Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2025, 2025:107

https://doi.org/10.28919/cmbn/9487

ISSN: 2052-2541

MACHINE LEARNING BASED HYDROLOGICAL MODEL: DEEP FEED FORWARD NEURAL NETWORK (DFFNN)

MISLAN[1,*], ANDREA TRI RIAN DANI[2,3], M. NUR ALFI SYAHRI[2]

[1]Department of Physics, Faculty of MIPA, Mulawarman University, Samarinda, Indonesia

[2]Department of Mathematics, Faculty of MIPA, Mulawarman University, Samarinda, Indonesia

[3]Doctoral Study Program MIPA, Faculty Science and Technology, Airlangga University, Surabaya, Indonesia

Abstract: This study applies the Deep Feed Forward Neural Network (DFFNN) model to forecast the number of rainy days in Indonesia, a tropical country characterized by significant seasonal rainfall variations. The data used includes daily rainfall data from the Juwata Tarakan Meteorological Station over the period from 2014 to 2024. DFFNN was chosen for its ability to handle non-linear relationships and temporal dependencies within complex time series data. Based on performance evaluation metrics such as MAPE, MSE, and RMSE, the model with a 4-hidden-layer architecture and a 4-5-3-5 neuron configuration, standardized using Z-Score, showed the best performance compared to other model combinations. The forecasting results reveal a seasonal pattern consistent with tropical rainfall cycles, with peaks in rainfall occurring from January to March and a decline observed between May and August. The model proved capable of providing accurate predictions for the number of rainy days, making it suitable for supporting early warning systems in mitigating natural disaster risks such as floods and landslides. However, the study also identified several limitations, particularly in the selection of input variables, hyperparameter tuning, and network architecture choices, which influence model performance. Therefore, further research is recommended to optimize these aspects to enhance prediction accuracy and stability in weather forecasting and disaster management.

Keywords: hydrology; climatology; forecasting; DFFNN; machine learning.

2020 AMS Subject Classification: 68T07.

*Corresponding author

E-mail address: mislan.jos@fmipa.unmul.ac.id

Received July 13, 2025

1. INTRODUCTION

Hydrology is a branch of science that studies the distribution and movement of water on the Earth's surface and its interaction with the surrounding environment. Hydrological data, such as rainfall, number of rainy days, river discharge, and humidity, are closely related to time series data, which are data sets organized over time [1]. Time series data in hydrology often contain non-linear patterns that are difficult to model, mainly because of the many factors that influence their variability [2]. The main difficulty in modeling hydrological data arises from several complex properties contained in the data [3]. One of the main challenges is the large number of zero values in rainfall data, which often occur on days without rain, causing problems in analysis. These zero values can interfere with the effectiveness of conventional statistical models such as moving averages, single exponential smoothing, and ARIMA, which are less effective in handling data that has an absence of phenomena (e.g., days without rain) compared to data that contains actual phenomena [2], [4].

In addition, hydrological data often exhibit highly complex non-linear patterns and are influenced by unpredictable seasonal factors, such as climate change or extreme natural phenomena such as El Niño [5], [6]. Unpredictable seasonal variations add to the complexity of modeling rainfall and other hydrological parameters. In addition, the complex interactions between different hydrological variables, such as rainfall, soil moisture, and streamflow, are often non-linear and interdependent, making them very difficult to model using conventional approaches that rely on the assumption of linear relationships. Hydrological data are also usually irregular and have significant gaps between measurements and long-term dependencies that are difficult to predict. Therefore, machine learning models, especially neural networks, offer a better solution, given their ability to handle non-linear and complex relationships and their adaptation to data containing zero values and unpredictable seasonal variations [7], [8], [9].

Along with machine learning technology, statistical models are increasingly developed and can be trained to recognize patterns more reliably. One of the well-known machine learning models is the neural network (NN), a computational model that imitates the way the human brain processes information, using many interconnected neurons [10], [11], [12]. The main advantage of neural networks is their ability to model very complex patterns, including non-linear data, making them more flexible than traditional methods [13], [14], [15]. A typical neural network architecture is the Feed Forward Neural Network (FFNN), which consists of an input layer, a hidden layer, and an output layer [16], [17]. This study used the Deep Feed Forward Neural Network (DFFNN), a type

of neural network with more hidden layers than the conventional FFNN [16], [18]. By increasing the number of hidden layers, the DFFNN model can learn more complex representations of the data, so it can handle patterns that are more difficult to detect in traditional models[19], [20].

The main advantage of DFFNN in hydrological modeling is its ability to identify complex non-linear relationships between hydrological parameters, which conventional statistical models cannot handle. DFFNN has high flexibility in handling various forms of data and can work well on large and unstructured data, such as hydrological time series data. The exemplary DFFNN architecture and appropriate data standardization techniques are essential for optimal model performance. This study tested two standardization techniques, namely z-score standardization and min-max standardization, to evaluate their effects on model convergence and accuracy. Standardization techniques are needed to avoid scale issues affecting the speed and quality of model training, especially on data with highly variable value ranges.

In addition, the model learning process is carried out using Backpropagation and Resilient Backpropagation, both of which are practical optimization algorithms in updating the weights of neural networks to minimize errors. Backpropagation works by calculating the error gradient at each layer of the network and updating its weights, while Resilient Backpropagation improves optimization performance in a faster and more stable way. By combining DFFNN with appropriate standardization techniques and optimization algorithms, this model can better model complex and non-linear hydrological data than conventional methods.

Thus, the novelty of this study lies in the application of Deep Feed Forward Neural Network (DFFNN) in hydrological data modeling, which not only overcomes the limitations of traditional models but also optimizes the use of standardization techniques and optimization algorithms in dealing with complex and dynamic natural phenomena. This study offers a more flexible and accurate approach to handling non-linear data and uncertainty in hydrological data, such as rainfall and river discharge. It significantly contributes to developing machine learning models that can be more effectively used for analysis and forecasting in hydrology.

