noise and then to draw conclusions on the stability of the established systems.

It is important to note that the learning process will be carried out by a part of the signal noted (Train), which represents nearly 80% of the size of the basic signal. However, 20% will represent the part (Test), which allow us to measure the performance of the learning carried out via the mean absolute percentage error (MAPE) given in (1). The next step is to make a prediction of 100 future observations in order to analyze the prediction of the system and its ability to detect the regularities of the signal and under what conditions, and if the system has taken into account the regularities of the signal (Tt and St). We apply the low-frequency filter (the moving average) by changing the window l to determine the period of the predicted signal. In Figure 2, we have displayed the whole layout of the proposed model.

$$MAPE = \left(\frac{1}{k}\sum_{x=1}^{k}\left|\frac{a_x - f_x}{a_x}\right|\right) \times 100 \tag{1}$$

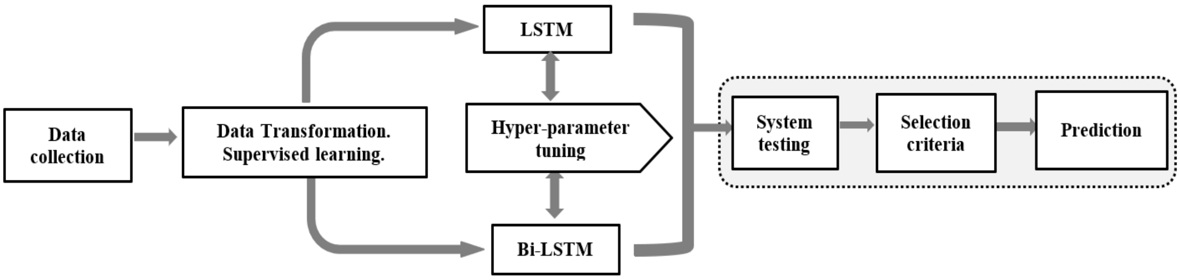


Figure 2. Layout of proposed method

2.3. Simulated data

The design of the methodology of this empirical analysis focuses on the use of several time series with different periods and variance σ^2 of white noise, we generate the time series via (2) and (3).

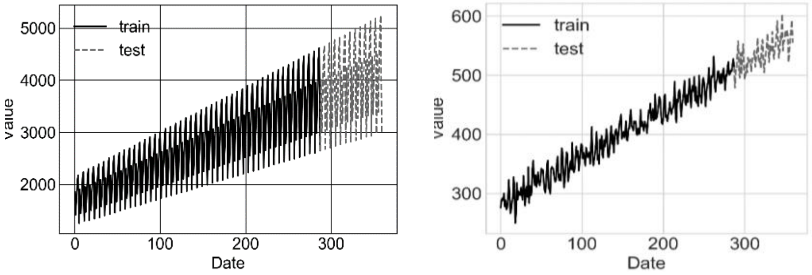
$$y(t) = SI(t) + T(t) + E(t) \tag{2}$$

$$y(t) = SI(t) \times T(t) + E(t) \tag{3}$$

In (2) and (3) are a characteristic of the additive (AM) and multiplicative (MM) model respectively, such that the *SI*(*t*) is the seasonality index, Table 1 shows the measures adopted for MM and MA. *T*(*t*) is the linear trend, and E(t) is the distribution error that follows the normal distribution $N(0, \sigma^2)$. Note that for each given *SI* we assign σ^2 three values σ=1, 5, 12. Indeed, controlling the seasonality index allows us to fix the period of seasonality and then see the reaction of the established systems with respect to these changes. On the other hand, the change σ allows us to test the robustness of the established assumptions with respect to the increase of the white noise energy in the signal. Figure 3 shows an example of the time series that we have generated with a seasonality index SI given in Table 1, for MM in Figure 3(a) and for an AM in Figure 3(b), the part *T* that characterizes the trend is given by *T(t)=0.8t+150* for any $t \in$ [0.359], i.e., this series as well as all the others generated from both MM and AM will have 360 observations. The white noise variance σ^2 of this time series is given by ($\sigma^2 = 144$).

