



ORIGINAL

Forecasting the EUR/USD Exchange Rate Using ARIMA and Machine Learning Models

Previsión del tipo de cambio EUR/USD utilizando ARIMA y modelos de aprendizaje automático

Said LAKHAL¹  

¹Polydisciplinary Faculty, Computer Science. Larache, Morocco.

Cite as: LAKHAL S. Forecasting the EUR/USD Exchange Rate Using ARIMA and Machine Learning Models. Data and Metadata. 2024; 3:.368.
<https://doi.org/10.56294/dm2024.368>

Submitted: 26-01-2024

Revised: 18-04-2024

Accepted: 26-08-2024

Published: 27-08-2024

Editor: Adrián Alejandro Vitón Castillo 

Corresponding Author: Said LAKHAL 

ABSTRACT

The present paper compared ARIMA with two machine learning algorithms, for forecasting USD/EUR exchange rate data. The experimental results indicated that the performance of ARIMA fell between that of recurrent neural networks and long short-term memory machine learning algorithms.

Keywords: Propose Autoregressive Integrated Moving Average; Long Short-Term Memory; Recurrent Neural Network; Forecasting Data.

RESUMEN

El presente artículo comparó ARIMA con dos algoritmos de aprendizaje automático para pronosticar datos del tipo de cambio USD/EUR. Los resultados experimentales indicaron que el rendimiento de ARIMA se situaba entre el de las redes neuronales recurrentes y el de los algoritmos de aprendizaje automático con memoria a corto plazo.

Palabras clave: Media Móvil Integrada Autorregresiva; Memoria Larga a Corto Plazo; Red Neuronal Recurrente; Pronóstico de Datos.

INTRODUCTION

Research context

In time series analysis, a stationary time series (STS) is one whose statistical properties, such as mean, variance, and autocorrelation, remain constant over time. In other words, an STS exhibits consistent statistical behavior over time, making forecasting feasible.⁽¹⁾

Generally, an STS may take one of four different models: autoregression (AR), moving average (MA), autoregression moving average (ARMA), or autoregression integration moving average (ARIMA).

The AR model implies a pure correlation between the time series values and their preceding and succeeding values. Such as, an AR(p) model uses p past data points to predict future behavior.⁽²⁾ Unlike AR, the MA process specifies that the output variable is cross-correlated with a non-identical, random variable.⁽³⁾ When the number of used random variables is q, the time series will be denoted by MA(q).

Sometimes, the two models AR and MA are combined to model the time series; in this case, an ARMA(p,q) model will be referred.⁽⁴⁾ However, in many cases, the available time series is not stationary, requiring transformation to establish stationary properties.

Among the famous transformation techniques used, there is an integration process. This method involves iteratively subtracting successive terms until stationarity is obtained. The order of integration, or the number of iterations until stationarity, is denoted by d , and the time series is indicated by $I(d)$.⁽⁵⁾

Once the integration process is finished, a suitable model for the time series will be searched: AR, MA, ARMA, or ARIMA. As a result, the obtained model is ARIMA (p, d, q), where p , d , and q represent the AR, integration, and MA components, respectively.

Frequently, time series analysis is employed to predict future data, facilitating decision-making in forex trading.

Based on principles and the time of inception, trading methods can be categorized into two classes: traditional and modern.

The former employs two types of analysis technical, where historical data is considered, and fundamental, where macroeconomic parameters play a crucial role.

The latter relies on algorithms to make decisions. Within this context, a distinction is drawn between trading algorithms and machine learning algorithms.

Trading algorithms are static, initially crafted by programmers to analyze data and make decisions without learning capabilities. In contrast, machine learning algorithms possess the ability to learn from the environment and make decisions based on changing conditions.⁽⁶⁾

The machine learning algorithms include decision tree regression, random forest regression, K-nearest neighbors, support vector machine (SVM)^(7,8,9) recurrent neural networks (RNN)^(10,11,12) and Long Short-Term Memory (LSTM).^(13,14,15)

Objectives of this study

The objective of this study is to compare the forecasting performances of the ARIMA model with LSTM and RNN models.

