

Temperature Forecast at Djuanda International Airport using ARIMA, ANN, and Hybrid ARIMA-ANN

Elly Pusporani^{1*}; Fitriana Nur Afifa²; Fidela Sahda Ilona Ramadhina³

¹⁻³Program Studi Statistika, Fakultas Sains dan Teknologi,
Universitas Airlangga, Surabaya, Indonesia, 60115

¹elly.pusporani@fst.unair.ac.id; ²fitriana.nur.afifa-2022@fst.unair.ac.id;

³fidela.sahda.ilona-2022@fst.unair.ac.id

Received: 28th April 2025/ **Revised:** 2nd September 2025/ **Accepted:** 18th September 2025

How to Cite: Pusporani, E., Afifa, F. N., & Ramadhina, F. S. I. (2025). Temperature forecast at Djuanda International Airport using ARIMA, ANN, and hybrid ARIMA-ANN. *ComTech: Computer, Mathematics and Engineering Applications*, 16(2), 139–151. <https://doi.org/10.21512/comtech.v16i2.13219>

Abstract - This research evaluates the performance of Artificial Neural Network (ANN) models in forecasting temperature at Djuanda Airport, comparing them with the traditional Autoregressive Integrated Moving Average (ARIMA) model and a hybrid ARIMA-ANN approach. Although statistical models such as ARIMA are widely applied, their capacity to capture nonlinear dynamics in tropical climate conditions is limited, particularly when the data exhibit irregular fluctuations that linear models cannot adequately represent. Forecasting temperatures in tropical airport settings, which is crucial for flight planning, operational safety, and the reliability of aviation operations, remains relatively underexplored. This gap underscores the importance of alternative modeling techniques that can effectively address nonlinear relationships. Using one year of observed data, the models are evaluated with three accuracy metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). The ANN model achieves the lowest error values (MAE 0.7630, MAPE 2.7067%, RMSE 1.0074) compared to both ARIMA and hybrid approaches. The metrics and the testing graph collectively indicate that ANN has a stronger ability to capture nonlinear temperature dynamics in tropical contexts. Nonetheless, the findings must be interpreted with caution due to the limited dataset and single case study. These limitations highlight the need for extended data and alternative architectures to improve forecasting accuracy and strengthen support for safer aviation operations.

Keywords: Artificial Neural Network (ANN) model, forecasting, weather, Surabaya

I. INTRODUCTION

Weather and climate have a profound impact on human life, particularly in regions with dynamic atmospheric conditions. Weather refers to short-term variations in atmospheric conditions over a small area, whereas climate represents the average weather over a more extended period and a broader region. In recent years, global climate change has made weather patterns increasingly unpredictable, particularly in tropical areas such as Indonesia (Leontinus, 2022). This challenge is further intensified by rapid urbanization and the urban heat island effect, which contribute to higher urban temperatures than those in the surrounding areas. As one of the largest cities in Indonesia, Surabaya faces erratic temperature changes. Additionally, data from the Meteorology, Climatology, and Geophysical Agency (BMKG) indicate an increasing trend in average temperatures in Surabaya over the past decade, which is attributed to the combined effects of human activities and global climate change (Wicaksono, 2024). This condition affects the comfort of community life, infrastructure planning, and energy needs in the region.

In addition to social and environmental impacts, temperature forecasting is crucial for aviation safety and efficiency. Air transportation plays a vital role in supporting Indonesia's economic growth and population mobility. Djuanda International Airport serves as the main gateway to eastern Indonesia, offering both domestic and international flights. Meteorological conditions, particularly air temperature, are critical for aircraft operations. Rising temperatures reduce air density, which in turn decreases aircraft lift and directly affects the takeoff and landing processes (Razzaaq et al., 2024). By combining the maximum air temperature and air pressure, the density altitude

can be calculated, providing essential information for flight operators to ensure safe operations (Kania, 2022).

Despite its importance, forecasting in tropical climates remains challenging because temperature data often exhibit irregular and nonlinear fluctuations. Traditional statistical models, such as the Autoregressive Integrated Moving Average (ARIMA), are designed to capture linear patterns but may not adequately represent the complexity of temperature dynamics in tropical regions. This limitation underscores the need for approaches that can handle both linear and nonlinear behaviors in time-series data. Artificial Neural Networks (ANN) have emerged as a promising alternative owing to their ability to approximate nonlinear functions and capture hidden patterns in time-series data.

Temperature data often fluctuate irregularly, underscoring the need for effective modeling methods. One such approach is the application of ANN, including hybrid models that combine ARIMA with ANN. This hybrid approach is expected to improve prediction accuracy by compensating for the weaknesses of each individual method, thereby creating a more adaptable model for forecasting weather conditions in Surabaya. Previous research demonstrates that an ANN utilizing the Backpropagation Neural Network technique, which incorporates rainfall and wind speed as variables, produces minimal error and is identified as the most effective model (Aruan et al., 2021). In addition, research conducted by Aleksandra has determined daily minimum and maximum temperature forecasts with 96% conformity to actual measurements using an ANN in Poland, which is located in a temperate transitional climate zone with compound climate variability (Bączkiewicz et al., 2021). In contrast, research conducted by Ihsan et al. (2024) employ the Hybrid ARIMA–NN method to forecast rainfall in Makassar City, developing a model that captures linear patterns in time-series data while simultaneously addressing nonlinear patterns, thus generating predictive values that are close to the actual observations with relatively low error rates.

This research focuses on accurately predicting temperatures in the Djuanda International Airport area, which experiences dynamic weather conditions and varying data distributions. By utilizing historical meteorological data from the weather station at Djuanda Airport, we identified the most effective models, including artificial neural networks, ARIMA, and hybrid approaches that combine both ARIMA and artificial neural networks. These models are expected to deliver high-accuracy weather predictions, which are vital for supporting flight operations and ensuring safety at Djuanda Airport.

Furthermore, the solutions developed in this research can be applied to other tropical cities facing similar challenges. By enhancing temperature prediction capabilities, we can implement more targeted mitigation and adaptation measures to address climate change, ultimately promoting environmental

sustainability and improving the overall quality of life within the community. This aligns with the achievement of Sustainable Development Goal (SDG) 13, which focuses on climate change by ensuring that communities can adapt to significant temperature fluctuations, making daily life safer and more comfortable.

II. METHODS

This research employs daily temperature data from Djuanda Airport, Surabaya, obtained from the National Oceanic and Atmospheric Administration (NOAA) database (NOAA, 2024). The dataset spans from December 2023 to December 2024 and consists of approximately 374 daily observations expressed in degrees Celsius. These data serve to identify temperature variation patterns and provide the basis for developing and evaluating forecasting models that capture short-term dynamics relevant to aviation operations and local climate conditions.

An ANN is a structure designed to solve certain types of problems by mimicking the way the human brain solves problems. These structures are designed to address specific issues, particularly those involving high-dimensional, nonlinear data, by emulating the problem-solving methods of the human brain (Schmidgall et al., 2024). The performance of ANNs is highly dependent on the network architecture and the relationship between the neurons. Neurons are arranged in several layers that form the basic structure of the brain. In general, an ANN consists of three main layers: the input layer, which receives data; the hidden layer, which processes information; and the output layer, which produces predictions or decisions based on the processed data, as shown in Figure 1 (Saputra et al., 2023).

