JTIM : Jurnal Teknologi Informasi dan Multimedia
p-ISSN : 2715-2529
e-ISSN : 2684-9151
https://journal.sekawan-org.id/index.php/jtim

Comparative Analysis of Stock Price Prediction Using Deep Learning with Data Scaling Method

I Nyoman Switrayana [1]*, Rifqi Hammad [2], Pahrul Irfan [3], Tomi Tri Sujaka [1], Muhammad Haris Nasri [4]

[1] Computer Science, Bumigora University, Indonesia
[2] Software engineering, Bumigora University, Indonesia
[3] Informatics Engineering, Univesity of Mataram, Indonesia
[4] Informatics Engineering, Bumigora University, Indonesia

* Correspondence: nyoman.switrayana@universitasbumigora.ac.id

Citation: Switarayana, I. N.; Hammad, R.; Irfan, P.; Sujaka, T. T.; Nasri, M. H. (**2025**). Comparative Analysis of Stock Price Prediction Using Deep Learning with Data Scaling Method. JTIM: Jurnal Teknologi Informasi Dan Multimedia, 7(1), 78-90. https://doi.org/10.35746/jtim.v7i1.650

Received: 28-11-2024
Revised: 27-12-2024
Accepted: 02-01-2025

Abstract: The dynamic and unpredictable nature of stock prices makes accurate forecasting an important challenge in financial analysis. This study aims to compare the performance of three deep learning models, namely, Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) in predicting stock prices on historical daily banking data from Yahoo Finance. The main objective is to determine the model that is best able to capture sequential patterns and temporal dependencies in stock price movements. Each model was trained and optimized through data scaling, namely MinMax Scaler and Standard Scaler, with performance evaluated using Root Mean Square Error (RMSE) as the primary metric. Results show that while the RNN provides a basic approach, the GRU and LSTM models produce higher prediction accuracy, with GRU achieving the lowest RMSE thanks to its better ability to maintain long-term dependencies. The RMSE achieved by RNN, GRU, and LSTM were 211.47, 158.89, and 197.45, respectively. The lowest error results were achieved when using MinMax Scaler. The use of MinMax Scaler here shows a better performance improvement with an average improvement of 22.57% compared to using Standard Scaler. This comparative analysis contributes to providing empirical insight into the relative effectiveness of the tested architectures. The findings suggest that the combination of GRU and MinMax Scaler can be a more reliable tool for financial forecasting, with the potential to develop more robust stock prediction applications under fluctuating market conditions.

Keywords: Stock Price; Prediction; Deep Learning; Data Scaling

1. Introduction

Stock price prediction is an important focus in financial technology research due to the volatility of the stock market, which affects investment decisions and economic stability [1][2]. Recent advances in deep learning have enabled new approaches to time series forecasting, especially for non-linear and dynamic stock price data, which is difficult to predict with traditional statistical methods[3]. Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM) models have been widely applied to capture temporal dependencies and patterns in financial time series, showing promising accuracy in stock price prediction despite requiring greater time and memory[4]. Challenges to stock price prediction models remain as their predictive performance is affected by market fluctuations and data quality. [5][6]. To overcome these challenges, the study of deep learning LSTM models and preprocessing methods to improve prediction accuracy is highly recommended [7]. Research [8] provides information related to the use of the RNN-LSTM hybrid model does not always produce the best performance where LSTM as a hidden layer, RNN works well only at the single step stage and not in the multi-step prediction model. The LSTM model in [9] provides the best prediction results compared to Artificial Neural Network (ANN) and Autoregressive Integrated Moving Average (ARIMA) models, but is highly dependent on data preprocessing.

Previous research shows that the LSTM model works well with large amounts of data [10]. The use of bi-LSTM by [11] showed a very reliable model performance, where the Mean Square Error (MSE) reached 0.00020 on the testing data. The research conducted data preprocessing before being included in bi-LSTM using MinMax Scaler. Comparison of bi-LSTM and LSTM models was also tested by [12] showing the bi-LSTM model has better performance. The use of the GRU model is also often superior to the combined LSTM-GRU model or LSTM alone [13]. Although LSTM models generally achieve high prediction accuracy, they tend to be computationally intensive. Alternatively, GRU models were explored as a more computationally efficient option while maintaining similar predictive rates. The proposed RNN-LSTM model [14] has significant potential in prediction with optimization using Adaptive Moment Estimation (Adam).

In a comparative analysis conducted by [15], GRU and LSTM showed better performance than vanilla RNN. However, recent studies have shown research gaps in a comprehensive comparison between these models, especially regarding the impact of data scale on prediction accuracy. Some of the above studies contributed to the exploration of deep learning methods for financial forecasting, emphasizing the need for model optimization through data preprocessing. Data scaling techniques, including MinMax and Standard Scaler, are essential in financial forecasting due to their ability to stabilize data variability and improve model convergence. Data preprocessing is especially important in stock forecasting due to the inherent noise and high frequency of stock data. By applying scaling techniques, the range of data can be adjusted to reduce volatility, thus allowing the model to identify underlying trends more effectively.

