Jurnal Matematika dan Statistika serta Aplikasinya, Vol. 14, No. 2, 258-268, 2026
DOI: https://doi.org/10.24252/msa.v14i2.70441

Rainfall Forecasting Using Generalized Feedforward Neural Network (GFNN) (Case Study: Rainfall Forecasting in Bandung City)

Reva Azzahra Pratama[1]*, Dewi Rachmatin[2], Fitri Rahmawati[3]
[1,2,3]*Mathematics Study Program, Universitas Pendidikan Indonesia, Indonesia*
***Corresponding author**: revaazzahrap1804@upi.edu

***Submission date**: 13 July 2026, **Revision**: 20 August 2026, **Accepted**: 14 September 2026

ABSTRACT
Accurate short-term rainfall forecasting remains challenging because daily rainfall exhibits high variability and nonlinear dependence on meteorological conditions. This study investigates the applicability of a Generalized Feedforward Neural Network (GFNN) for one-step-ahead daily rainfall forecasting in Bandung City using five meteorological variables rainfall, air temperature, relative humidity, wind speed, and wind direction from NASA POWER data covering 2016–2025. The novelty of this study lies in the application and empirical evaluation of the GFNN shortcut-connection architecture to multivariate daily rainfall forecasting in Bandung, where the model simultaneously exploits direct input–output relationships and nonlinear transformations through the hidden layer. The study further evaluates whether GFNN provides predictive information beyond a persistence baseline rather than assessing the model solely through its absolute forecasting error. After percentile-based clipping, min-max normalization, and sliding-window input construction, six GFNN configurations with 6–11 hidden neurons were evaluated. The selected 5-6-1 architecture achieved RMSE values of 8.9018 mm for the training data and 9.8949 mm for the testing data. On the same testing period, GFNN reduced the RMSE of the persistence baseline from 12.3140 mm to 9.8949 mm, corresponding to a 19.6% reduction in prediction error. The results demonstrate that GFNN can capture useful predictive patterns in Bandung's daily rainfall data and provides a measurable improvement over the naïve persistence approach. However, the model tends to underestimate rainfall above approximately 40 mm, indicating that its current formulation is more suitable for general short-term rainfall forecasting than for extreme-rainfall prediction. The findings provide an empirical basis for further development of GFNN-based rainfall forecasting using longer temporal windows, imbalance-aware loss functions, and ground-station observations.

KEYWORDS
Generalized Feedforward Neural Network, rainfall, forecasting, Root Mean Square Error, Bandung City.

1. INTRODUCTION

Rainfall is one of the elements of the hydrological cycle that is most difficult to predict because it varies greatly across both space and time [1]. This uncertainty has a significant impact on hydrometeorological disaster mitigation, particularly floods, since unpredictable rainfall intensity can damage infrastructure, disrupt mobility, and harm the economy and social life. Geomorphologically, Bandung City forms a basin surrounded by mountains and river catchment areas [2], causing rainwater from higher elevations to flow and accumulate in the central and southern parts of the city, and several areas such as Pasteur, Pagarsih, Gedebage, and Antapani frequently experience flooding as a result of this basin topography [3]. According to the Bandung City Fire and Disaster Management Agency (Diskar PB), six floods and six landslides occurred in various locations up

to February 2025 due to high rainfall intensity [4], underscoring the urgency of an accurate rainfall forecasting model to support disaster-risk mitigation in the city.

Rainfall forecasting remains challenging because rainfall is a complex phenomenon influenced by many meteorological factors such as temperature, air pressure, humidity, and wind [5]. Statistical approaches such as ARIMA and linear regression are commonly used because they can capture seasonal patterns in time series data, but they are limited in recognizing the nonlinear patterns and extreme fluctuations that frequently appear in rainfall data [6]. Singular Spectrum Analysis (SSA) has been used to decompose trend, oscillation, and noise components in rainfall data, but it still struggles with sudden pattern changes and extreme fluctuations [7]. Fuzzy Time Series (FTS) has also been applied for rainfall forecasting without requiring a specific distributional assumption, but its accuracy strongly depends on the selection of linguistic intervals, which is often subjective [8].

