I.T.C

Content list available at ITC

Techno-Science Research Journal

Journal Homepage: http://techno-srj.itc.edu.kh/

Techno-Science Research Journal

An Empirical Investigation of Gold Price Forecasting Using ARIMA Compare with LSTM Model

Seng Hak Leng [1*], Sokkhey Phauk [1], Sothea Has [2]

[1] Department of Applied Mathematics and Statistics, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia

[2] Laboratoire de Probabilités, Statistique et Modélisation, Université Paris Cité, 85 boulevard Saint-Germain, 06e arr., Paris, France

Received: 29 April 2023; Accepted: 13 July 2023; Available online: December 2023

Abstract: *Time series forecasting is a well-established research domain, particularly in finance and econometrics, with a multitude of methods and algorithms proposed to achieve accurate future trend predictions. This study aims to examine the effectiveness of two popular models, ARIMA and LSTM, for predicting trends in gold prices in finance and econometrics. Monthly global gold prices from January 2010 to December 2022 are analyzed, with a training set from January 2010 to December 2020, a validation set of 12 months randomly selected from the training set, and a test set from January 2021 to December 2022. The results show that the LSTM model with a forget gate cell at 600 epochs yields the highest accuracy in term of RMSE, MAPE and SMAPE, surpassing all other models, including the ARIMA model. The study also suggests that increasing the number of epochs beyond 600 does not lead to significant improvements in the LSTM model's performance. While the ARIMA model is simpler to implement and requires less time for parameter tuning and training, it is less accurate than the LSTM model. Incorporating a peephole connection to the LSTM cell does not improve the model's accuracy or training speed. The study's outcomes provide valuable insights into optimal practices for gold price prediction, with implications for decision-making and risk management processes.*

Keywords: ARIMA; LSTM; Deep Learning; Peephole Connection; Time Series Analysis; RNN

1. INTRODUCTION

Time series analysis utilizes mathematical techniques to investigate sequential data collected at regular intervals over time. It has broad applications in multiple fields, including finance, economics, retail stock, healthcare, and environmental studies. The three key components of time series data, namely the trend, seasonality, and residual, are crucial for achieving accurate data interpretation and comprehension [1]. The examination of time series data enables the identification of patterns and information, which can be utilized to enhance the precision of forecasting through statistical techniques and mathematical models [2]. In this research article, we will study time series analysis on the gold price to forecast and conduct empirical investigations on the gold price based on the auto-regressive integrated moving average (ARIMA) model and compare it with the long-short term memory (LSTM) model.

Gold is a significant commodity exchanged incessantly throughout history [3]. Even a marginal enhancement in gold prediction methodologies can result in substantial profits [4]. Despite the pandemic disruptions, gold has remained a stable "safe haven." [5]. The Autoregressive Integrated Moving Average (ARIMA) model has been widely employed for time series forecasting to predict gold prices [6-8] evaluated several forecasting methods including Random Walk, Exponential Smoothing State Space Model, ARIMA, and Vector Autoregression to predict gold prices for 24 months. Their findings showed that ARIMA had the highest accuracy with an RMSE of 95.70. With the surging prominence of deep learning, scholars have utilized Long Short-Term Memory (LSTM) models for time series forecasting and have emphasized the significance of considering LSTM hyperparameter sensitivity [9-11]. These studies have also employed LSTM models with peephole connections to explore multivariate forecasting and the summarization of letters in text [12-14]. Expanding upon previous studies, our research will utilize the ARIMA model, LSTM model with forget gate cell, and LSTM model with peephole connection gate cell to investigate the implementation

[*] Corresponding author: Seng Hak Leng
E-mail: leng_senghak@gsc.itc.edu.kh; Tel: +855-12 184 848

of univariate time series forecasting. We will then carry out an empirical analysis of our results.

The paper will be partitioned into five discrete sections, each of which will perform a distinctive function. The first section will furnish a comprehensive outline of the research. The second section will explicate the methodology utilized to execute the research, and the third section will focus on the examination and interpretation of the outcomes derived from the experiments performed. The final section will proffer a definitive synopsis of the discoveries made and possible areas for future investigation.

2. METHODOLOGY

2.1. Dataset

In the context of our empirical inquiry, we intend to study the mean monthly gold price emanating from the London Bullion Market Association [15] over a span of 13 years, commencing from 2010 and culminating towards the end of 2022 as shown in Fig. 1.

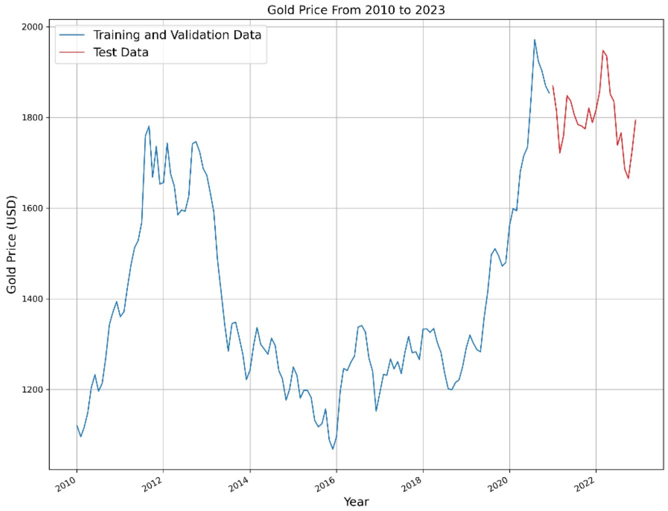

Fig. 1. Average monthly gold price from 2010 to 2023.

