Available online at
https://journal.rescollacomm.com/index.php/ijqrm/index

e-ISSN 2721-477X
p-ISSN 2722-5046

International Journal of Quantitative Research and Modeling

Vol. 6, No. 1, pp. 42-47, 2025

Implementation of the Gated Recurrent Unit (GRU) Model for Bank Mandiri Stock Price Prediction

Renda Sandi Saputra[1*], Dede Irman Pirdaus, Moch Panji Agung Saputra[3]

[1,2]*Department of Informatics, Faculty of Technology and Information, University of Informatics and Business, Bandung, Indonesia*

[3]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang, Indonesia*

**Corresponding author email: renda.ss21@student.unibi.ac.id*

Abstract

Stock price prediction is a crucial aspect in the financial world, especially in making investment decisions. This study aims to analyze the performance of the Gated Recurrent Unit (GRU) model in predicting Bank Mandiri (BMRI.JK) stock prices using historical data for five years. Stock data was collected from Yahoo Finance and normalized using Min-Max Scaling to improve model stability. Furthermore, the windowing technique was applied to form a dataset that fits the architecture of the time series forecasting-based model. The developed GRU model consists of two GRU layers with 128 neuron units, two dropout layers to prevent overfitting, and one output layer with one neuron to predict stock prices. Model evaluation was carried out using the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2 Score) metrics. The experimental results show that the GRU model is able to produce predictions with a high level of accuracy, indicated by the R^2 Score value of 0.9636, which indicates that the model can explain 96.36% of stock price variability based on historical data.

Keywords: Stock price prediction, gated recurrent unit (GRU), deep learning, time series, mandiri bank stock.

1. Introduction

In the world of finance, stock price prediction is one of the most important aspects for investors and capital market players (Soni et al., 2022). Prediction accuracy can provide significant benefits, both in making short-term and long-term investment decisions. However, the dynamic and fluctuating nature of the stock market makes stock price prediction a major challenge (Sharma et al., 2020). Various factors such as global economic conditions, monetary policy, and market sentiment can affect stock price movements, so more sophisticated analysis methods are needed to improve prediction accuracy.

The development of artificial intelligence technology, especially in the field of deep learning, has opened up new opportunities in stock market analysis and prediction. Time series forecasting-based models, such as the Gated Recurrent Unit (GRU), are increasingly used because of their ability to handle historical data with complex patterns. GRU is a development of LSTM which has a simpler architecture with fewer parameters, making it more efficient in the training and execution process (Li et al., 2023; Wang et al., 2020). The advantage of GRU lies in the gating mechanism, which allows the model to remember important information from historical data and ignore irrelevant information (Martina and Girana, 2024). This makes GRU an efficient alternative in predicting time series data, including stock prices.

Gated Recurrent Unit (GRU) is a type of artificial neural network based on Recurrent Neural Network (RNN) designed to overcome the problem of vanishing gradients in sequential data processing (Oğuz and Ertuğrul, 2023; Lee and Singh, 2023). GRU has a simpler structure than Long Short-Term Memory (LSTM) because it only uses two main gates, namely the reset gate and the update gate, which function to control the flow of information and maintain relevant information in the long term (Ghadimpour and Ebrahimi, 2022). The reset gate determines how much old information will be forgotten, while the update gate regulates how much new information will be stored. With this mechanism, GRU can handle long-term dependencies in sequential data more efficiently and with lighter computation than LSTM

Prediction accuracy can provide significant benefits, both in short-term and long-term investment decision making. However, the dynamic and volatile nature of the stock market makes stock price prediction a major challenge. Various factors such as global economic conditions, monetary policy, and market sentiment can affect stock price movements, so more sophisticated analysis methods are needed to improve prediction accuracy. This study aims to analyze the performance of the Gated Recurrent Unit (GRU) model in predicting Bank Mandiri (BMRI.JK) stock prices using historical data for five years.

2. Methodology

2.1. Data Collection

The data used in this study is Bank Mandiri stock price data (BMRI.JK) obtained from Yahoo Finance. The data period used covers 5 years, starting from January 1, 2020 to January 1, 2025. This data is downloaded automatically using the yfinance library in Python, which allows historical data retrieval based on a specified time range (Lewinson, 2022; Fleischer et al., 2022). From all the data obtained, this study will use the closing stock price as the main variable that will be displayed in Figure 1.

