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

International Journal of Quantitative Research and Modeling

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

Vol. 6, No. 1, pp. 33-41, 2025

Comparison of Stock Price Forecasting with ARIMA and Backpropagation Neural Network (Case Study: Telkom Indonesia)

Katherine Liora Carissa[1*], Betty Subartini[2], Sukono[3]

[1,2,3]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang, Indonesia*

**Corresponding author email: katherine21001@mail.unpad.ac.id*

Abstract

The growth of capital market investors in Indonesia is increasing every year. The most popular investment instrument is stocks. One of the stocks on the Indonesia Stock Exchange (IDX) is the Telkom Indonesia (TLKM). Through stock investment, investors can make a profit by utilizing stock prices in the market. However, stock price fluctuations are uncertain. Therefore, modeling is needed to be able to predict stock prices more accurately. The purpose of this study was to find an appropriate time series model and Neural Network model architecture, and to measure the accuracy of the two models in predicting future stock prices of TLKM. The study was conducted using the Autoregressive Integrated Moving Average (ARIMA) model and Backpropagation Neural Network (BPNN). For comparison, the Mean Absolute Percentage Error (MAPE) method was used. The data used in both models were the stock prices of Telkom Indonesia (TLKM) from September 1, 2023 to September 30, 2024. The result shows that the best ARIMA model, selected based on the least Akaike Information Criterion (AIC) value, is ARIMA(0,1,3) with a MAPE value of 1.20%. Meanwhile, the best BPNN model selected from the smallest testing Mean Squared Error (MSE) value, is BPNN(1,3,1) with a MAPE value of 1.17%. Among those two models, the BPNN model is more accurate because it has less MAPE value compared to the ARIMA one. The results of this research can be considered in forecasting TLKM stock price in the future.

Keywords: Stocks; ARIMA; Backpropagation; Neural Network; MAPE

1. Introduction

Forecasting is a process to predict future events by analyzing data sequences that are chronologically indexed (Taslim and Murwantara, 2024; Wei et al., 2019). This process plays an important role in the finance industry, particularly in investment. The data used in this process is called time series data. In Indonesia, the number of capital market investors is growing reaching 13 million single investor identification (SID) in June 2024. Of the 13 million SID, 5.7 million of them are stock investors.

Through stock investment, investors can gain profit by leveraging stock price changes in the market. However, stock price fluctuations are uncertain. Therefore, modelling is required in order to make more accurate stock price predictions. One of the time series models often used in forecasting is the Autoregressive Integrated Moving Average (ARIMA) model. In addition, there are various developments of Artificial Intelligence (AI) models that can be used as well, such as Backpropagation Neural Network (BPNN).

The comparison of forecasting using ARIMA and BPNN has been conducted before with various objects. For example, research done by Sukono et al., (2019) using Gross Regional Domestic Product (GRDP) of Bandung Regency, shows that the Mean Absolute Percentage Error (MAPE) value for BPNN model is less than the ARIMA model and therefore more accurate. Another research by Kittichotsatsawat et al. (2023) shows that the BPNN model appeared to perform better than ARIMA in forecasting Arabica coffee yields. However, the application of these models are rarely used in the Indonesian stock market. This study contributes by comparing the ARIMA and BPNN models to a stock listed on the Indonesia Stock Exchange (IDX), namely Telkom Indonesia (TLKM).

The purpose of this study is to determine the appropriate time series model and Neural Network model architecture, and to compare the accuracy of the two models in predicting TLKM stock price in the future.

2. Literature Review

2.1. Autoregressive Integrated Moving Average (ARIMA)

ARMA model combines the Autoregressive (AR) and Moving Average (MA) model. Autoregressive model is a time series model which assumes that the current time series data is influenced by the time series data value in the past. On the other hand, Moving Average model is a time series model with the assumption that the current time series data depends on the residual value of the data in the previous period. According to Wei (2006), the equation of the ARMA model, where p represents the order of the AR model and q represents the order of the MA model, and can be denoted as ARMA(p, q), is as follows

$$z_t = \mu + \phi_1 z_{t-1} + \cdots + \phi_p z_{t-p} - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (1)$$

with z_t random variable stock price at time t; $\phi_1, \dots, \phi_p$ AR model parameters; $\theta_1, \dots, \theta_q$ MA model parameters, and ε_t value of the residual at time t.

