Journal of Economics and Sustainable Development
Volume: 09 / N°: 01 (2026), P:134-150
ISSN : 2661-7986 EISSN: 2773-2606

Gold Price Prediction Based on ARIMA and LSTM Neural Network

Rouaba Mohammed [*1]

[1] Laboratory for the development of Algerian economic institutions, Ibn Khaldoun University Tiaret (Algeria), mohammed.rouaba@univ-tiaret.dz

Received: 02/09/2025 ***Accepted:13/01/2026*** ***Published:01/03/2026***

Abstract:
This study compares the forecasting performance of the Autoregressive Integrated Moving Average (ARIMA) model and the Long Short-Term Memory (LSTM) neural network in predicting daily closing prices of gold. Using a dataset covering the period from January 1, 2023, to May 25, 2025, both models were implemented in Python and evaluated using standard error metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).
The ARIMA model was constructed following differencing and optimal order selection based on the Akaike Information Criterion. The LSTM model employed a multi-layer architecture refined through experimental trials. Results indicate that the ARIMA model outperformed the LSTM network across all metrics, suggesting its greater suitability for modeling short-term gold price movements in this context.
Keywords: Gold price forecasting, ARIMA, LSTM, Time series, financial modeling.
JEL Classification: C53; G17; E37

[*] ***Corresponding author.***

Introduction

In the precious metals market -especially gold- researchers use a variety of methods to forecast price movements. Traditionally, the most common approaches have been technical analysis and fundamental analysis. Technical analysis relies on the basic laws of supply and demand to interpret market trends and anticipate future price shifts. Fundamental analysis, on the other hand, examines economic and financial factors that influence prices, aiming to identify underlying patterns that can guide investment decisions. While these methods differ in approach, both focus on predicting price trends and require a solid understanding of economics and investment theory.

With the ongoing advancement in computing power and the success of machine learning in various domains, predictive models based on machine learning have increasingly been applied to gold price forecasting. These models leverage historical data and deep learning techniques to detect patterns, simulate market behavior, and improve predictive accuracy over time. One of the key strengths of machine learning lies in its ability to identify technical indicators that influence price movements and to operate efficiently in large-scale, data-rich environments.

This study aims to answer the question:

"What is the optimal model between ARIMA and LSTM for predicting the daily closing prices of Gold?"

Literature review

Traditional time series models, particularly the Autoregressive Integrated Moving Average (ARIMA) framework, have served as foundational tools for forecasting gold prices due to their interpretability and robustness in capturing linear dependencies in historical data.

Siami- Namini et al. (2018) conducted one of the earliest comparative analyses between ARIMA and Long Short- Term Memory (LSTM) models in financial time series forecasting, demonstrating that LSTM architectures significantly reduced forecasting errors by 84–87 percent compared to ARIMA, underlining the potential of deep learning in this domain (Siami-Namini et al., 2018).

More recently, an empirical investigation using monthly global gold prices from 2019 to 2022 found that both ARIMA and LSTM models exhibit strengths in different phases of the time series, with LSTM providing edge in capturing nonlinear patterns, while ARIMA offered consistent performance during stable market periods(Madhika et al., 2023).

Setyowibowo et al. (2022) enhanced the classical ARIMA approach by integrating it with a Generalized Autoregressive Conditional

Heteroskedasticity (GARCH) model, achieving an RMSE of 2.375 for daily gold price forecasts, which highlights the utility of hybrid models in addressing volatility clustering (Setyowibowo et al., 2022).

LSTM networks have been applied extensively for their ability to learn long- term dependencies in sequential financial data. For example, the SHS Web of Conferences (2024) compared LSTM with linear regression for gold price prediction and reported a mean squared error (MSE) of 376.603 for LSTM, indicating its superior performance over linear baselines (Gong, 2024).

A hybrid study published in PPHM Journal (2025) evaluated ARIMA, LSTM, and their combination, demonstrating that the ARIMA- LSTM hybrid outperformed stand- alone models on multiple error metrics, thereby validating the complementary strengths of statistical and neural approaches (Mun et al., 2025).

Another comparative analysis by Siami Namin et al. (2019) extended the evaluation to Bidirectional LSTM (Bi-LSTM), finding that Bi-LSTM models converge more slowly but yield higher accuracy than both unidirectional LSTM and ARIMA in gold price predictions (Siami-Namini et al., 2019).