To achieve this objective, the paper is structured as follows:

- First, a literature review explores the use of the ARIMA model and the two machine learning algorithms in trading.
- Second, the used methods are exposed, including: data collection and analysis.
- Third, the development of model is made.
- Fourth, the results section illustrated the forecasting data, by using the three models; as well as the comparison between models in terms of: the shape of the curves, the mean absolute error (MAE), and the mean square error (MSE) relative to the observed data.
- Fifth, a discussion section is developed to describe the obtained results.
- Finally, a conclusion summarized the contributions of the paper.

Literature review

Islam MS et al.⁽¹⁶⁾ have investigated the prediction of future closing prices for four currency pairs, employing a blend of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). This comparative analysis aims to gauge the efficiency of the hybrid GRU-LSTM model relative to standalone LSTM and GRU models, with a focus on minimizing Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

This exploration into predictive models sets the foundation for understanding their effectiveness in financial forecasting. Simultaneously, Xinwei C et al.⁽¹⁰⁾ has delved into portfolio optimization using a recurrent neural network model, showcasing superior performance compared to the Dow Jones Industrial Average index, characterized by higher investment returns and lower risks.

The demonstration of enhanced portfolio management strategies underscores the potential of machine learning techniques in financial decision-making. David A et al.⁽¹⁷⁾ the researchers have devised an ARMA-GARCH model fused with machine learning methods to detect intraday patterns in defense stock and FOREX markets.

Their findings offer initial insights into market predictability, particularly during geopolitical events, affirming the efficacy of the proposed model. This validation of predictive models highlights the importance of integrating machine learning with traditional financial analysis methods.

Jaimin S et al.⁽¹⁸⁾ has critically evaluated stock price prediction techniques, encompassing ARIMA, LSTM, Hybrid LSTM, CNN, and Hybrid CNN models. By scrutinizing their limitations, accuracy, and margin of errors. The study advocates for hybrid models as a means to achieve efficient stock market forecasting, transcending the limitations of individual approaches.

This comparative analysis emphasizes the necessity of leveraging diverse methodologies to enhance predictive accuracy in financial markets. Lastly, a comparative analysis between neural network and traditional techniques for ETF.⁽¹⁹⁾

In addition, this study explores various neural network architectures to address a specific problem, enriching the understanding of predictive methodologies in financial contexts. This investigation into predictive methodologies expands the repertoire of tools available for financial analysts and researchers.

METHOD

The data was collected based on the monthly USD/EUR exchange rate over six years. This data was analyzed by using the three models. The established model was used to forecast the future monthly exchange rate for one year. These results were compared to the observed data to evaluate the performance of the established models.

Data collection

The monthly USD/EUR exchange rate from January 2018 to December 2023 was collected from the Yahoo Finance website. The data from the first five months was used for the training phase, and the data from the last month was used for the test phase.

Data analysis

ARIMA analysis

The stationary of a time series can be examined through graphical representation (figure 1) as well as ADF and KPSS tests: The first allows the verification of the presence or absence of trend and/or seasonality. Contrarily, the second provides statistical indicators and thresholds for comparison to assess stationary.

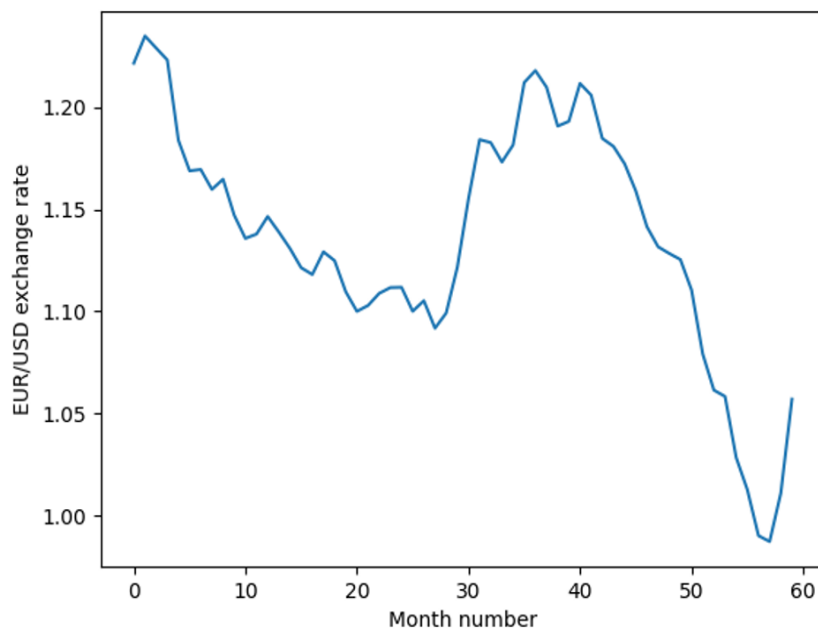


Figure 1. The EUR/USD exchange rate over time from January 2018 to December 2022

The absence of any trend and seasonality across the graph can be easily observed. This enhances the hypothesis of stationary. But the results of ADF and KPSS tests are necessary to confirm this finding.

