Malaysian Journal of Mathematical Sciences 18(2): 399–422 (2024)
https://doi.org/10.47836/mjms.18.2.11

Malaysian Journal of Mathematical Sciences

Journal homepage: https://mjms.upm.edu.my

Implementation of Long Short-Term Memory for Gold Prices Forecasting

Nurhambali, M. R.*[1], Angraini, Y.[1], and Fitrianto, A.[1]

[1] *Department of Statistics, IPB University, Indonesia*

E-mail: rizky2710.edu@gmail.com
**Corresponding author*

Received: 9 August 2023
Accepted: 29 February 2024

Abstract

Gold is a form of investment known as a safe haven asset because of its stability in unstable market conditions. Gold price forecasting is important for investors as decisions making tool. This study aims to study the best long short–term memory (LSTM) hyperparameters (optimizer, learning rate, and epoch) from cross–validation for forecasting. LSTM, as part of deep learning methods, is developed based on a RNN widely used in time series forecasting. LSTM is superior compared to other methods for its ability to minimize errors and forecast for long–term periods. Walk–forward validation with sliding and extending window scenarios as a form of cross–validation is used to see the method's accuracy. The used data is sourced from the World Gold Council with daily data periods for January 1, 2003, to December 31, 2023. The optimizer used is Adam and RMSProp, each with learning rate values of 0.01, 0.001, 0.0001, and epoch values of 100, 500, 1000. The best model uses the Adam optimizer, a learning rate of 0.01, and an epoch value of 100 with a MAPE value of 0.4867% in the validation process. Forecasting results show a tendency for gold prices to increase in the next eight years.

Keywords: deep learning; forecasting; gold; hyperparameter; LSTM.

1 Introduction

Forecasting is one way to obtain information in the future by utilizing information in the past [24]. Currently, forecasting is developing a lot, not only using classical methods, such as smoothing or autoregressive integrated moving average (ARIMA), but also using deep learning. Deep learning has a learning layer which makes the accuracy and performance of deep learning better compared to other algorithms [39][3]. One of the deep learning methods commonly used for forecasting is long short-term memory (LSTM).

LSTM is the development of a neural network, namely a recurrent neural network (RNN). It has hyperparameters that can determine the level of reliability and performance of the model [49]. The hyperparameters commonly used in LSTM include learning rate, epoch, and optimizer. LSTM is often used for forecasting time series data because it is considered superior and reliable in predicting long-term periods compared to other algorithms [54][1]. This is supported by a research by Adhinata and Rakhmadani [2] and Bodapati et al. [9] who made predictions using Covid-19 data, showing that LSTM was quite successful in minimizing errors in fluctuating data. However, even though a method is said to be good, it is still important to evaluate it to see its accuracy, such as by using time series cross validation which will divide the data into training data and test data by taking into account the time sequence [32]. According to Khalid et al. [35], a type of cross validation that can be used to avoid errors in forecasting is walk forward validation.

Forecasting results are widely used for making decisions and planning in the future, one of which is in making investments. One type of investment that is quite popular is gold. According to Puspitasari et al. [43], gold is referred to as a safe haven asset, which is an asset that is considered to have a value that continues to increase or stays the same even though market conditions are unstable. Rising gold prices will encourage investors to choose to invest in gold rather than in the stock market because with relatively lower risk, gold can provide good returns with increasing prices [16]. The stability of gold prices was proven during the 2008-2009 global financial crisis where many commodity prices fell by around 40%, but global gold prices tended to increase by an average of 6% [15].

Gold price data is time series data because it is composed of a series of information recorded over a certain period of time so that this data allows for forecasting. Forecasting the price of gold using the LSTM method was previously carried out by Vidya and Hari [50] which shows that the LSTM is better than the ARIMA, matrix estimation of variance, deep regression, support vector regression, and convolutional neural network (CNN) for gold price data for the period January 2, 1979 to July 31, 2020. Another study by Yurtsever [53] with multivariate gold data for the period 2001–2021 also shows that the use of LSTM is better than bidirectional–LSTM and gated recurrent unit (GRU). However, the two studies did not use cross validation so that a study which combines the LSTM hyperparameter and walk forward validation is needed in forecasting gold prices. The LSTM model is applied to gold price research because it has been proven to be one of the superior algorithms in forecasting. This is expected to assist investors in determining the investment strategy to be carried out.

2 Related Works

2.1 Long short-term memory (LSTM)

Long short-term memory (LSTM) is a method of developing a recurrent neural network (RNN) type neural network, first introduced by Hochreiter and Schmidhuber [23] in 1997. The development of RNN in LSTM lies in the special gate mechanism for controlling memory cell access. Memory cells as information storage areas will get information selected by the gate mechanism (Hochreiter and Schmidhuber [23] in Manowska [36]). According to Kalchbrenner et al. [25], the gate mechanism in the LSTM is composed of three vector gate units, the input gate, the forget gate, and the output gate. Each vector gate unit has a different function. The input vector gate functions to control the number of input vectors that will affect the memory. The forget vector gate controls the amount of old memory to be deleted. The output vector gate controls the amount of memory stored in the hidden state.

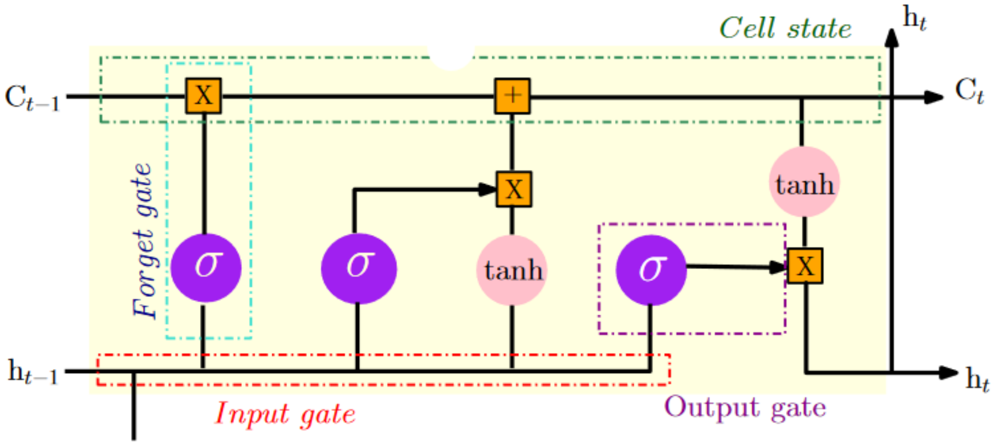


Figure 1: Example of an LSTM architecture [34].

Information :
x_t = t-time input value;
h_t = t-time output value;
c_{t-1}, c_t = cell state time-(t-1) and t.

LSTM comprises several general stages, as illustrated in Figure 1. Each gate has an activation vector that will process values x_t and h_{t-1} with a *sigmoid* activation function that produces values x in the range 0 to 1 and a *tanh* activation function that has values x in the range -1 to 1. Based on Sarangapani [46], the formula for the *tanh* activation function and *sigmoid*, respectively, can be seen in Equations (1) and (2):

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{1}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{2}$$

where x is the input value and e is the exponential value.

In addition, weight matrices (W) and bias vector parameters (b) for each gate will be studied during training. According to Goyal et al. [19], the first stage begins with determining the information to be deleted by the memory cell block at the forget gate (f_t) with the *sigmoid* activation function following Equation (3):

$$f_t = \sigma(W_f \cdot [h_{t-1},x_t] + b_f). \tag{3}$$

The second stage is determining the information stored in the cell state. This stage consists of two parts; the first is updating the value by the input gate (i_t) with the *sigmoid* activation function as Equation (4), and the second is the creation of a new value candidate vector (c_t) with the *tanh* activation function following Equation (5):

$$i_t = \sigma(W_i \cdot [h_{t-1},x_t] + b_i), \tag{4}$$

$$\tilde{c} = tanh(W_c \cdot [h_{t-1},x_t] + b_c). \tag{5}$$

The third stage is updating the old cell state into the new cell state following Equation (6):

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}. \tag{6}$$

The final stage in LSTM is determining the output by the output gate (o_t) by following Equation (7) which is based on the results of updating the cell state using the *sigmoid* activation function. The results of Equation (7) will then be combined with the *tanh* activation function to produce the output (h_t) as shown by Equation (8):

$$o_t = \sigma(W_o \cdot [h_{t-1},x_t] + b_o), \tag{7}$$

$$h_t = o_t \times tanh(c_t). \tag{8}$$

In addition to a gate mechanism, the LSTM modeling process can be formed using several hyperparameters. Hyperparameters are elements in deep learning that are not updated during the training process but can affect the algorithm's accuracy [21]. The LSTM hyperparameters commonly used include learning rate and epoch. The learning rate controls changes in the weights that are updated during the training process, while the epoch indicates the number of algorithms executed for the entire data [6][10]. However, other hyperparameters can still be used, namely optimizer. The optimizer is an optimization method for increasing accuracy in machine learning models. The optimizers commonly used in LSTM include adaptive momentum (Adam) and root mean square propagation (RMSProp). Adam is a free stochastic gradient optimization method by Kingma and Ba [27]. Adam stochastically determines the step size and learning rate using random probability distribution sampling. At the same time, RMSProp is a technique for reducing noise in a neural network by smoothing errors as they propagate through the network [37]. RMSProp addresses the disappearing gradient problem using a squared gradient moving average to normalize the gradient. This normalization balances the step size (momentum), reducing the step for large gradients so they do not explode and increasing the step for small gradients so they do not disappear [45].