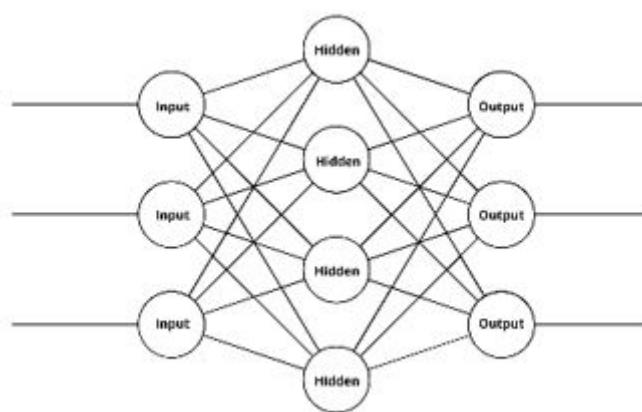


Figure 1 General Model Neural Network

One of the crucial aspects of a Neural Network is its capacity to detect nonlinear patterns in data. The detection of nonlinearity can be evaluated using the Terasvirta Test, which is developed based on neural network models. The Terasvirta Test, first introduced by Terasvirta in 1993, follows a systematic procedure

$$y'_t = \frac{y_t - \min(y_t)}{\max(y_t) - \min(y_t)} (\text{new max}(y_t) - \text{new min}(y_t) + \text{new min}(y_t)) \quad (1)$$

$$\hat{y}_t = \frac{y'_t - \text{new min}(y_t)}{\text{new max}(y_t) - \text{new min}(y_t)} (\max(y_t) - \min(y_t) + \min(y_t)) \quad (2)$$

(Katabba & Estefani, 2023). First, regress y_t on $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ with a constant of 1 and calculate the value of the residual $\hat{u}_t$. Second, regress $\hat{u}_t$ on $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ with a constant of 1 and v additional predictors, including quadratic and cubic terms, which are approximations of the Taylor series. Then, the coefficient of determination R^2 from the regression in (b) is then calculated. The next step is to calculate the test statistic $\chi^2 = nR^2 \sim \chi^2_{(\alpha, v)}$. The null hypothesis is that the model has a linear pattern. In contrast, the alternative hypothesis is that the model has a nonlinear pattern, with the conclusion rejecting H_0 if $\chi^2 > \chi^2_{(\alpha, v)}$ or $p\text{-value} < \alpha$.

In ANN, the data used is normalized data, which involves scaling the data to the range of 0,1 (min) to 1 (max) using the binary sigmoid formula (Chandra et al., 2022), as shown in Equation (1). With y_t is the observation data and y'_t is the normalized data. The estimated result of ANN $\hat{y}_t$ must be converted into the form of data before normalization ($\hat{y}_t$) through the denormalization process (Sakti et al., 2024) shown in Equation (2).

An activation function is used to perform the transformation in an ANN. Activation functions are used to perform transformations in Artificial Neural Networks. Commonly used activation functions include the Sigmoid Function and the Tanh Function, as shown in Equations (3) and (4).

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

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

The activation function used depends on the nature of the data and the type of output desired (Montgomery et al., 2015). ANN models require a training process, in which parameters such as weights and biases are estimated by minimizing the overall squared error. One popular algorithm used to train ANNs is the backpropagation algorithm, which is a form of the steepest descent method. The algorithm for backpropagation consists of four steps, which is the Initialization, Activation, Weight Training, and Iteration steps (Novita & Putri, 2021).

The initialization step provides initial values for the parameters required by the neural network, such as weights and thresholds. The output units ($y_k, k = 1, 2, 3, \dots, m$) receive the target pattern from the learning input pattern and calculate the error information, as shown in Equation (5). Then calculate the weight correction (to correct the value w_{jk}) shown in Equation (6). Also, calculate the bias correction (to correct the value of w_{0k}) as shown in Equation (7). Subsequently,

δ_k is sent to the units in the layer below.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (5)$$

$$\Delta w_{jk} = a \delta_k z_j \quad (6)$$

$$\Delta w_{0k} = a \delta_k \quad (7)$$

The next step is activation, which determine the actual output in the hidden layer and calculate the exact output in the output layer. The hidden units ($Z_j, j = 1, 2, 3, \dots, p$) sum the input deltas of the units in the layer above them, as shown in Equation (8). This value is multiplied by the derivative value of the activation function to calculate the error information displayed in Equation (9). Then calculate the weight correction (to correct the value of v_{ij}) shown in Equation (10). Then also calculate the bias correction (to correct the value of v_{0j}) as shown in Equation (11).

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (8)$$

$$\delta_j = \delta_{in_j} \cdot f'(Z_{in_j}) \quad (9)$$

$$\Delta v_{jk} = a \delta_j x_i \quad (10)$$

$$\Delta v_{0j} = a \delta_j \quad (11)$$

The third step is weight training, which calculate the error gradient at the output and hidden layers. The output units ($Y_k, k = 1, 2, 3, \dots, m$) correct their biases and weights ($j = 1, 2, 3, \dots, p$) using the formula shown in Equation (12). The hidden units ($Z_j, j = 1, 2, 3, \dots, p$) correct their biases and weights ($i = 1, 2, 3, \dots, n$) as shown in Equation (13). Lastly the fourth and final step is iteration. During this step, the process is repeated (iteration) until the minimum error is obtained.

$$w_{jk}(\text{new}) = w_{jk}(\text{old}) + \Delta w_{jk} \quad (12)$$

$$v_{ij}(\text{new}) = v_{ij}(\text{old}) + \Delta v_{ij} \quad (13)$$

The ARIMA model is a time series analysis method used to forecast future data based on past data. Several stages are involved in the ARIMA model, including identifying time series patterns, checking the significance of parameters, the diagnostic test stage (which involves normality and white noise tests), and the prediction or forecasting stage (Pradana et al., 2022). The ARIMA model assumes that time series data are stationary in terms of mean and variance, meaning that the average variance of the data remains constant (Amaly et al., 2022). The identification of

time series data stationarity in variance can be assessed using the Box-Cox Plot, which indicates a rounded value of 1 (Nanlohy & Loklomin, 2023). Concurrently, the stationarity of time series data in the mean can be identified through observations on the Autocorrelation Function (ACF) plot and the Augmented Dickey-Fuller (ADF) test, which determines whether a unit root is present in the model. The ADF test statistic is shown in Equation (14).

$$ADF = \frac{\hat{\beta}_c}{SE(\hat{\beta}_c)} \quad (14)$$

The null hypothesis in the ADF test is that the data are not stationary in the mean, and the alternative hypothesis is that the data are stationary in the mean, with the criteria to reject H_0 if $p\text{-value} < \alpha$. The differencing process is used to make the data stationary in terms of the mean, and the Box-Cox transformation is employed to make the data stationary in terms of the variance (Guobadia & Uadiale, 2024). The Box-Cox transformation is a power transformation on the response, where the transformation is Z_t^λ , where λ , often referred to as the shape parameter, is the parameter that needs to be estimated (Maulana & Hajarisman, 2023). The ARIMA model aims to determine the statistical relationship between the variable to be predicted and its historical values (Geurts et al., 1970). The ARIMA (p, d, q) model has the following general form, as shown in Equation (15).

$$\theta_p(B)(1-B)^d y_t = \theta_q(B) a_t \quad (15)$$

With B as the backshift operator, a_t is the error at time t , p is the AR order, q is the MA order, and d is the number of differencing processes. In ARIMA, there is a diagnostic test that consists of a white noise test and a normality test on the residual model. The white noise test uses the L-Jung Box test to determine whether the model residuals follow a white noise distribution. The test statistic Q is shown in Equation (16).