2. PRELIMINARIES

A. Neural Network

Artificial Neural Networks (ANNs) constitute an artificial-intelligence paradigm for information processing that emulates the functional properties of the human nervous system[15], [21], [22]. Grounded in neurobiological principles, these models reproduce a network of

elementary processing units neurons that individually execute simple computations yet collectively yield highly complex operations through their interconnections [23], [24]. Effective computation follows a prior learning phase, and, mirroring their biological analogue, ANNs display adaptivity: the strengths of the inter-neuronal links (synaptic weights) update dynamically, allowing continuous assimilation of new information. The classical ANN framework rests on four core postulates:

a. Information is handled by basic processing units termed neurons.
b. Neurons are linked via pathways that transmit activation signals.
c. Each linkage carries an adjustable weight that modulates the transmitted signal.
d. A neuron's output is produced by an activation function that combines its weighted inputs.

By reproducing the brain's plastic and dynamic connectivity, ANNs furnish an adaptive information-processing architecture capable of refining its internal representations with experience [25], [26]. In the form of a mathematical model, the network architecture can be described in Equation (1).

$$x_t = f^{\circ}\left(w^b + \sum_{j=1}^{l} w_j^{\circ} f^h\left(w_j^b + \sum_{i=1}^{p} w_{ji}^h x_{t-i}\right)\right) \quad (1)$$

where:

x_t : network output at time *t*

f° : activation function applied to the output layer

w^b : bias weight vector for output layer

l : number of neurons in the hidden layer

w_j° : weight vector from hidden layer to output

f^h : activation function on the hidden layer

w_j^b : bias weight vector in the hidden layer

p : number of time steps (lags) considered as input to the hidden layer

w_{ji}^h : the weight vector connecting the input to the hidden layer

x_{t-i}: input at time $t-i$, which is the previous data (lagged input).

f^h is the activation function in the hidden layer and the weight vector of the network consisting of the weight of the input layer to the hidden layer (w_{ij}), the weight of the bias to the

hidden layer (w_j^b), the weight of the hidden layer to the output (w_j) and the weight of the bias to the output (w_b) [27].

1. Output at time t (x_t)

   The output at time t, x_t, is generated by applying the output activation function f° to the total sum of the bias weights in the output layer and the result of the processed hidden layer.

2. Summation of the hidden layer

   Inside the brackets, there is the sum of the weights connecting the neurons in the hidden layer to the output layer. This is done for each neuron in the hidden layer, calculated from the activation function in the hidden layer.

3. Activation function on hidden layer

   f° is the activation function applied to the linear combination of the biased weights w_j^b and the input influenced by the weights w_{ji}^h for the delayed input data at time step $t-i$.

4. Input at time $t-i$

   The input to the hidden layer is not only based on the current data, but also data at previous times. The input x_{t-i} at time $t-i$ indicates the temporal (time-dependent) nature of this model.

B. Deep Feed Forward Neural Network

A Deep Neural Network (DNN) is a multilayer artificial-neural architecture whose depth exceeds three layers, specifically, an input layer, at least two hidden layers (≥2 hidden layers), and an output layer hence the term "deep" [16], [21]. The optimisation procedure used to estimate the network parameters is generally known as deep learning, while the resulting architecture itself is referred to as a DNN[16].

The Feed Forward Neural Network (FFNN) is a flexible subclass of DNN frequently applied to nonlinear regression, dimensionality reduction, and modelling of nonlinear dynamic systems [18], [20]. An FFNN comprises numerous processing units (neurons) arranged in sequential layers with strictly unidirectional connections: the output of one layer becomes the input to the next, and no feedback loops are present. Each connection carries an adjustable weight; learning involves iteratively updating these weights and their associated biases so that the network maps input signals to target outputs with minimal error. The aggregate of weighted inputs at each neuron is passed through an activation function, introducing nonlinearity and enhancing representational

capacity[28].

The predictive performance of an FFNN is governed chiefly by three factors: (i) the network architecture (e.g., number of layers and neurons per layer), (ii) the training or optimisation algorithm employed, and (iii) the choice of activation function [10], [29]. For an FFNN with ppp input variables and a single hidden layer containing mmm neurons, the functional mapping is expressed mathematically in Equation (2).

$$f(x_t, \mathrm{v}, \mathrm{w}) = g_2\left\{\sum_{j=0}^{m} v_j, g_1\left[\sum_{i=0}^{p} w_{ji}, x_{it}\right]\right\} \quad (2)$$

where:

$f(x_t, \mathrm{v}, \mathrm{w})$: the output function generated by the neural network at time *t*, with parameters *v* and *w*

x_t : input vector at time *t*. the network receives this input from outside to process

w : the weight vector connecting the input and hidden layer. these weights control how much influence input x_{it} has on the activation of neurons in the hidden layer. w_{ji} is the weight that connects input x_{it} to neuron *j* in the hidden layer.

g_1 : activation function for the hidden layer that converts the linear combination of inputs into a non-linear output

g_2 : activation function for the output layer that is used to convert the result of the linear combination of the hidden layers into the final output.

w_{ji}, v_j : weights estimated by the backpropagation process

1. **Hidden Layer**

   In this neural network, we have m neurons in the hidden layer. Each neuron in the hidden layer receives inputs from all p inputs (including bias), and the weight connecting the input x_{it} to the hidden neuron j is w_{ji}. The activation function g_1 is then applied to the sum. The linear combination for neuron j in the hidden layer can be written in Equation (3).

$$z_j = \sum_{i=0}^{p} w_{ji} x_{it} \quad (3)$$

   Here, x_{it} is the input to neuron j at time t, and w_{ji} is the weight connecting x_{it} to neuron j. The activation function g_1 is applied to the result:

$$h_j = g_1(z_j) = g_1\left(\sum_{i=0}^{p} w_{ji} x_{it}\right) \quad (4)$$

Here, h_j is the output of neuron j in the hidden layer.