Table 1. Seasonal indexes used for simulated monthly data

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
SI MM	0.8	0.85	0.87	0.95	0.99	0.97	0.96	1.04	0.99	1.07	1.25	1.85
SI AM	89	96	99	105	119	98	92	136	93	149	168	194

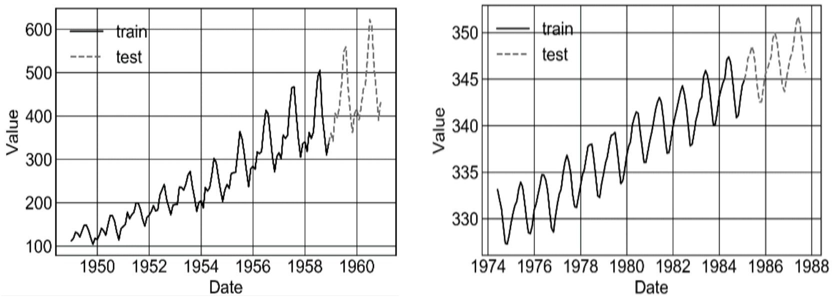


(a) (b)

Figure 3. Plot the simulated time series (a) through the multiplicative model and (b) through the additive model

2.4. Real data

Figure 4 represents the real database; Figure 4(a) shows the monthly evolution of total passengers in an international airport in the period from January 1949 to December 1960. Box and Jenkin applied their methodology to this time series and proposed linear models of class SARIMA that they developed [5]. The time series shows an upward tendency and a seasonality of period *p=12* that changes in fluctuation in the course of the time. While Figure 4(b) illustrates the evolution of the air's CO_2 concentration from May 1974 to September 1987.



(a) (b)

Figure 4. Plot the real time series (a) monthly evolution of total passengers in an international airport in the period from January 1949 to December 1960 and (b) evolution of CO_2 concentration in the air in the period May 1974 to September 1987

2.5. Modeling strategy

The databases we manipulated in this study are univariate time series. We implemented two models of recurrent neural networks in particular LSTM and Bi-LSTM, using libraries such as NumPy [29], Pandas [30], Keras and TensorFlow [31]. For a given signal with fixed period p and white noise variance σ^2, we performed in the learning by train part while varying the size of the input vector ve and took the following values: 3, 4, 9 and 12. In other words, we have performed for a given signal four tests, this allows us to note the correlations of the different parameters of the system. Figure 5 shows the neural structure adapted for the two models, LSTM and Bi-LSTM. Noting that the structure is the same for both models, it is composed of an input layer with ve inputs and connected to 256 neurons of the first hidden layer. The neural structure has six hidden layers, the choice of the number of neurons is based on the remark of Moolayil [32], concerning the number of hidden layers, which is one of the questions of this study. How to choose the number of hidden layers intelligently in relation to the particularity of the signal to guarantee the performance of the learning. We conducted experiments in this direction, but they did not lead to consistent results.

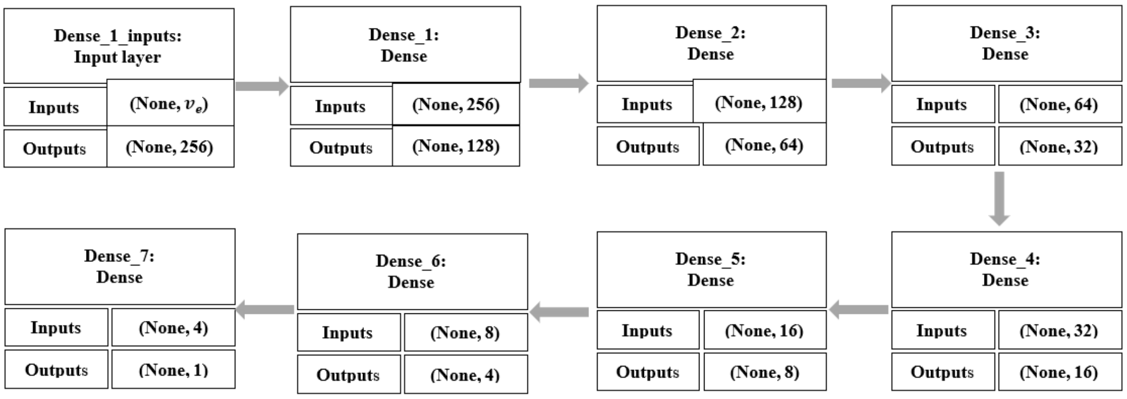


Figure 5. RNN (LSTM and Bi-LSTM) structure with 256 input neurons and 7 hidden layers

Table 2 shows the different parameters of the neural architectures established. For the RNNs model LSTM and its extension Bi-LSTM, the activation function is the ReLU, the optimization algorithm held is the Adam's and the cost function employed is the mean-squared error (MSE). The learning algorithm tries to minimize the cost function, which characterizes the distance between real and predicted values at the same time, by adjusting the weights and bias of the system. The starting point of these parameters impacts the learning performance, a decision is made to initialize the weights and bias of the neural architecture to the same values for all performed tests.