2.2 Walk forward validation

Walk forward validation is the most common method of backtesting cross-validation, a form of time series cross-validation method to measure the goodness of a strategy or model that generally uses past observations without any retraining. According to Ostermann et al. [40], walk-forward validation can be done with two approaches: sliding windows and expanding windows. Sliding windows use a constant window shift inside and outside the sample data, as illustrated in Figure 2(a). The window is the part that contains all of the training data and test data. The size of the training data, test data, and window shifts are kept the same so that several training-test pairs are produced [13]. Meanwhile, expanding windows adds observations from the test dataset to expand the training size from an initial size to a maximum size, as illustrated in Figure 2(b).

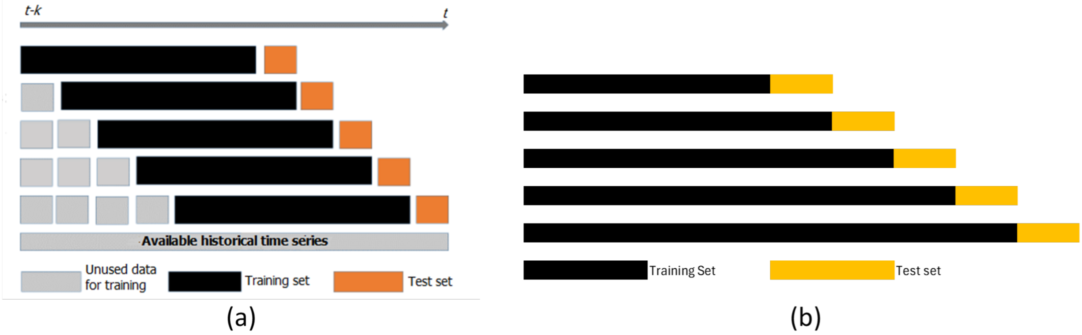


Figure 2: Walk-forward validation illustration: (**a**) sliding window (modified from Khalid et al. [35]); (**b**) expanding window.

2.3 Accuracy size

Accuracy is one of the important measurements in determining the model's goodness. Forecast accuracy can only be determined by considering how well the model performs on new data not used during model building [24]. Accuracy measures generally used for forecasting include root mean square error (RMSE) and mean absolute percentage error (MAPE). RMSE is a model evaluation method by calculating the average value of the sum of the squared errors. A low RMSE value indicates that the accuracy is improving with the variety of values produced by the model approaching the diversity of the original values [22]. MAPE is a measure of accuracy that calculates the difference between actual and estimated data in percentage form [52]. The RMSE and MAPE values, according to Poon [42], can be calculated sequentially using Equations (9) and (10), and the classification of MAPE values, according to Lewis [28], is shown in Table 1:

$$RMSE = \sum_{t=1}^{n} \left(\frac{(x_t - \hat{x}_t)^2}{n} \right)^{\frac{1}{2}}, \tag{9}$$

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{x_t - \hat{x}_t}{x_t} \right| \times 100\%, \tag{10}$$

where x_t is t-time current input value, $\hat{x}_t$ is t-time forecast value, and n is number of forecast periods.

Table 1: Classification of MAPE values by Lewis [28].

MAPE(%)	Forecasting Power
<10	High
10–20	Good
20–50	Reasonable
>50	Inaccurate

3 Materials and Methods

3.1 Data

Secondary data is used in the study which is obtained from the World Gold Council. The data is daily gold price data for five working days (Monday to Friday). It consists of 5,478 data with the period used from 1 January 2003 to 31 December 2023. Data obtained through the site https://www.gold.org/goldhub/data/gold-prices.

3.2 Data analysis procedures

The stages of analysis carried out in this study are as follows.

1. Perform data preprocessing.

   It is done by viewing and changing the input data type to suit the application programming language and performing data imputation. The gold data used in this study is daily data on weekdays, so it is necessary to impute it to fill in data values on weekends and holidays. The imputation method to be used is linear interpolation. According to Chapra [11], linear interpolation is interpolation to estimate the value of two known values. The linear interpolation formula can be seen in Equation (11):

   $$f_1(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{(x_2 - x_1)}(x - x_1). \tag{11}$$

   Information :
   x_1 = time period before missing data;
   x_2 = time period after missing data;
   x = time period of missing data;
   $f(x_1)$ = data value one period before missing data;
   $f(x_2)$ = data value one period after missing data;
   $f_1(x)$ = missing data values to be imputed.

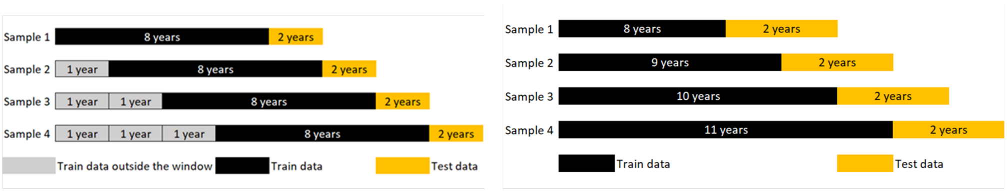
2. Exploring the data to understand and obtain the characteristics of the data, as well as evaluating the autocorrelation function (ACF) plot and Box-Cox plot to see the stationarity of the data. Data is stationary if the ACF plot decreases rapidly and has a rounded value (λ) equal to one or a value of one in the confidence interval of the Box-Cox plot [51].

3. Create a walk-forward validation scenario.

   Walk-forward validation begins with sample preparation. Samples from sliding windows walk-forward validation are obtained by first determining the length of the training data, test data, and the window shift distance. The length for both data and shift distances are determined subjectively with a training data length of eight years, a testing data length of two years, and an annual shift distance. Next, for expanding windows walk-forward validation, the initial size for training data length is eight years, testing data length is two years, and expanding training is each year. Each walk-forward validation approach has an equal sample, as many as 12 samples. The final sample formed is shown in Figure 3 and more clearly, the division of data in the Table 2. The end of observation does not end on December 31, 2023, because the length of the data used in this study uses the multiplication of seven days a week by 52 weeks a year, so there are remaining days that cannot be used as a sample.

Table 2: Period of training data and test data for each sample.

No.Sample	Sliding Window		Expanding Window	
	Train	Test	Train	Test
1	01-Jan-2003 to 21-Dec-2010	22-Dec-2010 to 18-Dec-2012	01-Jan-2003 to 21-Dec-2010	22-Dec-2010 to 18-Dec-2012
2	31-Dec-2003 to 20-Dec-2011	21-Dec-2011 to 17-Dec-2013	01-Jan-2003 to 20-Dec-2011	21-Dec-2011 to 17-Dec-2013
3	29-Dec-2004 to 18-Dec-2012	19-Dec-2012 to 16-Dec-2014	01-Jan-2003 to 18-Dec-2012	19-Dec-2012 to 16-Dec-2014
4	28-Dec-2005 to 17-Dec-2013	18-Dec-2013 to 15-Dec-2015	01-Jan-2003 to 17-Dec-2013	18-Dec-2013 to 15-Dec-2015
5	27-Dec-2006 to 16-Dec-2014	17-Dec-2014 to 13-Dec-2016	01-Jan-2003 to 16-Dec-2014	17-Dec-2014 to 13-Dec-2016
6	26-Dec-2007 to 15-Dec-2015	16-Dec-2015 to 12-Dec-2017	01-Jan-2003 to 15-Dec-2015	16-Dec-2015 to 12-Dec-2017
7	24-Dec-2008 to 13-Dec-2016	14-Dec-2016 to 11-Dec-2018	01-Jan-2003 to 13-Dec-2016	14-Dec-2016 to 11-Dec-2018
8	23-Dec-2009 to 12-Dec-2017	13-Dec-2017 to 10-Dec-2019	01-Jan-2003 to 12-Dec-2017	13-Dec-2017 to 10-Dec-2019
9	22-Dec-2010 to 11-Dec-2018	12-Dec-2018 to 08-Dec-2020	01-Jan-2003 to 11-Dec-2018	12-Dec-2018 to 08-Dec-2020
10	21-Dec-2011 to 10-Dec-2019	11-Dec-2019 to 07-Dec-2021	01-Jan-2003 to 10-Dec-2019	11-Dec-2019 to 07-Dec-2021
11	19-Dec-2012 to 08-Dec-2020	09-Dec-2020 to 06-Dec-2022	01-Jan-2003 to 08-Dec-2020	09-Dec-2020 to 06-Dec-2022
12	18-Dec-2013 to 07-Dec-2021	08-Dec-2021 to 05-Dec-2023	01-Jan-2003 to 07-Dec-2021	08-Dec-2021 to 05-Dec-2023



(a) (b)

Figure 3: Illustration of walk-forward validation with (**a**) sliding window and (**b**) expanding window approach.