2. Output Layer

Once we have the outputs of all the neurons in the hidden layer, they are combined with the weights v_j to produce the final output of the network. The linear combination in the output layer written in Equation (5).

$$y = \sum_{j=0}^{m} v_j h_j = \sum_{j=0}^{m} v_j g_1\left(\sum_{i=0}^{p} w_{ji} x_{it}\right) \quad (5)$$

After that, the activation function g_2 is applied to generate the final output in Equation (6).

$$f(x_t, v, w) = g_2\left(\sum_{j=0}^{m} v_j g_1\left(\sum_{i=0}^{p} w_{ji}, x_{it}\right)\right) \quad (6)$$

C. Algorithm Rprop+ and Backpropagation

Backpropagation refers to a multilayer neural-network training procedure. Like other learning algorithms for artificial neural networks, it seeks an optimal trade-off between memorising the patterns presented during training and correctly generalising to novel inputs that resemble those patterns[30]. Fundamentally, backpropagation applies a gradient-descent strategy to adjust the synaptic weights linking successive layers the input, hidden and output layers, so that the discrepancy between observed targets and the network's predictions is driven to a minimum . The backpropagation algorithm in the FFNN model consists of three stages, namely feedforward of the input pattern, calculating the error and adjusting the weights. In this algorithm, the initial weights are determined in advance. In the feedforward stage, the output of the initial points is calculated. weighting is done using a gradient-based method to get a smaller error. A stopping criterion is determined to stop the iteration process [31].

Neural Network Rprop (Resilient Propagation) is a neural network learning algorithm developed to overcome the weaknesses in neutral network backpropagation which usually uses a sigmoid activation function. This activation function will bring inputs with an unlimited range to an output value with a limited range between 0 and 1. One of the characteristics of the sigmoid

function is that the gradient will approach zero if the input is given very much. This near-zero gradient has implications for low weight changes. If the weights do not change enough, the algorithm will be very slow to approach its optimum value [32], [33].

The Rprop algorithm uses the sign (positive or negative) of the gradient to indicate the direction of the weight adjustment. The size of the weight change is determined by the adjustment value (Δ_0). The adjustment value has a lower limit (Δ_{min})) and an upper limit (Δ_{max}). Other Rprop parameters are the decreasing factor ($\eta -$) and increasing factor ($\eta +$) or can be said to be the learning rate as in backpropagation. The learning rate values that are often used are 1.2 for ($\eta +$) and 0.5 for ($\eta -$) [21], [32], [34]. The rules for adjusting the value are as follows:

$$\Delta_{ij}{}^{(t)} = \begin{cases} \eta^{+}\Delta_{ij}{}^{t}, if \dfrac{\partial E^{(t)}}{\partial W_{ij}} \dfrac{\partial E^{(t-1)}}{\partial W_{ij}} > 0 \\ \eta^{-}\Delta_{ij}{}^{t}, if \dfrac{\partial E^{(t)}}{\partial W_{ij}} \dfrac{\partial E^{(t-1)}}{\partial W_{ij}} > 0 \\ \Delta_{ij}, else \end{cases} \quad (7)$$

where:

η^{-} : learning rate decrease (decreasing factor)

η^{+} : learning rate enhancer (enhancement factor)

$\frac{\partial E^{(t)}}{\partial W_{ij}}$: slope of the error curve against the weights (gradient) at the current iteration

$\frac{\partial E^{(t-1)}}{\partial W_{ij}}$: slope of the error curve against the weights (gradient) at the previous iteration.

D. Data Standardization

Min–max normalization constitutes a linear rescaling technique that projects each observation of a variable onto a predetermined numerical interval, most commonly [0, 1], while maintaining the original proportional distances among data points [35]. By anchoring the minimum observed value to the lower bound of the interval and the maximum to the upper bound, the procedure guarantees that all transformed values remain within the specified range, thereby facilitating fair comparison across variables with differing units or magnitudes. This method can use the following formula in Equation (8).

$$X_{new} = \frac{X - min\,(X)}{max(X) - min\,(X)} \quad (8)$$

where:

X_{new} : the new value from the normalized results

X : old value

$max(X)$: maximum value in the dataset

$min\ (X)$: minimum value in the dataset.

Z-score normalization (also referred to as standardization) rescales a quantitative variable by removing its location μ (the sample mean) and scaling by its dispersion σ (the sample standard deviation), thus producing dimensionless values characterised by zero mean and unit variance [35]. This transformation is advantageous when the true extrema of the data are unknown or uninformative, and when subsequent statistical or machine-learning procedures assume features of comparable scale [36]. This method can use the following formula in Equation (9).

$$X_{new} = \frac{X - \mu}{\sigma} = \frac{X - mean\ (X)}{StdDev(X)} \tag{9}$$

where:

X_{new} : the new value from the normalized results

X : old value

μ : population mean

σ : standard deviation value

E. Model Goodness Criteria

1. Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) is the absolute value of the percentage error of the data against the mean which can be formulated in Equation 10.

$$MAPE = \sum \frac{\frac{\sum|Actual - Prediction|}{Actual} \times 100}{n} \tag{10}$$

2. Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is the sum of the squares of the error or difference between the actual value and the predicted value, then divide the sum by the number of times of the forecasting data and then pull the root, can be formulated in Equation (11).

$$RMSE = \sqrt{\frac{\sum(Actual - Prediction)^2}{n}} \tag{11}$$

3. Mean Square Error (MSE)

Mean Square Error (MSE) is the result of the subtraction between the actual value and the squared predicted value, then the sum of these results with n is the number of periods used for calculation.

$$MSE = \frac{\sum(Actual - Prediction)^2}{n-1} \quad (12)$$

F. Data and Data Sources

This study uses monthly rainy-day data obtained from Juwata Tarakan Meteorological Station. The data used covers the period from January 2014 to December 2024, so it consists of 132 monthly observations. The data is used to analyze rainfall patterns in the long term as well as the basis for modeling and forecasting processes. The following table displays a snapshot of the data used in this study:

TABLE 1. Number of Rainy Days Juwata Station, Tarakan

Year	Month	Number of Rainy Days
2014	January	17
	February	14
	March	20
⋮	⋮	⋮
2024	October	26
	November	29
	December	26

