JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
https://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Zein, M. H., Yudistira, N., & Adikara, P. P. (2024). Indonesian stock price prediction using neural basis expansion analysis for interpretable time series method. *Journal of Information and Communication Technology, 23*(3), 361-392. https://doi.org/10.32890/jict2024.23.3.1

Indonesian Stock Price Prediction Using Neural Basis Expansion Analysis for Interpretable Time Series Method

*1Muhamad Harun Zein, 2Novanto Yudistira & 3Putra Pandu Adikara
Informatics Engineering Department,
Brawijaya University, Indonesia

*1mharunzein@student.ub.ac.id
2yudistira@ub.ac.id
3adikara.putra@ub.ac.id
*Corresponding author

Received: 3/9/2023 Revised: 3/7/2024 Accepted: 8/7/2024 Published: 28/7/2024

ABSTRACT

The stock market is an attractive investment venue for many individuals and companies. However, unexpected share price fluctuations can cause significant financial losses. In stock investment, predicting stock price movements is the most frequently discussed topic because it allows investors to make the right investment decisions to make big profits. Therefore, a model is needed to predict future stock prices, one strategy for maximising investment profits. New state-of-the-art deep learning architectures for time series forecasting are being developed yearly, making them more accurate than ever. The most commonly used network for such a solution is Long Short-Term Memory (LSTM) architecture, but it has limitations such as long training time and interpretability. This study aims to evaluate another state-of-the-art solution, Neural Basis Expansion Analysis for Interpretable Time

Series (N-BEATS), in comparison with LSTM by utilising historical data of PT Bank Central Asia Tbk (one of the banking companies in Indonesia) from 25 March 2013 to 21 March 2023. N-BEATS is a relatively new variable method that can produce accurate predictions using neural networks. This architecture has advantages such as interpretability, seamless applicability across diverse target domains without requiring modifications, and fast training. Based on tests carried out with prediction errors measured using the Mean Average Percentage Error (MAPE), it was found that the N-BEATS model outperformed the LSTM model with a MAPE value of 1.05 percent. In conclusion, this research shows the use of a new method of deep learning algorithms to predict stock prices, which contributes to facilitating stock buying and selling decisions by investors.

Keywords: Prediction, stock price, neural basis expansion analysis for interpretable time series, mean absolute percentage error.

INTRODUCTION

The rapid increase in stock investors in Indonesia, as evidenced by statistical data from the Indonesian Central Securities Depository, underscores the growing interest in stock market participation and the importance of effective investment decision-making. Predicting stock price movements is paramount for investors seeking to maximise returns and mitigate risks in this volatile market environment. Traditional machine learning approaches, such as Support Vector Machine (SVM) and linear regression, have been utilised to forecast stock prices with varying degrees of success. Setiawan et al. (2018) applied SVM to predict the Indonesian stock prices of PT Astra International Tbk, PT Garuda Indonesia Tbk, and PT Indosat Tbk, yielding accuracy rates ranging from 52.06 percent to 55.52 percent. Similarly, Keren and Oetama (2019) utilised linear regression to predict the stock price of PT Bank Central Asia Tbk, albeit with a high Mean Absolute Percentage Error (MAPE) of 234.64. These findings highlight the limitations of traditional machine learning models in capturing the complex dynamics of stock price movements, particularly in highly volatile markets.

In contrast, deep learning methodologies have emerged as promising alternatives for stock price prediction due to their ability to capture nonlinear relationships and temporal dependencies in data. Tiyara (2021) demonstrated the efficacy of the Long Short-Term Memory (LSTM) model in forecasting the stock prices of PT Bank Rakyat

Indonesia Tbk and PT Bank Central Asia Tbk, achieving significantly lower MAPE values than traditional methods. These models have proven their efficiency in effectively handling time-series data (Bhanusree et al., 2023). Furthermore, recent advancements in deep learning have led to the development of novel models like Neural Basis Expansion Analysis for Interpretable Time Series (N-BEATS), which offer enhanced performance in time series forecasting tasks. Bulatov (2020) compared N-BEATS with LSTM and Autoregressive Integrated Moving Average (ARIMA) models, revealing N-BEATS superior performance in evaluation metrics. This underscores the potential of N-BEATS as a state-of-the-art solution for stock price prediction, owing to its deep neural architecture and advantageous characteristics such as interpretability and fast training. Oreshkin et al. (2019) provide insights into the architecture and features of N-BEATS, highlighting its suitability for predicting time series data, including stock prices. The incorporation of backward and forward residual links, along with fully connected layers, enables N-BEATS to capture complex temporal patterns and exhibit robust performance across diverse domains without necessitating extensive modifications.

Given the demonstrated advantages of deep learning models, particularly N-BEATS, the researcher's intention to apply N-BEATS for stock price prediction in the Indonesian market is well-founded. By leveraging the capabilities of N-BEATS, investors can gain valuable insights into future stock price movements, facilitating informed investment decisions and potentially maximising returns. In summary, while traditional machine learning models have limitations in predicting stock prices accurately, deep learning methodologies, including LSTM and N-BEATS, offer promising solutions for overcoming these challenges. Integrating advanced predictive models into investment strategies can empower investors to navigate the complexities of the stock market more effectively, ultimately contributing to improved market efficiency and investor outcomes.

LITERATURE REVIEW

Stock

Shares are one of the increasingly popular types of investment products in recent times. According to the Indonesia Stock Exchange (IDX), shares represent a person's or entity's (business entity) capital participation in a company or limited liability corporation. By providing such capital, the party possesses a claim to the company's earnings, a claim to the company's assets, and the right to attend the General Meeting of Shareholders (GMS) (Bursa Efek Indonesia, 2023). There are two advantages investors gain from buying or owning shares: dividends and capital gain. Dividends are a portion of the company's profits obtained and distributed to its shareholders as a reward for their willingness to invest in the company (Rudianto, 2012). Capital gain is the financial gain generated when an asset's selling price exceeds its initial purchase price. It represents the positive difference between the two amounts, indicating a profitable outcome.

As an investment instrument, stocks carry capital loss and liquidation risks. Capital loss is the opposite of capital gain, incurred when an investor sells shares lower than the purchase price. Meanwhile, the possibility of liquidation risk emerges when a corporation where an individual holds shares is officially declared insolvent by a court or undergoes dissolution. In this case, shareholders' claims are given the lowest priority after all the company's debts (from the proceeds of the sale of corporate assets) have been settled. If there is a surplus from the sale of the company's assets, the remainder is distributed proportionally among all shareholders. However, shareholders will not receive any proceeds from the liquidation if the company has no remaining wealth. This situation presents the most significant risk to shareholders. Therefore, shareholders need to monitor the company's developments continuously.

In the secondary market or daily stock trading activities, stock prices fluctuate through rises and falls. Stock prices are dictated by the interplay between the availability and desire for stocks. To elaborate, stock prices are subject to the impact of the availability and desire surrounding these stocks. These factors are propelled by diverse elements, including stock-specific aspects like a company's performance and the industry it operates in, as well as macroeconomic considerations such as interest rates, inflation, exchange rates, and non-economic variables such as social and political factors.

Time Series Analysis

Time series analysis is employed to uncover growth patterns or changes in the past that can be utilised to predict future patterns. While time series analysis does not provide definite answers about the future, it holds considerable significance in forecasting and minimises forecasting errors (Draper & Smith, 1998). Time series data consists of variables collected based on the time series of specific categories or individuals over a certain period of time. The time units can be seconds, minutes, hours, days, weeks, months, or years. Numerous data collections manifest as time series, comprising monthly records of goods dispatched from a manufacturing facility, weekly data reflecting road accident occurrences, daily measurements of precipitation, and hourly observations concerning the output yield of a chemical procedure (Box et al., 2015).

Neural Basis Expansion Analysis for Interpretable Time Series Forecasting Architecture

In the fourth Makridakis competition, known as M4 (Makridakis et al., 2020), the winning solution was a hybrid model of Exponential Smoothing and Recurrent Neural Networks (ES-RNN) developed by Uber (Smyl et al., 2018). One year later, Elemental AI (co-founded by Yoshua Bengio) released N-BEATS, a pure deep-learning model that outperformed the M4-winning ES-RNN model by 3 percent (Oreshkin et al., 2019). N-BEATS is an advanced deep learning method that employs both backward and forward residual connections and a deep stack of fully connected layers to predict univariate time series data. Each stack comprises a sequence of blocks, each made up of multi-layer fully connected and infused with rectified linear unit (ReLU) nonlinearities (Kabir et al., 2024). N-BEATS framework offers several advantageous characteristics, including interpretability, ease of implementation across different target domains without requiring modifications, and efficient training. In contrast to other architectures, N-BEATS does not depend on utilising specialised techniques for time series feature engineering or input scaling (Oreshkin et al., 2019).