The 156 observations will be divided into a training and test set for an investigation based on prior research. The data depicts non-stationarity, requiring the use of gold price data from 2010 to 2021 as the training dataset, with 12 random observations as the validation set, and gold price data from 2021 to 2023 as the testing dataset. Standardization scaling will be conducted on the training set to optimize model accuracy [16-17].

2.2. Conceptual model and processing

The ARIMA model is a classical linear statistical model that predicts linearly future observations based on past observations [18]. The model is a composition of an Autoregressive (AR) process and a Moving Average (MA) process, producing an Autoregressive Moving Average (ARMA) model, which assumes stationarity [19]. The ARIMA model extends the ARMA model by incorporating an integrated term "I," which transforms the non-stationary series into a stationary one through differencing [20].

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \qquad \text{(Eq. 1)}$$

Eq. (1) represents ARMA(p,q) model equation, where xt is a stationary variable, c is a constant, ϕi represents autocorrelation coefficients at lags 1 to p, εt is a Gaussian white noise with zero mean and variance σε2 (>0), θi denotes weights applied to current and previous values, and θ0 = 1 and θi ≠1. By performing differencing to attain a stationary series with order d, the ARMA model is transformed into an ARIMA(p,d,q) model, with p, d, and q as its three parameters.

2.2.1 Augmented Dickey-Fuller Test (ADF)

The Augmented Dickey-Fuller (ADF) test, denoted by Eq. (2), to address non-stationary sequential data caused by a unit root in univariate time series.

$$\Delta y_t = \mu + \beta_t + \gamma y_{t-1} + \sum_{i=1}^{p} \Delta y_{t-i} \mu_t \qquad \text{(Eq. 2)}$$

In Eq. (2), Δ symbolizes the difference operator, β represents the trend coefficient, and μt denotes the white noise. The ADF test includes an AR(p) process with lagged p, as illustrated in Eq. (2), with the test statistic computed through OLS regression of the coefficients γ in the equation. The ADF test hypotheses are H0: γ = 0, indicating a unit root within the series, and HA: γ < 0, the alternative hypothesis. However, the ADF test is sensitive to lag order in finite samples, which can result in lag adjustment issues [21].

Table 1 Integral square error (ISE) rating

Rating	ISE Value
Excellent	< 3.0
Very good	3.0-6.0
Good	6.0-10.0
Fair	10.0-25.0
Poor	> 25.0

2.2.2 Kwiatkowski–Phillips–Schmidt–Shin Test (KPSS)

The Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test assesses stationarity in time series data with the test statistic expressed by Eq. (3).

$$KPSS = \frac{1}{N^2 \hat{\sigma}_N^2} \sum_{i=1}^{N} S_n^2 \qquad \text{(Eq. 3)}$$

where a consistent estimator, estimates the residuals' long-run variance, σ2. The cumulative sum of deviations from the time series trend, S, represents the test's integral component, while N denotes the number of observations. The comparison of the unit root test outcomes can facilitate the comprehension of the series' stationarity properties. When the ADF test fails to reject the null hypothesis, and the KPSS test rejects it, the non-stationarity of the series is more credible [22].

2.2.3 Bayesian Information Criterion (BIC)

The Bayesian Information Criterion (BIC) or Schwarz criterion selects the best model by penalizing the number of parameters to avoid overfitting. The BIC value is given by Eq. 4, where k denotes the number of parameters and $\hat{\sigma}_e^2$ is the error variance, calculated using Eq. 5 with *n* representing the model's observations.

$$BIC = n\ln(\hat{\sigma}_e^2) + k\ln(n) \qquad \text{(Eq. 4)}$$

$$\hat{\sigma}_e^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x}_i)^2 \qquad \text{(Eq. 5)}$$

In Eq. 5, x_i represents the actual observation, while $\bar{x}_{\mathrm{i}}$ is the forecasted value by a specific ARIMA model for the corresponding lag *i*. The model with the smallest BIC value is chosen [23].

2.3 Long short-term memory

In the second part of our experiment, we will use the LSTM algorithm, a type of RNN, to forecast gold prices. We will employ two LSTM architectures, LSTM with forget gate and LSTM with peephole connection, to address the issue of vanishing and exploding gradients during training, as described by [24].

2.3.1 Long short-term memory with forget gate

Despite the Long Short-Term Memory (LSTM) model's success in mitigating the issues of vanishing and exploding gradients in Recurrent Neural Networks (RNNs), it suffers from the inability to reset the internal state, resulting in network breakdown. To overcome this limitation, an LSTM model incorporating a forget gate was proposed. In our experimental setup, we employ the LSTM cell incorporating a forget gate, as illustrated in Fig. 2. The LSTM cell consists of three gates, namely the forget gate, input gate, and output gate, as depicted in the figure.

The computation within each LSTM cell can be expressed using Eq. 6 through Eq. 11, where the symbols *W*, *b*, *σ*, and *tanh* denote the weights, biases, sigmoid, and tan-hyperbolic functions, respectively. Furthermore, the symbols i_t, f_t, h_t, and o_t represent the input gate, forget gate, hidden state, and output gate, respectively, at a given timestamp *t*.