The limitations of statistical methods in capturing complex relationships among meteorological variables have driven the use of Artificial Neural Network (ANN)-based approaches, which can recognize nonlinear patterns more flexibly through layered processing units and nonlinear activation functions. Hybrid approaches such as ARIMA-NN have been used to combine the strength of ARIMA's linear pattern recognition with the nonlinear recognition capability of ANN [6]. Among the more specific ANN variants, the Multilayer Perceptron (MLP) is the most widely used base architecture, but its purely feedforward structure tends to require relatively long training times and is prone to overfitting when the number of parameters is not proportional to the amount of data [9]. The Time-Delay Neural Network (TDNN) introduces a time-delay window at the input layer to account for short-term temporal patterns, although its performance depends on having a large amount of historical data to fill that window adequately [9]; the Nonlinear Autoregressive Exogenous Neural Network (NARX-NN) extends this capability by linking rainfall with exogenous variables simultaneously, but adds structural complexity and more parameters to train [10]; and the Focused Time-Delay Neural Network (FTDNN) concentrates its delay taps only at the input layer, making it more compact than TDNN, but this characteristic also makes it prone to overfitting when the data contains noise, as is common in rainfall series [12]. The Generalized Feedforward Neural Network (GFNN) offers an alternative by allowing shortcut connections that skip one or more hidden layers, enabling the network to capture more complex input-output relationships than conventional feedforward architectures [11]. The differences in structural characteristics, data requirements, and noise robustness among these methods are summarized comparatively in **Table 1**.

The comparison in **Table 1** shows that classical statistical methods (ARIMA, SSA, FTS) excel in interpretability and require relatively little data, but are consistently limited in capturing the nonlinear patterns and extreme fluctuations that characterize daily rainfall. More complex ANN variants such as TDNN, NARX-NN, and FTDNN improve temporal or exogenous modeling capability, but the resulting trade-off is an increased need for historical data, structural complexity, or sensitivity to noise. The implication for the choice of approach in this study is that an architecture is needed that can capture linear and nonlinear components simultaneously without adding structural complexity or excessive historical-data requirements for the one-step-ahead (H+1) forecasting case. GFNN was chosen in this study because its shortcut connection structurally accommodates this need: the linear signal from input to output is carried directly, while the hidden layer still captures the nonlinear component, so the model does not need to rely entirely on network depth or a long time window to represent complex input-output relationships. This characteristic makes GFNN a methodologically relevant choice for the short-term rainfall forecasting problem addressed in this study, without claiming that this architecture is empirically superior to all the methods discussed, since this study does not perform a direct comparison against them. Compared with LSTM or GRU, which require more parameters and training data to make optimal use of their gating mechanisms [15], GFNN was chosen because its simpler structure is considered adequate for the one-step-ahead (H+1) forecasting scheme with a short input window used in this study.

Previous studies have shown that GFNN can produce more accurate predictions than other ANN methods, including for daily rainfall forecasting in Bangkok, Thailand, where GFNN outperformed the persistence method for forecasts 1 to 6 hours ahead [11]. GFNN has also demonstrated strong generalization ability in other fields, such as epilepsy diagnosis with a classification accuracy of 86.67% [13], modeling wind-speed distribution for turbine energy-output estimation [16], and hybrid wavelet-based runoff forecasting [17]. Although it has been applied in hydrology and time series forecasting more broadly, no study has specifically applied GFNN to daily rainfall forecasting in Bandung City, which constitutes the research gap that this study aims to address. This study therefore applies a GFNN model to forecast daily one-step-ahead (H+1) rainfall in Bandung City using ten years of NASA POWER satellite data (2016-2025), evaluates its accuracy using the Root Mean Square Error (RMSE)

Table 1. Critical Comparison of Rainfall Forecasting Methods

Method	Main Characteristics	Strengths	Limitations	Relevance to Short-Term Rainfall Forecasting
ARIMA / Linear Regression [6]	Linear statistical model based on autocorrelation and seasonal components in historical data	Simple to interpret; relatively low historical-data requirement; able to capture seasonal patterns	Limited in recognizing nonlinear patterns and extreme fluctuations	Adequate as a basic benchmark, but less suitable for highly fluctuating rainfall
SSA [7]	Decomposes the time series into trend, oscillation, and noise components	Separates signal from noise components without requiring a specific distributional assumption	Struggles with sudden pattern changes and extreme fluctuations	Useful for exploring series components, but less responsive to short-term extreme events
FTS [8]	Modeling based on fuzzy sets and linguistic intervals	Does not require a data-distribution assumption	Accuracy depends heavily on the choice of linguistic intervals, which tends to be subjective	Flexible, but sensitive to heuristic parameter settings
MLP [9]	Standard feedforward ANN with sequential connections between layers	Able to model nonlinear relationships in general	Relatively long training time; prone to overfitting	The basis of ANN architecture, but lacks an explicit pathway for the linear component
TDNN [9]	MLP with a time-delay window at the input layer	Accounts for short-term temporal patterns	Requires a large amount of historical data to adequately fill the time window	Relevant for lagged patterns, but less efficient for an H+1 scheme with a short window
NARX-NN [10]	Autoregressive ANN linking the target with exogenous variables	Able to relate rainfall to external meteorological variables	More complex structure and more parameters	Suitable when exogenous relationships dominate, but adds computational load
FTDNN [12]	TDNN with delay taps centered at the input layer	More compact structure than TDNN	Prone to overfitting when the data contains noise	Structurally efficient, but less stable for noisy rainfall data
GFNN (this study) [11, 13, 14]	MLP with a shortcut connection from the input directly to the output, alongside the hidden-layer pathway	Captures linear and nonlinear components simultaneously without adding recurrent complexity	Performance is sensitive to the choice of hidden-neuron count and weight initialization	The two-pathway structure suits the needs of H+1 forecasting, combining linear and nonlinear components without long-term temporal dependence

metric, and produces a one-day-ahead rainfall forecast as a demonstration of the model's practical application.