The primary scientific contributions of this work are threefold. First, it introduces a comparative framework that systematically evaluates the performance of ARIMA and LSTM models in the context of gold price forecasting, offering empirical insights into the strengths and weaknesses of each approach. Second, by leveraging machine learning techniques on high-frequency financial data, this study demonstrates how data-driven models can complement or even surpass traditional econometric approaches in forecasting precision. Third, the findings of this research have practical implications for investment decision-making and risk management, providing a methodological foundation for future studies aiming to integrate statistical and deep learning models in commodity price prediction.

1- Model Theoretical Basis

1-1- ARIMA Time Series Model

1-1-1- Introduction to ARIMA Model Theory

In the 1970s, statisticians Jenkins and Box proposed the ARIMA model, also known as the autoregressive integrated moving average model. As a nonlinear stochastic process mathematical model based on statistical principles, the ARIMA model is mainly used for short-term prediction of time series variables. Although individual time series values are unpredictable, the overall sequence exhibits regularity, which can be mathematically expressed through the ARIMA model, enabling short-term

prediction of time series values. The ARIMA model includes three types: AR(p) model, MA(q) model, and ARIMA(p,d,q) model. The ARIMA model usually requires various treatments to achieve stationarity for non-stationary time series, with differencing techniques being particularly suitable for short-term trend stationarity.

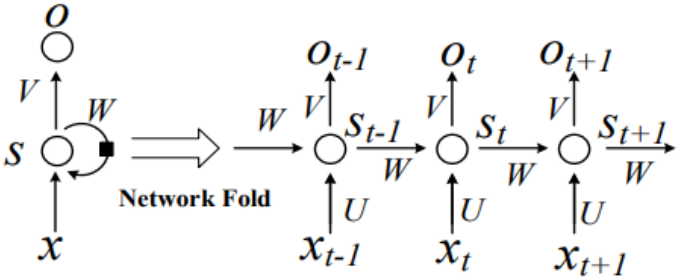
1-1-2- ARIMA Model Building Steps

The ARIMA model building process includes four steps: First, perform a stationarity test on the original series. If it is non-stationary, differencing is generally required to obtain stationary time series data. Second, determine the model and use the AIC criterion to determine the order. Third, estimate the model parameters using the least squares method and verify their rationality. Fourth, after diagnostic analysis, use the generated model to predict data and compare it with actual data to verify its accuracy. If the parameters cannot be precisely determined, a new model needs to be constructed.

1-2- Long Short-Term Memory Neural Network

Since neural networks can better fit nonlinear time series data like stock prices, BackPropagation (BP) neural networks were proposed in 1986 by Rumelhart and McClelland. BP neural networks are multi-layer feedforward neural networks trained using the error backpropagation algorithm. The structure of BP neural networks is characterized by full connectivity between neurons in adjacent layers, but no connections within the same layer, making it difficult to utilize information from previous layers when processing time series. This led to the development of RNNs, whose neural network structure is shown in Figure 1.

Figure number (01): Structural Diagram of the RNN Neural Network



Source: (Gao et al., 2018, p. 3)

Where x_{t-1}, x_t and x_{t+1} are the inputs at times $t-1$, t, and $t+1$, respectively; s_{t-1}, s_t, and s_{t+1} are the intermediate results at times $t-1$, t, and $t+1$, respectively; and o_{t-1}, o_t, and o_{t+1} are the final outputs at times

$t-1$, t, and $t+1$, respectively. W, U, and V are the parameters to be trained. The output at time t, o_t, is:

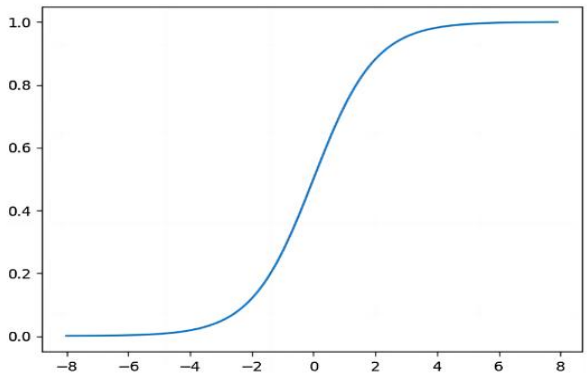
$$O_t = g(V \cdot S_t) \quad (1)$$

Where s_t is calculated as:

$$S_t = f(U \cdot x_t + W \cdot S_{t-1}) \quad (2)$$

The activation function used in the model is the Sigmoid function, as shown in Figure 2. When the input value reaches a certain level, the output changes little, leading to a decrease in the error gradient. The essence of backpropagation is to continuously adjust the fitting function by reducing the error to update the weights. Over time, the gradient decreases, and some layers stop learning when the gradient approaches zero. These layers usually stop learning early, causing the RNN to forget information from earlier layers. Additionally, RNNs face the problem of gradient explosion due to increasingly complex calculations, limiting their prediction capabilities.

