

Forecasting financial budget time series: ARIMA random walk vs LSTM neural network

Maryem Rhanoui¹, Siham Yousfi², Mounia Mikram³, Hajar Merizak⁴

¹IMS Team, ADMIR Laboratory, Rabat IT Center, ENSIAS, Mohammed V University in Rabat, Morocco

²Meridian Team, LYRICA Laboratory, School of Information Sciences, Rabat, Morocco

³SIP Research Team, Rabat IT Center, EMI, Mohammed V University in Rabat, Morocco

⁴LRIT, Faculty of Science, Mohammed V University in Rabat, Morocco

Article Info

Article history:

Received Aug 8, 2019

Revised Oct 22, 2019

Accepted Nov 1, 2019

Keywords:

ARIMA,
Deep learning,
Financial time series
LSTM,
Machine learning,
Random walk,
RNN

ABSTRACT

Financial time series are volatile, non-stationary and non-linear data that are affected by external economic factors. There is several performant predictive approaches such as univariate ARIMA model and more recently Recurrent Neural Network. The accurate forecasting of budget data is a strategic and challenging task for an optimal management of resources, it requires the use of the most accurate model. We propose a predictive approach that uses and compares the Machine Learning ARIMA model and Deep Learning Recurrent LSTM model. The application and the comparative analysis show that the LSTM model outperforms the ARIMA model, mainly thanks to the LSTMs ability to learn non-linear relationship from data.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Maryem Rhanoui,
IMS Team, ADMIR Laboratory,
Rabat IT Center, ENSIAS,
Mohammed V University in Rabat, Morocco.
Email: mrhanoui@gmail.com, sihamyousfi@research.emi.ac.ma, mikram.mounia@gmail.com

1. INTRODUCTION

Budget decision-making for an organization suffers from a lack of conceptual foundation, given the amount of information that is difficult to control. To remedy this problem and to control and monitor the consumption of the allocated budget, any organization sees itself in the need to rethink its budget by completing its information system by establishing a decision-making platform that allows uninitiated users to exploit and process stored data to identify opportunities that create a real competitive advantage and eliminate or mitigate risks. Forecasting financial data is different from typical deep learning applications, such as image recognition, as it does not consist on replicating tasks that humans can easily do [1]. Financial time series are usually non-stationary and non-linear [2].

Machine learning and deep learning are the marriage of massive data, analytical methods and statistics that apply and assist decision making, by synthesizing knowledge and recorded data history. Applying deep learning methods to these problems can produce more useful results than standard methods in finance. The autoregressive integrated moving average (ARIMA) model, and its subclass model, the Random Walk, is a safe bet and is commonly used in this context. However, the ARIMA model can hardly identify the nonlinear patterns, LSTM Neural Networks are of the most advanced deep learning architectures that learns from sequential and time-series data.

In this paper, we propose an approach that uses and compares two predictive models: a linear machine learning model and a nonlinear deep learning model, in order to develop an understanding of the

information available on the organization's own budget and predict its evolution. The added value of predictive models is their qualitative analysis that the organization can use to assist decision making. The remainder of this paper is organized as follow: in Section II we present the background knowledge, section III presents some related works, sections IV and V presents the forecasting models using ARIMA (0,1,0) Random Walk and LSTM Neural Network, finally, section VI synthesizes and compares both models.

2. BACKGROUND AND CONTEXT

2.1. ARIMA time series

Time Series are used to examine observations over time, with the goal of predicting future values. For example, predict the budget based on data from previous years. Time series are the fact of observing in a regular interval of time $[t_2 - t_1 = t_3 - t_2]$ a variable indexed by time $\{X_t, t_i \in T\}$, such that $T = \{t_1, \dots, t_n\}$ is the space of time. Time Series focuses on a single variable that is observed in different periods. A time series is the resultant of different components namely:

- Trend: evolution of the series in the long term.
- regular time interval.
- Residual (noise): irregular variation in a time interval.

2.1.1. ARIMA

ARIMA stands for Autoregressive (AR) Integrated (I) Moving Average (MA), also known as the Box-Jenkins approach. An ARIMA model is specified by the 3 parameters (p, d, q), such as:

- p is the number of autoregressive terms [AR (p)]
- d is the number of differentiation [I (d)]
- q is the number of moving averages [MA (q)]

A sequence $\{X_t, t_i \in T\}$ is called ARIMA process of order (p, d, q) ARIMA (p, d, q) if it can be written in the following formula:

2.1.2. Box-jenkins

Figure 1 shows the Box-Jenkins method [3] summarizes the ARIMA process in three main steps:

- Identification: This first step is to break down the time series according to the three processes: AR (autoregressive), I (integrated) and MA (moving average). This step obviously makes it possible to specify the parameters p, d and q, while first checking the stationarity of the series. Specification of the parameters p, q is done thanks to the autocorrelation functions and the partial autocorrelation which we will discuss in detail in the realization part. The parameter d is the order of differentiation.
- Estimation: The second step of the Box-Jenkins procedure is to estimate the parameters of the appropriate models by providing the orders p, d and q. the estimation is done using non-linear methods.
- Diagnosis: The last step of the Box-Jenkins method concerns the verification of the relevance of the model. That is, to verify that the estimated model is adapted to the data available. To do this we refer to statistical tests.