This study aims to fill the research gap by conducting a comparative analysis on RNN, GRU, and LSTM models in predicting stock prices. The main contribution of this research is the integration of data scaling techniques into the preprocessing pipeline for each model, to systematically assess their impact on prediction accuracy. Using Root Mean Square Error (RMSE) as an evaluation metric, this research evaluates the performance of each model with each data scaling to understand its effect on different architectures. This research provides new insights into how scaling affects model performance in volatile financial markets, an area that has rarely been explored before. The novelty of this research lies in the comparative analysis of three recurrent neural architectures combined with scaling techniques, an approach that has not been comprehensively addressed in previous stock price prediction studies. Furthermore, this study uses the same hyperparameters for all architectures, ensuring a fair and accurate comparison between the RNN, GRU, and LSTM models. By analyzing these models under consistent conditions with identical datasets and scaling techniques, this research provides valuable insights in selecting suitable models for financial forecasting. Moreover, the insights gained from this research can assist financial analysts in developing more accurate prediction models, which can ultimately support better investment decision-making in volatile markets.

2. Materials and Methods

This research uses the Cross-Industry Standard Process for Data Mining (CRISP-DM) approach which consists of six stages: Business Understanding, Data Understanding, Data

Preparation, Modeling, Evaluation, and Deployment [16]. In this research, only up to the evaluation stage. Figure 1 shows the flow of the CRIPS-DM methodology. This methodology was chosen because of its sequential analysis structure, making it easier to explore and develop appropriate predictive models for time-series data.

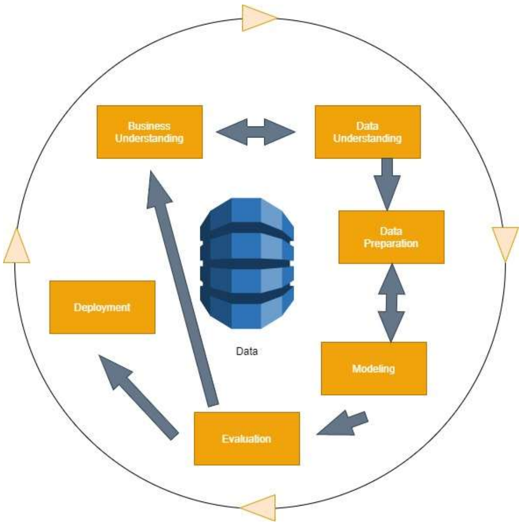


Figure 1. CRISP-DM Methodology [17]

2.1. *Business Understanding*

This stage begins with understanding the problem and research objectives, namely the prediction of banking stock prices taken from Yahoo Finance. The stock data used is the daily stock price of Mandiri Bank. With the ability to accurately predict stock prices, this research aims to help financial actors make better decisions. In addition, this research focuses on comparing the performance of three deep learning models: Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM). These methods were chosen due to their proven ability to model sequential data, which is a key characteristic of stock price time series. While there are numerous deep learning approaches available, these three represent state-of-the-art methods for sequence modeling, each with unique mechanisms for handling temporal dependencies. The study also emphasizes the role of data scaling in reducing prediction errors, a critical yet underexplored aspect of improving model accuracy. However, it is acknowledged that these models do not fully represent all deep learning methods, highlighting the need for ongoing exploration of other architectures and techniques.

2.2. *Data Understanding*

This stage involves an initial exploration of the stock price data, which includes the Open, High, Low, Close, Adj Close, and Volume. This process aims to understand trend patterns, volatility and data structure. The data will be analyzed to detect missing values that may affect the model results. The results of this exploration will determine the need for further preprocessing techniques to ensure that the data meets the requirements of the time-series model.

2.3. *Data Preparation*

At this stage, data preparation is carried out by preprocessing before the data is included in the model. Preprocessing is one of the important stages to be carried out in order to prepare good data that is ready to be used in the modeling stage. At this stage, the data will be prepared through several preprocessing steps. First, Handling Missing Values, where Missing values will be removed from the dataset. Second, Data Sharing where the dataset is divided into training data and test data with a certain proportion, ensuring that the model can generalize well on new data and avoid overfitting. Third, the data is processed with various scaling techniques, including MinMax Scaler and Standard Scaler. Scaling, also known as data normalization, helps the model to adapt more quickly in the training process and also reduces the possibility of overfitting. Based on several studies, normalization has been shown to improve accuracy/minimize error in deep learning-based prediction models. Data normalization is applied to reduce variability and accelerate model convergence during training, which has been shown to improve prediction performance in previous studies. In this study, each scaling method will be compared to the model performance results to assess the impact of normalization on model performance. To calculate MinMax Scaler, we can use formula (1) and Standard Scaler using formula (2).

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

$$X_{scaled} = \frac{X - \mu}{\sigma} \tag{2}$$

X_{min}	: The minimum value in the original data..
X_{max}	: The maximum value in the original data.
X	: The original data value to be scaled.
μ	: The mean value of the data.
σ	: The standard deviation of the data.
X_{scaled}	: The value of the data after scaling.

2.4. *Modeling*

This stage involves developing prediction models using three types of deep learning architectures: Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM). Each model has its own advantages in processing time-series data, and the three models are applied to understand their performance in predicting stock prices. The modeling process includes several sub-stages, namely architecture selection, hyperparameter adjustment, training, and model evaluation. The following is a detailed explanation for each model.

2.4.1. Recurrent Neural Network (RNN)

RNN is a type of neural network specifically designed for sequential or time-series data. The RNN architecture is illustrated in figure 2**.** The RNN architecture has a feedback loop connection, which allows information from previous steps to be passed on to the next step. Although RNNs are effective for sequential data, they have a weakness in handling long-term dependencies due to the vanishing gradient problem, which makes the gradient decrease drastically as layers and time steps are added in training. In this case, t represents time, x is the input layer, S is the hidden layer, and O is the output layer. W, U, V are used as weights on each corresponding layer [18].