Figure number (02): Sigmoid function

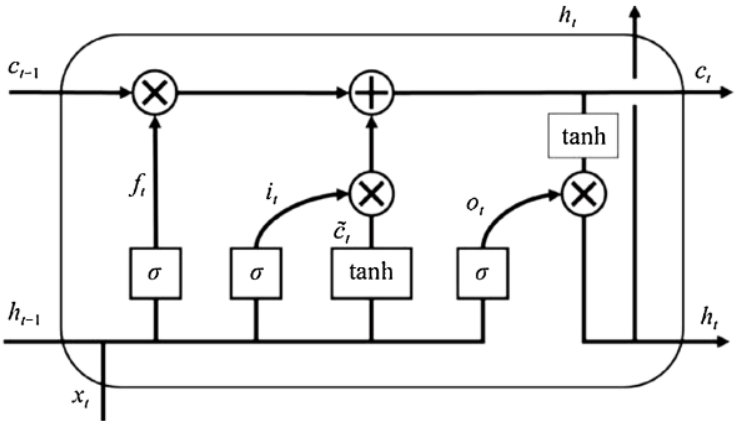


Source: (Ngah et al., p. 4883)

Hochreiter and Schmidhuber proposed the Long Short-Term Memory (LSTM) neural network structure (Hochreiter & Schmidhuber, 1997) to address the shortcomings of RNNs. LSTM is a special type of RNN designed to solve issues like gradient explosion, making time series prediction more accurate. The core of LSTM lies in its cell state and gate structure, which includes the forget gate, input gate, and output gate, acting as the "memory" of the neural network. The forget gate controls the retention of the previous memory unit by passing the previous hidden state and current input information through the Sigmoid function. The input gate controls the retention of new candidate cell states. The input gate determines the extent to which the current cell state (Xiao & Yin, 2019, p. 5) controls

the current output. The neuron structure of the LSTM neural network is shown in Figure 3, where f_t is the forget gate at time t, i_t is the input gate at time t, o_t is the output gate at time t, c_t is the cell state (long-term memory) at time t, h_t is the memory body (short-term memory) at time t, $\tilde{c}_t$ is the candidate state at time t, and σ represents the Sigmoid function.

Figure number (03): Neuronal Structure Diagram of the LSTM Neural Network



Source: (Xiao & Yin, 2019, p. 5)

The output of LSTM at time t, h_t, is:

$$h_t = o_t \cdot \tanh(c_{t-1}) \quad (3)$$

Where:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (4)$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (5)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (6)$$
$$\tilde{c} = \tanh(W_c x_t + U_c h_{t-1}) \quad (7)$$
$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (8)$$

In the above equations, W, U, and b are the weight matrices and bias vectors that need to be trained.

2- Empirical Analysis

2-1- Data Source

The data used in this study were obtained through the Yahoo Finance API using Python, covering daily trading information for gold futures from January 1, 2023, to May 25, 2025. The dataset includes the opening price, closing price, highest and lowest prices, and trading volume. For the purpose of this empirical analysis, only the daily closing prices were used, as they are considered to best reflect market consensus at the end of each

trading session. The selected data serve as the basis for training and evaluating both the ARIMA and LSTM models. A sample of the dataset is presented in Table 1.

Table number (01): Daily Trading Data of Gold Futures

Price	Close	High	Low	Open	Volume
Date					
2025-05-19	3228.9	3241.0	3228.3	3234.4	266
2025-05-21	3309.3	3317.5	3290.2	3293.4	979
2025-05-22	3292.3	3328.0	3282.7	3327.3	1210
2025-05-23	3357.7	3366.5	3285.5	3295.1	238139