G. Research Stages

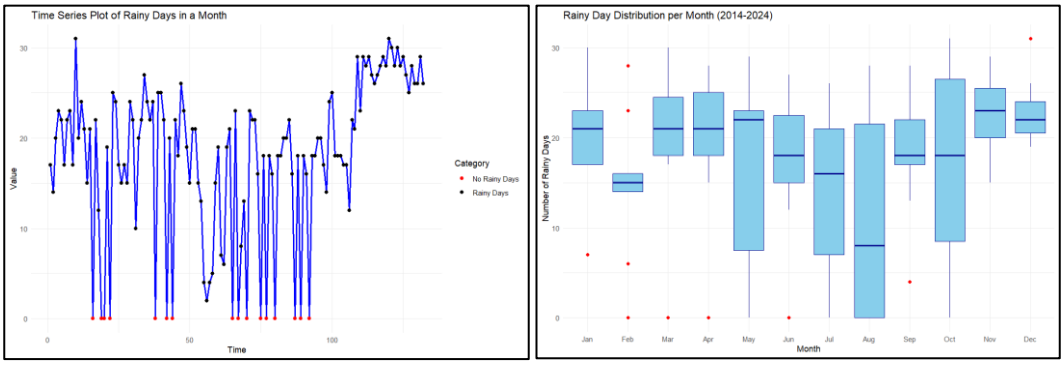
Data analysis in this study consisted of the following steps:

1. Exploring the data using Time Series Plot and Boxplot.
2. Dividing the data into in-sample and out-of-sample sets with a 90:10 proportion.
3. Plotting the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) with the in-sample data.
4. Modeling with Deep Feed Forward Neural Network (DFFNN).
5. Selecting the best model based on the smallest MAPE across all combinations of hidden layers and neurons.
6. Performing forecasting for the next 24 periods using the best model.

3. MAIN RESULTS

A. *Exploring the data using Time Series Plot and Boxplot*

This step involves visualizing the data to understand its overall trend, seasonality, and the presence of any outliers or anomalies.



(a) (b)

FIGURE 1. Data Exploration (a) time series plot; (b) boxplot

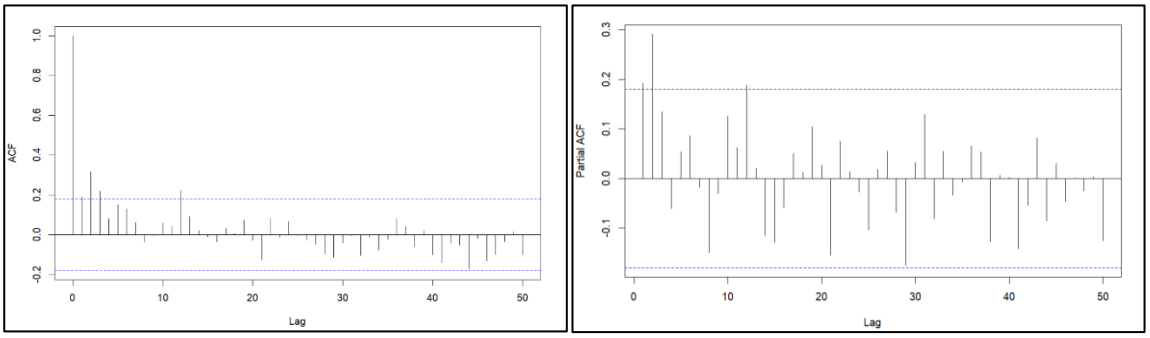
The boxplot distribution plot shows the variation in the number of monthly rainy days over the period 2014-2024. The months of October to December tend to have a consistently high number of rainy days, with a median of close to 25 rainy days per month. In contrast, months such as February, May, July and August show large variability with outliers on both the low and high sides. This indicates the instability of the rainy season in these months. The time series plot reinforces this by showing sharp fluctuations over time, including periods of no rain at all (red dots) that appear sporadically. On the other hand, the later period of the data shows a more stable and increasing trend in the number of rainy days. This pattern, which is not fully seasonal and exhibits dynamic complexity, supports the use of nonlinear models such as the Deep Feed Forward Neural Network for more accurate analysis and forecasting.

B. *Dividing the data into in-sample and out-of-sample sets with a 90:10 proportion*

The dataset is split into 90% training data (in-sample) and 10% testing data (out-sample) to ensure the model is trained on most of the data while being evaluated on unseen data.

C. *Plotting the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) with the in-sample data*

ACF and PACF are used to identify the significant lags in the data. These significant lags are then used to determine the number of input layers for the model.



(a) (b)

FIGURE 2. (a) ACF Plot; (b) PACF Plot

The ACF and PACF analysis results show that lag 1 to lag 3 has a significant autocorrelation value, while the lag after that tends to decrease and is within the significance limit. Similarly, in the PACF graph, the partial correlation looks quite strong until the 3rd lag and starts to weaken afterwards. This pattern shows that the value of rainy days in the previous three months has an influence on the current value. Based on this, three input lags ($t-1$, $t-2$, and $t-3$) were selected as input layers in the modeling.

D. Modeling with Deep Feed Forward Neural Network (DFFNN)

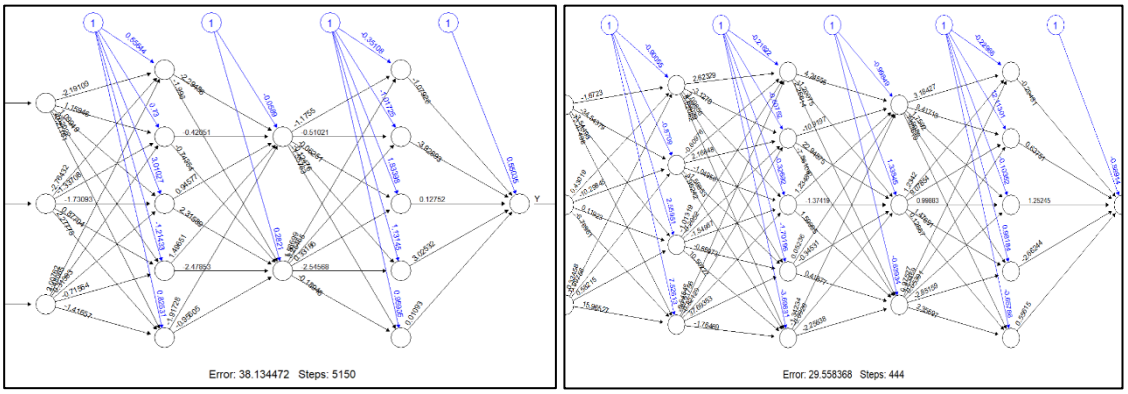
A feedforward neural network is used for modeling, with experiments conducted using 1 to 4 hidden layers and 1 to 5 neurons in each layer to evaluate the best configuration.

