Volume 10 Number 3, August 2025, 607-620

COMPARATIVE ANALYSIS OF NEURAL NETWORK MODEL SELECTION AND DATA TRANSFORMATION FOR RAINFALL FORECASTING

Regita Putri Permata[1*], Andrea Tri Rian Dani[2]
[1]Department of Mathematics, Telkom University, Bandung, West Java, Indonesia
[2]Department of Mathematics, Mulawarman University, Samarinda, East Kalimantan, Indonesia
*Correspondence: regitapermata@telkomuniversity.ac.id

ABSTRACT
The selection of input models in neural networks significantly influences predictive accuracy in time series forecasting. This study evaluates different input models for neural networks in rainfall prediction using data from the Wonorejo Reservoir, Surabaya. The neural network inputs are determined based on significant lags identified through the Partial Autocorrelation Function (PACF) and ARIMA models. Simulation results indicate that the best Feed Forward Neural Network (FFNN) model utilizes PACF-derived input lags and is trained using the Rprop+ algorithm with a logistic activation function. Meanwhile, the optimal Deep Learning Neural Network (DLNN) model employs the Rprop- algorithm with a logistic activation function. The best-performing model for rainfall forecasting, based on the lowest Root Mean Squared Error of Prediction (RMSEP), is the FFNN model with an (8,4,1) architecture. To further refine the model, we applied a stepwise selection process to eliminate non-significant lag inputs. However, results show that this optimization had no substantial impact, as RMSEP increased after the stepwise procedure.
Keywords: ARIMA, Feed Forward Neural Network, Deep Learning Neural Network

How to Cite: Permata, R., & Dani, A. T. R. (2025). Comparative Analysis Of Neural Network Model Selection and Data Transformation For Rainfall Forecasting. *Mathline: Jurnal Matematika dan Pendidikan Matematika, 10*(3), 607-620. http://doi.org/10.31943/Mathline.v10i3.763

PRELIMINARY

Forecasting theory increasingly focuses on neural networks (NN), leading to practical applications in time series prediction and explanatory forecasting (Makridakis et al., 2020; Tran et al., 2021). Notwithstanding their theoretical capabilities, neural networks have yet to demonstrate their efficacy in predicting compared to established statistical methods, such as ARIMA or Exponential Smoothing (Ge et al., 2022; Permata et al., 2024). NN provides a range of flexibility in the modeling process, including selecting activation functions, suitable network topologies for input, hidden, and output nodes, and learning methodologies (Jeremy & Suhartono, 2021). Their legitimate and reliable application is often perceived as an art and a science (Putri et al., 2021; Wanto et al., 2021).

Prior studies demonstrate that the economic identification of input variables and lags for forecasting an unknown data-generating process, in the absence of domain knowledge, presents a critical challenge in model definition (Wang et al., 2021; Apaydin et al., 2020). Box and Jenkins' study has opened the path for further research on the subject, particularly in determining the most relevant input variables in time series data for Neural Network model specifications (Jeremy & Suhartono, 2021; Pontoh et al., 2022). This task becomes increasingly crucial as complex time series components, such as deterministic or stochastic trends and seasonality, interact in a linear or nonlinear model with pulses, level shifts, structural breaks, and various noise distributions (Munandar et al., 2023; Suhartono et al., 2019). While numerous statistical methods have been established to aid in identifying linear dependencies, their application in nonlinear prediction has yet to be thoroughly examined. In a separate study, your work on the comparison of single and multiple hidden-layer networks (Narvekar & Fargose, 2015; Tran et al., 2021; Putri et al., 2021) has opened up new avenues for research and is integral to our understanding of why single hidden-layer networks converge to linear target functions faster than multiple hidden-layer networks (Author et al., 2010; Makridakis et al., 2017). To achieve the best prediction outcomes, the correct model selection requires information criteria; in this case, we utilize Root Mean Squared Error Prediction (RMSEP).

Advancements in computational analytics have facilitated real-time solutions for practical forecasting challenges. However, the training process for NN models can be computationally expensive. To enhance efficiency, effective data preprocessing techniques are crucial. Normalization, in particular, plays a vital role in improving predictive reliability by ensuring consistent data scaling before training (Rahman et al., 2015). This study examines how different normalization techniques contribute to improved forecasting accuracy, considering multiple factors influencing data transformation. This study illustrates how normalization methods enhance predictive accuracy. Multiple factors for data normalization are also taken into account.