$$Q = n(n+2) \sum_{k=1}^K \frac{\rho_k^2}{n-k} \quad (16)$$

Where n is the amount of data, k is the lag, K is the maximum lag, and ρ_k is the lag k autocorrelation function value. The null hypothesis in the white noise test is that the residuals follow a white noise distribution, and the alternative hypothesis is that the residuals do not follow a white noise distribution. If the $Q > \chi^2_{\left(\frac{\alpha}{df}, k-p-q\right)}$ or $p\text{-value} < \alpha$, then the null

hypothesis is rejected (Katabba & Estefani, 2023). The normality test uses the Kolmogorov-Smirnov test to determine whether the residuals have a normal distribution. The test statistic D is shown in Equation (17).

$$D = \text{Sup}_x |F_n(x) - F_0(x)| \quad (17)$$

D is the test statistic, $F_n(x)$ is the cumulative value based on the data, and $F_0(x)$ is the cumulative probability value under $H_0 P(Z < Zt)$. The null hypothesis in the normality test is that the residual data are normally distributed, and the alternative hypothesis is that the residual data are not normally distributed. If $D > D\alpha$, with $D\alpha$ being the critical value of the Kolmogorov-Smirnov test or $p\text{-value} < \alpha$, then the decision is to reject H_0 with the conclusion that the residual data is not normally distributed (Katabba & Estefani, 2023).

In addition to the fulfilled diagnostic test, the selection of a good ARIMA model is based on a low value of the Akaike Information Criterion (AIC). AIC is a statistical criterion that balances model fit and complexity. AIC specifically emphasizes the trade-off between goodness of fit and model complexity to achieve optimal estimation (Pangaribuan et al., 2023). The AIC formula is shown in Equation (18), where $\hat{\Sigma}_p$ is the residual covariance estimation matrix for the AR(p) model, T is the number of residuals, and K is the number of variables (Muslihin & Ruchjana, 2023).

$$AIC(p) = \log|\hat{\Sigma}_p| + \frac{2}{T} pK^2 \quad (18)$$

Zhang, as cited in Laily et al. (2021), developed a hybrid model to enhance forecasting accuracy by considering several key factors. First, selecting a forecasting method that aligns with the specific characteristics of the dataset is often a challenging task. Second, real-world time series data are not purely linear or nonlinear but are a combination of both. Third, various studies have indicated that no single forecasting method is universally effective under all conditions. Therefore, hybrid models integrate both linear and nonlinear components to improve prediction accuracy. One widely used hybrid approach is the Hybrid ARIMA-NN, which combines the ARIMA model to handle stochastic and nonlinear patterns (Ihsan et al., 2024). In general, the Hybrid ARIMA-NN model is presented in Equation (19), where L_t represents the linear component derived from the ARIMA model.

$$y_t = L_t + N_t \quad (19)$$

Meanwhile, N_t is a nonlinear component obtained by modeling the residuals of these two components. These two components are estimated from the data. First, the ARIMA model is used to model the linear component, so the residuals from the linear model will only contain nonlinear relationships. Suppose e_t is the residual at time t from the linear model shown in Equation (20), where $\hat{L}_t$ is the estimated value at time t obtained from the relationship.

$$a_t = y_t - \hat{L}_t \quad (20)$$

Residuals play a crucial role in determining the adequacy of a linear model. A linear model is considered inadequate if a linear correlation structure persists in the residuals. However, residual analysis is unable to detect nonlinear patterns in the data. Currently, there is no general diagnostic test for nonlinear autocorrelation relationships. Therefore, even if a model has passed the diagnostic test, it may still be inadequate if the nonlinear relationship has not been adequately modeled (Ihsan et al., 2024). The presence of significant nonlinear patterns in the residuals indicates the limitations of the ARIMA model. By modeling the residuals using an ANN, nonlinear relationships can be found. With n nodes, the ANN model for the residuals is given by Equation (21), where f is a nonlinear function determined by the neural network, and ε_t represents a random error.

$$\varepsilon_t = f(a_{t-1}, a_{t-2}, \dots, a_{t-n}) + \varepsilon_t \quad (21)$$

So, if the selection of function f is not appropriate, the error (ε_t) will be random. Therefore, identifying the right model is crucial. Suppose the approximation of the above equation is denoted as $\hat{N}_t$, then the joint approximation can be shown in Equation (22).

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \quad (22)$$

In selecting the best model, performance evaluation is crucial to determine the extent to which the model can provide accurate predictions. Various metrics are used to measure the error rate and accuracy of the model. One commonly used approach is to calculate the error value. Data accuracy parameters in detecting errors include Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) (Hilal et al., 2024).

$$MAE = \frac{\sum_{t=1}^n |y_t - \hat{y}_t|}{n} \quad (23)$$

$$MAPE = \left(\frac{100}{n} \right) \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (24)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (25)$$

With y_t is the actual data at time t , $\hat{y}_t$ is the predicted data at time t , and n is the number of data observations. A model has excellent criteria if the MAPE value is $< 10\%$ and good if the MAPE is between 10% - 20% . The following outlines the stages of the analysis method used in this research.

The ANN analysis begins with examining the descriptive statistics to understand the basic characteristics of the temperature data. Next, the Terasvirta test is applied to check whether the data

follow a linear or nonlinear pattern. The data are then split into training and testing sets in a 90:10 ratio. A Partial Autocorrelation Function (PACF) plot is used to identify the best lag value k . Afterward, the data are normalized using the min-max method to improve the performance of the neural network. The normalized data are combined with lagged data by shifting the time base back by k to prepare the input features. A neural network model is then defined, including its parameters. A trial-and-error process is conducted to test different numbers of hidden nodes using the sigmoid activation function. Each configuration is evaluated by predicting the testing data, and the best number of nodes is selected based on the lowest MAE, MAPE, and RMSE values of the denormalized results. The structure of the final ANN model is visualized, and equations are developed to describe how input data move through the layers to produce the output.

The ARIMA analysis begins by verifying whether the data is stationary. If the variance is not stable, a Box-Cox transformation is applied. If the mean is still not stable after the transformation, differencing is used to stabilize the data. Once the data is stationary, ACF and PACF plots are used to identify the best ARIMA model. After selecting a model, the significance of its parameters is tested. Then, the residuals are checked to ensure they behave like white noise and are typically normally distributed. The best model is chosen based on the principle of parsimony, which means selecting the simplest model that fits well. Finally, the model is used to make forecasts, and its accuracy is measured using MAE, MAPE, and RMSE based on actual data.

The hybrid ARIMA-NN process starts by calculating residuals, which are the differences between actual data and the ARIMA model's predicted values. These residuals are used as training data. ACF and PACF plots of the residuals are then used to choose input variables based on significant lags. The residual data is normalized using min-max normalization, and lagged versions are created by shifting the time base back by kk . Next, a neural network model is defined with its parameters. This network is used to predict the residuals for the testing data. The predicted residuals are added to the ARIMA forecasts to get the final predictions. The best model is chosen by comparing MAE, MAPE, and RMSE values for different neural network configurations. Finally, the structure and equation of the hybrid ARIMA-NN model are created to describe how the final predictions are made.