ARIMA model combines the AR model, differencing process, and MA model. Differencing process is done when the time series data is not stationary. According to Wei (2006), the equation of the ARIMA model, where p represents the order of the AR model, q represents the order of the MA model, and d represents the order of differentiation, and can be denoted as ARIMA(p, d, q), is as follows:

$$\phi_p(B)(1-B)^d z_t = \mu + \theta_q(B)\varepsilon_t \quad (2)$$

where $\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ AR operator; $\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$ MA operator; B backshift operator, and $(1 - B)$ differencing operator.

2.2. Artificial Neural Network (ANN)

ANN is a computational model that aims to imitate how the human brain functions (Islam et al., 2019). This model consists of several layers, each layer processes information and passes it to the next layer. A relationship known as weight is present in the process (Mustafidah and Suwarsito, 2020).

Activation function is a mathematic function applied to input signal to produce the output of a neuron (Works, 2021). One of the most commonly used functions is the binary sigmoid function because its value is in between $[0,1]$ and it is easily derived. According to Sukono et al. (2019), the equation of binary sigmoid function is as follows:

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

The first derivative of this function is as follows:

$$f'(x) = f(x)[1 - f(x)] \quad (4)$$

2.2.1. Backpropagation Neural Network (BPNN)

BPNN is a supervised learning approach method used to train ANN (Singh et al., 2022). Backpropagation consists of several layers, including input layer, hidden layer, and output layer. This method can be denoted as BPNN(a, b, c) with a the number of input layer units, b the number of hidden layer units, and c the number of output layer units.

Backpropagation network training consists of three phases, namely the feedforward phase for input training patterns, the error value calculation and backpropagation phase, and the weight adjustment phase (Fausett, 1994). In the first phase, the input signal x_i is calculated forward using the activation function to produce the network output y_k. The output is then compared with the expected target value. In the second phase, the difference between the network output value and the expected target value (error) is calculated using the Mean Squared Error (MSE) method and propagated backward starting from the path directly related to the units in the output layer. In the last phase, the weights in all layers are adjusted simultaneously.

2.2.2. Levenberg-Marquardt Method

Levenberg-Marquardt method is an algorithm used to train Backpropagation Neural Network. This method is able to converge faster than other optimization methods. According to Sukono et al., (2019) and Fausett (1994), the Levenberg-Marquardt Backpropagation algorithm is as follows:

Step 1: Initialize weights and biases with small random values along with maximum epoch and MSE target.

Step 2: Determine the required paramaters, such as: (i) Initialization $epoch = 0$, (ii) *Levenberg-Marquardt* parameter (μ), $\mu > 0$, and (iii) Factor beta parameter (β), which is used to be multiplied or divided by μ.

Step 3: Follow the steps below for each pair of training data, if epoch< maximum epoch or MSE > MSE target, do the following steps.

Phase I: Feedforward

Step 4: The input signal x_i is received by each input unit x_i, $i = 1,2, \dots, n$, and then passed to every unit in the hidden layer.

Step 5: Each unit of the hidden layer $z_j, j = 1,2, \dots, p$ adds up the bias v_{0_j} and the weighted input signals v_{ij},

$$z\_in_j = v_{0_j} + \sum_{i=1}^{n} x_i v_{ij}. \tag{5}$$

Apply the activation function in order to get the output signal,

$$z_j = f\left(z\_in_j\right) = \frac{1}{1 + e^{-z\_in_j}}. \tag{6}$$

Then, the signal z_j is sent to every unit in the layer above (output layer).

Step 6: Each unit of the output layer $y_k, k = 1,2, \dots, n$ adds up the bias w_{0_k} and the weighted input signals w_{j_k},

$$y\_in_k = w_{0_k} + \sum_{j=1}^{p} z_j w_{jk}. \tag{7}$$

Apply the activation function in order to get the output signal,

$$y_k = f(y\_in_k) = \frac{1}{1 + e^{-y\_in_k}}. \tag{8}$$

Phase II: Backpropagation

Step 7: Calculate the error,

$$error_k = e_k = t_k - y_k, \tag{9}$$

where t_k is utput target at k. Calculate the Mean Squared Error (MSE) using the formula below:

$$MSE = \frac{\sum_{i=1}^{n} error_k^2}{n}. \tag{10}$$

Then, the error vector can be formed as follows:

$$\boldsymbol{e} = (e_1 \;\; e_2 \;\; \dots e_n)^T. \tag{11}$$

Step 8: Calculate the output delta at each unit at the output layer $y_k, k = 1,2, \dots, m$ as follows:

$$\delta_{2_k} = error_k f'(y\_in_k). \tag{12}$$

Calculate its weight correction term (used to update w_{j_k} later),

$$\phi_{2_{jk}} = \delta_{2_k} z_j; k = 1,2, .. m; j = 0,1, \dots p. \tag{13}$$

Calculate its bias correction term (used to update w_{0_k} later),

$$\beta_{2_k} = \delta_{2_k}; k = 1,2, .. m. \tag{14}$$

Step 9: Each units from hidden layer $(z_j, j = 1,2, \dots, p)$ adds its delta inputs,

$$\delta\_in_j = \sum_{k=1}^{m} \delta_{2_k} w_{jk,} \tag{15}$$

multiplies by the derivative of the activation function to calculate the hidden delta,

$$\delta_{1_j} = \delta\_in_j f'(z\_in_j). \tag{16}$$

Calculate its weight correction term (used to update v_{ij} later),

$$\phi_{1_{ij}} = \delta_{1_j} x_i. \tag{17}$$

Calculate its bias correction term (used to update v_{0j} later),

$$\beta_{1_j} = \delta_{1_j}. \tag{18}$$

Step 10: Forming the Jacobian matrix $\boldsymbol{J}$,

$$\boldsymbol{J} = \left[\phi_{1_{11}} \cdots \phi_{1_{np}} \quad \beta_{1_1} \cdots \beta_{1_p} \quad \phi_{2_{11}} \cdots \phi_{2_{pm}} \quad \beta_{2_1} \cdots \beta_{2_m}\right]^T. \tag{19}$$

Phase III: Update weights and biases

Step 11: Calculate the new weights and biases as follows:

$$w_{new} = w_{old} - [\boldsymbol{J}^T\boldsymbol{J} + \mu\boldsymbol{I}]^{-1}\boldsymbol{J}^T\boldsymbol{e}, \tag{20}$$

with,

$$\boldsymbol{w} = [w_{11} \cdots w_{pm} \quad w_{0_1} \cdots w_{0_m} \quad v_{11} \cdots v_{np} \quad v_{0_1} \cdots v_{0_p}]^T. \tag{21}$$

Step 12: Modify μ

If $MSE_{new} \leq MSE_{old}$, then: (i) $\mu' = \frac{\mu}{\beta}$, (ii) $epoch = epoch + 1$, and (iii) Go back to step 4.

If $MSE_{new} > MSE_{old}$, then: (i) $\mu' = \mu\beta$ and (ii) Go back to step 11.

Step 13: The training process is stopped when *epoch* = *epoch* maksimum or MSE ≤ MSE target.

2.2.3. Normalization and Denormalization

Normalization is done to ensure that the data used is within a specific range of values. Min-max normalization changes the range of data values to be between 0 and 1 based on the smallest and biggest values in the data. The formula for data normalization is as follows:

$$X_n = \frac{(Z - Z_{\min})}{Z_{max} - Z_{min}}. \tag{22}$$

On the other hand, the process of returning data to its original value before the normalization process is called denormalization. Denormalization can be done with following equation:

$$Z = (X_n)(Z_{max} - Z_{min}) + X_{min} \tag{23}$$

with X_n normalized data values; Z actual data values; Z_{max} maximum value of the actual data; and Z_{min} minimum value of the actual data.

3. Materials and Methods

3.1. Materials

This study used the daily closing price of a stock listed on the Indonesia Stock Exchange, namely Telkom Indonesia (TLKM) from September 1, 2023 to September 30, 2024 with a total of 258 observations. The data was taken from www.yahoo.finance.com website.