After running the ADF and KPSS algorithms based on the time series values, they return critical values and correspondence with significance levels to validate two hypotheses, H0_ADF and H1_ADF, as well as H0_KPSS and H1_KPSS.

$$ADF: \begin{cases} \text{Null Hyp } (H_0): \text{The time series} \\ \text{has a unit root (non - stationary).} \\ \text{Alter Hyp } (H_1): \text{The time series} \\ \text{is stationary.} \end{cases} \quad (1)$$

$$KPSS: \begin{cases} \text{Null Hyp } (H_0): \text{The time series} \\ \text{trend - stationary.} \\ \text{Alter Hyp } (H_1): \text{The time series} \\ \text{not trend - stationary.} \end{cases} \quad (2)$$

The results of the two algorithms are illustrated in table 1.

Test	_stat	p_val	1 %	5 %	10 %	Result
ADF	0,036	0,036	-3,55	-2,91	-2,59	Stationary
KPS	0,08	0,08	0,739	0,46	0,34	Stationary

By considering a significance level of 5 %, in the first part, it is easy to observe that the ADF_Statistic and the p_value are both less than the critical Value and 0,05, respectively. Hence, the null hypothesis can be rejected, indicating that the series is stationary.

In the second part, the KPSS_Statistic is less than the critical value, and the p_value is more than 0,05. Therefore, the null hypothesis cannot be rejected, indicating that the series is trend stationary.

In summary, the ADF and KPSS tests confirm the time series stationary.

LSTM model

The LSTM is a type of deep recurrent neural network model. It remembers relevant past knowledge and forgets irrelevant data. This is ensured through the use of different activation function layers called gates. In this case, three gates are considered: the input gate, the output gate, and the forget gate, as shown in figure 2.