III. RESULTS AND DISCUSSIONS

Descriptive statistics are applied to identify the general characteristics of the data. In this research, descriptive analysis is performed using time series plots to show the data patterns. The data visualization is presented in Figure 2.

Based on Figure 2, the air temperature at Djuanda Airport, Surabaya, has exhibited substantial

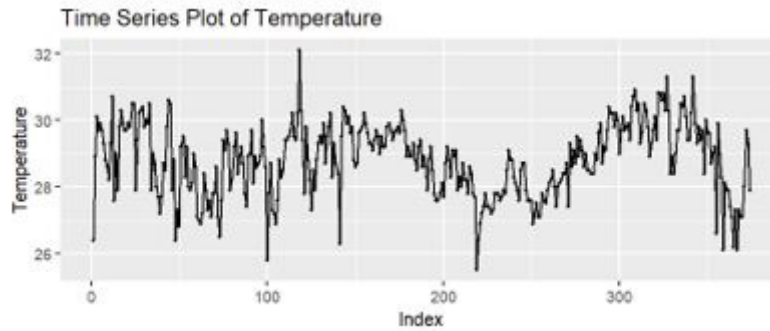


Figure 2 Time Series Plot

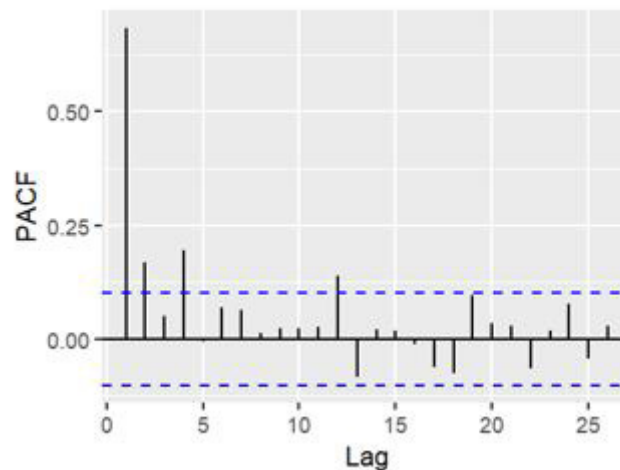


Figure 3 PACF Plot

fluctuations over the past year. The temperature generally ranged between 25.50°C and 32.10°C, with a mean of 28.82°C and a variance of 1.09°C. Irregular extreme increases and decreases indicate considerable variability in temperature dynamics throughout the year.

For the ANN model, the selection of input lags is chosen based on the PACF. PACF is widely used to detect the direct relationship between a series and its lagged values after controlling for the intermediate lags, which makes it appropriate for identifying informative predictors. PACF plots are depicted in Figure 3.

Based on Figure 3, and using 95% confidence bounds as the significance criterion, the PACF shows notable spikes at lags 1, 2, 4, and 12. These lags are incorporated as input features for the ANN to ensure that the model captures statistically relevant temporal dependencies rather than relying on arbitrary lag choices. Before implementing the model, the Terasvirta test is applied to examine whether the data follows a linear or nonlinear structure. The results of the Terasvirta test are presented in Table 1 and provide important evidence for selecting the appropriate modeling approach.

Table 1 Terasvirta Test

	Prob	Conclusion
χ^2	12.46	Reject H_0 , data is nonlinear
p - value	0.002	

According to Table 1, the data exhibit a nonlinear pattern, which emphasizes the necessity of using an activation function in the modeling process. Prior to implementing the artificial neural network (ANN) model, the data undergoes min-max normalization, as described by the formula in Equation (1). Table 2 presents the values before and after the min-max normalization process.

Table 2 Normalization result

Index	Actual	Normalization
1	26.4	0.2091
2	28.9	0.5121
...	...	...
373	29.3	0.5606
374	27.9	0.3909

Table 3 displays the descriptive statistics for both the raw and normalized data. The comparison shows that normalization changes the mean, variance, minimum, and maximum values. These results confirm that normalization adjusts the data scale while preserving the number of observations.

Table 3 Descriptive Statistics

	N	Mean	Var	Min	Max
Raw	374	28.82	1.09	25.50	32.10
Normalized	374	0.5019	0.0176	0.1	0.9

The algorithm employed in this research is backpropagation, which is widely used for training ANN due to its ability to minimize error through iterative weight adjustments. To construct the ANN model, several parameters are applied, ensuring that the network architecture is tailored to the characteristics of the dataset. These parameters are systematically presented in Table 4, providing a clear overview of the model configuration used in the analysis.

The number of hidden neurons is selected within the range of 2 to 7. A relatively small number of nodes is chosen due to the limited dataset size, as a simpler architecture is more appropriate for mitigating overfitting and maintaining the model's generalization capability. The ANN model output, generated using the parameters presented in Table 4, is subsequently converted back to the original units through denormalization using Equation (2). Following this process, MAE, MAPE, and RMSE are calculated based on the output and testing data for each trial node, with the results presented in Table 5.

Table 4 ANN Parameter

Parameter	Value	Description
Input layer	4 (lag1,2,4,12)	Based on PACF
Hidden layer	2,3,4,5,6,7	Trial Error
Output layer	1	Predicted value
Learning rate	0.001	
Activation function	Sigmoid, Tanh	Trial error

Table 5 MAE, MAPE dan RMSE Value

	Nodes	MAE	MAPE %	RMSE
Sigmoid	2	0.7486	2.6585	1.0049
	3	0.7031	2.4953	0.9684
	4	0.7260	2.5770	0.9887
	5	0.7472	2.6517	1.0160
	6	0.7405	2.6262	1.0032
Tanh	7	0.7339	2.6035	0.9932
	2	0.7539	2.6751	1.0077
	3	0.7496	2.6623	1.0057
	4	0.7496	2.6600	1.0016
	5	0.7672	2.7251	1.0498
	6	0.7550	2.6782	1.0036
	7	0.7381	2.6196	1.0101

Based on Table 5, the ANN model with a sigmoid activation function and three nodes shows the best performance, as indicated by the lowest MAE, MAPE, and RMSE values. The resulting optimal network architecture is 4-3-1. A graphical representation of this architecture is provided in Figure 4. Based on Figure 4, the model equation is shown in Equation (26).

Next, $\hat{y}^j$ will be denormalized using Equation (3). The first step in ARIMA modeling is to examine the stationarity of the time series data. The ACF plot shown in Figure 5 displays a gradual dying-down pattern. This pattern indicates that the data is not yet stationary in the mean.

However, the data maintains a constant variance, as indicated by the rounded values on the control charts, which exceed 1. This result suggests that while the series requires differencing to achieve mean stationarity, the variance remains stable and does not require transformation. The corresponding visualization is presented in Figure 6, which illustrates the stability of variance across the observed data.

