Scientific Journal of Informatics
Vol. 10, No. 3, Aug 2023

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Comparison of ARIMA and GRU Models for High-Frequency Time Series Forecasting

Mochamad Ridwan[1], Kusman Sadik[2*], Farit Mochamad Afendi[3]
[1,2,3]Department of Statistics, Faculty of Mathematics and Natural Sciences,
IPB University, Indonesia

Abstract.
Purpose: The purpose of this research is to assess the efficacy of ARIMA and GRU models in forecasting high-frequency stock price data, specifically minute-level stock data from HIMBARA banks. In time series analysis, time series data exhibit interesting interdependence among observations. Despite its popularity in time series forecasting, the ARIMA model has limitations in capturing complicated nonlinear patterns. Forecasting high-frequency data is becoming more popular as technology advances and more high-frequency data becomes available.
Methods: In this study, we compare the ARIMA and GRU models in forecasting minute-level stock prices of HIMBARA banks. The data used consists of 62,921 minute-level stock data points for each bank in the HIMBARA group, collected in the year 2022. The GRU model was chosen because it is capable of capturing complex nonlinear patterns in time series data. Each method's predicting performance is assessed using the Mean Absolute Percentage Error (MAPE) statistic.
Results: In terms of forecasting accuracy, the GRU model outperforms the ARIMA model. The GRU model achieves a MAPE of 0.77% for BMRI stock, while the ARIMA model achieves a MAPE of 4.09%. The GRU model predicts a MAPE of 0.34% for BBRI stock, while the ARIMA model predicts a MAPE of 3.02%. For BBNI stock, the GRU model obtains a MAPE of 0.63%, while the ARIMA model achieves a MAPE of 1.52%. The GRU model achieves a MAPE of 0.58% for BBTN stock, while the ARIMA model achieves a MAPE of 6.2%.
Novelty: In terms of minute-level time series data modeling, research in Indonesia has been limited. This study adds a new perspective to the discussion by comparing two modeling approaches: the traditional ARIMA model and the sophisticated deep learning GRU model, both of which are applied to high-frequency data. Beyond the present scope, there are several promising future directions to pursue, such as anticipating intraday stock fluctuations. This unexplored zone not only contributes to the field of financial modeling but also has the ability to uncover intricate patterns in minute-level data, an area that has not been extensively studied in the Indonesian context.

Keywords: ARIMA, GRU, Time series, Forecasting, High-frequency data.

Received June 2023 / **Revised** July 2023 / **Accepted** August 2023

INTRODUCTION

Time series data is a sequence of observations collected over a defined time period [1]. Time series data is common in many sectors, including economics, business, engineering, natural sciences, and social sciences. The interdependence of closely spaced observations is a feature of time series data, making the study of their dependency patterns highly attractive. Box and Jenkins established and developed time series analysis in 1970, and it has since become widely utilized for anticipating future data. Forecasting is a future event prediction or estimation activity that acts as a tool for successful and efficient planning [2], [3].

Financial time series are the subject of a study that has been substantially developed in time series analysis. However, previous research has primarily focused on low-frequency data (daily, weekly, and monthly) for modeling and forecasting reasons. With the expansion of trade recording and the availability of financial data acquired at higher frequencies in recent decades, it has grabbed the attention of scholars and developed as a new research topic in econometrics and statistics [4]. The growing dominance of electronic trading

*Corresponding author.
Email addresses: moch.ridwan@apps.ipb.ac.id (Ridwan)
DOI: 10.15294/sji.v10i3.45965

enables the recording of market activities at high frequencies and with a high level of precision, resulting in comprehensive datasets. The increasing popularity of studies on high-frequency data is also driven by technological advancements in trading systems, particularly in trade recording, and the rising importance of intra-day trading.

The association between model flexibility and precision level is significantly impacted by data sample size. Shorter time intervals can be employed to improve the available sample size and hence enlarge the data sample. High-frequency data refers to data at the transactional level that has the maximum possible frequency [5]. The main characteristic of high-frequency data, especially in trading transactions, is the irregular spacing of time, also known as irregularly spaced time. As an example, the average stock in the Russell 3000 has around 2,100 ticks every day. As a result, the size of the HFD dataset for common stocks in the Russell 3000 in one day can be 2,100 times bigger than the size of the same day's closing data [5].

Several time series modeling methods have been extensively developed, and one of the most commonly used methods is ARIMA (Autoregressive Integrated Moving Average). ARIMA is capable of analyzing univariate data that contains trends, seasonality, and cyclic patterns [6]. The correlation structures in time series data are both linear and nonlinear. The ARIMA approach can address linear correlations between variables, but it cannot capture nonlinear patterns [7]. High-frequency financial time series exhibit unique characteristics, such as variations over time and a tendency towards nonlinearity. These characteristics are accompanied by significant stochastic volatility in the sample data, making forecasting challenging, especially when relying solely on the ARIMA model [8]. Stochastic volatility in financial markets is depicted as the process of fluctuation in the rate of returns that cannot be predicted using available information. It refers to the unpredictable and random nature of volatility in asset prices and the difficulty of forecasting future volatility accurately. Stochastic volatility models are designed to capture this inherent uncertainty in financial markets and are used to improve the accuracy of forecasting asset prices and risk management [9].