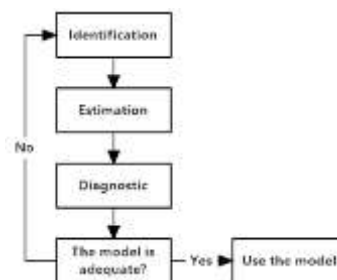


Figure 1. Box jenkins

2.1.3. Random walk

Random walk are stochastic processes formed by successive summation of independent, identically distributed random variables [4]. In the arima models, random walk corresponds to the ARIMA model (0,1,0).

2.2. Deep learning

Deep Learning [5] is a subfield of machine learning inspired by the structure and function of the brain. Deep Learning is a specific approach, less than 5 years old, used to build and form neural networks, which are considered very promising decision nodes. An algorithm is considered deep if the input data is passed through a series of non-linear transformations before they are output.

2.2.1. Neural network

An artificial neural network is inspired by the functioning of biological neurons. Written in the form of an algorithm, the neural network can modify itself according to the results of its actions, which allows it to learn and solve problems without human intervention. A neural network consists of three parts, an input layer, hidden layers, and an output layer. The input layers are a series of neurons containing the input signal that will be transmitted to the hidden layers, these layers represent the heart of the neural network it is at this level where the relations between the different variables are highlighted. The end result, often a prediction result, is at the output layer.

2.2.2. Recurrent neural network (RNN)

A recurrent neural network operates from sequential data, and learns from the succession of previous states. Each output depends on the calculation done downstream. In principle, RNNs can learn to map one variable sequence to another. RNNs are equivalent to very deep neural networks that share model parameters and receive input at each time step. An RNN is essentially characterized by the fact that it contains at least one return connection so that the activations circulate in loops. Recursion at the hidden layer of RNNs can act as a memory mechanism for networks (because the output at time t is a function of all previous inputs). At each time step, the learned recursion weights can decide which information to forget and which ones to keep in order to relay them over time. Among the main problems of an RNN, unjustified amplification of weights and the model being unable to learn training data. This problem is known by the "Exploding Gradients". The second problem with simple RNNs is that they do not preserve the information for a long time, so at some point the neural network can no longer connect the relationships between the data and as a result it would have difficulties to learn long-term additions. This problem is known as the "Vanishing gradient problem". To overcome exploding/vanishing gradient problems, a new concept has been introduced: "LSTM" abbreviation of Long Short-Term Memory.

2.2.3. LSTM

This concept was first introduced [6], it is an extension of recurrent neural networks to extend their memory. LSTMs allow RNNs to remember their entries over a long period of time, as an LSTM can write and delete information from its memory [7]. Figure 2 shows this memory behaves like a blocked cell ie the cell decides to store or delete information, depending on the importance it attributes to it. The attribution of importance is done through weights, which are also learned by the algorithm. It simply means that it learns over time what information is important and which is not. It is a gate mechanism and memory cell.

- Forget Gate: This block is responsible for resetting the memory cell (state cell). That is, the previously given information is no longer useful for the LSTM to learn more.
- Input Gate: This block takes the responsibility to add the information to the memory cell.
- Output Gate: This block is responsible for selecting useful information from the current memory cell.

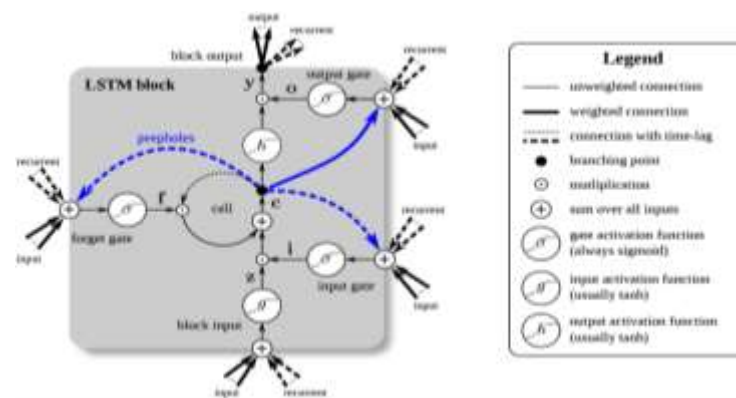


Figure 2. LSTM block [8]