This paper focuses on Computational Intelligence techniques in rainfall forecasting, specifically using Neural Networks and time series approaches to predict rainfall occurrence in a case study of Wonorejo, Surabaya (Mislan & Dani, 2024). The dataset is selected for its distinct seasonal patterns, encompassing both dry and rainy periods (Andriyana et al., 2024). Traditional statistical approaches alone may be insufficient in capturing the nonlinear dependencies present in rainfall data. Thus, this research aims to identify the most suitable NN input model by integrating statistical techniques, such as significant PACF lags and ARIMA modeling, to construct meaningful input variables.

This study is limited to rainfall forecasting at Wonorejo Reservoir, focusing solely on historical rainfall data from this specific location. The NN model developed is trained and validated using data from this region, and thus, its applicability to other geographic areas with different climatic conditions is not addressed. The research does not explore external environmental factors such as wind patterns, temperature fluctuations, or topographical influences, which may also impact rainfall predictions. Additionally, while this study compares NN-based forecasting with statistical methods like ARIMA, the evaluation is constrained to performance metrics such as RMSEP, without incorporating alternative validation techniques like probabilistic forecasting or uncertainty quantification. Further research is needed to assess the generalizability of the proposed approach to other locations with varying meteorological patterns.

METHODS

1. Autoregressive Integrated Moving Average (ARIMA p,d,q)

ARIMA is frequently referred to as the Box-Jenkins time series approach (A. T. R. Dani et al., 2023). ARIMA is highly accurate for both short and long-term forecasting. The ARIMA model combines the autoregressive (AR) and moving average (MA) models. (Hutagalung & Sari, 2024; Zen et al., 2023). According to Box and Jenkins, the ARIMA model (p,d,q) in Eq. 1.

$$\phi_p(B)(1-B)^d Y_t = \theta_0 + \theta_q(B)a_t. \tag{1}$$

2. Feed Forward Neural Network (FFNN)

The Feed Forward Neural Network (FFNN) is a prevalent nonlinear model extensively utilized for time series forecasting. FFNN has an input, hidden, and output layer (Tran et al., 2021). Each layer comprises components known as neurons. Each neuron receives information exclusively from the neurons in the preceding layer (Suhartono et al., 2019). The FFNN model for univariate time series data with p inputs, q hidden neurons, and a single output, represented as FFNN (p, q), can be articulated as follows.

$$\hat{Y}_{(t)} = f^o\left[\sum_{j=1}^{q}\left[w_j^o f_j^h\left(\sum_{i=1}^{p} w_{ji}^h X_{i(t)} + b_j^h\right) + b^o\right]\right], \tag{2}$$

Where:

w_{ji}^{h}	: the weights that connect input layer to hidden layer,
w_{j}^{o}	: the weights that connect hidden layer to output layer.
$f(.)$	: activation function,
b^{o} and b_{j}^{h}	: the biases
$X_{i(t)}$	: the input values
$\hat{Y}_{(t)}$	: the predicted output values.

In this study, we compared two algorithm namely Rprop- and Rprop+ and use two activations namely logistics and tanh (Wang et al., 2021). Which is given logistics activation function follows.

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

Tangent hyperbolic (tanh) function as the activation function, which is given as follows.

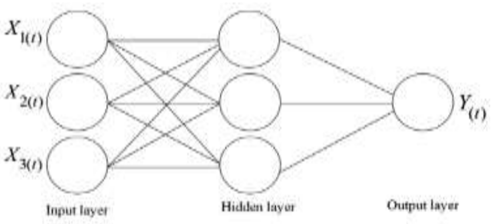
$$f(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{4}$$

3. Deep Learning Neural Network (DLNN)

Deep Learning Neural Networks (DLNN) are Feed Forward Neural Networks (FFNN) that include several hidden layers. (Makridakis et al., 2018). In the time series model, the DLNN model with two hidden layer is given as follows.

$$\hat{Y}_{(t)} = f^{o}\left[\sum_{i=1}^{s}\alpha_i f_i^{h_2}\left[\sum_{j=1}^{r}\beta_{ij} f_j^{h_1}\left(\sum_{t=1}^{p}\gamma_{it}X_{i(t)}+b_j^{h_1}\right)+b_i^{h_2}\right]+b^{o}\right]+\varepsilon_t \tag{5}$$

The architectures of FFNN and DLNN are shown in Fig. 1.