4. Doing modeling with LSTM for one sample walk-forward validation.

   (a) Before modeling, pre-processing is carried out by combining training data and test data from point (3). Next, standardize the data because LSTM works better with centralized and scaled data [26]. Standardization will cause the data to have a mean of zero and a variance of one. According to Schmuller [47], standardization is carried out by following Equation (12):

$$x_t^* = \frac{x_t - \bar{x}}{\sigma_x}, \tag{12}$$

where x_t^* is t-time standardized input value, x_t is t-time input value, $\bar{x}$ is the average data value, and σ_x is the standard deviation of x.

(b) Initialize the LSTM hyperparameters to be processed.
Referring to Alhamdani [4] and Yurtsever [53], the learning rate values to be used include 0.01; 0.001; and 0.0001; and epochs with values 100, 500, and 1000. In addition, Adam optimizer and RMSProp are used. All hyperparameters are then combined with the sliding and expanding windows scenario, as shown in Table 3. Each walk-forward scenario will get 18 hyperparameter combinations, as shown in Table 3 for the sliding and expanding window scenario. The total combinations produced are 36. However, the architecture and other hyperparameters in the LSTM are determined following the running data analysis procedure.

Table 3: Formation of a walk–forward validation sample.

No.	**Walk-Forward Scenario**	**Hyperparameters**		
		Optimizer	**Learning Rate**	**Epoch**
1.	Sliding Window / Expanding Window	Adam	0.01	100
2.				500
3.				1000
4.			0.001	100
5.				500
6.				1000
7.			0.0001	100
8.				500
9.				1000
10.		RMSProp	0.01	100
11.				500
12.				1000
13.			0.001	100
14.				500
15.				1000
16.			0.0001	100
17.				500
18.				1000

(c) Conduct training and fit the LSTM model for one sample walk-forward validation.

(d) Evaluate the model with RMSE.

5. Repeat step (4) for all samples from each walk-forward validation approach.

6. Calculates the RMSE mean value of each hyperparameter combination for each scenario and retrieves the best hyperparameters.

7. Handling underfitting and overfitting problems often encountered in using LSTM with early stopping. Early stopping will stop the process more quickly when there is a decrease in validation and maintain optimal conditions without running the entire epoch [29][5].

8. Validate the best hyperparameter combination in point (6) for all data and calculate the MAPE value. Each hyperparameter in point (6) is combined again to see the possibility of another hyperparameter combination that produces the best forecast outside the combination of point (6). In this step, all data will again be divided into training and test data using proportions that refer to data patterns. Then, the Diebold-Mariano (DM) test will be used to test the significant differences in accuracy using forecast errors [14] [41]. The alternative hypothesis used in the Diebold-Mariano test is that models with higher accuracy values are better than those with lower ones. Next, the residuals from the model will be subjected to diagnostic tests.

9. Do forecasting with all data with the best hyperparameter point (8), which has the smallest MAPE value.

4 Results

4.1 Preprocessing and data exploration

The data preprocessing begins by changing the data format. The data format for the date variable, previously of type POSIXtc, is converted to datetime so that it can be read and processed according to its data type using the programming language. Next, the data preprocessing imputes data for values on price variables on weekends (Saturday-Sunday) and holidays with linear interpolation. The amount of data that previously consisted of 5,478 data increased to 7,670 after imputation.

After preprocessing the data, data exploration is carried out. Data exploration begins by looking at the descriptive statistics of the data and obtaining an average daily gold price of 1,197.1680 USD/troy ounce with a spread of (standard deviation) 480.4087 from the average. The standard deviation value, which is quite large, indicates that the spread of the data to the average is also large, as seen in Figure 4. The highest gold price data reached 2,078.4 USD/troy ounce on December 28, 2023 (green dot), while the lowest gold price touched 319.9 USD/troy ounce on April 7, 2003 (red dot).

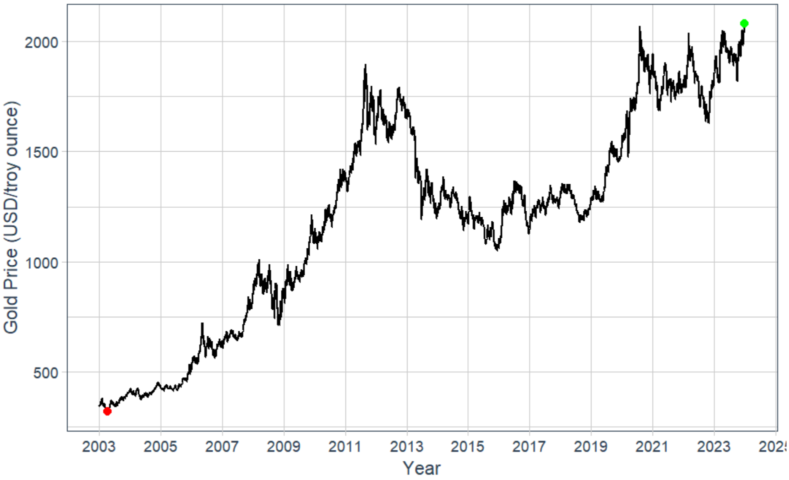
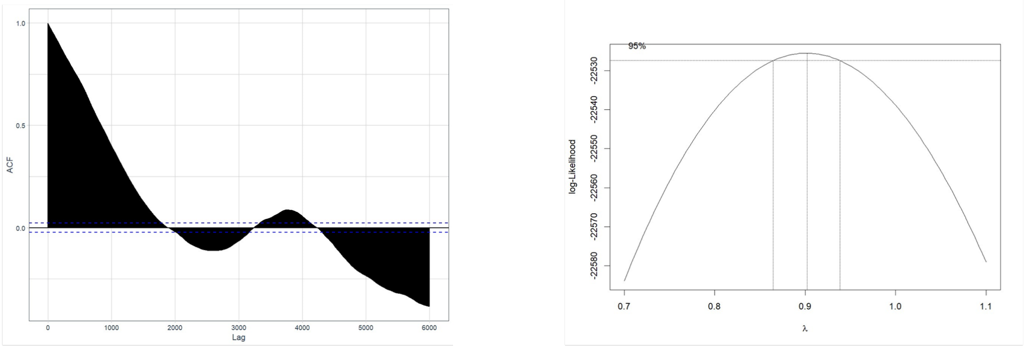


Figure 4: Gold prices for the period 2003-2023.

In addition to descriptive statistics, exploration is done by creating time series plots. Figure 4 plots daily gold price time series data with the x-axis as the period and the y-axis as the gold price per troy ounce in USD. Figure 4 shows that the data has a trend from the beginning to the end of the period. The price of gold had an extremely positive trend around 2008 to mid-2012. The global financial crisis caused this, so investors prefer to invest in gold as a safe asset. Then, there was a negative trend from 2012 to 2013 due to economic recovery, and it fluctuated for the next period until the end of 2019. However, these fluctuations still showed stability. The gold price then experienced an extreme increase again in 2020 until mid-2021. The highest spike in the daily gold price occurred in mid-2020. This was caused by the effects of the Covid–19 pandemic, so investors re-switch to stable investment instruments such as gold. After the situation improved enough and vaccinations began to spread, the global economy also recovered so that investors turned back to investing in high-risk instruments, which had an impact on the price of gold, which seemed to decline. However, after that, the price of gold remained stable despite fluctuating.

4.2 Data stationery

The fluctuating price of gold, as shown in Figure 4, indicates that the data is not stationary on the average because the data does not spread around the average value. Figure 5(a) shows the ACF plot, which seems to decrease slowly and is outside the interval of the standard error value (blue dotted line). Therefore, it can be identified that the world gold price data is not stationary on average. Apart from not being stationary on average, the gold price data is also suspected to be non-stationary on the range because it looks disproportionately widening and narrowing. Figure 5(b) shows a rounded value (λ) of 0.9020, and at a 95% confidence interval, the value of λ has a lower limit of 0.8657 and an upper limit of 0.9383. The interval does not contain a value of one, so the closing gold price data is not stationary in variance.



