



Comparison Of Long Short-Term Memory and Gated Recurrent Unit Methods with RMSProp Optimization in Forecasting Stock Closing Prices

Rani Anggraini*, Arief Agoestanto

Department of Mathematics, Faculty of Mathematics and Natural Science,
Universitas Negeri Semarang, Semarang, 50229, Indonesia

Article Info

Article History:

Received 03-07-2025

Accepted 20-10-2025

Published 31-10-2025

Keywords:

Forecasting, LSTM, GRU,
RMSprop Optimization,
Stocks

Abstract

Retail sector stocks are one of the key indicators supporting consumer growth. PT Sumber Alfaria Trijaya Tbk, the operator of the Alfamart minimarket chain, is a leading retail issuer on the Indonesia Stock Exchange, listed in the LQ45 index with a Return on Equity (ROE) of 23.5%. The seasonal, volatile, and macroeconomic-sensitive nature of its stock price makes it difficult to forecast with conventional linear methods. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were chosen because their gating mechanisms overcome the vanishing gradient problem and capture nonlinear temporal dependencies, and RMSprop was chosen because it adapts the learning rate of each weight to the recent gradient magnitude, which stabilizes training on noisy data. This study aims to compare LSTM and GRU with RMSprop optimization in forecasting the closing price of PT Sumber Alfaria Trijaya Tbk shares. The data consist of 1,211 daily closing prices from January 2, 2020, to December 30, 2024, normalized with min-max scaling and split 80:20 into training and test data. Each model was tested on a grid of 36 combinations of time step, batch size, units, and epochs, evaluated with RMSE and MAPE, and the mean error at each hyperparameter level was analyzed to assess sensitivity. The best LSTM (time step 60, batch size 8, units 100, epochs 50) yielded an RMSE of 56.948 and a MAPE of 1.409%, whereas the best GRU (time step 30, batch size 8, units 50, epochs 150) yielded an RMSE of 52.689 and a MAPE of 1.343%. GRU with RMSprop optimization is therefore more effective in the best case, although the average errors of both models are practically equal. A short window and a small batch size are recommended as the starting configuration for forecasting retail stocks, with the forecasts used as decision support alongside fundamental analysis.

How to cite:

Anggraini, R., & Agoestanto, A. (2025). Comparison of Long Short-Term Memory and Gated Recurrent Unit Methods with RMSprop Optimization in Forecasting Stock Closing Prices. *UNNES Journal of Mathematics*. 14 (2): 1-10.

1. Introduction

Forecasting is the ability to predict what will happen in the future based on analysis of past and present data (Moran et al., 2016). In the context of capital markets, the ability to make predictions is used to aid strategic decision-making in order to generate profits for capital market players. However, predictions in capital markets face complex challenges due to high volatility influenced by various economic and non-economic factors.

With the advancement of technology, investment activity in the capital market continues to increase. Based on data from the Otoritas Jasa Keuangan (OJK), the number of capital market investors in Indonesia as of October 22, 2024 reached more than 14 million investors. This number increased by more than 2 million new investors or 16.81% ytd compared to the end of 2023. This growth indicates that the Indonesian public is increasingly interested in investing in the capital market, particularly in stocks.

Retail stocks are one of the sectors that play an important role in supporting the growth of public consumption. This sector has relatively stable growth in line with population growth and evolving consumption patterns. One of the most prominent retail issuers listed on the Indonesia Stock Exchange (IDX) is PT Sumber Alfaria Trijaya Tbk. This company is known as the manager of the Alfamart minimarket chain, which is spread throughout Indonesia. Alfamart is also listed as a constituent of the LQ45 index, one of the major stock indices on the IDX, and has a relatively strong Return on Equity (ROE) of 23.5%. Retail sector stocks exhibit seasonal characteristics, short-term volatility, and are highly sensitive to changes in consumer behavior and macroeconomic conditions such as inflation and purchasing power. These characteristics result in non-linear and dynamic stock price patterns. Stock investments in the capital market are one of the most popular investment options among investors, as stocks are perceived to offer higher returns, high liquidity, and ease of transaction (Tarno et al., 2022). Although offering significant profits, stock investments are also prone to risks of loss that can occur at any time. Stock prices fluctuate constantly, so stock price forecasting is very important.