N-BEATS architecture comprises a series of stacks, each containing multiple basic blocks. A detailed depiction of the architecture can be observed in Figure 1. On the other hand, Figure 2 provides a specific representation of the internal structure of the basic block, which exhibits a fork-like design. The input to the model is derived from the time series and encompasses a look-back period of nH, where n signifies the number of data points necessary to forecast a single data point in the future. H represents the forecast horizon, indicating the number of data points to be predicted. In this instance, the value of H is 5, resulting in a look-back period equivalent to 15 data points. Consequently, a 15-dimensional input from the look-back period is fed through four layers of Fully-Connected (FC) + ReLU stacks, subsequently divided into two segments. These two segments are then processed by additional FC layers, yielding two outputs: a 15-dimensional backcast vector and a 5-dimensional forecast vector (representing five predicted data points). Thus, the basic block provides predictions for both the forthcoming data points and the subsequent input data in the form of the backcast. Operations in N-BEATS architecture can be seen in Equations 1 and 2.

Figure 1

N-BEATS Architecture

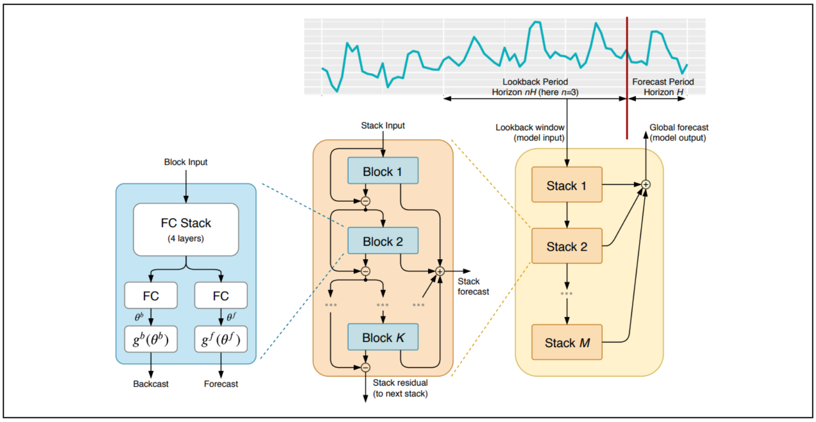
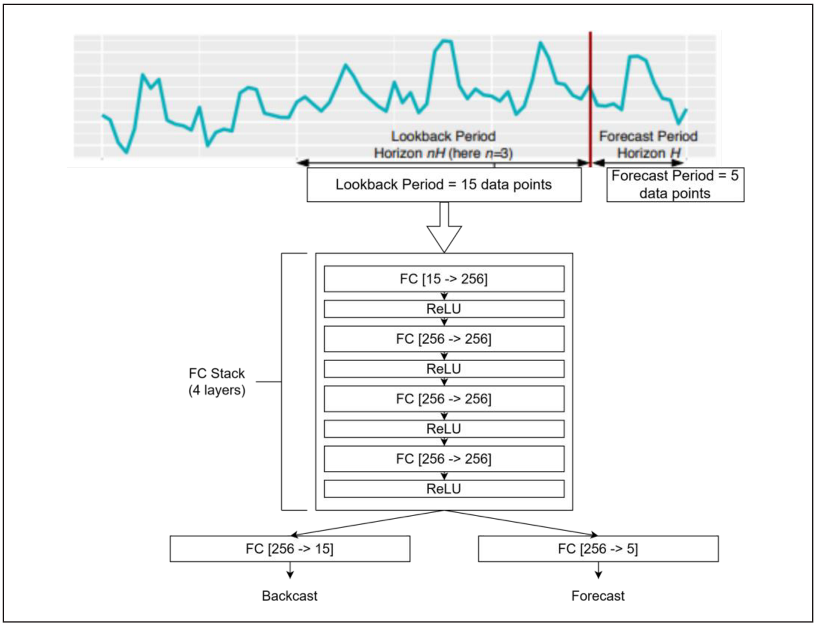


Figure 2

N-BEATS Basic Building Block



$$FC = \sum(w \times x) + b \tag{1}$$

$$x_\ell = x_{\ell-1} - \hat{x}_{\ell-1} \tag{2}$$

where *w* is weight, *x* is input and *b* is bias.

Multiple elementary N-BEATS blocks are merged into a unified stack, with the arrangement of the elementary blocks within the stack adhering to a dual residual stacking method. This stacking approach is termed "dual residual stacking" due to the presence of two arithmetic operations generated by the elementary block: backcast and forecast (Oreshkin et al., 2019). The backcast and forecast vectors are two results derived from the initial fundamental block following the input processing during the look-back period. The backcast vector is subsequently utilised to determine the input for the subsequent block, which entails the subtraction of the elements within the backcast from the newly obtained look-back input. By subtracting the backcast from the new look-back input, a vector is obtained that only incorporates information that was not adequately captured by the first block. This vector is then passed as input to the subsequent block (Oreshkin et al., 2019). Therefore, the input to each subsequent block will consist of the subtraction of the elements from the output and the backcast input of the preceding block.

The output generated by the last block in the stack is referred to as the stack backcast output. The forecast output from all blocks within the stack is utilised to calculate the stack forecast output, which is the summation based on the elements of all the outputs. Each stack is also used to construct a larger stack. The preceding explanation indicates that each input stack serves as the backcast output of the previous stack, encompassing the elements not learned by the preceding stack (Bulatov, 2020). The output of the stack forecast is aggregated across elements to generate the final global forecast vector. The discrepancy between the actual values and predictions is calculated using the MAPE. At the end of each training cycle, the model's gradient is updated based on the loss value (Oreshkin et al., 2019). The formula for calculating dual residual stacking can be observed in Equations 3 and 4.

$$x_\ell = x_{\ell-1} - \hat{x}_{\ell-1} \tag{3}$$

$$\hat{y} = \sum_\ell \hat{y}_\ell \tag{4}$$

where x_ℓ is the input for the next block. The $x_{\ell-1}$ is input for the current block, $\hat{x}_{\ell-1}$ is the backcast residual in the current block, and $\hat{y}$ is the final global forecast vector. Last, $\hat{y}_\ell$ is the forecast vector on the current block.

The proposed application of the N-BEATS model for stock price prediction in the Indonesian market represents a significant advancement in financial forecasting. While previous studies have explored various predictive models, including traditional machine learning algorithms and deep learning techniques such as LSTM, adopting N-BEATS introduces a novel approach with several distinctive contributions. Firstly, N-BEATS stands out for its innovative architecture, characterised by backward and forward residual links and a deep stack of fully connected layers. This design enables N-BEATS to more effectively capture complex temporal patterns and dependencies within stock market data than traditional models. By incorporating past and future information through residual links, N-BEATS can leverage a broader range of historical data to inform its predictions, enhancing its forecasting accuracy and robustness. Moreover, N-BEATS offers inherent interpretability, a feature often lacking in complex deep-learning models. The transparent architecture of N-BEATS allows researchers and investors to understand how predictions are generated, facilitating insights into the underlying factors driving stock price movements. This interpretability is crucial for building trust in the predictive model and aiding decision-making processes, particularly in high-stakes financial environments.

Additionally, N-BEATS demonstrates seamless applicability across diverse target domains without requiring extensive modifications. This versatility is particularly advantageous in stock price prediction, where market dynamics can vary widely across different companies, sectors, and regions. By leveraging N-BEATS adaptability, researchers can develop a unified framework for forecasting stock prices across various Indonesian companies, providing investors with a comprehensive and consistent predictive toolset. Furthermore, the fast training capabilities of N-BEATS enable efficient model deployment and iteration, facilitating real-time decision-making in dynamic market environments. The rapid training process reduces computational overhead and accelerates the implementation of predictive models into investment strategies, enhancing their practical utility for investors seeking timely insights into stock market trends.

Overall, the proposed application of the N-BEATS model for stock price prediction in the Indonesian market represents a novel contribution to financial forecasting. By leveraging N-BEATS innovative architecture, interpretability, versatility, and efficiency, the proposed work aims to provide investors with a powerful tool for making informed investment decisions and confidently navigating the complexities of the Indonesian stock market.

