e-ISSN 2620 – 8369
VOLUME 8 (ISSUE 2) DECEMBER 2024: 179-190
https://doi.org/10.21009/JSA.08205

# THE APPLICATION OF THE ARTIFICIAL NEURAL NETWORK (ANN) METHOD FOR FORECASTING THE SOUTHERN OSCILATION INDEX (SOI)

**Fathia Syahla Az Zahra [1*], Bagus Sumargo[2], Dania Siregar[3], Auria Yusrin Fathya[4]**

*[1,2,3,4] Statistics Study Program, Universitas Negeri Jakarta*
*Jln. Rawamangun Muka, East Jakarta, DKI Jakarta, 13220, Indonesia*

Corresponding author's e-mail: * **fathiaazzahra02@gmail.com**

*ABSTRACT*

*Article History:*
*Received: 4, December 2024*
*Revised: 24, December 2024*
*Accepted: 27, December 2024*
*Published: 31, December 2024*
*Available online.*

*Keywords:*
*Artificial Neural Network, Backpropagation, Southern Oscillation Index, Forecasting.*

Indonesia's seasons are influenced by global phenomena such as ENSO. This phenomenon affects rainfall intensity in Indonesia through its two main phases: El Nino and La Nina. One method to detect these events is by analyzing the Southern Oscillation Index (SOI). A highly accurate SOI forecasting model is critical for both short-term and long-term development planning, particularly in anticipating future extreme seasons. One of the methods used for forecasting is the Artificial Neural Network (ANN). This study aims to develop an ANN model capable of predicting the SOI index. Based on forecasting using training data, the optimal model architecture identified is 12-7-1, which achieved the smallest MSE value of 0.0095 and a MAPE of 17.6851. With an error rate below 20%, the 12-7-1 architecture demonstrates strong forecasting capabilities. The study forecasts the SOI index for the next 12 months, indicating a trend from negative values at the beginning of the year to more positive values toward the year's end.



*How to cite this article:*
F.S.A. Zahra, B. Sumargo, D. Siregar, A.Y. Fathya, "THE APPLICATION OF THE ARTIFICIAL NEURAL NETWORK (ANN) METHOD FOR FORECASTING THE SOUTHERN OSCILATION INDEX (SOI)", Journal Statistika dan Aplikasinya, vol. 8, iss. 2, pp. 179 – 190, December 2024


Journal e-mail: *jsa@unj.ac.id*
Research Article · Open Access

## 1. INTRODUCTION

ENSO (*El Nino Southern Oscillation*) is a phenomenon involving deviations in sea surface, characterized by warmer or cooler water than normal conditions in the central and eastern Pacific Ocean. ENSO is a *non*-periodic global climate system with two phases, El Nino, and La Nina. El Nino refers to the warm phase, marked by an increase in sea surface temperatures in the equatorial Pacific Ocean, while La Nina is the cool phase signified by decrease in sea surface temperatures in the same region. El Nino can lead to a drop in sea surface temperatures in Indonesian waters, while La Nina tends to cause an increase in sea surface temperatures in Indonesian waters (Kovats, 2000; Xiao & Mechoso, 2009; Luo et al., 2010; Aldrian et al., 2011; Wang et al., 2017; Fitria & Pratama, 2013).

Indonesia is located near the equator, positioned between the continents of Asia and Australia and the Indian and Pacific Oceans. Its equatorial location makes Indonesia's seasonal system influenced by various factors. The onset of the rainy and dry seasons does not occur consistently each year (Tjasyono, 2008). According to Yamagata et al. (2002), this variability is due to Indonesia's seasons being affected by global phenomena such as ENSO.

The ENSO phenomenon influences rainfall intensity in Indonesia. ENSO phenomenon has significant impacts on the climate and oceans, especially in the eastern part of Indonesia (Aldrian & Susanto, 2003). Both climate anomalies, El Nino and La Nina are crucial in shaping the annual climate, typically causing changes in rainfall volume (Hidayat & Ando 2014). In addition to affecting rainfall intensity, ENSO also affects extreme weather, prolonged dry and rainy seasons, and droughts across various regions in Indonesia (Irawan, 2006). If the El Nino phenomenon occurs during the dry season, it tends to exacerbate the impact of drought, and if it happens during the rainy season, it reduces rainfall during that period. Conversely, when the La Nina phenomenon occurs, it generally increases rainfall intensity, both during the dry and rainy seasons (Aldrian, 2002 & 2008; Utami et al., 2011).

