Dynamic Analysis of the Jakarta Composite Index: A Time Series Approach Using Monthly Data from 2009 to 2024

Elvano Jethro Mogi Pardede  
*Statistics Department*  
*School of Computer Science, Bina Nusantara University*  
Jakarta, Indonesia 11480  
[elvano.pardede@binus.ac.id](mailto:elvano.pardede@binus.ac.id)

Rasyad Muhammad Ramdhanazuri  
*Statistics Department*  
*School of Computer Science, Bina Nusantara University*  
Jakarta, Indonesia 11480  
[rasyad.ramdhanazuri@binus.ac.id](mailto:rasyad.ramdhanazuri@binus.ac.id)

*Abstract*—This study explores the dynamics of the Jakarta Composite Index (IHSG) using monthly data from January 2009 to April 2024. The IHSG serves as a key indicator of stock market performance in Indonesia, influenced by various economic and political events both domestically and internationally. The primary objective is to identify the most effective time series prediction model for forecasting IHSG movements by evaluating multiple models, including the Naïve method, Double Moving Average (DMA), Double Exponential Smoothing (DES), Time Series Regression, ARIMA, and Neural Networks (NN). The analysis reveals that while the DMA model with a window size of 5 shows lower errors on the testing dataset, the significant difference between training and testing errors indicates overfitting. This overfitting suggests that the DMA model captures noise within the training data, leading to poor generalization on unseen data. On the other hand, the NN model with a hidden layer size of 5 demonstrates robust performance with minimal differences between training and testing errors, indicating a good fit. This balance makes the NN model a more reliable and effective choice for forecasting the IHSG.

Keywords—time series analysis, neural networks, stock market, IDX composite, double moving average, double exponential smoothing

# Introduction

The Jakarta Composite Index (IHSG) serves as the primary indicator used to measure the performance of the stock market in Indonesia. Reflecting the performance of stocks across various industrial sectors, IHSG provides a comprehensive overview of the overall condition of the capital market.[1] As an economic barometer, IHSG not only mirrors investor sentiment towards domestic economic conditions but is also influenced by a multitude of external factors such as global economic conditions, monetary policies, and geopolitical turmoil.

In recent decades, global stock markets, including Indonesia's, have experienced significant dynamics. The fluctuations in IHSG are often influenced by various economic and political events at both national and international levels. For instance, changes in interest rate policies by Bank Indonesia, international trade policies, and global commodity prices such as oil and gas, all have the potential to affect the movement of IHSG. Additionally, domestic factors like economic growth, inflation rates, and political stability play crucial roles in determining the direction of IHSG.

Time series analysis is one of the methods used to understand and predict the movement of IHSG. This analysis helps identify historical patterns and trends that may recur in the future. By employing this approach, analysts and investors can make more accurate predictions about the future performance of IHSG, thereby aiding in better investment decision-making.

This study aims to analyze the dynamics of IHSG movements from January 2009 to April 2024, gaining deeper insights into existing trends and patterns. Additionally, the study seeks to identify the most effective time series prediction model for forecasting IHSG movements based on historical data. Finally, the study evaluates the performance of these prediction models by comparing errors on test data, thus identifying the best model for predicting IHSG.

# Methodology

## Dataset

This study uses monthly IHSG data from January 2009 to April 2024 [2] to explore the dynamics of the index movement. This period was chosen because it covers a variety of different economic phases, ranging from the post-2008 global financial crisis period, an era of rapid economic growth, to the COVID-19 pandemic and its impact on the stock market. This dataset provides 3 variables to be used for time series analysis.

Dataset variables

| Variable | Variable Description |
| --- | --- |
| Year () | This variable represents the year in which the IHSG data was recorded. It ranges from 2009 to 2024 |
| Month | This variable denotes the month of the year when the IHSG data was recorded, ranging from January to December. |
| IHSG () | The value of the composite stock price index, which is the main focus of the study |

## Forecasting Method

### Naïve Method

The naive trend forecasting method is a simple yet effective approach for predicting future values in a time series. This method assumes that the trend observed in the most recent period will continue. The forecast for the next period is calculated by adding the most recent change in the series to the last observed value. Specifically, the naive trend forecast formula is given by:



Where in (1), ​ represents the forecast for the next period, is the value at the current period, and ​ is the value at the previous period. This approach is particularly useful for datasets with a consistent trend and is favored for its simplicity and ease of implementation. [3]

### Double Moving Average

The double moving average method is an extension of the simple moving average technique, used to smooth out fluctuations in a time series and to forecast future values. It involves calculating two moving averages: the first moving average smooths the original series, and the second moving average smooths the first moving average. This method helps in identifying and projecting trends more accurately. The formulas involved in the double moving average method [4] are as follows:

2

Equation (2) calculates the first moving average ​, which is the average of the last n observations of the time series.