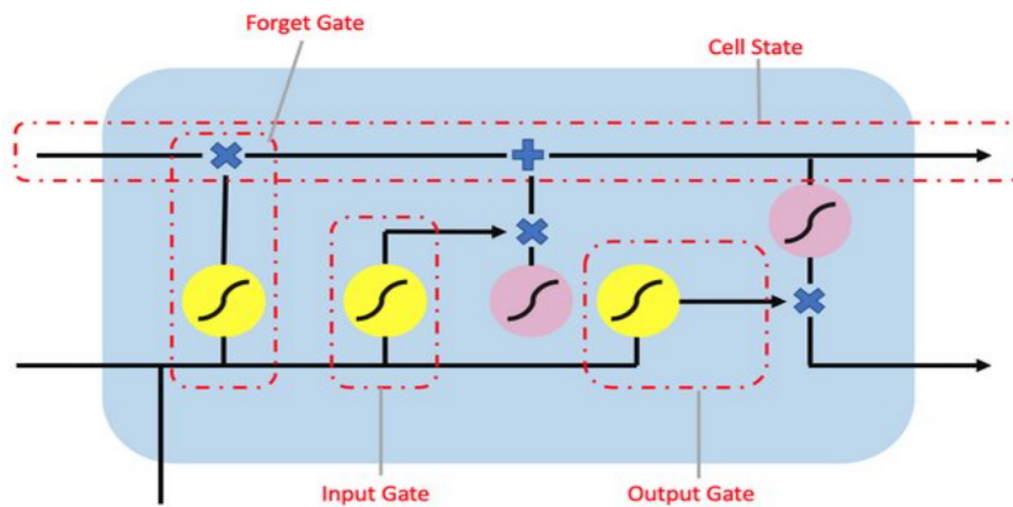


Figure 2. Architecture of LSTM⁽²⁰⁾

RNN model

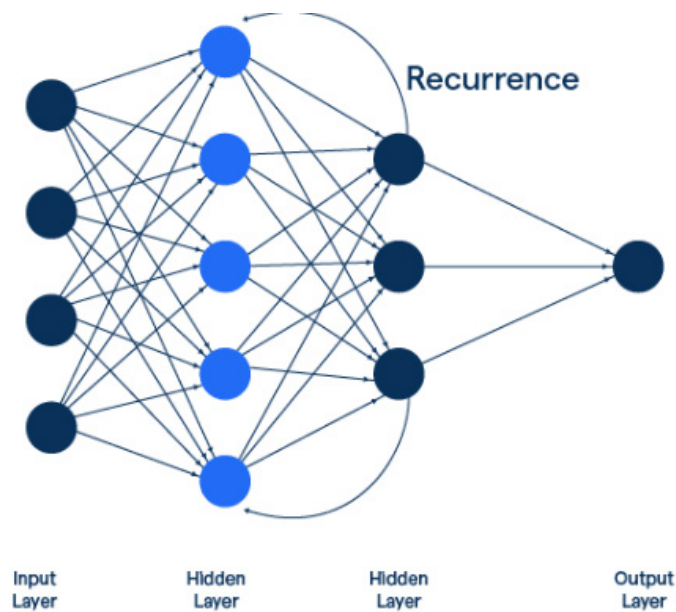


Figure 3. Architecture of RNN

The RNN addresses the limitations of traditional neural networks by introducing a recurrent connection that allows information to flow from one time step to the next. This way, an initial memory is maintained, where the output of each step is fed back as an input to the next step.

The RNN architecture enables it to capture information from previous steps and utilize it in the current step, allowing the model to learn temporal dependencies and handle input of variable length (figure 3).

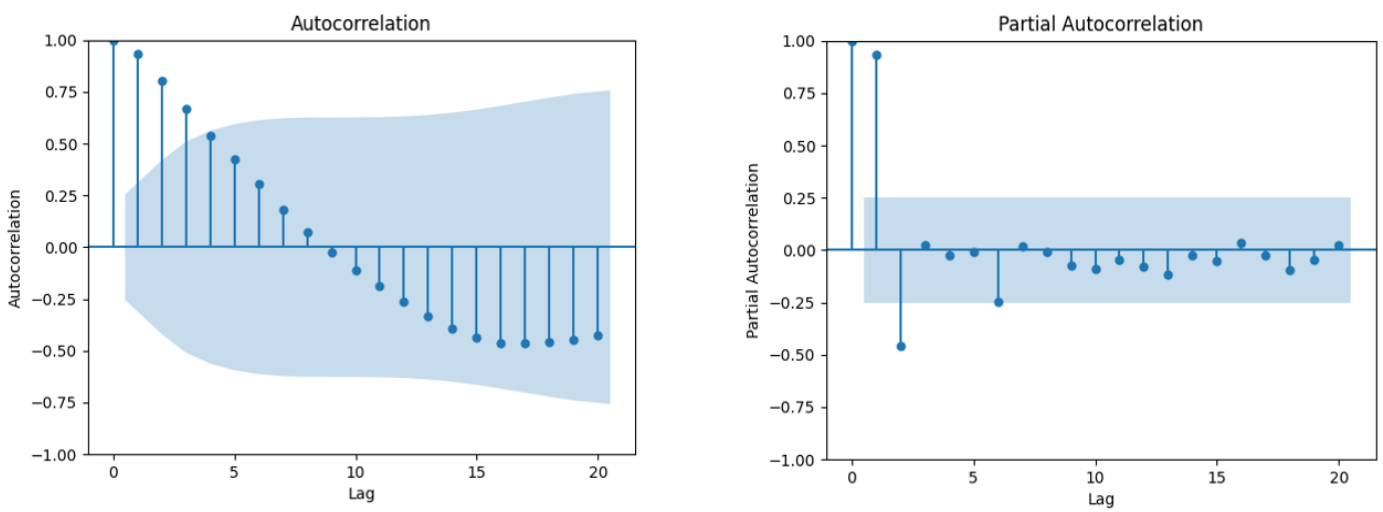
DEVELOPMENT

The ARIMA model is based on a set of parameters illustrated in table 2:

Table 2. The meaning of parameters used in the ARIMA model	
Parameters	Meaning
p	Number of autoregressive terms
d	Number of nonseasonal differences needed for stationarity
q	Number of lagged forecast errors in the prediction equation
T	Total number of observations in the time series
\bar{x}	Sample mean of x_T
\bar{x}_{t-1}	Sample mean of x_{T-1}
$\bar{\epsilon}_{t-1}$	Sample mean of ϵ_{t-1}
ϕ_1	AR coefficient
θ_1	MA coefficient
C	Constant
P'	Estimator of parameter P
MAE	Mean Absolute Error
MSE	Mean Squared Error
y_i	Rate at the i^{th} month

To specify an ARIMA model, there are three parameters to define: p , d , and q , such that the model is denoted as ARIMA(p,d,q) (table 1). These parameters correspond respectively to the auto-regression, integration and moving average components. While the time series is stationary, the value of parameter d will be 0.