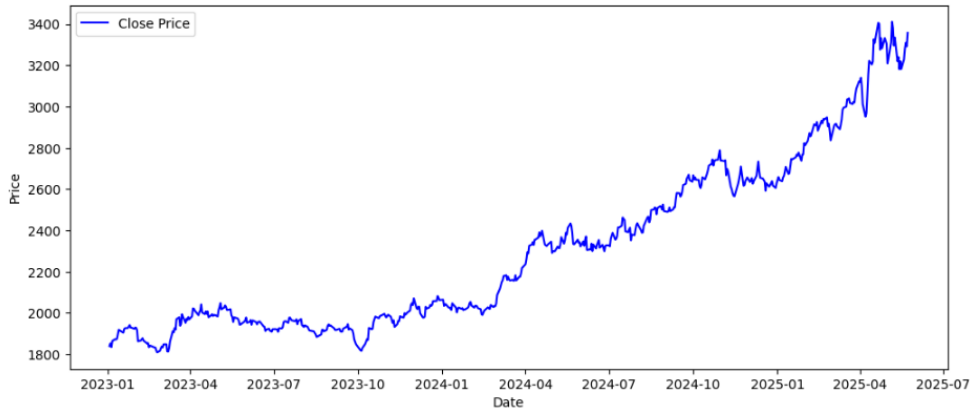
Source: Prepared by the researcher based on Python outputs.

2-2- ARIMA Model Establishment and Analysis

2-2-1- ADF Test and Data Stationarity Processing

Using Python, the trend of gold's daily closing price was plotted, as shown in Figure 4. The horizontal axis represents the date, while the vertical axis indicates the closing price of gold futures.

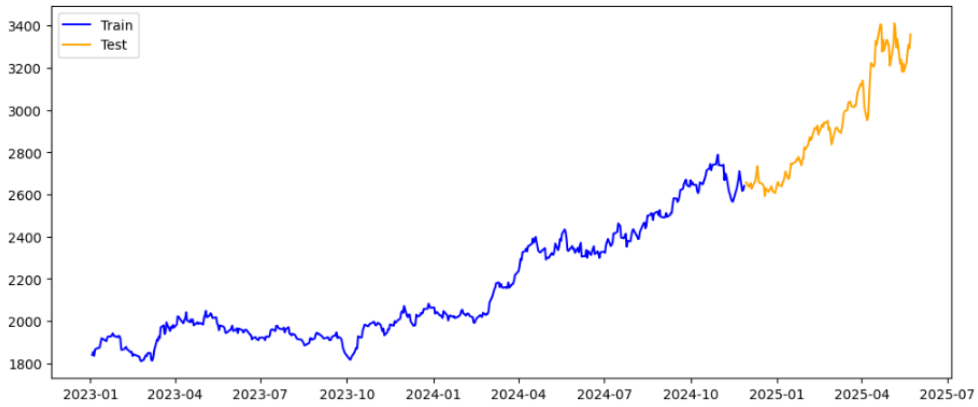
Figure number (04): Trend of Gold Daily Closing Prices



Source: Prepared by the researcher based on Python outputs.

The dataset was divided into two subsets: a training set comprising 80% of the total observations, and a testing set accounting for the remaining 20%. This partitioning was conducted to evaluate the model's generalization capability on previously unseen data, thereby assessing its predictive performance beyond the training sample.

Figure number (05): Train and test data



Source: Prepared by the researcher based on Python outputs.

The original time series displays a clear upward trend, suggesting that it is likely non-stationary. To confirm this, the Augmented Dickey-Fuller (ADF) test was conducted, the results presented in Table 2. The test yielded a p-value of 0.994, which is greater than the 0.05 significance level. Therefore, the null hypothesis of a unit root cannot be rejected, indicating that the original closing price series is non-stationary.

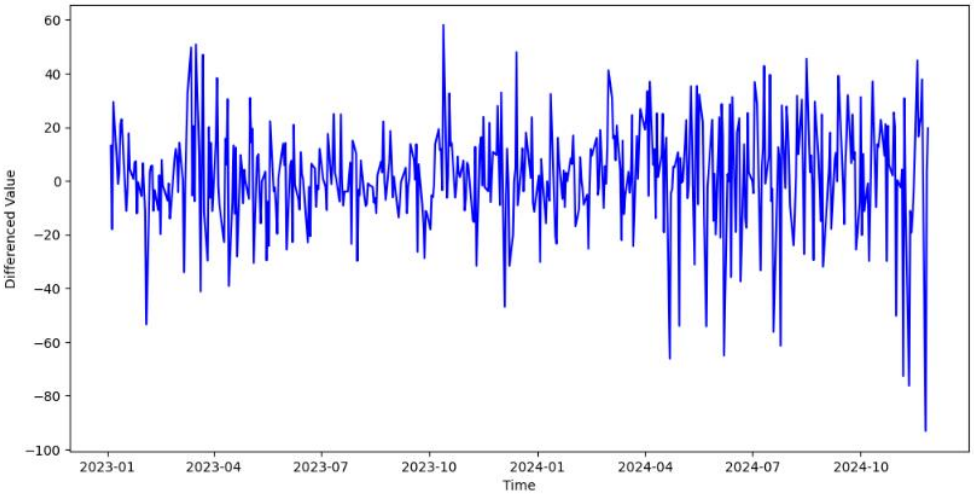
To address this issue, first-order differencing was applied to the series. The transformed series is shown in Figure 9, where the vertical axis represents the day-to-day differences in closing price. The differenced series appears to fluctuate around a constant mean, suggesting stationarity. This was confirmed by a second ADF test on the differenced series, with results summarized in Table 3. The test yielded a p-value of 0.000, allowing the null hypothesis to be rejected at the 5% level. Thus, the first-order differenced series is considered stationary and suitable for time series modeling.