Artificial intelligence technology is rapidly advancing, and one of the branches that has garnered significant attention from researchers is machine learning. Deep learning, a crucial part of machine learning, involves algorithms that model high-level abstractions in data by using a series of layered and deep non-linear transformation functions. Deep learning models are built with numerous layers of neurons to automatically find patterns and crucial elements in complex data. The capability of deep learning to process intricate and extensive datasets makes it highly effective in various applications, such as facial recognition, image analysis [10], natural language processing, and more. With the advancements in this technology, there is great hope for utilizing artificial intelligence to enhance various aspects of human life and advance various fields of science [11]. Models in deep learning are fundamentally built based on Artificial Neural Networks (ANNs), a research area that has been active since the 1980s. However, it has recently experienced a significant resurgence due to the advancements in faster computers and parallel processing capabilities, particularly utilizing GPUs (Graphics Processing Units) [12]. Time series data forecasting can also be accomplished using the Artificial Neural Network (ANN) approach. A commonly used neural network method for processing time series data is the Recurrent Neural Network (RNN). However, RNN faces a major issue known as vanishing gradient, where the gradients of the function decrease exponentially when processing long sequential data. This problem hinders RNNs from capturing long-term dependencies, leading to reduced forecasting accuracy [13].

Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models were created to solve the limitations of RNN. Both the LSTM and GRU models are intended to avoid the vanishing gradient problem that frequently happens in standard RNNs while processing extended data sequences. The vanishing gradient problem reduces the model's capacity to retain important information from earlier time steps, which is critical for capturing the long-term dependencies of time series data. In a study by Yamak et al., which compared GRU and LSTM for time series forecasting on a Bitcoin dataset, GRU demonstrated superior performance compared to LSTM [14]. In the study conducted by Chung et al., various types of RNN variants, particularly focusing on LSTM and GRU, were compared in the context of polyphonic music modeling and sound signal modeling. The results showed that both LSTM and GRU models were capable of providing good accuracy in time series forecasting for these specific applications [15]. GRU was first introduced in 2014 by Kyung Hyun Cho as a variation of LSTM. One of the main advantages of GRU over LSTM is its simpler computation, which makes it more computationally efficient and faster to train. Despite its simplicity, GRU has shown comparable accuracy to LSTM in various tasks, making it an attractive alternative for time series modeling and other sequential data tasks. Additionally, GRU is known to expose more complete memory from the hidden layers compared to LSTM. This ability to capture and retain relevant information from past time steps allows GRU to effectively model long-range dependencies in the data, just like LSTM [16].

Referring to the background, it is highlighted that the ARIMA model has certain limitations, including large deviations in forecasting high-frequency time series data. High-frequency financial time series exhibit characteristics such as time-varying variations, non-linear behavior, and significant stochastic volatility. Some studies have also shown that in certain cases, GRU performs better than LSTM. Given this context, comparing ARIMA and GRU becomes an interesting area of research, especially for high-frequency data. Moreover, there is a lack of research on the use of GRU models for forecasting in Indonesia.

The purpose of this study is to apply the ARIMA and GRU models to minute-level stock data from HIMBARA (The Association of State-Owned Regional Development Banks) member banks and assess forecast accuracy using MAPE (Mean Absolute Percentage Error). Minute-level stock data falls under high-frequency data as it is recorded and updated every minute during stock trading hours, providing a higher level of detail to reflect real-time stock price changes. Similar research with this specific focus is scarce in Indonesia, making this study valuable in providing new insights and contributing to the development of more accurate forecasting methods for high-frequency data.

METHODS

Data Collection

The empirical data used in this study are stock price data from four large-capitalized banks that are members of The Association of State-Owned Regional Development Banks (HIMBARA). Banks in this category include Bank Mandiri, Bank BNI, Bank BTN, and Bank BRI. The study's stock price data is collected at 1-minute intervals from January 1, 2022, to December 31, 2022, obtaining 59,532 data points for each bank.

High-Frequency Data

High-frequency data, often known as tick-by-tick data, is information captured at the individual transaction level [5]. In high-frequency data, information is recorded and updated each time there is a price change, which is commonly referred to as a "tick." Each tick represents a single transaction or price update in the market, and it provides real-time information on the price movements and trading activity of an asset or security. High-frequency data is obtained directly from the market, capturing the smallest price fluctuations and allowing for a more detailed analysis of intraday trading behavior and market dynamics [17]. The book "Encyclopedia Of Financial Models" explains that there are two main reasons for the interest in high-frequency data. Firstly, such data allows for capturing intriguing events and phenomena in the stock market with remarkable precision. It enables financial institutions to measure intraday trading risk accurately and identify short-term trading opportunities, thereby enhancing their decision-making processes. Second, high-frequency data can be used to increase forecasting model accuracy [5]. By having direct access to more detailed transaction information from the market, forecasting models can be built to consider rapid price changes. In this context, analysis using high-frequency data can help generate more accurate predictions about stock prices and other financial assets' movements.

High-frequency data has been the principal focus of research for individuals seeking a more in-depth understanding of the financial market. This type of data has advantages in explaining micro market structures, involving trading rules that influence stock prices in the capital market [18]. According to the book "An Introduction To High-Frequency Finance," high-frequency data refers to a significant volume of data, with the number of observations in one day of the stock market being similar to the daily data in 30 years [17]. High-frequency data allows for the examination of the financial market at various periods, ranging from minutes to years, and can reflect aggregation factors of four to five times [17].

Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average (ARIMA) is a popular statistical method for forecasting time series data, particularly non-stationary data. A time series data, denoted by Y_t , is said to follow an ARIMA model if the d-th difference of $W_t = \nabla^d Y_t$ follows an Autoregressive Moving Average (ARMA) process. If $\{W_t\}$ follows an ARMA(p,q) model, then $\{Y_t\}$ is an ARIMA(p,d,q) process, where p denotes the autoregressive parameter order, d the differencing order, and q the moving average parameter order. When a time series can be written in the following equation form, it is said to follow an ARIMA(p,d,q) process.:

$$Y_t = (1 + \emptyset_1)Y_{t-1} + (\emptyset_2 - \emptyset_1)Y_{t-2} + \cdots + (\emptyset_p - \emptyset_{p-1})Y_{t-p} - \emptyset_p Y_{t-p-1} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{1}$$

with parameters ϕ and θ, which are the autoregressive and moving average parameters that are stationary and invertible, respectively, and ε_t is *white noise* [19].