One way to detect El Niño and La Niña events is by examining the Southern Oscillation Index (SOI). The SOI represents the difference in atmospheric pressure at sea level between Tahiti (eastern Pacific) and Darwin (western Pacific), which is caused by the temperature differences in the sea surface between these two regions (Zakir et al., 2009).

Climatologists and meteorologists often use the SOI to assess the strength of ENSO. Historical extreme seasonal events in various parts of the world are closely associated with the intensity and duration of the SOI. Developing a highly accurate SOI forecasting model is essential for effective long-term and short-term planning. Especially in anticipating future extreme seasonal changes.

In statistics, one method for predicting the future is forecasting. Forecasting is used to estimate what will happen in the future based on past data. Predictions about the future are made using historical data from a variable, which is referred to as a time series (Makridakis, 1983).

One of the methods used for forecasting is the Autoregressive Integrated Moving Average (ARIMA) method. ARIMA was developed by George E. P. Box and Gwilym M. Jenkins in 1970 for time series analysis and forecasting (Cryer & Chan, 2008). This method is reliable for short-term forecasting; however, its accuracy tends to decline when applied to long-term forecasting (Brockwell & Davis, 1996). For forecasting SOI data, it is expected that the predictions can be made over the long term.

In this case, a machine learning approach can be used, as machine learning develops models that can learn from data and make predictions or decisions without being explicitly programmed for specific tasks. One such machine learning approach is the Artificial Neural Network (ANN). Unlike ARIMA, ANN models do not require assumptions (Kusumadewi, 2014).

Artificial Neural Networks (ANN) are a forecasting method that can be used to predict nonlinear time series (Hikmah, 2017). According to Fausett (1994), ANN has become an important research subject in the fields of artificial intelligence and cognitive computing. ANN is capable of modeling and solving various problems, such as pattern recognition, prediction, and nonlinear data processing.

## 2. METHODS

### Data

The data used in this study are secondary data. The data were obtained from the official website of the Australian Government Bureau of Meteorology (www.bom.gov.au). The dataset consists of monthly SOI data from January 2004 to December 2023, with a total of 240 observations.

### Research Method

*Artificial Neural Network*

*Artificial Neural Network* (ANN) is an information processing system that resembles biological neural networks. ANN is an artificial representation of the human brain, designed to simulate the learning processes of the human brain (Fausett, 1994). *Artificial Neural Network* consists of interconnected neurons. The neurons are linked by weights, which transmit signals from one neuron to another. Each neuron can receive multiple inputs and generate a single output.

*Artificial Neural Network Architecture*

In general, the architecture of an Artificial Neural Network (ANN) can be observed from the number of layers and the number of neurons in each layer. The ANN consists of three layers: the input layer, hidden layer, and output layer (Pradana et al., 2022).

*Weight Initialization*

Weights represent the information used by the network to solve problems. Weight initialization is performed randomly. Random initialization is the most used method for initializing weights (Kholis & Rofii, 2017). According to Fausett (1994), weight initialization is done with small random numbers. Initial weight initialization is carried out before the training process begins and is applied to each neuron that is interconnected.

*Binary Sigmoid Activation Function*

The activation function is used to update the weight values based on all input values. In backpropagation, the activation function is crucial, as it determines the magnitude of the weights (Puspitaningrum, 2006). Activation function in ANN, when the output values need to be in the range from 0 to 1, the binary sigmoid function is commonly used because it produces values within the range (0, 1). The binary sigmoid function is defined as follows (Fausett, 1994).

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

where $f(x)$ is the binary sigmoid activation function, and $e^{-x}$ is the exponential value.

*Data Normalization*

Normalization is the process applied to raw data to ensure that the data used falls within a specific range (Budianita & Prijodiprodjo, 2013). The goal of the normalization process is to transform the data so that its range is not too wide, while still preserving the original characteristics of the data. Normalization is necessary because the activation function used is the binary sigmoid function, which is an asymptotic function that never reaches 0 or 1. Therefore, data normalization is performed by transforming the data into a smaller interval using the following formula (Siang, 2005):

$$X' = \frac{0{,}8(x-a)}{b-a} + 0{,}1 \tag{2}$$

where $X'$ is the normalized data, $x$ is the original data, $a$ is the minimum value, and $b$ is the maximum value.