Table number (02): ADF Test Results Table

Metric	Value
ADF Statistic	0.997575
p-value	0.994240
Lags Used	0.000000
Number of Observations Used	600.000000
Critical Values (1%)	-3.441296
Critical Values (5%)	-2.866369
Critical Values (10%)	-2.569342

Source: Prepared by the researcher based on Python outputs.

Figure number (06): First-Order Differenced Time Series Diagram



Source: Prepared by the researcher based on Python outputs.

Table number (03): First-Order Differenced ADF Test Results Table

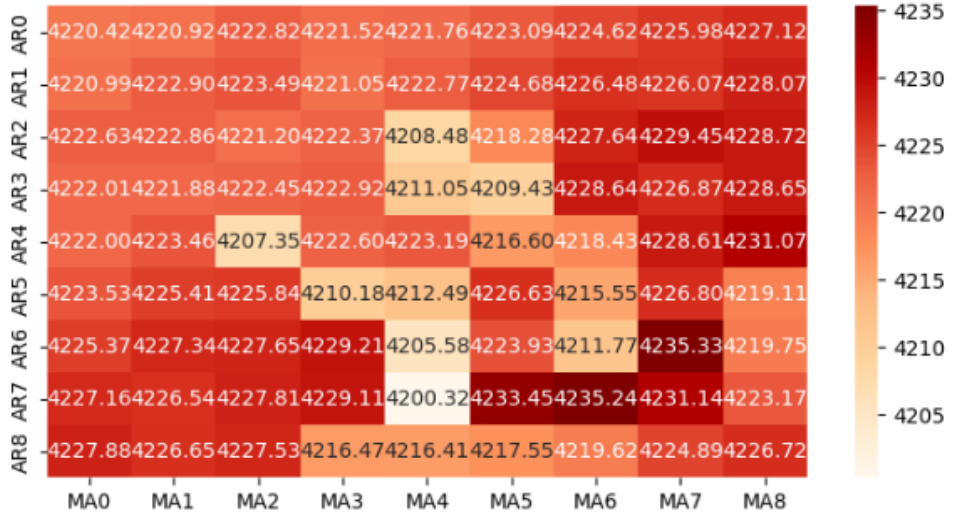
Metric	Value
ADF Statistic	-25.253708
p-value	0.000000
Lags Used	0
Number of Observations Used	599
Critical Values (1%)	-3.441314
Critical Values (5%)	-2.866377
Critical Values (10%)	-2.569346

Source: Prepared by the researcher based on Python outputs.

2-2-2- Model Order Determination

After first-order differencing, the data series becomes stationary, so the model is preliminarily determined as d = 1. To ensure the model's optimality, the AIC criterion is used to determine the optimal p and q values. The traditional method involves trying different p and q values, but this paper uses Python to find the p and q values with the smallest AIC value, as shown in Table 4. The ARIMA(7,1,4) model has the smallest AIC value, so it is selected as the final model.

Table number (04): AIC Values for Different ARIMA Models

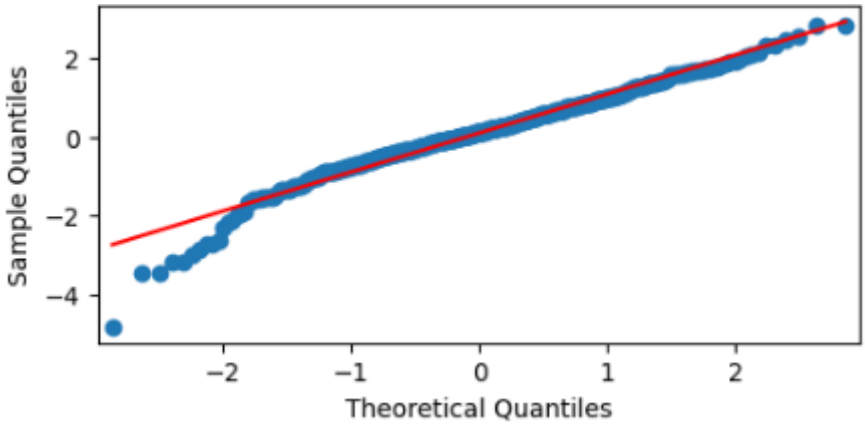


Source: Prepared by the researcher based on Python outputs.