ARIMA uses numerous iterative phases to get the optimum model. The initial phase is model identification and selection, and for non-stationary data, differencing may be conducted once or twice to establish stationarity. When the data has reached stationary, the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots are evaluated to determine the AR and MA model orders. The second phase is parameter estimation, which entails optimization techniques that make use of metrics like the Akaike Information Criterion (AIC) and/or the Bayesian Information Criterion (BIC). Finally, a diagnostic checking phase is performed to find the optimal final model based on residual analysis. ARIMA is a data-driven linear technique that adapts time series data parameters. As a result, nonlinearity in the data has a major impact on the ARIMA model's performance. This is a restriction of the ARIMA model because major non-linear data patterns might limit the ARIMA model's usefulness.

Gated Recurrent Unit (GRU)

Kyunghyun Cho was the first to introduce GRU in 2014. It is an RNN-based algorithm that is comparable to LSTM but has a simpler architecture. The fundamental problem of RNN is the vanishing and exploding gradient issue, which occurs due to continuous multiplication during Backpropagation Through Time (BPTT) [20]. GRU addresses this problem by using gates, namely the update gate and reset gate, as shown in Figure 1.

The initial stage in developing a GRU model is to compute the update gate (Z_t) using the formula in equation (2), which defines how much past information must be preserved.

$$z_t = \sigma(w^{(t)}x_t + u^{(t)}h_{t-1} + b) \tag{2}$$

where w and u are weights, x_t is the input, , h_{t-1} is the hidden state, and b is the bias.

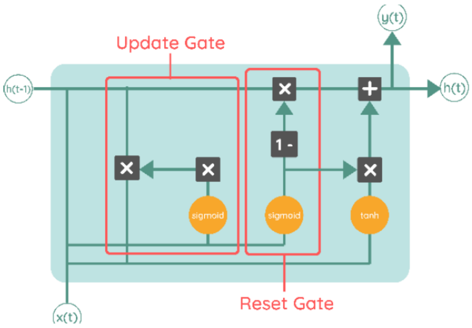


Figure 1. GRU architecture

The reset gate (r_t) is then calculated using equation (3), which defines how much of the prior information should be deleted and how to combine the incoming input with the old information. The reset gate formula is as follows :

$$r_t = \sigma(w^{(r)}x_t + u^{(r)}h_{t-1} + b) \tag{3}$$

Next, compute the candidate hidden state (h'_t) that will be used by the reset gate to retain important information from the past. Equation (4) expresses the candidate hidden state :

$$h'_t = tanh\,(wx_t + r_t \odot uh_{t-1}) \tag{4}$$

where $\odot$ is the hadamard product.
The final step is to compute the hidden state (h_t) using equation (5). This hidden state serves as the output (y_t)

$$h_t = z_t \odot h_{t-1} (1 - z_t) \odot h'_t \tag{5}$$

GRU features various hyperparameters that might affect prediction results, such as the number of hidden layers, batch size, and learning rate drop. The number of hidden layers denotes the depth of the training process, batch size indicates how frequently the weights are updated, and learning rate drop denotes the number of iterations used to establish the learning rate [20].

METHODS

The flow of this research can be seen in Figure 2.

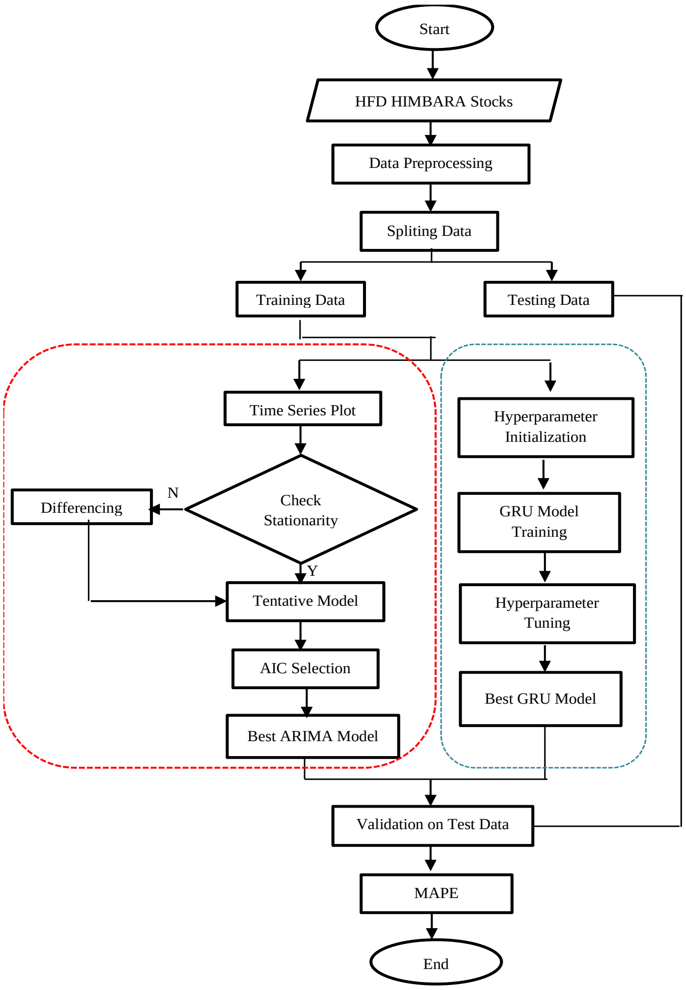


Figure 2. Stage of research

Preprocessing

This stage entails cleaning the data and dividing it into training and testing data. The data cleaning process involves removing data outside the trading hours of the Indonesia Stock Exchange (BEI), such as deleting data on holidays (Saturdays, Sundays, and public holidays) as well as collective leave days that occurred in 2022. BEI trading hours are from 09:00 to 15:00, with a break from 11:30 to 13:30. The dataset is divided into training and testing data in order to evaluate the forecasting model. The training data consists of closing stock prices from January 1, 2022, to October 31, 2022, while the testing data consists of closing stock prices from November 1, 2022, to December 31, 2022. The primary purpose of separating the data into training and testing subsets is to assess how effectively the model can generalize patterns and trends within the data. By dividing the data into two independent groups, we can measure how well the model can apply what it has learnt from the training data to the testing data. The splitting of training and testing data is particularly beneficial in detecting overfitting, which occurs when a model performs well on training but badly on testing data.