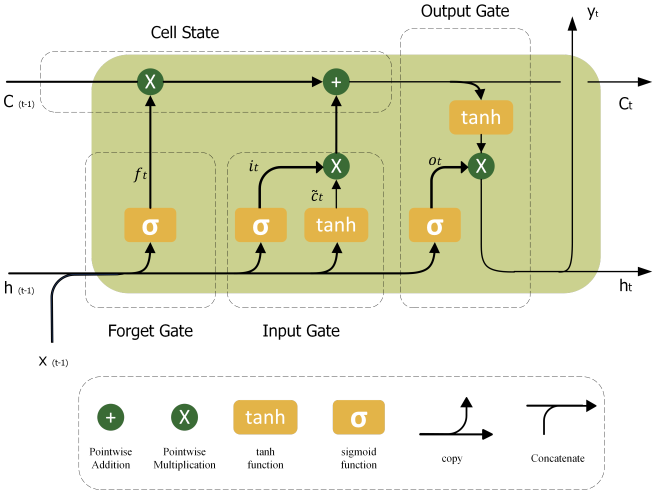

Fig. 2. LSTM with forget gate cell architecture.

$$f_t = \sigma\left(W_{fh}h_{t-1} + W_{fx}x_t + b_f\right) \qquad \text{(Eq. 6)}$$

$$i_t = \sigma\left(W_{ih}h_{t-1} + W_{ix}x_t + b_i\right) \qquad \text{(Eq. 7)}$$

$$\tilde{c}_t = \tanh\left(W_{\tilde{c}_t h}h_{t-1} + W_{\tilde{c}_t x}x_t + b_{\tilde{c}}\right) \qquad \text{(Eq. 8)}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c_t \qquad \text{(Eq. 9)}$$

$$o_t = \sigma\left(W_{oh}h_{t-1} + W_{ox}x_t + b_o\right) \qquad \text{(Eq. 10)}$$

$$h_{t-1} = o_t \cdot \tanh(c_t) \qquad \text{(Eq. 11)}$$

2.3.2 Long short-term memory with peephole connect

Felix A. Gers et al. [25] proposed a peephole connection from the previous cell state in the input to improve the LSTM model's performance. Fig. 3 shows the LSTM cell architecture with a peephole connection.

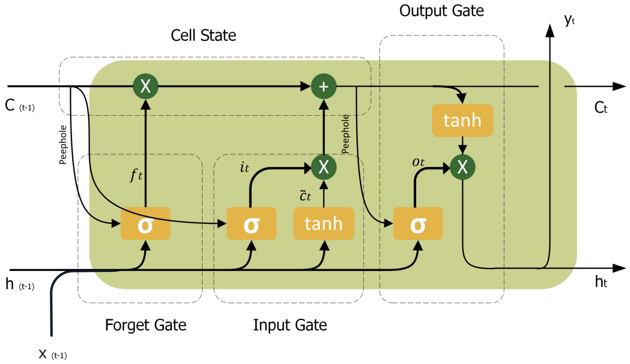

Fig. 3. LSTM with peephole connection cell architecture.

$$f_t = \sigma\left(W_{fh}h_{t-1} + W_{fx}x_t + P_f \cdot c_{t-1} + b_f\right) \qquad \text{(Eq. 12)}$$

$$i_t = \sigma\left(W_{ih} h_{t-1} + W_{ix} x_t + P_i \cdot \mathrm{c}_{t-1} + b_i\right) \quad \text{(Eq. 13)}$$

$$\tilde{c}_t = \tanh\left(W_{\tilde{c}_t h} h_{t-1} + W_{\tilde{c}_t x} x_t + b_{\tilde{c}}\right) \quad \text{(Eq. 14)}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad \text{(Eq. 15)}$$

$$o_t = \sigma\left(W_{oh} h_{t-1} + W_{ox} x_t + P_o \cdot \mathrm{c}_t + b_o\right) \quad \text{(Eq. 16)}$$

$$h_{t-1} = o_t \cdot \tanh(c_t) \quad \text{(Eq. 17)}$$

The primary difference between the peephole and forget gate LSTM cells is the inclusion of the previous cell state in the forget gate, input gate, and output gate. The LSTM cell's mathematical representation using Eq. 12 through Eq. 17 is possible based on the Fig. 3 illustration. Here, *W* and *b* denote the weights and bias, respectively, while *σ* and *tanh* signify the sigmoid and tan-hyperbolic activation functions. *P* denotes the coefficient of the peephole connection from the previous cell state to the respective gate at a given time *t*.

2.4 Model evaluation metrics

This study will assess the model's effectiveness using the root mean square error (RMSE), mean absolute percentage error (MAPE), and symmetric mean absolute percentage error (SMAPE) metrics to ensure a fair comparison.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(Y_{\text{actual,i}} - Y_{\text{forecast,i}}\right)^2} \quad \text{(Eq. 18)}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{\left(Y_{\text{actual,i}} - Y_{\text{forecast,i}}\right)}{Y_{\text{actual,i}}}\right| \quad \text{(Eq. 19)}$$

$$SMAPE = \frac{1}{m}\sum_{i=1}^{m}\frac{2\cdot\left|Y_{\text{actual,i}} - Y_{\text{forecast,i}}\right|}{\left|Y_{\text{actual,i}}\right| + \left|Y_{\text{forecast,i}}\right|} \quad \text{(Eq. 20)}$$

RMSE, expressed in Eq. 18, is useful in identifying outliers and quantifies the difference between actual and forecasted values. MAPE, computed using Eq. 19, represents an alternative metric for evaluating regression models, similar to a weighted Mean Absolute Error (MAE) regression. SMAPE, introduced as a solution to MAPE's unboundedness, is expressed in Eq. 20, and produces symmetrical errors in asymmetric forecasting ranges, avoiding the generation of unbounded, extremely large, or infinite errors commonly associated with MAPE. Additionally, SMAPE is more resilient to outliers, assigning less weight to them relative to other measures that lack error thresholds [26-27].