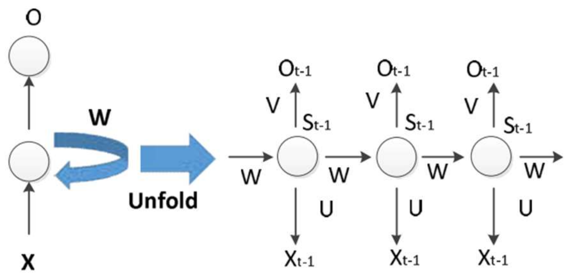


Figure 2. RNN Architecture [3]

2.4.2. Gated Recurrent Unit (GRU)

GRU is an extension of RNN that aims to overcome the vanishing gradient problem by introducing a "gates" mechanism that allows for longer information retention. GRU has two types of gates, namely reset gate and update gate, which regulate which information needs to be passed on to the next time. This model is proven to be more memory and computationally efficient than LSTM. Figure 3 shows the architecture of GRU. Equation (3-6) used in GRU in processing information refers to [18].

$$z_t = \sigma\,(w_z x_t + u_z h_{t-1} + b_z) \tag{3}$$
$$r_t = \sigma\,(w_r x_t + u_r h_{t-1} + b_r) \tag{4}$$
$$\tilde{h} = tanh\,(w_h x_t + r_h * u_h h_{t-1} + b_h) \tag{5}$$
$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h} \tag{6}$$

z_t : Update gate.
r_t : Reset gate.
$\tilde{h}$: Where the previous information is stored by the reset gate.
h_t : The final storage used by the update gate.
x_t : Input data.
b : Bias.
w and u : The weight.

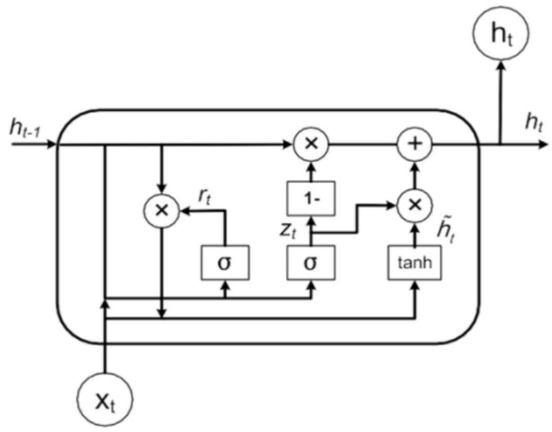


Figure 3. RNN Architecture [18]

2.4.3. Long Short-Term Memory (LSTM)

LSTM is designed to overcome the vanishing gradient problem in RNN to process sequential data. LSTM uses three types of gates: input, forget, and output gates, which serve to regulate the flow of information and retain important information in the long run [6] [7]. LSTM is widely used for stock price prediction and is proven to be effective in predicting data with high volatility [8] [10] [19] [20]. In LSTM, the neurons in the hidden layer in RNN are replaced by LSTM cells that function to maintain previous information. The LSTM architecture is shown in Figure 4. Equations (7-12) are the equations used in the LSTM model at each gate in Figure 4 according to the explanation [21].

$$f_t = \sigma\,(W_{fh}[h_{t-1}], W_{fx}[x_t], b_f) \tag{7}$$
$$i_t = \sigma\,(W_{ih}[h_{t-1}], W_{ix}[x_t], b_i) \tag{8}$$
$$\bar{C}_t = tanh\,(W_{ch}[h_{t-1}], W_{cx}[x_t], b_c) \tag{9}$$
$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \tag{10}$$

$$O_t = \sigma\,(W_{oh}[h_{t-1}], W_{ox}[x_t], b_o) \tag{11}$$

$$h_t = O_t * \tanh\,(C_t) \tag{12}$$

f_t : Forget gate.
i_t : Input gate.
O_t : Output gate.
x_t : Input value nilai input.
W : Weight.
b : Bias.
h_{t-1} : Output from time t-1.
C_{t-1} : The memory cell state of the previous cell.
h_t : The final output.

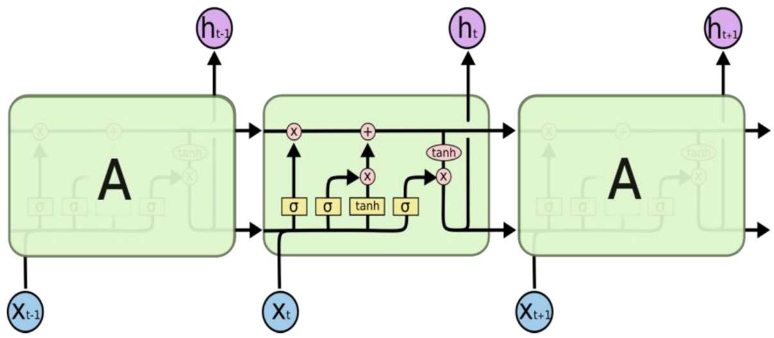


Figure 4. LSTM Architecture [12]

2.5. Evaluation

The evaluation was conducted using Root Mean Square Error (RMSE), an evaluation metric that measures the mean square difference between predicted and actual values. RMSE is a commonly used way to measure model error from predicted data that is quantitative in nature. RMSE is used to determine the size of the distribution of the deviation of data points from the linear regression line or to determine the concentration of data around the linear regression line [13]. RMSE was chosen because it is relevant in measuring numerical prediction error on time-series data. The model will be evaluated on each data normalization method and the results will provide information on the impact of different data scaling on the accuracy of the model in the stock price prediction scenario. RMSE provides information on how accurate the model is in predicting the target value. The lower the RMSE value, the better the model. RMSE is calculated by formulas (13) and (14). Mean Square Error (MSE) is the sum of the squares of the errors between the actual value and the forecasting value and divided by the number of forecasting times [4] [14] [22]. Where is y the actual data/value, $\hat{y}$ is the predicted value, and n is the amount of data. Then to calculate RMSE just find the root of MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_1 - \hat{y}_1)^2 \tag{13}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_1 - \hat{y}_1)^2} \tag{14}$$

