MORFAI JOURNAL Multidiciplinary Output Research For Actual and International Issue
Multidisciplinary Output Research For Actual and International Issue (MORFAI Journal) ISSN (e): 2808-6635
Volume 6 No. 5 (2026)

OPTIMIZATION OF LSTM HYPERPARAMETERS USING GREY WOLF OPTIMIZATION AND HYPERBAND FOR PREDICTING BRI STOCK PRICES

Eko Budi Pratama[1], Rudi Nurdiansyah[2]*, Muhammad Ali Ridho[3], Irsyad Khoirun Ramadhan[4], Ibnu Toharri[5]
[1,2,3,4,5]Universitas Negeri Malang, Malang
E-mail: eko.budi.2205166@students.um.ac.id[1], rudi.nurdiansyah.ft@um.ac.id[2]*, muhammad.ali.2205166@students.um.ac.id[3], irsyad.khoirun.2205166@students.um.ac.id[4], ibnu.toharri.2205166@students.um.ac.id[5]

Received : 20 June 2026 Accepted : 21 July 2026
Revised : 29 June 2026 Published : 27 July 2026

Abstract

This research compares the performance of the Hyperband and Grey Wolf Optimization (GWO) algorithms in optimizing the hyperparameters of the Long Short-Term Memory (LSTM) model for predicting the stock price of Bank Rakyat Indonesia (BRI). Stock price prediction is a complex problem due to the dynamic and nonlinear characteristics of time series data. The dataset used consists of historical BRI stock data for the 2018-2025 period, obtained from Yahoo Finance. The optimization process was performed on several LSTM hyperparameters, including units, learning rate, dropout, optimizer, batch size, and epoch. Model performance was evaluated using Root Mean Square Error (RMSE). The results show that the LSTM-GWO model outperforms the LSTM-Hyperband model, with an RMSE of 75.93 compared to 77.60. However, Hyperband is more computationally efficient than GWO. The findings indicate that the GWO algorithm is more effective at improving prediction accuracy, while Hyperband excels in tuning efficiency. This study is expected to contribute to the development of hyperparameter optimization methods for deep learning models used in time-series-based stock price prediction.

Keywords: *Grey Wolf Optimization, Hyperband, Hyperparameter Optimization, LSTM, Stock Price Prediction*

INTRODUCTION

The capital market plays a vital role in a country's economy because it serves as both a means of raising funds and an investment vehicle for the public (Purnamasari, 2025). One form of investment in the capital market is stock investment. Stock investment offers high profit potential but also carries a high level of risk due to price movements that tend to be volatile (Duha & Putra, 2025). Several factors influence changes in stock prices, such as economic and political conditions, as well as company performance. These factors make it difficult to predict price movements accurately, particularly for stocks with high trading volume. High trading volume also makes stock price movement patterns more complex and interesting to study.

Stock price forecasting is performed using historical time-series data. The patterns in stock price time-series data are dynamic and nonlinear, making them quite difficult to model using conventional methods (Setiyani, 2025). Technological advancements have driven the use of deep learning-based approaches in time-series data forecasting because they are capable of learning complex data patterns (Andika & Kusrini, 2025). The Deep Learning model used to predict stock prices is the Long Short-Term Memory (LSTM). LSTM is designed to handle long-term dependencies in sequential data, enabling it to better capture nonlinear patterns (Hochreiter & Schmidhuber, 1997). This capability makes the LSTM model particularly relevant for predicting stock prices, which exhibit fluctuating patterns. Previous research has shown that the LSTM model is capable of providing good prediction results on time series data, including stock data (Remetwa et al., 2025).

Although it performs well, the predictive results of the LSTM model are heavily influenced by the choice of hyperparameters, such as the number of neurons, learning rate, batch size, and epochs (Andika & Kusrini, 2025). Manually determining hyperparameters is often time-consuming and does not necessarily yield optimal performance. Therefore, an optimization method is needed to help identify the best combination of hyperparameters more effectively and efficiently. One widely used method in hyperparameter optimization is Hyperband. This algorithm works by evaluating several hyperparameter configurations and halting the training process for configurations that show poor performance early on (S. Li et al., 2024). Through this mechanism, the hyperparameter search process can be conducted more efficiently. Research conducted by Eren et al., (2025) shows that Hyperband offers better time efficiency compared to random search and Bayesian optimization in air quality prediction. In addition to Hyperband, metaheuristic-based optimization methods are also increasingly being used to solve complex and nonlinear optimization problems (Salgotra et al., 2024). One such metaheuristic algorithm is Grey Wolf Optimization (GWO). This algorithm is inspired by the leadership behavior and hunting strategies of grey wolves in their search for optimal solutions (Mirjalili et al., 2014). GWO is known for its good balance between exploration and exploitation, making it effective for optimization processes (Yosviansyah & Rizal, 2024). Several studies have shown that using GWO for LSTM hyperparameter optimization can improve the performance of predictive models. Khayat et al., (2025) were able to improve predictive performance compared to several benchmark models in an energy consumption forecasting case.

Based on previous research, Hyperband and GWO have been widely used to improve the performance of LSTM models. However, studies specifically comparing the performance of these two algorithms in optimizing LSTM model hyperparameters for BBRI stock price prediction remain limited. Therefore, this study was conducted to analyze and compare the performance of the Hyperband and GWO algorithms in optimizing LSTM model hyperparameters for BBRI stock price prediction. Model performance was evaluated using Root Mean Square Error (RMSE) because this metric is sensitive to large prediction errors, thereby providing a more representative picture of model performance (Mubarak et al., 2025). This study is expected to contribute to the development of hyperparameter optimization methods for deep learning models, particularly in the prediction of stock price time series data.