3

Equation (3) calculates the second moving average , which is the moving average of the first moving averages over the same period of n.

4

5

6

Equation (6) provides the forecast for p periods into the future, where is the level component from (4) and is the trend component from (5).

### Double Exponential Smoothing

Double exponential smoothing, also known as Holt’s linear trend method, is a forecasting technique used to account for data with trends. It extends simple exponential smoothing by adding a component to capture the trend in the data. This method uses two equations to update the level and trend estimates at each time step, and a forecast equation for future values. The formulas involved [4], [5] are:

7

8

Equation (7) updates the smoothed level by combining the actual value ​ with the previous level adjusted by the previous trend, where is the smoothing parameter for the level with  and the initial value of the level in (8).

9

10

Equation (9) updates the trend estimate ​ by combining the difference between the current and previous levels with the previous trend, where is the smoothing parameter for the trend with and the initial value of the trend in (10).

11

Equation (11) forecasts the value p periods ahead by adding the current level ​ and trend ​.

### Time Series Regression

#### Linear Trend

Time series regression is a fundamental approach in time series forecasting, especially when examining trends over time.

12

Equation (12) [4] represents a simple linear regression model where ​ is the value of the time series at time , is the intercept, is the slope (which indicates the rate of change over time), and represents time periods.

13

14

In practical terms, this regression model allows analysts to quantify and predict how the time series evolves over time. The intercept provides a baseline value, while the slope indicates whether the series is increasing or decreasing over time and at what rate. By fitting this linear relationship to historical data, analysts can extrapolate future values of based on the established trend. This method is foundational in economics, finance, and other fields where understanding and predicting temporal patterns are crucial for decision-making.

#### Exponential Trend

The exponential trend model is used in time series analysis to capture growth or decay patterns that follow an exponential curve over time.

15

16

17

18

Equation (15) represents an exponential trend model where denotes the value of the time series at time , (16) represents the initial value of the series when , (17) signifies the growth rate parameter, (18) transforms the original values into their natural logarithms, making the data linearizable for regression analysis. In practice, this exponential trend model is particularly useful for forecasting phenomena that exhibit compounded growth or decay, such as population growth, economic trends with compound interest, or technological adoption rates. By fitting this model to historical data, analysts can project future values of ​ based on the established exponential growth or decay pattern. This approach provides insights into the magnitude and pace of change over time, helping decision-makers anticipate future trends and plan accordingly. [6]

#### Quadratic Trend

The quadratic trend model is a regression model used in time series analysis to capture nonlinear trends over time.

19

Equation (19) represents a quadratic trend model where ​ represents the value of the time series at time , is the intercept (the value of when ), is the linear coefficient (which captures the linear trend component), and is the quadratic coefficient (which captures the curvature or acceleration of the trend over time). The linear coefficient () determines the rate of change of with respect to . A positive indicates an increasing trend, and otherwise. The quadratic coefficient () is the term captures the curvature of the trend over time. A positive c indicates the trend is accelerating upwards, and otherwise. [6]

The quadratic trend model is useful when the underlying data exhibits a curvature or acceleration that cannot be captured by a simple linear trend. It allows analysts to understand and forecast more complex patterns in time series data, providing insights into the direction and pace of change over time. This makes it applicable in various fields such as economics (e.g., analyzing economic growth patterns), demographics (e.g., population trends), and physics (e.g., modeling the trajectory of objects under acceleration).

#### Lag-1 Trend

The lag-1 trend model is a type of autoregressive model commonly used in time series analysis to incorporate the effect of the previous time period's value ​ on the current time period .

20

Equation (20) represents a lag-1 trend model where represents the value of the time series at time , s the intercept term, which represents the baseline value of when , is the coefficient of (capturing the linear trend component over time), and is the coefficient of ​, which quantifies how much the previous time period's value influences the current value. This coefficient () reflects the impact of the previous time period’s value on the current value . A positive indicates a positive correlation between and . [6]

#### Lag-2 Trend

A different model of the Lag-1 trend model that using the second of the previous time period’s value .