The differencing process is essential for achieving mean stationarity. In this research, first-order differencing is applied to obtain mean stationarity. This conclusion is supported by the ACF and PACF plots in

$$\begin{aligned}
 f_j^{(1)} &= \sigma \left(\sum_{i=1}^4 W_{ij}^{(1)} x_i + b_j^{(1)} \right); j = 1, 2, 3 \\
 f_1^{(1)} &= \sigma \left((y'_{t-1}(0.17612) + y'_{t-2}(1.01635) + y'_{t-4}(-2.39467) + y'_{t-12}(0.34512)) - 1.29603 \right) \\
 f_2^{(1)} &= \sigma \left((y'_{t-1}(-0.80982) + y'_{t-2}(-0.6079) + y'_{t-4}(-0.57186) + y'_{t-12}(1.02736)) + 0.52203 \right) \\
 f_3^{(1)} &= \sigma \left((y'_{t-1}(-1.2287) + y'_{t-2}(-0.88003) + y'_{t-4}(0.01151) + y'_{t-12}(0.80882)) - 0.5485 \right) \\
 \hat{y}^j &= \left(\sum_{k=1}^3 W_k^{(output)} f_k^{(1)} + b^{(output)} \right) \\
 \hat{y}^j &= \left(-0.18191f_1^{(1)} - 1.06285f_2^{(1)} - 0.79652f_3^{(1)} \right) + 1.00944
 \end{aligned} \tag{26}$$

Figure 7, as well as by the results of the Augmented Dickey-Fuller (ADF) test, which yield a p-value of 0.010, as presented in Table 6.

Table 6 ADF Test

	Prob	Conclusion
χ^2	-9.907	Reject , H_0
p - value	0.010	Data is stationary in mean

After achieving stationarity in the mean, ARIMA modeling is conducted with a differencing order of $d = 1$. The estimation results include the significance of the parameters, the white noise test, and the AIC values. These results for each ARIMA model are summarized in Table 7.

Based on Table 7, the best-fitting model is ARIMA (1,1,1). Although all models indicate a non-normal distribution, the Central Limit Theorem (CLT) suggests that with a sufficiently large sample size, the distribution can be treated as approximately normal. The parameter estimates of the selected model are provided in Table 8. Based on Table 8, the equation of the ARIMA (1,1,1) model is shown in Equation (27) where y_t is time series data and a_t is the error.

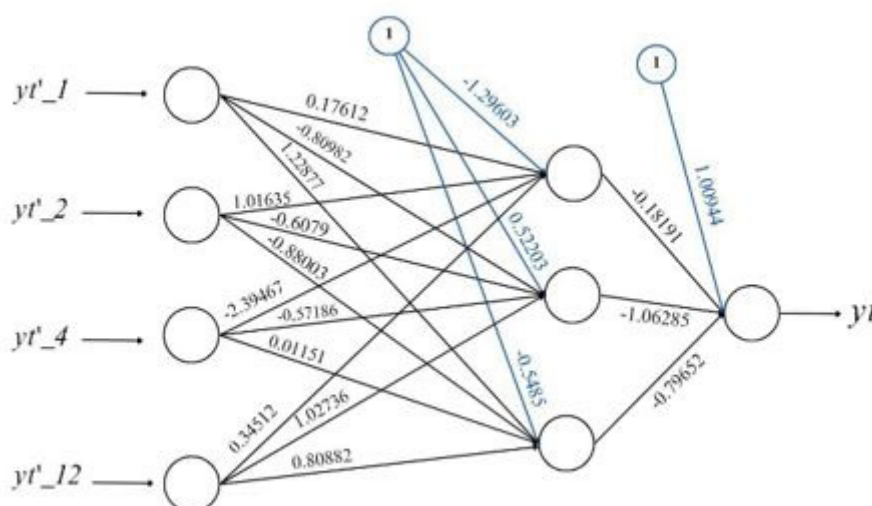


Figure 4 Model Architecture

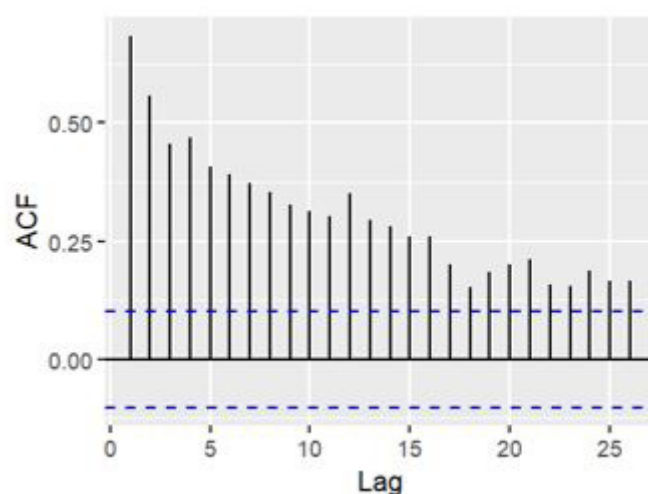


Figure 5 ACF Plot

Table 7 Model Selection

ARIMA (p,d,q)	Parameter	White Noise	AIC
(1,1,0)	Significant	√	754.57
(1,1,1)	Significant	√	718.66
(0,1,1)	Significant	√	744.30
(2,1,1)	Insignificant	√	720.39
(1,1,2)	Insignificant	√	720.47

Table 8 Parameter Coefficient

Parameter	Coefficient	Std. Error	p-value
AR1	0.5128	0.0636	0.0000
MA1	-0.9042	0.0305	0.0000