The urgency of this research lies in the gap between the rapid growth of stock market participation and the limited empirical guidance on how recurrent neural networks should be configured for volatile retail-sector stocks. Inaccurate closing-price forecasts can lead investors to mistime transactions and bear avoidable losses, particularly for stocks whose prices respond sharply to inflation, purchasing power, and seasonal consumption. At the same time, the accuracy of deep learning forecasts depends strongly on hyperparameter choices such as the time step, batch size, number of units, and number of epochs, and earlier applications of LSTM and GRU to stock forecasting have not always produced convincing results (Lawi et al., 2022). A systematic and reproducible comparison of both architectures under controlled settings is therefore needed so that investors, analysts, and researchers can select a forecasting model on the basis of evidence rather than trial and error.

Deep learning is a branch of machine learning whose algorithms are inspired by the structure of the human brain, known as neural networks (Alfarizi et al., 2023). Among the neural network architectures, Recurrent Neural Networks (RNN) are designed for sequential data, but standard RNNs suffer from the vanishing gradient problem when they learn long dependencies. LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Cho et al., 2014) overcome this problem with gating mechanisms that regulate which information is retained or discarded, which makes them suitable for the nonlinear and noisy dynamics of stock prices. Empirical comparisons of gated recurrent units found GRU comparable to LSTM, with both better than traditional recurrent units (Chung et al., 2014), so the choice between the two has to be tested for each data set. Model accuracy also depends on the optimizer, which adjusts the weights to minimize the loss function (Setiawan & Kartikasari, 2022). RMSprop was selected because it divides each weight update by a moving average of the recent squared gradients (Tieleman & Hinton, 2012), so the effective learning rate adapts to each weight and training remains stable on noisy and non-stationary data such as stock prices.

Previous studies have applied LSTM and GRU to stock price forecasting. Arfan and Lussiana (2020) showed that LSTM predicted stock prices in Indonesia better than SVR, and Nilsen et al. (2022) found that GRU was more accurate than ARIMA for the closing prices of PT Bank Central Asia Tbk and PT Kertas Tjiwi Kimia Tbk. Khairunisa and Hendikawati (2024) compared LSTM and GRU while varying the learning rate, the number of epochs, and the number of neurons, and found only a marginal advantage for GRU (MAPE of 1.27% versus 1.28%). Meri Aryati et al. (2024) also compared the two methods and reported that their relative performance changed with the parameter combination, while Setiawan and Kartikasari (2022) combined LSTM with RMSprop and Adam optimization to predict cocoa prices. To the authors' knowledge, however, few studies have varied the time step, batch size, units, and epochs jointly for both architectures under one fixed optimizer for a retail-sector stock listed in the LQ45 index.

The novelty of this study lies in three aspects. First, it focuses on a retail-sector issuer, PT Sumber Alfaria Trijaya Tbk, whose price dynamics are shaped by seasonal consumption and macroeconomic conditions, using daily data from 2020 to 2024 that cover the COVID-19 pandemic and the subsequent recovery. Second, RMSprop is held fixed as the optimizer so that differences in performance can be attributed to the architecture and to four hyperparameters—time step, batch size, units, and epochs—which are varied in a full grid of 36 combinations for each model. Third, besides reporting the best configuration, the study analyzes the mean effect of every hyperparameter level on RMSE and MAPE, so that the findings show not only which model is best but also how sensitive each model is to its settings.

This study aims to compare the performance of LSTM and GRU with RMSprop optimization in forecasting the closing price of PT Sumber Alfaria Trijaya Tbk shares. Specifically, it addresses three research questions: (1) which combination of time step, batch size, units, and epochs gives the best LSTM model with RMSprop optimization; (2) which combination gives the best GRU model with RMSprop optimization; and (3) which of the two models is more accurate according to RMSE and MAPE. The results are expected to give investors and researchers an evidence-based reference for configuring recurrent neural network forecasts of retail-sector stocks.

The theoretical basis of the methods and the evaluation metrics used in this study is described below.

Long Short-Term Memory (LSTM) is a specialized RNN that can model long-range dependencies and temporal sequences with higher accuracy than conventional RNN (Yudistira et al., 2023). LSTM emerged as a solution to address the vanishing gradient problem in RNNs when processing long sequential data. The memory cell component of LSTM enables the understanding of complex data, as it can store information for extended periods (Akbar et al., 2024). The LSTM architecture consists of three gates: the input gate, forget gate, and output gate. Additionally, the LSTM process includes computational stages such as Candidate Cell State, Update Cell State, and Hidden State. The LSTM algorithm equation is as follows.