2-2-3- Model Testing

After determining the model parameters, the model needs to be tested. First, the residuals are tested for normality using a QQ plot, as shown in Figure 7. The QQ plot shows that most data points are close to the straight line, indicating that the residuals follow a normal distribution. Second, the Durbin-Watson test is performed, and the resulting statistic is 1.9861, as shown in Table 5. This value is sufficiently close to the ideal value of 2, indicating the absence of significant first-order autocorrelation in the residuals. Accordingly, the residuals can be considered approximately independent.

Figure number (07): QQ Plot



Source: Prepared by the researcher based on Python outputs.

Table number (05): Model Prediction Results

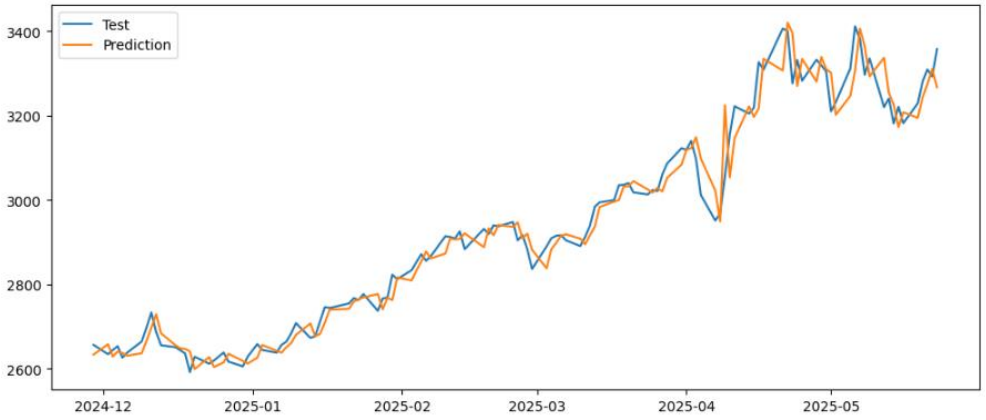
	c
Durbin-Watson test on residuals	1.986176
Test for all AR roots outside unit circle (>1)	True
Test for all MA roots outside unit circle (>1)	True

Source: Prepared by the researcher based on Python outputs.

2-2-4- Model Prediction

Based on the ARIMA (7,1,4) model, The model's predictive performance was assessed using the test data. The predicted and actual data are visualized in Figure 8. The horizontal axis represents the date, the red line represents the predicted values, and the blue line represents the test values.

Table number (08): Model Prediction Results



Source: Prepared by the researcher based on Python outputs.

Table (08) below, shows the performance evaluation metrics for this methodology.

Table number (06): Performance Evaluation Metrics for the ARIMA Model

Metric	Value
MAE	31.418978
MSE	1839.671265
RMSE	42.891388

Source: Prepared by the researcher based on Python outputs.

2-3- LSTM Neural Network Model Establishment

2-3-1- Data Standardization

To eliminate the influence of different scales on the calculation results, the data is standardized using the $0 \sim 1$ mean normalization method. The standardization function is as follows:

$$X_{\text{new}} = \frac{X_{\text{old}} - \text{mean}}{\text{std}}$$

2-3-2- Loss Function Selection

The mean squared error (MSE) is selected as the loss function, calculated as follows:

$$\text{MSE} = \frac{\sum_{i=1}^{n} (\text{Predicted Value}_i - \text{Actual Value}_i)^2}{N}$$

2-3-3- Model Construction

The LSTM neural network architecture was constructed using the Keras API in Python. The model was designed to capture both short-term and long-term dependencies in the daily closing price series while preventing overfitting through appropriate regularization techniques.