TABLE 2. Deep Feed Forward Neural Network Modelling Result

Model	Hidden Layer	Training			Testing			Runtime
		MSE	RMSE	MAPE	MSE	RMSE	MAPE	
Z-Score Standardized								
DFFNN "Sigmoid" & "Backprop"	H1=5 H2=4	57,363	7,574	14,640	12,740	3,569	5,088	0,2427
	H1=5 H2=2 H3=5	51,386	7,168	13,310	13,358	3,655	5,358	0,8038
	H1=4 H2=5 H3=3	58,028	7,618	14,699	14,588	3,819	5,713	0,3913

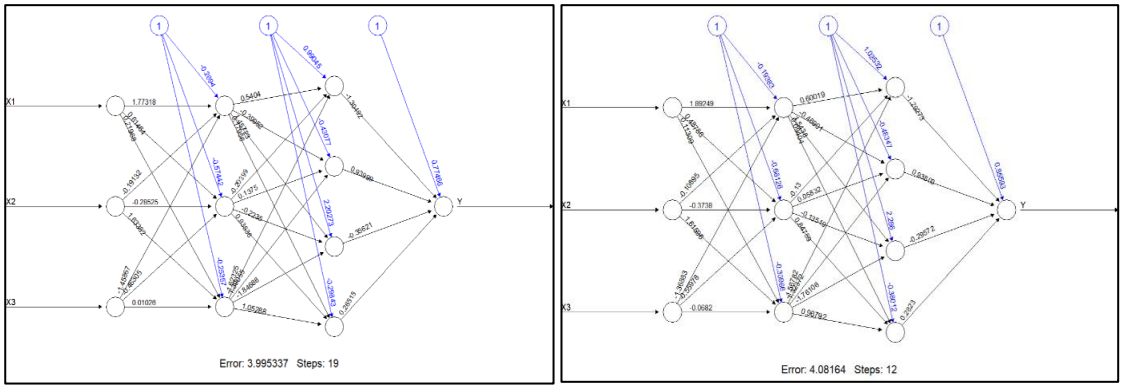
Model	Hidden Layer	Training			Testing			Runtime
		MSE	RMSE	MAPE	MSE	RMSE	MAPE	
	H4=2							
DFFNN "Sigmoid" & "Rprop+"	H1=4 H2=5	41,337	6,429	11,544	31,234	5,589	8,709	0,0493
	H1=5 H2=5 H3=5	37,682	6,138	11,114	16,592	4,073	5,307	0,0408
	H1=4 H2=5 H3=3 H4=5	39,830	6,311	11,151	17,090	4,134	5,297	0,0865
Min-Max Standardized								
DFFNN "Sigmoid" & "Backprop"	H1=3 H2=4	68,217	8,259	15,794	67,254	8,201	13,589	0,0022
	H1=5 H2=4 H3=1	69,928	8,362	15,970	102,703	10,134	16,888	0,0036
	H1=1 H2=5 H3=5 H4=5	69,622	8,344	16,095	88,401	9,402	15,640	0,0029
DFFNN "Sigmoid" & "Rprop+"	H1=3 H2=4	66,774	8,171	15,681	51,626	7,185	11,835	0,0037
	H1=2 H2=5 H3=4	71,454	8,453	16,092	108,661	10,424	17,395	0,0032
	H1=4 H2=5 H3=3 H4=2	67,332	8,206	15,936	77,017	8,776	14,563	0,004

The results of all modeling combinations are presented in Table 2, which contains the Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) values for training and testing data, as well as the computation time (runtime) of each model. Here is the architecture of the 4 models tested with the smallest MAPE Training of each combination of hidden layers and neuron.



(a) (b)

FIGURE 4. Best Each Model with Z-Score Standardized Architecture (a) DFFNN "Sigmoid" & "Backprop"; (b) DFFNN "Sigmoid" & "Rprop+"



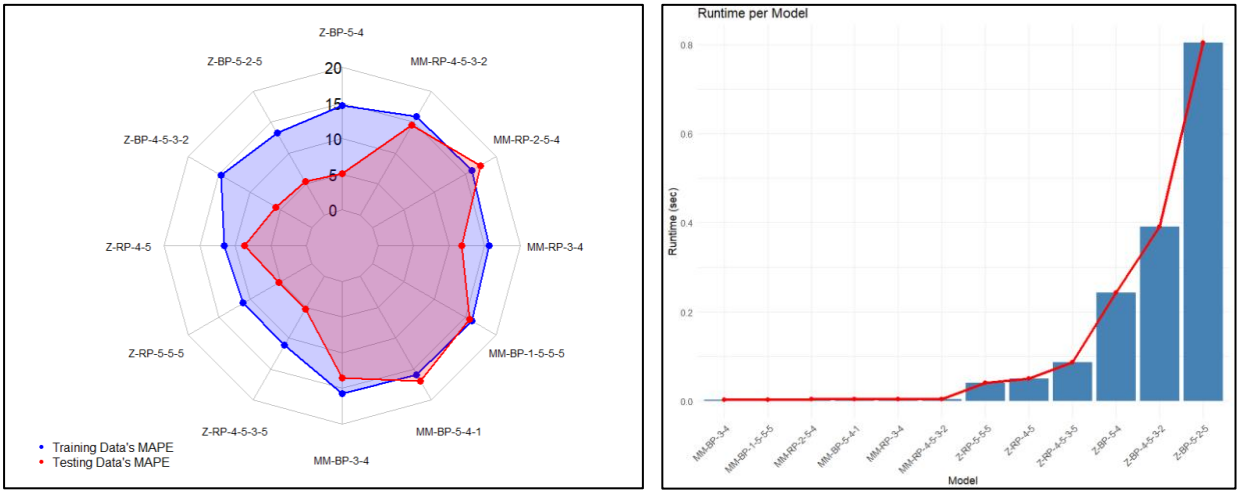
(a) (b)

FIGURE 5. Best Each Model with Min-Max Standardized Architecture (a) DFFNN "Sigmoid" & "Backprop"; (b) DFFNN "Sigmoid" & "Rprop+"

E. Best Model Selection

The model configurations are evaluated using Mean Absolute Percentage Error (MAPE), and

the model with the smallest MAPE is selected as the best.