(a) (b)

Figure 5: ACF plot (**a**) and Box-Cox plot (**b**).

According to Lin and Feng [31], using deep learning methods, especially neural networks, does not require stationary assumptions because they can handle nonlinear relationships and large dimensions. In addition, Long et al. [33] mention LSTM as an algorithm that can overcome linearity obstacles in the autoregressive-moving average (ARMA) model and stationary assumptions. Therefore, gold price data that is not stationary regarding mean and variance does not need to be handled.

4.3 LSTM model evaluation walk forward validation results

Model evaluation begins with building the LSTM model used in the data processing. LSTM is built with an architecture in the form of layers and initialization of hyperparameters such as batch size, epoch, and learning rate. In addition to the epoch value and learning rate, this study initializes the hyperparameters randomly based on experiments. The LSTM model is built with two LSTM layers, each consisting of 50 neuron units and one dense layer. The optimizers used are Adam and RMSProp. According to Mehmood et al. [37], Adam is an efficient optimizer, while RMSProp can better deal with noise. Furthermore, the batch size is determined by taking the common factor of the amount of data between the training data and the test data. In this study, the batch size used was 56.

Model evaluation is obtained from the average RMSE value of all samples in each model combination with two kinds of walk-forward scenarios. Table 4 and Figure 6 show the difference in walk-forward approaches in which the sliding window has a bigger average RMSE than the expanding window. This is due to the fact that the training data information continues to grow as the data period increases, while the length of the test data is always kept fixed. The model with the smallest RMSE value is the best, as shown in Table 4. Evaluation of the LSTM model from two scenarios walk-forward shows the same hyperparameter values in getting the smallest RMSE value. Using the RMSProp optimizer produces a smaller RMSE average and standard deviation. In addition, Figure 6 shows there are some trends, such as the use of the RMSProp optimizer being more stable than Adam and small epochs being better than large epochs. However, the combination of hyperparameters also determines the model's goodness. The Adam optimizer cannot produce output results at a combination of 1000 epochs with a learning rate of 0.0001. This can indicate that the combination produces divergence, especially when using a small learning rate [30].

Table 4: Comparison of the best hyperparameters for each walk-forward scenario.

Walk-Forward	**Hyperparameters**			**RMSE**	
Scenario	**Optimizer**	**Learning Rate**	**Epoch**	**Mean**	**Standard Deviation**
Sliding Window	Adam	0.01	100	30.3908	34.6692
	RMSProp	0.001	500	18.6431	12.5548
Expanding Window	Adam	0.01	100	12.8135	4.4375
	RMSProp	0.001	500	11.8783	3.7719

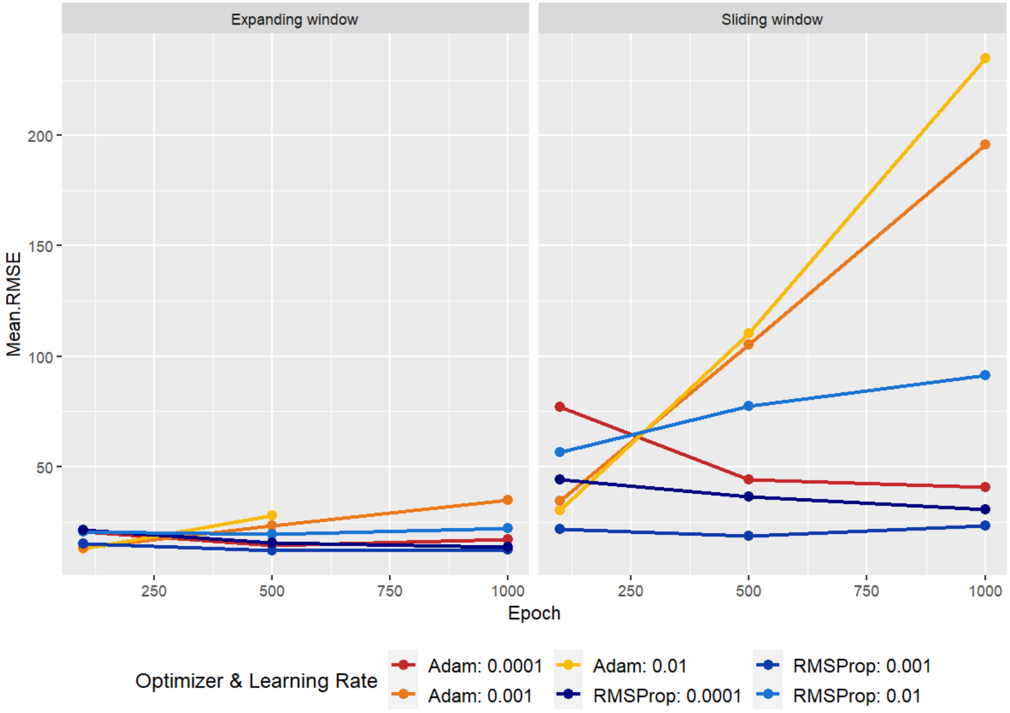


Figure 6: Average RMSE hyperparameter combination results for each walk-forward approach.

Figure 6 shows that the RMSProp optimizer produces a smaller RMSE average if combined with neither too big nor too small learning rate and epoch. Meanwhile, the Adam optimizer produces a smaller RMSE average value if combining a small learning rate with big epoch or big learning rate with small epoch. However, the epoch on RMSProp did not show a tendency to increase or decrease the average RMSE value. These two results are by Guha's research [20] with MNIST and CIFAR-10 data to see the effect of learning rates on various optimizers. Research with MNIST data shows RMSProp and Adam working better with a learning rate of 0.01, 0.001, and 0.003, and using different epochs on the RMSProp optimizer shows that the accuracy tends to be stable.

Figure 6 shows that many combination especially in the use of sliding window produces a fairly high RMSE accuracy value. Underfitting is a condition when the model is unable to get a low enough error value in the training process so that it does not represent a complete picture of the data pattern, while overfitting is a condition when the model has poor performance on invisible data but is good in the training process [17] [38]. Following Liang and Cai [29] in their study of standard monthly loan rates on American peer-to-peer (P2P) lending platforms, the problem of underfitting LSTM can be handled by adjusting the value of the learning rate. The learning rate value will be adjusted randomly from the learning rate value that produces the smallest RMSE value. Furthermore, the problem of overfitting LSTM can be handled by using early stopping.

Table 5: Evaluate the best hyperparameter model using early stopping.

Walk-Forward	**Hyperparameters**			**RMSE**	
Scenario	**Optimizer**	**Learning Rate**	**Epoch**	**Mean**	**Standard Deviation**
Sliding Window	Adam	0.01	100	23.1508	14.6143
	RMSProp	0.001	500	25.3310	15.7483
Expanding Window	Adam	0.01	500	12.8617	4.6503
	RMSProp	0.001	500	12.8293	4.7589

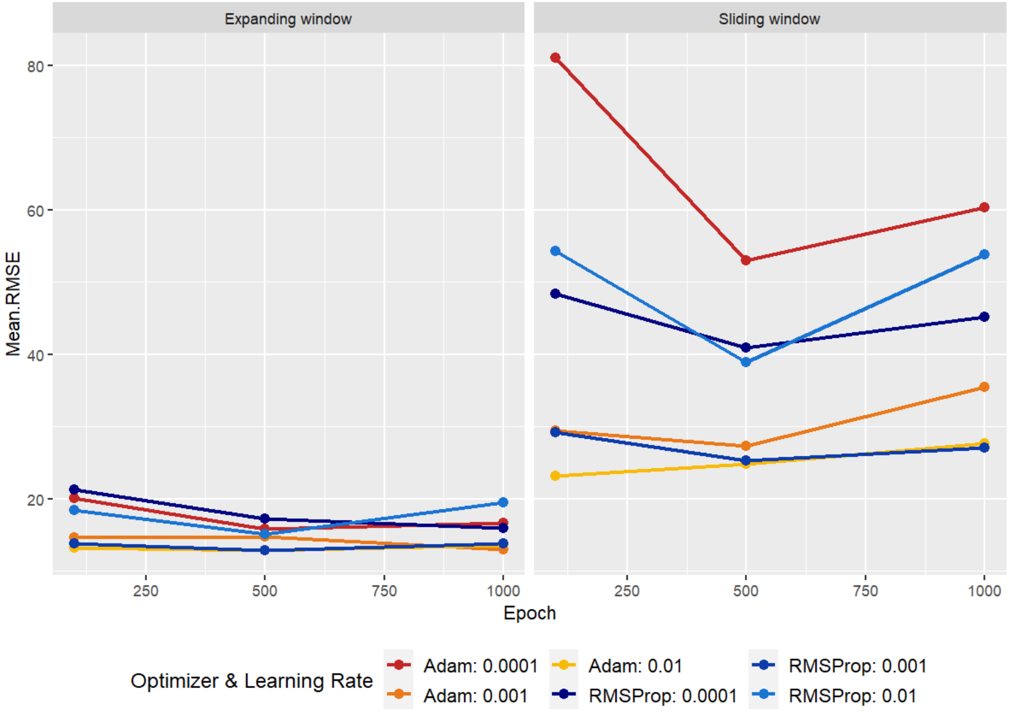


Figure 7: Average RMSE hyperparameter combination results using early stopping.