3. RELATED WORKS

ARIMA time series are a widely used technique in econometrics for financial time series [9], several ARIMA model were proposed to analyze and forecast stock markets [10-12]. Furthermore, ARIMA is used for water budget/consumptions prediction [13-14] and electricity demand [15]. More recently, there has been a growing interest in the use of deep learning models [16-17], especially recurrent models such as LSTM Neural Network for the prediction of financial time series, in particular in the stock market [18-20]. In [18] proposed a modeling and prediction of China stock returns using LSTM architecture with an approved accuracy of 27.2%, in [19] analyzed the applicability of recurrent neural networks for stocks market prices movements prediction. Finally [20] proposed an accurate prediction of Shanghai Composite Index and Dow Jones Index. Concerning the budget analysis and forecasting, very little work were found comparing or even applying the two techniques, most of them has been applied to the stock price as previously mentioned. This is mainly due to the difficulty of obtaining relevant datasets, and the volatile nature of these data. None of its authors compared the performance of LSTM and ARIMA models.

4. FORECASTING USING ARIMA RANDOM WALK

4.1. Data sets

We use a Dataset that treats the actual budget consumed by a governmental organization. The data contained in this Dataset dates from 1976 to 2016 with an annual periodicity Figure 3 The values in the Dataset are expressed in billion dollars. The richness and the history of the data allow us to optimize the relevance of our analysis. The purpose of this analysis is to predict the budget for the upcoming years. Given that time series treat a single time-dependent variable that will predict future values based on previously observed values. The Dataset records and processes the annual budget. That said, the “Times series” model is well adapted to this case. In order to apply the ARIMA model, we follow the Box- Jenkins method.

Year	Budget		
2007-01-01	2.863068		
2008-01-01	3.326275		
2009-01-01	4.077475	mean	1.918888550732
2010-01-01	3.404600	std	1.113732387288
2011-01-01	3.509081	min	0.428870310000
2012-01-01	3.576168	25%	1.028252168000
2013-01-01	3.470366	50%	1.588911888000
2014-01-01	3.619008	75%	2.788388888000
2015-01-01	3.772713	max	4.077475000000
2016-01-01	3.880913		

Figure 3. Data description

4.2. Preprocessing

This phase is necessary as it allows the preparation of data and make them in accordance with our needs. We want to track and predict the evolution of the overall budget of the organization. This Dataset breaks down the budget by offices and services for each year, so we will consolidate the budget consumed for each year only. To build an ARIMA (Time series) model, it is desirable to store the time (for our case years) in one column and the variable on which we will apply the model in another column.

4.3. Analysis

In this third phase of the process, we analyze the behavior of the time series, in order to extract the useful information used to build the model Figure 4. We notice through this graph that the series is growing, so the budget is growing over time. There was a decline between the years 2010-2015 that intersects with the period of the economic crisis. The average of the series tends to change. The graph shows that the series is not stationary, to ensure the stationarity of the series, Figure 5 we refer to the test “Augmented Dickey-Fuller test”. This test is based on two assumptions:

- The null hypothesis: the series can be represented by a unit root, so it is not stationary.
- The alternative hypothesis: reject the null hypothesis, suggests that the series has no unit root, which means that it is stationary.

We interpret the test result using the p value generated by the test:

- $p\text{-value} > 0.05$: Failed to reject the null hypothesis, the data has a unit root and is non-stationary.
- $p\text{-value} < 0.05$: Reject the null hypothesis, the data have no unit root and are stationary.

Applying this test for our time series, we obtain the following result. The value p is greater than 0.05 so this time series is nonstationary.

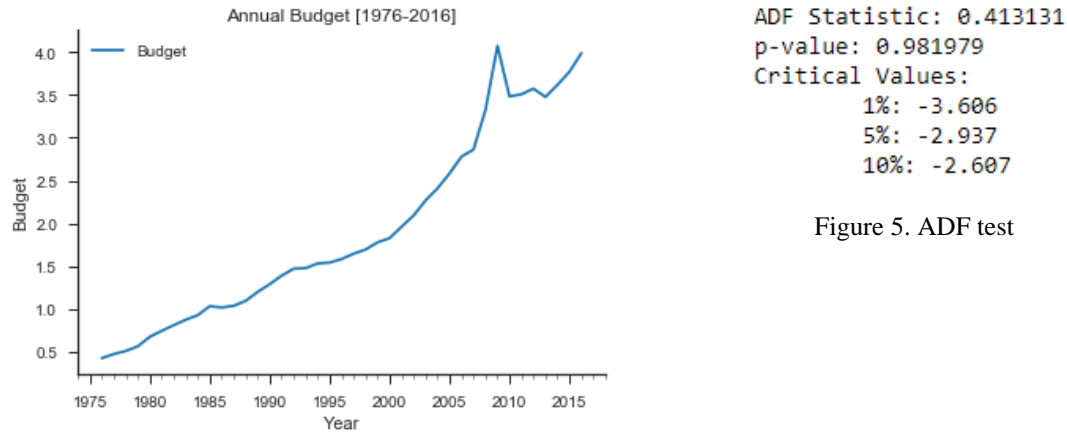


Figure 4. Plot of the series

The autocorrelation Figure 6 shows that the peaks at each offset break the confidence interval ± 1.96 , so the series is not a white noise, that is, there is no temporal dependency. Thus, the peaks of each ACF offset decrease very slowly, which means that the terms of the series are correlated over several periods in the past. After having analyzed the series we will begin the crucial phase which allows to elaborate the adequate ARIMA model.