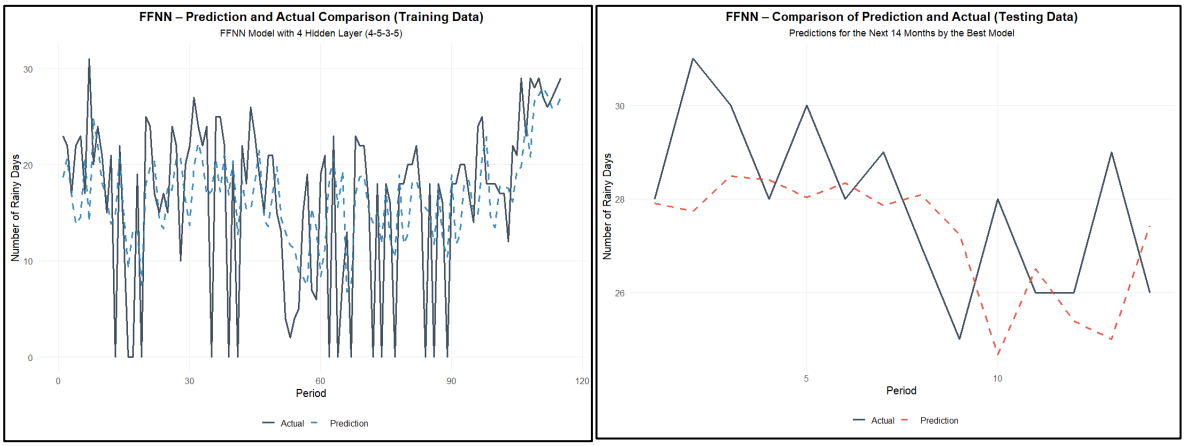


(a) (b)

FIGURE 6. (a) Radar Chart; (b) Runtime Barplot

Based on the radar chart, the model with a 4-hidden-layer architecture and a 4-5-3-5 neuron configuration, standardized using Z-Score (Z-RP-4-5-3-5), exhibits the smallest MAPE value on the testing data compared to all other models. This indicates that the model is highly capable of generalizing to unseen data. Notably, the difference between the training and testing MAPE values is minimal, further reinforcing the model's ability to perform well on both training and testing datasets without overfitting. This close alignment between training and testing performance is a key consideration in selecting this model as the best candidate.

Additionally, the runtime analysis via the bar plot highlights that the Z-RP-4-5-3-5 model has an efficient training time, which is considerably lower than some other models with even lower accuracy. This makes the model not only effective in prediction but also computationally efficient, ensuring a balanced trade-off between accuracy and resource usage. Regarding input selection, three-time lags (t-1, t-2, and t-3) were included as input layers to allow the model to capture time-dependent patterns effectively, while avoiding noise from irrelevant or insignificant lags. This configuration enables the model to leverage the temporal dependencies in the data, enhancing its predictive capability for time-series forecasting tasks.



(a) (b)

FIGURE 7. Comparison Chart (a) Training Data; (b) Testing Data

Based on the graphs and tables shown, the DFFNN model with 4 hidden layers (4-5-3-5 configuration) performs well in predicting the number of rainy days. Graph (a) displays the prediction results against the training data (in-sample), where the prediction line appears to follow the actual data pattern quite accurately. This shows that the model can learn historical patterns, including seasonal fluctuations and trends, effectively. Graph (b) shows the prediction results on the test data (out-sample). Although there is a slight deviation between the prediction and the actual data, the general movement pattern is still in line. This indicates that the model not only fits the training data but is also able to make reasonably good predictions on new data that has never been seen before.

F. Performing Forecasting for the Next 24 Periods Using the Best Model

The best model, based on the evaluation metrics, is used to forecast the next 24 periods. The forecasting results can be seen in Table 3.

Year	Month	Forecast	Year	Month	Forecast
2025	January	26.173	2026	January	17.998
	February	25.528		February	18.573
	March	23.731		March	18.594
	April	20.106		April	17.963
	May	14.832		May	17.205

Year	Month	Forecast	Year	Month	Forecast
	June	13.740		June	17.034
	July	13.818		July	17.616
	August	14.439		August	18.204
	September	14.507		September	18.467
	October	15.044		October	18.242
	November	15.969		November	17.682
	December	16.999		December	17.278

TABLE 3. Number of Rainy Days Forecasting

Based on the forecasting results presented in Table 3, the pattern of rainy days shows a seasonal trend consistent with the tropical climate in Indonesia. In general, the period from January to April exhibits a high number of rainy days, peaking in January, which indicates the peak of the rainy season in Indonesia, typically occurring between November and March. Meanwhile, the period from May to August shows a decrease in the number of rainy days, aligning with the dry season, which spans from May to October in most tropical regions. From September to December, the number of rainy days increases again, marking the beginning of the rainy season.

This seasonal pattern aligns well with the results from the best model, which is the model with a 4-hidden-layer architecture and a 4-5-3-5 neuron configuration, standardized using Z-Score (Z-RP-4-5-3-5). This model demonstrated the best performance in forecasting the number of rainy days, reflecting the characteristic seasonal pattern found in Indonesia. Furthermore, the model effectively captured time-dependent rainfall patterns, with excellent accuracy as evidenced by the low MAPE between training and testing data. The model's ability to predict fluctuations in rainy days according to seasonal trends makes it particularly useful for early warning systems, enabling more accurate anticipation of extreme conditions such as floods and landslides, which typically occur during the peak of the rainy season.

With this seasonal pattern, an early warning system is crucial for mitigating risks related to high rainfall, especially in months with a very high number of rainy days, such as January, February, and December. One potential impact of heavy rainfall is flooding, which can be prevented through the maintenance and expansion of flood control infrastructure such as drainage systems, embankments, and flood gates. Additionally, the public should be educated on evacuation routes and precautionary measures during the rainy season.