Long Short-Term Memory Architecture

LSTM is a type of Recurrent Neural Network (RNN), which is also a strong artificial neural network, so it is suitable for solving problems from time series data (Yu et al., 2019). LSTM networks can store previous information for long periods using unique memory cells. LSTM gates solve short-term memory problems and can also be used to organise data flows (Prakash & Kumar, 2023). The gateway can store necessary data sequentially while discarding useless information. The architecture of LSTM can be seen in Figure 3. The LSTM architecture consists of a cell state and three gates: input, forget, and output. Cell state functions channel information from one sequence to another, with changes to the value being passed through 3 gates. The forget gate functions to determine the current input and the previous hidden state were stored or discarded with an output in the form of a forget gate value (f_t). The input gate functions to update the value of the cell state by calculating candidate cell states ($\hat{C}_t$) and input gate (i_t). And the output gate functions to calculate the value that becomes the next hidden state. The hidden state captures the network's understanding of the data (Ehteram, 2024). Equations 5, 6, 7, 8, 9 and 10 are the formulas for input gate, forget gate and output gate.

$$i_t = \sigma\left(w_f * [h_{t-1}, x_t] + b_i\right) \tag{5}$$

$$f_t = \sigma\left(w_i * [h_{t-1}, x_t] + b_f\right) \tag{6}$$

$$g_t = tanh(w_c * [h_{t-1}, x_t] + b_c) \tag{7}$$

$$C_t = f_t * C_{t-1} + i_t * g_t \tag{8}$$

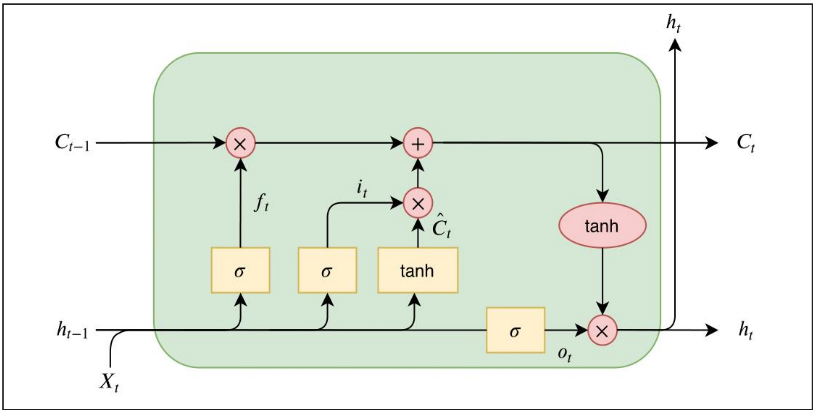
$$O_t = \sigma(w_o * [h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = O_t * tanh(C_t) \tag{10}$$

where i_t is the gate input, f_t is forget gate, O_t is the gate output, σ is the sigmoid function, $tanh$ is the tanh function, w is the weight, x_t is the current input, b is bias, g_t is a candidate for a new cell state, C_t is the new cell state, h_{t-1} is the hidden state of the previous LSTM block and h_t is the new hidden state.

Figure 3

LSTM Architecture

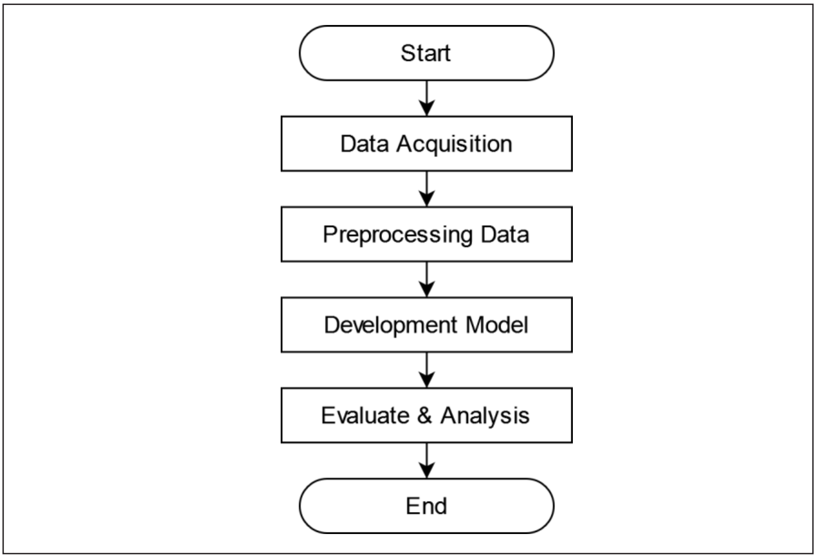


METHODOLOGY

This research employs an experimental study strategy that tests the model for stock price prediction and evaluates the utilised model. The process involves designing the steps carried out throughout the research. The research strategy flowchart can be observed in Figure 4.

Figure 4

Flowchart of Steps Performed in the Experiment



Data Acquisition

The initial step involves obtaining the dataset from the Yahoo Finance website. The dataset used was PT Bank Central Asia Tbk share price data from 25 March 2013 to 21 March 2023 with 2487 rows and six columns of data obtained: open, high, close, adjusted close and volume. This research focused on 'close' price data, which were used in data preprocessing by conducting min-max normalisation.

Preprocessing Data

The N-BEATS model requires no data preprocessing or specific feature engineering (Bulatov, 2020). Therefore, min-max normalisation is only used in the LSTM model. The dataset containing stock prices divided the training and testing data by comparing 80 percent of training data and 20 percent of testing data. Then, each data goes to a function called create_sequence, which converts the data into several sequences. The complete preprocessing flow diagram can be seen in Figures 5 and 6.

Figure 5

Flowchart of Preprocessing Data

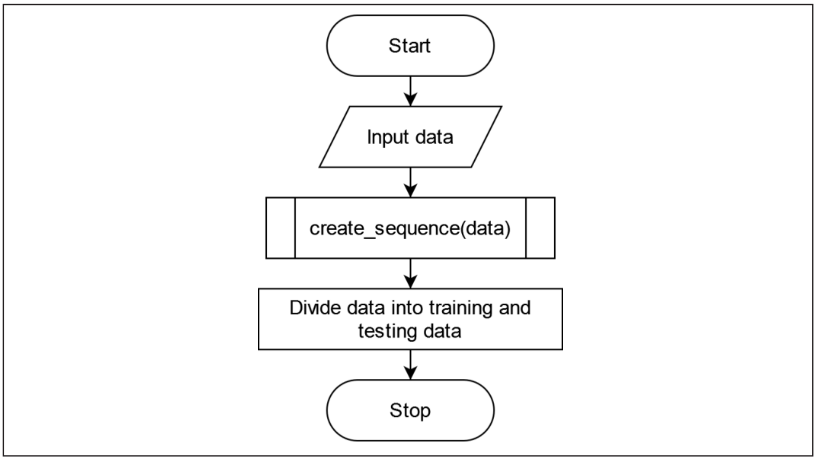
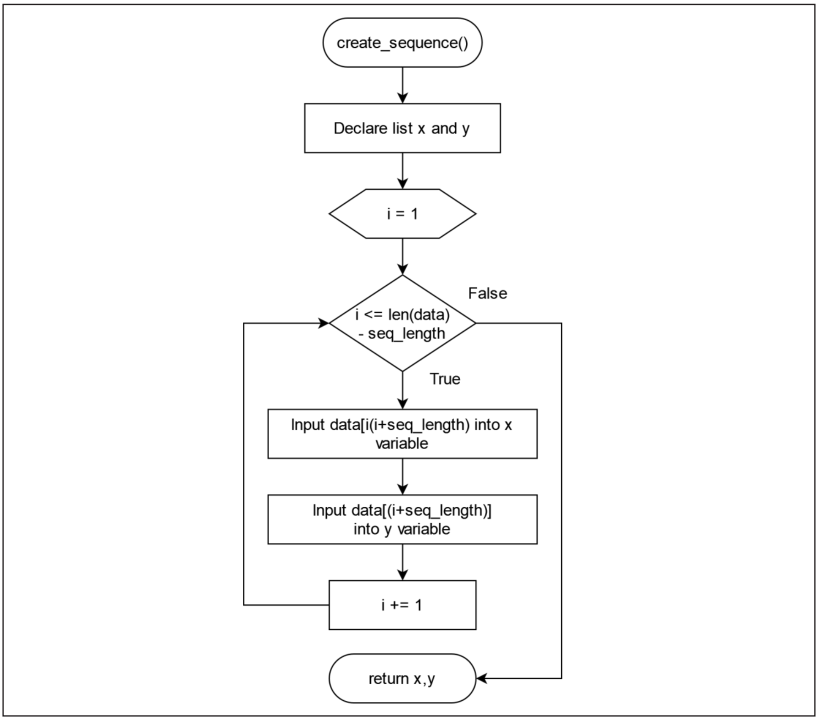


Figure 6

Flowchart of Create_Sequence Function



Min-Max Normalization

Normalisation is a way to change the form of data into another form without losing the information contained therein. One normalisation formula that can be used is min-max. The min-max formula adjusts data in the range 0 to 1. Min-max normalisation minimises errors when testing prediction models (Rizkilloh & Widiyanesti, 2022). The calculation process was also faster because the data value would be smaller. The Min-max normalisation formula can be seen in Equation 11.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{11}$$

where, X' is result data of min-max normalisation, X is data before normalisation, X_{max} and X_{min} are the largest and smallest data from all data.

Denormalisation

Denormalisation is the opposite of normalisation, where the value of the data that changes due to the normalisation process is returned to its original value. The formula for denormalisation from min-max normalisation is shown in Equation 12.

$$X = X'(X_{max} - X_{min}) + X_{min} \tag{12}$$

where, X' is result data of min-max normalisation, X is data before normalisation, X_{max} and X_{min} is the largest and smallest data from all data.