Thus, the model becomes ARIMA ($p,0,q$), or simply ARMA(p,q). In following, the values of p and q will be determined. For this purpose, they are two methods: ACF and PACF analyses (figure 2), as well as AIC and BIC value calculations.



Autocorrelation Function (ACF)

Partial Autocorrelation Function (PACF)

Figure 4. ACF and PACF

Based on figure 4, a sharp drop has been observed after the first lag, indicating the possibility of an AR(1) model. In other side, a sharp drop has been remarked after the first lag in figure 2.b, indicating the probability of an MA(1) model. But, before deciding that the time series follows an ARMA(1,1) model, the results of AIC and BIC values are necessary: after varying the values of p and q between 1 and 10 and calculating the corresponding AIC and BIC values, the value (p,q)=(1,1), leads to the lowest AIC and BIC values. These outputs confirm the results obtained previously. Consequently, the chronological series follows an ARMA(1,1). The validation of the model has been done based on the comparison between the ACF of residuals and the ACF of white noise (figure 5). If they are similar, then ARMA(1,1) is a good approximation for the time series.

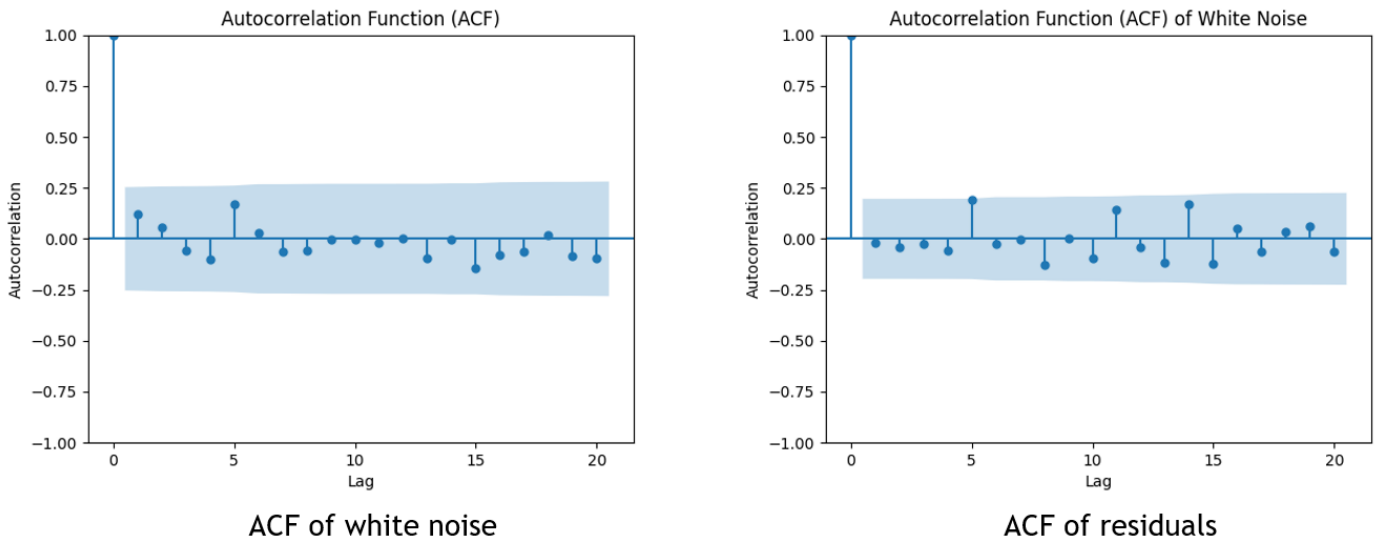


Figure 5. ACFs of residuals and white noise

Referring to figure 5, the complete similarity between the ACFs of residuals and white noise is evident. Therefore, ARMA(1,1) is a good approximation for this time series.