LITERATURE REVIEW

Based on Table 1, various previous studies have shown that hyperparameter optimization is a key factor in improving the performance of LSTM models for predicting sequential time-series data . Various optimization methods, such as Bayesian Optimization (Andika & Kusrini (2025), Dilawar & Shahbaz (2025), Eren et al. (2025), Maulana et al. (2026)), Hyperband (Andika & Kusrini (2025), Eren et al. (2025), Song et al. (2025)), Particle Swarm Optimization (PSO) (Tanjung et al. (2025)), Genetic Algorithm (Dhake et al. (2025)), Heap-Based Optimizer (Dhake et al. (2025), Putra & Nurmawati (2024)), Manual tuning (Paygude et al. (2023)), Hyperparameter tuning (Deswal (2024)), and Grid Search (Putra & Nurmawati (2024)), have been shown to improve prediction accuracy in several cases involving stock price prediction, air quality, solar irradiance, commodity forecasting, and financial distress. Differences in the solution-search characteristics of each optimization method also influence the performance of the resulting models.

Most studies still focus on using a single optimization method, whether metaheuristic or non-metaheuristic, without conducting a comprehensive comparative analysis. Research on LSTM model hyperparameter optimization for predicting stock prices in the Indonesian banking sector remains relatively limited. Some studies also still rely on manual tuning or conventional methods, so the effectiveness of each optimization algorithm has not been comprehensively evaluated.

Given these conditions, there remains a research gap in the comparative analysis of hyperparameter optimization methods for LSTM models in predicting stock prices in the Indonesian banking sector. Therefore, this study proposes the application and comparison of GWO and Hyperband in the hyperparameter optimization of the LSTM model for predicting the stock price of Bank Rakyat Indonesia (BBRI). This research aims to compare the effectiveness of both optimization methods in producing the best hyperparameter configuration to improve the model's predictive performance.

Table 1. Literature review

No	Author & Year	Dataset	Model	Optimization	Evaluation Metric	Research Gap
1	Andika & Kusrini, (2025)	Purchasing of Natural Rubber Raw Materials	LSTM Bi-LSTM Stacked LSTM	*Bayesian optimization*, *Hyperband*, and Optuna	MAE & MSE	Does not compare metaheuristic algorithms
2	Dhake et al., (2023)	Solar Irradiance Forecasting	LSTM	Heap-Based Optimizer and Genetic Algorithm	MSE, MSLE, RMSE	Compares only metaheuristic algorithms
3	Dilawar & Shahbaz, (2025)	Heavy Industrial Motor Vibration Prediction	Stacked LSTM	*Bayesian optimization*	MSE & MAE	Uses only Bayesian optimization
4	Eren et al., (2025)	Urban Air Quality Prediction	Model LSTM	Random Search, Bayesian Optimization, and Hyperband	RMSE, MSE, MAE, R^2	Does not use metaheuristic algorithms
5	Maulana et al., (2026)	Stock Price Prediction	LSTM	Bayesian Optimization	MAPE & R^2	Does not use metaheuristic algorithms
6	Deswal, (2024)	Stock Price Prediction	LSTM	Hyperparameter Tuning	MSE & MAE	Does not discuss optimization algorithms or metaheuristics
7	Paygude et al., (2023)	Stock Price	LSTM	Manual	RMSE & MAPE	Does not use optimization algorithms
8	Song et al., 2025)	Corporate Financial Distress Prediction	CNN, Bi-LSTM	Hyperband	Accuracy, Precision, Recall, F1-Score	Does not use metaheuristic algorithms
9	Tanjung et al., (2025)	Forecasting Shallots and Garlic	LSTM	PSO	RMSE, MAPE, R^2	Uses only metaheuristic algorithms
10	Putra & Nurmawati, (2024)	Stock Price	BiLSTM & LSTM	Grid Search	MAPE, RMSE, MAE	Uses only grid search

METHOD

Research Design

This research is a comparative study of the algorithms used in optimizing the LSTM model to predict BBRI stock prices. The research employs a quantitative approach based on historical data. Figure 1 shows the research framework used in this study. There are four main stages: Raw Data, Data Preparation, Modeling, and Evaluation.

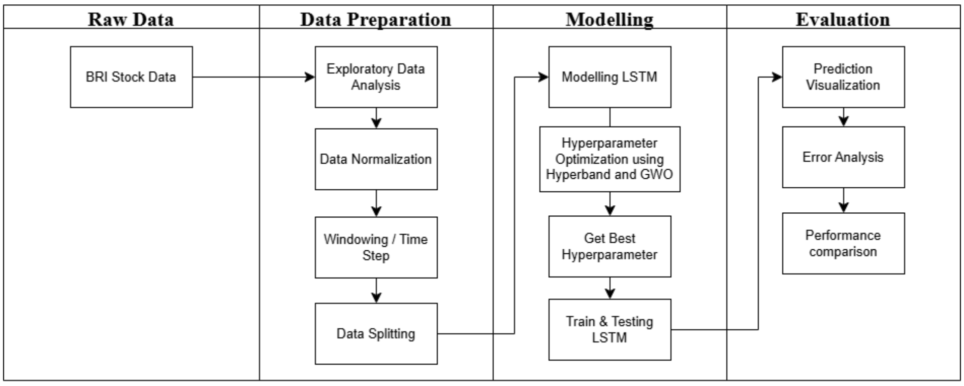


Figure 1: Research Framework

The raw data was obtained from the website https://finance.yahoo.com/ using the stock code "BBRI.JK." The data used consists of BBRI stock prices from 2018 to 2025. The BBRI.JK stock code refers to PT Bank Rakyat Indonesia (Persero) Tbk. BRI was selected because it is one of the banks with the highest transaction volumes in Indonesia. BBRI stock is also a blue-chip stock with a large market capitalization. BBRI stock is nicknamed the "people's stock" because it has the largest number of retail investors in Indonesia (Anam, 2026).