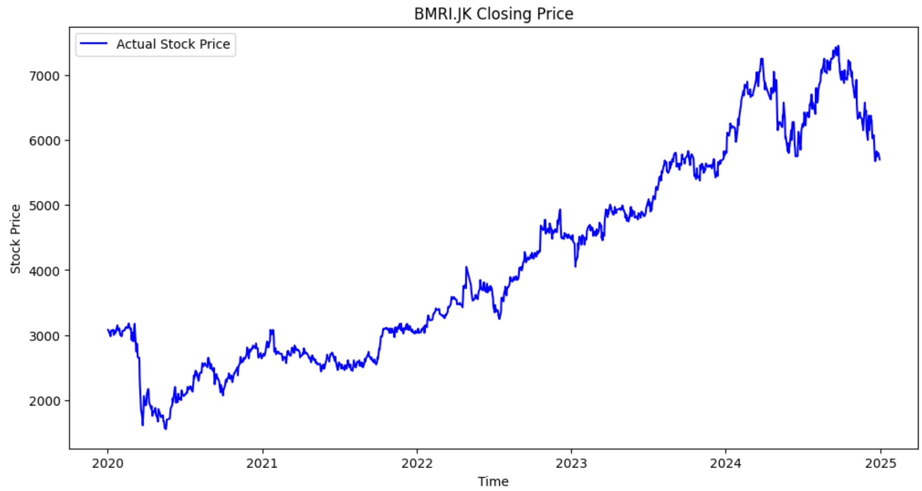

Figure 1: Mandiri stock closing price chart (BMRI.JK)

Based on Figure 1, the stock price of BMRI.JK shows an upward trend (bullish) from 2020 to 2025. In early 2020, the stock experienced a significant decline from the level of 3000 to 1500, most likely influenced by the impact of the COVID-19 pandemic. However, after that period, the stock showed a strong and consistent recovery. From 2021 to 2024, the stock price continued to experience a gradual increase until it reached its highest level of around 7000. Entering 2025, there was a correction that brought the price to the level of 5700-6000. It can be seen that at the beginning of each year the price also experienced a decline and then continued to rise.

2.2. Data Preprocessing

In the preprocessing stage, several steps are taken to prepare the data before being entered into the Gated Recurrent Unit (GRU) model. These steps include data normalization, feature formation, and division of training and testing data, which aims to improve the performance of the model in predicting Bank Mandiri (BMRI.JK) stock prices.

2.2.1. Data Normalization

The first process in preprocessing is data normalization, which aims to align the data scale to better suit the characteristics of the artificial neural network-based model. The Min-Max Scaling technique is used in this study, with a range scale (0,1) using the *MinMaxScaler(feature_range=(0,1))* function. This normalization is important to reduce the dominance of large values and accelerate the convergence process during model training. After normalization, all closing stock price values are converted into a range of 0 to 1 so that they are more stable for processing by Recurrent Neural Network (RNN)-based models such as GRU.

2.2.2. Feature Formation with Windowing

To capture the pattern of stock price movements within a certain time span, a windowing technique is applied with a lookback parameter = 20. This means that each data sample will be formed using information from the previous 20 days to predict the stock price the next day. This process is carried out with the *create_features* (data, lookback) function, which produces two variables X (Features), namely a set of stock price data from the previous 20 days as model input and Y (Label), namely the stock price on the 21st day that will be predicted by the model.

2.2.3. Training and Testing Data Division

The normalized data is then divided into training data (training set) and testing data (testing set). The proportion of data division is carried out with a ratio of 60:40, where 60% of the data is used for training, while the remaining 40% is used for testing. This separation is done by determining the separator index based on the length of the data using *split_ratio* = 0.6. After the separation, the training and testing data are reprocessed with the *create_features()* function, so that the *X_train*, *y_train* pairs are formed for training and *X_test*, *y_test* for testing.

2.2.4. Adjusting Data Dimensions for the GRU Model

Because the GRU model requires input in three-dimensional format, a reshaping process is carried out on the training and testing data using *np.reshape()*. This transformation changes the data dimensions to (number of samples, window length, number of features) so that they are in accordance with the needs of the GRU model.

2.3. Model Development

At this stage, the Gated Recurrent Unit (GRU) model is built which is designed to predict stock prices based on previously processed historical data.

2.3.1. Model Architecture

The developed model consists of several main layers, namely:

a) The first GRU layer with 128 neuron units using the ReLU activation function. This layer has a *return_sequences=True* parameter, which allows data to be forwarded to the next GRU layer.
b) The dropout layer is 0.2, which functions to reduce the risk of overfitting by randomly ignoring a number of units during the training process.
c) The second GRU layer with 128 neuron units using the ReLU activation function, but with *return_sequences=False* because this is the last layer of the GRU network before entering the fully connected layer.
d) The second dropout layer is 0.2, to increase model generalization.
e) The output layer (Dense layer) with 1 neuron unit, which functions to generate stock price predictions based on patterns that have been learned from previous data.