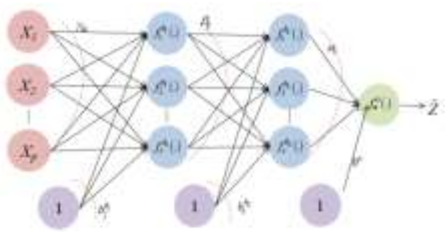


Figure 1. Visualization the architecture of FFNN (left) and DLNN (right)

4. Evaluation Model using Root Mean Square Error Prediction (RMSEP)

To evaluate the forecasting performance of the Feedforward Neural Network (FFNN) models, we employed the Root Mean Square Error of Prediction (RMSEP) as the primary evaluation metric. RMSEP is a widely used measure to assess the accuracy of predictive models, especially in regression and time series forecasting contexts. It is defined as the square root of the mean of the squared differences between the actual observed values and the predicted values:

$$RMSEP = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

where y_i denotes the actual observed value, $\hat{y}_i$ is the predicted value, and nnn is the total number of test data points. The RMSEP metric is particularly useful because it penalizes larger errors more heavily due to the squaring operation, thus providing a sensitive measure of model accuracy. Lower RMSEP values indicate better model performance.

5. Input Selection

Stepwise selection expands the forward selection technique by allowing the removal of input variables at any later iteration. This method is commonly employed to create linear regression models, and stepwise regression is a good illustration. This wrapper method repeatedly builds linear models by estimating their coefficients and adding new input variables. The input variables are kept after analyzing the newly formed model's coefficients. After k+1 input variables are judged to be no better than the previous k, the selection procedure is repeated until the model meets optimality criteria, such as the AIC. Figure 2 shows the neural network with the considerable lag added by the preceding phases after the input models impacted by the stepwise procedure were determined.

Figure 2. Diagram of Stepwise

6. Input Variable Selection for Neural Network

Our research here takes advantage of recently discovered rainfall data from Wonorejo. The dataset covers 1998–2018, with monthly rainfall data included. We create two sets of data: one which is used for training and the other which is used for testing. The training set contains data from January 1998 to December 2016, while the remaining data is used for testing.

The correct amount of hidden units, or the complexity of functional form, and the essential input variables must be selected before a network design can be specified. In this study, we used one hidden layer and tried 1 to 10 neuron units with ten replications in

FFNN models, compared two algorithms, namely Rprop- and Rprop+[1] and used two activations, namely logistics and tanh. Previous research in neural network modeling asserts that determining the pertinent time series delays by means of seasonal 12 differencing requires just an examination of the AR terms derived by PACF-analysis.

This flowchart illustrates the modeling and evaluation process in time series analysis using ARIMA and neural networks. The process begins with data transformation and progresses through model training and evaluation. The process starts with data transformation to ensure optimal data quality before analysis. Next, the Partial Autocorrelation Function (PACF) is used to identify significant lag dependencies. Based on the PACF results, significant lag inputs are generated and utilized in the ARIMA model as well as for input selection in neural network models. Two types of neural network models are employed: Feedforward Neural Network (FFNN) and Deep Learning Neural Network (DLNN). After training these models, the results are compared and evaluated to determine the best approach for time series forecasting.

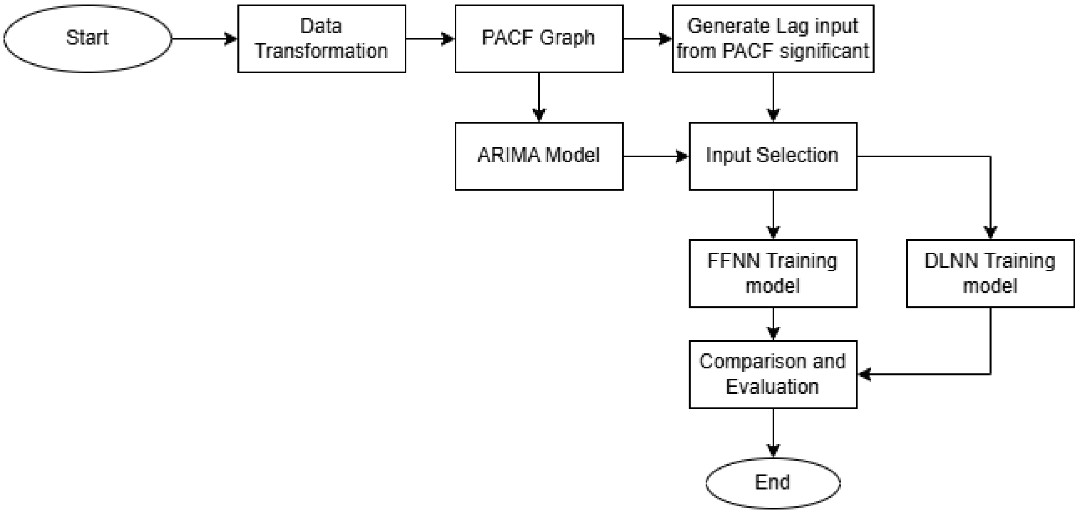


Figure 3. Flowchart

RESULT AND DISCUSSION

1. Exploratory Data

The average rainfall in the Wonorejo Surabaya during the 1998-2006 period was 213.52 mm. The time series plot shows that rainfall is seasonal in a year and has a