2. LITERATURE REVIEW

2.1 Rainfall and Time Series Forecasting

Rainfall is the height of rainwater collected in a rain gauge placed on a flat, non-absorbent, non-infiltrating, and non-flowing surface, measured in millimeters (mm), where 1 mm of rainfall is equivalent to 1 liter of water collected over an area of 1 square meter [18]. Time series forecasting requires attention to the underlying data pattern horizontal, trend, seasonal, or cyclical since the accuracy of a forecasting method depends on how well it reproduces the historical behavior of the series. Quantitative forecasting methods, which rely on historical numerical data, are generally categorized separately from qualitative methods that rely on expert judgment.

2.2 Artificial Neural Network (ANN) and Generalized Feedforward Neural Network (GFNN)

An Artificial Neural Network (ANN) is a branch of artificial intelligence that mimics the workings of biological neural networks through a parallel computing system consisting of many simple interconnected processing units (artificial neurons) [19]. A standard ANN architecture consists of an input layer, one or more hidden layers, and an output layer, with each neuron applying a (generally nonlinear) activation function to its weighted input to produce an output signal [20]. The GFNN is a development of the Multilayer Perceptron (MLP) that differs in the flexibility of its connection structure: whereas MLP only allows sequential, layer-to-layer connections, GFNN allows shortcut connections that skip one or more hidden layers, so that the network simultaneously carries a nonlinear signal through the hidden layer and a linear signal directly from input to output [14]. This dual-pathway structure improves the model's ability to capture both linear and nonlinear components of a complex data-generating process.

The activation function used on both the hidden and output layers in this study is the hyperbolic tangent (tanh) function, chosen because its output range of [-1,1] matches the range used for data normalization and because its zero-centered property helps accelerate training convergence compared to functions such as sigmoid [21]. The tanh function is defined as:

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

where $f(x)$ is the activation output within the interval [-1,1], x is the net input signal to the neuron, and eis Euler's constant ($e \approx 2.71828$). GFNN is typically trained using the backpropagation algorithm, which propagates the prediction error backward through the network to update the weights of both the indirect (hidden-layer) and direct (shortcut) pathways [11].

3. METHODOLOGY

3.1 Research Data

The data used in this study are secondary daily time-series data obtained from the NASA POWER (Prediction of Worldwide Energy Resources) satellite platform for the geographic coordinates of Bandung City (-6.9175° latitude, 107.6191° longitude, elevation 921.68 m), covering the period 1 January 2016 to 31 December 2025, for a total of 3,653 daily records. Five variables were used: daily rainfall (x_1, mm/day), air temperature at 2 m (x_2, °C), relative humidity at 2 m (x_3, %), wind speed at 10 m (x_4, m/s), and wind direction at 10 m (x_5, °), with rainfall serving as both a predictor and the target variable for the following day ($t+1$).

3.2 Data Pre-Processing

Data pre-processing was carried out in four sequential steps. First, all 3,653 records were checked for the NASA POWER missing-value code (-999); none were found, so the dataset was complete and no imputation was required. Second, extreme values were handled using percentile-based clipping, chosen over trimming because it preserves the sample size and the continuity of the daily series [22]:

$$x_i^{*} = \min\left(x_i, P_{99,9}\right) \tag{2}$$

where x_i is the original value, P99.9 is the 99.9th percentile of the data distribution, and x_i^{*} is the value after clipping. Applying this threshold to the rainfall series yielded a clipping value of 81.94 mm/day, affecting only 4 of 3,653 records (0.11%); the mean rainfall shifted only slightly from 9.14 mm to 9.07 mm and the standard deviation from 11.55 mm to 10.78 mm, so the dominant right-skewed shape of the distribution was preserved.

Third, because all five variables have very different scales and the tanh activation function has an output range of [-1,1], all variables were normalized to [-1,1] using min-max normalization, with the minimum and maximum values computed only from the training subset to avoid data leakage:

$$x_{norm} = 2 \times \left(\frac{x - x_{min}}{x_{max} - x_{min}}\right) - 1 \tag{3}$$

where x is the original value, x_{min} and x_{max} are the minimum and maximum values of the training data, and x_{norm} is the normalized result. Fourth, input–target pairs were formed using a sliding window of size 1 [23], where the five meteorological variables on day t serve as the input vector and the rainfall value on day $t+1$ is the target, producing 3,652 usable samples. The dataset was then split sequentially (to preserve temporal order) into 90% training data (3,286 samples; 1 Jan 2016–28 Dec 2024) and 10% testing data (366 samples; 29 Dec 2024–31 Dec 2025).