*Backpropagation*

Backpropagation was popularized by Rumelhart, Hinton, and Williams in 1986 for use in neural networks (Kusumadewi, 2004). Backpropagation is one of the most used ANN algorithms to solve problems related to identification, prediction, pattern recognition, and others (Anwar, 2011). Backpropagation uses a weight adjustment pattern to minimize the error between the forecasted output and the actual output (Junaidi et al., 2022). Backpropagation is a procedure that iteratively adjusts the connection weights in the network to minimize the discrepancy between the actual network output and the desired output (Rumelhart, Hinton, & Williams, 1986). The backpropagation training algorithm, according to Fausett (1994), is as follows:

**Phase I : *Feedforward Propagation***

The first step is to initialize the weights and biases with small random numbers. Then, each input value ($x_i, i = 1, \dots, n$) receives a signal and passes this signal to all units in the hidden layer. The hidden layer units ($z_j, j = 1, \dots, p$) compute the weighted sum of the input signals,

$$z\_in_j = v_{0j} + \sum_{n=1}^{i} x_i \; v_{ij} \tag{3}$$

Next, the activation function is applied to compute the output signal,

$$z_j = f(z\_in_j) \tag{4}$$

where $z\_in_j$ is the $z$ *input* to the $j$-th hidden unit, $v_{0j}$ is the bias weight for the $j$-th *hidden layer*, $x_i$ is the input value from the $i$-th unit in the *input layer*, $v_{ij}$ is the weight from the $i$-th unit in the *input layer* to the $j$-th unit in the hidden layer, and $z_j$ is the output of the $j$-th hidden unit.

For the output layer, the process is similar. Each output unit ($y_k, k = 1, \dots, m$) computes the weighted sum of the input signals,

$$y\_in_k = w_{0k} + \sum_{k=1}^{p} z_j \; w_{jk} \tag{5}$$

Then, the activation function is applied to compute the output signal,

$$y_k = f(y\_in_k) \tag{6}$$

where $y\_in_k$ is the $k$-th input observation for the output layer, $w_{0k}$ is the bias weight for the $k$-th unit in the hidden layer, $z_j$ is the $j$-th unit in the hidden layer. $w_{jk}$ is the weight from the $j$-th unit in the hidden layer to the $k$-th unit in the output layer, and $y_k$ is the $k$-th output observation in the output layer.

Next, the error is calculated using the Mean Squared Error (MSE) function before backpropagation,

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(t_k - y_k)^2 \tag{7}$$

where $n$ is the number of data points, $t_k$ is the target output for the $k$-th unit, and $y_k$ is the output from $k$-th unit in the output layer.

**Phase II : *Backpropagation***

Each output neuron ($y_k, k = 1, \dots, m$) receives the target pattern corresponding to the input pattern during training. Then, the error correction factor for each layer is computed,

$$\delta_k = (t_k - y_k)y_k(1 - y_k) \tag{8}$$

where $\delta_k$ is the error correction factor for the weight $w_{jk}$.

The weight correction is computed for later use to adjust $w_{jk}$,

$$\Delta w_{jk} = \alpha\delta_k\, z_j \tag{9}$$

The bias correction for later use to adjust $w_{0k}$ is calculated as,

$$\Delta w_{0k} = \alpha\delta_k \tag{10}$$

where $\Delta w_{0k}$ is the bias correction for the $k$-th output layer, $\Delta w_{jk}$ is the weight correction from the $j$-th unit in the hidden layer to the $k$-th unit in the output layer, $\alpha$ is the learning rate, and $\delta_k$ is the error correction factor for the weight $w_{jk}$.

Each hidden unit ($z_j, j = 1, \dots, p$) computes the sum of the inputs from the above layer (output layer),

$$\delta\_in_j = \sum_{i=1}^{m}\delta_k\, w_{jk} \tag{11}$$

Then, the derivative of the activation function is applied to compute the error correction factor,

$$\delta_j = \delta\_in_j\, z_j(1 - z_j) \tag{12}$$

where $\delta\_in_j$ is the $\delta$ *input* to the $j$-th hidden unit, $\delta_k$ is the error correction factor for the weight $w_{jk}$, $w_{jk}$ is the weight from the $j$-th unit in the hidden layer to the $k$-th unit in the output layer, is the error correction factor for the weight $v_{ij}$.