3. RESULTS AND DISCUSSION

3.1. ARIMA model parameters estimation

We conducted ADF and KPSS tests on the dataset to determine unit roots and stationarity.

Table 2. ADF and KPSS test results on the training set

Stationary Test		ADF Test		KPSS Test	
		d = 0	d = 1	d = 0	d = 1
Statistics		-1.585	-4.513	0.551	0.122
p-value		0.491	0.0002	0.030	0.100
Critical values	1%	-3.474	-3.475	0.739	0.739
	5%	-2.880	-2.881	0.463	0.463
	10%	-2.577	-2.577	0.347	0.347

The results in Table 1 show that the training dataset has a unit root when using zero order difference, but is stationary when using first-order difference (d=1), as indicated by the rejection of the null hypothesis in ADF test and acceptance of the alternative hypothesis in KPSS test at a 5% significance level. Therefore, we opt for the model's first-order difference.

Table 3. BIC values for ARIMA model at first differences

d = 1	MA(0)	MA(1)	MA(2)	MA(3)	MA(4)
AR(0)	-9.63	-14.84	-10.19	-6.572	-4.599
AR(1)	-13.93	-12.12	-9.37	-4.94	-0.42
AR(2)	-9.58	-9.66	-4.41	-0.59	4.18
AR(3)	-5.04	-4.82	-0.56	-2.73	2.03
AR(4)	-0.66	0.11	4.45	1.05	10.70

Table 2 displays the Bayesian Information Criterion (BIC) values for various ARIMA models at the first-order difference. The preferred model is ARIMA(4,1,1) with a BIC value of 0.11.

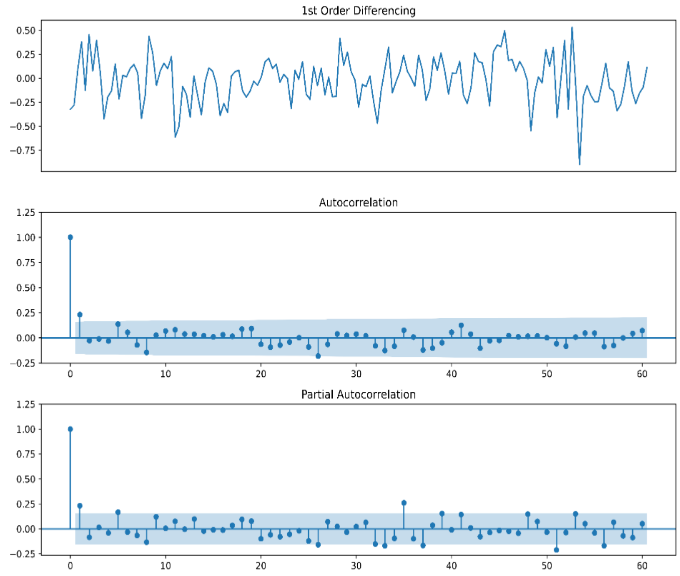

Fig. 1 Autocorrelation and Partial Autocorrelation plot

The autocorrelation (AC) and partial autocorrelation (PAC) plots of the series are presented in Fig. 4. The plot includes the first-order differencing plot, AC plot, and PAC plot of the series. The AC plot shows a cut-off at lag 1, and the PAC plot displays decay at lag 4, supporting the model selected using the BIC values from the grid search method. Thus, the optimal forecasting model will be ARIMA(4,1,1).

3.2 LSTM model parameters estimation

This research aims to optimize the Long Short-Term Memory (LSTM) model by tuning hyperparameters, including the number of layers, hidden layers per cell, and learning rate. The LSTM model will be trained using an NVIDIA RTX 3050Ti with 4GB of RAM, PyTorch, and the Adam optimizer with a learning rate of 0.01. Data preprocessing will be carried out on an AMD Ryzen7 5800h CPU with 8 cores and 16GB of RAM, utilizing a batch size of 24 and shuffling facilitated by the PyTorch data loader.

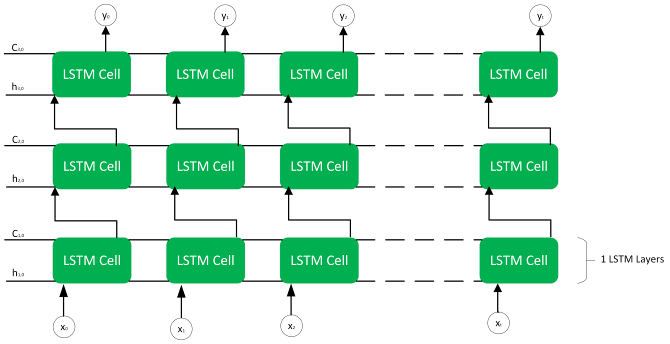

Fig 2. LSTM Network with 3 LSTM Layers.