Development Model

The network architecture used is the N-BEATS model. The first step of this architecture is to enter data into a function, namely NBeats, which produces output in the form of prediction results. The flow of the NBeats function is called nested loops and conditional statements. In the conditional statement, a condition is met. That is, if the block and stack are equal to one, the data received from the NBeats function were thrown to the NBeatsBlock function, producing output in the form of backcast and forecast vector variables. Then, the data received by the NBeats function were reduced to a forecast vector, which becomes a variable called residual. However, if these conditions are not met, a process occurs in the form of entering residual variables into the NBeatsBlock function, which will produce output in the form of backcast and block_forecast vector variables. The forecast variable is added to the block_forecast variable, which becomes a variable called forecast. The backcast variable reduces the residual variable, which becomes a variable called residual. The conditional statement process is carried out repeatedly as many times as there are blocks. This loop is also carried out repeatedly, as many times as there are stacks. In the NBeatsBlock function, the forward propagation calculation of the N-BEATS model occurs in the order of Linear 1, ReLU, Linear 2, ReLU, Linear 3, ReLU, Linear 4, ReLU, and Linear 5. Linear 5 is divided into two parts, namely, the backcast variable and the forecast. The complete network architecture of the N-BEATS model used is shown as a flow diagram in Figures 7, 8, and 9.

Figure 7

Flowchart of Network Architecture

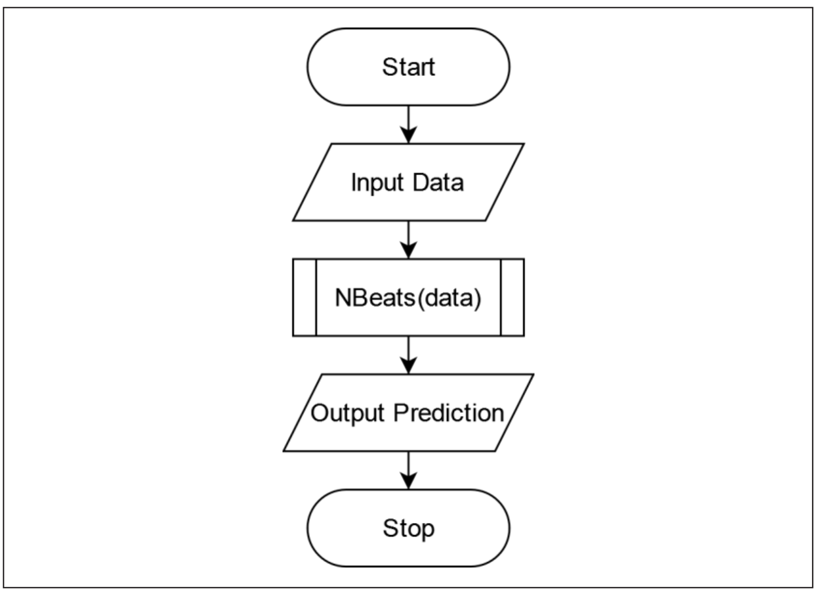


Figure 8

Flowchart of NBeats Function

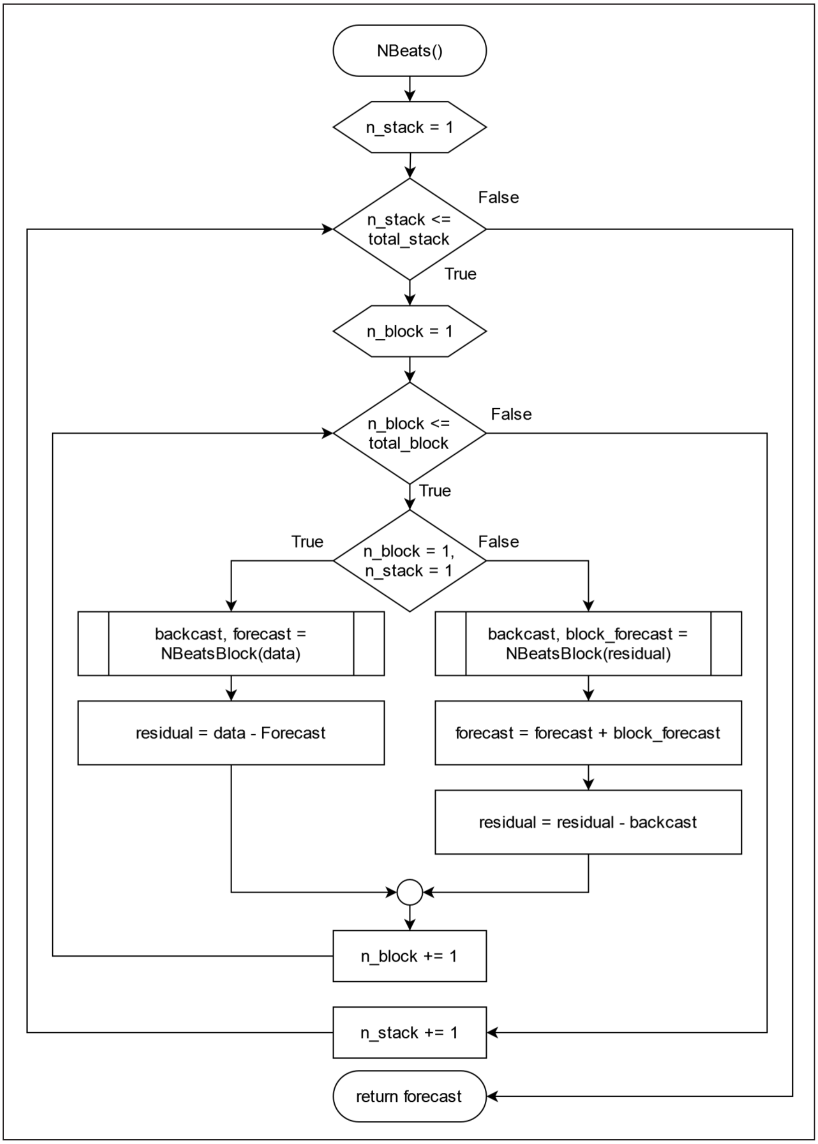
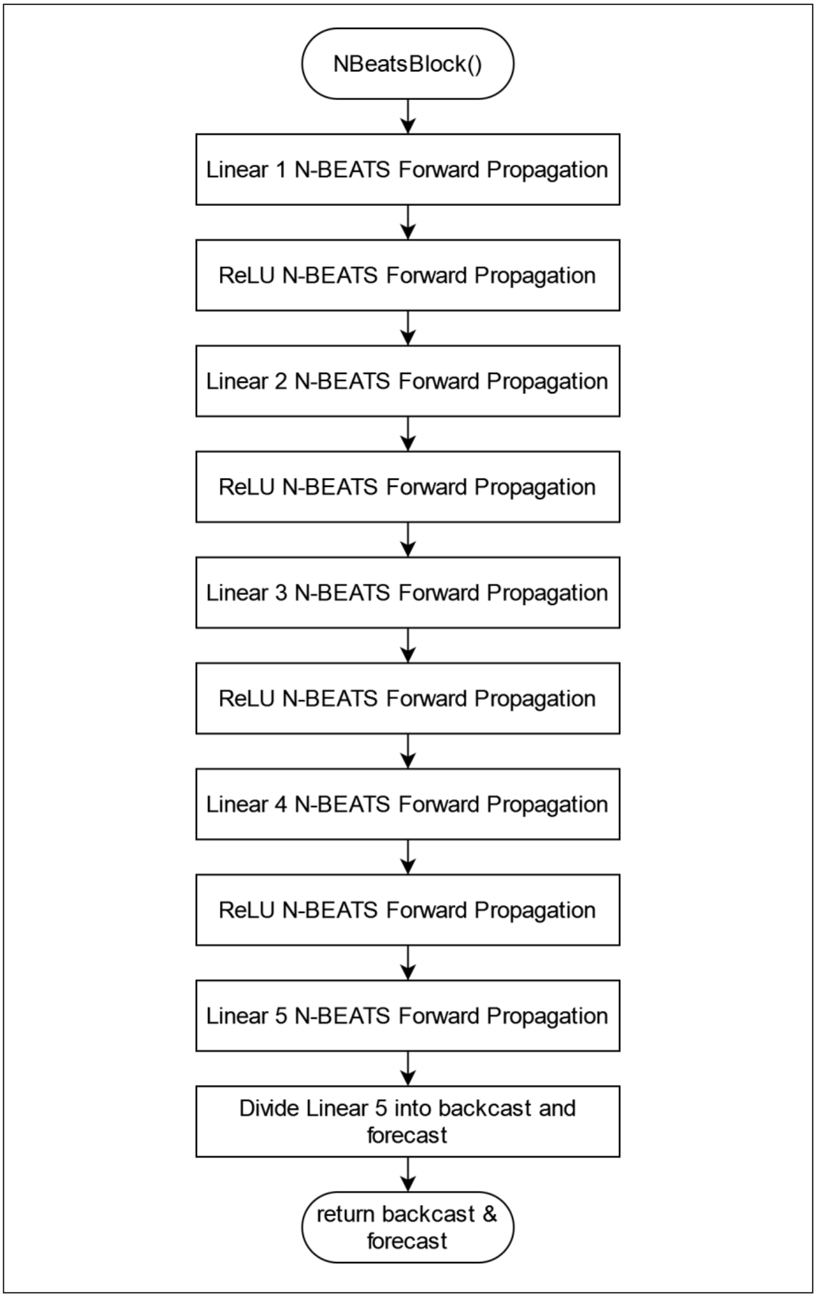