Table 2. Parameter settings of the studied approaches

Models	Parameters	Values
LSTM and Bi-LSTM	Input	ve
	Output	1
	Hidden Layer	7
	Neurons	256-128-64-32-16-8-4
	Optimizer	Adam
	Features	1
	activation function	ReLU
	Loss function	MSE
	Training epochs	500

3. RESULTS AND DISCUSSION

3.1. Simulated data results and discussion

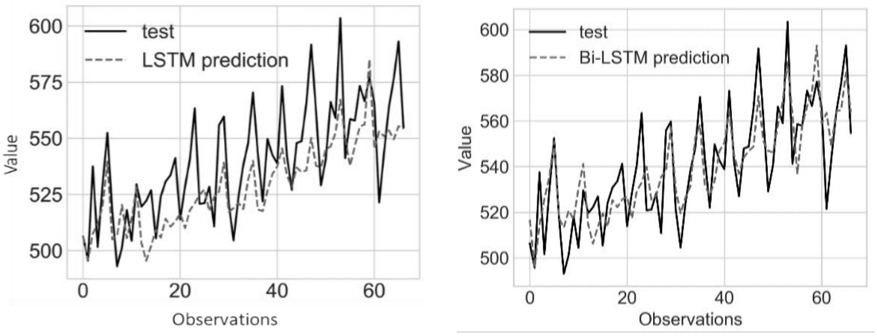
We established LSTM and Bi-LSTM to try to answer the question reported in section 2. We adopted the same parameters indicated in Table 2 of section 2 for all the neural architectures established. The purpose of this study does not take into consideration the comparison of the different architectures such as Adam algorithm, adaptive gradient algorithm (AdaGrad) and stochastic gradient descent (SGD) or the different existing cost functions. Table 3 shows the results of the tests carried out using the methodology reported in section 2 to generate the time series, according to AM and MM characterized respectively by the formulas 2 and 3. We raise two remarks: firstly, there is a correlation between the period of the signal and the size of the input vector ve, meaning that, to guarantee the relative performance of the learning, it is more appropriate to choose $ve = p$, and this is for the two extensions of the recurrent neural networks LSTM and Bi-LSTM. Secondly, the white noise impacts the learning performance.

Figure 6 shows the learning result for AM, Figures 6(a) and 6(b) illustrate the performance of the LSTM and Bi-LSTM models, respectively. The MAPE, as shown in Table 3, is of order 0.53 and 0.49, respectively. Figure 7 characterizes the learning results for MM, Figures 7(a) and 7(b) illustrate the performance of the LSTM and Bi-LSTM models, respectively. The MAPE, as shown in Table 3, is of the order 0.17 and 0.09 respectively.

Remember that the comparison of the two models LSTM and Bi-LSTM is not the goal of this work. The initial question was mainly focused on the stability of learning, and to ensure an adequate model for signals, characterized by a seasonal and a trend via RNNs models. Two interesting results are deduced: first, a significant correlation exists between the size of the input vector of the system ve and the period of the signal, second, the noise has an impact on the learning.

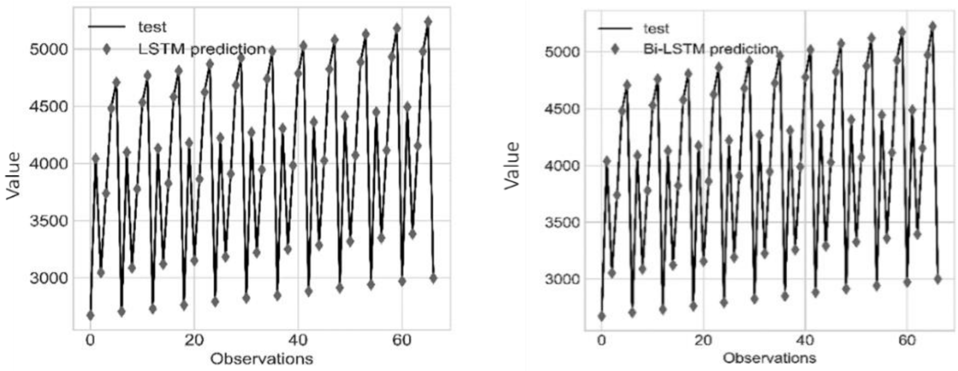
Table 3. Simulation result for neural networks