3. Results and Discussion

3.1. Business Understanding

At the Business Understanding stage, this research predicts the share price of Bank Mandiri using deep learning to provide reliable predictive tools for investors in the face of market volatility. The stock market is dynamic and affected by various external factors, so accurate predictions can help in making better investment decisions. This research uses historical stock price data from Yahoo Finance to train three deep learning models, namely RNN, GRU, and LSTM, to compare their effectiveness. By applying MinMax Scaler and Standard Scaler as scaling techniques, the research examines how data preprocessing can improve prediction accuracy. The prediction results are expected to provide empirical insight into the best model that captures stock price patterns, as well as provide guidance for further research. The success of the model in predicting stock prices will be very useful for investment strategies and business decisions in the financial sector.

3.2. Data Understanding

The data used is Mandiri Bank stock data sourced from https://finance.yahoo.com. This dataset is bank Mandiri's stock price data for the period September 2020-2024. The data is downloaded in the form of Comma Separated Values (CSV) format which is then transformed into a dataframe using the Pandas module to be used in the next process. Table 1 shows a sample of bank stock data that has been downloaded.

Table 1. Bank share price data

Date	Open	High	Low	Close	Adj Close	Volume
2023-10-02	6025	6100	6025	6050	5755,643555	33296500
2023-10-03	6000	6075	6000	6075	5779,427246	37093100
2023-10-04	6075	6125	5975	6125	5826,994629	69154000
…	…	…	…	…	…	…
2024-09-25	7350	7350	7050	7200	7200	192197100
2024-09-26	7175	7200	7100	7175	7175	173044000
2024-09-27	7025	7125	7000	7050	7050	161405200

The dataset consists of several important attributes that support comprehensive stock price analysis. The explanation of each attribute/feature in the dataset is as stated by [23]. Where, Date records the date of each trade data, providing a time context for monitoring trends and patterns in stock price movements. Open is the initial trading price of the stock on the day, providing an early snapshot of market sentiment. High and Low show the highest and lowest prices reached during the day, reflecting the range of volatility of the stock throughout the day. Close, or closing price, reflects the last price at the end of trading, often used as a benchmark for daily performance analysis. The Adj Close (Adjusted Close) is the closing price adjusted for factors such as dividends and stock splits, which is particularly useful for long-term analysis as it takes into account relevant value changes. Finally, Volume describes the number of shares traded on the day, which indicates the level of market activity and interest in the stock. There were no missing values in the data. The data is then divided into training and test data, where training data is taken 80% of the total data and test data the remaining 20%.

3.3. Data Preparation

The results of the bank stock closing price scale transformation applied with two commonly used scaling techniques in modeling: MinMax Scaler and Standard Scaler are shown in Table 2. Both methods are used to scale the data to make it more uniform, so that the deep learning model can process the data more efficiently and reduce potential bias caused by differences in value ranges between features.

Table 2. Comparison of MinMax Scaler results with Standard Scaler

Date	Close	MinMax_Scaler	Standard_Scaler
2023-10-02	6050	0,222222222	-0,815946233
2023-10-03	6075	0,236111111	-0,768397854
2023-10-04	6125	0,263888889	-0,673301096
…	…	…	…
2024-09-25	7200	0,861111111	1,371279199
2024-09-26	7175	0,847222222	1,32373082
2024-09-27	7050	0,777777778	1,085988925

Table 2 illustrates the closing prices of bank stocks that have been processed using two scaling methods: MinMax Scaler and Standard Scaler. In the first column, the closing price of the stock on a particular date is recorded, which shows the fluctuation of the market price. The use of MinMax Scaler transforms the stock price, which was originally in a larger range of values, into a more controlled range between 0 and 1. As a result, the stock price values that were previously in the range of 5650 to 7450 are now reduced to smaller, standardized numbers. For example, on 2023-10-02, the closing price of 6050 was transformed to 0.22 after MinMax Scaler was applied. Meanwhile, the Standard Scaler transforms the data so that it has a distribution with a mean of 0 and a standard deviation of 1, although the resulting range of data values can vary greatly. For example, on the same date (2023-10-02), the closing price of stock 6050 was transformed to -0.81, reflecting that it was below the average stock price in the dataset. This difference shows that Standard Scaler measures how far a stock price value is from its average value in terms of standard deviation, while MinMax Scaler focuses more on normalizing values within a fixed range between 0 and 1.

3.4. *Modelling*

The scaled data will be used as input in each stock price prediction model, namely RNN, GRU, and LSTM. Each model is designed with an architectural structure consisting of two layers blocks, where each block has 50 neuron units. These neuron units use a 'tanh' activation function that is often used in recurrent networks to model sequential relationships and capture temporal patterns in time series data such as stock prices. In the first block, the neuron layer is equipped with a dropout layer of 0.2 to help reduce the possibility of overfitting that may occur during the training process. This dropout technique will randomly turn off some neurons in the training process, which encourages the network to learn more general features and reduce overreliance on specific features [24]. The model architecture is shown in figure 5.