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

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

where, i_t is the input gate, f_t is the forget gate, $\tilde{C}_t$ is the candidate cell state, C_t represents the update cell state, h_t is the hidden state, W is the weight, b is the bias, and x_t indicates the input value.

GRU has unit gates that can be used to modulate data flow within the unit. The main task of GRU is to address the vanishing gradient problem that occurs in standard neural networks (Islam & Hossain, 2021). Unlike LSTM, the GRU architecture uses update gates and reset gates to address the vanishing gradient problem and also decides what the output will be. GRU also includes Candidate Hidden State and Current Hidden State. The GRU algorithm equation is as follows.

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z) \quad (7)$$

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r) \quad (8)$$

$$\tilde{h}_t = \tanh(W_h \cdot x_t + U_h \cdot (r_t * h_{t-1}) + b_h) \quad (9)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (10)$$

where, r_t is the reset gate, z_t is the update gate, $\tilde{h}_t$ is the candidate hidden state, W and U are weights, b is the bias, and x_t indicates the input value.

Root Mean Square Propagation (RMSprop) is an optimization method that maintains a moving average of the squared gradients for each weight (Yotanka et al., 2020). RMSprop is an optimization function that utilizes the magnitude of the latest gradient to normalize the gradient. The RMSprop equation is as follows.

$$E[g^2]^{(t)} = \beta E[g^2]^{(t-1)} + (1 - \beta) \left(\frac{\partial c}{\partial w} \right)^2 \quad (11)$$

$$w_{ij}^{(t)} = w_{ij}^{(t-1)} - \frac{\eta}{\sqrt{E[g^2]}} \frac{\partial c}{\partial w_{ij}} \quad (12)$$

where, $E[g]$ is the exponential average of the squared gradient (moving average), $\frac{\partial c}{\partial w}$ is the gradient of the loss function, η is the learning rate, β is the *decay rate*, usually 0.9.

Performance measurement is used to evaluate models by measuring error rates. Lower RMSE and MAPE values indicate better model accuracy.

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2} \quad (13)$$

$$MAPE = \frac{1}{n} \sum \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (14)$$

Where, $\hat{y}_i$ is the predicted value, y_i is the actual value, n is the amount of data.

2. Method

The data used in this study are secondary data of the daily stock prices of PT Sumber Alfaria Trijaya Tbk (AMRT.JK) obtained from Yahoo Finance (<https://finance.yahoo.com/>). The data set contains the date, open, close, high, low, adjusted close, and volume. Only the closing price is used as the forecast variable because it represents the final market valuation of each trading day. The observation period runs from January 2, 2020, to December 30, 2024, and contains 1,211 daily observations. All computations were carried out in Python on Google Colab.

The preprocessing stage consisted of three steps: (i) checking for missing values, (ii) min-max normalization, which rescales each closing price to the interval [0, 1], and (iii) splitting the data in chronological order with an 80:20 ratio into 968 training observations and 243 test observations, so that the models are tested only on prices that occur after the training period. The normalized series was then converted into supervised-learning samples with a sliding window: the closing prices of the previous k trading days (the time step, k) form the input sequence, and the closing price of the next trading day is the target.

Two recurrent architectures, LSTM and GRU, were built, and both were trained with the RMSprop optimizer. For each architecture a full grid of hyperparameters was evaluated (Table 2): time step (30 and 60), batch size (8, 16, and 32), number of units (50 and 100), and number of epochs (50, 100, and 150). This gives $2 \times 3 \times 2 \times 3 = 36$ combinations per architecture and 72 trained models in total.

Each trained model was used to forecast the test data, and the predictions were transformed back to the original price scale before the errors were calculated. Accuracy was measured with RMSE (Equation 13) and MAPE (Equation 14); lower

values indicate a more accurate model. The analysis proceeded in three steps: (1) the best configuration of each architecture was identified as the combination with the lowest RMSE and MAPE; (2) the two best models were compared with each other and with the actual closing prices in graphical and tabular form; and (3) the mean RMSE and MAPE at every hyperparameter level were calculated over all 36 combinations to examine how sensitive each architecture is to the individual hyperparameters. The research procedure is summarized in Figure 1.