Table 6 and the y-axis from Figure 7 show that early stopping can reduce the sliding window scenarios' average and standard deviation RMSE value. The successful use of early stopping is combined with a random adjustment of the learning rate around the best value. The results are that the average RMSE value tends to produce random values and is greater than the optimum learning rate in Table 6, even produce NaN values. The optimal learning rate value according to initial initialization is in line with the statement of Goodfellow et al. [17], who stated that the search for the optimum learning rate involves values with a logarithmic function scale (0.1, 0.01, 0.001, 0.0001, 0.00001). Therefore, the hyperparameter values for the best models in Table 5 will still be used at the validation stage.

Table 6: Evaluate the model by adjusting the learning rate and using early stopping.

Walk-Forward	Hyperparameters			RMSE	
Scenario	Optimizer	Epoch	Learning Rate	Mean	Standard Deviation
Sliding Window	Adam	100	0.03	90.1299	102.5337
			0.05	150.4352	160.0882
			0.08	NaN	
	RMSProp	500	0.003	28.5388	17.1501
			0.005	27.7234	16.7459
			0.008	37.1435	32.7617
Expanding Window	Adam	500	0.03	45.5488	98.6409
			0.05	86.8005	169.0489
			0.08	NaN	
	RMSProp	500	0.003	15.5992	7.5193
			0.005	16.0798	8.7189
			0.008	19.7050	19.7704

4.4 Model validation

The best LSTM model with the obtained hyperparameters is then validated using the entire data. The data is again divided into training data and test data with a ratio of 60:40. This comparison was chosen because it can be seen that the distribution of training data and test data has similar data patterns compared to the distribution using other proportion values. The learning rate and epoch values will be combined based on the optimizer to see whether other combinations outside the values of Table 5 produce better accuracy. Each model that is formed is calculated by MAPE value to see the level of accuracy of the model that RMSE cannot classify. A comparison of the MAPE values for each hyperparameter combination is shown in Table 7.

Table 7: Comparison of MAPE between validation models.

Model	Optimizer	Learning Rate	Epoch	MAPE (%)
1	Adam	0.01	100	0.4867
2	Adam	0.01	500	0.5479
3	RMSProp	0.001	100	0.7244
4	RMSProp	0.001	500	0.6244

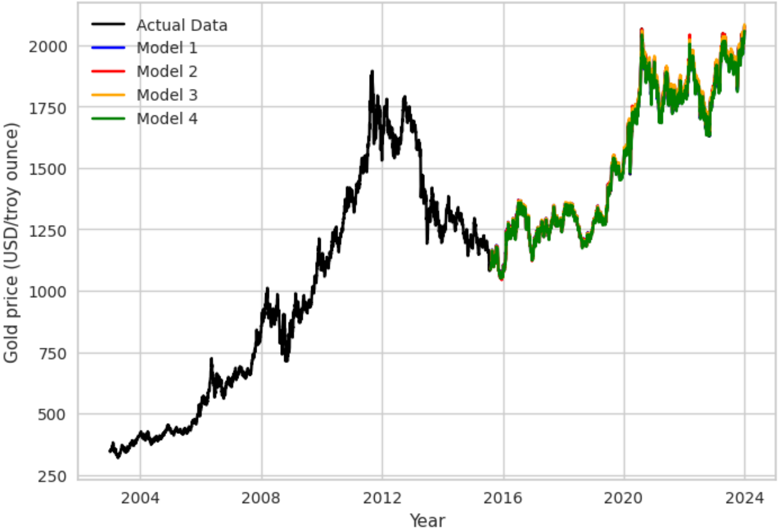


Figure 8: Results of validation of all model.

Table 7 demonstrated the validation results that a learning rate of 0.01 produces better forecasting accuracy than a learning rate of 0.001. Almost all uses of a learning rate of 0.01 produce forecasting accuracy, which is classified as a high forecasting power according to Table 1. A learning rate of 0.001 is also classified as a high forecasting power. Figure 8 shows forecast results from all models in the validation process that resemble the original data pattern. These results make it difficult to select the best model, so the Diebold-Mariano test was conducted using a 5% significance level. The null hypothesis used is like in Table 8 against the alternative hypothesis that the first model is better than the second model. The results show that Model 1 is significantly different from Model 3 and Model 4. Model 2 is also significantly different from Model 3. This means the null hypothesis is rejected, and the alternative hypothesis is accepted. Based on the amount of significance against other models, Model 1 is the best and will be tested for the diagnostic of the residuals, including tests for normality, autocorrelation, and heteroscedasticity.

Table 8: Diebold-Mariano test between all validation models.

Null Hypothesis	DM-test Statistic	P-value	
Model 1 $\leq$ Model 2	-1.2816	0.1000	
Model 1 $\leq$ Model 3	-2.4575	0.0070	**
Model 1 $\leq$ Model 4	-3.3451	0.0004	***
Model 2 $\leq$ Model 3	-2.9159	0.0018	**
Model 2 $\leq$ Model 4	-1.5520	0.0603	.
Model 4 $\leq$ Model 3	1.4889	0.9317	

Significance codes:***: 0.001, **: 0.01, *: 0.05, .: 0.1.

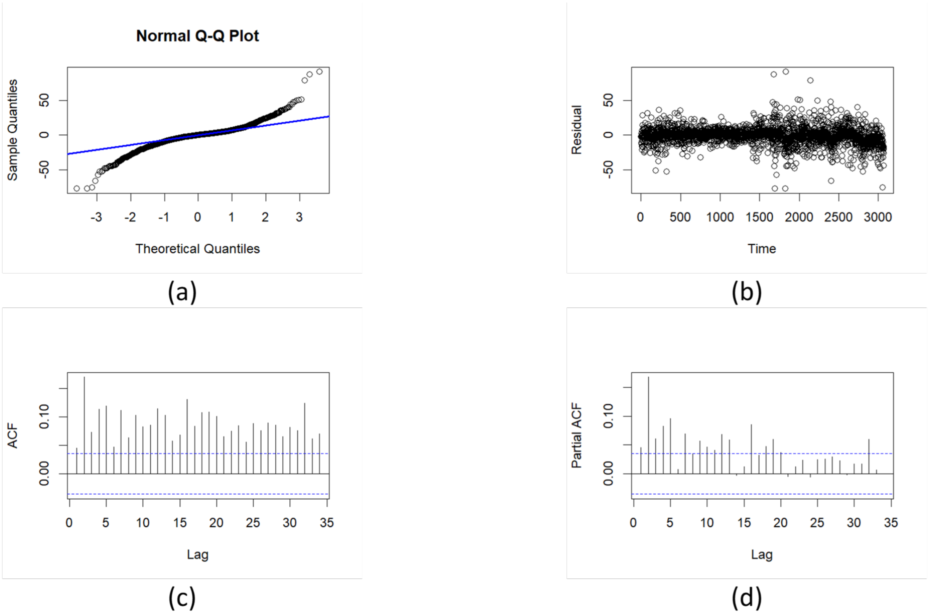


Figure 9: Explorative diagnostics test: (**a**) normal quantile-quantile plot, (**b**) time series plot, (**c**) ACF plot, and (**d**) PACF plot of the residuals of LSTM Model 1.

Model diagnostic tests, both exploratory and formal tests, were performed on the residuals of Model 1. It can be seen in Figure 9 that all assumptions are violated. First, normal quantile-quantile plots were used to test the normality of the residuals, showing that the residuals did not follow the normal line. This is also supported by a formal test with the Anderson-Darling test, resulting in a p-value of 0.0000, less than the 5% significant level. This means the result rejects the null hypothesis (the means follow a normal distribution). Then, the residual plot to check the heterogeneity test of variance shows it has different bandwidths over several periods, so the model tends to be heterogeneous. The Ljung-Box test of squared residuals also supports the results, which have a p-value of 0.0000, less than the 5% significant level, which means that the result rejects the null hypothesis (homogeneous variance). Last, for autocorrelation, ACF and partial ACF (PACF) show that the first 35 lags have significant autocorrelation. The Ljung-Box test of residuals also supports the results, which has a p-value of 0.0119, less than the 5% significant level, and it means that the result rejects the null hypothesis (no autocorrelation in the time series). Even though it violates all assumptions, according to de Guia et al. [12], neural networks (where LSTM is included) can assume any function, even for complicated patterns, so they have no assumptions about data, errors, or targets. Therefore, violations of this assumption will not be addressed.