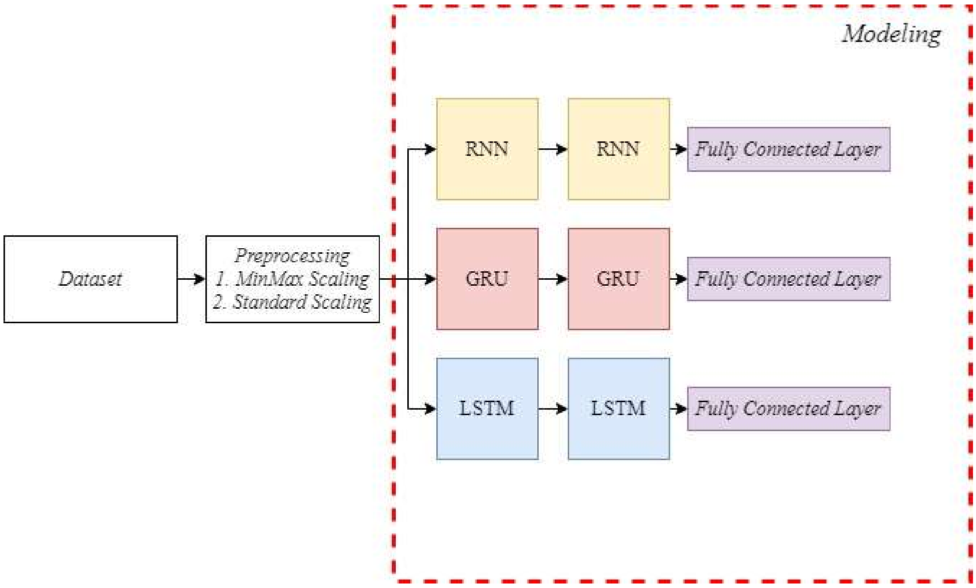


Figure 5. Prediction model architecture using RNN, GRU, and LSTM

The last layer in each model is the dense layer or fully connected layer which consists of 1 neuron to produce a single output, which is the predicted closing price of the stock. This layer is responsible for integrating all the information from the previous layers and generating the optimal price forecast. To optimize the model, the Adam optimizer algorithm is used, as it is able to adapt the learning rate during training, thus accelerating convergence to the optimal solution. As a loss function, Mean Square Error (MSE) was applied to measure how far the model's predictions are from the true values, where a lower MSE value indicates a more accurate prediction. Each model was trained using batch sizes of 64 and 100 epochs, which means the model will process the entire dataset 100 times to strengthen the understanding of patterns in the data. The use of 100 epochs was deemed sufficient to train the model as done in [19].

3.5. Evaluation

In the evaluation stage, the performance of the three deep learning models used in this study is evaluated based on the prediction error value calculated using RMSE. In this experiment, two different scaling techniques, MinMax Scaler and Standard Scaler, were applied to prepare the data before the model training process. Scaling the data aims to improve the convergence of the model as well as optimize the performance in overcoming scale differences between features in the dataset. Table 3 presents the prediction error values measured using RMSE for three models, namely RNN, GRU, and LSTM, with the application of two different scaling methods: MinMax Scaler and Standard Scaler

Table 3. Comparison of MinMax Scaler results with Standard Scaler

Model	**RMSE**	
	MinMax Scaler	**Standard Scaler**
RNN	211,47	348,62
GRU	**158,89**	171,04
LSTM	197,45	250,76
Average	189,27	256,81

Based on Table 3, of the three models tested, GRU provides the best prediction results with the lowest RMSE of 158.89 using MinMax Scaler, followed by LSTM with an RMSE of 197.45, and RNN with the highest RMSE of 211.47. These results demonstrate the superiority of GRU in capturing temporal patterns in stock price time-series data, despite its simpler architecture compared to LSTM. LSTM, with a unique architecture capable of maintaining long-term dependencies through a cell memory mechanism, performed quite well but slightly higher than GRU. Meanwhile, RNN showed the lowest performance among the three models, indicating its limitation in maintaining complex long-term patterns. In addition, the use of Standard Scaler tends to increase the RMSE values for all three models, where the average RMSE for Standard Scaler is higher than MinMax Scaler, indicating that the scale of the data has a significant impact on the prediction accuracy. Figure 6 presents a performance comparison graph of the three deep learning models, using two different scaling methods: MinMax Scaler and Standard Scaler. This graph illustrates the RMSE values produced by each model after scaling the data. This comparison aims to show the effect of scaling techniques on the performance of stock price prediction performed by these models.