$$(1 - \phi_1 B)(1 - B)y_t = (1 + \theta_1 B)a_t$$

$$(1 - 0.5128B)(y_t - y_{t-1}) = (1 - 0.9042B)a_t$$

$$y_t = 1.5128y_{t-1} - 0.5128y_{t-2} + a_t - 0.9042a_{t-1} \quad (27)$$

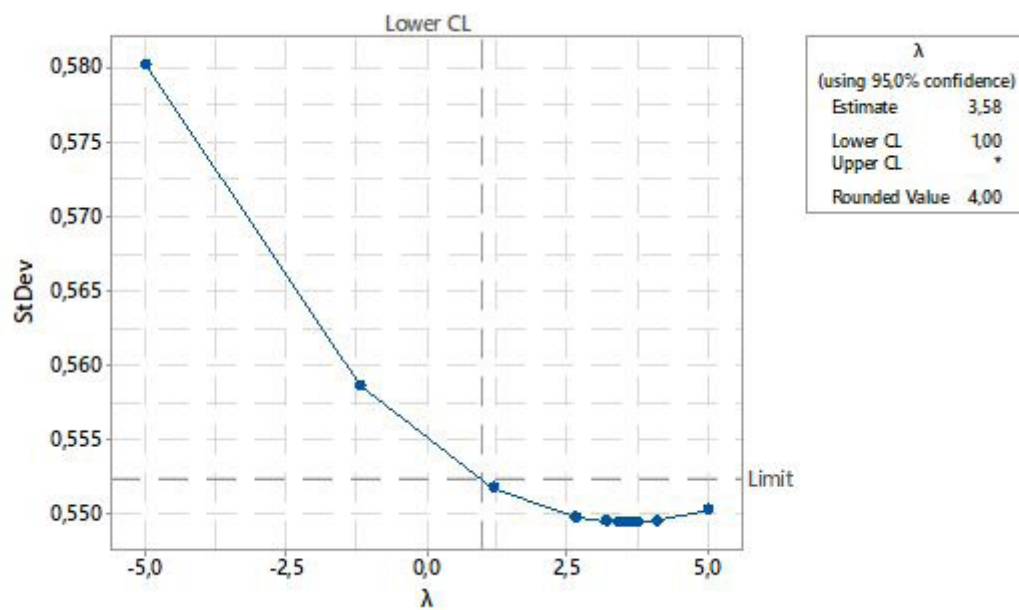


Figure 6 Control Chart

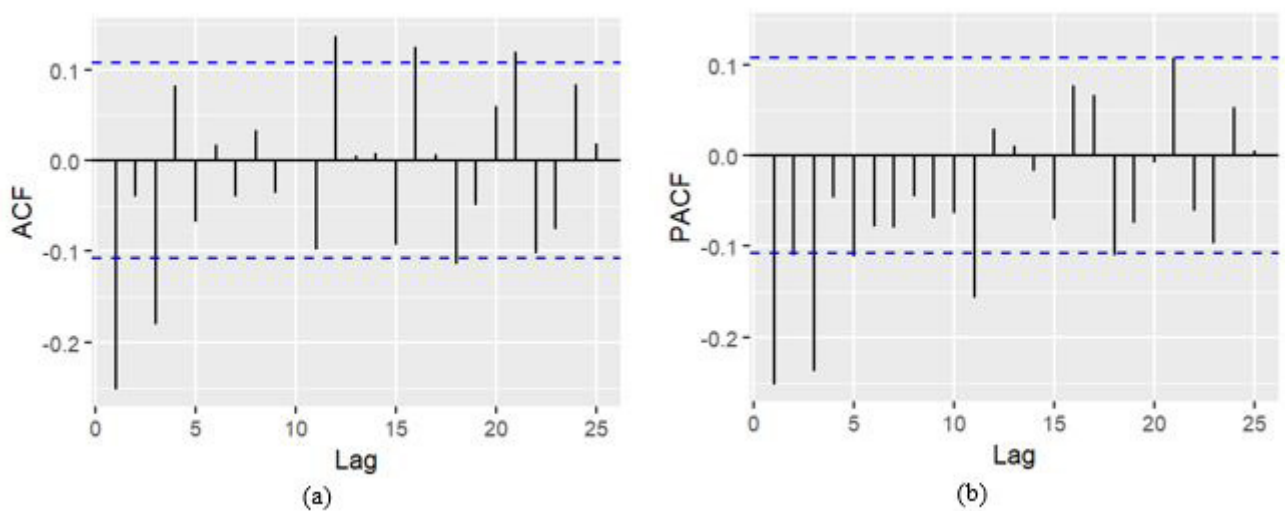


Figure 7 (a) ACF and (b) PACF Plot after Differencing

The ARIMA (1,1,1) model achieves an MAE of 1.5249, an MAPE of 5.5389%, and an RMSE of 1.9338, indicating excellent prediction performance. The first step in constructing the hybrid ARIMA-ANN model is to examine the linearity of the residuals from the ARIMA (1,1,1) model. This step ensures that the residuals contain nonlinear patterns that the ANN can capture. The test results are presented in Table 10.

Table 9 Terasvirta Test

	Prob	Conclusion
χ^2	8.778	Reject H_0
p - value	0.012	

The residuals of the ARIMA (1,1,1) model in Table 9 show a nonlinear pattern. This indicates the need for an activation function, such as the one in the ANN model, to capture the underlying data structure. To determine suitable inputs, the PACF of the residual series is examined, as it detects remaining autocorrelations not explained by the linear ARIMA structure.

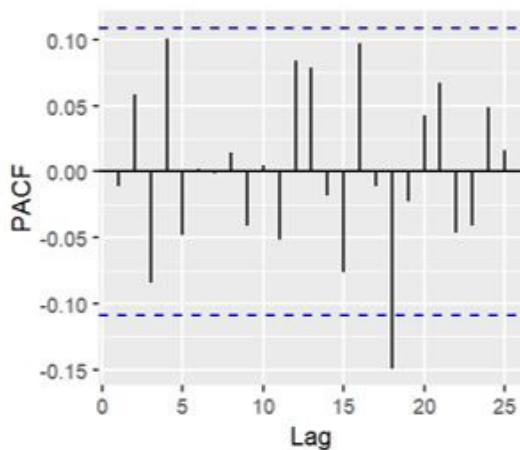


Figure 8 PACF Plot of Residual ARIMA (1,1,1)

The PACF plot in Figure 8 shows a single significant spike at lag 18, indicating residual dependence at this lag. Accordingly, lag 18 is selected as the sole input feature for the ANN, ensuring that the hybrid model focuses on statistically significant nonlinear dependencies. The algorithm and parameters for ARIMA (1,1,1) residual modeling are summarized in Table 10.

Table 10 ANN for Residual Parameter

Parameter	Value	Description
Input layer	1	Based on the PACF lag
Hidden layer	2,3,4,5,6,7	Trial Error
Output layer	1	Predicted value
Learning rate	0.001	
Activation function	Sigmoid, Tanh	Trial error

The best model is determined by comparing the MAE, MAPE, and RMSE values. These error measures provide an objective basis for evaluating the model's accuracy. The results of this comparison are presented in Table 11.

Table 11 MAE, MAPE dan RMSE Value

	Nodes	MAE	MAPE %	RMSE
Sigmoid	2	1.5758	5.7187	1.9822
	3	1.5525	5.6365	1.9594
	4	1.5577	5.6551	1.9651
	5	1.5573	5.6528	1.9619
	6	1.5722	5.7064	1.9802
	7	1.5852	5.7520	1.9928
Tanh	2	1.5630	5.6735	1.9697
	3	1.5747	5.7127	1.9747
	4	1.5757	5.7196	1.9881
	5	1.5666	5.6856	1.9709
	6	1.5657	5.6851	1.9804
	7	1.5790	5.7290	1.9824

Based on Table 11, the best model is achieved using a sigmoid activation function with three nodes. The resulting optimal network architecture is 1-3-1. A visualization of this architecture is provided in Figure 9. The model equation based on Figure 8 for ANN residuals can be written as follows in Equation (28).

$$\begin{aligned}
 f_1^{(1)} &= \sigma(a'_{t-18}(0.29399) + 0.82181) \\
 f_2^{(1)} &= \sigma(a'_{t-18}(0.1765) + 1.04074) \\
 f_3^{(1)} &= \sigma(a'_{t-18}(1.84388) - 0.94403) \\
 \hat{N}_t &= \left(\sum_{k=1}^3 W_k^{(output)} f_k^{(1)} + b^{(output)} \right) \\
 \hat{N}_t &= (-1.56435 f_1^{(1)} + 1.08954 f_2^{(1)} - 0.09995 f_3^{(1)}) + 0.90012
 \end{aligned} \tag{28}$$

The combination of Equations (27) and (28) will form an ARIMA(1,1,1)-ANN(1-3-1) model, which can be mathematically written as follows in Equation (29). The best model is determined by evaluating the MAE, MAPE, and RMSE values of each method. These performance measures are used to compare and assess model accuracy. The detailed results are presented in Table 12.

$$\begin{aligned}
 y_t &= (1.5128y_{t-1} - 0.5128y_{t-2} + a_t - 0.9042a_{t-1} + \\
 &\quad ((-1.56435 f_1^{(1)} + 1.08954 f_2^{(1)} - 0.09995 f_3^{(1)}) + 0.90012)
 \end{aligned} \tag{29}$$

Table 12 MAE, MAPE, and RMSE Value

Method	MAE	MAPE %	RMSE
NN	0.7630	2.7067	1.0074
ARIMA	1.5249	5.5389	1.9338
ARIMA-NN	1.5525	5.6365	1.9594

According to Table 12, the single ANN method shows the best performance, yielding lower MAE, MAPE, and RMSE values than the other methods. This suggests that the ANN model is more effective at capturing data patterns. Figure 10 presents a comparison of the ANN, ARIMA, and ARIMA-ANN models based on both training and testing data.