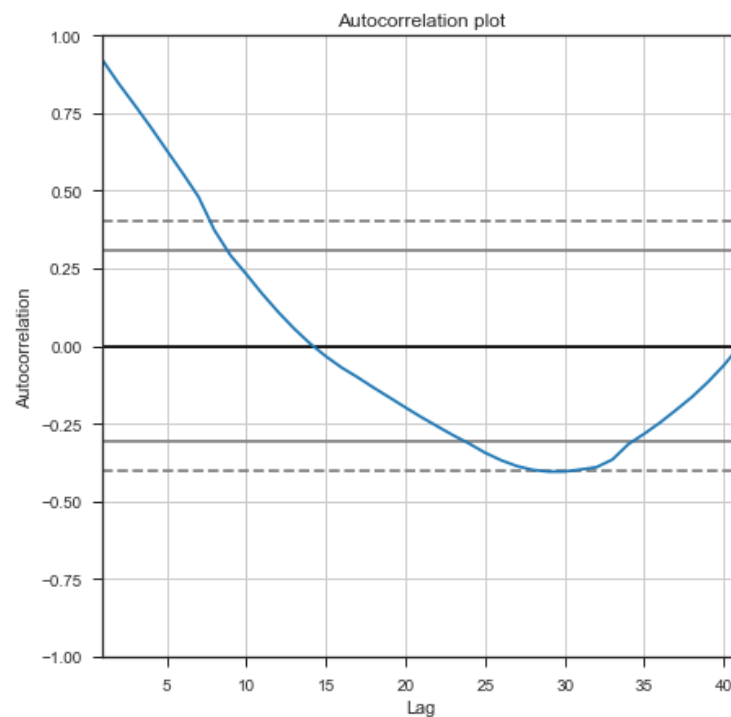


Figure 6. Autocorrelation graph of the time series

4.4. Model Selection and Construction

The first step of the Box-Jenkins method that we follow to build our ARIMA model concerns the identification of parameters. The ARIMA process is applied on a stationary series, but our series is not. To make the series stationary, we differentiate it a first time and then apply the “ADF” test to check the stationarity of the resulting series. After that the series is stationary, the next step is to determine the parameters p and q of the process. To do this, we will plot the autocorrelation graph and the partial autocorrelation of the differentiated series. The plot of the ACF and PACF Figure 7 shows that the series defines a random walk because only the first peak breaks the confidence interval. That said, the parameters p and q are equal to zero.

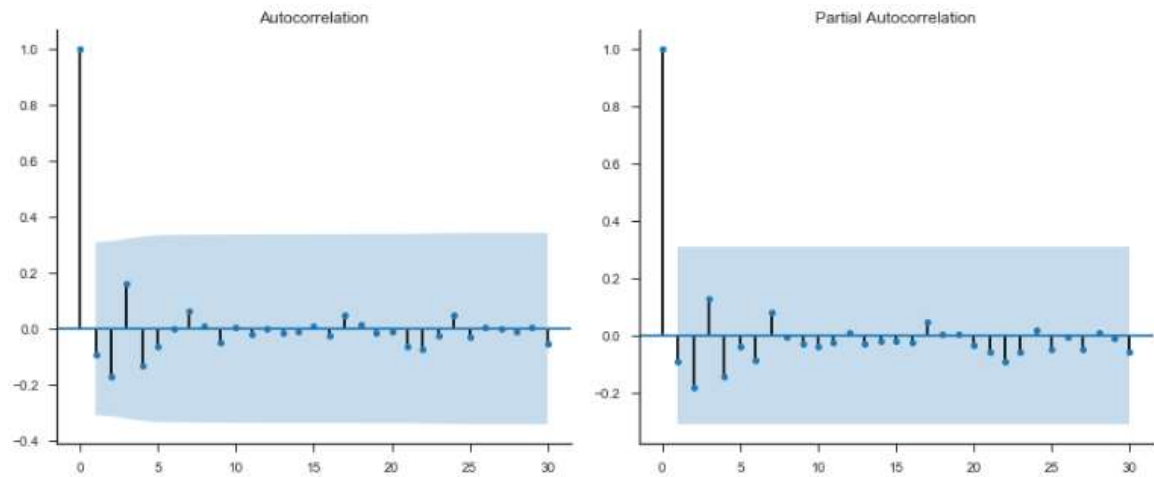


Figure 7. Plot of the ACF and PACF

After identifying the parameters of the ARIMA model (0,1,0) we apply this model to our data. The first step in applying a machine learning model is to divide the data into two subsets:

- Training
- Test

We then apply the ARIMA model (0,1,0) to our training data and predict the test data to evaluate the accuracy of the model. The following Figure 8 compares the test data set (expected values) with the values generated by the ARIMA model (0,1,0). We note that the predicted values follow the evolution of the test values with a small margin of error. We need to evaluate the performance of this model. The Box-Jenkins method gives recommendations for determining the parameters p , d and q , but this is not necessarily the best model for the time series studied. To judge the relevance of our model, we will test ARIMA (0,1,1) and ARIMA (1,1,0), then we will compare the AIC to determine the best model. The best model is the one with the lowest AIC, so ARIMA (0,1,0) is the most suitable for our case.

Table 1. Comparison of the aic of the different models

ARIMA Model	AIC
ARIMA (0,1,0)	-22.100
ARIMA (1,1,0)	-20.493
ARIMA (0,1,1)	-20.682

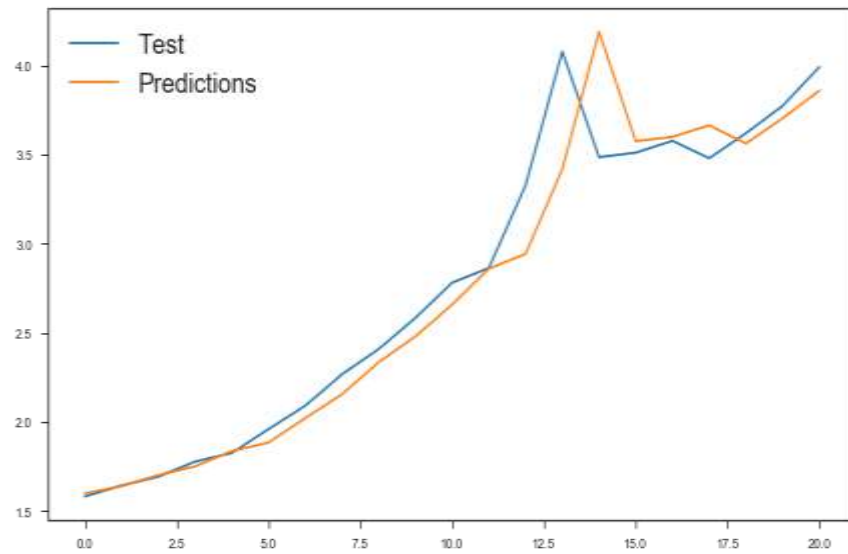


Figure 8. Real and estimated values

4.5. Diagnosis

The 3rd step of the Box-Jenkins process is the diagnosis of the model. To judge the precision of the model we will study the distribution of residual. Figure 9 suggest a Gaussian type distribution. The plot of density shows a slight shift toward zero. The autocorrelogram of the residuals shows that no autocorrelation coefficient is significantly different from zero. Given the above results, we can validate the ARIMA random walk model (0,1,0) that we proposed.

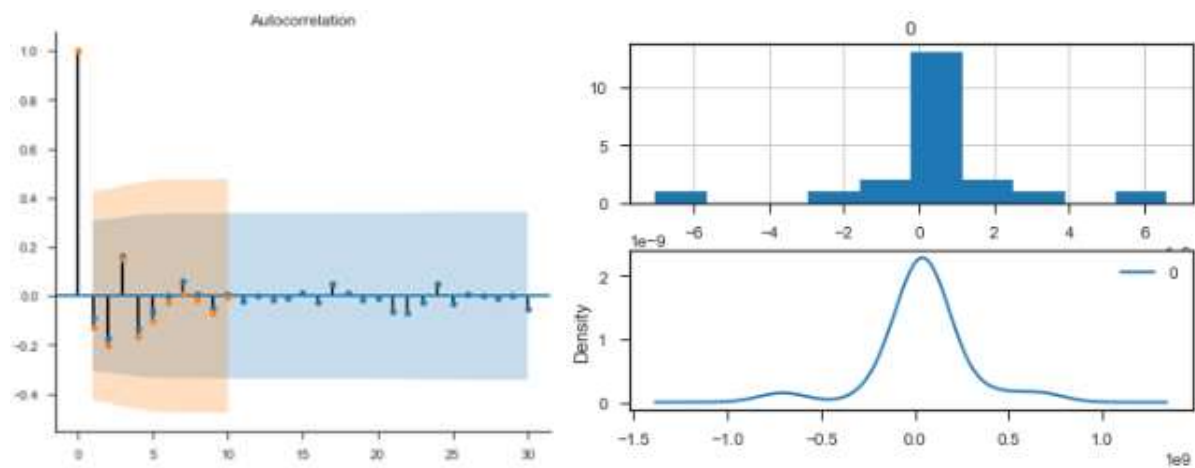


Figure 9. Diagnosis of the ARIMA model

4.6. Model application and prediction

The purpose of the work is the prediction of the budget for the coming years in order to have a clear and concise idea. For this, we apply the constructed model to predict the evolution of the budget for the next seven years. Figure 10 shows the predicted values for the next seven years. In the Figure 11, we will graphically show the evolution of the budget in the next years based on the history provided.