The network comprises two stacked LSTM layers. The first LSTM layer contains 50 memory cells and is configured with 'return_sequences=True' to propagate temporal features to the subsequent layer. This is followed by a dropout layer with a dropout rate of 0.2, which randomly deactivates 20% of the neurons during training to enhance model generalization. The second LSTM layer also consists of 50 memory cells and is connected to a fully connected Dense output layer with a single neuron. A linear activation

function is used in the output layer to align with the continuous nature of the forecasted variable.

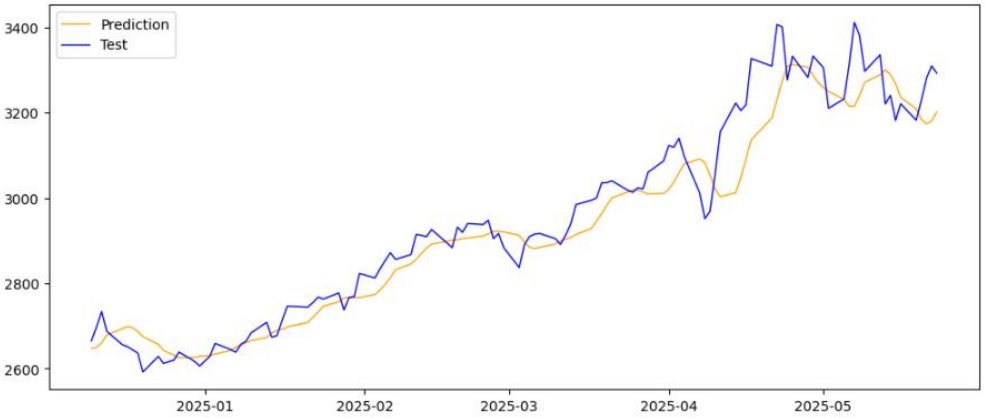
The model was compiled with the Mean Squared Error (MSE) loss function and the Adam optimizer, initialized with a learning rate of 0.001. During training, early stopping was employed based on the validation loss, with a patience of 10 epochs to halt training when no improvement was observed. Additionally, model checkpoints were saved at the epoch corresponding to the lowest validation loss, ensuring the best-performing weights were retained.

Training was performed for a maximum of 100 epochs using a batch size of 32. This configuration balances computational efficiency and model accuracy, leveraging the LSTM's capability to learn complex, nonlinear patterns in financial time series data while mitigating the risks of overfitting.

2-3-4- Model Prediction

The model is trained using the training data set and validated using the test set. The predicted and actual data are visualized in Figure 9.

Figure number (09): Model Prediction Results



Source: Prepared by the researcher based on Python outputs.

Table (08) below, which shows the performance evaluation metrics for this methodology.

Table number (08): Performance Evaluation Metrics for the LSTM Model

Metric	Value
MAE	58.599804
MSE	5801.610138
RMSE	76.168301

Source: Prepared by the researcher based on Python outputs.

3-Results and discussion

This section presents and analyzes the empirical results obtained from both the ARIMA and LSTM models developed for forecasting the daily closing prices of gold. The comparison focuses on predictive accuracy, model fit, and statistical robustness, with a view to determining the more effective approach.

3-1-ARIMA Model Performance

Following the application of the Augmented Dickey-Fuller test, the original series was found to be non-stationary, as indicated by a p-value of 0.994. First-order differencing successfully addressed this issue, yielding a stationary series validated by a subsequent ADF test with a p-value of 0.000. The optimal ARIMA configuration was identified as ARIMA(7,1,4), selected based on the minimum AIC value.

The model's residuals exhibited no significant autocorrelation, as evidenced by a Durbin-Watson statistic of 1.9861, approximating the ideal value of 2. The residuals also demonstrated approximate normality, as confirmed by the QQ plot analysis. These diagnostic results suggest that the ARIMA model is statistically adequate and well-specified.

In terms of predictive accuracy, the ARIMA model yielded a Mean Absolute Error (MAE) of 31.42, a Mean Squared Error (MSE) of 1839.67, and a Root Mean Squared Error (RMSE) of 42.89. The relatively low error metrics confirm the model’s effectiveness in capturing the linear structure of gold price dynamics over the study period.

3-2-LSTM Model Performance

The LSTM neural network was constructed with a three-layer architecture optimized through multiple experimental iterations. The final structure consisted of two hidden LSTM layers with 50 nodes each, followed by a dense output layer. The model was trained using the Adam optimizer and employed the Mean Squared Error (MSE) as the loss function.

Compared to ARIMA, the LSTM model produced higher error metrics: an MAE of 58.60, an MSE of 5801.61, and an RMSE of 76.17. These

results indicate that the LSTM model underperformed relative to the ARIMA model for this specific dataset and forecasting task.