Heavy rainfall can also lead to landslides in areas with steep slopes, requiring intensive monitoring of soil conditions and the construction of infrastructure that can prevent landslides, such as slope stabilization techniques and reforestation. In the agricultural sector, farmers should be provided with information on rainfall patterns to adjust planting schedules and take preventive actions against potential crop damage from flooding or waterlogging. The impact of heavy rainfall can also disrupt transportation networks, particularly in areas prone to flooding or road damage. Therefore, it is important to carry out routine maintenance on roads and bridges, as well as provide an early warning system for hazardous travel conditions. Lastly, high rainfall can increase the risk of waterborne diseases, such as cholera, and enhance mosquito breeding sites, leading to diseases like malaria and dengue. Therefore, the early warning system should include vector control measures and water quality monitoring to protect public health. Overall, with an effective early warning system and proper mitigation strategies, the negative impacts of high rainfall can be significantly reduced, improving community preparedness for natural disasters and minimizing potential losses.

4. CONCLUSIONS

This study successfully demonstrates the application of the Deep Feed Forward Neural Network (DFFNN) for forecasting the number of rainy days, providing a robust approach to hydrological modeling. The selected model, with a 4-hidden-layer architecture and a 4-5-3-5 neuron configuration, standardized using Z-Score, emerged as the most effective in terms of forecasting accuracy, as evidenced by the smallest MAPE value on both training and testing datasets. The model's ability to generalize well to unseen data without overfitting is a critical advantage, ensuring that it provides reliable predictions across different time periods.

The forecasting results reveal a seasonal pattern in the number of rainy days, consistent with the tropical climate of Indonesia, where the rainy season peaks from January to March and decreases during the dry season. This model captures the time-dependent nature of rainfall data effectively, making it suitable for applications in early warning systems. By accurately predicting high rainfall months, the model can inform preventive measures against flooding, landslides, and other weather-related disasters. Moreover, the model's ability to predict variations in rainfall supports proactive planning in sectors such as agriculture and transportation.

However, the study also acknowledges several limitations, including the challenges in determining appropriate input variables, optimizing hyperparameters, and selecting the best

network architecture. The performance of the model is highly sensitive to these choices, and future studies could benefit from exploring more advanced techniques for hyperparameter tuning and input selection to enhance the model's accuracy and robustness further.

Overall, the study highlights the potential of machine learning models, particularly DFFNNs, in enhancing the accuracy and efficiency of hydrological forecasting. The results contribute to the development of more effective tools for managing the risks associated with extreme weather events, improving disaster preparedness and response strategies.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] C.A. Zafra-Mejía, H.A. Rondón-Quintana, C.F. Urazán-Bonells, Arima and Tfarima Analysis of the Main Water Quality Parameters in the Initial Components of a Megacity's Drinking Water Supply System, Hydrology 11 (2024), 10. https://doi.org/10.3390/hydrology11010010.

[2] H. Hartmann, Comparison of Precipitation Rates from Global Datasets for the Five-Year Period From 2019 to 2023, Hydrology 12 (2025), 4. https://doi.org/10.3390/hydrology12010004.

[3] M.J.S. Safari, S. Rahimzadeh Arashloo, A. Danandeh Mehr, Rainfall-runoff Modeling Through Regression in the Reproducing Kernel Hilbert Space Algorithm, J. Hydrol. 587 (2020), 125014. https://doi.org/10.1016/j.jhydrol.2020.125014.

[4] H. Wong, W. Ip, R. Zhang, J. Xia, Non-parametric Time Series Models for Hydrological Forecasting, J. Hydrol. 332 (2007), 337-347. https://doi.org/10.1016/j.jhydrol.2006.07.013.

[5] T. Kim, M. Jehanzaib, Drought Risk Analysis, Forecasting and Assessment Under Climate Change, Water 12 (2020), 1862. https://doi.org/10.3390/w12071862.

[6] Mamenun, Y. Koesmaryono, A. Sopaheluwakan, R. Hidayati, B.D. Dasanto, R. Aryati, Spatiotemporal Characterization of Dengue Incidence and Its Correlation to Climate Parameters in Indonesia, Insects 15 (2024), 366. https://doi.org/10.3390/insects15050366.

[7] I. Kamil, R. Jayadi, A.P. Rahardjo, Impact of Climate Change on Inundation in the Katingan Tidal Agricultural Lowland, Central Kalimantan, ASEAN Eng. J. 14 (2024), 197-206. https://doi.org/10.11113/aej.v14.20495.

[8] A. Sudradjat, B.S. Muntalif, N. Marasabessy, F. Mulyadi, M.I. Firdaus, Relationship Between Chlorophyll-A, Rainfall, and Climate Phenomena in Tropical Archipelagic Estuarine Waters, Heliyon 10 (2024), e25812. https://doi.org/10.1016/j.heliyon.2024.e25812.

[9] C. Payus, L. Ann Huey, F. Adnan, A. Besse Rimba, G. Mohan, S. Kumar Chapagain, G. Roder, A. Gasparatos, K. Fukushi, Impact of Extreme Drought Climate on Water Security in North Borneo: Case Study of Sabah, Water 12 (2020), 1135. https://doi.org/10.3390/w12041135.

[10] D. Munandar, B.N. Ruchjana, A.S. Abdullah, H.F. Pardede, Literature Review on Integrating Generalized Space-Time Autoregressive Integrated Moving Average (GSTARIMA) and Deep Neural Networks in Machine Learning for Climate Forecasting, Mathematics 11 (2023), 2975. https://doi.org/10.3390/math11132975.

[11] X. Zhang, Q. Zhang, G. Zhang, Z. Nie, Z. Gui, A Hybrid Model for Annual Runoff Time Series Forecasting Using Elman Neural Network with Ensemble Empirical Mode Decomposition, Water 10 (2018), 416. https://doi.org/10.3390/w10040416.

[12] G. Zhang, Time Series Forecasting Using a Hybrid Arima and Neural Network Model, Neurocomputing 50 (2003), 159-175. https://doi.org/10.1016/s0925-2312(01)00702-0.