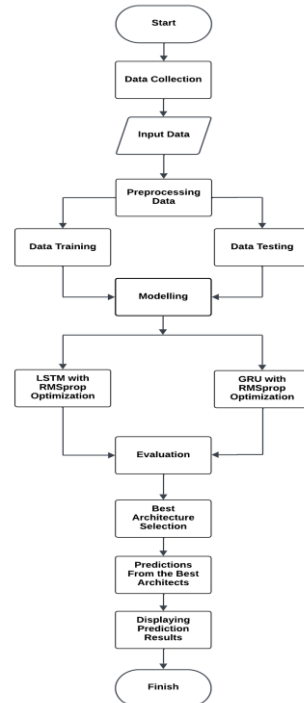


Figure 1 Research methodology

3. Results and Discussion

The dataset collected and used in this study is the Sumber Alfaria Trijaya Tbk (AMRT.JK) stock price dataset taken from the Yahoo Finance website in CSV format. The data used covers the period from January 2, 2020, to December 30, 2024, and consists of 1,211 daily data points. This study uses stock closing prices, as closing prices represent the final value of the stock at the end of the trading day. Therefore, the close column was selected as the primary variable. The following is a visualization of the daily movement of closing prices for PT Sumber Alfaria Trijaya Tbk over the past five years.



Figure 2 Alfamart stock chart

Figure 2 shows Alfamart's stock chart over the past five years. Overall, the stock price has experienced a consistent upward trend in the medium to long term, with several significant fluctuations during the period from 2020 to 2024. In mid-2021, the stock began to experience a significant increase, marked by a gradual surge in price.

In the preprocessing stage, there are several steps, including checking for missing values, dividing the data, and normalizing the data. After checking for missing values, the number of missing values in the close column is 0. This indicates that the dataset is complete, so no missing value handling is required. Next, the dataset is divided using an 80:20 ratio, with 80% of the data used for training and 20% for testing. Based on the total data count of 1,211, the training data consists of 968 data points, while the testing data consists of 243 data points. The next step is data normalization, which involves transforming the original data range into a range of 0 to 1 using min-max scaling. Table 1 below shows the results of the stock data after normalization.

Date	Before Normalization	After Normalization
2020-01-02	875.00	0.08488964
2020-01-03	865.00	0.08149406
2020-01-06	845.00	0.0747028
...	...	...
2024-12-24	2780.00	0.73174873
2024-12-27	2800.00	0.7385399
2024-12-30	2850.00	0.75551783

This research was conducted using experiments on several combinations of hyperparameters to find the best architecture. Several hyperparameters used in this research were the number of time steps, batch size, units, and epochs. Meanwhile, the parameter that had been determined was RMSprop optimization. The hyperparameter settings are shown in Table 2.

Hyperparameter	Value
Time step	30, 60
Batch size	8, 16, 32
Units	50, 100
Epoch	50, 100, 150

The following are the results of testing the LSTM and GRU methods with RMSprop optimization based on model evaluation using the RMSE and MAPE metrics at time step 30.

Hyperparameter			LSTM		GRU	
Batch Size	Units	Epoch	RMSE	MAPE	RMSE	MAPE
8	50	50	92.173	2.595%	54.983	1.380%
		100	109.798	3.433%	59.286	1.595%
		150	136.123	4.315%	52.166	1.326%
	100	50	59.256	1.521%	63.106	1.663%
		100	65.522	1.762%	85.020	2.528%
		150	75.607	2.223%	54.859	1.367%
16	50	50	96.481	2.860%	123.438	3.810%
		100	66.351	1.664%	95.499	2.905%
		150	66.027	1.754%	69.079	1.906%
	100	50	90.233	2.653%	64.448	1.713%
		100	76.708	2.175%	56.173	1.438%
		150	57.537	1.472%	55.897	1.416%
32	50	50	126.232	3.803%	79.661	2.142%
		100	67.174	1.742%	228.562	7.648%
		150	62.562	1.615%	118.063	3.637%
	100	50	71.746	1.800%	93.475	2.745%
		100	109.885	3.227%	85.101	2.449%
		150	60.947	1.552%	58.361	1.524%

Table 3 shows the results of the LSTM model evaluation, which performed best at a batch size of 16, 100 units, and 150 epochs with an RMSE of 58.066 and a MAPE of 1.494%. The GRU model shows the best performance at a batch size of 8, 50 units, and 150 epochs with an RMSE of 52.689 and a MAPE of 1.343%.

The following are the results of testing the LSTM and GRU methods with RMSprop optimization based on model evaluation using the RMSE and MAPE metrics at time step 60.