BBRI.JK stock data was collected over a 7-year period, from January 1, 2018, to December 31, 2025. The dataset consists of 1,966 rows and 6 columns: date, open, high, low, close, and volume. The "close" column was selected as the target variable because it represents the final price agreed upon by the market at the end of the trading session. The second stage involved data preparation to ensure the data was ready for use prior to the modeling process. The acquired dataset was first processed through data preprocessing to ensure it met the model's requirements. This stage includes several processes: data cleaning, data transformation, and splitting the data into training and testing sets. Data cleaning involves checking for missing values, followed by identifying patterns in the data.

The data, after undergoing normalization, is then shaped into a sequence using a windowing technique with a time step of 60. This time step is used so that the LSTM model can learn historical patterns and temporal relationships in stock price data more effectively. Additionally, a time step of 60 was chosen because it strikes a balance between the model's ability to capture data patterns and computational efficiency (S. Li et al., 2024). The data is then split into training and testing sets in an 80:20 ratio. This data split is performed to train the model using the majority of the historical data and to evaluate the model's ability to make predictions on data it has never seen before. The training data is used in the model training process, while the testing data is used to measure the model's predictive performance. The third stage is modeling, in which the model is used for analysis to predict stock prices. This stage applies the Hyperband and GWO algorithms to find the optimal hyperparameters for the LSTM.

LSTM

The LSTM model is an architecture derived from the RNN (Recurrent Neural Network) and was designed to address the limitations of RNNs. It can be viewed as an evolution of RNN architecture. This model was first introduced by Sepp Hochreiter and Jurgen Schmidhuber in 1997. The advantage of LSTM is its ability to process information more accurately; it incorporates an architecture that allows previous outputs to be reused as inputs, enabling the model to capture both short-term and long-term patterns (Nurashila et al., 2023). Figure 2 shows the general architecture of an LSTM, which consists of a forget gate, an input gate, and an output gate (Djaballah et al., 2024).

The forget gate determines whether information should be retained or discarded using a sigmoid activation function.

$$F_t = \sigma\left(W_f \times [h_{t-1}, X_t] + b_f\right)$$

The input gate determines how much new information will be stored in the memory cell's state. This process is designed to filter out irrelevant information so that it does not enter main memory.

$$i_t = \sigma(W_i \times [h_{t-1}, X_t] + b_i)$$

The output gate is used to control which information from the memory cell is sent as the output.

$$O_t = \sigma(W_o \times [h_{t-1}, X_t] + b_o)$$

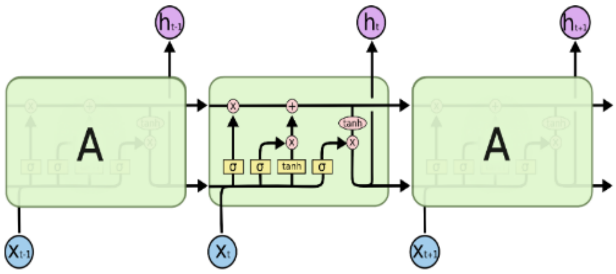


Figure 2: LSTM Architecture
Source: (Nurashila et al., 2023)

Building the LSTM model architecture shown in Figure 2 requires hyperparameters to train the model. Selecting the right parameters is crucial so that the model can learn patterns effectively and avoid overfitting. Hyperparameters are parameters specified before the model training process begins. They are used to configure deep learning models, such as LSTM, to achieve optimal performance. The optimized hyperparameters include the number of neurons, learning rate, number of epochs, dropout rate, batch size, and time steps (Nurmawati et al., 2025). The best parameters are identified using the Hyperband and GWO algorithms to obtain the most optimal values. The resulting hyperparameters are used by the model to perform the prediction process.

Referring to Andika & Kusrini, (2025), Table 2 shows the range of LSTM hyperparameters used in this study. The hyperparameters explored included units, dropout rate, learning rate, activation function, optimizer, batch size, and epochs. A range of units from 16 to 128 was used to determine the model's capacity to capture data patterns. The dropout rate, ranging from 0.0 to 0.5, serves to improve model generalization and prevent overfitting. The learning rate, ranging from 0.0001 to 0.01, controls the magnitude of weight updates during the training process. Activation functions used to learn complex patterns include ReLU, Tanh, and Sigmoid. The optimizers evaluated to minimize the loss function include Adam, Nadam, RMSprop, and SGD. The batch size processed in a single training iteration ranges from 16, 32, 64, and 128. The number of epochs used is 50, 100, and 200. Hyperparameter combinations will be optimized using Hyperband and GWO to produce the best combination.

Table 2. LSTM Hyperparameters

Hyperparameters	**Value**	**Description**
Units_n	16, 32, 48, 64, 80, 96, 112, 128	Determining the model's ability to capture data patterns
Dropout rate	0.0-0.5	Improving model generalization and preventing overfitting
Learning Rate	0.0001-0.01	Adjusting the magnitude of weight updates during the training process
Activation	ReLU, Tanh, Sigmoid	Learning more complex and nonlinear patterns
Optimizer	Adam,Nadam,RMSprop,SGD	Minimizing the loss function by updating the model's weights
Batch Size	16, 32, 64, 128	The amount of data processed in a single training iteration
Epoch	50, 100, 200	Determining the number of training iterations for the model across the entire training dataset

Hyperband

One of the algorithms frequently used for hyperparameter optimization is the Hyperband algorithm. This is because the Hyperband algorithm can efficiently test tens to hundreds of parameter combinations. The Hyperband algorithm's approach combines random search and successive halving, which eliminates fewer promising configurations early on (Duha & Putra, 2025). This makes the tuning process faster and significantly improves prediction results. According to Song et al., (2025), the steps of the Hyperband algorithm are as follows:

1. Parameter Initialization

   Determine the resource budget (R), bandwidth factor (η), and initial resource (r_0). R represents the total amount of available training resources. The bandwidth factor (η) is a parameter that controls the rate at which hyperparameter configurations are eliminated at each stage. The initial resource (r_0) is the initial number of resources allocated to each hyperparameter configuration during the initial evaluation stage.