2.3.2. Model Compilation

The model is compiled using the Mean Squared Error (MSE) loss function, which is a commonly used metric in regression problems to measure the difference between actual and predicted values. In addition, model optimization is performed using the Adam algorithm with a learning rate of 0.0001. The selection of this optimizer is based on its superiority in adaptively adjusting parameters, thereby accelerating the model convergence process.

2.3.3. Regularization and Callbacks Strategy

To improve the stability and performance of the model during training, several regularization techniques and an early stopping strategy are used through the callbacks mechanism, EarlyStopping, which monitors the validation loss value and will stop training if there is no improvement for 5 consecutive epochs (*patience* = 5). This prevents overfitting and inefficient use of computing time. In addition, the model will use the best weights before the process stops (*restore_best_weights* = True). ModelCheckpoint, which aims to save the model with the best performance based on the lowest validation loss value (*save_best*_only = True). This model will be saved in the best_model.h5 file to ensure that the best model is used in the subsequent evaluation and prediction process.

2.4. Model Training

The model training process is carried out using a previously processed dataset. The model is trained for 30 epochs with a batch size of 20 to ensure that the learning process takes place optimally. The training data is used to update the model weights, while the testing data is used as validation data to evaluate the model's performance on previously unseen data. During training, the model uses callbacks techniques, namely EarlyStopping to stop training if the validation loss does not show improvement in 5 consecutive epochs, and ModelCheckpoint to save the best model weights. In addition, the *shuffle=False* option is applied to maintain the data order, considering that the model learns sequential patterns in historical stock price data.

2.5. Model Evaluation

To evaluate the accuracy of the model, evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2 Score) are used. Each of these metrics is calculated using the following formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \quad (1)$$

$$RMSE = \sqrt{MSE} \quad (2)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \tilde{y}_i| \quad (3)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \tilde{y}_i}{y_i}\right| \times 100\% \quad (4)$$

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y}_i)^2} \quad (5)$$

Where y_i is the actual value, $\tilde{y}_i$ is the predicted value, $\bar{y}_i$ is the average of the actual values and n is the number of samples.

3. Results and Discussion

3.1. Training and Validation Analysis

The results of model training are shown in the loss function graph which will be displayed in Figure 2.

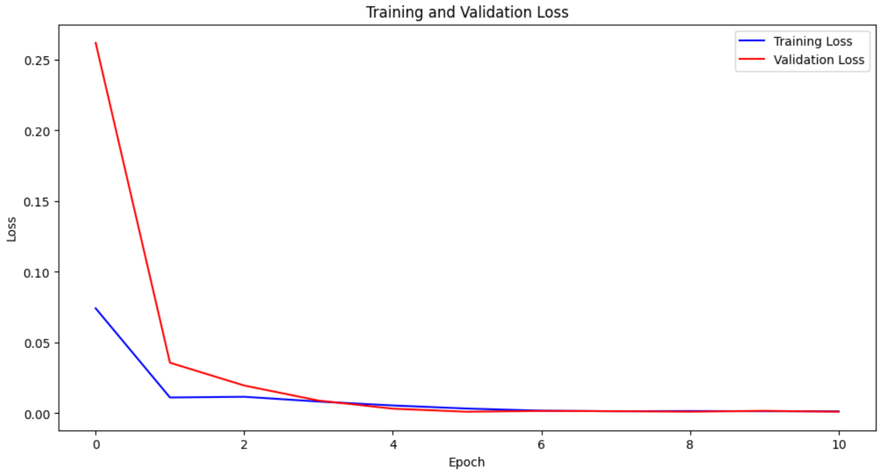

Figure 2: Model training process

At the beginning of training, the loss value for the validation data (red line) is much higher than the training loss (blue line). However, as the number of epochs increases, both losses decrease significantly, indicating that the model successfully learns from the given data. After a few epochs, both training loss and validation loss tend to approach zero and become stable. The model was originally planned to be trained for 30 epochs, but the training process was stopped early due to the early stopping mechanism. This indicates that the model has converged well without any significant indication of overfitting, as there is no large gap between the training loss and validation loss at the end of training.

3.2. Model Evaluation

Model evaluation is conducted using several main evaluation metrics, namely Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2 Score). The results of the model evaluation can be seen in Table 1.

Table 1: Results of stock price prediction model evaluation

Evaluation Metrics	Mark
Mean Squared Error (MSE)	24,896.4120
Root Mean Squared Error (RMSE)	1.577.860
Mean Absolute Error (MAE)	1.242.339
Mean Absolute Percentage Error (MAPE)	2.04%
R-squared (R^2 Score)	0.9636

Based on the evaluation results, the model shows very good performance with an MSE value of 24,896.4120 and an RMSE of 157.7860, indicating that the average squared error in the prediction is relatively small. In addition, the MAE value of 124.2339 indicates that the average absolute error of the model is quite low. The MAPE reaching 2.04% indicates that the relative error in the prediction is quite small, so the model has a good level of accuracy. The R-squared (R^2) value of 0.9636 indicates that the model can explain 96.36% of the variability of stock price data, indicating that the model has a high level of fit to the actual data.