[13] I.D. Mienye, T.G. Swart, G. Obaido, Recurrent Neural Networks: a Comprehensive Review of Architectures, Variants, and Applications, Information 15 (2024), 517. https://doi.org/10.3390/info15090517.

[14] Y. Wang, L. Huang, Fourier series neural networks for regression, in: 2018 IEEE International Conference on Applied System Invention (ICASI), IEEE, 2018, pp. 716-719. https://doi.org/10.1109/icasi.2018.8394358.

[15] M. Mislan, A.T.R. Dani, Forecasting Maximum Water Level Data for Post Sangkuliman Using an Artificial Neural Network Backpropagation Algorithm, J. Teor. Apl. Mat. 8 (2024), 465. https://doi.org/10.31764/jtam.v8i2.20112.

[16] S. Suhartono, D.E. Ashari, D.D. Prastyo, H. Kuswanto, M.H. Lee, Deep Neural Network for Forecasting Inflow and Outflow in Indonesia, Sains Malays. 48 (2019), 1787-1798. https://doi.org/10.17576/jsm-2019-4808-26.

[17] M. Khashei, M. Bijari, An Artificial Neural Network (p,d,q) Model for Timeseries Forecasting, Expert Syst. Appl. 37 (2010), 479-489. https://doi.org/10.1016/j.eswa.2009.05.044.

[18] I.E. Livieris, E. Pintelas, S. Stavroyiannis, P. Pintelas, Ensemble Deep Learning Models for Forecasting Cryptocurrency Time-Series, Algorithms 13 (2020), 121. https://doi.org/10.3390/a13050121.

[19] P.P. Phyo, Y. Byun, Hybrid Ensemble Deep Learning-Based Approach for Time Series Energy Prediction, Symmetry 13 (2021), 1942. https://doi.org/10.3390/sym13101942.

[20] K. He, Q. Yang, L. Ji, J. Pan, Y. Zou, Financial Time Series Forecasting with the Deep Learning Ensemble Model, Mathematics 11 (2023), 1054. https://doi.org/10.3390/math11041054.

[21] I. Nusrat, S. Jang, A Comparison of Regularization Techniques in Deep Neural Networks, Symmetry 10 (2018), 648. https://doi.org/10.3390/sym10110648.

[22] R.P. Permata, R. Ni'mah, A.T.R. Dani, Daily Rainfall Forecasting with Arima Exogenous Variables and Support Vector Regression, J. Varian 7 (2024), 177-188. https://doi.org/10.30812/varian.v7i2.3202.

[23] M. Narvekar, P. Fargose, Daily Weather Forecasting Using Artificial Neural Network, Int. J. Comput. Appl. 121 (2015), 9-13. https://doi.org/10.5120/21830-5088.

[24] T.T.K. Tran, S.M. Bateni, S.J. Ki, H. Vosoughifar, A Review of Neural Networks for Air Temperature Forecasting, Water 13 (2021), 1294. https://doi.org/10.3390/w13091294.

[25] G. Zhang, M. Qi, Neural Network Forecasting for Seasonal and Trend Time Series, Eur. J. Oper. Res. 160 (2005), 501-514. https://doi.org/10.1016/j.ejor.2003.08.037.

[26] P. Escudero, W. Alcocer, J. Paredes, Recurrent Neural Networks and Arima Models for Euro/Dollar Exchange Rate Forecasting, Appl. Sci. 11 (2021), 5658. https://doi.org/10.3390/app11125658.

[27] W. Sulandari, Subanar, M.H. Lee, P.C. Rodrigues, Indonesian Electricity Load Forecasting Using Singular Spectrum Analysis, Fuzzy Systems and Neural Networks, Energy 190 (2020), 116408. https://doi.org/10.1016/j.energy.2019.116408.

[28] N. Suhermi, Suhartono, D.D. Prastyo, B. Ali, Roll Motion Prediction Using a Hybrid Deep Learning and Arima Model, Procedia Comput. Sci. 144 (2018), 251-258. https://doi.org/10.1016/j.procs.2018.10.526.

[29] Y. Wang, R. Zou, F. Liu, L. Zhang, Q. Liu, A Review of Wind Speed and Wind Power Forecasting with Deep Neural Networks, Appl. Energy 304 (2021), 117766. https://doi.org/10.1016/j.apenergy.2021.117766.

[30] S. Chen, G. Fang, X. Huang, Y. Zhang, Water Quality Prediction Model of a Water Diversion Project Based on the Improved Artificial Bee Colony–backpropagation Neural Network, Water 10 (2018), 806. https://doi.org/10.3390/w10060806.

[31] A. Atiya, S. El-Shoura, S. Shaheen, M. El-Sherif, A Comparison Between Neural-Network Forecasting Techniques-Case Study: River Flow Forecasting, IEEE Trans. Neural Netw. 10 (1999), 402-409. https://doi.org/10.1109/72.750569.

[32] C. Chen, J. Lin, Applying Rprop Neural Network for the Prediction of the Mobile Station Location, Sensors 11 (2011), 4207-4230. https://doi.org/10.3390/s110404207.

[33] K.R. Kashyzadeh, N. Amiri, S. Ghorbani, K. Souri, Prediction of Concrete Compressive Strength Using a Back-Propagation Neural Network Optimized by a Genetic Algorithm and Response Surface Analysis Considering the Appearance of Aggregates and Curing Conditions, Buildings 12 (2022), 438. https://doi.org/10.3390/buildings12040438.

[34] M. Fayaz, H. Shah, A.M. Aseere, W.K. Mashwani, A.S. Shah, A Framework for Prediction of Household Energy Consumption Using Feed Forward Back Propagation Neural Network, Technologies 7 (2019), 30. https://doi.org/10.3390/technologies7020030.

[35] L. Peng, D. Gao, Y. Bai, A Study on Standardization of Security Evaluation Information for Chemical Processes Based on Deep Learning, Processes 9 (2021), 832. https://doi.org/10.3390/pr9050832.

[36] H. Anysz, A. Zbiciak, N. Ibadov, The Influence of Input Data Standardization Method on Prediction Accuracy of Artificial Neural Networks, Procedia Eng. 153 (2016), 66-70. https://doi.org/10.1016/j.proeng.2016.08.081.