2. Calculating the Maximum Number of Rounds (S_{max})

This calculation indicates the maximum number of iterations or brackets that can be performed during the optimization process. The purpose of this stage is to ensure that the hyperparameter evaluation process can be carried out efficiently by determining a resource allocation strategy. The mathematical formula for this stage is as follows.

$$S_{max} = [log_{\eta} \frac{R}{r_0}]$$

3. Perform several iterations of hyperparameter configuration and the elimination process.
   Several rounds are performed from $s = 0$ to $s = S_{max}$. In each s-th round, the number of configurations and the number of resources are initialized. In the initial stage, the results are assigned to a training resource allocation of r.

$$n = \left[\frac{R}{r_0(s+1)\eta^s}\right], r = r_0 \cdot \eta^s$$

   In each s-th iteration, the number of hyperparameter configurations (n_i) and the number of resources (r_i) used for each configuration in the I-th subphase (from 0 to s) are determined.

$$n = \left[\frac{n}{\eta^i}\right], r = r_0 \cdot \eta^i$$

   Based on Successive Halving, low-performance configurations are eliminated. Resources are allocated to configurations with good performance. The elimination ratio for this stage is $1 - \frac{1}{\eta}$, so only $\frac{1}{\eta}$ of the best configurations are retained for the next stage.
4. Selecting the Optimal Configuration
   The configuration that delivered the best performance at the end of all rounds.

Referring to (L. Li et al., 2018), the hyperband parameter settings used in this study are shown in Table 3. This study used a max_epochs value of 200 for the Hyperband algorithm based on experimental results with resource budgets of 50, 100, and 200 epochs. In the hyperparameter search process, the validation loss (val_loss) metric was used with a factor value of 3, which retained one-third of the best configurations at each selection stage, in accorandce with the Hyperband algorithm's recommendations.

Table 3. Hyperband Settings

Setting	Value	Description
Objective	Val_Loss	Determining performance during the tuning process
Resource budget (R),	50, 100, 200	The maximum number of epochs assigned to each hyperparameter configuration
Bandwidth factor (η),	3	Setting the elimination rate to determine how many of the best configurations are retained

Grey Wolf Optimization (GWO)

The GWO is an algorithm that draws inspiration from the hunting behavior of the grey wolf (Canidae Lupus). The grey wolf is an apex predator that lives and hunts in packs, employing strategies of searching for, surrounding, and attacking prey. These packs are considered to be at the top of the food chain and possess a highly structured social hierarchy (Yosviansyah & Rizal, 2024). The wolf leadership hierarchy is divided into four types: alpha (α), beta (β), delta (δ), and omega (ω). The social hierarchy is represented mathematically, with the optimal solution represented by the alpha wolf (α), followed by the beta (β) and delta (δ) wolves, and other potential solutions represented by the omega wolf (ω) (Mahardhika et al., 2025).

The GWO algorithm is mathematically described by the following equation:

$$\vec{D}_{\alpha} = |C_1 . \vec{X}_{\alpha} - \vec{X}_i|; \ \vec{D}_{\beta} = |C_2 . \vec{X}_{\beta} - \vec{X}_i|; \ \vec{D}_{\delta} = |C_3 . \vec{X}_{\delta} - \vec{X}_i|$$

$$\vec{X}_1 = \vec{X}_{\alpha} - A_1 .(\vec{D}_{\alpha}); \ \vec{X}_2 = \vec{X}_{\beta} - A_2 .(\vec{D}_{\beta}); \ \vec{X}_3 = \vec{X}_{\delta} - A_3 .(\vec{D}_{\delta})$$

$$\vec{X}_1(t+1) = \frac{X_1 + X_2 + X_3}{3}$$

Let $A = 2a.r_1 - a$ and , $C = 2.r_2,\ r_1, r_2$ where r_1 and r_2 are random values in the range [0, 1], and the value of a decreases linearly from 2 to 0 with each iteration. The values of A and C are vector coefficients. The value of X is the position of the grey wolf, while D is a vector representing the distance between the wolf's position and its prey, t is the time step, and $\vec{X}_1(t+1)$ is the wolf's position at the next time step. As for the GWO algorithm, the process of finding the best hyperparameters uses a population-based approach. The optimization process is carried out by evaluating hyperparameter combinations and calculating the error value as a fitness function. Then, the positions of the three best solutions (alpha, beta, and delta) are updated. This process is repeated until a predetermined condition is reached. The GWO settings used to identify the optimal hyperparameter combinations are presented in Table 4. Referring to Zhou et al., (2025), a population size of 60 and a maximum of 30 iterations constitute a configuration that demonstrates rapid convergence. Meanwhile, Kazemi et al., (2026) use a population size of 10 with 20 iterations to configure the GWO parameters, whereas Hao et al., (2024) employ a population size of 10 with 50 iterations.

Table 4. GWO Settings

Setting	Value	Description
Population	10 and 60	Determines the number of searches during the optimization process
Iteration	20, 30, and 50	The maximum number of optimization iterations executed to determine the best parameter configuration

Evaluation Metrics

After obtaining the best hyperparameters for each of the two algorithms, the models were evaluated using the RMSE metric and compared across the two algorithms. In this study, the evaluation metric used was RMSE (Root Mean Square Error). RMSE is the square root of the difference between the actual values and the predicted values, divided by the total number of data points. This evaluation metric is used to measure the average prediction error. A model is considered better if its RMSE value is smaller (Listyaningrum et al., 2025). Mathematically, RMSE is expressed by the following equation, where Y' is the predicted value, Y is the actual value, and n is the number of data points.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y - Y')^2}$$

RESULTS AND DISCUSSION

This study begins by applying LSTM model without hyperparameter optimization to predict Bank Rakyat Indonesia's stock price. The RMSE obtained is 84.24, using a configuration parameter consisting of a time step of 60, two LSTM layers of 64 units each, a dropout rate of 0.2, the Adam optimizer, a batch size of 32, and 100 epochs. Figure 3 shows a visualization of the actual and predicted prices from the LSTM without an optimization algorithm.