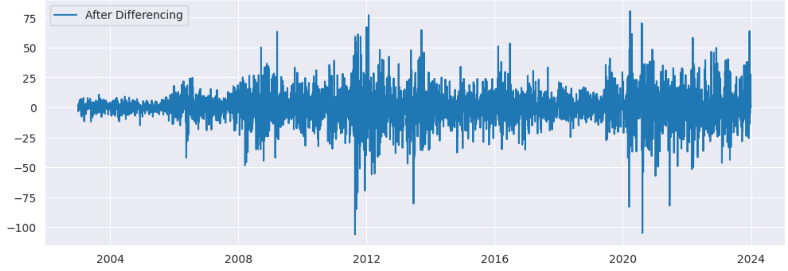
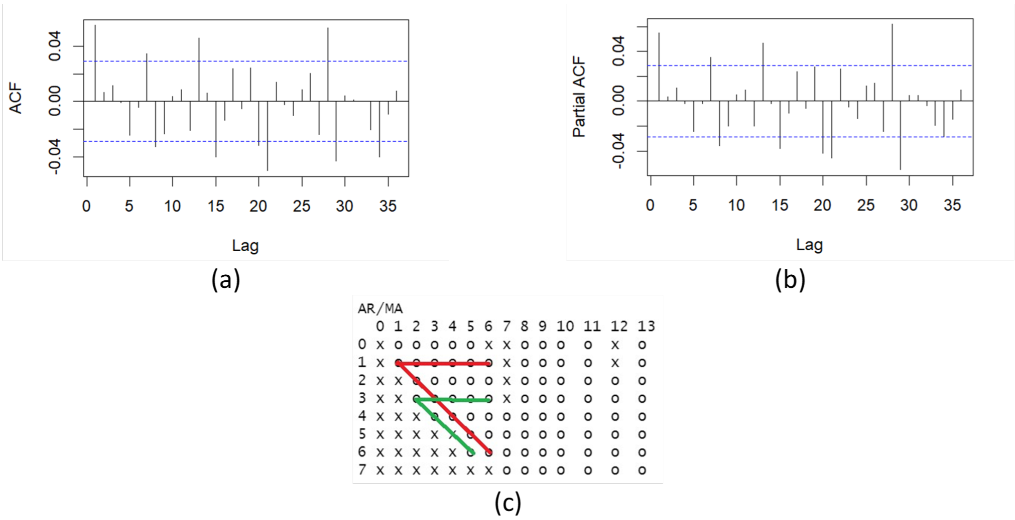


Figure 10: Gold price time series after differencing.

As a benchmarking, the conventional modelling method, Box-Jenkins ARIMA, was also tried with the same data conditions. Box-Jenkins ARIMA was chosen because, according to Ramli et al. [44], the ARIMA model can model non-stationary data with simple differencing, i.e., one or two levels of differencing. Based on data exploration, it was found that the data was not stationary, so one-time differencing was performed. Figure 10 show the result was that the data was stationary in the mean. Moreover, this is reinforced by the formal Augmented Dickey-Fuller test, which yields a p-value of 0.01, thus rejecting the null hypothesis (data is not stationary).



```
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x o o o o o x x o o o  o  x  o
1 x o o o o o o x o o o  o  x  o
2 x x o o o o o x o o o  o  o  o
3 x x o o o o o x o o o  o  o  o
4 x x x o o o o o o o o  o  o  o
5 x x x x x o o o o o o  o  o  o
6 x x x x x o o o o o o  o  o  o
7 x x x x x x x o o o o  o  o  o
```

Figure 11: (**a**) ACF plot, (**b**) PACF plot, and (**c**) EACF plot.

Table 9: Parameter estimation of ARIMA(3,1,2).

Model	Parameter	Coefficient	P-value	AIC
ARIMA(3,1,2)	AR(1)	-0.1994	0.0006	34,191.47
	AR(2)	-0.8912	0.0000	
	AR(3)	0.0724	0.0000	
	MA(1)	0.2574	0.0000	
	MA(2)	0.9103	0.0000	

Furthermore, model identification was carried out using ACF, PACF, and extended ACF (EACF) plots as shown in Figure 11, and the estimation of model parameters was continued. Based on the ACF plot, it can be identified that the model formed is ARIMA(0,1,1), based on the PACF plot, it can be identified that the model formed is ARIMA(1,1,0), and based on EACF plot it can be identified that the model formed is ARIMA(1,1,1) and ARIMA(3,1,2). Next, parameter estimation is carried out for the entire identified model. As a result, ARIMA(3,1,2) is the best model based on the smallest AIC value and significant parameters. The estimated parameter values are shown in Table 9.

Exploratively from Figure 12, all showed assumption violations. However, with formal tests, the Anderson-Darling normality test for the residuals and the Ljung-Box variance test on the squared residuals of the model yielded a p-value of 0.0000. In contrast, the Ljung-Box autocorrelation test on the residuals yielded a p-value of 0.8471, thus not rejecting the null hypothesis. As a result, the assumptions of normality and homogeneity of variance are violated. Without ad-

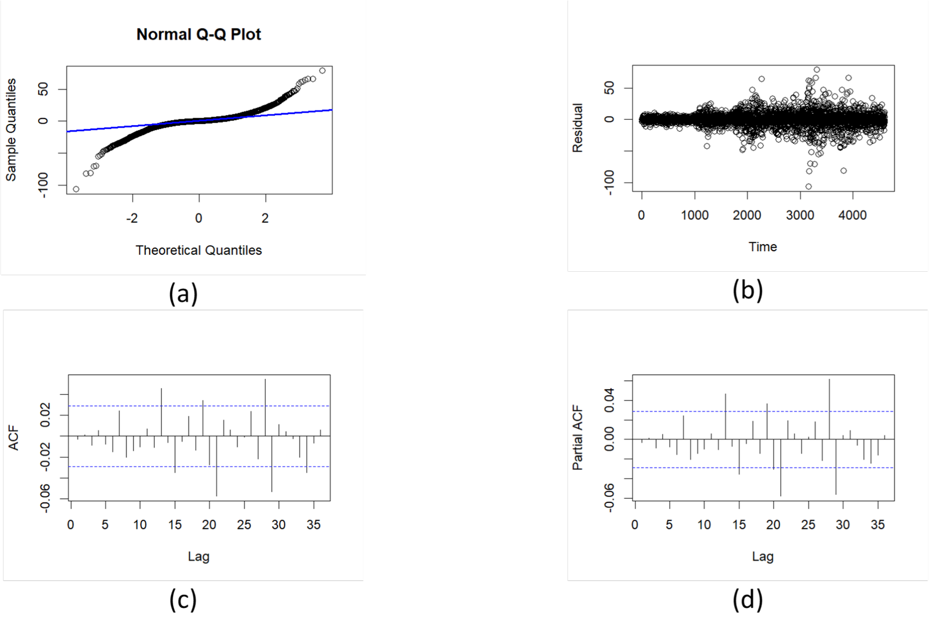


Figure 12: Explorative diagnostics test: (**a**) normal quantile-quantile plot, (**b**) time series plot, (**c**) ACF plot, and (**d**) PACF plot of the residuals of ARIMA(3,1,2).

dressing this, the modelling is continued with overfitting, which is to increase the order p and q of the initial model to find the better model, and the results are not better, so the ARIMA(3,1,2) model is still used. Figure 13 depicts the forecast process, and validation results were obtained with a MAPE of 26.2920%, which was classified as reasonable forecasting. This result shows that LSTM is better than ARIMA, especially in long-term data validation.

4.5 Forecasting

The best model validation results, Model 1, are used to make forecasting. The forecasting in question predicts the value of data after the last data. In this study, the latest data was from December 31, 2023. Data values after December 31, 2023, will be predicted for the next 3,068 days, according to the amount of test data in the validation process, or around eight years. Forecasting values show that the price of gold, which begins with a decline, will continue to increase in the long term, although it appears to decline around 2028-2029, as shown in Figure 14. The price of gold is predicted to increase by around 700 USD/troy ounce in seven years after decline in one year, with a sharp increase occurring between 2024 and 2026. The average forecast for gold prices in 2024, it is 1,321.10 USD/troy ounce; in 2025, it was 1,925.97 USD/troy ounce; in 2026, it was 2,046.86 USD/troy ounce; in 2027 it was 2,075.46 USD/troy ounce; in 2028 it was 1,910.40 USD/troy ounce; in 2029 it will be 1,893.86 USD/troy ounce; in 2030 it will be 1,997.59 USD/troy ounce, and in 2031 it will be 2,079.32 USD/troy ounce.