21

#### Lag-1 Lag-2 Trend

A combination model of the Lag-1 trend model and Lag-2 trend model

22

### Autoregressive Integrated Moving Average (ARIMA)

ARIMA (Autoregressive Integrated Moving Average) models are powerful tools in time series forecasting, capable of capturing both autocorrelation and trend components in data.

23

24

25

Equation (23) breaks down into several components essential for understanding ARIMA's functionality. The represents the time series data, is the backward shift operator (which shifts the series back by one time), (24) represents the autoregressive (AR) polynomial, and (25) represents the moving average (MA) polynomial. The parameter denotes differencing, a process used to stabilize non-stationary time series by transforming them into stationary series. [7]

The ARIMA model operates by fitting these polynomials to historical data, where incorporates lagged values of , represents the differencing operator applied times to achieve stationarity, is a constant term, and models the moving average of the errors . The AR part (24) models the impact of previous values on the current value, while the MA part (25) accounts for the influence of past error terms. Together, these components allow ARIMA to flexibly model a wide range of time series patterns, making it a versatile tool for forecasting future values based on historical data patterns and trends.

### Neural Network

Neural Network is an adaptive statistical model that is customized like the human brain. Neural networks are used as statistical tools in various fields such as engineering, econometrics, psychology, and others. In a neural network there are two structures, namely neurons and network units. Where the unit neurons are connected to each other by a weighted link. Each unit of neurons contains information about the characteristics of a pattern to be analyzed. While the network is a weighted link to do learning. The learning is obtained by modifying the weight value contained in each network between unit neurons. [8]

26

This equation encapsulates the recursive nature of neural networks, where the current output is determined by a combination of a bias term (), the influence of the previous output (), and an added noise or error component (). This autoregressive model highlights how neural networks leverage past information to inform current predictions, thus enabling them to capture temporal dependencies and patterns within the data. Understanding this formula is important for holding how neural networks process sequential information and evolve over time.

## Model Evaluation

### Root Mean Square Error (RMSE)

27

where is the forecast error for the i-th observation (i.e., the difference between the actual value , and the forecasted value , and n is the number of observations. RMSE measures the square root of the average of the squared errors, giving more weight to larger errors. It is useful for capturing the magnitude of errors and is sensitive to outliers.[9]

### Mean Absolute Error (MAE)

28

where is the absolute forecast error for the i-th observation. MAE calculates the average of the absolute errors, providing a straightforward measure of forecast accuracy. Unlike RMSE, it treats all errors equally, making it less sensitive to outliers.[9]

### Mean Absolute Percentage Error (MAPE)

29

where is the forecast error and is the actual value for the i-th observation. MAPE expresses the error as a percentage of the actual values, making it easy to interpret and compare across different time series. It is particularly useful when the scale of the data varies.[10]

# Result and Discussion

## Data Preparation

Before initiating on time series prediction modeling, it is crucial to understand the nature of the dataset. Observing the line plot of the IHSG dataset, it becomes evident that the overall trend is upward, despite some periods of decline. The downward movements do not exhibit any clear seasonality, suggesting that the IHSG dataset primarily exhibits a trend rather than seasonal patterns. Therefore, appropriate forecasting methods that account for the trend will be employed.

IHSG Line Plot

In this study, a variety of forecasting methods will be compared, ranging from the simplest to more complex approaches. These methods include the naive method, double moving average, double exponential smoothing, time series regression, ARIMA (Autoregressive Integrated Moving Average), and neural networks.

## Modeling & Evaluation

### Naïve Model

In the naïve model, the model is so simple that it only requires two previous data to predict the current data. For instance, at time , the model is given by:

From that, we got the predictions of equal to 1238.29. After performing similar calculations for all the data points, we then compute the RMSE, MAE, and MAPE to evaluate the performance of the naïve model:

Evaluation of Naïve Model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Training** | | | **Testing** | | |
| *RMSE* | *MAE* | *MAPE* | *RMSE* | *MAE* | *MAPE* |
| Naive | 328.346 | 217.345 | 0.047 | 277.041 | 240.344 | 0.035 |

After the Naïve model performance evaluation results (Table II), the naive model from the training data gets an RMSE value of 328.346, MAE value of 217.345, and MAPE value of 4.7%. And the naïve model from the testing data gets an RMSE value of 277.041, MAE value of 240.344, and MAPE value of 3.5%. This indicates that naïve model has a balance between training and testing errors, which means that the model is both generalizable and robust.