The weight correction for $v_{ij}$ is computed as,

$$\Delta v_{ij} = \alpha\delta_j x_i \tag{13}$$

The bias correction for $v_{0j}$ is calculated as,

$$\Delta v_{0j} = \alpha\delta_j \tag{14}$$

where $\Delta v_{ij}$ is the weight correction for $v_{ij}$, $\Delta v_{0j}$ is the bias correction for $v_{0j}$, $\alpha$ is the learning rate constant, $\delta_j$ is the error correction factor for the weight $v_{ij}$, $x_i$ is the input value in the $j$-th unit of the input layer.

**Phase III : *Weight and Bias Update***

Each output unit ($y_k, k = 1, \dots, m$) and each hidden unit ($z_j, j = 1, \dots, p$) computes the weight update,

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \tag{15}$$

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \tag{16}$$

Each output unit ($y_k, k = 1, \dots, m$) and hidden unit ($z_j, j = 1, \dots, p$) computes the bias update,

$$w_{0k}(new) = w_{0k}(old) + \Delta w_{0k} \tag{17}$$

$$v_{0j}(new) = v_{0j}(old) + \Delta v_{0j} \tag{18}$$

Finally, the stopping condition is checked when the target error is achieved. If the stopping condition is met, the network training is stopped. Otherwise, the forward propagation steps are repeated until the weight and bias changes reach the stopping condition.

*Data Denormalization*

Denormalization is the process of converting normalized values back to their original scale, where the initial data was within the range [0, 1]. The data is then transformed into its original or pre-normalization values using the following formula (Siang, 2005):

$$x = \left[\frac{(X' - 0{,}1)}{0{,}8}\right] + (b - a) + a \tag{19}$$

where $X'$ is the predicted value from the normalized data, $x$ is the predicted value with the original range, $a$ is the minimum data value, and $b$ is the maximum data value.

*Mean Squared Error*

According to Suryaningrum (2015), *Mean Squared Error* (MSE) is another method for evaluating forecasting methods. Each error or residual is squared. MSE is the average squared difference between the forecasted values and the observed values. The formula for calculating MSE is as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{20}$$

where $n$ is the number of data points, $y_t$ is the actual value for period-$t$, $\hat{y}_t$ is the forecasted value for period-$t$.

*Forecasting Accuracy*

Forecasting accuracy can be calculated by determining the error rate. One method for calculating forecasting error is the Mean Absolute Percentage Error (MAPE). MAPE is calculated by measuring the percentage difference between the forecasted values and the actual data. The formula for calculating MAPE is as follows:

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|^2 \tag{21}$$

where $n$ is the number of data points, $\hat{y}_t$ is the forecasted value for period-$t$, and $y_t$ is the actual value for period-$t$. The MAPE criteria are as shown in Table 1 (Lewis, 1982).

**Table 1. The MAPE Criteria**

| MAPE | Forecasting Ability |
|---|---|
| < 10% | Excellent |
| 10% - 20% | Good |
| 20% - 50% | Fair |
| > 50% | Poor |

## 3. RESULTS

### Descriptive Analysis

As shown in Figure 1, the SOI index is not always stable. The highest SOI value reached 27,1 in December 2010, indicating the occurrence of the La Niña phenomenon, which can cause increased rainfall intensity in Indonesia. The lowest SOI value of -28,6 occurred in February 2005, indicating the opposite phenomenon, El Niño, which can lead to a decrease in rainfall intensity in Indonesia. The average SOI index during the period from 2004 to 2023 is 1,36, with a standard deviation of 10,37.

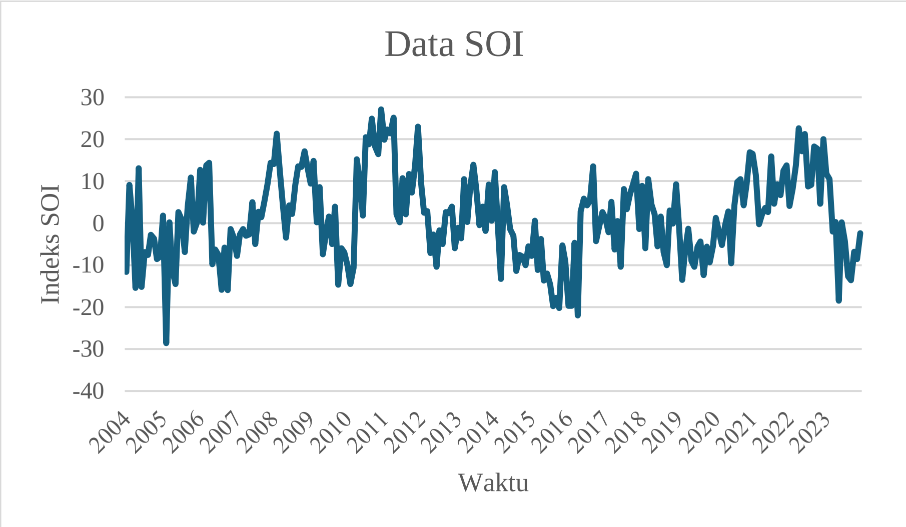


**Figure 1. SOI Index Graph**

### Arsitektur Model ANN

Table 2 presents the ANN model architectures used.