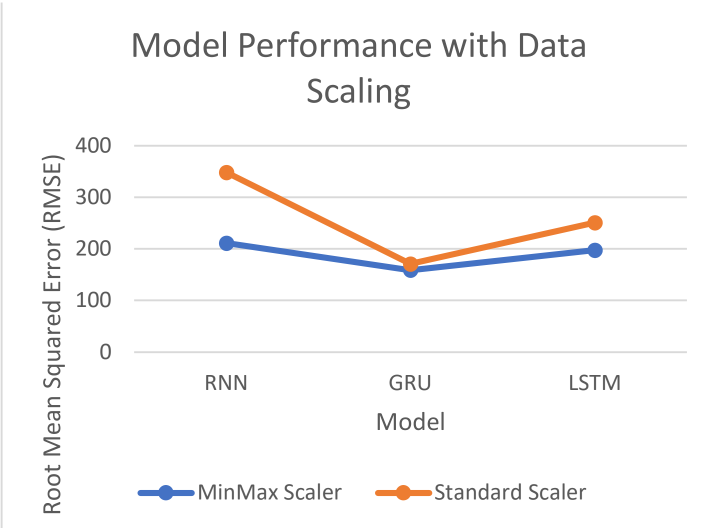
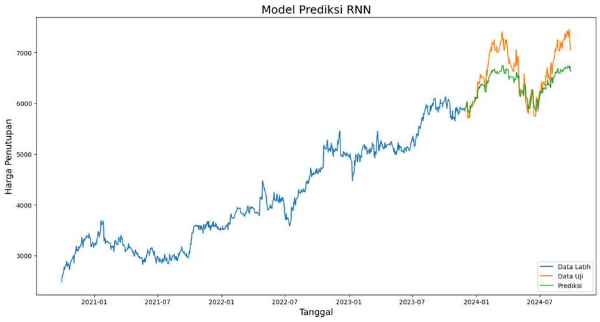
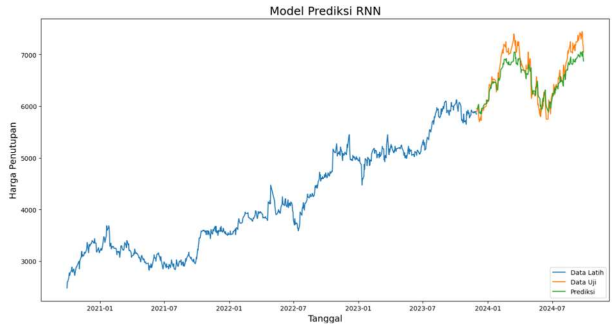


Figure 6. Model performance comparison graph using MinMax Scaler and Standard Scaler

From Figure 6, it can be observed that the MinMax Scaler provides consistently lower RMSE values across the three models, indicating a better performance improvement compared to the Standard Scaler. This figure provides a clear visual representation of the advantage of using MinMax Scaler in improving prediction accuracy. Based on the results obtained in Table 3, the difference in RMSE values between using MinMax Scaler and Standard Scaler for the RNN model is 137.15, which indicates that using MinMax Scaler provides better results in improving prediction accuracy. The RMSE difference in the GRU model is 12.15, indicating a still significant performance improvement, although not as large as in the RNN model RMSE improvement. The LSTM model has an RMSE difference of 53.31, indicating that LSTM also benefits from the use of MinMax Scaler to capture temporal patterns more accurately. The average RMSE of the model using MinMax Scaler is 189.27, while the average RMSE on Standard Scaler is 256.81. This difference shows that MinMax Scaler provides an average performance improvement of about 22.57% over Standard Scaler. This indicates that MinMax Scaler is more effective in retaining the original distribution information of the data, which is important for deep learning models to accurately identify time-series patterns. With the range of values standardized into a fixed scale of 0-1, MinMax Scaler helps the model capture small variations in stock prices, which is crucial in financial applications. Figure 7, Figure 8, and Figure 9 show the prediction trendlines of the RNN, GRU, and LSTM models on bank stock data, with Standard Scaler and MinMax Scaler, respectively. In each figure, the testing data and prediction data are shown to give an idea of how well the models follow the actual price movement patterns.

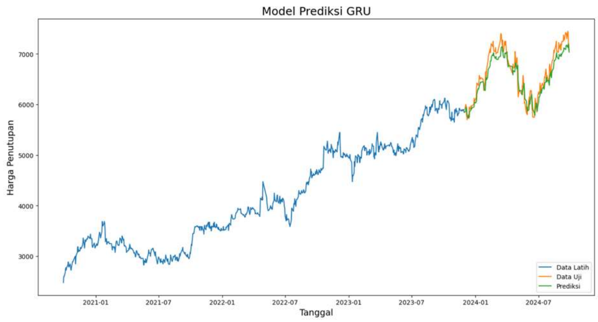


RNN+*Standard Scaler*

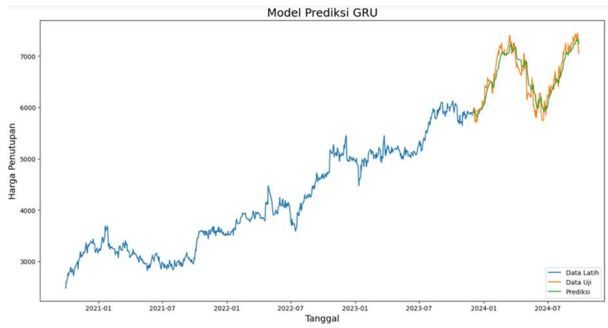


RNN+*MinMax Scaler*

Figure 7. Trendline prediction using RNN model on Standard Scaler and MinMax Scaler

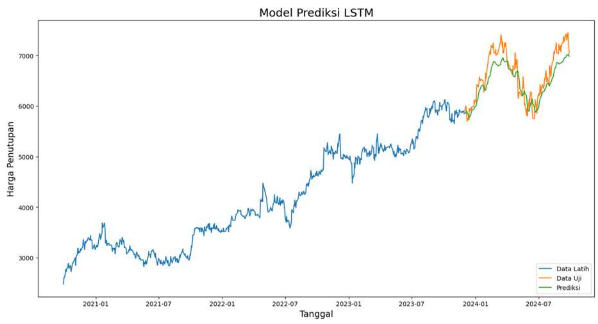


GRU+*Standard Scaler*

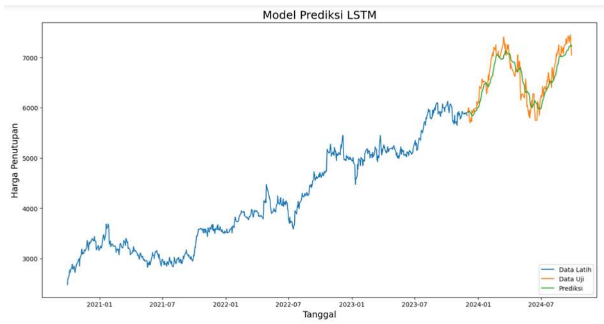


GRU+*MinMax Scaler*

Figure 8. Trendline prediction using GRU model on Standard Scaler and MinMax Scaler