Based on Figure 10, all methods capture the training data pattern reasonably well; however, ANN shows superior performance on the testing data. The fundamental difference in model structures can explain

this. ARIMA, as a linear model, relies on past values and error terms, making it less effective in capturing irregular fluctuations or sudden shifts in temperature data. Even when combined in the hybrid ARIMA-ANN, the residual component still inherits ARIMA's linearity, limiting its capacity to fully represent complex dynamics. In contrast, an ANN is a data-driven nonlinear model that can approximate highly complex relationships without prior assumptions about the data structure. Its hidden layers and activation functions enable the network to adapt flexibly to abrupt variations and nonlinear dependencies, which are characteristic of meteorological data. This flexibility explains why ANN achieved lower error values compared to ARIMA and hybrid ARIMA-ANN in this research. Similar conclusions have been drawn in prior research on gold prices (Ulandari, 2023), rainfall (Ihsan et al., 2024), and agricultural commodities (Palupi et al., 2023), further reinforcing that ANN is well-suited for forecasting tasks dominated by nonlinear patterns.

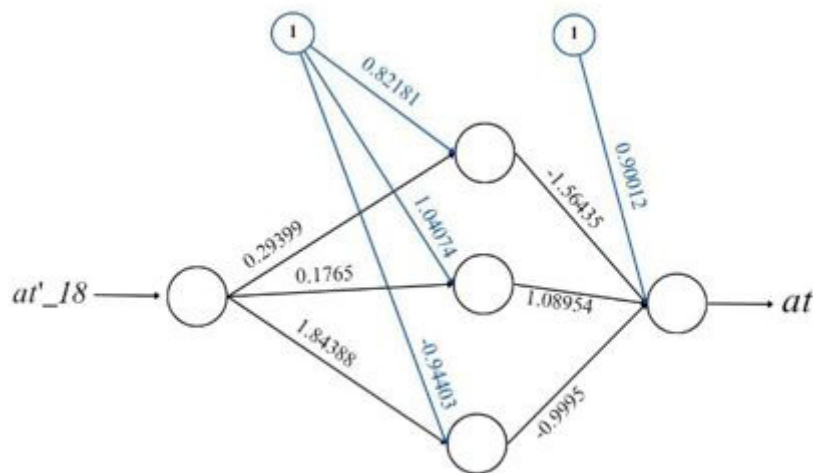


Figure 9 Model Architecture

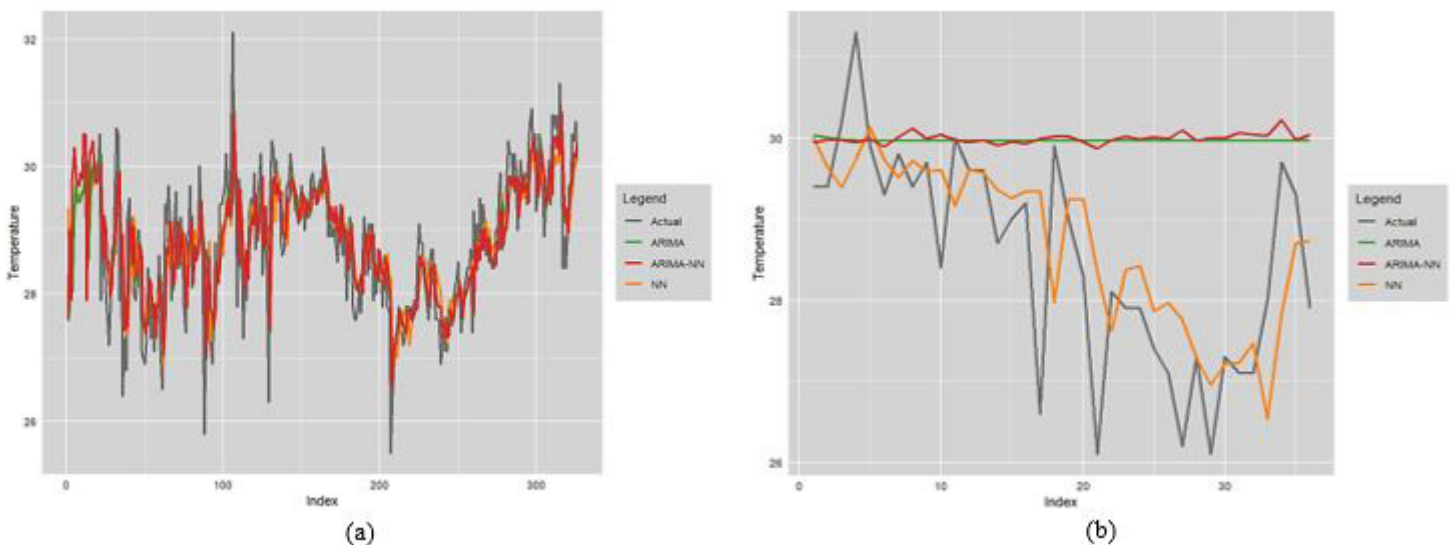


Figure 10 Comparison Fitted Plot on Data (a) Training (b) Testing

IV. CONCLUSIONS

The research findings demonstrate that the ANN model effectively captures the nonlinear patterns of temperature data at Djuanda Airport during both training and testing phases. Compared to the ARIMA and hybrid ARIMA-ANN models, the ANN model achieves lower MAE, MAPE, and RMSE values, indicating relatively lower prediction errors. Although the hybrid ARIMA-ANN model is designed to integrate linear and nonlinear characteristics for improved accuracy, in this case, the pure ANN approach shows greater adaptability to the data dynamics.

These results are consistent with previous evidence on the strengths of ANN in handling strongly nonlinear time series. However, it should be interpreted with caution, given the research's limitations, including its focus on a single location and the use of a fixed ANN configuration. From a practical perspective, more accurate temperature forecasts can support aviation stakeholders by enhancing flight scheduling, improving operational safety margins, and contributing to more efficient energy management at airport facilities. Future works should optimize ANN architectures and extend the analysis to multiple sites, longer horizons, and alternative models (e.g., TSR, Exponential Smoothing, SVR). Additionally, future research should explore higher-frequency data, such as hourly observations, to enhance accuracy and generalizability for aviation operations.

AUTHOR CONTRIBUTIONS

Conceived and designed the analysis, E. P.; Collected the data, F. S. I. R.; Contributed data or analysis tools, E. P., F. N. A.; Performed the analysis, E. P., F. N. A., F. S. I. R.; Wrote the paper, F. N. A., F. S. I. R.