Fig. 5 shows a model architecture with 3 LSTM layers. The number of LSTM layers will be determined through a tuning process that explores LSTM architectures with forget gate and peephole connections, depicted in Fig. 2 and 3, respectively. Each LSTM cell contains 6 neurons, and the tuning process spans from 1 to 4 layers over 1200 epochs [28], with a learning rate of 0.01 during layer and hidden neuron tuning. Fig. 6 displays validation set cost across epochs for various numbers of LSTM layers.

The validation cost results for various numbers of Long Short-Term Memory (LSTM) layers are depicted in Fig. 6. The 4-layer LSTM network, utilizing the LSTM forget gate cell, demonstrated the lowest validation cost at roughly 900 epochs. However, the 3-layer LSTM network, utilizing the peephole connection LSTM cell, demonstrated the lowest validation cost, but at around 600 epochs.

Fig. 7 illustrates the use of a four-layer LSTM model with an LSTM forget gate cell and a three-layer LSTM model with peephole connection LSTM cell, where the number of hidden neurons is varied (3, 6, 9, and 12). At 800 epochs, the LSTM model with an LSTM forget gate cell and 6 hidden neurons has the lowest validation cost, while the LSTM with a peephole connection LSTM cell and 9 hidden neurons has the lowest validation cost. Hyperparameter optimization was then conducted to determine the optimal learning rate for both models based on the validation cost, with the learning rate ranging from 0.1, 0.01, 0.001, and 0.0001 on the Adam optimizer. The optimal learning rate was chosen based on the lowest validation cost.

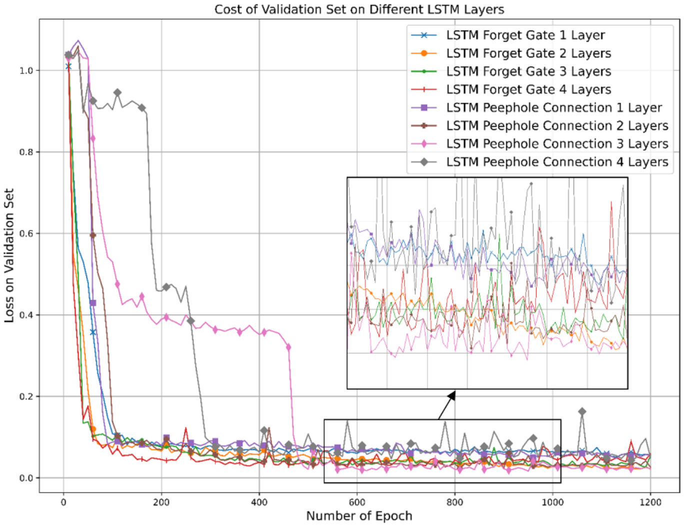

Fig. 3. Cost validation set by epoch in different LSTM layers.

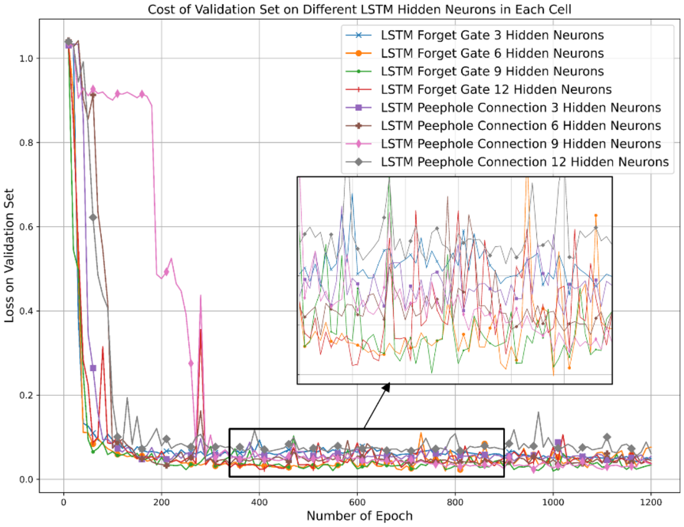

Fig. 4. Cost of Validation set by varying number of Neurons in LSTM cell.

Fig. 8 demonstrates that a learning rate of 0.01 yields optimal performance for both LSTM models. The LSTM model with LSTM forget gate cell comprises four layers with nine hidden neurons and the same learning rate, while the LSTM model with LSTM peephole connection cell consists of three layers with nine hidden neurons and the same learning rate. In the hyperparameter fine-tuning process, Fig. 6 recommends training for 600 and 900 epochs, Fig. 7 recommends 800 epochs, and Fig. 8 suggests around 800 epochs. We will evaluate both LSTM architectures' performance on the test set for the suggested numbers of epochs: 600, 800, and 900.