LSTM+*Standard Scaler*



LSTM+*MinMax Scaler*

Figure 9. Trendline prediction using LSTM model on Standard Scaler and MinMax Scaler

Each model, including RNN, LSTM, and GRU, was implemented using the TensorFlow framework with the same architectural structure and configuration as explained in modeling stage. Figure 7 shows the trendline of the RNN model predictions with a poor fit to the test data (20% of the total dataset), especially when using the Standard Scaler. While the MinMax Scaler shows improvement, the difference in predictions remains visible at some points, indicating the limitations of the RNN in capturing complex stock price patterns. Figure 8 shows the prediction results of the GRU model, which appears to more accurately follow the trendline of the test data. This visualization confirms the superiority of GRU, especially on the MinMax Scaler, where the prediction pattern fits the testing data better than the Standard Scaler. This shows that GRU is able to capture temporal dependencies better, thus providing more accurate prediction results in accordance with the findings of [15]. Figure 9 displays the trendline of the LSTM model prediction, which also shows an improvement in accuracy when using MinMax Scaler over Standard Scaler. Despite the fit, the LSTM predictions still show a slight mismatch at some points compared to GRU, which in this case produces the most consistent and accurate prediction performance. This may be due to the LSTM model showing strong performance in short-term stock price prediction, but it struggles to model long-term dependencies effectively, requiring modifications such as Extended Long Short-Term Memory (xLSTM) [22]. Various optimizations on LSTM also need to be done in order to produce a more reliable model as done by [25]. In contrast to GRU, in this study, just the basic model is able to provide better performance than the models that have been tested.

4. Conclusions

The conclusion of this research shows that the Gated Recurrent Unit (GRU) model with MinMax Scaler provides the most accurate stock price prediction results compared to the Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) models in the analysis of daily stock data. The GRU model achieved the lowest RMSE of 158.89, demonstrating its superiority in capturing the temporal patterns and dependencies of dynamic time-series data. Each model, including RNN, LSTM, and GRU, was implemented with the same architectural structure and configuration, consisting of two layers of 50 neuron units with 'tanh' activation functions, a dropout rate of 0.2, adam optimizer, and a dense layer with a single output neuron. The application of MinMax Scaler consistently improves the prediction performance compared to Standard Scaler, with an average accuracy improvement of about 22.57%. This indicates that MinMax Scaler is more effective in retaining the original distribution information of the data, which is important for deep learning models to accurately identify time-series patterns. With the range of values standardized into a fixed scale of 0–1, MinMax Scaler helps the model capture small variations in stock price. These findings enrich the stock prediction literature by showing that the combination of GRU and MinMax Scaler is highly reliable in stock price forecasting. In addition, these results open up opportunities for further research, such as the exploration of hybrid or multimodal model architectures, to improve the robustness and accuracy of predictions under various volatile market conditions.

.

References

[1] F. Nur Iman and D. Wulandari, "Prediksi Harga Saham Menggunakan Metode Long Short Term Memory," *Log. J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 3, pp. 601–616, 2023. https://journal.mediapublikasi.id/index.php/logic/article/view/1855

[2] M. R. Pahlawan, A. Djunaidy, R. A. Vinarti, and D. S. Informasi, "Prediksi Indeks Harga Saham Menggunakan Model Hibrida," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 4, 2022. https://doi.org/10.35957/jatisi.v9i4.3065

[3] M. A. D. Suyudi, E. C. Djamal, A. Maspupah, J. Informatika, and F. Sains, "Prediksi Harga Saham menggunakan Metode Recurrent Neural Network," in *Section SNATI 2019: Bidang Sistem Cerdas, Informatika, Text Mining & Visi Komputer*, 2019, pp. 33–38. https://doi.org/10.35957/jatisi.v9i4.3065

[4] P. Soni, Y. Tewari, and D. Krishnan, "Machine Learning Approaches in Stock Price Prediction: A Systematic Review," *J. Phys. Conf. Ser.*, vol. 2161, no. 1, 2022. https://doi.org/10.1088/1742-6596/2161/1/012065

[5] A. Agusta, I. Ernawati, and A. Muliawati, "Prediksi Pergerakan Harga Saham Pada Sektor Farmasi Menggunakan Algoritma Long Short-Term Memory," *Inform. J. Ilmu Komput.*, vol. 17, no. 2, pp. 164–173, 2021. https://doi.org/10.52958/iftk.v17i2.3651

[6] C. C. Sumarga, D. E. Herwindiati, and J. Hendryli, "Rancangan Sistem Prediksi Harga Saham dengan Menggunakan Metode LSTM dan ARMA klasik," *J. Ilmu Komput. dan Sist. Inf.*, vol. 11, no. 1, 2023. https://doi.org/10.24912/jiksi.v11i1.24075