An ARMA(1,1) model is expressed as follows:

$$x_t = c + \phi_1 x_{t-1} + \epsilon_t + \theta_1 \epsilon_{t-1} \quad (3)$$

The specification of the model requires the clarification of three parameters: c, ϕ_1 and θ_1 . These parameters will be estimated based on the following expressions:

$$\hat{c} = \bar{x} - \hat{\phi}_1 \bar{x}_{t-1} - \hat{\theta}_1 \bar{\epsilon}_{t-1} \quad (4)$$

$$\hat{\phi}_1 = \frac{\sum_{t=2}^T (x_t - \hat{c} - \hat{\theta}_1 \bar{\epsilon}_{t-1})(x_{t-1} - \bar{x}_{t-1})}{\sum_{t=2}^T (x_{t-1} - \bar{x}_{t-1})^2} \quad (5)$$

$$\hat{\theta}_1 = \frac{\sum_{t=2}^T (x_t - \hat{c} - \hat{\phi}_1 x_{t-1} - \epsilon_t) \epsilon_{t-1}}{\sum_{t=2}^T \epsilon_{t-1}^2} \quad (6)$$

After solving the system formed by equations (4,5,6): the obtained values are: c=1,1389, $\phi_1 = 0,9382$ and $\theta_1 = 0,5544$.

Thus, the time series can be expressed as follows:

$$x_t = 1.1389 + 0.9382x_{t-1} + \epsilon_t + 0.5544\epsilon_{t-1} \quad (7)$$

The evaluation of forecasting performance is based on MAE and MSE, such as:

$$MAE = \frac{1}{12} \sum_{i=1}^{12} |y_i - \hat{y}_i| \quad (8)$$

$$MSE = \frac{1}{12} \sum_{i=1}^{12} (y_i - \hat{y}_i)^2 \quad (9)$$

RESULTS

The ARIMA model established in equation (7) has been used to forecast the months of the year 2023. The results are illustrated in figure 6.

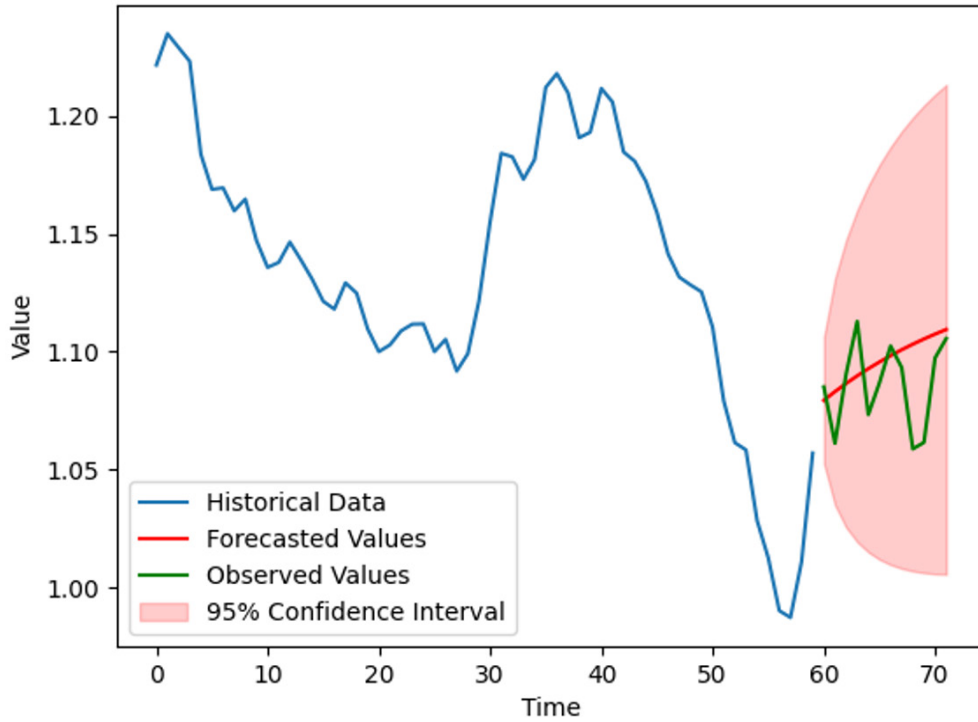


Figure 6. ARMA(1,1) forecasting

In order to make a shape comparison between the three models, figure 7 serves this purpose.