DATA AVAILABILITY

The data that support the findings of this study are openly available in the NOAA database at <https://www.ncei.noaa.gov/cdo-web/>

REFERENCES

- Amaly, M. H., Hirzi, R. H., & Basirun. (2022). Perbandingan metode ANN backpropagation dan ARMA untuk peramalan inflasi di Indonesia. *Jambura Journal of Probability and Statistics*, 3(2), 61–70. <https://doi.org/https://doi.org/10.34312/jjps.v1i1.15440>
- Aruan, N. M., Panggabean, D. A. H., & Sihombing, F. M. (2021). Prediksi tinggi curah hujan dan kecepatan angin berdasarkan data cuaca dengan penerapan algoritma Artificial Neural Network (ANN). *SEMINASTIKA*, 3(1), 1–7. <https://doi.org/10.47002/seminastika.v3i1.237>
- Bączkiewicz, A., Wątróbski, J., Sałabun, W., & Kołodziejczyk, J. (2021). An ANN model trained on regional data in the prediction of particular weather conditions. *Applied Sciences (Switzerland)*, 11(11), 4757. <https://doi.org/10.3390/app11114757>
- Chandra, R., Chaudhary, K., & Kumar, A. (2022). Comparison of data normalization for wine classification using K-NN algorithm. *IJIS: International Journal of Informatics and Information Systems*, 5(4), 175–180. <https://doi.org/10.47738/ijis.v5i4.145>
- Geurts, M., Box, G. E. P., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control Holden-Day series in time series analysis and digital processing. In *Holden Day*. <https://doi.org/10.2307/3150485>
- Guobadia, E. K., & Uadiale, K. K. (2024). Effect of Box-Cox transformation on a k-th weighted moving average processes for time series. *African Multidisciplinary Journal of Sciences and Artificial Intelligence*, 1(1), 655–668. <https://doi.org/https://doi.org/10.58578/AMJSAI.v1i1.3755>
- Hilal, Y. N., Nainggolan, G. D. A., Syahputri, S. H., & Kartiasih, F. (2024). Comparison of ARIMA and LSTM methods in predicting Jakarta sea level. *Jurnal Ilmu Dan Teknologi Kelautan Tropis*, 16(2), 163–178. <https://doi.org/10.29244/jitkt.v16i2.52818>
- Ihsan, H., Irwan, I., & Nensi, A. I. E. (2024). Implementation of backpropagation and hybrid ARIMA-NN methods in predicting accuracy levels of rainfall in Makassar city. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, 18(4), 2435–2448. <https://doi.org/10.30598/barekengvol18iss4pp2435-2448>
- Kania, D. D. (2022). Sektor penerbangan global dalam isu perubahan iklim : Dampak dan mitigasinya. *Jurnal Manajemen Transportasi & Logistik (JMTRANSLOG)*, 8(2), 133. <https://doi.org/10.54324/j.mtl.v8i2.713>
- Katabba, Y. I., & Estefani, K. (2023). Penerapan model Self-Exciting Threshold Autoregressive (SETAR) nonlinear dalam memodelkan data harga minyak sawit (FCPOc1). *Mathematical Sciences and Applications Journal*, 4(1), 33–39. <https://doi.org/10.22437/msa.v4i1.28292>
- Laily, V. O. N., Suhartono, Pusporani, E., & Atok, R. M. (2021). A novel hybrid MGSTAR-RNN model for forecasting spatio-temporal data. *Journal of Physics: Conference Series*, 1752(1), 012011. <https://doi.org/10.1088/1742-6596/1752/1/012011>
- Leontinus, G. (2022). Program dalam pelaksanaan tujuan pembangunan berkelanjutan (SDGs) dalam hal masalah perubahan iklim di Indonesia. *Jurnal Samudra Geografi*, 5(1), 43–52. <https://doi.org/10.33059/jsg.v5i1.4652>
- Maulana, C., & Hajarisman, N. (2023). Penerapan transformasi Box Cox untuk mengatasi masalah ketidakstasioneran dan pola periodik dalam data deret waktu pada ekspor bidang pertanian di Indonesia. *Bandung Conference Series: Statistics*, 3(2), 763–772. <https://doi.org/10.29313/bcss.v3i2.9371>
- Montgomery, D. C., Jennings, C. L., & Murat, K. (2015). *Introduction to time series analysis and forecasting*.

John Wiley & Sons.

- Muslihin, K. R. A., & Ruchjana, B. N. (2023). Model Autoregressive Moving Average (ARMA) untuk peramalan tingkat inflasi di Indonesia. *Limits: Journal of Mathematics and Its Applications*, 20(2), 209. <https://doi.org/10.12962/limits.v20i2.15098>
- Nanlohy, Y. W. A., & Loklomin, S. B. (2023). Model Autoregressive Integrated Moving Average (ARIMA) untuk meramalkan inflasi Indonesia. *VARIANCE: Journal of Statistics and Its Applications*, 5(2), 201–208. <https://doi.org/10.30598/variancevol5iss2page201-208>
- NOAA. (2024). *Climate data records*. National Oceanic and Atmospheric Administration. <https://www.noaa.gov/cdo-web/>
- Novita, R., & Putri, A. (2021). Analisis algoritma backpropagation Neural Network dalam peramalan jumlah benih ikan. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 7(2), 201–207.
- Palupi, S. P., Sadik, K., & Afendi, F. M. (2023). *Perbandingan performa metode hybrid ARIMA-SVM dan ARIMA-NNAR pada peramalan data deret waktu* [IPB University]. <http://repository.ipb.ac.id/handle/123456789/123662>
- Pangaribuan, J. J., Fanny, F., Barus, O. P., & Romindo, R. (2023). Prediksi penjualan bisnis rumah properti dengan menggunakan metode Autoregressive Integrated Moving Average (ARIMA). *Jurnal Sistem Informasi Bisnis*, 13(2), 154–161. <https://doi.org/10.21456/vol13iss2pp154-161>
- Pradana, D. A. P., Mahananto, F., & Djunaidy, A. (2022). Sistem peramalan menggunakan Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX) untuk harga minyak sawit Indonesia. *Jurnal Teknik ITS*, 11(2). <https://doi.org/10.12962/j23373539.v11i2.86373>
- Razzaaq, M., Lubis, L. H., & Sirait, R. (2024). Studi pengaruh suhu dan tekanan udara terhadap gaya angkat pesawat tahun 2014-2021 di Bandara Internasional Kualanamu Deli Serdang. *Relativitas: Jurnal Riset Inovasi Pembelajaran Fisika*, 7(2), 102–111. <https://doi.org/10.29103/relativitas.v7i2.18923>
- Sakti, A. I., Saputra, L., Suhendra, H., Halim, N., Alviari, I., Ilham, M. R. N., Putri, M. H. N., & Dalimunthe, D. Y. (2024). Implementasi Artificial Neural Network (ANN) dalam memprediksi nilai tukar Rupiah terhadap Dolar Amerika. *Euler: Jurnal Ilmiah Matematika, Sains Dan Teknologi*, 12(2), 124–130. <https://doi.org/10.37905/euler.v12i2.26654>
- Saputra, R., Sunardiyo, S., Nugroho, A., & Subiyanto. (2023). Analisis arsitektur jaringan syaraf tiruan-multilayer perceptron untuk efektivitas estimasi beban energi listrik PT. PLN (Persero) UP3 Salatiga. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(3), 664. <https://doi.org/10.26760/elkomika.v11i3.664>
- Schmidgall, S., Ziaei, R., Achterberg, J., Kirsch, L., Hajiseyedrazi, S. P., & Eshraghian, J. (2024). Brain-inspired learning in artificial neural networks: A review. *APL Machine Learning*, 2(2), 1–13. <https://doi.org/10.1063/5.0186054>
- Ulandari, R. (2023). *Perbandingan metode Autoregressive Integrated Moving Average dengan metode hybrid Autoregressive Integrated Moving Average Jaringan Saraf Tiruan pada peramalan harga emas*. Universitas Hasanuddin.
- Wicaksono, A. (2024). *Anomali suhu udara bulan Oktober 2024*. Badan Meteorologi, Klimatologi, Dan Geofisika. <https://www.bmkg.go.id/iklim/anomali-suhu-udara/anomali-suhu-udara-bulan-oktober-2024>