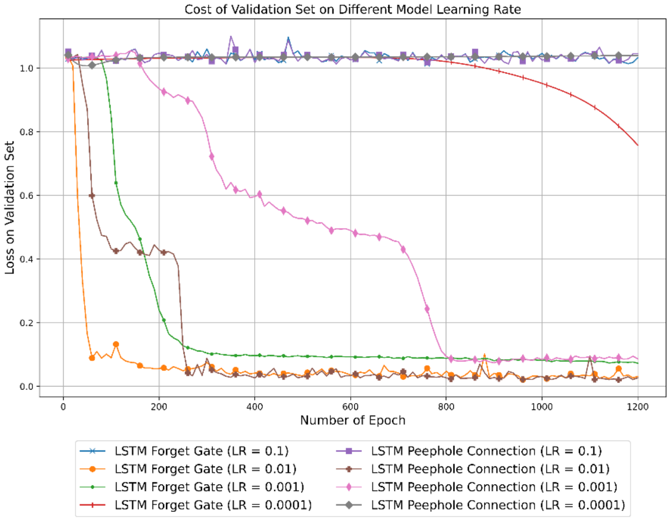

Fig. 5. Cost of validation by adjusting Model Learning Rate.

3.3 Empirical Investigation of ARIMA and LSTM forecasting

The optimal ARIMA(4, 1, 1) model, identified based on the training set, was trained and the residuals were analyzed. Fig. 9 displays the residual plot, indicating initial significant fluctuations in the residuals that gradually center around zero in later steps.

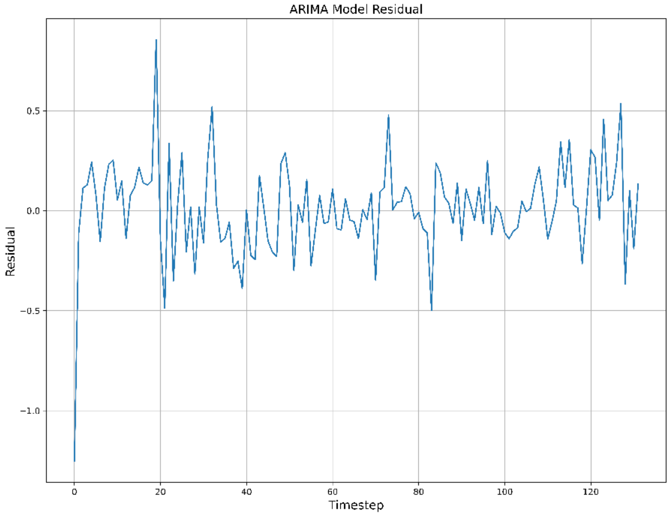

Fig. 6. ARIMA(4, 1, 1) model residual.

The density plot of the residuals, as illustrated in Fig. 10, demonstrates a distribution that closely approximates normality. This observation suggests that the model accurately captures the underlying patterns of the data and that any errors present are randomly and uniformly distributed.

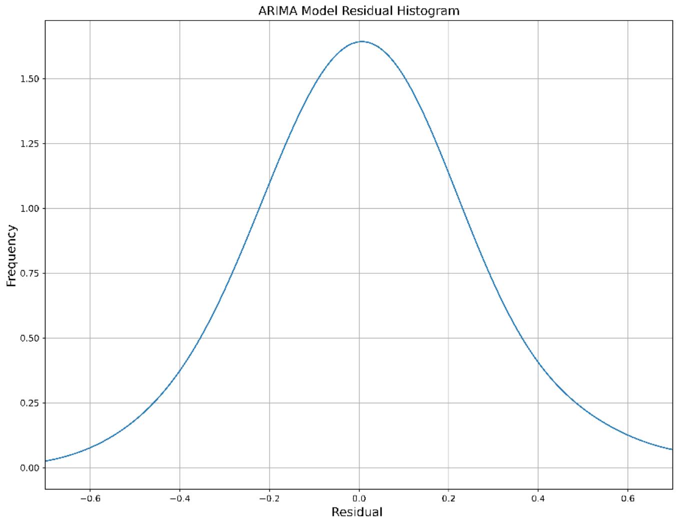

Fig. 7. ARIMA(4, 1, 1) model residual density.

By employing the Autoregressive Integrated Moving Average (ARIMA) model and Long Short-Term Memory (LSTM) models with different cell architectures. Fig. 11 illustrates the results obtained from the ARIMA(4, 1, 1) forecasting and the LSTM models with LSTM forget gate cell and peephole connection cell.

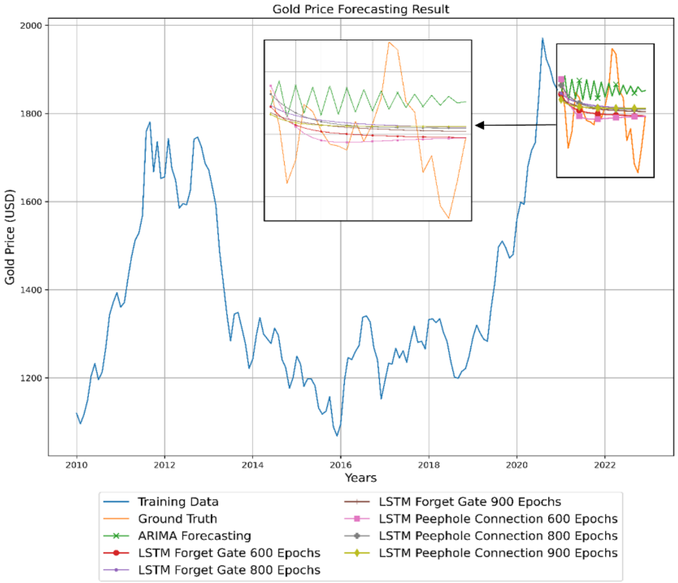

Fig. 8. ARIMA(4, 1, 1) and LSTM model Forecasting on Test Set.