2016-12-31	3.858657681979
2017-12-31	3.944602363957
2020-12-31	4.030547045936
2019-12-31	4.116491727915
2020-12-31	4.202436409893

Figure 10. Prediction of the budget for the next seven years

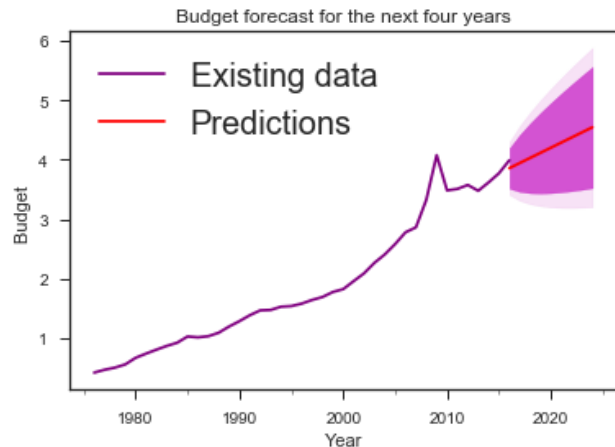


Figure 11. Prediction of the budget for the next years

Forecasts and the associated confidence interval that we generated are used to better understand time series and predict what to expect. Forecasts show that the budget should continue to grow at a steady pace. As long as we are planning the budget for years to come, it is natural for us to become less confident in our values. This is reflected in the confidence intervals generated by our model, which grow as we move further into the future.

5. FORECASTING USING LSTM

In this section we want to predict the budget for years to come using Deep Learning via the LSTM architecture. Note that we previously predicted the budget using the ARIMA model. We use Tensorflow and Keras libraries to implement this architecture. Budget data is saved as sequences. To manage the dependence of the sequences we use the recurrent neural networks, precisely an LSTM since it preserves the information for a long duration and allows to model the most sophisticated dependencies in our time series. This model supports a very large volume of data. An LSTM has three parameters: one parameter to write the information in the memory, the other to read it and the last one to delete it.

In this phase we discuss the architecture of the proposed LSTM model:

- The second step is to standardize the data, ie the data must belong to the scope of the activation function.
- Choice of the activation function. For this study we opted for a sigmoid function that outputs values between 0 and 1. We use the hyperbolic tangent function.
- “Batch size” is the number of samples that will be propagated in the neural network.
- The neural network requires only one output to estimate the budget for the next year.
- After building the model, it is important to thoroughly evaluate the model. To do this, we use the RMSE cost function. This function calculates the error between the predicted data and the test data.

5.1. Model construction

In this step we build an appropriate LSTM model for our case study. An LSTM model assumes that our data is divided into input X and output Y components. For our case, we use the previous observations for each time step as our inputs and the output will be the observation of the current time step. The following Figure 12 represents the overall configuration of our LSTM.

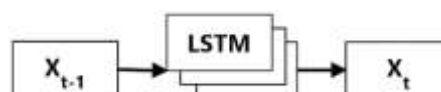


Figure 12. Global diagram of the LSTM

To build an LSTM model we must first transform the training and test Dataset into a three-dimensional array of “samples, features and timesteps”. We use an input layer, a hidden layer containing LSTM blocks and a single output layer. We made use of the default activation function of an LSTM: sigmoid. The model is trained 1000 times with batch size equals to 10. We use ADM optimization algorithm for updating weights. We used the following rules to determine the number of layers and the number of neurons in each layer:

- Input layer: logically we have one entry, the number of neurons contained in this layer is determined by the number of quantifiable columns. For our case we want to follow the evolution of the budget, so the number of neurons in the input layer is 1.
- Output layer: each neural network has a single output. Since we want to control the budget, our output layer contains only one neuron.
- Hidden layer: the size of this layer, that is to say the number of neurons, is to determine.