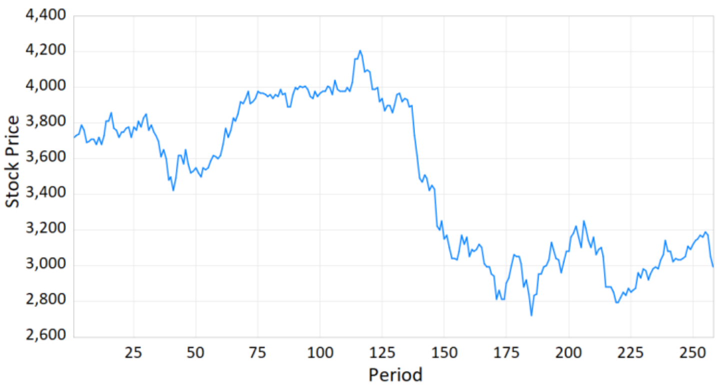

Figure 1: TLKM stock price plot

3.2. Methods

This research was conducted to find the most accurate model to predict TLKM stock price. The models used are the ARIMA and BPNN models. For ARIMA model, the steps are as follows: (i) Identify the order value of p and q by Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF), (ii) Parameter estimation using Maximum Likelihood Estimation (MLE), (iii) Residual diagnostic test for white noise and normally distributed assumptions, and (iv) Forecasting. For BPNN model, the steps are as follows: (i) Parameters, weights, and biases initialization, (ii) Feedforward, (iii) Backpropagation, (iv) Updating weights and biases, and (v) Forecasting using the best BPNN model, decided based on the smallest MSE testing value. Then, compare both model with Mean Absolute Percentage Error (MAPE) to get the more accurate model between ARIMA and BPNN to predict TLKM stock price.

4. Results and Discussion

4.1. Data Analysis Using ARIMA

In this section, the process of determining the best ARIMA model for TLKM stock price is explained.

4.1.1. Stationary Test

This test was done using the Augmented Dickey-Fuller (ADF) test with the help of Eviews 12 software. The hypothesis used was H_0: data is not stationary and H_1; data is stationary. From the test results, a p-value of 0.8210 was obtained at a significance level (α) of 5%. Therefore p-value $> \alpha$ was applied, so H_0 was accepted meaning the data is not stationary. Therefore, differencing process was done starting from the first order. Then, the differenced data was tested again. From the test results, the p-value was 0.0000. Since p-value $< \alpha$, H_1 was accepted meaning the data is already stationary and could proceed to the next step.

4.1.2. Model Identification

The model identification for p and q order can be done using ACF and PACF plots, with the help of Eviews 12 software. While, the d order was determined by the differencing order which is one. Thus, the model used is a model of ARIMA$(p, 1, q)$. Both the ACF and PACF plot decreased exponentially from lag 3. Therefore, the temporary models to be tested further are ARIMA(3,1,0), ARIMA(0,1,3), and ARIMA(3,1,3).

4.1.3. Selection of the Best Model

Choosing the best model can be done using the Akaike Information Criterion (AIC), where the smallest AIC value shows the best model. Based on the results done using Eviews 12 software, the AIC value from each model is as follows: ARIMA(3,1,0) model with AIC value of 10.8220, ARIMA(0,1,3) model with AIC value of 10.8214, and ARIMA(3,1,3) model with AIC value of 10.8383. The best model is ARIMA(0,1,3) because it has the least AIC value.

4.1.4. Parameter Estimation and Significance Test (Test statistics t)

Parameter estimation was done using Eviews 12 software. The estimation results for μ is -2.9321, ϕ_1 is -0.0038, ϕ_2 is 0.0569, and ϕ_3 is 0.1534. However, ϕ_3 parameter was the only parameter that was significant at a significance level of 5%. Therefore, it is necessary to re-estimate the model using only ϕ_3 parameter. The re-estimation result for ϕ_3 is 0.1525. So, the equation of ARIMA(0,1,3) for TLKM stock price is as follows:

$$z_t = -0.1525\varepsilon_{t-3} \tag{24}$$

Against the model equation (24), it is required to test the significance with statistical test t. The test hypothesis used is $H_0: \theta_3 = 0$ (parameter is not significant) and $H_1: \theta_3 \neq 0$ (parameter is significant). From the test results, a p-value of 0,0136 was obtained at a significance level (α) of 5%. Therefore p-value $< \alpha$ was applied, so H_1 was accepted meaning the parameter is significant.

4.1.5. Diagnostic Test

Diagnostic test is needed because ARIMA model residual is assumed to be white noise and normally distributed. White noise test is done using the Ljung-Box test to determine if there is an autocorrelation in the residual. The hypothesis used is H_0: residual is white noise and H_1: residual is not white noise. From the test results, a p-value of 0.6930 was obtained at a significance level (α) of 5%. Therefore p-value $> \alpha$ was applied, so H_0 was accepted meaning the residual is white noise. Normally distributed test can be done using the Jarque-Bera test. However, because the data used has large enough sample sizes (more than 30), the normally distributed test can be skipped (Pallant, 2011).