These results indicate that the LSTM model can effectively capture temporal dependencies in stock price data. However, its predictive performance remains dependent on the manual hyperparameter tuning process. To improve the model's performance, hyperparameter optimization is performed using Hyperband and GWO.

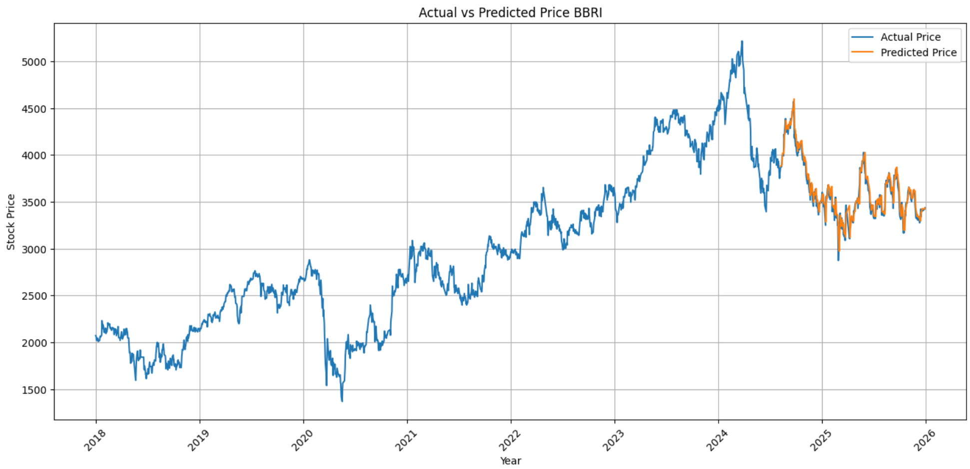


Figure 3: Predicted Price Pure LSTM

Table 5 shows the best hyperparameters from the Hyperband algorithm used during the model training and testing phases. Table 5 shows that increasing the resource budget from 100 to 200 yields only a slight improvement in RMSE, indicating that the LSTM model has reached convergence. Nevertheless, a *max_epochs* value of 200 still produces the best performance and was selected as the optimal configuration because it strikes a balance between prediction accuracy and computational efficiency.

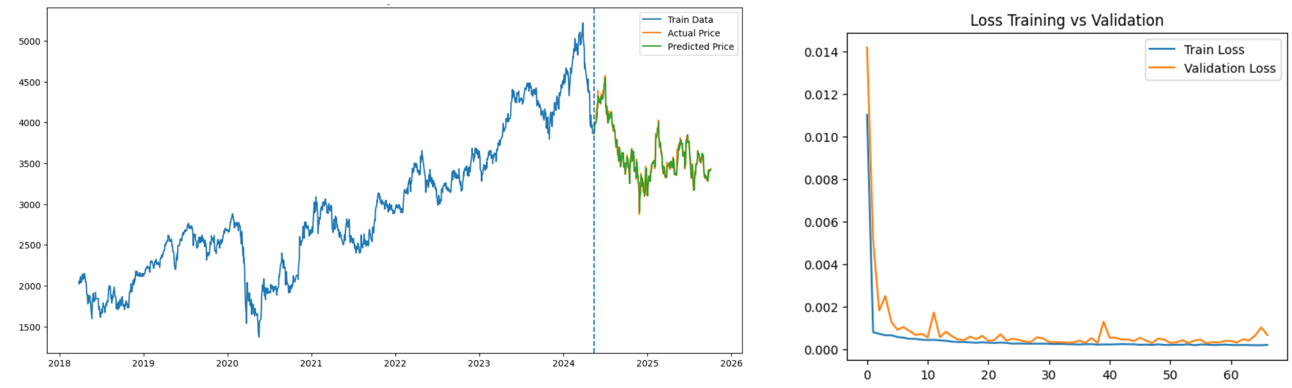
Table 5. Hyperband Hyperparameter Results

Parameter Settings	Best Hyperparameter	RMSE	Computation Time
R = 50, factor = 3, Train epoch = 200	'units': 32, 'dropout': 0.0, 'learning_rate': 0.01, 'activation': 'tanh', 'optimizer': 'adam', 'batch_size': 16	83.56	77 minutes
R = 100, factor = 3, Train epoch = 200	'units': 64, 'dropout': 0.1, 'learning_rate': 0.01, 'activation': 'tanh', 'optimizer': 'nadam', 'batch_size': 32	77.64	219 minutes
R = 200, factor = 3, Train epoch = 200	'units': 128, 'dropout': 0.0, 'learning_rate': 0.001, 'activation': 'relu', 'optimizer': 'adam', 'batch_size': 16	77.60	271 minutes

The RMSE values obtained show that the Hyperband algorithm is capable of finding a sufficiently good combination of hyperparameters to improve the LSTM model's performance in predicting BBRI stock prices. The use of the Adam optimizer and the ReLU activation function helps the model learn temporal patterns in the time-series data more stably. Additionally, a batch size of 16 allows for a more detailed learning process on the training data.

Figure 4(a) shows a graph of the Predicted Price, indicating that the model performs well in most tests. Notably, the model is able to track the movements of the actual data, although there are some discrepancies at certain points.