We tested various LSTM configurations using different numbers of blocks. Table 2 shows compares the different configurations based on the RMSE. It reveals that in the best configuration is the one with 2 LSTM blocks. Figure 13 describes the adopted LSTM architecture, it illustrates the flow of an X time series through an LSTM layer. Note that Y represents the output and c represents the memory. The first LSTM unit takes the initial state of the network and the first-time step of the sequence X1 and then calculates the first output Y1 and the memory c1. At time t, the unit takes the current state of the network (c_{t-1}, Y_{t-1}) and the next time step of the sequence X_t , then calculates the output X_t and the memory c_t . Each LSTM unit behaves like a mini-memory where the forget gate, input gate and output gate have weights that are learned during the training procedure.

Table 2. LSTM configurations

Model	RMSE	MSE	MAE
1 Block LSTM	0.226	0.051	0.120
2 Block LSTM	0.222	0.049	0.119
3 Block LSTM	0.281	0.079	0.178
4 Block LSTM	0.229	0.052	0.122
5 Block LSTM	0.284	0.081	0.153
10 Block LSTM	0.296	0.087	0.157
20 Block LSTM	0.298	0.089	0.158
50 Block LSTM	0.264	0.070	0.149
100 Block LSTM	0.261	0.068	0.147
300 Block LSTM	0.257	0.066	0.144
500 Block LSTM	0.257	0.066	0.144

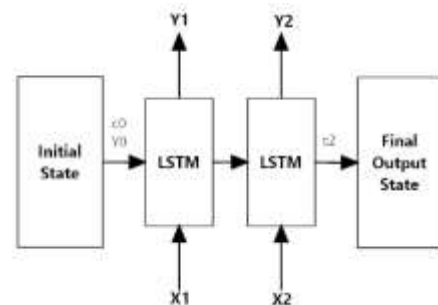


Figure 13. LSTM Architecture

6. MODELS SYNTHESIS AND COMPARAISON

In order to predict the evolution of the budget we followed two methods: ARIMA and LSTM. In this section we evaluate the two models developed to predict the evolution of the budget. This study compares the performance of two techniques for predicting financial time series. The goal being the prediction of the budget for the years to come. Below, we visualize Figure 14 the predictions of the two models. After making the stationary series we applied the ARIMA model using different settings, the best model retained was the random walk ARIMA (0,1,0). Then we developed an LSTM architecture based on different parameter settings, the best configuration was two LSTM blocks contained in the hidden layer. To evaluate the models, we used the RMSE calculation, this measure allows to calculate the difference of the residues between the predicted values and the values recorded in the data set. The Table 3 shows compares the two models that we established based on the RMSE. It is recalled that the statistical quantity RMSE (the root of the squared mean error) is a measure widely used to evaluate the accuracy of the model and calculate the difference between the actual values of the Dataset, and the value predicted by a model.

We notice a slight difference between the two RMSEs, such as the RMSE of the LSTM model is smaller. So it is obvious that the LSTM architecture outperforms ARIMA's performance. In order to validate the choice of the predictive model, and to estimate to what extent the choice is precise we will compare the MAE and MSE of the two models. MAE as its name suggests is the average of absolute errors. The absolute error is the absolute value of the difference between the expected value and the actual value. The Table 4 shows compares the MAE of both models. We note that the MAE of the LSTM model is smaller, which confirms its good performance in predicting more accurate values for the studied time series. MSE is the average of the squares of the difference between the actual observations and those predicted. This

calculation allows us to evaluate the accuracy of a model. The Table 5 shows the LSTM model is once again better than the ARIMA model.