4.1.6. Forecasting with ARIMA Model

After going through many tests, the best estimator of the ARIMA(0,1,3) model can be used for forecasting TLKM stock price. The comparison plot between actual data and forecasting results of TLKM stock price from September 1, 2023 to September 30, 2024 can be shown as in the graph in Figure 2.

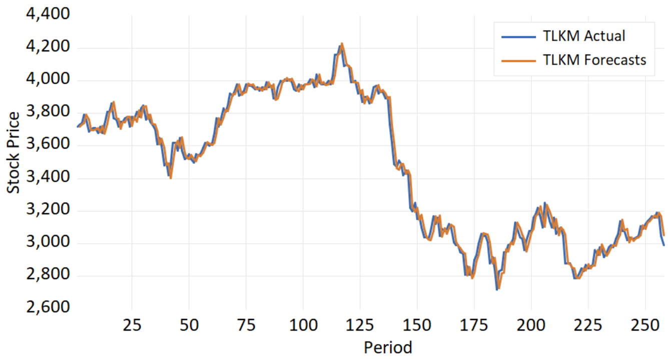

Figure 2: Plot of actual data and forecast results of the ARIMA model for TLKM

In Figure 2, the blue line shows the actual value of the data and the orange line shows the forecast value. It can be seen that the two lines are close together on the comparison chart. This indicates that the ARIMA model used can capture the data movement pattern well. Then, the MAPE calculation was carried out using Eviews 12 software, obtaining a MAPE of 1.20% for TLKM.

4.2. Data Analysis Using BPNN

In this section, the process of determining the best BPNN model for TLKM stock price is explained.

4.2.1. Data Normalization

Before starting the analysis process with BPNN, the data is first normalized using equation (22). This is done since the activation function used is a binary sigmoid function, which has a range value between 0 and 1. After the data is normalized, it was then used for the design of network structure, as follows.

4.2.2. Network Architecture Formation

After the stock price data is normalized, the next stage is the formation of the network architecture, such as the division of input and output variables, the division of training and testing data, determining the number of hidden and output units, and the maximum epoch and MSE target as follows:

a) Divide the normalized data into input and output variables. Determining the lag combination of input variables is done to determine the amount of past data used to predict future values. The lag combination used is obtained through a trial and error process. In this process, testing is carried out starting from small to larger lags. There are six input combinations tested in this research as follows:
    a. Input at the 1st lag (X_{t-1}),
    b. Input at the 1st lag (X_{t-1}) and 2nd lag (X_{t-2}),
    c. Input at the 1st lag (X_{t-1}), 2nd lag (X_{t-2}), and 3rd lag (X_{t-3}),
    d. Input at the 1st lag (X_{t-1}), 2nd lag (X_{t-2}), 3rd lag (X_{t-3}), and 4th lag (X_{t-4}),
    e. Input at the 1st lag (X_{t-1}), 2nd lag (X_{t-2}), 3rd lag (X_{t-3}), 4th lag (X_{t-4}), and 5th lag (X_{t-5}),
    f. Input at the 1st lag (X_{t-1}), 2nd lag (X_{t-2}), 3rd lag (X_{t-3}), 4th lag (X_{t-4}), 5th lag (X_{t-5}), and 6th lag (X_{t-6}).

b) Dividing the data into training and testing data with a ratio of 80:20. Training data is used so that the model can learn data patterns to be evaluated later using testing data.

c) Determining the number of hidden layers and hidden layer units. Hidden layers are used to introduce non-linearity using activation functions. The use of many hidden layers can slow down the training process, so in this study one hidden layer was used. The number of hidden units used was obtained through trial and error. In this process, testing was carried out starting from small hidden units to larger ones. The number of units tested was one to six units.

d) In this study, one output layer unit was used which was the result of forecasting TLKM stock price.

e) Epoch is a parameter for the maximum number of iterations in the training process. One epoch means that all training data is processed once completely. The maximum epoch is determined to set the model training time limit.

f) In this study, a maximum epoch of 1000 was used.

g) In this study, the MSE target used was 0.0001 to ensure a high level of accuracy in predicting stock prices. There are two types of MSE obtained, namely MSE training and MSE testing. The best Backpropagation Neural Network (BPNN) model network architecture is selected based on the smallest MSE testing.