Figure 4(b) shows that the validation loss still experiences significant fluctuations, especially in the early epoch. However, in subsequent periods, the model was able to learn the historical data patterns well. The difference between the training loss and the validation loss was also not too large, so the model showed no signs of significant overfitting. Hyperband was able to produce a model through a relatively fast tuning process by eliminating suboptimal parameter configurations in the early stages of training.



(a) (b)

Figure 4: (a) LSTM-Hyperband Predicted Price Chart, (b) LSTM-Hyperband Loss Model

For the hyperparameter optimization of LSTM using GWO algorithm, the best hyperparameters for the GWO algorithm are shown in Table 6. The comparison results in Table 5 show that the configuration with a population of 60 and 30 iterations yields the lowest RMSE value of 75.93 compared to the other configurations, although it requires a longer computation time. These results indicate that increasing the number of search agents and iterations expands the exploration of the hyperparameter space, thereby yielding more optimal parameter combinations.

Table 6. GWO Hyperparameter Results

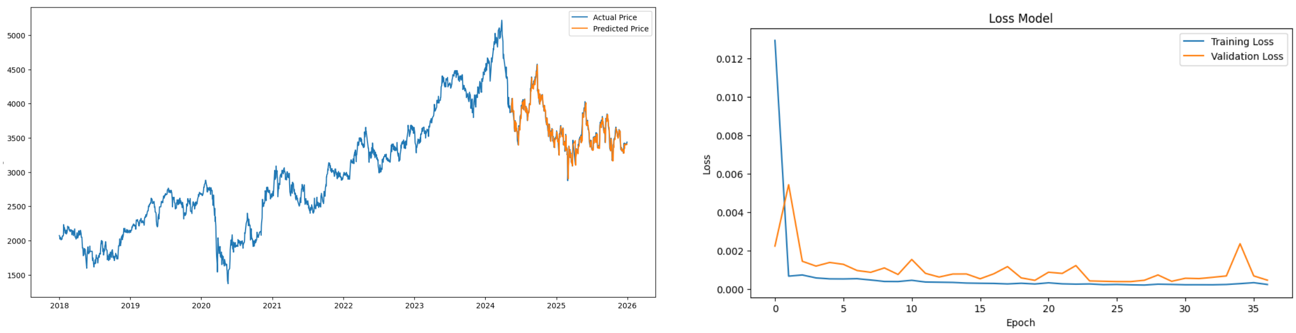
Population	Iteration	Best Hyperparameter	RMSE	Computation Time
10	20	'units': 64 'dropout': 0.07103537769881944 'learning_rate': 0.004546912258211178 'activation': 'tanh' 'optimizer': 'adam' 'batch_size': 16 'epochs': 50	80.83	297 Minutes
10	50	'units': 80 'dropout': 0.00571573571252854 'learning_rate': 0.00840979321425187 'activation': 'tanh' 'optimizer': 'adam' 'batch_size': 16 'epochs': 50	77.91	226 Minutes
60	30	'units': 96 'dropout': 0.0007674827092990222 'learning_rate': 0.003874791128170863 'activation': 'tanh' 'optimizer': 'adam' 'batch_size': 16 'epochs': 100	75.93	927 Minutes

The results show that the GWO algorithm is capable of producing better prediction performance than Hyperband because it achieves a lower RMSE value. A lower RMSE value indicates a smaller prediction error, which indicates that the model output is more consistent with the actual data. This is due to the exploration and exploitation mechanisms in GWO, which are performed iteratively by updating the wolf's position to obtain the optimal combination of hyperparameters. Using a larger number of units allows the model to learn more complex data patterns, while a smaller learning rate helps stabilize the training process.

However, improving performance using GWO requires more computational time compared to Hyperband. This is because the fitness evaluation process is performed on the entire population at every iteration, resulting in a significant increase in the number of model evaluations. With a population of 60 and 30 iterations, the optimization process results in a larger total number of model evaluations, thus requiring greater computational resources. Figure 5(a) shows a visualization of the predicted prices from the LSTM-GWO model, which aligns well with the actual data. The prediction patterns closely follow the data patterns across most of the test models.

In the LSTM-GWO loss model shown in Figure 5(b), the validation loss for the GWO model spiked initially, although it eventually followed the subsequent historical pattern. The spike that occurred in the early epochs did not indicate significant overfitting. The closeness of the two curves indicates that the model performs well when using GWO. This suggests that hyperparameter optimization using GWO is capable of producing more optimal parameter configurations, resulting in a model with relatively stable performance. Based on Table 6, the optimal hyperparameter combination generated by the GWO algorithm consists of 96 units, a dropout rate of 0.0007674827092990222, a learning rate of 0.003874791128170863, the tanh activation function, the Adam optimizer, a batch size of 16, and 100 epochs. This combination produced the lowest RMSE value among the other combinations.

Overall, the results of this study show that GWO is more effective at obtaining optimal hyperparameters to improve the prediction accuracy of the LSTM model, while Hyperband is superior in terms of computational efficiency.



(a) (b)

Figure 5: (a) LSTM-GWO Predicted Price Chart, (b) LSTM-GWO Model Loss Chart

Prediction Results

A comparison graph between actual data and prediction results shows that LSTM model, optimized using GWO, is able to accurately track the price movement patterns of Bank Rakyat Indonesia stock. The resulting prediction patterns exhibit trends similar to the actual data, both during uptrends and downtrends. Although there are slight discrepancies in predictions during periods of high volatility, the model remains capable of consistently capturing the direction of stock price movements. These results indicate that hyperparameter optimization using GWO enhances the model's generalization ability in learning stock time-series data patterns.

Furthermore, the training loss and validation loss graphs for both models show a significant decrease in loss during the early epochs, followed by a tendency toward stability in subsequent epochs. The relatively small difference between training loss and validation loss indicates that the models did not experience significant overfitting during the training process. However, based on the resulting loss patterns, the LSTM-GWO model demonstrated greater training stability and better generalization ability compared to LSTM-Hyperband. This suggests that the GWO algorithm is more effective at finding the optimal combination of hyperparameters in LSTM models for stock price prediction.