[7] C. G. K. Simatupang, W. Swastika, and T. R. Suganda, "Perancangan Aplikasi Berbasis Web untuk Prediksi Harga Saham Dengan Metode LSTM," *Sainsbertek J. Ilm. Sains dan Teknol.*, vol. 3, no. 1, 2022. https://doi.org/10.33479/sb.v3i1.212

[8] I. Akil and I. Chaidir, "Prediksi Harga Saham Twitter Dengan Long Short-Term Memory Recurrent Neural Network," vol. 17, no. 1," *INTI Nusa Mandiri*", pp. 1–7, 2022. https://doi.org/10.33480/inti.v17i1.3277

[9] Q. Ma, "Comparison of ARIMA, ANN and LSTM for Stock Price Prediction," *E3S Web Conf.*, vol. 218, pp. 1–5, 2020. 10.1051/e3sconf/202021801026

[10] R. Julian and M. R. Pribadi, “Peramalan Harga Saham Pertambangan Pada Bursa Efek Indonesia (BEI) Menggunakan Long Short Term Memory (LSTM),” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 3, 2021. https://doi.org/10.35957/jatisi.v8i3.1159

[11] C. Han and X. Fu, “Challenge and Opportunity: Deep Learning-Based Stock Price Prediction by Using Bi-Directional LSTM Model,” *Front. Business, Econ. Manag.*, vol. 8, no. 2, pp. 51–54, 2023.

[12] D. I. Puteri, “Implementasi Long Short Term Memory (LSTM) dan Bidirectional Long Short Term Memory (BiLSTM) Dalam Prediksi Harga Saham Syariah,” *Euler J. Ilm. Mat. Sains dan Teknol.*, vol. 11, no. 1, pp. 35–43, 2023. https://doi.org/10.34312/euler.v11i1.19791

[13] W. Hastomo, A. S. B. Karno, N. Kalbuana, E. Nisfiani, and L. ETP, “Optimasi Deep Learning untuk Prediksi Saham di Masa Pandemi Covid-19,” *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 2, pp. 133–140, 2021. https://doi.org/10.26418/jp.v7i2.47411

[14] S. J. Pipin, R. Purba, and H. Kurniawan, “Prediksi Saham Menggunakan Recurrent Neural Network (RNN-LSTM) dengan Optimasi Adaptive Moment Estimation,” *J. Comput. Syst. Informatics*, vol. 4, no. 4, pp. 806–815, 2023. https://doi.org/10.47065/josyc.v4i4.4014

[15] A. S. Saud and S. Shakya, “Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE,” *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 788–798, 2020. https://doi.org/10.1016/j.procs.2020.03.419

[16] A. Pambudi, “Penerapan Crisp-Dm Menggunakan Mlr K-Fold Pada Data Saham Pt. Telkom Indonesia (Persero) Tbk (Tlkm) (Studi Kasus: Bursa Efek Indonesia Tahun 2015-2022),” *J. Data Min. dan Sist. Inf.*, vol. 4, no. 1, p. 1, 2023. https://doi.org/10.33365/jdmsi.v4i1.2462

[17] I. N. Switrayana, D. Ashadi, H. Hairani, and A. Aminuddin, “Sentiment Analysis and Topic Modeling of Kitabisa Applications using Support Vector Machine (SVM) and Smote-Tomek Links Methods,” *Int. J. Eng. Comput. Sci. Appl.*, vol. 2, no. 2, pp. 81–91, 2023. https://doi.org/10.30812/ijecsa.v2i2.3406

[18] Y. Zhu, “Stock price prediction using the RNN model,” *J. Phys. Conf. Ser.*, vol. 1650, no. 3, 2020. https://doi.org/10.1088/1742-6596/1650/3/032103

[19] G. Budiprasetyo, M. Hani’ah, and D. Zahira Aflah, “Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM),” *J. Nas. Teknol. dan Sist. Inf.*, vol. 03, no. 2022, pp. 164–172, 2023. https://doi.org/10.25077/teknosi.v8i3.2022.164-172

[20] A. Satyo and B. Karno, “Prediksi Data Time Series Saham Bank BRI Dengan Mesin Belajar LSTM (Long ShortTerm Memory),” *Journal of Informatic and Information Security*," vol. 1, no. 1, pp. 1–8, 2020. https://doi.org/10.31599/jiforty.v1i1.133

[21] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “The comparison of forecasting analysis based on the ARIMA-LSTM hybrid models,” *Proc. - 2021 Int. Conf. E-Commerce E-Management, ICECEM 2021*, pp. 185–188, 2021. https://doi.org/10.1109/ICECEM54757.2021.00044

[22] X. Fan, C. Tao, and J. Zhao, “Advanced Stock Price Prediction with xLSTM-Based Models: Improving Long-Term Forecasting,” *Preprints*, 2024.https://doi.org/10.20944/preprints202408.2109.v1

[23] M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, and G. Wills, “A survey on machine learning for stock price prediction: Algorithms and techniques,” *FEMIB 2020 - Proc. 2nd Int. Conf. Financ. Econ. Manag. IT Bus.*, pp. 63–71, 2020. https://doi.org/10.5220/0009340700630071

[24] I. N. Switrayana, S. Hadi, and N. Sulistianingsih, “A Robust Gender Recognition System using Convolutional Neural Network on Indonesian Speaker,” *Sist. J. Sist. Inf.*, vol. 13, no. 3, pp. 1008–1021, 2024. https://doi.org/10.32520/stmsi.v13i3.3698

[25] B. Gülmez, “Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm,” *Expert Syst. Appl.*, vol. 227, no. January, p. 120346, 2023. https://doi.org/10.1016/j.eswa.2023.120346