3.3 GFNN Model Architecture and Training

The GFNN architecture consists of an input layer, one hidden layer, and an output layer, with a shortcut connection linking the input layer directly to the output layer [13]. The number of hidden neurons was searched within the range suggested by Fletcher and Goss [24], i.e., $2\sqrt{n+m}$ to $2n+1$, where n is the number of input neurons and m is the number of output neurons. For

the hidden-layer output, each neuron j receives a weighted sum of the inputs plus a bias, passed through the *tanh* activation function:

$$h_j = tanh\left(\sum_{i=1}^{n} v_{ij}x_i + b_j\right) \tag{4}$$

where v_{ij} is the weight from input neuron i to hidden neuron j, and b_j is the hidden-layer bias. The network output combines the nonlinear signal from the hidden layer with the linear signal carried by the shortcut connection:

$$\hat{y}_{norm} = tanh\left(\sum_{j=1}^{p} w_j h_j + \sum_{i=1}^{n} u_i x_i + b\right) \tag{5}$$

where w_j is the weight from hidden neuron j to the output neuron, u_i is the shortcut-connection weight from input i directly to the output, and b is the output bias. During training, the squared-error loss is minimized using the backpropagation algorithm with gradient descent, in which the output error is propagated backward to compute the gradients for both the indirect path (v, w) and the direct path (u), and all weights and biases are updated online (per sample) with a learning rate α [11, 15]. Training stops when either the maximum number of epochs or a target error is reached, whichever occurs first.

After training, the network's normalized output $\hat{y}_{norm}$ is transformed back to the original scale (mm) through denormalization, obtained analytically as the inverse of equation (3):

$$\hat{y} = \frac{(\hat{y}_{norm} + 1)}{2} \bullet (x_{max} - x_{min}) + x_{min} \tag{6}$$

Model performance is evaluated using the Root Mean Square Error (RMSE) between the denormalized predictions and the actual rainfall values on the testing data:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{7}$$

where n is the number of testing samples, $\hat{y}_i$ is the i-th predicted value, and y_i is the i-th actual value in millimeters. As an additional benchmark, model performance is compared against a persistence (naïve) forecasting model, defined as $y_{t+1} = y_t$, which is a common minimum baseline in time-series forecasting evaluation [25].

4. RESULT & DISCUSSION

4.1 Data Description and Exploration

Table 2 presents the descriptive statistics of the five variables computed from the full 3,653 day dataset.

Table 2. Descriptive Statistics of the Input and Target Variables

Variable	Description	Min	Max	Mean	Std Dev	Unit
x_1	Rainfall	0.00	226.27	9.14	11.55	mm/day
x_2	Air Temperature	17.90	24.06	21.68	0.88	°C
x_3	Humidity	72.22	96.94	88.85	3.53	%
x_4	Wind Speed	0.61	6.87	2.20	0.84	m/s
x_5	Wind Direction	0.20	359.90	162.11	82.51	°

Based on **Table 2**, the rainfall variable (x_1) has a minimum value of 0.00 mm, a maximum value of 226.27 mm, a mean (μ) of 9.14 mm, and a standard deviation (σ) of 11.55 mm. The ratio of the standard deviation to the mean gives a Coefficient

of Variation (CV) of 1.26, or 126.3%. A CV exceeding 100% indicates that the standard deviation is larger than the mean, reflecting a high degree of variation and a wide spread [26].

Table 2 also shows that the five variables have very different scales: rainfall ranges from 0-226.27 mm, while wind direction reaches nearly 360°. This substantial difference in scale is the fundamental reason why the data must be normalized before being fed into the neural network, since without normalization, variables with larger scales would dominate the learning process and disrupt the model's convergence.

A similar pattern appears when comparing the maximum value with the mean: the maximum value (226.27 mm) is more than 24 times the mean (9.14 mm), far exceeding what a symmetric distribution would show. Together with the CV above, this ratio indicates a right-skewed rainfall distribution, a characteristic that was taken into account when applying percentile-based clipping during data preprocessing. **Figure 1** shows the Pearson correlation matrix among the five input variables used in this study.