p	σ^2	ve	MAPE					MAPE			
			Additive model		Multiplicative model			Additive model		Multiplicative model	
			LSTM	Bi-LSTM	LSTM	Bi-LSTM		LSTM	Bi-LSTM	LSTM	Bi-LSTM
$p = 3$	σ = 1	$ve = 3$	0.01	0.04	0.03	0.12	$p = 6$	2.23	1.72	2.79	1.29
		$ve = 4$	10.77	13.61	12.65	13.76		10.18	6.49	17.11	12.36
		$ve = 6$	5.61	8.37	23.46	11.54		0.03	0.02	0.02	0.02
		$ve = 12$	1.47	1.16	10.34	1.98		1.48	0.53	1.38	2.11
	σ = 5	$ve = 3$	0.09	0.14	0.08	0.14		4.83	6.35	3.65	2.32
		$ve = 4$	12.42	12.27	7.38	14.61		18.43	21.73	14.57	17.52
		$ve = 6$	10.35	11.48	6.29	16.92		0.12	0.21	0.17	0.08
		$ve = 12$	3.29	9.11	5.81	10.73		12.37	2.43	16.49	19.53
	σ = 12	$ve = 3$	1.91	1.45	2.41	1.78		11.62	18.42	13.64	15.11
		$ve = 4$	17.19	25.18	24.73	38.41		20.46	17.31	19.43	26.59
		$ve = 6$	13.22	24.62	21.22	21.39		1.48	0.82	0.42	0.53
		$ve = 12$	10.41	12.43	16.73	18.37		4.41	2.93	14.28	22.48
$p = 4$	σ = 1	$ve = 3$	8.83	3.78	6.48	9.57	$p = 12$	2.87	1.39	2.47	5.83
		$ve = 4$	0.02	0.01	0.05	0.15		17.32	10.45	9.33	10.82
		$ve = 6$	3.56	7.51	7.51	10.35		5.63	2.54	1.43	4.69
		$ve = 12$	2.84	4.68	3.71	10.51		0.21	0.01	0.02	0.06
	σ = 5	$ve = 3$	14.64	12.63	13.27	15.61		3.67	4.78	15.22	16.43
		$ve = 4$	0.12	0.17	0.28	0.18		21.86	14.26	8.62	15.73
		$ve = 6$	18.51	12.85	18.29	23.51		6.17	4.34	3.45	5.22
		$ve = 12$	6.42	3.57	8.38	17.32		0.37	0.15	0.11	0.68
	σ = 12	$ve = 3$	14.18	8.52	19.27	11.69		3.78	4.38	17.64	16.44
		$ve = 4$	2.79	0.27	0.74	0.42		20.41	15.26	13.63	12.85
		$ve = 6$	21.48	14.27	18.63	17.65		9.54	5.86	4.57	4.12
		$ve = 12$	6.92	6.27	16.38	16.84		0.53	0.49	0.17	0.09



(a) (b)

Figure 6. Predicted values versus true values on the training data ($\sigma = 12$) for AM, in (a) the LSTM model results and (b) the Bi-LSTM model results

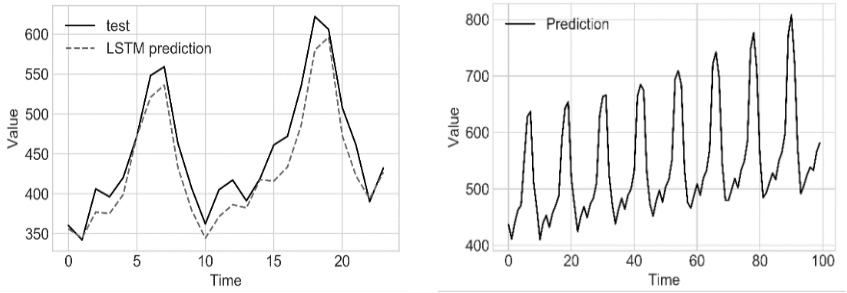


(a) (b)