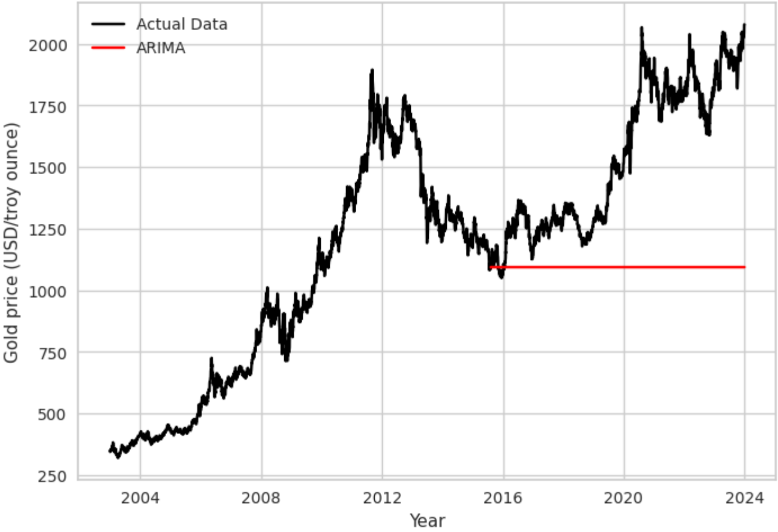


Figure 13: Validation forecast model of ARIMA(3,1,2).

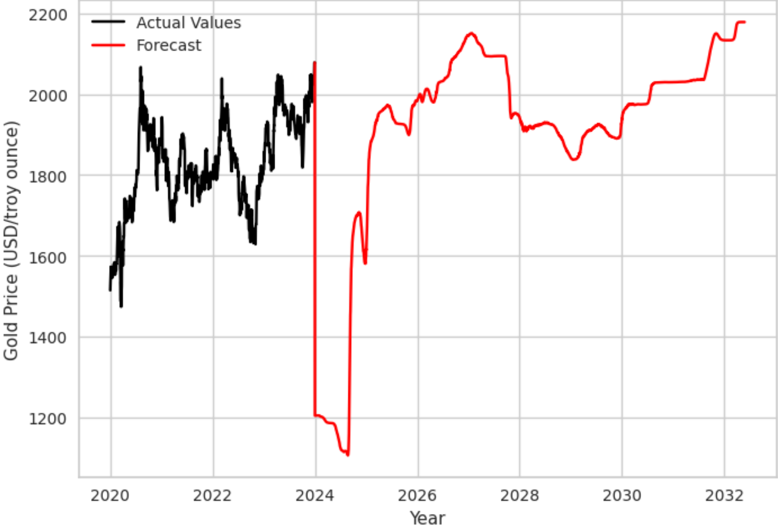


Figure 14: Forecast results with Model 1.

According to the World Bank [8], the price of gold is predicted to decline due to the global economic recovery and reduced inflationary pressure. Furthermore, in the long term, it is stated that inflation and interest rates are the main factors for gold price movements. In contrast, the volatility of gold prices in the short term will likely continue, given the increasing geopolitical and economic uncertainties. The forecasting results that researchers do are different from those of the World Bank, so the value of these results may need to be corrected and appropriate. However, these results still follow the statement of Soelistijo et al. [48], who stated that the value of gold would continue to strengthen in the long term. Gold has a special place in world trade, and investment dues have a value never eroded by inflation. The condition of increasing gold prices can be used as an option for saving/investing. Meanwhile, gold purchases can be made when needed [7].

5 Conclusions

The daily gold price, which is not stationary in mean and variance, can be modeled with LSTM. The walk-forward validation process helps determine the best hyperparameters from various scenarios. The best LSTM model obtained results from a combination of hyperparameter optimizer Adam, learning rate 0.01, and epoch 100. This model has been able to follow the actual value pattern and produces an accuracy of 0.4867% MAPE value in the validation process, which can be classified as having high forecasting power. Forecast results with the best model show that gold prices tend to increase in the next eight years. The condition of increasing gold prices can be used as the right time to invest.

This study applies the method with univariate data that still ignores classical statistical forecasting assumptions. Further research can add covariates to the model and handle assumption violations, such as heteroscedasticity problems. Heteroscedasticity of gold data can be done with the Box-Cox transformation process, as done by Gopal et al. [18] with Malaysian gold price data. In addition, the LSTM method can be combined with classical statistical methods (ARIMA) or other deep learning methods, such as artificial neural networks, CNN, or GRU, to find a better model.

Acknowledgement We would like to thank Department of Statistics, IPB University, for all the support and facilities provided.

Conflicts of Interest All authors declare no conflict of interest.

References

[1] H. Abbasimehr, M. Shabani & M. Yousefi (2020). An optimized model using LSTM network for demand forecasting. *Computers & Industrial Engineering*, *143*, 106435. https://doi.org/10.1016/j.cie.2020.106435.

[2] F. D. Adhinata & D. P. Rakhmadani (2021). Prediction of Covid-19 daily case in Indonesia using long short term memory method. *Teknika*, *10*(1), 62–67. https://doi.org/10.34148/TEKNIKA.V10I1.328.

[3] O. Al Qasem, M. Akour & M. Alenezi (2020). The influence of deep learning algorithms factors in software fault prediction. *IEEE Access*, *8*, 63945–63960. https://doi.org/10.1109/ACCESS.2020.2985290.

[4] F. D. S. Alhamdani, G. I. Marthasari & C. S. K. Aditya (2021). Prediksi harga emas menggunakan metode time series long short-term memory neural network. *Jurnal Repositor*, *3*(4). https://doi.org/10.22219/repositor.v3i4.31959.

[5] M. Almousa, T. Zhang, A. Sarrafzadeh & M. Anwar (2022). Phishing website detection: How effective are deep learning-based models and hyperparameter optimization? *Security and Privacy*, *5*(6), e256.

[6] M. Y. Aristyanto & R. Kurniawan (2021). Pengembangan metode neural machine translation berdasarkan hyperparameter neural network. In *Seminar Nasional Official Statistics*, volume 2021 pp. 935–946. https://doi.org/10.34123/semnasoffstat.v2021i1.789.

[7] K. Aswin & S. Thangavel (2023). The predictive analysis for economic development and financial status of India in 2023. *BOHR International Journal of Financial Market and Corporate Finance*, *2*(1), 10–16. https://doi.org/10.54646/bijfmcf.013.

[8] J. Baffes, D. Cosic & V. Kshirsagarm (2023). *Commodity market outlook*. World Bank Group, Washington DC.

[9] S. Bodapati, H. Bandarupally & M. Trupthi (2020). COVID-19 time series forecasting of daily cases, deaths caused and recovered cases using long short term memory networks. In *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pp. 525–530. IEEE. https://doi.org/10.1109/ICCCA49541.2020.9250863.

[10] J. Brownlee. Difference between a batch and an epoch in a neural network 2022.

[11] S. C. Chapra (2012). *Applied numerical methods with MATLAB for engineers and scientists, 3rd edition*. McGraw-Hill, New York.

[12] J. D. De Guia, R. S. Concepcion, H. A. Calinao, J. Alejandrino, E. P. Dadios & E. Sybingco (2020). Using stacked long short term memory with principal component analysis for short term prediction of solar irradiance based on weather patterns. In *2020 IEEE REGION 10 CONFERENCE (TENCON)*, pp. 946–951. IEEE. https://doi.org/10.1109/TENCON50793.2020.9293719.

[13] M. L. De Prado (2018). *Advances in financial machine learning*. John Wiley & Sons, New Jersey.

[14] F. X. Diebold & R. S. Mariano (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *20*(1), 134–144. https://doi.org/10.1198/073500102753410444.

[15] Z. Enslin, E. Du Toit & J. J. Szczygielski (2018). An investigation into the changing relationship between the gold price and South African gold mining industry returns. *South African Journal of Business Management*, *49*(1), 1–11. https://doi.org/10.4102/sajbm.v49i1.232.

[16] A. Fairuzie, A. Siagian & Y. Stefhani (2022). Analisis pengaruh earning per share, harga emas dunia, inflasi terhadap harga saham perusahaan sektor pertambangan di bursa efek Indonesia pada masa pandemi Covid-19. *Jurnal Manajemen*, *6*(2), 37–52. https://doi.org/10.54964/manajemen.v6i2.202.

[17] I. Goodfellow, Y. Bengio & A. Courville (2016). *Deep learning*. MIT press, Cambridge, Massachusetts.