Figure 9

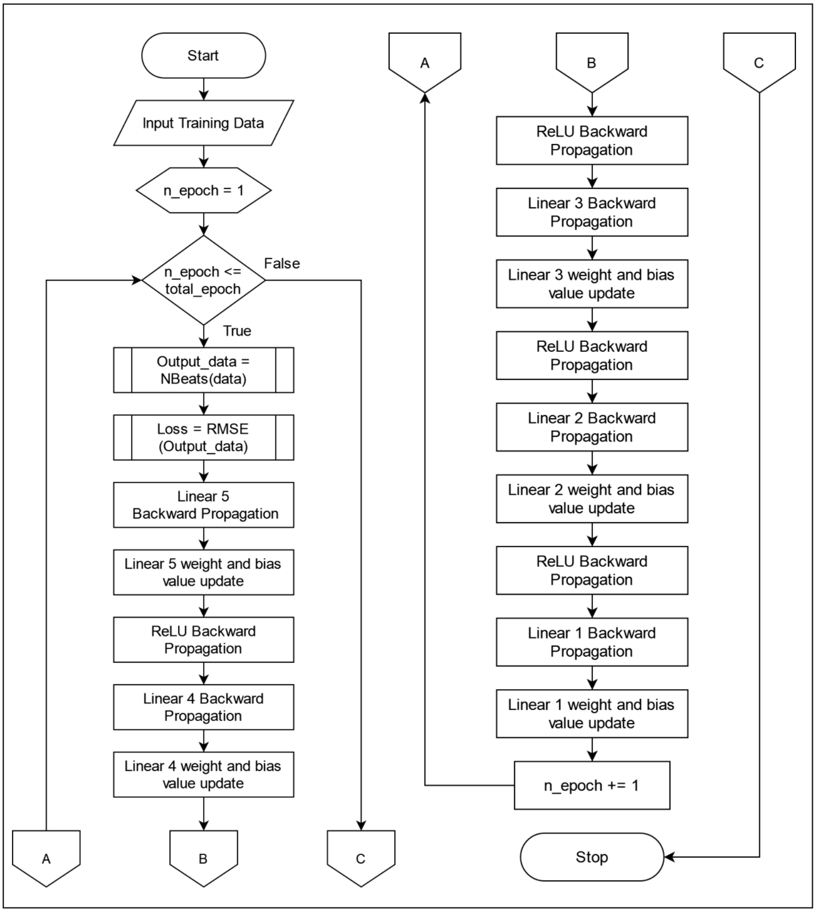
Flowchart of NBeatsBlock Function



The N-BEATS model training process is carried out to obtain optimal weight and bias values, which will later be used in the testing stage. The training flow diagram for the model can be seen in Figure 10.

Figure 10

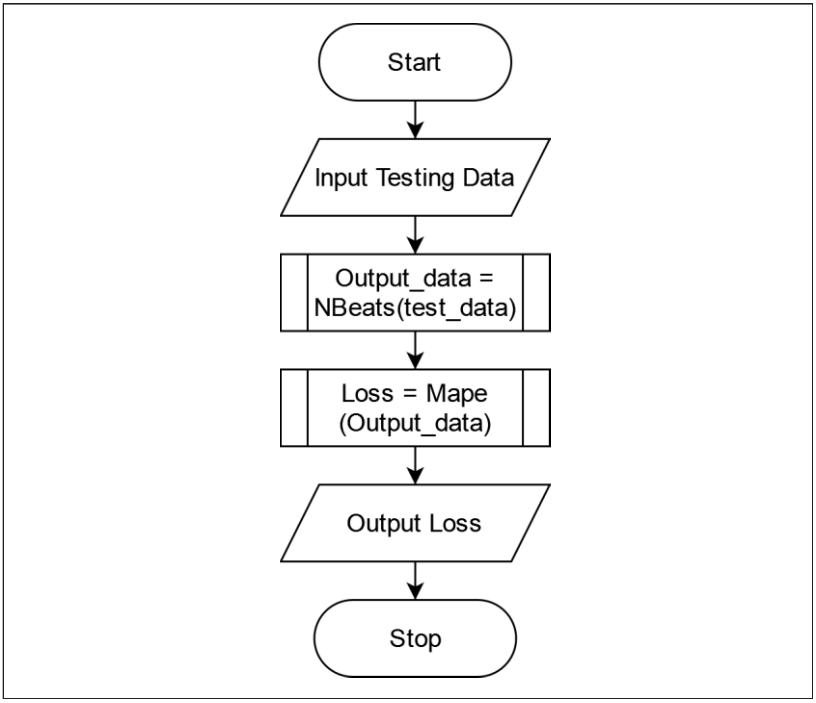
Flowchart of N-BEATS Training Model



The model testing process is carried out after the training, and the final weights and biases are obtained. Testing was carried out using the stock price test dataset to determine the model's performance. The flow diagram of testing the model can be seen in Figure 11.

Figure 11

Flowchart of Testing Model



Evaluation and Analysis

A metric is required to measure a prediction model's adequacy, such as Root Mean Squared Error (RMSE) and MAPE. RMSE is the square root of the average of the squared differences between estimated values and actual values. RMSE is scale-dependent, so it cannot be used to compare two different datasets. However, it is useful for indicating the magnitude of errors, whether positive or negative, and it is always greater than or equal to zero (though achieving zero is practically impossible) (Yadav & Thakkar, 2024). MAPE is the absolute value of the percentage error of data to the mean. MAPE indicates the magnitude of prediction errors compared to the original values (Laung-Iem & Thanarak, 2021). The resulting prediction criteria are divided into four based on the MAPE value, as seen in Table 1 (Chang et al., 2007). RMSE and MAPE can be calculated according to Equations 13 and 14.

Table 1

Category of MAPE

MAPE	Category
<10%	Excellent Forecasting Ability
10-20%	Good Forecasting Ability
20-50%	Reasonable Forecasting Ability
>50%	Bad Forecasting Ability

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{13}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{d_i}\right| \times 100\% \tag{14}$$

where N is the total test data, y_i is the actual data, and $\hat{y}_i$ is the predicted data.

Using the stock price data of PT Bank Central Asia Tbk from 25 March 2013 to 21 March 2023, the evaluation focused on the N-BEATS model's parameter configurations, including sequence length, block layer, layer size, block, and stack used. In addition, it also evaluated the LSTM model by analysing the model parameter configurations in the form of hidden size, number of layers, and sequence length. The resulting MAPE loss values from these various configurations were compared. Subsequently, a conclusion was drawn regarding the configuration that achieves the lowest MAPE value.

RESULTS AND DISCUSSION

The results of the evaluation based on the previously established design are explained in this section. It was performed using different model configurations, and the optimal results were chosen based on the smallest MAPE value. Each configuration was tested ten times, and the average value was taken as the final result. The same set of hyperparameters was used for each test, including the Adam optimiser, a learning rate of 0.01, and 150 epochs.

N-BEATS Based on Layer Sizes

The model was tested on layer sizes using the following configuration: a sequence length of 5, 3 block layers, 3 blocks, a stack of 10, and a training-to-test data ratio of 80:20. The tested layer sizes were 8, 16, 32, 64, and 128. The results of the average MAPE values are presented in Table 2. The final result of Table 2 can be represented as a line graph, as shown in Figure 12. Based on the tests conducted on the layer size, the highest MAPE value was obtained at a layer size of 8. Subsequently, the MAPE value decreased when the layer size was increased to 16 and 32, which resulted in the lowest MAPE value of 1.24 percent. Then, the MAPE value increased when the layer size was further increased from 32 to 64 and 128. Based on these findings, it can be said that the layer size in the performance of the N-BEATS model does not necessarily follow the rule that larger or smaller sizes are always better. Tests must be conducted using different quantities. Moreover, increasing the layer size will affect the training time due to the increased computational processes, making the process more demanding.