**Table 2. The list of ANN model architectures to be developed**

| Architecture |
|---|
| 12 – 5 – 1 |
| 12 – 6 – 1 |
| 12 – 7 – 1 |
| 12 – 8 – 1 |
| 12 – 9 – 1 |
| 12 – 10 – 1 |

### Training Model of ANN

Table 3 shows the results of each architecture that achieved the target error at the last epoch. The next step involved forecasting using the training data for each architecture to select the best one based on MSE and MAPE values.

**Table 3. MSE Results of Each Architecture in the Final Epoch of Training Data**

| Architecture | MSE | *Epoch* |
|---|---|---|
| **12-5-1** | 0,0078 | 315 |

| | | |
|---|---|---|
| **12-6-1** | 0,0087 | 306 |
| **12-7-1** | 0,0094 | 716 |
| **12-8-1** | 0,0083 | 348 |
| **12-9-1** | 0,0091 | 310 |
| **12-10-1** | 0,0098 | 644 |

As shown in Table 4, the lowest MSE and MAPE values from the training data forecast were achieved by the 12-7-1 architecture, with an MSE of 0.0095 and a MAPE of 17.6851. Therefore, the 12-7-1 architecture was identified as the best model.

**Table 4. MSE and MAPE Results from Forecasting the Training Data**

| Architecture | MSE | MAPE |
|---|---|---|
| **12-5-1** | 0,0109 | 19,2809 |
| **12-6-1** | 0,0103 | 18,4122 |
| **12-7-1** | 0,0095 | 17,6851 |
| **12-8-1** | 0,0111 | 19,4064 |
| **12-9-1** | 0,0109 | 19,2333 |
| **12-10-1** | 0,0128 | 18,2269 |

Figure 2 shows the results of forecasting the training data for the 12-7-1 architecture.

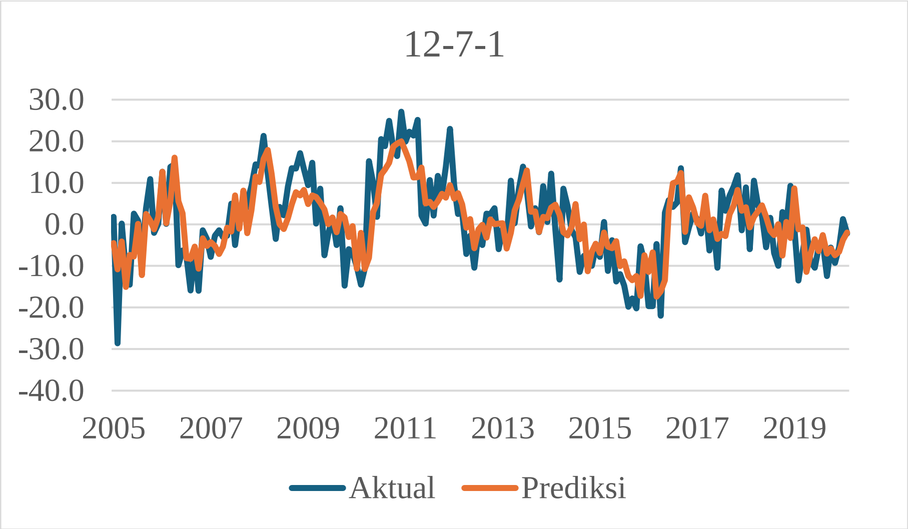


**Figure 2. Forecasting Results for Training Data of the 12-7-1 Architecture**

**Forecasting for Testing Data**

Table 5. presents the MSE and MAPE values achieved by 12-7-1 architecture model, with a MSE of 0,0136 and a MAPE of 17,3304. It can be seen that the MAPE value from the developed model is below 20%, indicating that the model has good forecasting capability.