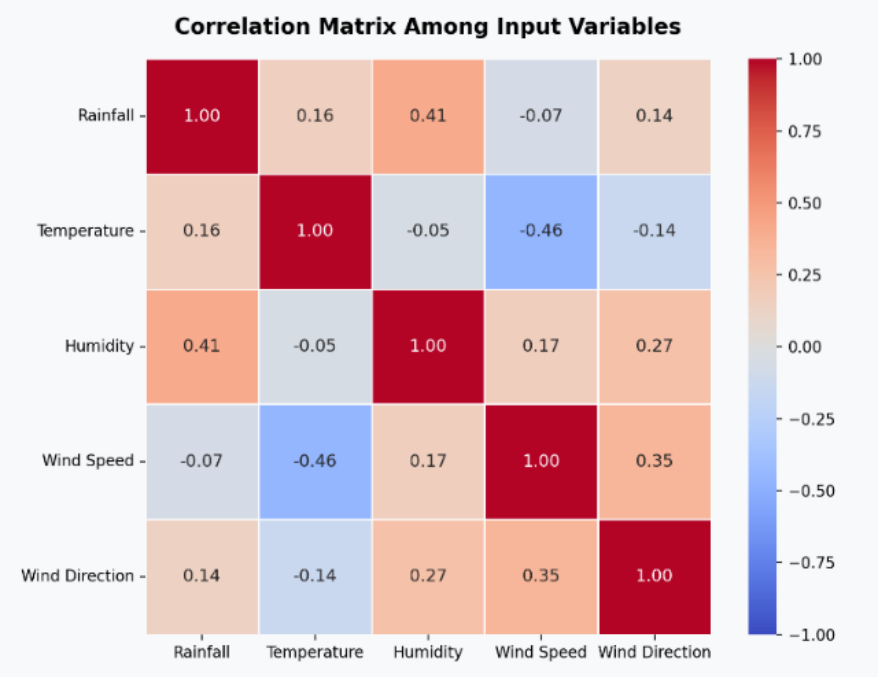


Figure 1. Correlation Matrix Among the Input Variables

Figure 1 shows that no pair of variables has a correlation exceeding ± 0.7, indicating no problematic multicollinearity. Rainfall has its strongest correlation with humidity ($r = 0.41$), while its correlations with temperature, wind speed, and wind direction are low. This low Pearson correlation reflects a weak linear association between the variables, not a direct indicator of the underlying form of the relationship-the Pearson coefficient is inherently sensitive only to linear patterns, so relationships that are non-monotonic, conditional, or involve interactions between variables are not adequately captured by this measure. This characteristic is one of the main considerations behind choosing a neural-network-based architecture such as GFNN, which is structurally designed to accommodate both linear and nonlinear relationships without requiring any prior assumption about the form of the relationship.

4.2 GFNN Architecture Determination

Using the Fletcher and Goss [24] rule with $n = 5$ input neurons and m=1 output neuron, the recommended hidden-neuron search range is 6 to 11. Six independent GFNN configurations were trained within this range while holding all other hyperparameters fixed ($\alpha = 0.01$, maximum epoch $= 1,000$, window size $= 1$, split ratio 90:10). The results are summarized in **Table 3**.

All configurations produced a positive difference between the test RMSE and the training RMSE, indicating that none of the architectures tested overfit the training data. The six configurations produced test RMSE values ranging from 9.8949 mm to 9.9807 mm, a relatively narrow range that did not increase monotonically with the number of hidden neurons. Because each

Table 3. Hidden-Neuron Selection Experiment Results

Hidden Neurons	Testing RMSE (mm)	Training RMSE (mm)	Difference (Test-Train)	Overfitting Indication
6	9.8949	8.9018	0.9931	No
7	9.9807	8.9272	1.0535	No
8	9.9742	8.9120	1.0622	No
9	9.9732	8.8868	1.0864	No
10	9.9362	8.8888	1.0474	No
11	9.9241	8.8720	1.0521	No

configuration was trained only once, without repeated runs or uncertainty estimation, this small RMSE difference between architectures is best interpreted as roughly comparable performance among the evaluated configurations rather than definitive proof that one architecture is superior. Based on these results, the 6-neuron configuration produced the lowest test RMSE (9.8949 mm) among the configurations evaluated and was therefore selected as the final architecture, yielding a 5-6-1 network structure with a shortcut connection from all five inputs directly to the output. **Figure 2** illustrates this selected architecture.