[18] K. Gopal, M. A. Rahim & M. Adam (2017). Box-cox transformation of monthly Malaysian gold price range. *Malaysian Journal of Mathematical Sciences*, *11*, 107–118.

[19] P. Goyal, S. Pandey & K. Jain (2018). *Deep learning for natural language processing: Creating neural networks with python*. Apress, New York.

[20] R. Guha (2023). Benchmarking gradient based optimizers' sensitivity to learning rate. *Available at SSRN 4318767*, pp. 1–33. http://dx.doi.org/10.2139/ssrn.4318767.

[21] J.-H. Han, D.-J. Choi, S.-U. Park & S.-K. Hong (2020). Hyperparameter optimization using a genetic algorithm considering verification time in a convolutional neural network. *Journal of Electrical Engineering & Technology*, *15*(2), 721–726. https://doi.org/10.1007/s42835-020-00343-7.

[22] H. W. Herwanto, T. Widiyaningtyas & P. Indriana (2019). Penerapan algoritme linear regression untuk prediksi hasil panen tanaman padi. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, *8*(4), 364–370.

[23] S. Hochreiter & J. Schmidhuber (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[24] R. J. Hyndman & G. Athanasopoulos (2021). *Forecasting: Principles and practice, 3rd edition*. OTexts, Melbourne, Australia.

[25] N. Kalchbrenner, I. Danihelka & A. Graves (2016). Grid long short-term memory. In *4th International Conference on Learning Representations, ICLR 2016*, pp. 1–15. https://doi.org/10.48550/arXiv.1507.01526.

[26] P. Khan, B. S. K. Reddy, A. Pandey, S. Kumar & M. Youssef (2020). Differential channel-state-information-based human activity recognition in IoT networks. *IEEE Internet of Things Journal*, *7*(11), 11290–11302. https://doi.org/10.1109/JIOT.2020.2997237.

[27] D. P. Kingma & J. Ba (2014). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations* (*ICLR 2015*), pp. 1–15. https://doi.org/10.48550/arXiv.1412.6980.

[28] C. D. Lewis (1982). *Industrial and business forecasting methods*. Butterworth Scientific, London. https://doi.org/10.1002/for.3980020210.

[29] L. Liang & X. Cai (2020). Forecasting peer-to-peer platform default rate with LSTM neural network. *Electronic Commerce Research and Applications*, *43*, 100997. https://doi.org/10.1016/j.elerap.2020.100997.

[30] F.-J. Lin, S.-Y. Chen, L.-T. Teng & H. Chu (2009). Recurrent functional-link-based fuzzy neural network controller with improved particle swarm optimization for a linear synchronous motor drive. *IEEE Transactions on Magnetics*, *45*(8), 3151–3165. https://doi.org/10.1109/TMAG.2009.2017530.

[31] S. Lin, Y. Feng et al. (2022). Research on stock price prediction based on orthogonal gaussian basis function expansion and pearson correlation coefficient weighted LSTM neural network. *Advances in Computer, Signals and Systems*, *6*(5), 23–30. https://doi.org/10.23977/acss.2022.060504.

[32] Z. Liu & X. Yang (2022). Cross validation for uncertain autoregressive model. *Communications in Statistics-Simulation and Computation*, *51*(8), 4715–4726. https://doi.org/10.1080/03610918.2020.1747077.

[33] B. Long, F. Tan & M. Newman (2023). Forecasting the monkeypox outbreak using ARIMA, prophet, NeuralProphet, and LSTM models in the United States. *Forecasting*, *5*(1), 127–137. https://doi.org/10.3390/forecast5010005.

[34] S. Mahjoub, L. Chrifi-Alaoui, B. Marhic & L. Delahoche (2022). Predicting energy consumption using LSTM, multi-layer GRU and drop-GRU neural networks. *Sensors*, *22*(11), 4062. https://doi.org/10.3390/s22114062.

[35] W. Mahmood & E. Avşar (2021). One step ahead prediction of ozone concentration for determination of outdoor air quality level. *MANAS Journal of Engineering*, *9*(1), 45–54. https://doi.org/10.51354/mjen.869736.

[36] A. Manowska (2020). Using the LSTM network to forecast the demand for hard coal. *Gospodarka Surowcami Mineralnymi-Mineral Resources Management*, *36*(4), 33–48. https://doi.org/10.24425/gsm.2020.133945.

[37] F. Mehmood, S. Ahmad & T. K. Whangbo (2023). An efficient optimization technique for training deep neural networks. *Mathematics*, *11*(6), 1360. https://doi.org/10.3390/math11061360.

[38] O. A. Montesinos López, A. Montesinos López & J. Crossa (2022). Overfitting, model tuning, and evaluation of prediction performance. In *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, pp. 109–139. Springer, Cham. https://doi.org/10.1007/978-3-030-89010-0_4.

[39] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab & A. Mosavi (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, *8*, 150199–150212. https://doi.org/10.1109/ACCESS.2020.3015966.

[40] A. Ostermann, A. Bajrami & A. Bogensperger (2024). Short-term forecasting of German generation-based CO2 emission factors using parametric and non-parametric time series models. *Energy Informatics*, *7*(1), 1–28. https://doi.org/10.1186/s42162-024-00303-9.

[41] R. K. Paul, M. Yeasin, P. Kumar, P. Kumar, M. Balasubramanian, H. Roy, A. Paul & A. Gupta (2022). Machine learning techniques for forecasting agricultural prices: A case of brinjal in Odisha, India. *Plos One*, *17*(7), e0270553. https://doi.org/10.1371/journal.pone.0270553.

[42] S.-H. Poon (2005). *A practical guide to forecasting financial market volatility*. John Wiley & Sons, West Sussex.

[43] I. F. Puspitasari, N. Andriyani & N. Hidayah (2022). Emas sebagai safe haven dan inflation hedging di tengah ketidakpastian perekonomian global selama pandemi Covid-19. *JURNAL PENDIDIKAN EKONOMI: Jurnal Ilmiah Ilmu Pendidikan, Ilmu Ekonomi Dan Ilmu Sosial*, *16*(2), 250–258. https://doi.org/10.19184/jpe.v16i2.33694.

[44] N. A. Ramli, T. Ismail & H. C. Wooi (2014). Application of fuzzy optimization and time series for early warning system in predicting currency crisis. *Malaysian Journal of Mathematical Sciences*, *8*(2), 239–253.

[45] Sanghvirajit. A complete guide to Adam and RMSprop optimizer 2021. https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be.

[46] J. Sarangapani (2018). *Neural network control of nonlinear discrete-time systems*. CRC press, Taylor & Francis Group, Boca Raton, Florida. https://doi.org/10.1201/9781420015454.

[47] J. Schmuller (2017). *Statistical analysis with R for dummies*. John Wiley & Sons, Hoboken, New Jersey.

[48] U. W. Soelistijo, P. L. Anjani, H. I. Pratama, H. La Pili & M. K. Herdyanti (2015). Trend of mineral commodity price and its impact on the Indonesia economy 1990-2025. *Earth Sciences*, *4*(4), 129–145. https://doi.org/10.11648/j.earth.20150404.11.

[49] J. N. Van Rijn & F. Hutter (2018). Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2367–2376. https://doi.org/10.1145/3219819.3220058.

[50] G. Vidya & V. Hari (2020). Gold price prediction and modelling using deep learning techniques. In *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 28–31. IEEE. https://doi.org/10.1109/RAICS51191.2020.9332471.

[51] S. Wahyuningsih, R. Goejantoro, M. Siringoringo, A. Saputra & S. Aminah (2019). Application seasonal autoregressive integrated moving average to forecast the number of East Kalimantan hotspots. In *Journal of Physics: Conference Series*, pp. 012085. IOP Publishing. https://doi.org/10.1088/1742-6596/1351/1/012085.

[52] I. Yulian, D. S. Anggraeni & Q. Aini (2020). Penerapan metode trend moment dalam forecasting penjualan produk CV. Rabbani Asyisa. *JURTEKSI* (*Jurnal Teknologi dan Sistem Informasi*), *6*(2), 193–200. https://doi.org/10.33330/jurteksi.v6i2.443.

[53] M. Yurtsever (2021). Gold price forecasting using LSTM, Bi-LSTM and GRU. *Avrupa Bilim ve Teknoloji Dergisi*, (31), 341–347. https://doi.org/10.31590/EJOSAT.959405.

[54] S. Zahara & M. B. Ilmiddafiq (2019). Prediksi indeks harga konsumen menggunakan metode long short term memory (LSTM) berbasis cloud computing. *Jurnal RESTI* (*Rekayasa Sistem Dan Teknologi Informasi*), *3*(3), 357–363. https://doi.org/10.29207/resti.v3i3.1086.