**Table 5. MSE and MAPE Results for Forecasting the Testing Data of the 12-7-1 Architecture**

| Architecture | MSE | MAPE |
|---|---|---|
| **12-7-1** | 0,0136 | 17,3304 |

Figure 3. shows the results of forecasting with the 12-7-1 architecture on testing data, where the model is still unable to accurately capture the actual pattern.

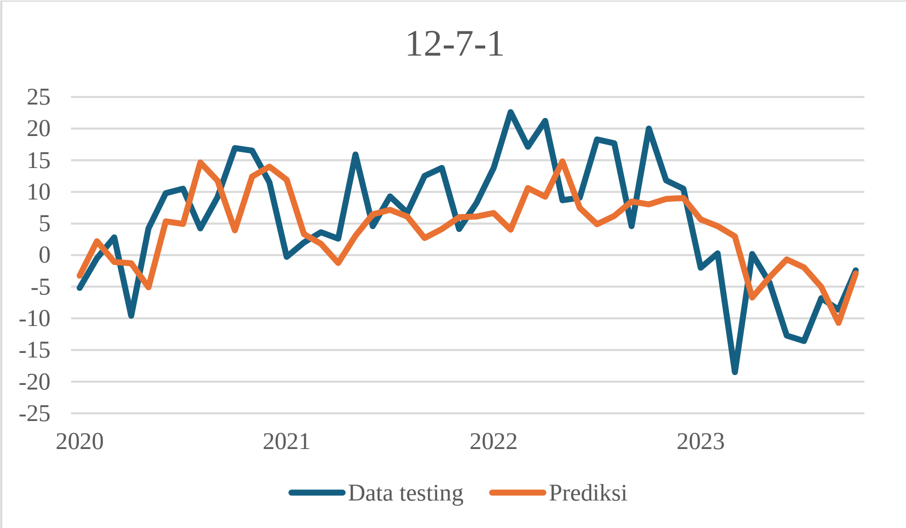


**Figure 3. Forecasting Results of the 12-7-1 Architecture ANN Model for Testing Data**

Based on the calculations performed, the results can be interpreted into the following formula:

$$z_j = f(v_{0j} + \sum_{n=1}^{i} x_i \ v_{ij}) \tag{22}$$

$$z_1 = f((-0{,}8070) + x_1 \ * (-0{,}6400) + x_2 * (3{,}3997) + x_3 * (-1{,}4594) + x_4 * (-0{,}2872) + x_5 * (-3{,}9883) + x_6 * (0{,}0507) + x_7 * (-6{,}7996) + x_8 * (4{,}2079) + x_9 * (-1{,}5605) + x_{10} * (-0{,}8725) + x_{11} * (0{,}1745) + x_{12} * (2{,}7903)$$

.

.

.

$$z_7 = f((-0{,}1811) + x_1 \ * (3{,}0954) + x_2 * (9{,}4052) + x_3 * (7{,}4076) + x_4 * (-3{,}3386) + x_5 * (0{,}6367) + x_6 * (2{,}8014) + x_7 * (-0{,}7642) + x_8 * (7{,}3212) + x_9 * (0{,}2957) + x_{10} * (-6{,}5738) + x_{11} * (-1{,}8439) + x_{12} * (-11{,}4480)$$

$z_j$ represent the *output* in the *hidden layer* after being activated using the binary sigmoid activation function. Then $v_{0j}$ is the weight between the input layer and the hidden layer. The mathematical model of the ANN obtained by applying the activation function is as follows:

$$y_k = f(w_{0k} + \sum_{k=1}^{p} z_j \ w_{jk}) \tag{23}$$

$$y_k = f((0{,}3940) + z_1 \ * (-0{,}9091) + z_2 * (-1{,}4965) + z_3 * (1{,}8943) + z_4 * (-0{,}9112) + z_5 * (-1{,}7816) + z_6 * (-1{,}3158) + z_7 * (-0{,}9794)$$

$y_k$ is the output of the forecasted SOI index value for the $k$-th instance after being activated using the binary sigmoid activation function. Meanwhile, $w_{0k}$ is the bias, and $w_{jk}$ is the weight between the *hidden layer* and the *output layer*, serving as an additional parameter that can take both positive and negative values, allowing it to adjust the *output* of the SOI index forecast.

**Future Forecasting**

Figure 4. shows the forecast for the next 12 months using the 12-7-1 architecture. The SOI index is expected to move from negative values at the start of the year to more positive values towards the end of the year.