[1] Rprop- (Resilient Propagation Minus) and Rprop+ (Resilient Propagation Plus) are two variations of the Resilient Backpropagation (Rprop) optimization algorithm used for training neural networks (Mosca and Magoulas 2015).

tendency to increase from October to March. Time series plot of Wonorejo rainfall are given Fig. 4.

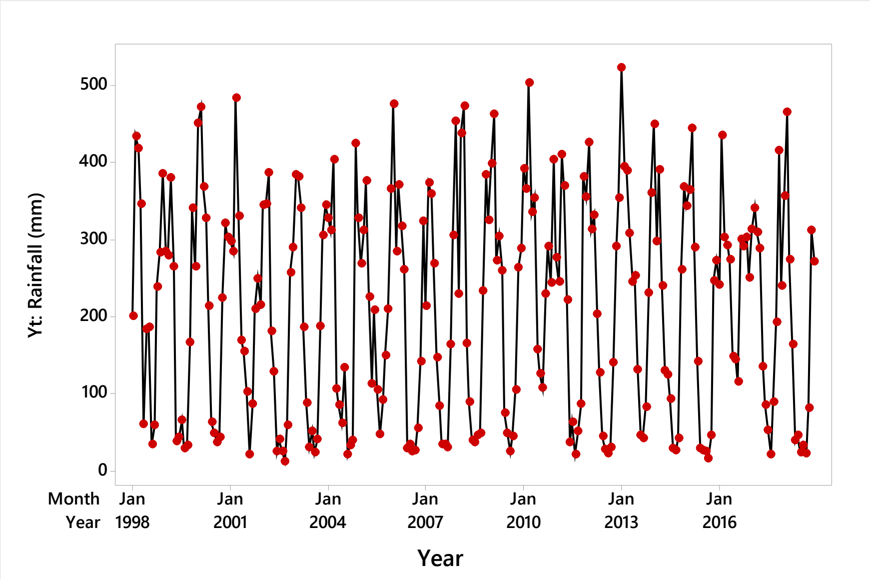


Figure 4. Time Series plot of Wonorejo rainfall

Figure 5 shows that in January, February, March, April, November and December the average value of rainfall is quite high. Meanwhile, in July, August and September have average rainfall was low, which indicates that there is a dry season and the season from the rainy season to the dry season or vice versa in the month. In 2016 there were differences in rainfall patterns in August and September so that it showed a higher rainfall pattern than other patterns.

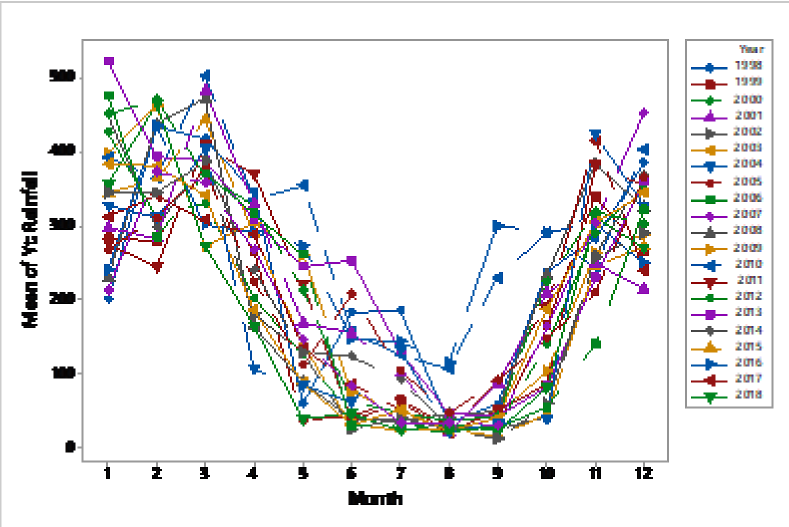


Figure 5. Line plots of Wonorejo rainfall

ARIMA Modeling

ARIMA (p,d,q) modeling is carried out following the Box-Jenkins stages, namely identifying data stationarity through stationarity in variance and mean; determining temporary ARIMA order (p,d,q) based on ACF and PACF graphs; parameter estimation; diagnostic examination, and forecasting. ARIMA models along with forecast accuracy are shown in Table 1.