Model Accuracy

One of the crucial stages in forecasting is the process of evaluating forecast accuracy. Forecast accuracy can be measured by evaluating how well a model works when applied to fresh data that was not utilized in the model's development [21]. The model's accuracy can be measured using numerous metrics, including root mean square error (RMSE), mean absolute deviation (MAD), and mean absolute percentage error (MAPE). The accuracy measures are determined using the following formulas based on forecast errors :

$$e_t = y_t - \hat{y}_t \tag{6}$$

where y_t represents the actual data value at time t dan $\hat{y}_t$ represents the forecasted (predicted) value at time t. This study uses the MAPE (Mean Absolute Percentage Error) as a measure of forecast accuracy. The formula to calculate the MAPE is as follows [22]:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{e_t}{y_t} \times 100\right| \tag{7}$$

RESULTS AND DISCUSSIONS

Stationaryity

The plot in Figure 3 shows that the stock price data of the four banks does not exhibit stationarity in terms of mean and variance. To address this non-stationarity, the data needs to undergo a differencing process. Differencing attempts to change data into a stationary form by calculating the difference between observed values at one time and observed values at a prior time. By performing differencing, we can remove trends and patterns that may exist in the original data and generate a new set of data that is more stationary. Stationary data has constant statistical properties over time, such as a constant mean and variance. The differencing process can help make the data easier to analyze and use in forecasting models.

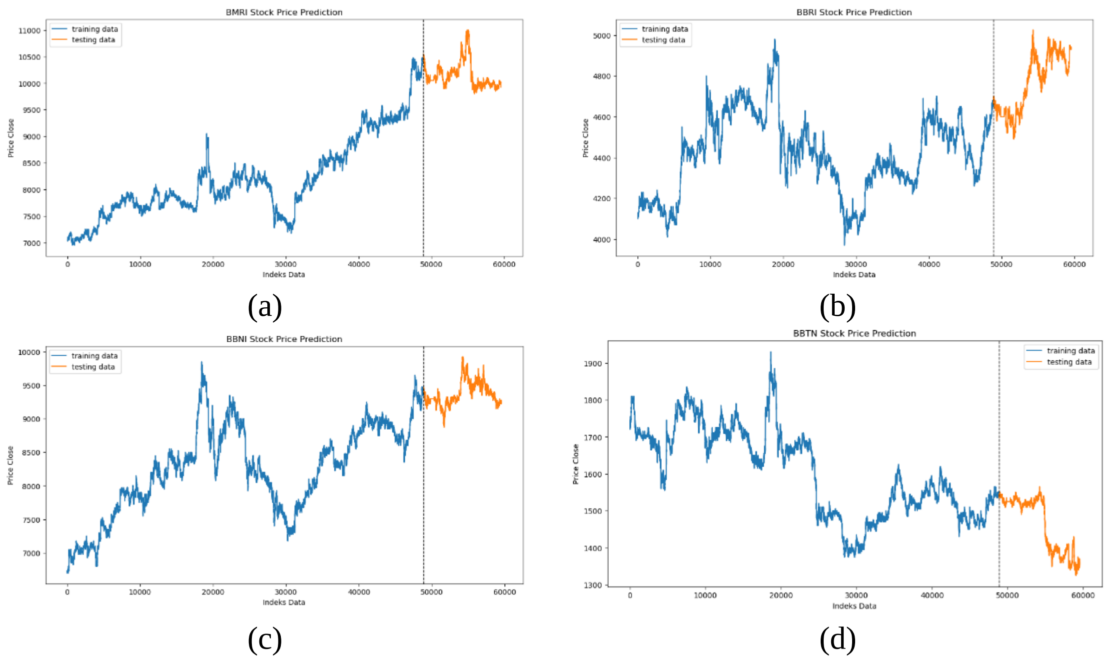


Figure 3. HIMBARA bank stock training and test data plots.
(a) BMRI, (b) BBRI, (c) BBNI, (d) BBTN.

Results of the Augmented Dickey-Fuller (ADF) test show that the training data from each bank is not significant at the significance level $\alpha = 0.05$, suggesting that the data is non-stationary. The ADF test plays a crucial role in ARIMA modeling as it helps to determine whether differencing is required before applying the ARIMA model to the time series data. In this research, the ADF test is run on the training data both before and after first-order differencing. Table 1 displays the results of the ADF test on the training data and the training data after first-order differencing.

Table 1. Augmented dickey-fuller (ADF) test results

Closing Stock		P-Value	Description
BMRI	Closing Stock	0,9633	Not Stationary
	Closing Stock D1	< 0,000	Stationary
BBRI	Closing Stock	0,1325	Not Stationary
	Closing Stock D1	< 0,000	Stationary
BBNI	Closing Stock	0,2978	Not Stationary
	Closing Stock D1	< 0,000	Stationary
BBTN	Closing Stock	0,4177	Not Stationary
	Closing Stock D1	< 0,000	Stationary

ARIMA Modeling

Several factors are used as guidance in selecting the optimum ARIMA model. The Akaike Information Criterion (AIC) is a popular metric for picking the best model. AIC combines the goodness of fit of the model and its complexity, therefore a model with a lower AIC value is deemed better. By comparing the AIC values of different ARIMA models, we can choose the most optimal model. Table 2 provides information about the best ARIMA orders for each member bank of HIMBARA, including the values of p, d, and q, which indicate the number of autoregressive order, differencing, and moving average order in the optimal ARIMA model. Using this information, we can build the appropriate ARIMA model for forecasting the closing stock prices of each bank with the expected accuracy.