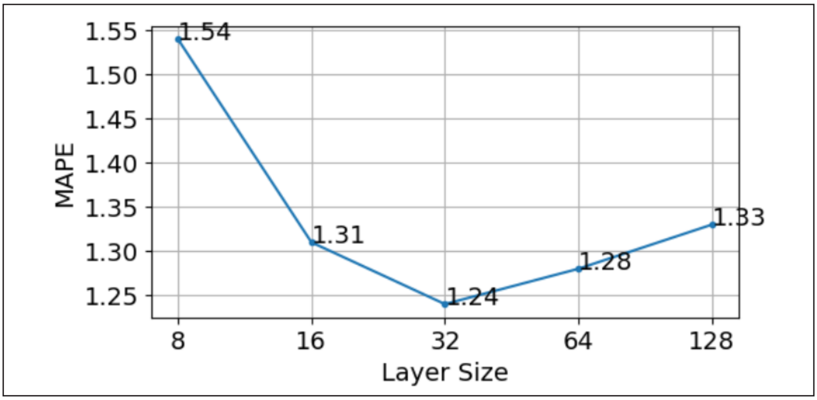
Table 2

Test Results of N-BEATS Based on Layer Size

	Layer Size				
	8	16	32	64	128
MAPE (%)	1.54	1.31	**1.24**	1.28	1.33

Figure 12

Graph of Test Results of N-BEATS Based on Layer Size



N-BEATS Based on the Number of Block Layers

The model was tested on the block layer using the following configuration: a sequence length of 5, a layer size of 32, 3 blocks, 10 stacks, and a training-to-testing data ratio of 80:20. The tested values for the block layer were 1, 2, 3, 4, 5, and 6. The results, represented by the average MAPE values, are presented in Table 3. The final results of Table 3 can be represented as a line graph, as shown in Figure 13. Based on the tests on the number of block layers, the highest MAPE value was obtained with 1 block layer.

Subsequently, the MAPE value continued to increase as the number of block layers was increased to 2 and 3, which resulted in the lowest MAPE value of 1.16 percent. Then, the MAPE value stabilised as the number of block layers increased from 3 to 4, 5, and 6. This indicates that the number of block layers impacts the model's performance. A lower number of block layers leads to a higher MAPE value and vice versa. However, increasing the number of block layers will affect the training time due to the increased computational processes, making the process more demanding.

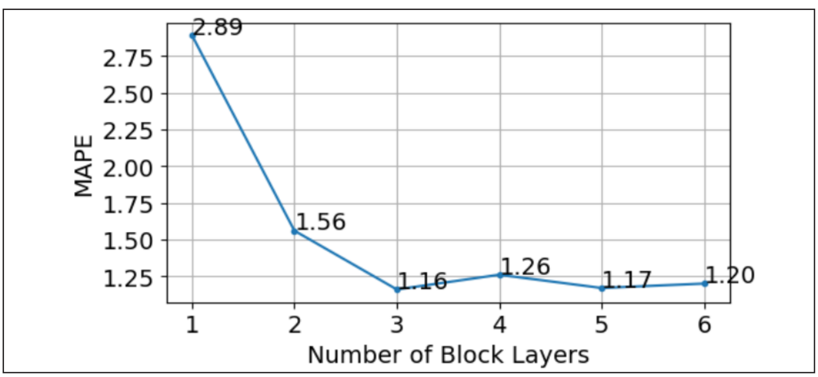
Table 3

Test Results of N-BEATS Based on Number of Block Layers

	Number of Block Layers					
	1	2	3	4	5	6
MAPE (%)	2.89	1.56	**1.16**	1.26	1.17	1.20

Figure 13

Graph of Test Results of N-BEATS Based on Number of Block Layers



N-BEATS Based on Number of Blocks

The model was tested on the number of blocks using the following configuration: a sequence length of 5, a layer size of 32, a block layer of 3, a stack of 10, and a training-to-testing data ratio of 80:20. The tested block values were 1, 2, 3, 4, and 5. The results, represented by the average MAPE values, are shown in Table 4. The final result of Table 4 can be represented as a line graph, as shown in Figure 14. Based on the test conducted on blocks, the highest MAPE value was obtained with a block count of 1. Subsequently, the MAPE value decreases as the block count increases to 2, 3, and 4, where the lowest MAPE value is 1.23 percent. Then, the MAPE value increases when the block count is further increased from 4 to 5. This indicates that the optimal block count is 4. Based on these findings, it can be stated that the performance of the N-BEATS model does not necessarily improve as the block count increases or decreases. Further testing is necessary using different block counts.

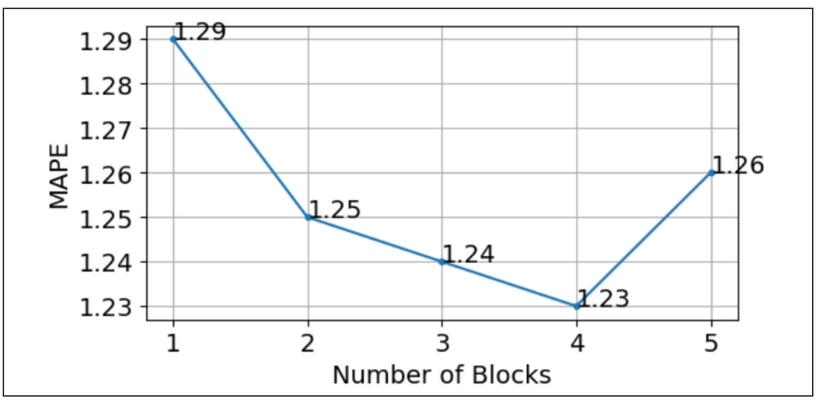
Table 4

Test Results of N-BEATS Based on Number of Blocks

	Number of Blocks				
	1	2	3	4	5
MAPE (%)	1.29	1.25	1.24	**1.23**	1.26

Figure 14

Graph of Test Results of N-BEATS Based on Number of Blocks



N-BEATS Based on Number of Stacks

The model was tested on various stack sizes using the following configuration: a sequence length of 5, a layer size of 32, a block layer of 3, a block size of 4, and a training-to-testing data ratio of 80:20. The tested stack sizes were 1, 5, 10, 15, 20, and 25. The results of the average MAPE values are presented in Table 5. The final result of Table 5 can be represented as a line graph, as shown in Figure 15. Based on the conducted tests on the stack size, the MAPE value decreased when the stack size increased from 1 to 5. However, the MAPE value remained unchanged when the stack size was increased from 5 to 10. Subsequently, the MAPE value decreased when the stack size was further increased from 10 to 15, reaching its lowest value of 1.25 percent. Then, the MAPE value increased when the stack size was further increased to 20 and 25, which represented the highest MAPE values. This indicates that the optimal stack size is 15. Based on these findings, it can be concluded that the stack size in the N-BEATS model's performance is not always better with a larger or smaller value. Further testing is necessary using different stack counts.

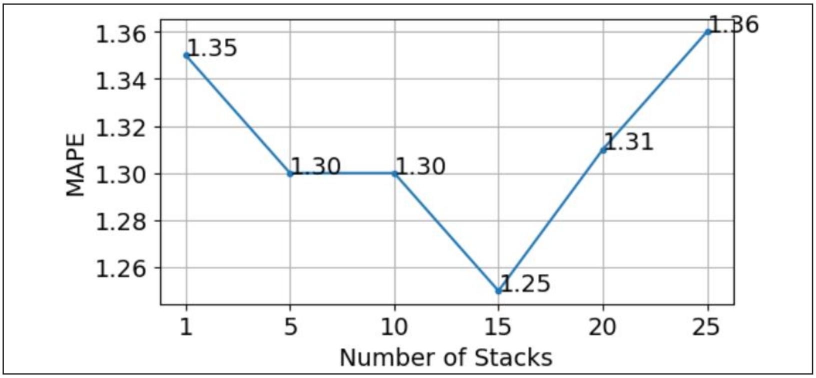
Table 5

Test Results of N-BEATS Based on Number of Stacks

	Number of Stacks					
	1	5	10	15	20	25
MAPE (%)	1.35	1.30	1.30	**1.25**	1.31	1.36

Figure 15

Graph of Test Results of N-BEATS Based on Number of Stacks



N-BEATS Based on Sequence Length

The model was tested on sequence length with the following configuration: a layer size of 32, 3 block layers, 4 blocks, 15 stacks, and a training-to-test data ratio of 80:20. The tested sequence lengths were 1, 3, 5, 7, 10, and 15. The results of the average MAPE values are presented in Table 6. The final results of Table 6 can be represented in the form of a line graph, as shown in Figure 16. Based on the conducted tests on sequence length, the lowest MAPE value was obtained at a sequence length of 1, which is 1.05 percent. Subsequently, the MAPE value increased when the layer size was increased to 3 and 5. Then, the MAPE value decreased when the layer size was further increased from 5 to 7. Then, the MAPE value increased again when the layer size was increased to 10 and 15, which were the highest MAPE values. Based on these results, it can be concluded that the sequence length impacts the model's performance. The higher the sequence length, the higher the MAPE value, and vice versa. A larger sequence length results in a larger data pattern, thereby increasing the MAPE value.