Table 1. Forecast evaluation of ARIMA

ARIMA models	White Noise	RMSEP
$(0,0,1)(0,1,1)^{12}$	Yes	65.552
$([1,3,5],0,0)(2,1,0)^{12}$	Yes	68.632

The ARIMA model is more accurate in ARIMA (0,0,1)(0,1,1)12 using RMSEP evaluation.

2. Feed Forward Neural Network (FFNN)

At the FFNN modeling stage, we form an NN architecture with a combination of neurons in each hidden layer. It also carries out a combination of algorithms and activation functions. The input variables, which are the initial values in the FFNN architecture, are seen based on the PACF graph of significant lags $\{y_{t-1}, y_{t-3}, y_{t-5}, y_{t-12}, y_{t-13}, y_{t-15}, y_{t-17}, y_{t-24}\}$

Table 2. Forecast evaluation of FFNN using standardized preprocessing

	Rprop-		Rprop+	
Neuron	Test tanh	Test log	Test tanh	Test log
1	85.644	85.733	85.697	85.689
2	75.953	75.858	100.358	75.577
3	90.809	92.706	117.529	67.238
4	95.274	58.882	78.562	74.887
5	127.755	87.463	132.249	147.302
6	148.567	87.884	147.779	130.245
7	101.211	116.546	147.047	96.924
8	83.6935	163.920	114.704	95.592
9	218.357	167.571	87.2509	137.986
10	179.264	124.907	188.453	153.474

Table 3. Forecast evaluation of FFNN using normalized preprocessing

	Rprop-		Rprop+	
Neuron	Test tanh	Test log	Test tanh	Test log
1	81.961	81.933	82.730	81.882
2	72.221	72.434	72.643	73.087
3	74.229	56.237	62.611	69.224
4	80.236	91.173	107.078	51.961
5	72.424	95.602	68.821	85.179
6	130.252	66.680	79.918	97.061
7	147.347	95.671	142.745	94.956
8	124.509	643.671	799.918	82.737
9	127.059	128.918	1396.57	97.498
10	138.653	98.194	99.694	98.048

Table 4. Forecast evaluation of FFNN using adjusted normalized preprocessing

	Rprop-		Rprop+	
Neuron	**Test tanh**	**Test log**	**Test tanh**	**Test log**
1	83.883	83.881	83.994	83.854
2	75.957	73.765	76.043	76.337
3	74.653	90.767	74.629	124.27
4	73.800	77.655	115.977	74.319
5	90.457	99.334	98.921	64.196
6	105.844	97.168	104.941	81.509
7	149.521	75.194	222.455	120.726
8	106.686	111.952	154.455	106.033
9	96.391	163.017	141.934	96.548
10	309.068	118.408	109.435	232.803

Based on Tables 2, 3, and 4, it can be concluded that the performance of the Feedforward Neural Network (FFNN) model is significantly influenced by the data preprocessing method, training algorithm (Rprop- or Rprop+), number of neurons, and activation function (tanh or logistic). Among the three tables, the normalized preprocessing method generally yields better performance, as indicated by lower RMSEP (Root Mean Square Error of Prediction) values compared to other preprocessing techniques. The best result is observed in Table 3**,** where using 4 neurons**,** the logistic activation function, Rprop+ algorithm, and normalized preprocessing produces the lowest RMSEP of 51.961. This combination significantly reduces prediction error. Moreover, the logistic activation function tends to outperform tanh in most cases, highlighting the importance of selecting an appropriate activation function when designing the FFNN architecture.

3. Feed Forward Neural Network (FFNN) using ARIMA Model

From the modeling of ARIMA, we use input model from ARIMA $([1,3,5],0,0)(2,1,0)^{12}$. ARIMA model selection using subset because the nonseasonal AR model and seasonal AR model with differencing 12.

Table 5. Forecast evaluation of FFNN input ARIMA model

	Rprop-		Rprop+	
Neuron	**Test tanh**	**Test log**	**Test tanh**	**Test log**
1	66.884	66.869	67.263	66.990
2	89.193	87.735	88.743	87.293
3	88.007	86.462	70.827	57.057
4	72.414	98.901	86.933	96.239
5	105.877	91.439	78.270	94.123
6	102.805	67.842	101.297	121.321
7	74.113	75.623	110.471	393.740
8	86.111	133.759	84.130	107.857
9	96.272	163.290	373.684	168.343
10	515.559	99.662	650.098	98.330

4. Deep Learning Neural Network Model

DLNN model input variables significant lag from PACF $\{y_{t-1}, y_{t-3}, y_{t-5}, y_{t-12}, y_{t-13}, y_{t-15}, y_{t-17}, y_{t-24}\}$. The algorithms tested using the DLNN method are Rprop + and Rprop- without replication. The DLNN architectures that will be applied are two hidden layers with 1 to 3 neurons in each layer. We didn't use replication because the value is not affected by adding nodes in the DLNN layer.