The optimization results show that the LSTM-GWO model produced a lower RMSE value (75.93) compared to the LSTM model without optimization (84.24). This reduction in the RMSE value indicates that GWO is capable of finding a more optimal combination of hyperparameters, thereby improving the model's generalization ability in predicting stock price movements. Figure 6 shows the BRI stock price forecast generated by LSTM-GWO.

Figure 6: Forecast BRI Stock Price LSTM-GWO

The results of stock price forecasting for Bank Rakyat Indonesia (BBRI) using the LSTM model with GWO indicate a trend of rising stock prices over the next year. The resulting forecasting pattern appears relatively stable and is able to follow historical stock data trends without showing overly extreme fluctuations. These results indicate that the LSTM-GWO model is capable of effectively learning the temporal patterns of stock data, thereby producing more consistent predictions that closely approximate actual patterns.

The superior forecast performance of the LSTM-GWO model aligns with evaluation results showing a lower RMSE value compared to other models. This suggests that hyperparameter optimization using GWO enhances the model's generalization ability in predicting stock price movements. However, the forecasting results in this study cannot be used as a definitive reference because stock price movements are influenced by various external factors, such as economic conditions, corporate policies, and market sentiment. Therefore, the forecasting in this study is primarily focused on evaluating the model's ability to learn historical patterns and predict future trends in stock prices.

CONCLUSION

This study successfully applied an LSTM model with hyperparameter optimization using Hyperband and GWO to predict the stock price of Bank Rakyat Indonesia (BBRI). The evaluation results show that both optimization methods were able to improve the model's performance compared to an LSTM without optimization, which yielded an RMSE of 84.24. The LSTM-GWO model delivered the best performance with an RMSE of 75.93, while LSTM-Hyperband achieved an RMSE of 77.6. These results indicate that GWO is more effective at identifying the optimal combination of hyperparameters, resulting in predictions that more closely align with actual data.

GWO requires more computational time compared to Hyperband, which offers the advantage of more efficient tuning by eliminating certain hyperparameter configurations early in the training phase. Therefore, the choice of optimization method can be tailored to balance the need for model accuracy and computational efficiency. The forecasting results also indicate a trend of rising BBRI stock prices in the coming period, although the predictions remain subject to various external factors such as economic conditions, corporate policies, and market sentiment.

For future research, it is recommended to use data with additional features, such as technical indicators and macroeconomic data, to improve the model's prediction accuracy. Additionally, other optimization methods or further development of the Long Short-Term Memory architecture can be applied for comparison to achieve better prediction performance.

REFERENCES

Anam, K. (2026, April 28). *Bukti BBRI Jadi Saham Rakyat, Jumlah Investor Ritel Capai 700 Ribu*. CNBC Indonesia. https://www.cnbcindonesia.com/market/20260428165642-17-730659/bukti-bbri-jadi-saham-rakyat-jumlah-investor-ritel-capai-700-ribu

Andika, R., & Kusrini, K. (2025). Optimasi Hyperparameter Model Lstm Dan Variannya Untuk Peramalan Pembelian Bahan Baku Karet Alam. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, *10*(3), 2627–2639. https://doi.org/10.29100/jipi.v10i3.7567

Deswal, V. (2024). Hyperparameter Tuning of the LSTM model for Stock Price Prediction. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(4), 705–712.

Dhake, H., Kashyap, Y., & Kosmopoulos, P. (2023). Algorithms for Hyperparameter Tuning of LSTMs for Time Series Forecasting. *Remote Sensing*, *15*(8), 2076. https://doi.org/10.3390/rs15082076

Dilawar, M., & Shahbaz, M. (2025). A Bayesian Optimized Stacked Long Short-Term Memory Framework for Real-Time Predictive Condition Monitoring of Heavy-Duty Industrial Motors. *Computers, Materials & Continua*, *83*(3), 5091–5114. https://doi.org/10.32604/cmc.2025.064090

Djaballah, S., Saidi, L., Meftah, K., Hechifa, A., Bajaj, M., & Zaitsev, I. (2024). A hybrid LSTM random forest model with grey wolf optimization for enhanced detection of multiple bearing faults. *Scientific Reports*, *14*, 23997. https://doi.org/10.1038/s41598-024-75174-x

Duha, A. M., & Putra, A. T. (2025). Optimasi Hyperparameter Pada Model Hybrid Bidirectional LSTM-GRU Untuk Prediksi Harga Saham Bank. *Bulletin of Computer Science Research*, *6*(1), 332–341. https://doi.org/10.47065/bulletincsr.v6i1.903

Eren, B., Erden, C., Atalı, A., & Ozdemir, S. (2025). A comparative analysis of hyperparameter optimization using LSTM-based deep learning models for urban air quality predictions. *Ain Shams Engineering Journal*, *16*(12), 103786. https://doi.org/10.1016/j.asej.2025.103786

Hao, J., Shang, S., Yuan, J., & Li, J. (2024). Do multisource data matter for NGP prediction? Evidence from the G-LSTM model. *Heliyon*, *10*(12), e33387. https://doi.org/10.1016/j.heliyon.2024.e33387

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kazemi, M., Samani, R. N., & Kariminejad, N. (2026). Optimizing LSTM networks and feature selection algorithms using GEE data. *PLOS ONE*, *21*(4), e0347858. https://doi.org/10.1371/journal.pone.0347858

Khayat, A., Kissaoui, M., Bahatti, L., Raihani, A., Errakkas, K., & Atifi, Y. (2025). Efficient day-ahead energy forecasting for microgrids using LSTM optimized by grey wolf algorithm. *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, *13*, 101054. https://doi.org/10.1016/j.prime.2025.101054

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). *Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization*.