Hyperparameter		LSTM			GRU	
Batch Size	Units	Epoch	RMSE	MAPE	RMSE	MAPE
8	50	50	65.768	1.711%	86.383	2.593%
		100	154.453	5.042%	57.071	1.515%
		150	60.200	1.607%	63.947	1.693%
	100	50	57.031	1.414%	85.415	2.492%
		100	85.668	2.561%	70.739	1.933%
		150	66.386	1.858%	52.471	1.334%
16	50	50	74.452	1.879%	57.810	1.427%
		100	68.742	1.808%	65.061	1.781%
		150	59.774	1.518%	56.750	1.488%
	100	50	67.493	1.703%	120.548	3.791%
		100	59.361	1.505%	86.412	2.567%
		150	76.679	2.112%	58.174	1.516%
32	50	50	130.423	3.957%	66.692	1.764%
		100	76.569	2.134%	78.496	2.188%
		150	69.755	1.903%	96.222	2.851%
	100	50	81.319	2.255%	116.352	3.576%
		100	83.076	2.378%	73.334	2.069%
		150	60.539	1.543%	100.878	3.030%

Table 4 shows the results of the LSTM model evaluation, which performed best at a batch size of 8, 100 units, and 50 epochs, with an RMSE of 56.948 and a MAPE of 1.409%. The GRU model shows the best performance at a batch size of 8, 100 units, and 150 epochs with an RMSE of 52.912 and a MAPE of 1.349%.

Across the 36 combinations evaluated for each model, the errors of both models stayed well below 10% (MAPE from about 1.3% to 5.0% for LSTM and from about 1.3% to 7.6% for GRU), and the training loss decreased consistently, which shows that RMSprop trained both architectures stably on this data set. This is consistent with the reason for choosing RMSprop: dividing each weight update by a moving average of recent squared gradients limits oscillations when the gradients of different weights differ greatly in scale. Stability does not mean insensitivity, however. The RMSE of the worst combination is about 2.7 times that of the best combination for LSTM (roughly 154 versus 57) and about 4.4 times for GRU (roughly 229 versus 52), so the choice of hyperparameters remains decisive. To summarize this sensitivity, Table 5 reports the mean RMSE and MAPE of all combinations at each hyperparameter level.

Table 5 Mean RMSE and MAPE of all combinations by hyperparameter level

Hyperparameter	Level	LSTM		GRU	
		RMSE	MAPE (%)	RMSE	MAPE (%)
Time step	30	82.8	2.34	83.2	2.40
	60	77.6	2.16	77.4	2.20
Batch size	8	85.7	2.50	65.5	1.78
	16	71.7	1.93	75.8	2.15
Units	32	83.4	2.33	99.6	2.97
	50	87.9	2.52	83.8	2.42
	100	72.5	1.98	76.7	2.18
Epoch	50	84.4	2.35	84.4	2.42
	100	85.3	2.45	86.7	2.55
	150	71.0	1.96	69.7	1.92
All combinations (n = 36)		80.2	2.25	80.3	2.30

Note: means are computed from the 36 combinations of each model in Tables 3 and 4; lower values indicate better accuracy.

Time step 30 produced the best GRU model, whereas time step 60 produced the best LSTM model. This agrees with Moodi et al. (2024), who also found that an LSTM with a time step of 60 performed better, while a GRU with a time step of 30 was highly accurate. Averaged over all combinations (Table 5), however, both models had slightly lower errors at time step 60 (mean RMSE of 77.6 for LSTM and 77.4 for GRU) than at time step 30 (82.8 and 83.2), so the effect of the window length is modest. The preference of GRU for the shorter window and of LSTM for the longer one is consistent with the view that the separate memory cell of LSTM suits longer sequences, but it should be read as a tendency of the best configurations rather than as a general rule.

Batch size 8 produced the lowest error for GRU on average (mean RMSE of 65.5, compared with 75.8 at batch size 16 and 99.6 at batch size 32), and four of the five best GRU combinations use batch size 8. For LSTM, however, the lowest mean error was obtained at batch size 16 (71.7) rather than at batch size 8 (85.7), even though the single best LSTM combination uses batch size 8. A small batch is therefore clearly beneficial for GRU, whereas LSTM appears to prefer a moderate batch size. This agrees in part with Arinal and Puspita (2025), who reported that smaller batch sizes gave lower MAPE and RMSE than larger ones. Small batches update the weights more often and introduce gradient noise that can help the optimizer escape poor solutions, which is useful for a training set of only 968 observations; large batches give fewer updates per epoch and slow down convergence.