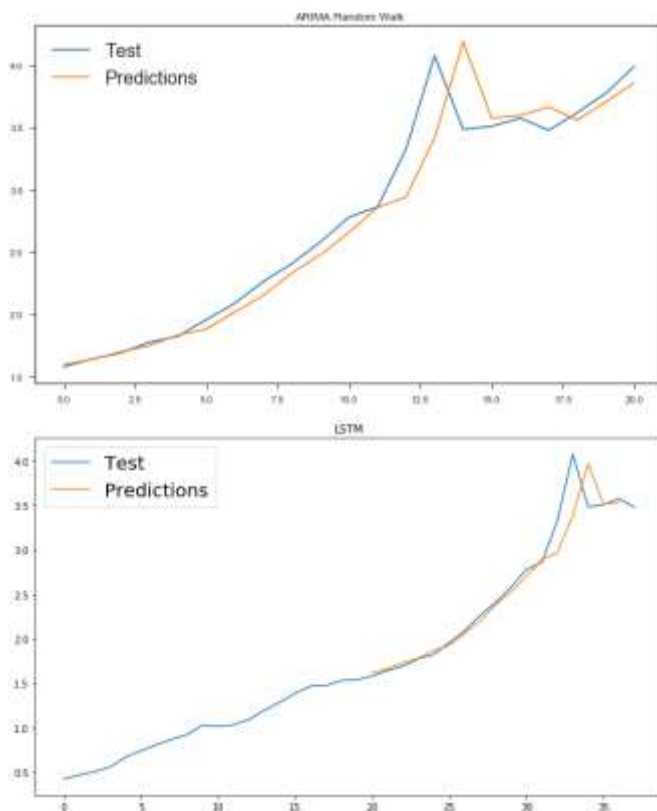


Figure 14. Visualization of the predictions of the two models

Table 3. RMSE comparison

Model	RMSE
Machine Learning: ARIMA	0.239
Deep Learning: LSTM	0.222

Table 4. MAE comparison

Model	MAE
Machine Learning: ARIMA	0.139
Deep Learning: LSTM	0.119

Table 5. MSE comparison

Model	MSE
Machine Learning: ARIMA	0.057
Deep Learning: LSTM	0.049

6.1. Major differences between ARIMA and LSTM

Although the research is recent, it is clear that LSTM architectures have great potential as candidates for time series modeling and forecasting. We study in the following Table 6 the major differences between an LSTM and ARIMA. The use of RNNs including the LSTM architecture, allows the setting of several parameters that we must adjust to obtain optimal performance on the forecasting tasks. Its difficulty lies in choosing the right parameters to find the right model architecture. An ARIMA model is simple to configure as it gives a good performance, this model also requires the identification of the parameters p , d and q such that p is the order of the autoregressive part (AR), of the order of differentiation and q the order of the moving average part (MA).

Table 6. Models comparison

ARIMA	LSTM
Linear model	Nonlinear model
Small amount of data	Large amount of data
Parametric model, that is to say for each series we have to define the parameters p , d and q	Non-parametric model, requires adjustment of some hyperparameters
Dedicated specifically for time series	Process sequential data

7. CONCLUSION

Defining an optimal model to forecast financial time series data is a challenging task because of the non-linearity, non-stationarity and volatility characteristics of this type of data. In this paper we compared two forecasting models for financial time series. This predictive analysis showed that, although the ARIMA model provides satisfactory results, the LSTM model outperforms the performance of the ARIMA model.

Deep Learning techniques, and the LSTM recurrent neural network in particular, can identify non-linear structures in financial time series. In future work, we investigate the application of Bidirectional recurrent neural networks for Random Walk time series [21] and extend the comparison with multivariate ARIMA [22]. Also, although it is not as widely used as RNN models for financial prediction, Convolutional Neural Networks (CNN) remains a promising approach [23] to be exploited for the prediction of financial times series.

REFERENCES

- [1] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3-12, 2017.
- [2] M. Sewell, "Characterization of Financial Time Series," *RN*, vol. 11, no. 01, pp. 01-01, 2011.
- [3] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [4] G. F. Lawler and V. Limic, *Random walk: a modern introduction*. Cambridge University Press, 2010.
- [5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85-117, 2015.
- [6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [7] A. Graves, "Long Short-Term Memory," ed: Springer Berlin Heidelberg, 2012, pp. 37-45.
- [8] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, 2017.
- [9] T. C. Mills and R. N. Markellos, *The econometric modelling of financial time series*. Cambridge University Press, 2008.
- [10] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497-505, 2005.
- [11] J. E. Jarrett and E. Kyper, "ARIMA modeling with intervention to forecast and analyze chinese stock prices," *International Journal of Engineering Business Management*, vol. 3, no. 3, pp. 53-58, 2011.
- [12] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on*, 2014 2014, pp. 106-112.
- [13] M. Birylo, Z. Rzepecka, J. Kuczynska-Siehien, and J. Nastula, "Analysis of water budget prediction accuracy using ARIMA models," *Water Science and Technology: Water Supply*, vol. 18, no. 3, pp. 819-830, 2018.
- [14] S. Razali, M. Rusiman, N. Zawawi, and N. Arbin, "Forecasting of water consumptions expenditure using Holt-Winter's and ARIMA," in *Journal of Physics: Conference Series*, 2018, vol. 995, no. 1: IOP Publishing, p. 012041.
- [15] K. Kandanand, "Electricity demand forecasting in buildings based on ARIMA and ARX models," in *Proceedings of the 8th International Conference on Informatics, Environment, Energy and Applications*, 2019: ACM, pp. 268-271.
- [16] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187-205, 2017.
- [17] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18569-18584, 2017.
- [18] K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," in *2015 IEEE International Conference on Big Data (Big Data)*, 2015/10 2015, pp. 2823-2824.
- [19] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017/05 2017, pp. 1419-1426.
- [20] Q. Jiang, C. Tang, C. Chen, X. Wang, and Q. Huang, "Stock Price Forecast Based on LSTM Neural Network," in *Proceedings of the Twelfth International Conference on Management Science and Engineering Management*, J. Xu, F. L. Cooke, M. Gen, and S. E. Ahmed, Eds., 2019: Springer International Publishing, pp. 393-408.
- [21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, 1997.
- [22] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural networks*, vol. 5, no. 6, pp. 961-970, 1992.
- [23] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.