### Double Moving Average

Comparing N in Double Moving Average

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Training** | | | **Testing** | | |
| *RMSE* | *MAE* | *MAPE* | *RMSE* | *MAE* | *MAPE* |
| DMA(2) | 369.761 | 265.012 | 0.0568 | 605.465 | 577.73 | 0.0831 |
| DMA(3) | 373.003 | 279.385 | 0.0589 | 442.264 | 412.542 | 0.0596 |
| DMA(5) | 394.008 | 294.744 | 0.0618 | 223.432 | 199.097 | 0.0287 |
| DMA(6) | 306.500 | 164.501 | 0.0364 | 553.524 | 528.781 | 0.0762 |

After evaluating the performance of the Double Moving Average (DMA) model with various window sizes n, we observe distinct differences in error metrics across the training and testing datasets. For n=2 and n=3, while the training errors (RMSE, MAE, and MAPE) are relatively low, the testing errors are considerably higher, indicating poor generalization. Although the DMA model with n=6 achieves the lowest training errors, the testing errors show a significant increase, suggesting overfitting. On the other hand, the DMA model with n=5 strikes a balance between training and testing errors. Despite having slightly higher training errors compared to n=6, it achieves the lowest testing errors (RMSE: 223.43, MAE: 199.10, MAPE: 0.0287), indicating better generalization and robustness. Therefore, we conclude that n=5 is the most suitable parameter for forecasting the IHSG using the double moving average method.

### Double Exponential Smoothing

Before going to the modeling stage, we need to determine the alpha and beta values first. To determine the alpha and beta values, we use the R program to get the best alpha and beta values. So, the output given by the R program is as follows:

|  |
| --- |
| ETS(M,Ad,N)  Call:  ets(y = data\_train)  Smoothing parameters:  alpha = 0.9481  beta = 1e-04  phi = 0.9727 |

1. Best alpha and beta output from R program

So, we will use alpha value is 0.95 and beta value is 0.0001 as double exponential smoothing parameters. By using them, we get double exponential smoothing model evaluation is as follows:

Evaluation of Double Exponential Smoothing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Training** | | | **Testing** | | |
| *RMSE* | *MAE* | *MAPE* | *RMSE* | *MAE* | *MAPE* |
| DESa | 230.012 | 164.502 | 0.0364 | 243.990 | 213.88 | 0.0310 |

1. The parameters: and

After the Double Exponential Smoothing model performance evaluation results (Table IV), the double exponential smoothing model from the training data gets an RMSE value of 230.021, MAE value of 164.501, and MAPE value of 3.6%. And the naïve model from the testing data gets an RMSE value of 243.99, MAE value of 213.878, and MAPE value of 3.1%. This indicates that double exponential smoothing model has a balance between training and testing errors, which means that the model is both generalizable and robust.

### Time Series Regression

In Time Series Regression models, we’re using six models to analysis the data, which are Linear Trend, Exponential Trend, Quadratic Trend, Lag-1 Trend, Lag-2 Trend, and Lag-1 Lag-2 Trend.

Significant & Residual Assumptions Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Significant & Residual Assumption** | | | |
| *Significant?* | *Normality* | *Independency* | *Homoscedasticity* |
| Linear Trenda | ✓ | 4.16E-06 | < 2.2e-16 | 8.79E-01 |
| Exponential Trendb | ✓ | < 2.2e-16 | < 2.2e-16 | 1.34E-04 |
| Quadratic Trendc | ✓ | 4.02E-03 | < 2.2e-16 | 4.91E-04 |
| Lag-1 Trendd | ✓ | 2.12E-05 | 0.988 | 0.04912 |
| Lag-2 Trende | ✓ | 0.000294 | 3.87e-11 | 0.04006 |
| Lag-1 Lag-2 Trend | X | - | - | - |

From the Significant & Residual Assumptions Test (Table V), Lag-1 Lag-2 Trend doesn’t pass the significant requirments. So, we don’t do residual assumptions test for that model. Otherwise, the remaining models don’t pass the residual assumptions test. Therefore, time series regression cannot be used in IHSG forecasting.