The graphical representation in figure 7 will be confirmed by a quantitative comparison, expressed by MAE and MSE. The numerical results are illustrated in table 3.

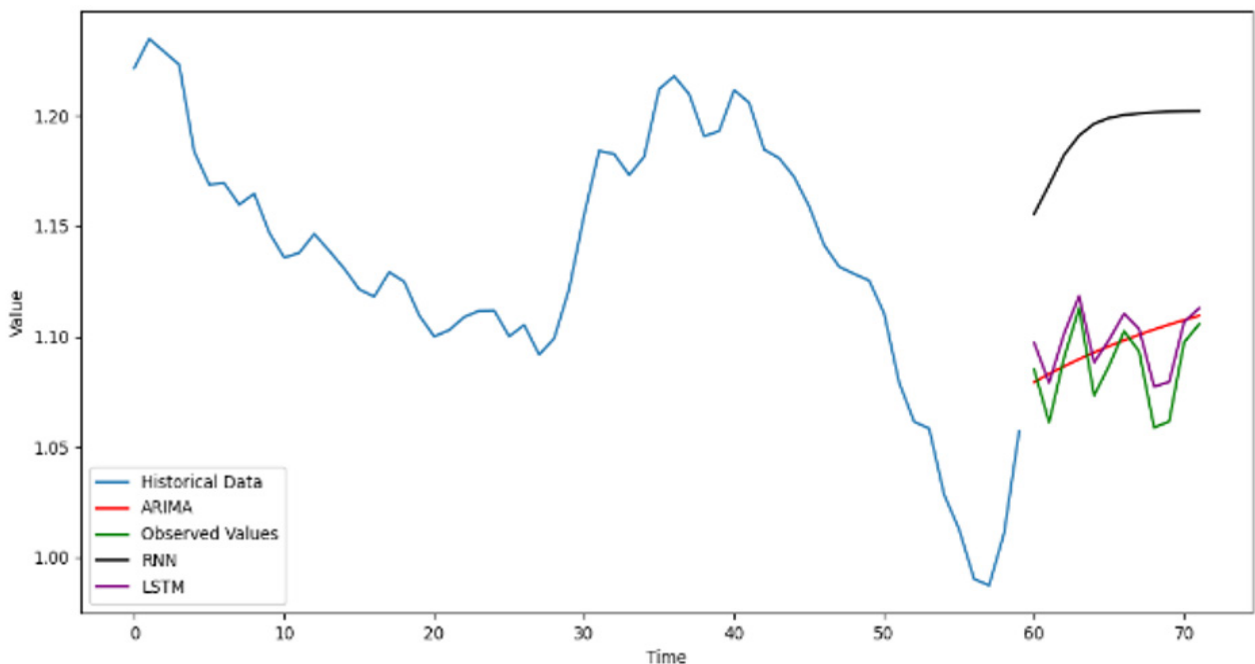


Figure 7. Forecasting by using the three models

Table 3. Numerical performance of the three models

	LSTM	RNN	ARMA
MAE	0,0179	0,0505	0,0164
MSE	0,00033	0,0029	0,00046

DISCUSSION

To describe the forecasted data, two approaches are followed: graphical representation and numerical calculation.

The graphical representation shows that the ARIMA forecasted data maintains a strictly increasing curve situated in the middle of the observed data. In contrast, the RNN forecasted data curve is positioned above the observed data with an increased slope. Finally, the LSTM forecasted data remains close to the observed data, following the same shape.

The error calculations reveal that the RNN has the highest values of MAE and MSE, while there is a minor difference between ARIMA and LSTM in terms of MAE. As well as, The LSTM's MSE is lower than that of ARIMA.

This comparison can be summarized in table 4.

Table 4. Comparison between the forecasted and the observed data by using the three models

	Curve shape	Curve position	MAE	MSE
ARIMA	Increased	Middle	Minimal	Average
LSTM	Similar	Near	Average	Minimal
RNN	Increased	Far	Maximal	Maximal

As a result, the LSTM model can be considered the most performant among the three models, given its curve shape and the minimal value of MSE.

CONCLUSIONS

In this work, the USD/EUR exchange rate data has been modeled using ARIMA, LSTM, and RNN models. After forecasting the data over a period of twelve months with these three models, the LSTM yields the best performance in terms of congruence with observed data and minimizing errors.