4.2.3. Training and Testing Process

The training process was carried out using 80% of the data to obtain the weights and biases trained using the Levenberg-Marquardt algorithm with a binary sigmoid activation function. Then, the testing process was carried out using the remaining 20% of the data to test the weights and biases that have been obtained. The training and testing process is carried out for each input combination with the number of hidden layer units one to six to obtain the best BPNN model network architecture in predicting TLKM stock price. The MSE resulting from the training and testing process on TLKM stock price data calculated using Python software can be seen in Table 1.

Table 1: MSE training and testing of TLKM stock price

No	BPNN Model	Training MSE	Testing MSE	No	BPNN Model	Training MSE	Testing MSE
1	BPNN(1,1,1)	0.00152355	0.00143671	19	BPNN(4,1,1)	0.00147831	0.00143038
2	BPNN(1,2,1)	0.00133549	0.00104754	20	BPNN(4,2,1)	0.00147785	0.00143112
3	BPNN(1,3,1)	0.00129972	0.00098483	21	BPNN(4,3,1)	0.00128438	0.00111195
4	BPNN(1,4,1)	0.00133552	0.00104782	22	BPNN(4,4,1)	0.00105133	0.00118488
5	BPNN(1,5,1)	0.00132411	0.00099251	23	BPNN(4,5,1)	0.00123039	0.00110114
6	BPNN(1,6,1)	0.00131125	0.00098828	24	BPNN(4,6,1)	0.00129634	0.0014227
7	BPNN(2,1,1)	0.00152909	0.00142959	25	BPNN(5,1,1)	0.00146469	0.00152509
8	BPNN(2,2,1)	0.00134155	0.00104795	26	BPNN(5,2,1)	0.00126038	0.00129685
9	BPNN(2,3,1)	0.00129524	0.00099945	27	BPNN(5,3,1)	0.00149809	0.00136711
10	BPNN(2,4,1)	0.00129764	0.00100697	28	BPNN(5,4,1)	0.00113571	0.00125947
11	BPNN(2,5,1)	0.0013087	0.00098575	29	BPNN(5,5,1)	0.24980255	0.6584132
12	BPNN(2,6,1)	0.00127818	0.00101995	30	BPNN(5,6,1)	0.00176332	0.00148551
13	BPNN(3,1,1)	0.00152943	0.00145445	31	BPNN(6,1,1)	0.00145268	0.00151786
14	BPNN(3,2,1)	0.00152661	0.00144883	32	BPNN(6,2,1)	0.00125287	0.00133264
15	BPNN(3,3,1)	0.00131837	0.00107187	33	BPNN(6,3,1)	0.00119763	0.00151087
16	BPNN(3,4,1)	0.24819011	0.6584132	34	BPNN(6,4,1)	0.00117449	0.00128245
17	BPNN(3,5,1)	0.00160018	0.00303003	35	BPNN(6,5,1)	0.00113781	0.00127506
18	BPNN(3,6,1)	0.00124258	0.00101355	36	BPNN(6,6,1)	0.00126522	0.00133276

Table 1 shows that the best Backpropagation Neural Network model architecture for TLKM stock price is BPNN(1,3,1) because it has the smallest MSE testing around 0.00098483. This model consists of 1 input unit, 3 hidden units, and 1 output unit.

4.2.4. Forecasting with BPNN Model

After determining the best BPNN model architecture, the next step is to forecast the stock price of TLKM from September 1, 2023 to September 30, 2024. Then, the forecast results are denormalized using equation (23). The comparison plot between the actual data and the forecast results can be seen in Figure 3.

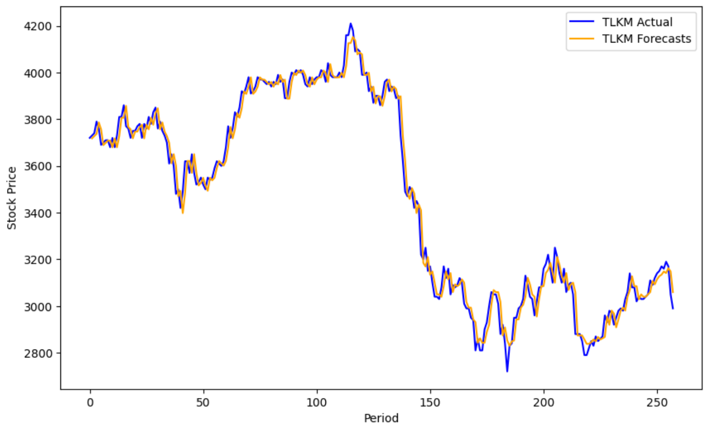

Figure 3: Plot of actual data and forecast results of the BPNN model for TLKM

In Figure 3, the blue line shows the actual value of the data and the orange line shows the forecast value. It can be seen that the two lines are close together on the comparison chart. This indicates that the BPNN model used can capture the data movement pattern well. Then, the MAPE calculation is carried out using Python software, obtaining a MAPE of 1.17% for TLKM.