As shown in Fig. 11, ARIMA(4, 1, 1) forecasting results in a fluctuating line compared to the smooth line obtained from the LSTM models. The LSTM models with peephole connection LSTM cell and forget cell can predict the overall trend of the gold price, whereas the ARIMA(4, 1, 1) model fails to capture this distinction. To conduct a fair evaluation, we calculated the error using the metrics described in Section 2.4.

Table 3 Forecasting Result on Test Set

Model	Number of Epoch	Forecasting Result			
		RMSE	MAPE	sMAPE	Time(s)
ARIMA(4, 1, 1)		87.2298	0.0388	0.0397	**0.0820**
LSTM Forget Gate	600	**66.5199**	**0.0282**	**0.0282**	21.7594
	800	68.8688	0.0294	0.0297	29.0852
	900	67.2441	0.0289	0.0290	32.8718
LSTM Peephole Connection	600	68.8480	0.0290	0.0289	28.5860
	800	69.5722	0.0295	0.0298	38.1608
	900	67.9864	0.0291	0.0293	43.0951

Table 3 presents the outcomes of a comparative analysis of various forecasting models with regard to their performance in predicting the test set. The results indicate that the ARIMA(4, 1, 1) model has the highest error rates in terms of RMSE, MAPE and sMAPE, with values of 87.2298, 0.0388, and 0.0397, respectively. In contrast, the LSTM model with a forget gate LSTM cell at 600 epochs achieved the best performance among all models, with an RMSE of 66.5199, MAPE of 0.0282, and sMAPE of 0.0282. The model's performance did not improve significantly with increased epochs, resulting in higher RMSE values at 800 and 900 epochs. The LSTM model with peephole connections performed better than the ARIMA(4, 1, 1) model but was inferior to the LSTM model with forget gate cells. The LSTM model with peephole connections did not perform better than forget cells. The LSTM model with peephole connections achieved its best performance at 900 epochs with an RMSE of 67.9864, MAPE of 0.0291, and sMAPE of 0.0293.

In terms of elapsed time, the ARIMA(4, 1, 1) model took the least amount of time to train, with only around 0.0820s. The model with the lowest RMSE took approximately 21.7594s to train, which was higher than the ARIMA model but lower than all other LSTM models. The LSTM model with peephole connection cells took longer to train compared to the LSTM model with forget gate cells, despite having fewer LSTM layers.

The investigation findings reveal that, in univariate time series analysis, the LSTM model outperforms the ARIMA model in the long-term forecasting of data due to its ability to capture long-term trends effectively. Incorporating a peephole connection to the LSTM cell does not enhance the model's performance in terms of accuracy or training speed. Through the process of hyperparameter tuning, it is evident that the LSTM model requires more effort and time to achieve optimal results.

4. CONCLUSIONS

The study employed the ARIMA(4, 1, 1) and LSTM models for forecasting gold prices. The ARIMA model demonstrated a good fit for the training set with residuals centered around zero and approximately normally distributed. The LSTM model with forget gate cell at 600 epochs provided the most accurate forecast for the test set with an RMSE of 66.5199, MAPE of 0.0282, and sMAPE of 0.0282. ARIMA was shown to be an easy method to implement and take less time in both parameter tuning and training. The LSTM model, on the other hands, take more efforts and time to train and tune but it provides more superior result. In univariate time series forecasting, the LSTM model with forget gate LSTM cell provides superior performance compared to the ARIMA model and the LSTM model with peephole connection cell.

To further advance this research, we propose studying an alternative LSTM model with a distinct cell architecture and exploring more robust methods for tuning the model's hyperparameters..

ACKNOWLEDGMENTS

I extend my sincere gratitude to Dr. PHAUK Sokkhey and Dr. HAS Sothea for their invaluable support and guidance throughout this project. Their profound insights, exceptional expertise, and unwavering encouragement were critical in enabling me to complete this work. Their mentorship has been a great source of inspiration and I am honored to have had the privilege of working with them.

REFERENCES

[1] Deb, C., Zhang, F., Yang, J., Lee, S. E., & Shah, K. W. (2017). A review on time series forecasting techniques for building energy consumption. Renewable and Sustainable Energy Reviews, 74, 902–924. https://doi.org/10.1016/j.rser.2017.02.

[2] Kirchgässner, G., Wolters, J., & Hassler, U. (2013). Introduction to Modern Time Series Analysis. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33436-8

[3] Aggarwal, R., & Lucey, B. M. (2007). Psychological barriers in gold prices? Review of Financial Economics, 16(2), 217–230. https://doi.org/10.1016/j.rfe.2006.04.001

[4] Liu, D., & Li, Z. (2017). Gold Price Forecasting and Related Influence Factors Analysis Based on Random Forest. In J. Xu, A. Hajiyev, S. Nickel, & M. Gen (Eds.), Proceedings of the Tenth International Conference on Management Science and Engineering Management (Vol. 502, pp. 711–723). Springer Singapore. https://doi.org/10.1007/978-981-10-1837-4_59

[5] Cui, M., Wong, W.-K., Wisetsri, W., Mabrouk, F., Muda, I., Li, Z., & Hassan, M. (2023). Do oil, gold and metallic price volatilities prove gold as a safe haven during COVID-19 pandemic? Novel evidence from COVID-19 data. Resources Policy, 80, 103133. https://doi.org/10.1016/j.resourpol.2022.103133