Figure 7. Predicted values versus true values on the training data ($\sigma = 12$), for MM in (a) the LSTM model results and (b) the Bi-LSTM model results

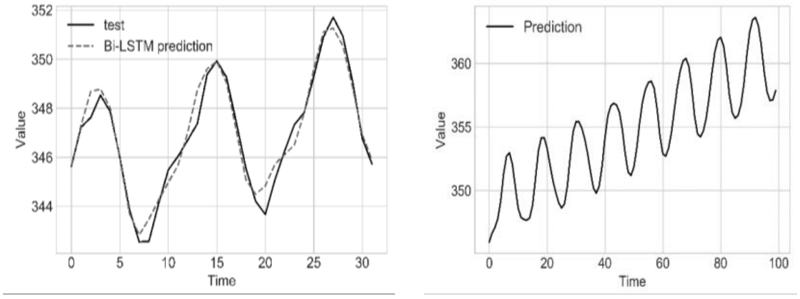
3.2. Real data results and discussion

We used the same parameters used previously as shown in Table 2 and the same RNNs on the two-time series, both additive and multiplicative, as described in subsection 3.3. We applied the low frequency filter (the moving average) to determine the period, we conduct tests by systematically changing the size of the input vector. The results of the system learning for the real data shows the potential of neural networks to model this class of time series by choosing appropriate parameters, in particular the size of the input vector, Figure 8(a) illustrates the performance of the LSTM models, the MAPE is of order 0.04, while Figure 8(b) shows the prediction of the two neural systems for 100 future observations. Moreover, Figure 9(a) shows the performance of the Bi-LSTM model, the MAPE is of order 0.05 while Figure 9(b) shows the prediction of the two neural systems for 100 future observations.



(a) (b)

Figure 8. LSTM model results for total passengers in an international airport data (a) predicted values versus true values on the training data and (b) prediction of 100 future observations



(a) (b)

Figure 9. Bi-LSTM model results for evolution of CO_2 concentration in the air data (a) predicted values versus true values on the training data and (b) prediction of 100 future observations

It is important to note that the learning performance depends on the size of the input vector *ve*, which corroborates the conclusion made for the simulated data, indeed, we did the learning by varying *ve*, for *ve=12* the system becomes efficient compared to other ve values. The first multiplicative signal of the monthly evolution of passengers has period *p=12* [5], for *ve* equal to 3, 6 and 9, MAPE is of the order, 12.35, 4.21 and 13.93 respectively, and this for LSTM model, the choice of *ve=12* allows an optimal performance of the order MAPE=0.04. The second additive signal has period *p=12* [28], for ve equal to 3, 6, 9 and 12, MAPE is of the order, 10.26, 6.74, 8.31, and 0.05 respectively, and this for Bi-LSTM model. The prediction of 100 future observations shows clearly that the system was able to learn the different features for the multiplicative and additive signals, such as the variation of seasonal component fluctuations over time.

Liu *et al.* study [19], adopts models such as, the convolutional neural network (CNN), FNN and a non-pooling CNN. Lui's study [19] also made a comparative study on the optimizer parameter using several types such as, Adadelta, AdaGrad, Adam, and SGD as well as the activation function namely, ReLU, Tanh, linear. They concluded that the choice of system parameters impacts learning performance, by indicating that choosing ReLU or linear activation functions and the Adam optimizer increases performance. Concerning this paper, we focused the research on other parameters of the system, specifically, the size of the input vector using RNN models.

To evaluate the performance of the neural network system, we made a comparison with the autoregressive moving average (ARMA) model, ARMA requires additional preprocessing to make both time series stationary. We created a 12-lag difference to remove the seasonality and then a 2-lag difference to remove the trends. The final model for the total passengers in an international airport data is ARMA with p=12 and q=1, which were selected by the autocorrelation function, the partial autocorrelation function and the Bayesian information criterion (BIC). For this model, the MAPE measure is of order 1.39. We find that the LSTM has obtained much better MAPE value than the ARMA.