The mean error decreased when the number of units increased from 50 to 100 in both models (LSTM: mean RMSE from 87.9 to 72.5; GRU: from 83.8 to 76.7), which supports the argument that a larger hidden state can represent more complex patterns. This is in line with Khairunisa and Hendikawati (2024), who reported that the number of neurons affects the accuracy of stock price forecasts. The best GRU combination, however, uses 50 units, which suggests that GRU, with fewer gates and parameters than LSTM, can reach its best accuracy with a smaller hidden state, whereas the larger configuration is more robust on average. Because more units also increase the risk of overfitting, the number of units should be tuned together with the number of epochs and the batch size.

The mean error was lowest at 150 epochs for both models (mean RMSE of 71.0 for LSTM and 69.7 for GRU), compared with 50 epochs (84.4 and 84.4) and 100 epochs (85.3 and 86.7). The fact that 100 epochs was no better than 50 epochs shows that longer training does not automatically improve generalization, and that results also vary between training runs because of random weight initialization. This is similar to the finding of Meri Aryati et al. (2024) that the effect of the number of epochs differs between LSTM and GRU across parameter combinations. Four of the five best GRU combinations use 150 epochs, whereas the best LSTM combinations are spread over 50 to 150 epochs, which suggests that GRU benefits more from longer training under RMSprop.

The best LSTM combination with RMSprop was obtained at a time step of 60, a batch size of 8, 100 units, and 50 epochs (RMSE of 56.948 and MAPE of 1.409%), whereas the best GRU combination was obtained at a time step of 30, a batch size of 8, 50 units, and 150 epochs (RMSE of 52.689 and MAPE of 1.343%). The best GRU is thus 7.5% lower in RMSE and 0.07 percentage points lower in MAPE. The margin is modest, and the mean errors over all 36 combinations are practically identical for the two models (mean RMSE of 80.2 for LSTM and 80.3 for GRU; mean MAPE of 2.25% and 2.30%), with GRU producing the lower RMSE in 19 of the 36 matched combinations. GRU is also more sensitive to its settings (standard deviation of RMSE of 32.9 versus 24.6 for LSTM). The advantage of GRU therefore lies in the best accuracy that can be reached when the hyperparameters are chosen carefully, not in uniformly better performance. The simpler structure of GRU may allow more efficient training and faster adaptation to short-term fluctuations, particularly at time step 30; this explanation is plausible but was not tested directly in this study.

The following is a graph that illustrates the comparison of actual prices and predicted prices from the LSTM and GRU methods.

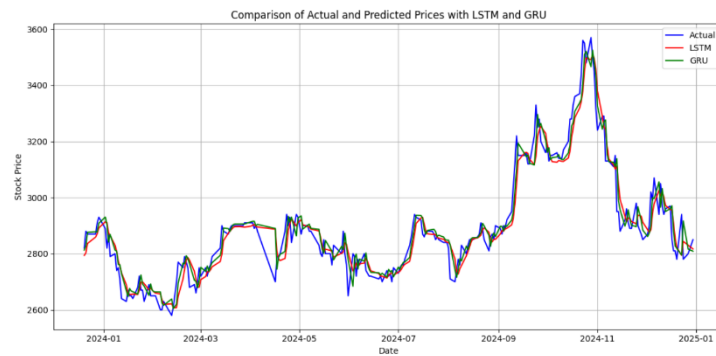


Figure 3 Graph comparison chart between actual prices and predicted prices

Based on Figure 3, it can be seen that there are differences in the degree of closeness between the prediction results and the actual stock prices for each method. In general, the predictions generated by the GRU method show a relatively smaller distance from the actual prices compared to the LSTM method. This indicates that the GRU model has a better ability to capture stock price movement patterns. The following are the prediction results for the sample data from November 8, 2023, to November 13, 2024.