Table 2. The best ARIMA models for the stocks of banks in HIMBARA

Closing Stocks	Best Model	AIC	MAPE (%)
BMRI	ARIMA(1,1,2)	416091	4,09
BBRI	ARIMA(5,1,1)	331666	3,02
BBNI	ARIMA(3,1,1)	417236	1,52
BBTN	ARIMA(2,1,1)	250720	6,20

Table 3 shows the Ljung-Box test values for the residuals of the four ARIMA models. The Ljung-Box test is used to determine the presence of autocorrelation in the ARIMA model residuals. The Ljung-Box test findings demonstrate that the p-values for the residuals in all four models exceed the predefined significance level of 0.05. This shows that there is insufficient evidence to reject the null hypothesis, implying that there is no significant autocorrelation in the ARIMA model residuals. Knowing that there is no significant autocorrelation in the ARIMA model residuals provides useful information about the underlying data structure and trends. This data is extremely useful for modeling and forecasting future data since it allows us to create more accurate models and make better forecasts. As a result, performing the Ljung-Box test for autocorrelation in the ARIMA model residuals provides a more in-depth understanding of the data and aids in the construction of more effective models.

Table 3. Ljung-box test for the residuals of the ARIMA model

Stocks	Best Model	Ljung-Box	P-Value
BMRI	ARIMA(1,1,2)	4,506	0,720
BBRI	ARIMA(5,1,1)	5,855	0,210
BBNI	ARIMA(3,1,1)	11,246	0,081
BBTN	ARIMA(2,1,1)	8,905	0,260

GRU Modeling

This research uses various combinations of hyperparameter settings to build the GRU model. Some important hyperparameters that influence the performance and learning of the GRU model are the number of neurons, learning rate, and epochs. The number of neurons refers to the number of units or "cells" in the GRU network. It can affect the capacity and complexity of the model. Too few neurons may not be sufficient to capture complex patterns in the data, while too many neurons can lead to overfitting. The learning rate controls how fast or slow the model updates its weights and biases during the learning process. It is essential in determining the size of the steps taken by the optimization algorithm when searching for the minimum or maximum of the objective function. Epochs refer to the number of times the entire dataset is passed through the model during the learning process. Each epoch consists of one forward and backward pass on the training data. To avoid overfitting, the number of epochs is adjusted using the early stopping mechanism. This technique stops the model training early if there is no significant improvement in performance or reduction in error. It prevents overfitting, saves time and computational resources, and can result in a more stable model. The combinations of hyperparameters used in this study can be seen in Table 4.

Table 4. GRU hyperparameter

Hyperparameter	Value
Look Back	50
Validation Split	0,2
Dropout	0,2
Jumlah Neuron	[32, 64, 128]
Learning rate	[0.001, 0.0001]
Epochs	[500] *Early Stop*

In the GRU model, a look-back value of 50 means that the model uses the past 50 time steps as inputs to predict the next time step. The Dropout value used is 0.2, which randomly drops out a portion of neurons during the training process. By applying Dropout, the GRU model becomes more robust and can generalize well to new data. The validation split value used is 0.2, which means that 20% of the training data is used as validation data during the training process of the GRU model.

These combinations of hyperparameter settings are applied to the training data of the four HIMBARA bank stocks, and the forecasting accuracy is tested using the test data. Table 5 shows the MAPE values on the test data for the GRU models of the four bank stocks. The best MAPE value for the BMRI stock using the GRU model is 0,77 with 128 neurons, a learning rate of 0.0001, and stopping at epoch 26. The BBRI stock gives a MAPE value of 0,34 with 32 neurons, a learning rate of 0.0001, and stopping at epoch 39. The BBNI stock gives a MAPE value of 0,63 with the same hyperparameter combination as BBRI, but stopping at epoch 21. The BBTN stock gives a MAPE value of 0,58 with 64 neurons, a learning rate of 0.0001, and stopping at epoch 61. The GRU model with the appropriate hyperparameter settings can be used to forecast HIMBARA bank stock time series data with good accuracy.

Table 5. MAPE value of the GRU model of HIMBARA bank

Stocks	Number of Neurons	Learning Rate	Epochs (Early Stop)	MAPE (%)
BMRI	32	0,001	500 (16)	1,51
		0,0001	500 (24)	1,00
	64	0,001	500 (16)	1,12
		0,0001	500 (32)	0,84
	128	0,001	500 (16)	1,47
		0,0001	**500 (26)**	**0,77**
BBRI	**32**	0,001	500 (16)	0,96
		0,0001	**500 (39)**	**0,34**
	64	0,001	500 (17)	1,04
		0,0001	500 (32)	0,35
	128	0,001	500 (20)	11,00
		0,0001	500 (51)	0,36
BBNI	**32**	0,001	500 (15)	1,92
		0,0001	**500 (21)**	**0,63**
	64	0,001	500 (16)	1,50
		0,0001	500 (24)	0,65
	128	0,001	500 (16)	1,57
		0,0001	500 (23)	0,67
BBTN	32	0,001	500 (34)	2,25
		0,0001	500 (45)	0,62
	64	0,001	500 (38)	1,89
		0,0001	**500 (61)**	**0,58**
	128	0,001	500 (53)	1,55
		0,0001	500 (53)	0,58

Overall, the MAPE values for the four HIMBARA bank stocks using GRU are better compared to using ARIMA. This indicates that GRU is more adaptive to the data patterns of these four stocks compared to

ARIMA. BBRI and BBNI stocks use a smaller number of neurons, making them computationally lighter because the number of parameters to be calculated and optimized is reduced. All GRU models tested on the training data show that using a learning rate of 0.0001 is better than using 0.001. This suggests that a smaller learning rate leads to slower convergence of the model but has the potential to achieve more accurate and stable results. However, using an extremely small learning rate can make the training process too slow, and the model may struggle to reach convergence. Thus, the effect of a smaller learning rate is a trade-off between convergence speed and accuracy/stability of the model.