4.3. MAPE Comparison of ARIMA and BPNN Models

To find out a better model in predicting TLKM stock price, it is necessary to conduct a comparative analysis of the level of prediction accuracy of each model. The level of prediction accuracy of the model can be calculated using the Mean Absolute Percentage Error (MAPE). The comparison of MAPE for the ARIMA and BPNN models for TLKM can be seen in Table 2.

Tabel 2. MAPE Comparison of ARIMA and BPNN Models

Issuer Code	ARIMA Model	MAPE	BPNN Model	MAPE
TLKM	ARIMA(0,1,3)	1,20%	BPNN(1,3,1)	1,17%

Based on Table 2, the best ARIMA model for predicting the stock price of TLKM is ARIMA(0,1,3) with a MAPE value of 1.20%, indicating that the model is very accurate and only the moving average (MA) component is significant. On the other hand, the best BPNN model for TLKM is BPNN(1,3,1) with a MAPE value of 1.17%, indicating that this model is better than ARIMA for TLKM.

5. Conclussion

This study has successfully use the Autoregressive Integrated Moving Average (ARIMA) and Backpropagation Neural Network (BPNN) model to predict TLKM stock price in the future. The result shows that the best ARIMA model, selected based on the least AIC value, is ARIMA(0,1,3) with a MAPE value of 1.20%. Meanwhile, the best BPNN model selected from the smallest testing MSE value, is BPNN(1,3,1) with a MAPE value of 1.17%. Among those two models, the BPNN model is more accurate because it has less MAPE value compared to the ARIMA one.

As a recommendation for further research, consider using the ARIMA-GARCH model to predict as well as analyze stock price volatility. Additionally, the use of other activation functions can be explored, such as Hyperbolic Tangent.

References

Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Hoboken: Prentice Hall, Inc.

Islam, M., Chen, G., & Jin, S. (2019). An Overview of Neural Network. *American Journal of Neural Networks and Applications*, *5*(1), 7–11.

Kittichotsatsawat, Y., Boonprasope, A., Rauch, E., Tippayawong, N., & Tippayawong, K. Y. (2023). Forecasting arabica coffee yields by auto-regressive integrated moving average and machine learning approaches. *AIMS Agriculture and Food*, *8*(4), 1052–1070.

Mustafidah, H., & Suwarsito, S. (2020). Performance of Levenberg-Marquardt Algorithm in Backpropagation Network Based on the Number of Neurons in Hidden Layers and Learning Rate. *JUITA: Jurnal Informatika*, *8*(1), 29–35.

Pallant, J. (2011). *SPSS SURVIVAL MANUAL*. New York: Routledge.

Singh, A., Kushwaha, S., Alarfaj, M., & Singh, M. (2022). Comprehensive Overview of Backpropagation Algorithm for Digital Image Denoising. *Electronics (Switzerland)*, *11*(10), 1–15.

Sukono, Subartin, B., Ambarwati, Napitupulu, H., Saputra, J., & Hidayat, Y. (2019). Forecasting model of gross regional domestic product (GRDP) using backpropagation of Levenberg-Marguardt method. *Industrial Engineering and Management Systems*, *18*(3), 530–540.

Taslim, D. G., & Murwantara, I. M. (2024). Comparative analysis of ARIMA and LSTM for predicting fluctuating time series data. *Bulletin of Electrical Engineering and Informatics*, *13*(3), 1943–1951.

Wei, L., Yu, Z., Jin, Z., Xie, L., Huang, J., Cai, D., He, X., & Hua, X. S. (2019). Dual Graph for Traffic Forecasting. *IEEE Access*, *4*, 1–10.

Wei, W. W. . (2006). *Nonstationary Time Series Models*. London: Pearson Education.

Works, R. (2021). An overview of activation functions used in Neuronal Networks: Application for plant disease diagnosis. *International Journal of Advanced Trends in Computer Science and Engineering*, *10*(1), 139–145.