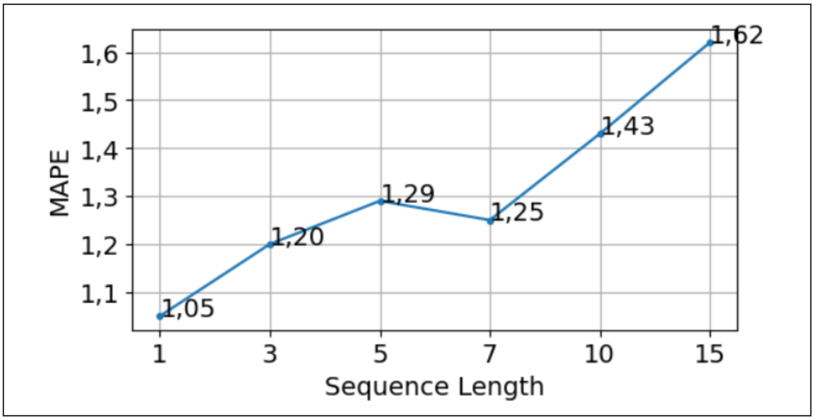
Table 6

Test Results of N-BEATS Based on Sequence Length

	Sequence Length					
	1	3	5	7	10	15
MAPE (%)	**1.05**	1.20	1.29	1.25	1.43	1.62

Figure 16

Graph of Test Results of N-BEATS Based on Sequence Length



LSTM Based on Hidden Size

The model was tested for hidden size with a configuration consisting of a single layer, a sequence length of 3, and a training-to-test data ratio of 80:20. The tested hidden size values were 8, 16, 32, 64, 128, and 256. The results of the average MAPE values are presented in Table 7. The final result of Table 7 can be represented as a line graph, as shown in Figure 17. Based on the testing conducted on the hidden size, the highest MAPE value was obtained at a hidden size of 8. Subsequently, the MAPE value decreased when the hidden size was increased to 16, 32, and 64, which yielded the lowest MAPE value of 1.67 percent. Then, the MAPE value increased when the hidden size was further increased from 64 to 128 and 256. This indicates that the hidden size impacts the model's performance. A smaller hidden size results in a higher MAPE value, whereas a larger one affects training time due to the increased computational processes, making the process more demanding.

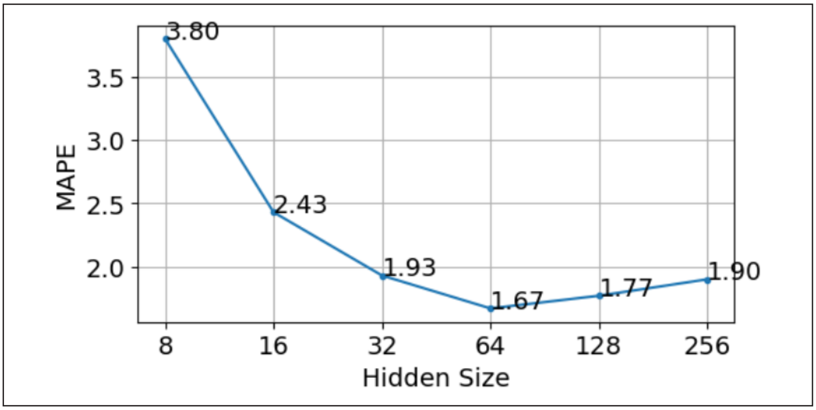
Table 7

Test Results of LSTM Based on Hidden Size

	Hidden Size					
	8	16	32	64	128	256
MAPE (%)	3.80	2.43	1.93	**1.67**	1.77	1.90

Figure 17

Graph of Test Results of LSTM Based on Hidden Size



LSTM Based on the Number of Layers

The model was tested on the number of layers using a configuration consisting of a hidden size of 64, a sequence length of 3, and a training-to-testing data ratio of 80:20. The tested layer values were 1, 2, 3, 4, and 5. The results of the average MAPE values are presented in Table 8. The final result of Table 8 can be represented as a line graph, as shown in Figure 18. Based on the testing conducted on the number of layers, the lowest MAPE value was obtained with 8 layers, which is 1.71 percent. Subsequently, the MAPE value increased when the number of layers was increased to 2 and 3, which were the highest MAPE values. Then, the MAPE value decreased when the number of layers was increased from 3 to 4 and 5. Based on these results, it can be concluded that the number of layers affects the model's performance. The more layers there are, the higher the MAPE value, and vice versa. This is because a larger number of layers leads to longer training time due to the increased computational processes, making the process more demanding.

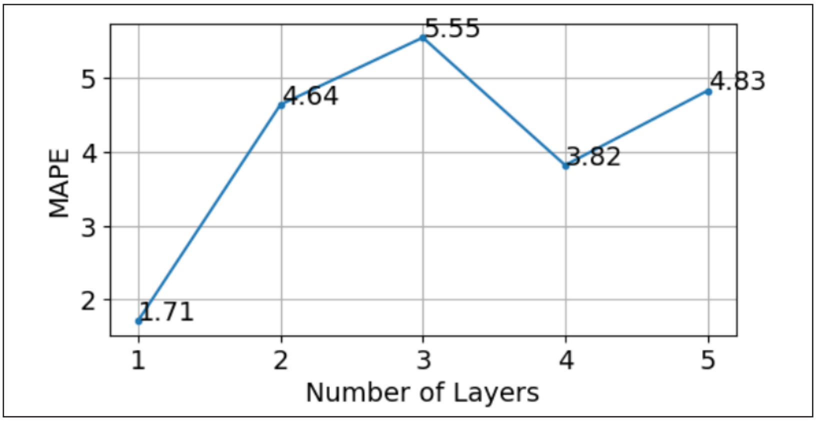
Table 8

Test Results of LSTM Based on Number of Layers

	Number of Layers				
	1	2	3	4	5
MAPE (%)	**1.71**	4.64	5.55	3.82	4.83

Figure 18

Graph of Test Results of LSTM Based on Number of Layers



LSTM Based on Sequence Length

The model was tested for sequence length using a configuration consisting of a hidden size of 64, a single layer, and a training-to-test data ratio of 80:20. The tested sequence lengths were 1, 3, 5, 7, 10, and 15. The results of the average MAPE values are presented in Table 9. The final result of Table 9 can be represented in the form of a line graph, as shown in Figure 19. Based on the conducted tests on sequence length, the highest MAPE value was obtained at a sequence length of 1. Subsequently, the MAPE value decreased when the sequence length was increased to 3, which resulted in the lowest MAPE value of 1.69 percent. Then, the MAPE value increased when the layer size was increased from 3 to 5, 7, 10, and 15. This indicates that the optimal sequence length is 3. Based on these findings, it can be concluded that the performance of the LSTM model does not necessarily improve with larger or smaller sequence lengths. Further testing with different values is required.