Model Validation

Figure 4 shows the train loss and validation loss graphs of the best model. The train loss graphs of all best models indicate a stable decreasing trend as the training progresses on the training data. This indicates that the models consistently learn the patterns in the training data and minimize prediction errors. The validation loss graphs also show a stable or consistent decreasing trend during the training on the training data. This indicates that the models can generalize well to the validation data, which they have never seen before. For the BBRI stock, the validation loss graph experiences fluctuations after a decrease at epoch 5, but it remains close to the train loss graph. The small difference between the train loss and validation loss graphs suggests that the model does not suffer from overfitting on the training data and can generalize well to new data. An ideal graph would show a stable decreasing trend for both train loss and validation loss, with a small difference between them. This indicates that the model has learned effectively, is not overfitting, and can generalize well to new data.

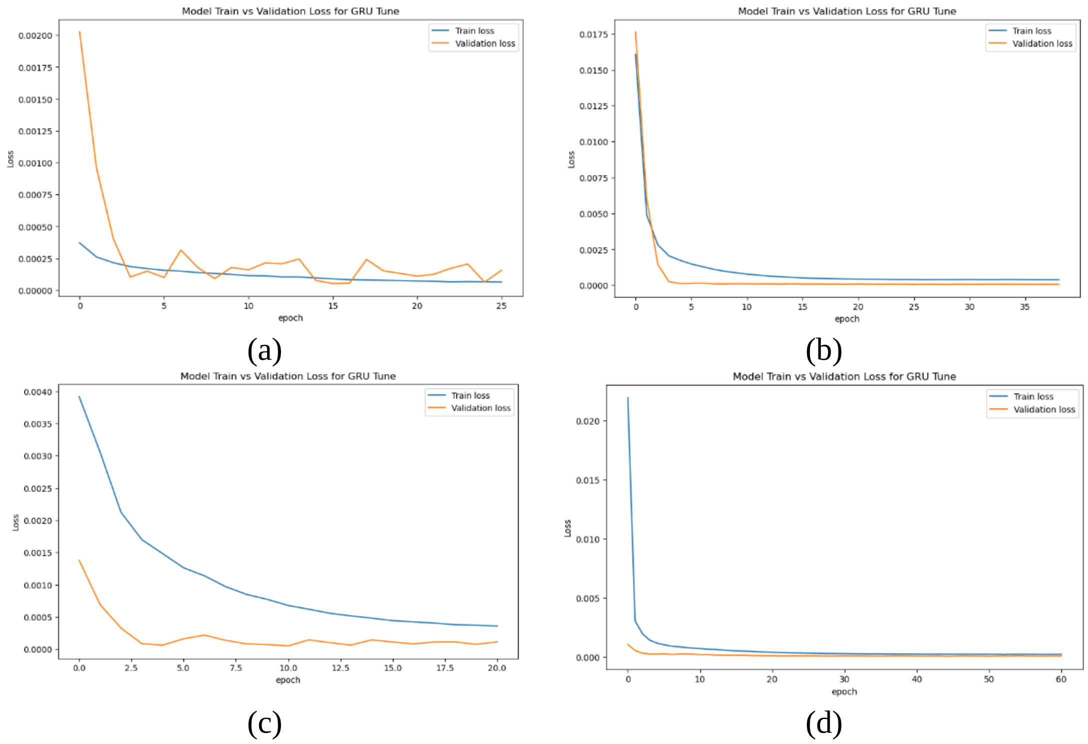


Figure 4. Train and validation loss for the best model of the HIMBARA bank stock.
(a) BMRI, (b) BBRI, (c) BBNI, (d) BBTN

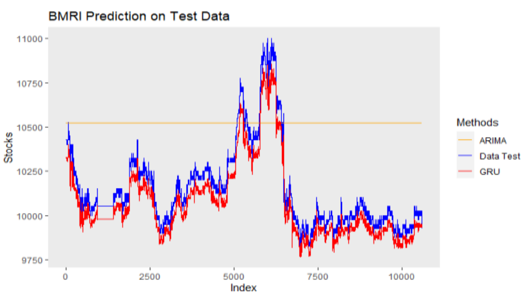
A comparison of the models (ARIMA and GRU) can be seen for each HIMBARA bank stock. The GRU model has a lower MAPE than the ARIMA model. This indicates that the GRU model is capable of providing more accurate predictions in modeling the data of the four bank stocks. For example, in the case of BBRI bank stock, the ARIMA(5,1,1) model gives a MAPE of 0.0302, whereas when the data is modeled with GRU, the MAPE drastically decreases to 0.0034. Table 7 shows the comparison of the three tested models, and it is evident that there is an improvement in forecasting accuracy from ARIMA to GRU for all HIMBARA bank stocks.