### ARIMA

Before going to the modeling stage, we need to do stationary tests for mean and variance. First, we’re going to do a stationary test to the variance using R program. So, the output given by the R program is as follows:

|  |
| --- |
| bcPower Transformation to Normality  Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  data\_train 1.6402 2 1.1729 2.1076  Likelihood ratio test that transformation parameter is equal to 0  (log transformation)  LRT df pval  LR test, lambda = (0) 61.28245 1 4.996e-15  Likelihood ratio test that no transformation is needed  LRT df pval  LR test, lambda = (1) 7.957132 1 0.0047898 |

1. Stationary test to the variance (R output)

From the output, the lambda (1) p-value is less than (0.05). Which the hypothesis is:

Since the lambda (1) p-value is less than , is rejected. This indicates that the data we have is not stationary with respect to variance. Therefore, we need to power transform the data by two. The next is do a stationary test to the mean using R program. So, the output given by the R program is as follows:

|  |
| --- |
| Augmented Dickey-Fuller Test  data: data\_pow  Dickey-Fuller = -2.5151, Lag order = 5, p-value = 0.3613  alternative hypothesis: stationary |

1. Stationary test to the mean (R output)

From the output, the Augmented Dickey-Fuller Test’s p-value is greater than (0.05). Which the hypothesis is:

Since the Augmented Dickey-Fuller Test’s p-value is greater than α, we fail to reject the null hypothesis . This indicates that the data is not stationary with respect to the mean. Therefore, we need to apply differencing to our data. The next step is to perform differencing using the R program and then conduct another stationarity test on the mean. The output provided by the R program is as follows:

|  |
| --- |
| Augmented Dickey-Fuller Test  data: d1  Dickey-Fuller = -5.0841, Lag order = 5, p-value = 0.01  alternative hypothesis: stationary |

1. Stationary test to the mean of differenced data (R output)

Thus, the Augmented Dickey-Fuller Test’s p-value using differenced data is less than , is rejected. This indicates that the data is now stationary with respect to the mean. Therefore, our data is ready for the next stage, which is the modeling stage.

Significant & Residual Assumptions Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Significant & Residual Assumption** | | |
| *Significant?* | *Normality* | *White Noise* |
| ARIMA(0,1,0) | X | - | - |
| ARIMA(0,1,1) | X | - | - |
| ARIMA(0,1,2) | X | - | - |
| ARIMA(0,1,3) | X | - | - |
| ARIMA(1,1,0) | X | - | - |
| ARIMA(1,1,1) | ✓ | 7.239e-10 | 0.9634 |
| ARIMA(1,1,2) | X | - | - |
| ARIMA(1,1,3) | X | - | - |

From the Significant & Residual Assumptions Test (Table VI), Only ARIMA(1,1,1) passed the significant requirement. Otherwise, the remaining models don’t pass the significant requirement. But ARIMA(1,1,1) doesn’t pass the residual assumption requirement. Therefore, ARIMA model cannot be used in IHSG forecasting.

### Neural Network

In neural network modeling, we iterate to get the best number of hidden layers and input layers for the data we have. Thus, the best number of hidden layers and input layers are 5 and 1. After that, we evaluate the model to produce the following evaluation values:

Evaluation of Every Best Model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Training** | | | **Testing** | | |
| *RMSE* | *MAE* | *MAPE* | *RMSE* | *MAE* | *MAPE* |
| NN | 222.340 | 150.023 | 0.032 | 236.781 | 210.502 | 0.030 |

After the Neural Network model performance evaluation results (Table VII), the neural network model from the training data gets an RMSE value of 222.340, MAE value of 150.023, and MAPE value of 3.2%. And the neural network model from the testing data gets an RMSE value of 236.781, MAE value of 210.502, and MAPE value of 3%. This indicates that neural network model has a balance between training and testing errors, which means that the model is both generalizable and robust.

## Model Evaluation Overall

Evaluation of Every Best Model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Training** | | | **Testing** | | |
| *RMSE* | *MAE* | *MAPE* | *RMSE* | *MAE* | *MAPE* |
| Naïve | 328.346 | 217.345 | 0.047 | 277.041 | 240.344 | 0.035 |
| DMAa | 394.008 | 294.744 | 0.062 | 223.432 | 199.096 | 0.029 |
| DESb | 230.012 | 164.501 | 0.036 | 243.990 | 213.878 | 0.031 |
| Regression | Does not pass the residual assumption test | | | | | |
| ARIMA | Does not pass the residual assumption test | | | | | |
| NNc | 222.340 | 150.023 | 0.032 | 236.781 | 210.502 | 0.030 |

1. Window size
2. and
3. Hidden layer

Table VIII provides a comprehensive evaluation of various forecasting models applied to the IHSG dataset, highlighting their performance through key error metrics: RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). These metrics are calculated for both the training and testing datasets, allowing for a detailed comparison of each model's accuracy and generalization capability.