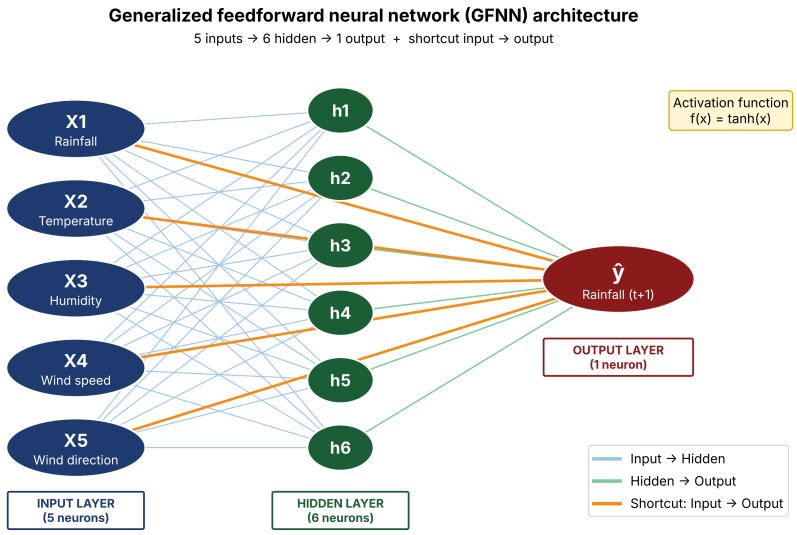


Figure 2. GFNN 5-6-1 Model Architecture

4.3 Model Training and Evaluation

The selected model was trained for a full 1,000 epochs using online (per-sample) backpropagation with $\alpha = 0.01$. **Figure 3** shows the training RMSE convergence curve on the normalized scale.

The curve shows three phases: a sharp decline from around 0.30 to 0.223 during epochs 0-200, as the randomly initialized weights moved quickly toward better values; a slowdown during epochs 200-600 as the weights approached the minimum region; and a stable phase from epoch 600 to the end, finishing at a normalized RMSE of 0.21788 at epoch 1,000. Because this value did not reach the error target of 0.01, training stopped due to the epoch limit rather than convergence to the target. This smooth decline without large oscillations indicates a stable training process under the learning rate $\alpha = 0.01$ used, although the smoothness of the curve alone does not prove that this α value is optimal, since this study did not perform a separate learning-rate sensitivity analysis. **Table 4** compares the training and test RMSE on the original (denormalized) scale for the final 5-6-1 model.

The small difference between the training RMSE (8.9018 mm) and the test RMSE (9.8949 mm) again shows that the model

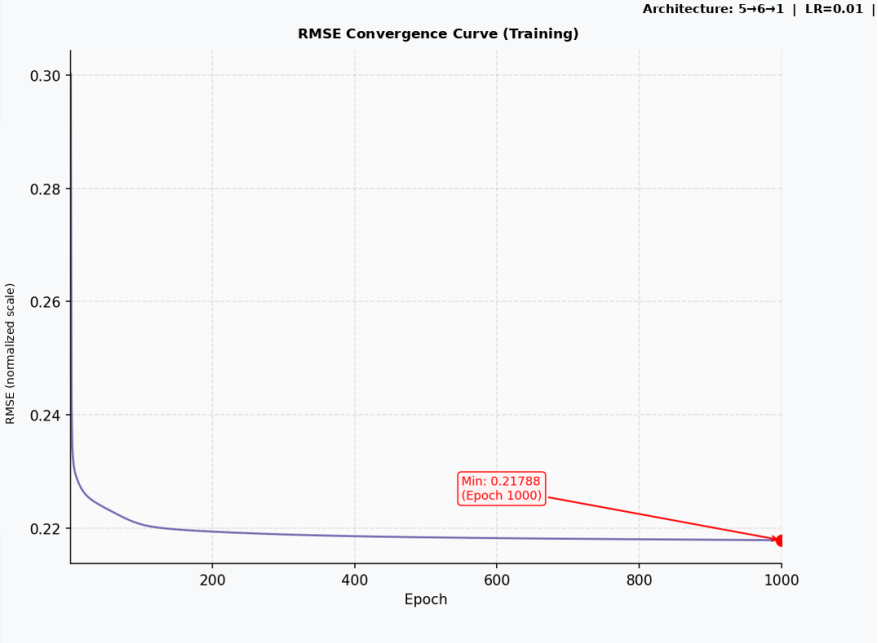


Figure 3. RMSE Training Convergence Curvee

Table 4. Comparison of Training and Testing RMSE for the GFNN 5-6-1 Model

Data Subset	Number of Samples	RMSE (Original Scale / mm)
Training Data	3,286	8.9018
Testing Data	366	9.8949

generalizes reasonably well and does not overfit. Because the rainfall series has a very high coefficient of variation (126.3%), the RMSE value cannot be interpreted on its own; the GFNN model is therefore compared with a persistence (naïve) baseline, which simply assumes that tomorrow's rainfall equals today's. The persistence model produced an RMSE of 12.3140 mm on the same 366 test samples, or GFNN reduced the RMSE by 2.4191 mm (19.6%) relative to the persistence model, confirming that GFNN captures genuine predictive structure in the data rather than merely following the previous day's value. **Figure 4** compares the actual and predicted rainfall values on the 366 test samples. The predicted curve follows the actual values reasonably well in the low-to-moderate rainfall range (0-25 mm), which dominates the test data, but the model consistently underestimates rainfall above approximately 40 mm. This is expected, since the training data is dominated by low rainfall values, so the network's weights are optimized to minimize error for the most frequently occurring values, while rare extreme events are underrepresented in the mapping the model has learned.