4. CONCLUSION

Analyzing and modelling time series allows the extraction of knowledge. In the present study, we have introduced the modelling of seasonal time series with a trend via a supervised learning technique, in particular the RNN method. For this, we have established both LSTM and Bi-LSTM models in order to propose an approach to construct neural systems allowing relatively efficient modelling. We conducted tests on real and simulated time series, and we simulated the additive and multiplicative classes, in order to test the ability of the established systems to detect the change in the fluctuation of the seasonal component over time. Based on 80% of the data, the two RNN extensions were able to predict the rest of the series, which was then validated with the remaining 20%. Tests are performed by varying the period p and σ^2 (the variance of the noise component), and we noted a significant correlation between the input vector size v_e and the period p. Indeed, to ensure relatively efficient learning, we recommend choosing the input vector size v_e equal to the signal period p. We have also concluded that noise has an impact on learning performance, as the increase of MAPE error depends on the noise component.

REFERENCES

[1] B. T. Khoa and T. T. Huynh, "Forecasting stock price movement direction by machine learning algorithm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 6, pp. 6625–6634, 2022, doi: 10.11591/ijece.v12i6.pp6625-6634.

[2] S. Bhanja and A. Das, "A hybrid deep learning model for air quality time series prediction," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 22, no. 3, pp. 1611–1618, Jun. 2021, doi: 10.11591/ijeecs.v22.i3.pp1611-1618.

[3] P. Mondal, L. Shit, and S. Goswami, "Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices," *International Journal of Computer Science, Engineering and Applications*, vol. 4, no. 2, pp. 13–29, Apr. 2014, doi: 10.5121/ijcsea.2014.4202.

[4] J. Kitchin, "Cycles and trends in economic factors," *The Review of Economics and Statistics*, vol. 5, no. 1, Jan. 1923, doi: 10.2307/1927031.

[5] S. Ray, S. S. Das, P. Mishra, and A. M. G. Al Khatib, "Time series SARIMA modelling and forecasting of monthly rainfall and temperature in the South Asian countries," *Earth Systems and Environment*, vol. 5, no. 3, pp. 531–546, Sep. 2021, doi: 10.1007/s41748-021-00205-w.

[6] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, Nov. 2003, doi: 10.1061/(ASCE)0733-947X(2003)129:6(664).

[7] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003, doi: 10.1016/S0925-2312(01)00702-0.

[8] T. Zaiyong, C. de Almeida, and P. A. Fishwick, "Time series forecasting using neural networks vs. Box-Jenkins methodology," *Simulation*, vol. 57, no. 5, pp. 303–310, Nov. 1991, doi: 10.1177/003754979105700508.

[9] S. Ho, M. Xie, and T. N. Goh, "A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction," *Computers and Industrial Engineering*, vol. 42, no. 2–4, pp. 371–375, 2002, doi: 10.1016/S0360-8352(02)00036-0.

[10] J. Idrais, Y. El Moudene, R. El Abassi, and A. Sabour, "The use of online social networks: comparing Moroccan and French communities on Facebook," in *Artificial Intelligence and Smart Environment*, 2023, pp. 890–895.

[11] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *European Journal of Operational Research*, vol. 160, no. 2, pp. 501–514, Jan. 2005, doi: 10.1016/j.ejor.2003.08.037.

[12] P. Almasinejad *et al.*, "Predicting the status of COVID-19 active cases using a neural network time series," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, pp. 3104–3117, 2022, doi: 10.11591/ijece.v12i3.pp3104-3117.

[13] S. Karasu and A. Altan, "Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization," *Energy*, vol. 242, Mar. 2022, doi: 10.1016/j.energy.2021.122964.

[14] V. Fernandez, J. Chavez, and G. Kemper, "Device to evaluate cleanliness of fiber optic connectors using image processing and neural networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 3093–3105, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3093-3105.

[15] J. Idrais, R. El Abassi, Y. El Moudene, and A. Sabour, "Characterizing community behavior in OSNs: Modeling and forecasting activity on Facebook using the SARIMA model," *Journal of Intelligent and Fuzzy Systems*, vol. 43, no. 3, pp. 3757–3769, Jul. 2022, doi: 10.3233/JIFS-213391.

[16] A. Kumar Dubey, A. Kumar, V. García-Díaz, A. Kumar Sharma, and K. Kanhaiya, "Study and analysis of SARIMA and LSTM in forecasting time series data," *Sustainable Energy Technologies and Assessments*, vol. 47, Oct. 2021, doi: 10.1016/j.seta.2021.101474.