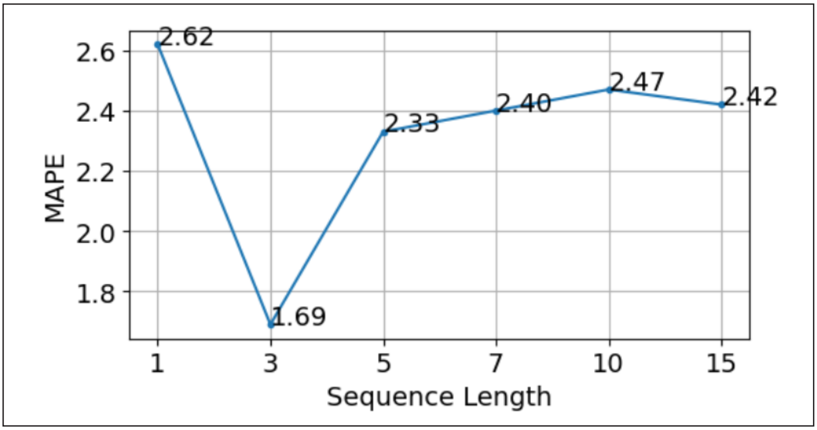
Table 9

Test Results of LSTM Based on Sequence Length

	Sequence Length					
	1	3	5	7	10	15
MAPE (%)	2.62	**1.69**	2.33	2.40	2.47	2.42

Figure 19

Graph of Test Results of LSTM Based on Sequence Length

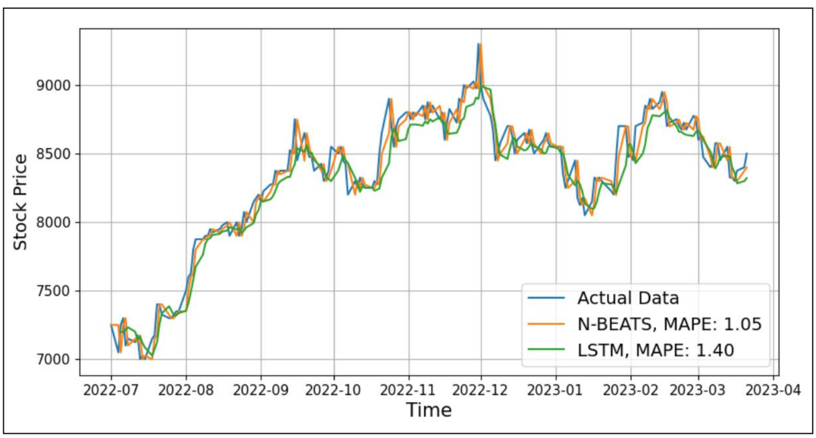


Comparison of Prediction for N-BEATS and LSTM

After conducting tests on the N-BEATS and LSTM models, a comparison was made between the prediction results of both methods to determine whether N-BEATS outperforms LSTM using its best-performing parameters obtained from the previous testing. In Figure 20, the data patterns generated by both methods tend to be similar, but the forecasting results from the N-BEATS method are closer to the actual data values compared to the LSTM method, and the evaluation scores of both methods indicate that the N-BEATS method with MAPE value of 1.05 percent outperforms LSTM with MAPE value of 1.40 percent. Based on the data patterns and evaluation scores of both methods in this study, it can be concluded that the N-BEATS method is superior to LSTM. This is due to the N-BEATS method's advantage in interpretation or interpretability, which refers to its ability to understand how the model makes predictions. The model can clearly understand the factors that influence its predictions. This method aims to uncover insights into the patterns and signals embedded in the time-series data.

Figure 20

Comparison of Prediction for N-BEATS and LSTM



Comparison of Prediction for N-BEATS and Other Models

Table 10

Comparison of N-BEAST with Other Models

	Models			
	N-BEAST	LSTM	RNN	ARIMA
MAPE (%)	**1.05**	1.40	1.67	1.07

The comparison of predictive models (Table 10), including N-BEATS, LSTM, RNN, and ARIMA, based on MAPE values, reveals insightful distinctions in their forecasting performance. N-BEATS exhibits the lowest MAPE percentage of 1.05 percent, indicating its superior accuracy in predicting stock price movements compared to the other models. LSTM follows closely with a MAPE of 1.40 percent, demonstrating its efficacy in capturing temporal dependencies within the data. Meanwhile, RNN and ARIMA show slightly higher MAPE values of 1.67 percent and 1.07 percent, respectively. Despite their relatively lower accuracy than N-BEATS and LSTM, these models still offer competitive performance in forecasting stock prices. However, the significantly lower MAPE of N-BEATS underscores its potential as a preferred choice for accurate and reliable stock price predictions, highlighting its unique contribution to financial forecasting. Furthermore, the capabilities possessed by N-BEATS, such as seamless applicability across diverse target domains without requiring modifications and fast training, also strengthen the reasons for choosing N-BEATS as a model for predicting stock prices. These results have significant implications for financial analysts and investors who rely on accurate stock price forecasts to inform their decisions. By leveraging the superior performance of N-BEATS, stakeholders can achieve more accurate predictions, potentially leading to better investment strategies and higher returns. Additionally, the efficiency and versatility of N-BEATS make it a valuable tool for various forecasting scenarios, enhancing its appeal in the financial sector and beyond. These findings also encourage further research and development in improving forecasting models to achieve even greater accuracy and reliability.

CONCLUSION

Based on the research findings of stock price prediction using the N-BEATS method, it can be concluded that the algorithm design using the N-BEATS method to predict stock prices requires several stages. The data used in this study consists of the stock prices of PT Bank Central Asia Tbk. The algorithm was tested using five scenarios to determine the optimal parameters. The testing scenarios included layer size, block layer, block, stack, and sequence length. The results of these tests indicate that the N-BEATS method performs optimally when the layer size is set to 32, the number of block layers is 3, the number of blocks is 4, the number of stacks is 15, and the sequence length is 1. The results of the N-BEATS method evaluation can be observed in the achieved MAPE value using the optimal parameters. The obtained MAPE value stands at 1.05 percent. This result is better than that of other deep learning models, such as the LSTM method, which is equal to 1.4 percent, and the RNN with 1.67 percent.

Several limitations were identified in this study. Firstly, this method has not been evaluated in combination with other architectures, such as DeepAR, XGBoost, or LGBM, which may potentially enhance the model's accuracy and robustness. Secondly, the testing was confined to specific hyperparameter combinations, indicating that future research should include a broader or more focused range of hyperparameters to explore the model's optimal performance. Thirdly, the dataset employed in this study was limited in both size and time range. Using a larger and more varied dataset, such as extending the time range or incorporating higher time resolution data (e.g., hourly or minute data), could provide deeper insights and contribute to a more robust model. Addressing these limitations in future research could yield more comprehensive and accurate results in stock price prediction.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sector.

REFERENCES

Bhanusree, Y., Kumar, S. S., & Rao, A. K. (2023). Time-distributed attention-layered convolution Neural Network with ensemble learning using Random Forest classifier for speech emotion recognition. *Journal of Information and Communication Technology, 22*(1), 49-76. https://doi.org/10.32890/jict2023.22.1.3

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. New York: John Wiley & Sons.

Bulatov, A. (2020). Forecasting Bitcoin Prices Using N-BEATS Deep Learning Architecture. Sheridan College, Institute of Technology and Advanced Learning.

Bursa Efek Indonesia. (2023). Retrieved 02 15, 2023, from https://idx.co.id/id/produk/saham/

Chang, P.-C., Wang, Y.-W., & Liu, C.-H. (2007). The development of a weighted evolving fuzzy neural network for PCB sales forecasting. *Expert Systems with Applications, 32*(1), 86-96. https://doi.org/10.1016/j.eswa.2005.11.021

Draper, N. R., & Smith, H. (1998). Applied regression analysis (Vol. 326). New York: John Wiley & Sons.

Ehteram, M. a. (2024). Read-First LSTM model: A new variant of long short term memory neural network for predicting solar radiation data. *Energy Conversion and Management,* 305, 118267. https://doi.org/10.1016/j.enconman.2024.118267

Kabir, M. K., Ghosh, K. K., Islam, M. F., & Uddin, J. (2024). Wildfire Prediction in the United States Using Time Series Forecasting Models. *Annals of Emerging Technologies in Computing (AETiC), 8*(2), 32-42. https://doi.org/10.33166/AETiC.2024.02.003

Keren, M., & Oetama, R. S. (2019). Prediksi prospek harga saham perusahaan perbankan Menggunakan Regresi Linear (Study Kasus Bank BCA Tahun 2015-2017). *JSI: Jurnal Sistem Informasi (E-Journal), 11*(1). https://doi.org/10.18495/jsi.v11i1.8105

Laung-Iem, K., & Thanarak, P. (2021). Forecasting of biodiesel prices in Thailand using time series decomposition method for long term from 2017 to 2036. *International Journal of Energy Economics and Policy, 11*(4), 593-600. https://doi.org/10.32479/ijeep.10666

Makridakis, S., Spiliotis, E., & Vassilios Assimakopoulos. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting, 36*(1), 54-74. https://doi.org/10.1016/j.ijforecast.2019.04.014

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437. https://doi.org/10.48550/arXiv.1905.10437

Prakash, V., & Kumar, D. (2023). A Modified Gated Recurrent Unit Approach for Epileptic Electroencephalography Classification. *Journal of Information and Communication Technology, 22*(4), 587-617. https://doi.org/10.32890/jict2023.22.4.3

Rizkilloh, M. F., & Widiyanesti, S. (2022). Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (LSTM). *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 6*(1), 25-31. https://doi.org/10.29207/resti.v6i1.3630

Rudianto. (2012). Pengantar akuntansi: Konsep dan teknik penyusunan laporan keuangan. (S. Saat, Ed.) Jakarta: Erlangga.

Setiawan, F. R., Umbara, R. F., & Rohmawati, A. A. (2018). Prediksi pergerakan harga saham dengan metode Support Vector Machine (SVM) menggunakan trend deterministic data preparation (studi kasus saham perusahaan PT Astra International Tbk, PT Garuda Indonesia Tbk, dan PT Indosat Tbk). eProceedings of Engineering, 5(3), 8356-8372.

Smyl, S., Ranganathan, J., & Pasqua, A. (2018). Uber. Retrieved 01 29, 2023, from https://www.uber.com/en-ID/blog/m4-forecasting-competition/

Tiyara, U. (2021). Prediksi Harga Saham Bank Bri Dan Bca Dengan Long Short Term Memory (Studi Kasus: Close Price Saham Bank Bri (Bbri) Dan Bank Bca (Bbca) Pada Februari 2016--Februari 2021).

Yadav, H., & Thakkar, A. (2024). NOA-LSTM: An efficient LSTM cell architecture for time series forecasting. *Expert Systems with Applications,* 238, 122333. https://doi.org/10.1016/j.eswa.2023.122333

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation, 31*(7), 1235-1270. https://doi.org/10.1162/neco_a_01199