3.3. Visualization of Prediction Results

The comparison between the actual stock price and the model prediction on the training data and testing data will be shown in Figure 3.

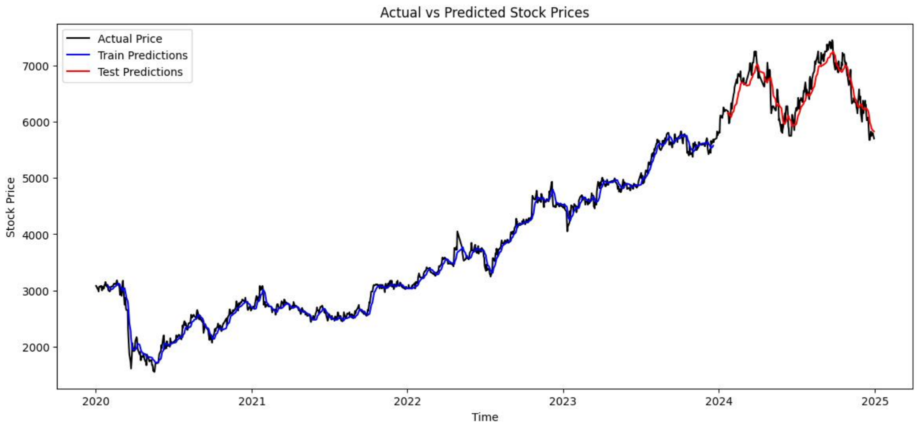

Figure 3: Comparison of actual and predicted prices

Looking at the comparison between actual and predicted prices in the second graph, the model shows excellent tracking ability. The predicted lines (blue for training and red for testing) are very close to the actual price movements (black). This is especially evident in the 2024-2025 period where the model is able to follow the increasing volatility well. Even in periods with more volatile movements, such as the sharp decline in early 2020 and the significant increase in 2024, the model is still able to produce accurate predictions.

4. Conclussion

The Gated Recurrent Unit (GRU) model is able to produce predictions with a very good level of accuracy. Key evaluation values such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2 Score) indicate that the model is able to capture stock price patterns well. Specifically, the R^2 Score value of 0.9636 indicates that the model can explain around 96.36% of stock price variability based on historical data.

References

Fleischer, J. P., von Laszewski, G., Theran, C., & Parra Bautista, Y. J. (2022). Time series analysis of cryptocurrency prices using long short-term memory. *Algorithms*, *15*(7), 230.

Ghadimpour, M., & Ebrahimi, S. B. (2022). Forecasting Financial Time Series Using Deep Learning Networks: Evidence from Long-Short Term Memory and Gated Recurrent Unit. *Iranian Journal of Finance*, *6*(4), 81-94.

Lee, T., & Singh, V. P. (2023). Deep learning: Long short-term memory in hydrological time series. In *Handbook of Hydroinformatics* (pp. 49-67). Elsevier.

Lewinson, E. (2022). *Python for Finance Cookbook: Over 80 powerful recipes for effective financial data analysis*. Packt Publishing Ltd.

Li, X. (2023). CNN-GRU model based on attention mechanism for large-scale energy storage optimization in smart grid. *Frontiers in Energy Research*, *11*, 1228256.

Martina, A., & Girana, I. (2024). Application of Singular Spectrum Analysis (SSA) Decomposition in Artificial Neural Network (ANN) Forecasting. *International Journal on Information and Communication Technology (IJoICT)*, *10*(1), 13-27.

Oğuz, A., & Ertuğrul, Ö. F. (2023). Introduction to deep learning and diagnosis in medicine. In *Diagnostic Biomedical Signal and Image Processing Applications with Deep Learning Methods* (pp. 1-40). Academic Press.

Sharma, A., Guha, D., Agarwal, H., & Harshadbhai, K. M. Stock Market Prediction and Investment using Deep Reinforcement Learning-a Continuous Training Pipeline.

Soni, P., Tewari, Y., & Krishnan, D. (2022). Machine learning approaches in stock price prediction: a systematic review. In *Journal of Physics: Conference Series* (Vol. 2161, No. 1, p. 012065). IOP Publishing.

Wang, C., Du, W., Zhu, Z., & Yue, Z. (2020). The real-time big data processing method based on LSTM or GRU for the smart job shop production process. *Journal of Algorithms & Computational Technology*, *14*, 1748302620962390.