Table 6. Forecast evaluation of DLNN-PACF using standardized preprocessing

Neuron	RPROP+		RPROP-	
	Tanh	Logistic	Tanh	Logistic
	RMSEP	RMSEP	RMSEP	RMSEP
1-1	67.251	67.076	67.245	67.096
1-2	66.021	69.988	66.142	67.244
2-1	63.402	62.548	67.255	66.966
2-2	67.285	66.952	64.752	77.899
3-1	81.865	91.923		
3-2	74.233	101.670		

Table 7. Forecast evaluation of DLNN-PACF using normalized preprocessing

Neuron	RPROP+		RPROP-	
	Tanh	Logistic	Tanh	Logistic
	RMSEP	RMSEP	RMSEP	RMSEP
1-1	63.717	63.816	63.165	63.723
1-2	75.864	75.666	75.143	77.301
2-1	75.469	64.850	78.878	60.537
2-2	64.145	64.877	62.778	63.775
3-1	74.664	75.870	74.736	70.666
3-2	72.510	70.259	63.631	58.744

Table 8. Forecast evaluation of DLNN-PACF using adjusted normalized preprocessing

Neuron	RPROP+		RPROP-	
	Tanh	Logistic	Tanh	Logistic
	RMSEP	RMSEP	RMSEP	RMSEP
1-1	65.316	65.310	65.364	64.983
1-2	67.452	77.425	82.532	61.499
2-1	66.839	68.093	81.649	66.918
2-2	64.916	64.956	64.808	64.873
3-1	77.120	69.032	62.702	64.051
3-2	75.879	65.669	64.942	65.733

Based on Table 6, Table 7, and Table 8, the best DLNN model results are obtained when using the Rprop-algorithm. However, when the standardized data as input selection, the convergence model is not achieved when the first neuron is more than two. Probably, it is because the uncontrolled weight and bias on it neuron make too many iterations needed to reach convergence. The convergence iteration criteria which has been declare at the beginning were not enough to make the model on neuron be convergence, so it must be modified again, moreover the more neurons the older time needed for converging model.

5. Evaluation

The use of Rprop- algorithm will produce a good DLNN when using normalized data as preprocessing data and logictic activation function. From the Neural Network input simulation, we found that FFNN method with PACF input using Rprop+ on neuron 4 has the best accuracy. Then we use stepwise on PACF input to get a significant lag. The result of stepwise method shows that significant lag input from PACF are $\{y_{t-1}, y_{t-3}, y_{t-5}, y_{t-12}, y_{t-15}\}$ and RMSEP value obtained by 59.759.

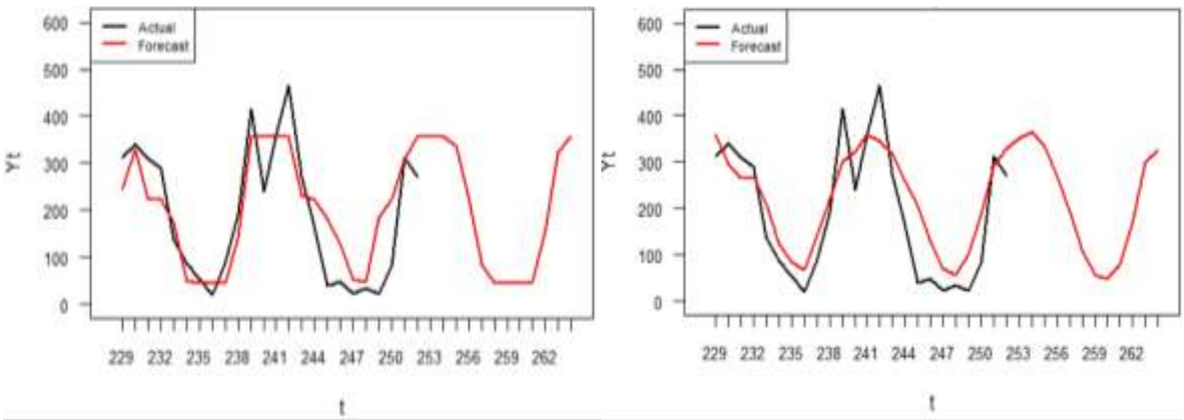


Figure 6. Rainfall forecast from (a) Lag PACF input model (b) Stepwise methods input models

The result show that FFNN (single hidden layer) not only fast but also more accurate than DLNN (multiple hidden layers) to estimate only in time series data as explained in Nakama (2011). The developement using stepwise to optimize the lag input for Neural Network. The stepwise process give us a glimps that it could make the process to estimate FFNN faster that without stepwise.