BIBLIOGRAPHIC REFERENCES

1. Peter JB, Richard AD. Introduction to Time Series and Forecasting. 2nd ed. Colorado: springer; 2002.
2. William WSW. Time series analysis, Univariate and multivariate methods. 2nd ed. New York: Addison-Wesley; 2006.
3. George P, Gwilym J, Gregorec R, Gretam L. Time Series Analysis Forecasting and Control. 4th ed. New Jersey: John Wiley & Sons; 2016.
4. Regina K, Agustín M. Notes on time series analysis ARIMA models and signal extraction. Banco de España, Servicio de Estudios Documento de Trabajo. 2023; 0012: 1-73.
5. Joseph LM, Kevin EB, Gemunu HG. Integration I(d) of Nonstationary Time Series Stationary and nonstationary increments. Texas: Center for Superconductivity University of Houston; 2008.
6. Ali S, Majid K. A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. Expert Systems With Applications. 2022;208(1):118-124.
7. Alireza S, Amir D, Mahdi MZ. Combined ensemble multi-class SVM and fuzzy NSGA-II for trend forecasting and trading in Forex markets. Expert Systems With Applications. 2021;185(10).
8. Jonatha SP, Raydonal O, Anderson A. A novel fusion Support Vector Machine integrating weak and sphere models for classification challenges with massive data. Decision Analytics Journal. 2024;11.

9. Chen L, Jiong RW. Jing Wan, Osman Taylan, Cheng-Wei Fei. Adaptive directed support vector machine method for the reliability evaluation of aeroengine structure. *Reliability Engineering and System Safety*. 2024;246.
10. Xinwei C, Adam F, Xujin P, Zenan Z, Vasilios K, Predrag S, Ivona B, Shuai L. A novel recurrent neural network based on line portfolio analysis for high frequency trading. *Expert Systems With Applications*. 2021;185.
11. Andrea C, Claudio G. Residual Echo State Networks: residual recurrent neural networks with stable dynamics and fast learning. *Neurocomputing*. 2024.
12. Mohammed SA, Yi MR, Feiyang O, Fahim A, Panagiotis DC. Model predictive control of non linear processes using transfer learning-based recurrent neural networks. *Chemical Engineering Research and Design*. 2024;205.
13. Chenjie S, Massimo DP. Improving trading technical analysis with TensorFlow Long Short Term Memory (LSTM) Neural Network. *The Journal of Finance and Data Science*. 2019; 5(1) :1-11.
14. Qian K, Dengxiu Y, Kang HC, Zhen W. Deterministic convergence analysis for regularized long short-term memory and its application to regression and multi-classification problems. *Engineering Applications of Artificial Intelligence*. 2024; 133(E).
15. Yildirim K, Sheldon BG, Hossein E, Dorcas SE, Farshad B, Yavuz CK, Aman A. Improving the accuracy of short-term multiphase production forecasts in unconventional tight oil reservoirs using contextual Bi-directional long short-term memory. *Geoenergy Science and Engineering*. 2024; 235.
16. Islam MS, Hossain E. Foreign exchange currency rate prediction using a GRU-LSTM hybrid network. *Soft Computing Letters*. 2021; 3.
17. David A, Belen SM, Antonio P. Hybrid ARMA-GARCH-Neural Networks for intraday strategy exploration in high-frequency trading. *Pattern Recognition*. 2024; 148.
18. Jaimin S, Darsh V, Manan S. A comprehensive review on multiple hybrid deep learning approaches for stock prediction. *Soft Computing Letters*. 2022; 16.
19. Karolina M, Agata K, Aleksandra M, Kateryna C, Klaudia K, Paweł G, Marcin H, Krzysztof L, Adrianna K, Marcin P, Artur R, Mykola D. Financial Time Series Forecasting: Comparison of Traditional and Spiking Neural Networks. *25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*. 2021; 192: 5023-5029.
20. Divish, Mina J , Benjamin R, Xin C, Graziela PF. Deep Learning with Dynamically Weighted Loss Function for Sensor-Based Prognostics and Health Management. *Sensors*. 2020; 20(723) :1-23.

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Said LAKHAL.

Investigation: Said LAKHAL.

Methodology: Said LAKHAL.

Drafting - original draft: Said LAKHAL.

Writing - proofreading and editing: Said LAKHAL.