[17] R. El Abassi, J. Idrais, Y. El Moudene, and A. Sabour, "Multivariate time series forecasting using recurrent neural network for a complex system," in *Artificial Intelligence and Smart Environment*, Springer International Publishing, 2023, pp. 649–654.

[18] M. Qi and G. P. Zhang, "An investigation of model selection criteria for neural network time series forecasting," *European Journal of Operational Research*, vol. 132, no. 3, pp. 666–680, Aug. 2001, doi: 10.1016/S0377-2217(00)00171-5.

[19] S. Liu, H. Ji, and M. C. Wang, "Nonpooling convolutional neural network forecasting for seasonal time series with trends," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 2879–2888, Aug. 2020, doi: 10.1109/TNNLS.2019.2934110.

[20] T. Guo, Z. Xu, X. Yao, H. Chen, K. Aberer, and K. Funaya, "Robust online time series prediction with recurrent neural networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2016, pp. 816–825, doi: 10.1109/DSAA.2016.92.

[21] D. A. Bashar, "Survey on evolving deep learning neural network architectures," *Journal of Artificial Intelligence and Capsule Networks*, no. 2, pp. 73–82, Dec. 2019, doi: 10.36548/jaicn.2019.2.003.

[22] K. Yang, Y. Liu, Y. Yao, S. Fan, and A. Mosleh, "Operational time-series data modeling via LSTM network integrating principal component analysis based on human experience," *Journal of Manufacturing Systems*, vol. 61, pp. 746–756, Oct. 2021, doi: 10.1016/j.jmsy.2020.11.020.

[23] C. Paoli, C. Voyant, M. Muselli, and M.-L. Nivet, "Forecasting of preprocessed daily solar radiation time series using neural networks," *Solar Energy*, vol. 84, no. 12, pp. 2146–2160, Dec. 2010, doi: 10.1016/j.solener.2010.08.011.

[24] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Computers and Industrial Engineering*, vol. 143, May 2020, doi: 10.1016/j.cie.2020.106435.

[25] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.

[26] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.

[27] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, Jul. 2005, doi: 10.1016/j.neunet.2005.06.042.

[28] Y. Zou, R. V Donner, N. Marwan, J. F. Donges, and J. Kurths, "Complex network approaches to nonlinear time series analysis," *Physics Reports*, vol. 787, pp. 1–97, Jan. 2019, doi: 10.1016/j.physrep.2018.10.005.

[29] S. Dabhi and M. Parmar, "Eigenvector component calculation speedup over NumPy for high-performance computing," *arXiv preprint arXiv:2002.04989*, Feb. 2020.

[30] M. Waskom, "Seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, Apr. 2021, doi: 10.21105/joss.03021.

[31] J. Dignam, P. Martin, B. Shastry, and R. Roeder, "TensorFlow: a system for large-scale machine learning," in *USENIX Symp. Oper. Syst. Des. Implement*, 2016, vol. 101, pp. 582–598.

[32] J. Moolayil, "Deep neural networks for supervised learning: regression," in *Learn Keras for Deep Neural Networks*, Berkeley, CA: Apress, 2019, pp. 53–99.

BIOGRAPHIES OF AUTHORS

Rida El Abassi he received the master's degree in Applied Mathematical Sciences in Engineering Sciences at the University Ibn Zohr, Ph.D. student at the University Ibn Zoher Agadir Morocco, Mathematical and Computer Engineering Laboratory (IMI). His research interests include the applications of the extraction of knowledge on big data and artificial intelligence. He can be contacted at email: rida.elabassi@edu.uiz.ac.ma.

Jaafar Idrais is a computer science engineer with a bachelor's degree in distributed computing systems. He obtained his Ph.D. in computer science from the University Ibn Zohr Agadir in 2022. His research focuses on online social network analysis and knowledge extraction from virtual communities. He is a researcher at the University Ibn Zohr Agadir. Mathematical Engineering and Computer Science Laboratory (IMI Lab) under the Faculty of Science of Agadir. He can be contacted at email: jaafar.idrais@edu.uiz.ac.ma.

Abderrahim Sabour received his Ph.D. degree in Computer Science from the University Mohammed VAgdal in 2007. His research interests include artificial intelligence and digital sociology. He is presently working as a Researcher Professor in the Department of Computer Science, Higher School of Technology, Ibn Zohr University, Agadir, Morocco. He can be contacted at email: ab.sabour@uiz.ac.ma.