CONCLUSION

Based on the results, normalization proved to be the most effective preprocessing method for NN input, leading to better performance in the forecasting model. The best activation function identified was logistic, combined with the Rprop+ algorithm. When analyzing rainfall data in Wonorejo Reservoir, the FFNN method outperformed both ARIMA and DLNN models, as it achieved the lowest RMSEP.

To optimize input selection, the stepwise method was applied, reducing the number of input variables from eight to five. While this process did not lead to significant changes in forecasting accuracy, it notably reduced computation time. However, when applied to input lag selection, the stepwise approach resulted in a higher RMSEP, indicating a trade-off between accuracy and faster estimation. In practice, this trade-off may be beneficial when computational efficiency is a priority, but it may not be ideal when accuracy is the primary concern.

This study highlights the importance of selecting appropriate preprocessing and input selection methods in neural network-based time series forecasting. However, it is limited to a specific dataset and parameter configurations. Future research should explore alternative optimization techniques, different neural network architectures, and the impact of hybrid approaches to further improve forecasting performance while balancing accuracy and computational efficiency.

ACKNOWLEDGMENT

We would like to thank the Ministry of Education, Culture, Research and Technology, Telkom University, and Mulawarman University

REFERENCES

Andriyana, Y., Falah, A. N., Ruchjana, B. N., Sulaiman, A., Hermawan, E., Harjana, T., & Lim-Polestico, D. Lou. (2024). Spatial Durbin Model with Expansion Using Casetti's Approach: A Case Study for Rainfall Prediction in Java Island, Indonesia. *Mathematics*, 12(15), 2304. https://doi.org/10.3390/math12152304

Apaydin, H., Feizi, H., Sattari, M. T., Colak, M. S., Shamshirband, S., & Chau, K. W. (2020). Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. *Water (Switzerland)*, 12(5), 1-18. https://doi.org/10.3390/w12051500

Dani, A. T., Fauziyah, M., & Sandariria, H. (2023). Forecasting The Search Trends of The Keyword "Sarung Wadimor" In Indonesia on Google Trends Data Using Time Series Regression with Calender Variation and Arima Box-Jenkins. *Jurnal Matematika, Statistika Dan Komputasi*, 19(3), 447–459. https://doi.org/10.20956/j.v19i3.24551

Ge, W., Lalbakhsh, P., Isai, L., Lenskiy, A., & Suominen, H. (2022). Neural Network-Based Financial Volatility Forecasting: A Systematic Review. *ACM Computing Surveys*, 55(1), 1-30. Association for Computing Machinery. https://doi.org/10.1145/3483596

Hutagalung, A. A., & Sari, R. F. (2024). Forecasting Indihome Users By Using Trigonometrics, Box Cox, Transformation, Arma Error, Trend, And Seasonal (TBATS) Methods. *Mathline : Jurnal Matematika Dan Pendidikan Matematika*, 9(1), 243–256. https://doi.org/10.31943/mathline.v9i1.579

Jeremy, N. H., & Suhartono, D. (2021). Automatic personality prediction from Indonesian user on twitter using word embedding and neural networks. *Procedia Computer Science*, 179, 416–422. https://doi.org/10.1016/j.procs.2021.01.024

Makridakis, S., Assimakopoulos, V., & Spiliotis, E. (2018). Objectivity, reproducibility and replicability in forecasting research. *International Journal of Forecasting*, 34(4), 835–838. Elsevier B.V. https://doi.org/10.1016/j.ijforecast.2018.05.001

Makridakis, S., Hyndman, R. J., & Petropoulos, F. (2020). Forecasting in social settings: The state of the art. *International Journal of Forecasting*, 36(1), 15–28. https://doi.org/10.1016/j.ijforecast.2019.05.011

Makridakis, S., Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning forecasting methods: Concerns and ways forward," PLOS ONE, vol. 13, no. 3, p. e0194889, Mar. 2018, doi: 10.1371/journal.pone.0194889.