Li, S., Wang, T., Li, G., Skulstad, R., & Zhang, H. (2024). Short-term ship roll motion prediction using the encoder–decoder Bi-LSTM with teacher forcing. *Ocean Engineering*, *295*, 116917. https://doi.org/10.1016/j.oceaneng.2024.116917

Listyaningrum, R., Purwanto, R., Prasetyanti, D. N., Vikasari, C., & Pratiwi, A. F. (2025). Pemanfaatan Algoritma Random Forest Regression dalam Memprediksi Kepuasan Mahasiswa Terhadap Dosen. *Infotekmesin*, *16*(2), 408–416. https://doi.org/10.35970/infotekmesin.v16i2.2808

Mahardhika, F., Yulianti, K., & Kustiawan, C. (2025). Algoritma Grey Wolf Optimizer dan Model Mean Absolute Deviation Untuk Optimisasi Portofolio Saham. *Jurnal EurekaMatika*, *13*(1), 15–24. https://doi.org/10.17509/jem.v13i1.69680

Maulana, M. I., Anam, S., & Bukhori, H. A. (2026). Bayesian-Optimized LSTM Framework for Accurate Stock Price Prediction. *Journal of Applied Informatics and Computing*, *10*(2), 1282–1290. https://doi.org/10.30871/jaic.v10i2.11756

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mubarak, M. N., Susandri, S., Zamsuri, A., & Ramadhani, M. (2025). Pendekatan Wavenet-Inspired Dan LSTM Untuk Prediksi Magnitudo Gempa Sebagai Upaya Transformasi Digital Mitigasi Bencana Di Indonesia. *SEMASTER: Seminar Nasional Teknologi Informasi & Ilmu Komputer*, *4*(1), 368–375.

Nurashila, S. S., Hamami, F., & Kusumasari, T. F. (2023). Perbandingan Kinerja Algoritma Recurrent Neural Network (Rnn) Dan Long Short-Term Memory (Lstm): Studi Kasus Prediksi Kemacetan Lalu Lintas Jaringan Pt Xyz. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, *8*(3), 864–877. https://doi.org/10.29100/jipi.v8i3.3961

Nurmawati, E., Abaysa, R., & Putra, R. A. (2025). Optimalisasi Portofolio Saham Syariah Berbasis Prediksi Menggunakan Long Short-Term Memory (LSTM). *Jurnal Informatika*, *10*(2).

Paygude, P., Chavan, P., Gayakwad, M., Gupta, K., Joshi, S., Gopika, G., Joshi, R., Gonge, S., & Kotecha, K. (2023). Optimizing Hyperparameters for Enhanced LSTM-Based Prediction System Performance. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(10s), 203–213. https://doi.org/10.17762/ijritcc.v11i10s.7620

Purnamasari, S. A. (2025). *Mekanisme Perkembangan Pasar Modal Sebagai Salah Satu Produk Investasi di Masyarakat*.

Putra, R. A., & Nurmawati, E. (2024). Prediction-based Stock Portfolio Optimization Using Bidirectional Long Short-Term Memory (BiLSTM) and LSTM. *Scientific Journal of Informatics*, *11*(3), 609–620. https://doi.org/10.15294/sji.v11i3.5941

Remetwa, D. P. A., Cahyadi, L., Ferdinand, F. V., Saputra, K. V. I., & Teja, K. (2025). Analysis Of Long Short – Term Memory (Lstm) Parameters In Predicting Ihsg. *JOHME: Journal of Holistic Mathematics Education*, *9*(2), 213–224. (Indonesia). https://doi.org/10.19166/johme.v9i2.10220

Salgotra, R., Sharma, P., Raju, S., & gandomi, A. H. (2024). A Contemporary Systematic Review on Meta-heuristic Optimization Algorithms with Their MATLAB and Python Code Reference. *Archives of Computational Methods in Engineering*, *31*(3), 1749–1822. https://doi.org/10.1007/s11831-023-10030-1

Setiyani, S. H. (2025). Prediksi Pergerakan Harga Saham Menggunakan Quantum Machine Learning Berbasis Variational Quantum Circuits. *Jurnal Informatika dan Teknik Elektro Terapan*, *13*(3S1). https://doi.org/10.23960/jitet.v13i3S1.8038

Song, Y., Chiangpradit, M., & Busababodhin, P. (2025a). Hyperband-Optimized CNN-BiLSTM with Attention Mechanism for Corporate Financial Distress Prediction. *Applied Sciences*, *15*(11), 5934. https://doi.org/10.3390/app15115934

Song, Y., Chiangpradit, M., & Busababodhin, P. (2025b). Hyperband-Optimized CNN-BiLSTM with Attention Mechanism for Corporate Financial Distress Prediction. *Applied Sciences*, *15*(11), 5934. https://doi.org/10.3390/app15115934

Tanjung, M. A., Sari, A. P., & Junaidi, A. (2025). Optimization of LSTM Hyperparameters Using PSO for Forecasting Shallots and Garlic. *Bit-Tech*, *8*(1), 416–426. https://doi.org/10.32877/bt.v8i1.2569

Yosviansyah, M. N., & Rizal, Y. (2024). Optimasi Persediaan Bahan Baku Dan Produksi Usaha Ganepo Putri Yose Dengan Menggunakan Algoritma Grey Wolf Optimizer. *MATHunesa: Jurnal Ilmiah Matematika*, *12*(1), 94–100. https://doi.org/10.26740/mathunesa.v12n1.p94-100

Zhou, W., Liu, S., Guo, J., Liu, N., Li, Z., & Xie, C. (2025). ARIMA-Kriging and GWO-BiLSTM Multi-Model Coupling in Greenhouse Temperature Prediction. *Agriculture*, *15*(8), 900. https://doi.org/10.3390/agriculture15080900