[6] Makala, D., & Li, Z. (2021). Prediction of gold price with ARIMA and SVM. Journal of Physics: Conference Series, 1767(1), 012022. https://doi.org/10.1088/1742-6596/1767/1/012022

[7] Ali, A., Ch, M. I., Qamar, S., Akhtar, N., Mahmood, T., Hyder, M., & Jamshed, M. T. (2016). Forecasting of Daily Gold Price by Using Box-Jenkins Methodology. International Journal of Asian Social Science, 6(11), 614–624. https://doi.org/10.18488/journal.1/2016.6.11/1.11.614.624

[8] Hassani, H., Silva, E. S., Gupta, R., & Segnon, M. K. (2015). Forecasting the price of gold. Applied Economics, 47(39), 4141–4152. https://doi.org/10.1080/00036846.2015.1026580

[9] Istiake Sunny, Md. A., Maswood, M. M. S., & Alharbi, A. G. (2020). Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model. 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), 87–92. https://doi.org/10.1109/NILES50944.2020.9257950

[10] Du, J., Liu, Q., Chen, K., & Wang, J. (2019). Forecasting stock prices in two ways based on LSTM neural network. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 1083–1086. https://doi.org/10.1109/ITNEC.2019.8729026

[11] Elsworth, S., & Güttel, S. (2020). Time Series Forecasting Using LSTM Networks: A Symbolic Approach. https://doi.org/10.48550/ARXIV.2003.05672

[12] Fu, L. (2020). Time Series-oriented Load Prediction Using Deep Peephole LSTM. 2020 12th International Conference on Advanced Computational Intelligence (ICACI), 86–91. https://doi.org/10.1109/ICACI49185.2020.9177688

[13] Yang, T., Wang, H., Aziz, S., Jiang, H., & Peng, J. (2018). A Novel Method of Wind Speed Prediction by Peephole LSTM. 2018 International Conference on Power System Technology (POWERCON), 364–369. https://doi.org/10.1109/POWERCON.2018.8601550

[14] Rahman, Md. M., & Siddiqui, F. H. (2021). Multi-layered attentional peephole convolutional LSTM for abstractive text summarization. ETRI Journal, 43(2), 288–298. https://doi.org/10.4218/etrij.2019-0016

[15] LBMA Gold Price. (2023). LBMA Precious Metal Prices. https://www.lbma.org.uk/prices-and-data/precious-metal-prices#/

[16] Wang, A. Y.-T., Murdock, R. J., Kauwe, S. K., Oliynyk, A. O., Gurlo, A., Brgoch, J., Persson, K. A., & Sparks, T. D. (2020). Machine Learning for Materials Scientists: An Introductory Guide toward Best Practices. Chemistry of Materials, 32(12), 4954–4965. https://doi.org/10.1021/acs.chemmater.0c01907

[17] Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., & Padma, V. (2020). Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 729–735. https://doi.org/10.1109/ICSSIT48917.2020.9214160

[18] Khashei, M., & Bijari, M. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Applied Soft Computing, 11(2), 2664–2675. https://doi.org/10.1016/j.asoc.2010.10.015

[19] Mills, T. C. (2019). Applied time series analysis: A practical guide to modeling and forecasting. Academic Press.

[20] Makridakis, S., & Hibon, M. (1997). ARMA Models and the Box-Jenkins Methodology. Journal of Forecasting, 16(3), 147–163. https://doi.org/10.1002/(SICI)1099-131X(199705)16:3<147::AID-FOR652>3.0.CO;2-X

[21] Cheung, Y.-W., & Lai, K. S. (1995). Lag Order and Critical Values of the Augmented Dickey–Fuller Test. Journal of Business & Economic Statistics, 13(3), 277–280. https://doi.org/10.1080/07350015.1995.10524601

[22] Nieh, C.-C., & Ho, T. (2006). Does the expansionary government spending crowd out the private consumption? The Quarterly Review of Economics and Finance, 46(1), 133–148. https://doi.org/10.1016/j.qref.2004.11.004

[23] Etebong P. Clement. (2014). Using Normalized Bayesian Information Criterion (Bic) to Improve Box—Jenkins Model Building. American Journal of Mathematics and Statistics, 4(5), 214–221.

[24] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2018). Recent Advances in Recurrent Neural Networks

http://arxiv.org/abs/1801.01078

[25] Felix A. Gers, Nicol N. Schraudolph, & Jurgen Schmidhuber. (2000). Learning Precise Timing with LSTM Recurrent Networks. CrossRef Listing of Deleted DOIs, 1. https://doi.org/10.1162/153244303768966139

[26] de Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2016). Mean Absolute Percentage Error for regression models. Neurocomputing, 192, 38–48. https://doi.org/10.1016/j.neucom.2015.12.114

[27] Chen, C., Twycross, J., & Garibaldi, J. M. (2017). A new accuracy measure based on bounded relative error for time series forecasting. PLOS ONE, 12(3), e0174202. https://doi.org/10.1371/journal.pone.0174202

[28] He, Z., Zhou, J., Dai, H.-N., & Wang, H. (2019). Gold Price Forecast Based on LSTM-CNN Model. 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 1046–1053. https://doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00188