Mislan, M., & Dani, A. T. R. (2024). Forecasting Maximum Water Level Data for Post Sangkuliman using An Artificial Neural Network Backpropagation Algorithm. *JTAM (Jurnal Teori Dan Aplikasi Matematika)*, 8(2), 465-578. https://doi.org/10.31764/jtam.v8i2.20112

Munandar, D., Ruchjana, B. N., Abdullah, A. S., & Pardede, H. F. (2023). Literature Review on Integrating Generalized Space-Time Autoregressive Integrated Moving Average (GSTARIMA) and Deep Neural Networks in Machine Learning for Climate Forecasting. *Mathematics*, 11(13), 3-25. Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/math11132975

Narvekar, M., & Fargose, P. (2015). Daily Weather Forecasting using Artificial Neural Network. *International Journal of Computer Applications*, 121(22), 9-13. DOI=10.5120/21830-5088

Pontoh, R. S., Toharudin, T., Ruchjana, B. N., Sijabat, N., & Puspita, M. D. (2022). Bandung Rainfall Forecast and Its Relationship with Niño 3.4 Using Nonlinear Autoregressive Exogenous Neural Network. *Atmosphere*, 13(2), 1-17. https://doi.org/10.3390/atmos13020302

Permata, R. P., Ni'mah, R., & Dani, A. T. R. (2024). Daily Rainfall Forecasting with ARIMA Exogenous Variables and Support Vector Regression. *Jurnal Varian*, 7(2), 177–188. https://doi.org/10.30812/varian.v7i2.3202

Putri, J. A., Suhartono, S., Prabowo, H., Salehah, N. A., Prastyo, D. D., & Setiawan, S. (2021). Forecasting Currency in East Java: Classical Time Series vs. Machine Learning. *Indonesian Journal of Statistics and Its Applications*, 5(2), 284–303. https://doi.org/10.29244/ijsa.v5i2p284-303

Rahman, N. H. A., Lee, M. H., Suhartono, & Latif, M. T. (2015). Artificial neural networks and fuzzy time series forecasting: an application to air quality. *Quality and Quantity*, 49(6), 2633–2647. https://doi.org/10.1007/s11135-014-0132-6

Suhartono, Setyowati, E., Salehah, N. A., Lee, M. H., Rahayu, S. P., & Ulama, B. S. S. (2019). A hybrid singular spectrum analysis and neural networks for forecasting inflow and outflow currency of bank Indonesia. *Communications in Computer and Information Science*, 937, 3–18. https://doi.org/10.1007/978-981-13-3441-2_1

Suhartono, Ashari, D. E., Prastyo, D. D., Kuswanto, H., & Lee, M. H. (2019). Deep neural network for forecasting inflow and outflow in Indonesia. *Sains Malaysiana*, 48(8), 1787–1798. https://doi.org/10.17576/jsm-2019-4808-26

Suhartono, B. S. S. Ulama, and A. J. Endharta. 2010. "Seasonal Time Series Data Forecasting by Using Neural Networks Multiscale Autoregressive Model." *American Journal of Applied Sciences* 7(10):1372–78. doi:10.3844/ajassp.2010.1372.1378.

Tran, T. T. K., Bateni, S. M., Ki, S. J., & Vosoughifar, H. (2021). A review of neural networks for air temperature forecasting. *Water (Switzerland),* 13(9), 1-15. MDPI AG. https://doi.org/10.3390/w13091294

Wang, Y., Zou, R., Liu, F., Zhang, L., & Liu, Q. (2021). A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy*, 304. https://doi.org/10.1016/j.apenergy.2021.117766

Wanto, A., Fauzan, M., Suhendro, D., Parlina, I., Damanik, B. E., Siregar, P. A., & Hidayati, N. (2021). Epoch Analysis and Accuracy 3 ANN Algorithm using Consumer Price Index Data in Indonesia. *Proceedings of the 3rd International Conference of Computer, Environment, Agriculture, Social Science, Health Science, Engineering and Technology ICEST*, 1, 35–41. https://doi.org/10.5220/0010037400350041

Zen, M. A., Wahyuningsih, S., Tri, A., Dani, R., Ekonomi, S., Bisnis, D., Statistika, S., Matematika, J., Matematika, F., Ilmu, D., Alam, P., Mulawarman, U., Samarinda, K., Timur, I., & Terapan, L. S. (2023). Aplikasi Pengelompokan Data Runtun Waktu dengan Algoritma K-Medoids. *INFERENSI*, 6(2), 2721–3862. https://doi.org/10.12962/j27213862.vxix.xxxx