4.4 One-Step-Ahead Forecasting Result

Using the meteorological data of 31 December 2025 as input, the trained GFNN model produced a rainfall forecast for 1 January 2026, as summarized in **Table 5**.

The 9.12 mm discrepancy between the predicted and actual values for this single date is comparable in magnitude to the model's overall testing RMSE of 9.8949 mm, and is consistent with the high day-to-day rainfall variability in Bandung reflected in its standard deviation (10.78–11.55 mm) and coefficient of variation (126.3%). This indicates that the discrepancy at this single prediction point is within the range of error already characterized by the model's aggregate testing performance, although a single-point forecast cannot by itself represent the model's overall performance.

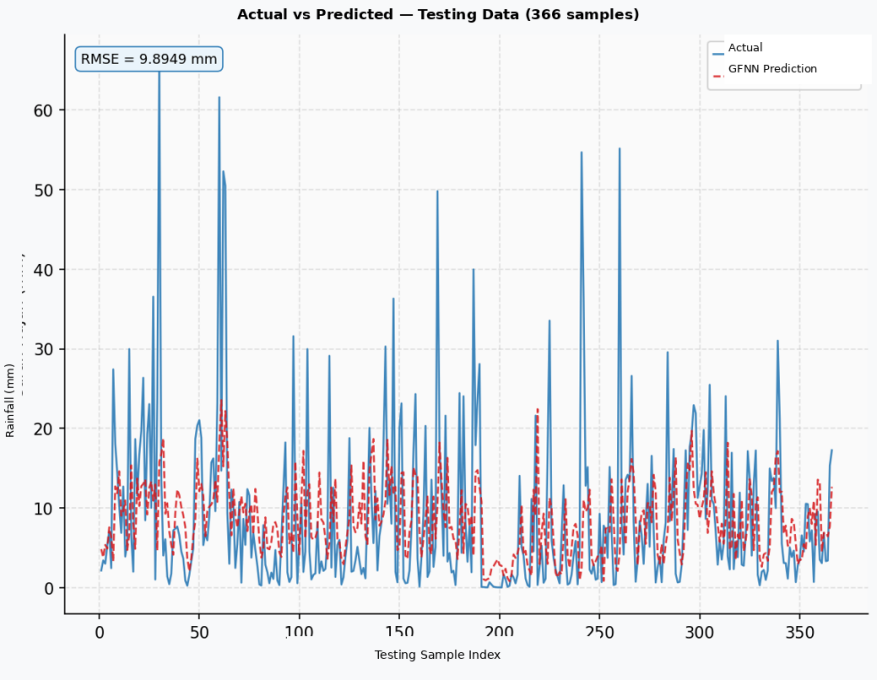


Figure 4. Actual vs. Predicted Values on the Testing Data

Table 5. One-Step-Ahead Rainfall Forecasting Result

Description	Value
Input Date	31 December 2025
Forecast Date	1 January 2026
Predicted Value	12.11 mm
Actual Value	2.99 mm
Difference	9.12 mm

5. CONCLUSION

This study implemented a Generalized Feedforward Neural Network to forecast one-step-ahead daily rainfall in Bandung City using ten years (2016–2025) of NASA POWER meteorological data. The implementation involved missing-value checking, percentile-based clipping (99.9th percentile) for extreme values, min-max normalization to [-1,1], and sliding-window construction of input–target pairs, followed by training a GFNN with a tanh activation function, a shortcut connection from input to output, a learning rate of 0.01, and backpropagation over 1,000 epochs. Testing six hidden-layer configurations (6–11 neurons) showed that none overfit the training data, and the 5-6-1 architecture achieved the lowest testing RMSE.

The selected GFNN 5-6-1 model achieved an RMSE of 8.9018 mm on the training data and 9.8949 mm on the test data, a difference indicating adequate generalization, and outperformed the persistence baseline (RMSE 12.3140 mm) by 19.6%, confirming that the model captures genuine predictive structure rather than a trivial carry-forward pattern. The model's performance needs to be distinguished between general rainfall conditions and extreme events: in the low-to-moderate rainfall range (0-25 mm) that dominates daily data in Bandung, the model performs well and consistently with the overall test RMSE; however, for rainfall above approximately 40 mm, the model systematically underestimates the actual values due to the dominance of low values in the training-data distribution. This difference in performance indicates that the model, in its current form, is reasonably well suited to general daily rainfall forecasting, while its application in contexts that require high sensitivity to extreme events, such as flood early-warning systems, requires further development before it can be fully relied upon for that

purpose. The one-step-ahead forecast for January 1, 2026 produced a value of 12.11 mm against an actual value of 2.99 mm, a difference of 9.12 mm that remains consistent with the model's overall test RMSE given the naturally high variability of daily rainfall in Bandung (coefficient of variation of 126.3%).