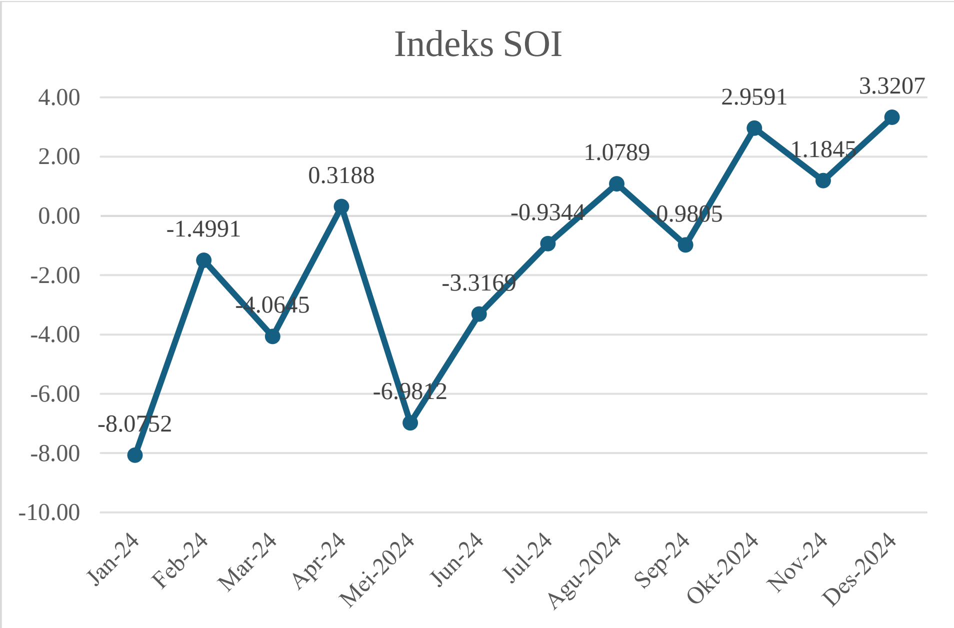

**Figure 4. Forecasting Results for the Next 12 Months**

## 4. DISCUSSIONS

The SOI index exhibits significant fluctuations, making it challenging to predict accurately. It is recommended that future studies using backpropagation should involve multiple trials with different network architectures to identify a model with improved accuracy.

## 5. CONCLUSION

Based on the forecasting for the training data, the model with the 12-7-1 architecture is the best, as it achieved the lowest MSE and MAPE values, with an MSE of 0.0095 and a MAPE of 17.6851. Meanwhile, the highest MSE and MAPE values were achieved by the 12-5-1 architecture model, with an MSE of 0.0109 and a MAPE of 19.2809.

Based on the forecasting for the training data, the MAPE values from the models developed were all below 20%, indicating that the models have good forecasting capabilities.

## 6. ACKNOWLEDGMENTS

We would like to express our gratitude to the Faculty of Mathematics and Natural Sciences, State University of Jakarta, for their support in this research.

## 7. REFERENCES


[1] Aldrian, E., Karmini, M., Budiman, "Adaptasi dan Mitigasi Perubahan Iklim di Indonesia," Pusat Perubahan Iklim dan Kualitas Udara, Kedeputian Bidang Klimatologi, Badan Meteorologi, Klimatologi, dan Geofisika, 2011.

[2] Aldrian, E., "*Forecasting : Spatial Patterns of ENSO Impact on Indonesian Rainfall*," Jurnal Sains & Teknologi Modifikasi Cuaca, 3(1), 5-15, 2002.

[3] Aldrian, E., Meteorologi Laut Indonesia. Jakarta: Puslitbang BMKG, 2008.

[4] Anwar, B., "Penerapan Algoritma Jaringan Syaraf Tiruan *Backpropagation* dalam Memprediksi Tingkat Suku Bunga Bank," Jurnal SAINTIKOM, Vol. 10. No. 2, 20011.

[5] *Bureau of Meteorology Australian Government*, "*Southern Oscillation Index Since 1876 Monthly Data*", [Online]. http://www.bom.gov.au/climate/enso/soi/ [Diakses pada: 9 April 2024].

[6] Brockwell, P. J., & Davis, R. A., *Introduction to Time Series and Forecasting.* New York: Springer, 1996.

[7] Cryer, J. D., & Chan, K.-S., *Time Series Analysis: With Applications to R*. Ed. ke-2. New York: Springer, 2008.