Table 6 Comparison of actual data and predicted results

Date	Actual	LSTM	GRU
2023-11-08	2820.00	2793.95	2815.33
2023-11-09	2880.00	2802.92	2829.77
2023-11-10	2870.00	2836.39	2877.54
2023-11-13	2870.00	2859.28	2880.39
2023-11-14	2910.00	2871.33	2879.71
...	...	...	...
2024-11-07	2780.00	2821.88	2819.24
2024-11-08	2940.00	2794.08	2795.30
2024-11-11	2780.00	2844.12	2916.62
2024-11-12	2800.00	2830.34	2816.66
2024-11-13	2850.00	2817.06	2808.89

Based on Table 6, the GRU produced predictions closer to the actual closing price than the LSTM on 8 of the 10 dates shown, which is consistent with the lower RMSE and MAPE of the GRU. Both models, however, lagged behind abrupt price changes: on 2024-11-08 the actual price rose to 2,940 whereas both models predicted about 2,795, and on 2024-11-11 the GRU prediction (2,916.62) overshoot the actual price of 2,780 by more than the LSTM prediction did. This behavior is typical of models that rely only on past closing prices and cannot anticipate news or macroeconomic shocks.

Compared with previous studies, the finding that GRU achieves the lowest error is in line with Khairunisa and Hendikawati (2024), who also found GRU slightly more accurate than LSTM for stock price forecasting, and with Nilsen et al. (2022), who found GRU more accurate than ARIMA. The present study extends this work in two ways. First, the analysis over the full grid shows that the advantage of GRU appears mainly in the best case: the mean RMSE and MAPE of the two models over all 36 combinations are practically equal, and GRU is more sensitive to the choice of hyperparameters. This nuance cannot be seen when only the best configuration is reported. Second, the study identifies a configuration pattern for a retail-sector stock in the LQ45 index—batch size 8 in both best models, and 100 units and 150 epochs being favorable on average—that was not found for this issuer in the literature reviewed here. Because the data sets, periods, and hyperparameter grids differ among studies, absolute error values are not directly comparable, and the comparison is limited to the direction and pattern of the results.

These findings have practical and methodological implications. For investors and analysts, GRU with RMSprop (time step 30, batch size 8, 50 units, 150 epochs) is a reasonable starting configuration for forecasting the closing price of retail-sector stocks such as AMRT. Its RMSE of 52.689 corresponds to about 1.8% of the last closing price in the data set (2,850), so the forecasts are useful as one input to decision making together with fundamental analysis and risk management, but not as a stand-alone trading signal, particularly around abrupt price changes. For researchers, the wide spread of errors across hyperparameter combinations indicates that models should be compared over a grid of settings and that mean performance should be reported together with best performance.

This study has several limitations. It uses a single stock and one training run per combination, and the best configurations were selected on the basis of the error on the test data without a separate validation set, so the reported errors may be optimistic and may change with a different random initialization. Future studies are recommended to (i) add a validation set or use walk-forward validation, (ii) repeat the training with different random seeds and test the significance of the difference between the models, (iii) include other stocks, sectors, and optimizers such as Adam, and (iv) add exogenous variables such as trading volume and macroeconomic indicators.

4. Conclusion

This study compared LSTM and GRU with RMSprop optimization in forecasting the closing price of PT Sumber Alfaria Trijaya Tbk shares, and the findings answer the three research questions. First, the best LSTM model was obtained with a time step of 60, a batch size of 8, 100 units, and 50 epochs, with an RMSE of 56.948 and a MAPE of 1.409%. Second, the best GRU model was obtained with a time step of 30, a batch size of 8, 50 units, and 150 epochs, with an RMSE of 52.689 and a MAPE of 1.343%. Third, the GRU was more accurate than the LSTM in terms of both RMSE and MAPE, but the margin is modest (7.5% lower RMSE and 0.07 percentage points lower MAPE), and the mean errors over all 36 combinations are practically identical for the two models. The advantage of GRU therefore lies in the best accuracy attainable with well-chosen hyperparameters, and GRU is more sensitive to those hyperparameters than LSTM.

Based on these findings, GRU with RMSprop optimization, a short window (time step 30), and a small batch size is recommended as the starting configuration for forecasting the closing price of this stock, with the forecasts used as decision support alongside fundamental analysis rather than as a stand-alone trading signal. Future research is recommended to use a validation set or walk-forward validation, repeat the training with different random seeds, and extend the analysis to other stocks, sectors, optimizers, and exogenous variables.