Table 6. Comparison of the models ARIMA and GRU

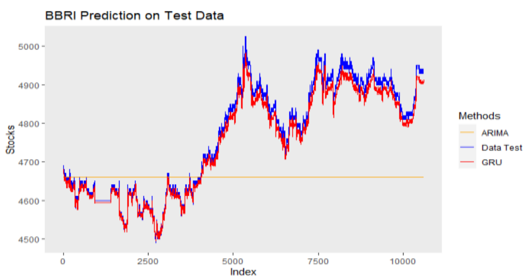
Stocks	Model	Number of Neurons	*Learning Rate*	*Epochs (Early Stop)*	MAPE (%)
BMRI	ARIMA(1,1,2)	-	-	-	4,09
	GRU	**128**	**0,0001**	**500 (26)**	**0,77**
BBRI	ARIMA(5,1,1)	-	-	-	3,02
	GRU	**32**	**0,0001**	**500 (39)**	**0,34**
BBNI	ARIMA(3,1,1)	-	-	-	1,52
	GRU	**32**	**0,0001**	**500 (21)**	**0,63**
BBTN	ARIMA(2,1,1)	-	-	-	6,20
	GRU	**64**	**0,0001**	**500 (61)**	**0,58**

Prediction of the Best Model

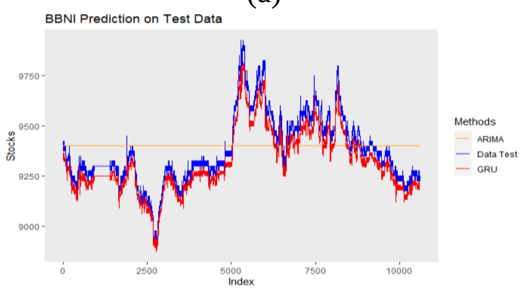
The best models of ARIMA and GRU that have been formed from the training data will be used to make predictions for the HIMBARA bank stock prices on the test data. The comparison of MAPE values has shown that the GRU model has the best prediction accuracy. The prediction plots of both models show a similar pattern. The prediction plots for the HIMBARA bank stock prices can be seen in Figures 5 and 6. Figure 5 shows the plot for the entire test data, while Figure 6 shows a zoomed-in view of a portion of the test data for the last day or the last 242 data points. It is evident from Figure 5 that the ARIMA model only produces a straight line for all stock prices. This is because the ARIMA model is designed to capture linear trend patterns in the data. In cases where the data has a clear linear trend, the ARIMA predictions may provide reasonably good results. However, the HIMBARA bank stock data does not have a clear trend, which causes the ARIMA prediction plot to be just a straight line.



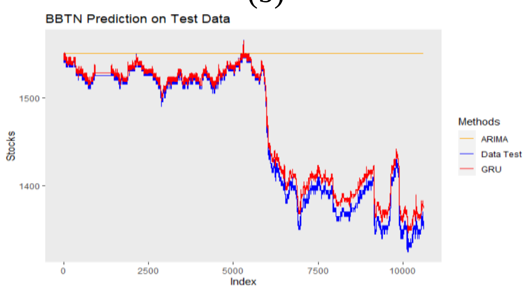
(a)



(b)



(c)



(d)

Figure 5. Prediction plot for all test data.
(a) BMRI, (b) BBRI, (c) BBNI, (d) BBTN

The use of the GRU model provides a better representation compared to ARIMA, where the GRU model can capture the data patterns in the test data. The GRU model is more adaptive in following the movements of the test data compared to ARIMA. The plots further emphasize that the GRU model is superior to the ARIMA model..

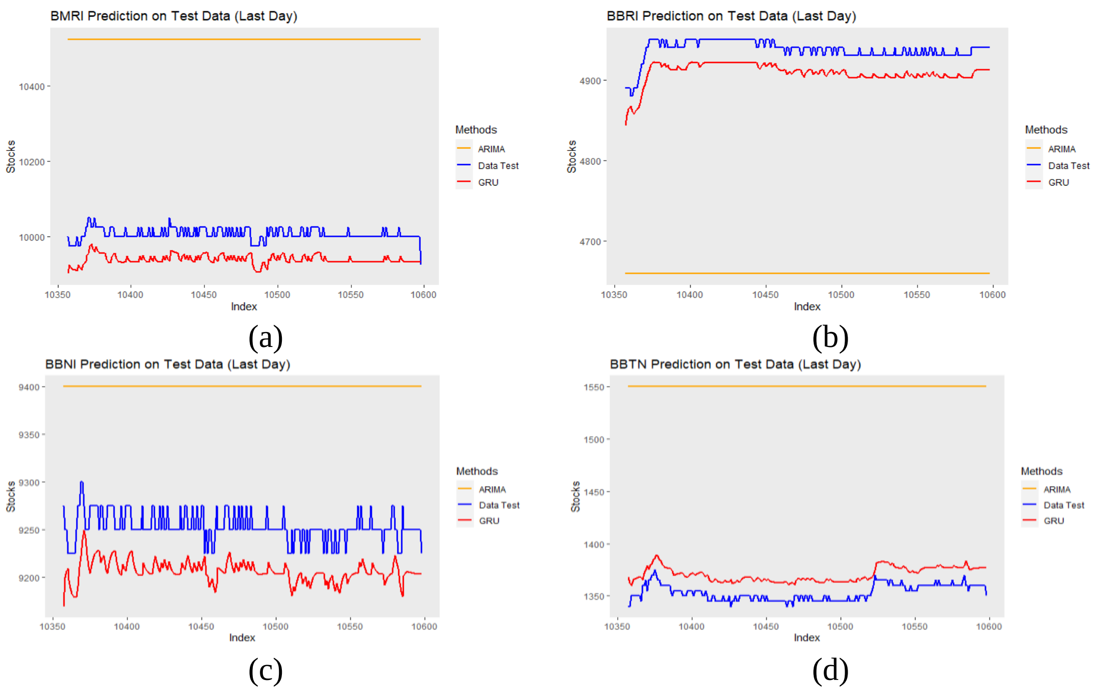


Figure 6. Zoomed-in prediction plot for the last day.
(a) BMRI, (b) BBRI, (c) BBNI, (d) BBTN

CONCLUSION

Based on the analysis, it can be concluded that using the GRU model is superior to the ARIMA model in forecasting high-frequency time series data, particularly in predicting the stock prices of HIMBARA member banks. This conclusion is supported by the comparison of MAPE values, where GRU provides lower MAPE values compared to ARIMA. The improved performance of GRU can be attributed to its ability to capture non-linear patterns that often appear in stock price time series data. While ARIMA has been widely used in time series forecasting, it has limitations in capturing more complex and non-linear patterns. On the other hand, GRU, being a recurrent neural network-based model, can adaptively adjust to complex and dynamic data patterns.