[8] Fausset, L., *Fundamentals of Neural Networks Architectures, Algorithms and Applications*. London: Prantice-Hall, Inc, 1994.

[9] Fitria, W., Pratama, M.S., "Pengaruh Fenomena El Niño 1997 dan La Niña 1999 Terhadap Curah Hujan di Biak," Jurnal Meteorologi dan Geofisika, 14(2), 65-74, 2013.

[10] Hikmah, A., "Peramalan Deret Waktu Menggunakan Autoregressive (AR), Jaringan Syaraf Tiruan Radial Basis *Function* (RBF), dan Hibrid AR-RBF pada Inflasi Indonesia," *Unnes Journal of Mathematics*, 2017.

[11] Junaidi., Mandasari, S., Franciska, Y., Fahmi, A., Rosnelly, R., "Implementasi Jaringan Syaraf Tiruan Menggunakan Algoritma Backpropagation Dalam Meramalkan Kebutuhan *Handsanitizer* Di Pemerintah Kota Medan," *Jurnal of Science and Social Research*, V(3): 671-676, 2022.

[12] Kholis, I., & Rofii, A., "Analisis Variasi Parameter *Backpropagation Artificial Neural Network* Pada Sistem Pengenalan Wajah Berbasis *Principal Component Anaysis*," *Ejournal Kajian Teknik Elektro*, Vol.2 No.1, 2017.

[13] Kovats, R. S., "*El Niño and Human Health. Bulletin of the World Health Organization*," 78(9), 1127-1135, 2000.

[14] Kusumadewi, S., *Membangun Jaringan Syaraf Tiruan menggunakan Matlab dan Excell Link*. Yogyakarta: Penerbit Gava Media, 2004.

[15] Kusumadewi, F., "Peramalan Harga Emas Menggunakan *Feedforward Neural Networks* Dengan Algoritma *Backpropagation*," Skripsi, Universitas Negeri Yogyakarta, 2014.

[16] Lewis, C. D., *International and Business Forecasting Methods*. Butterworths: London, 1982.

[17] Luo, J.J., Zhang, R., Behera, S.K., Masumoto, Y., Jin, F.F., Lukas, R., Yamagata, T., "*Interaction between El Nino and Extreme Indian Ocean Dipole. Journal of Climate*," 23(3), 726-742, 2010.

[18] Makridakis, S., Wheelwright, S.C., & McGee, V.E., *Forecasting : Methods and Applications.* Ed. ke-2. New York: John Wiley & Sons, 1983.

[19] Pradana, D. G., Alghifari, M. L., Juna, M. F., Palaguna, S. D., "Klasifikasi Penyakit Jantung Menggunakan Metode *Artificial Neural Network*," *Indonesian Journal of Data and Science*, Vol 3, No 2, pp 55-60, 2022.

[20] Puspitaningrum, D., *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: Andi, 2006.

[21] Rumelhart, D. E., Hinton, G. E., Williams, R. J., "*Learning representations by back-propagating errors*," *Nature* 323, 533-536, 1986.

[22] Siang, J. J., *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Penerbit Andi, Yogyakarta, 2005.

[23] Tjasyono, B., *Sains Atmosfer*. Jakarta: Puslitbang BMKG, 2008.

[24] Utami, A.W., Jamhari., Hardyastuti, S., "El Nino, La Nina, dan Penawaran Pangan di Jawa, Indonesia," Jurnal Ekonomi Pembangunan, 12(2), 257-271, 2011.

[25] Wang, C., Deser, C., Yu, J.Y., DiNezio, P., Clement, A., "*El Nino and Southern Oscillation* (ENSO): *A Review*," *Coral Reefs of the Eastern Tropical Pacific*, 8, 85-106, 2017.

[26] Xiao, H., Mechoso, C.R., "*Seasonal Cycle– El Niño Relationship: Validation of Hypotheses*," *Journal of the Atmospheric Sciences*, 66(6), 1633-1653, 2009.

[27] Yamagata, T., Behera, S. K., Rao, S. A., Guan, Z., Ashok, K., Saji, H. N., "*The Indian Ocean Dipole: a physical entity*," *CLIVAR Exchanges*, 24, pp 15–18, 20–22, 2002.

[28] Zakir, A., Sulistya, W., Khotimah, M. K., *Perspektif Operasional Cuaca Tropis*. Jakarta: Puslitbang BMKG, 2009.