These findings show that GFNN is able to capture the nonlinear and stochastic characteristics of daily rainfall in Bandung and offers an improvement over the naïve baseline. Future research is recommended to explore techniques for handling imbalanced data, such as weighted loss functions, to reduce the underestimation of extreme rainfall, to test longer sliding-window sizes (e.g., 3, 5, or 7 days) to capture short-term temporal cycles, to validate NASA POWER data against ground-station measurements (BMKG) given the tendency of satellite reanalysis data to overestimate rainfall in tropical highland regions, and to compare GFNN with other time series deep learning architectures such as LSTM or GRU.

REFERENCES

[1] M. N. French, W. F. Krajewski, and R. R. Cuykendall, "Rainfall forecasting in space and time using a neural network," *Journal of Hydrology*, vol. 137, no. 1–4, pp. 1–31, 1992.

[2] Geografi.org, "Kondisi geografi kota bandung," https://www.geografi.org/2017/12/kondisi-geografi-kota-bandung.html, 2017.

[3] Universitas Gadjah Mada, "Iabi: Banjir bandung akibat konversi lahan das citarum," https://ugm.ac.id/id/berita/12719-iabi-banjir-bandung-akibat-konversi-lahan-das-citarum/, 2016.

[4] R. Kurniawan, "Kota bandung diterjang 12 bencana awal 2025," Metrotvnews, 2025, https://www.metrotvnews.com/read/NP6C3gE2-kota-bandung-diterjang-12-bencana-awal-2025.

[5] S. F. Rohmana, A. Rusgiyono, and S. Sugito, "Penentuan faktor-faktor yang mempengaruhi intensitas curah hujan dengan analisis diskriminan ganda dan regresi logistik multinomial," *Jurnal Gaussian*, vol. 8, no. 3, pp. 398–406, 2019.

[6] H. Ihsan, Irwan, and A. I. E. Nensi, "Implementation of backpropagation and hybrid arima-nn methods in predicting accuracy levels of rainfall in makassar city," *Barekeng*, vol. 18, no. 4, pp. 10–14, 2024.

[7] T. K. Febrianti, W. Sulandari, and H. Pratiwi, "Peramalan curah hujan di kota bandung menggunakan singular spectrum analysis," *Jurnal Ilmiah Matematika*, vol. 8, no. 2, pp. 56–65, 2021.

[8] E. Habinuddin, "Penerapan fuzzy time series untuk memprediksi curah hujan kota bandung," *Jurnal Digit*, vol. 12, no. 2, p. 115, 2022.

[9] N. A. Charaniya and S. V. Dudul, "Design of neural network models for daily rainfall prediction," *International Journal of Computer Applications*, vol. 61, no. 14, pp. 23–27, 2013.

[10] M. Shoaib, "Impact of wavelet transformation on data driven rainfall-runoff models," ResearchSpace Repository, 2016.

[11] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi, "An artificial neural network model for rainfall forecasting in bangkok, thailand," *Hydrology and Earth System Sciences*, vol. 13, no. 8, pp. 1413–1425, 2009.

[12] N. A. Charaniya and S. V. Dudul, "Focused time delay neural network model for rainfall prediction using indian ocean dipole index," in *Proc. 4th Int. Conf. Computational Intelligence and Communication Networks (CICN)*, 2012, pp. 851–855.

[13] P. A. Kharat and S. V. Dudul, "Epilepsy diagnosis based on generalized feed forward neural network," *Interdisciplinary Sciences – Computational Life Sciences*, vol. 4, no. 3, pp. 209–214, 2012.

[14] G. Arulampalam and A. Bouzerdoum, "A generalized feedforward neural network architecture for classification and regression," *Neural Networks*, vol. 16, no. 5–6, pp. 561–568, 2003.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[16] A. N. Celik and M. Kolhe, "Generalized feed-forward based method for wind energy prediction," *Applied Energy*, vol. 101, pp. 582–588, 2013.

[17] M. Shoaib *et al.*, "A comparative study of various hybrid wavelet feedforward neural network models for runoff forecasting," *Water Resources Management*, vol. 32, no. 1, pp. 83–103, 2018.

[18] R. Y. Tallar, *Dasar-Dasar Hidrologi Terapan*. Bandung: Ideas Publishing, 2023.

[19] L. V. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1994.

[20] J. J. Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya dengan Matlab*. Yogyakarta: Penerbit Andi, 2005.

[21] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2009.

[22] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA: Morgan Kaufmann, 2012.

[23] S. Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[24] D. Fletcher and E. Goss, "Forecasting with neural networks," *Information & Management*, vol. 24, no. 3, pp. 159–167, 1993.

[25] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. Melbourne, Australia: OTexts, 2018.

[26] Sudjana, *Metoda Statistika*, 6th ed. Bandung: Tarsito, 1996.