References

- Akbar, J., Yudono, M. A. S., & Kharisma, I. L. (2024). Peramalan harga Bitcoin Cash-USD (BCH-USD) pada time frame harian menggunakan LSTM. *Jurnal Mnemonic*, 7(2).
- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python sebagai bahasa pemrograman untuk machine learning dan deep learning. *Karimah Tauhid*, 2(1).
- Arfan, A., & Lussiana, E. T. P. (2020). Perbandingan algoritma long short-term memory dengan SVR pada prediksi harga saham di Indonesia. *PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika*, 13(1), 33–43. <https://doi.org/10.33322/petir.v13i1.858>
- Arinal, V., & Puspita, M. (2025). Peningkatan akurasi nilai harga saham menggunakan metode long short-term memory (LSTM) pada PT Unilever Tbk. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi (JIMIK)*, 6(1). <https://doi.org/10.35870/jimik.v6i1.1190>
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Association for Computational Linguistics. <https://doi.org/10.3115/v1/D14-1179>
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. arXiv. <https://doi.org/10.48550/arXiv.1412.3555>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Islam, M. S., & Hossain, E. (2021). Foreign exchange currency rate prediction using a GRU-LSTM hybrid network. *Soft Computing Letters*, 3, 100009. <https://doi.org/10.1016/j.socl.2020.100009>
- Khairunisa, N. K., & Hendikawati, P. (2024). Long short-term memory and gated recurrent unit modeling for stock price forecasting. *Jurnal Matematika, Statistika dan Komputasi*, 21(1), 321–333. <https://doi.org/10.20956/j.v21i1.35930>
- Lawi, A., Mesra, H., & Amir, S. (2022). Implementation of long short-term memory and gated recurrent units on grouped time-series data to predict stock prices accurately. *Journal of Big Data*, 9(1). <https://doi.org/10.1186/s40537-022-00597-0>
- Meri Aryati, N. W., Wiguna, I. K. A. G., Putri, N. W. S., Widiartha, I. K. K., & Ginantra, N. L. W. S. R. (2024). Komparasi metode LSTM dan GRU dalam memprediksi harga saham. *Jurnal Media Informatika Budidarma*, 8(2), 1131. <https://doi.org/10.30865/mib.v8i2.7342>

- Moodi, F., Jahangard, A., Zare Chahooki, M. A., Rafsanjani, A. J., & Zarifzadeh, S. (2024). Advanced stock price forecasting using a 1D-CNN-GRU-LSTM model. *Journal of AI and Data Mining*, 12(3), 393–408. <https://doi.org/10.22044/jadm.2024.14831.2581>
- Moran, K. R., Fairchild, G., Generous, N., Hickmann, K., Osthus, D., Priedhorsky, R., Hyman, J., & Del Valle, S. Y. (2016). Epidemic forecasting is messier than weather forecasting: The role of human behavior and internet data streams in epidemic forecast. *The Journal of Infectious Diseases*, 214(Suppl. 4), S404–S408. <https://doi.org/10.1093/infdis/jiw375>
- Nilsen, A., Handari, B. D., & Hertono, G. F. (2022). Comparison of autoregressive integrated moving average (ARIMA) model and gated recurrent unit (GRU) model in predicting stock prices. *AIP Conference Proceedings*, 2391. <https://doi.org/10.1063/5.0073838>
- Setiawan, Y., & Kartikasari, P. (2022). Prediksi harga jual kakao dengan metode long short-term memory menggunakan metode optimasi root mean square propagation dan adaptive moment estimation dilengkapi GUI RShiny. *Jurnal Gaussian*, 11(1), 99–107. <https://ejournal3.undip.ac.id/index.php/gaussian/>
- Tarno, T., Trimono, T., Maruddani, D. A. I., Wilandari, Y., & Utami, R. S. (2022). Risk assessment of stocks portfolio through ensemble ARMA-GARCH and value at risk (case study: INDF.JK and ICBP.JK stock price). *Media Statistika*, 14(2), 125–136. <https://doi.org/10.14710/medstat.14.2.125-136>
- Tieleman, T., & Hinton, G. (2012). Lecture 6.5—RMSprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 26–31.
- Yotenka, R., Fikri, F., & Huda, E. (2020). Implementasi long short-term memory pada harga saham perusahaan perkebunan di Indonesia. *Jurnal UJMC*, 6(1), 9–18.
- Yudistira, N., Alfiansih, L. M. D., Andriyani, N. I., Wafa, E., Mualida, N., Maghfiroh, N. A., & Nurdian, I. W. (2023). *Prediksi deret waktu menggunakan deep learning*. UB Press.