While LSTM networks generally excel in capturing nonlinear and long-range dependencies in time series data, their performance is highly sensitive to the selection of hyperparameters, training data characteristics, and network complexity. In this case, the relatively short time span and limited data size may have constrained the LSTM's learning capacity and generalization performance.

3-3-Comparative Discussion

The comparative analysis highlights several important insights. First, the ARIMA model, despite its linear structure, demonstrated superior predictive accuracy over the LSTM model in forecasting short-term gold price movements. This may be attributed to the stationarity achieved through differencing and the relatively stable pattern of the gold price series within the study period.

Second, the LSTM model's weaker performance underscores the challenges associated with training deep learning models on limited financial data. Although LSTM networks are theoretically better suited for handling complex and nonlinear patterns, they may require larger datasets and more extensive tuning to outperform traditional time series models.

Third, these findings reaffirm the importance of model selection based on the nature of the data and the forecasting objective. For relatively stable and short-term price forecasting tasks, traditional statistical models like ARIMA may offer more reliable and interpretable results. Conversely, for longer-term predictions or more volatile datasets, the flexibility of LSTM and other deep learning models may become advantageous.

Conclusion

This study set out to evaluate and compare the performance of two prominent forecasting models -ARIMA and LSTM- in predicting the daily closing prices of gold. By employing a dataset comprising gold futures data from January 1, 2023, to May 25, 2025, the analysis aimed to assess the accuracy, robustness, and applicability of both models in the context of commodity price forecasting.

The empirical findings demonstrate that the ARIMA model outperformed the LSTM neural network across all performance metrics, including MAE, MSE, and RMSE. Despite the growing appeal of deep learning approaches in financial modeling, the LSTM model, in this case, failed to surpass the performance of its statistical counterpart. This outcome may be attributed to the limited size of the dataset, the relatively short forecasting horizon, and

the predominantly linear behavior exhibited by the gold price series over the study period.

The ARIMA model, after appropriate differencing to ensure stationarity, proved to be well-suited for capturing short-term price dynamics. Its residual diagnostics confirmed model adequacy, and its predictive accuracy was statistically robust. In contrast, the LSTM model, though theoretically capable of capturing nonlinearities and complex temporal dependencies, was likely constrained by insufficient data and the sensitivity of neural networks to hyperparameter configurations.

The comparative analysis reinforces the notion that model selection should be grounded in empirical validation rather than theoretical expectations. While machine learning methods offer powerful tools for modeling financial time series, their effectiveness is highly context-dependent. Traditional econometric models such as ARIMA remain relevant, especially when data are limited and exhibit regular patterns.

Future research may benefit from expanding the dataset both in terms of temporal scope and feature diversity, incorporating macroeconomic indicators, market sentiment variables, and external shocks. Moreover, hybrid models that combine the strengths of statistical and deep learning approaches such as ARIMA-LSTM architectures warrant further investigation for improved forecasting performance in volatile commodity markets.

Bibliography

Gao, M., Shi, G., & Li, S. (2018). Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network. *Sensors*, *18*(12), 4211.

Gong, W. (2024). Research on gold price forecasting based on lstm and linear regression. SHS Web of Conferences,

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Madhika, Y. R., Kusrini, K., & Hidayat, T. (2023). Gold Price Prediction Using the ARIMA and LSTM Models. *Sinkron: jurnal dan penelitian teknik informatika*, *7*(3), 1255-1264.

Mun, W. K., Sufahani, S. F., Kamil, A. A., & Nawawi, M. K. M. (2025). PREDICTION OF GOLD PRICES USING HYBRID MODEL ARIMA-LSTM. *Advances and Applications in Statistics*, *92*(5), 749-766.

Ngah, S., Rohani , A. B., Embong, A., & Razali, S. Two-steps implementation of sigmoid function for artificial neural network in field programmable gate array. *ARPN Journal of Engineering and Applied Sciences*, 4882-4888.

Setyowibowo, S., As' ad, M., Sujito, S., & Farida, E. (2022). Forecasting of daily gold price using arima-garch hybrid model. *Jurnal Ekonomi Pembangunan*, *19*(2), 257-270.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series. 2018 17th IEEE international conference on machine learning and applications (ICMLA),

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. *arXiv preprint arXiv:1911.09512*.

Xiao, Y., & Yin, Y. (2019). Hybrid LSTM neural network for short-term traffic flow prediction. *Information*, *10*(3), 105.