Starting with the Naïve model, it is evident that this simple approach, which relies only on the two previous data points to predict the future value, yields moderate errors. Specifically, the Naïve model shows an RMSE of 328.346 and an MAE of 217.345 on the training set, with slightly lower errors on the testing set (RMSE: 277.041, MAE: 240.344). The MAPE values further illustrate that the Naïve model has a relatively high percentage error, indicating it might not be the most reliable for precise forecasting.

The Double Moving Average (DMA) model, particularly with a window size of n=5, shows a significant improvement in performance on the testing set, with an RMSE of 223.432 and an MAE of 199.096, which are lower compared to other models. However, the training errors for this model are much higher, with an RMSE of 394.008 and an MAE of 294.744. This significant difference between the training and testing errors suggests that the DMA model is overfitting to the training data. Overfitting occurs when a model learns the noise and details in the training data to the extent that it performs poorly on new, unseen data. This makes the DMA model less reliable for forecasting, as its predictions may not generalize well to future values.

In contrast, the Double Exponential Smoothing (DES) model demonstrates balanced accuracy across both training and testing datasets. The training errors (RMSE: 230.012, MAE: 164.501) and testing errors (RMSE: 243.990, MAE: 213.878) are close, which indicates that the model is neither underfitting nor overfitting. The MAPE values for DES are also relatively low, suggesting that this model maintains a consistent prediction accuracy.

However, both the Regression and ARIMA models fail to meet the necessary residual assumptions for valid predictions. This failure indicates that these models are not suitable for the IHSG dataset, as their residuals do not exhibit the required properties, such as constant variance or lack of autocorrelation, making their predictions unreliable.

Notably, the Neural Network (NN) model with a hidden layer size of 5 emerges as the best-performing model among all evaluated. The NN model achieves the lowest RMSE (222.340) and MAE (150.023) on the training set, and similarly low errors on the testing set (RMSE: 236.781, MAE: 210.502). The MAPE values further confirm its superiority, reflecting minimal percentage errors. Despite the DMA model having slightly better testing errors, the NN model is favored because of the minimal difference between its training and testing errors. This indicates excellent generalization capability, meaning the NN model effectively captures the underlying patterns in the data without overfitting. This robustness makes the NN model highly reliable for forecasting future values.

# Conclution

While the DMA model shows lower testing errors, the significant disparity between its training and testing errors highlights its susceptibility to overfitting, making it a less reliable choice for forecasting. On the other hand, the Neural Network model, despite having slightly higher testing errors, demonstrates a good fit due to its consistent performance across training and testing datasets. This makes the NN model not only the best among the evaluated methods but also a robust and reliable choice for future time series forecasting tasks.

##### References

[1] P. Adam, “Statistical Characteristics of Jakarta Composite Index (JCI) Dynamics based on Short Term Data Represented in Candles,” *International Journal of Economics, Finance and Management Sciences*, vol. 2, no. 2, p. 138, 2014, doi: 10.11648/j.ijefm.20140202.14.

[2] BPS, “Transaksi dan Indeks Saham di Bursa Efek.” May 2024. [Online]. Available: https://www.bps.go.id/id/statistics-table/2/MTI1IzI%3D/transaksi-dan-indeks-saham-di-bursa-efek.html

[3] P. C. Padhan and others, “Use of univariate time series models for forecasting cement productions in India,” *International Research Journal of Finance and Economics*, vol. 83, pp. 167–179, 2012.

[4] R. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, 3rd ed. Melbourne, Australia: OTexts, 2021. Accessed: Jun. 14, 2024. [Online]. Available: https://otexts.com/fpp3/

[5] E. S. Gardner and Ed. Mckenzie, “Forecasting Trends in Time Series,” *Manage Sci*, vol. 31, no. 10, pp. 1237–1246, Oct. 1985, doi: 10.1287/mnsc.31.10.1237.

[6] D. Doane and L. Seward, *Applied Statistics in Business and Economics*, 3rd ed. New York: The McGraw-Hill, 2011.

[7] W. A. Woodward, B. P. Sadler, and S. Robertson, *Time Series for Data Science*. New York: Chapman and Hall/CRC, 2022. doi: 10.1201/9781003089070.

[8] H. Abdi, D. Valentin, and B. Edelman, *Neural Networks*, 124th ed. Sage, 1999.

[9] T. O. Hodson, “Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not,” *Geosci Model Dev*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/gmd-15-5481-2022.

[10] J. McKenzie, “Mean absolute percentage error and bias in economic forecasting,” *Econ Lett*, vol. 113, no. 3, pp. 259–262, Dec. 2011, doi: 10.1016/j.econlet.2011.08.010.