Analysis of prediction plots shows that GRU is more responsive to fluctuations and changes in the stock price data of HIMBARA banks. It accurately follows trends and provides predictions that are closer to the actual values. On the contrary, ARIMA tends to provide smoother predictions and is less responsive to sudden changes in data.

These results indicate that applying the GRU model to high-frequency time series data can offer advantages in terms of more accurate and adaptive forecasting. By utilizing GRU's ability to capture complex non-linear patterns, stock price modeling becomes more effective. In the context of applications in the financial and investment markets, choosing the right forecasting model is crucial to reduce risks and optimize investment outcomes. Therefore, methods like GRU that demonstrate better performance in modeling high-frequency stock price data can be a valuable contribution to practitioners and academics in conducting analysis and forecasting in the future.

REFERENCES

[1] R. Wulandari, B. Surarso, Irawanto, and F. Farikhin., "The Forecasting of Palm Oil Based on Fuzzy Time Series-Two Factor," *J. Soft Comput. Explor. 2*, vol. 1, pp. 11–16, 2021.

[2] I. Aksan, K. Nurfadilah, P. S. Matematika, U. Islam, and N. Alauddin, "Aplikasi Metode Arima Box-Jenkins Untuk Meramalkan Penggunaan Harian Data Seluler," vol. 2, no. 1, pp. 5–10, 2020.

[3] W. F. Abror and M. Aziz, "Journal of Information System Bankruptcy Prediction Using Genetic Algorithm-Support Vector Machine (GA-SVM) Feature Selection and Stacking," vol. 1, no. 2, pp. 103–108, 2023.

[4] N. Hautsch, *Econometrics of financial high-frequency data*. Springer Science & Business Media, 2012.

[5] F. J. Fabozzi, *Encyclopedia of Financial Models*. John Wiley & Sons, 2012.

[6] Paridi, “Perbandingan Metode ARIMA (Box Jenkins), ARFIMA, Regresi Spektral dan SSA dalam Peramalan Jumlah Kasus Demam Berdarah Dengue di Rumah Sakit Hasan Sadikin Bandun,” *JISIP (Jurnal Ilmu Sos. dan Pendidikan)*, vol. 3, 2019.
[7] A. O. B. Puka, “Model Hybrid ARIMAX-QR dan QRNN Untuk Peramalan Inflow dan Outflow Uang Kartal di Bank Indonesia Provinsi NTT dan Nasional,” pp. 20–51, 2017.
[8] Z. Li, “On the forecasting of high-frequency financial time series based on ARIMA model improved by deep learning,” no. December 2019, pp. 1–17, 2020, doi: 10.1002/for.2677.
[9] D. C. Lesmana and R. Budiarti, “Penentuan Harga Opsi Dengan Volatilitas Stokastik Menggunakan Metode Monte Carlo,” vol. 3, no. 1, pp. 80–92, 2021.
[10] K. L. Kohsasih and B. H. Hayadi, “Classification SARS-CoV-2 Disease based on CT-Scan Image Using Convolutional Neural Network,” *Sci. J. Informatics*, vol. 9, no. 2, pp. 197–204, 2022, doi: 10.15294/sji.v9i2.36583.
[11] A. A. P. Karo, “IMPLEMENTASI DEEP LEARNING PADA ROBOT KRAKATAU FC UNTUK MENENTUKAN SISI TIANG GAWANG PADA KONTES ROBOT HUMANOID INDONESIA,” *Univ. Teknokr. Indones.*, 2020.
[12] I. Cholissodin, A. A. Soebroto, U. Hasanah, and Y. I. Febiola, *AI, Machine Learning & Deep Learning*. Universitas Brawijaya, 2020.
[13] K. S. Nugroho *et al.*, “DETEKSI DEPRESI DAN KECEMASAN PENGGUNA TWITTER,” no. Ciastech, pp. 287–296, 2021.
[14] P. T. Yamak, “A Comparison between ARIMA , LSTM , and GRU for Time Series Forecasting,” 2017.
[15] J. Chung, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling arXiv : 1412 . 3555v1 [cs . NE] 11 Dec 2014,” no. December, 2014.
[16] B. Van Merri, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” 2013.
[17] R. Gençay, M. Dacorogna, U. A. Muller, O. Pictet, and R. Olsen, *An introduction to high-frequency finance*. Elsivier, 2001.
[18] T. G. Andersen, “Some reflections on analysis of high-frequency data,” *J. Bus. Econ. Stat.*, vol. 18, no. 2, pp. 146–153, 2000, doi: 10.1080/07350015.2000.10524857.
[19] K.-S. Chan and J. D. Cryer, *Time series analysis with applications in R*. springer publication, 2008.
[20] U. I. Arfianti, D. C. R. Novitasari, N. Widodo, M. Hafiyusholeh, and W. D. Utami, “Sunspot Number Prediction Using Gated Recurrent Unit (GRU) Algorithm,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 2, p. 141, 2021, doi: 10.22146/ijccs.63676.
[21] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.
[22] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.