STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–21.
Published online in International Academic Press (www.IAPress.org)

Hydrological Ensemble Modeling of Juwata Station Maximum Rainfall using FFNN and RNN

Mislan [1,*], Andrea Tri Rian Dani [2,3]

[1]*Department of Physics, Faculty of Mathematics and Natural Sciences, Mulawarman University, Samarinda, Indonesia*
[2]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Mulawarman University, Samarinda, Indonesia*
[3]*Doctoral Study Program MIPA, Faculty of Science and Technology, Airlangga University, Surabaya, Indonesia*

Abstract This study aims to develop a more accurate and reliable hydrological forecasting model for predicting maximum rainfall at Juwata Station in Tarakan, North Kalimantan. The objective is to address the limitations of traditional statistical approaches when handling highly dynamic and non-linear hydrometeorological patterns. To achieve this, we apply an ensemble modeling approach that integrates four neural network architectures: FFNN Backpropagation, FFNN Rprop, Elman RNN, and Jordan RNN. The methods involve training each model independently, evaluating their predictive performance, and combining their outputs using a weighted averaging scheme that assigns greater influence to models with lower error metrics. The analysis focuses on the ability of each architecture to capture non-linear relationships and temporal dependencies within the rainfall data. The findings show that the ensemble model produces consistently lower SMAPE, MSE, and RMSE values compared to any single model or two-model combination, demonstrating superior predictive capability. The novelty of this study lies in its integration of multiple feedforward and recurrent neural network structures into a unified ensemble specifically tailored for tropical rainfall forecasting, where atmospheric variability is high. This ensemble framework offers a more robust foundation for predicting extreme rainfall events, supporting flood risk mitigation and regional infrastructure planning.

Keywords Climate; Tropical Rainforest; Ensemble Forecasting; Neural Network; Rainfall; Time Series.

DOI: 10.19139/soic-2310-5070-3845

1. Introduction

Kalimantan Island is nicknamed the world's heart because of its biodiversity in the global tropical rainforest belt. Dense forests, warm sea temperatures, and equatorial alignment give this region very high annual rainfall [1], [2]. Extreme rainfall on this island can reach hundreds of millimeters per month, which is crucial in hydrological balance and causes potential flash floods and erosion if not adequately mitigated [3], [4]. The intensity and frequency of extreme rainfall in Kalimantan are strongly influenced by global and regional climate oscillations ranging from the Asia-Australia monsoon and the Madden-Julian Oscillation to El Niño and La Niña anomalies[5]. When El Niño is strong, the convection pattern shifts to the eastern Pacific so that rainfall decreases; conversely, La Niña and the shifting monsoon often trigger sudden heavy rain [6]. These non-linear and seasonal patterns challenge classical statistical models and open up strong opportunities for machine-learning approaches that can capture complex relationships [7].

Juwata Tarakan Climatology Station is located in the northern coastal area of Kalimantan, has low and flat topography characteristics, and is directly facing the Sulawesi Sea. This location makes it very vulnerable to the influence of atmospheric dynamics such as sea-land wind convergence, moisture intrusion from the sea, and its

*Correspondence to: Mislan (mislanfmipa@gmail.com). Department of Physics, Faculty of Mathematics and Natural Sciences, Mulawarman University, Samarinda, Indonesia

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 202x International Academic Press

proximity to the center of tropical storm formation in the western Pacific Ocean. The combination of these factors produces a distinctive extreme rainfall pattern that does not always follow the trend from inland areas. These fluctuations are significant because the maximum rainfall can trigger disasters such as inundation, flash floods, and disruption to sanitation systems and clean water distribution [8], [9]. Therefore, accurate predictions of maximum rainfall at this location are urgently needed for infrastructure planning, hydrometeorological risk mitigation, and water resource resilience in the rapidly developing North Kalimantan.

The time series analysis approach is the primary method to study this phenomenon [11], [12]. Time series analysis is a branch of statistics that studies the behavior and structure of data patterns based on their chronological order [13], [14]. In hydrological data, especially rainfall, time series often show seasonal patterns, long-term trends, and non-linear fluctuations [15], [16]. In recent decades, conventional approaches such as ARIMA, SARIMA, exponential smoothing, and moving averages have been widely used but have limitations in handling data that fluctuates sharply due to dynamic atmospheric [16], [1] and ocean factors [17]. With the big data and artificial intelligence (AI) era, neural network-based methods have become a superior alternative because they can build complex non-linear models without assuming rigid data distribution [18], [19], [20]. Various neural network architectures such as Backpropagation , Resilient Propagation (Rprop), and the Elman and Jordan neural network model allow learning from past patterns [17], [21], [22]. They can adapt to the dynamics of climate change, making them very relevant for predicting maximum rainfall in tropical areas such as Tarakan.

This study's four neural network architectures have unique working principles and characteristics. Backpropagation is a basic supervised learning algorithm that uses the backpropagation method to minimize errors by gradually adjusting weights with gradient derivatives [23]. However, this process may exhibit slow convergence and can be sensitive to local minima, depending on the optimization settings and network structure. Resilient Propagation (Rprop) was developed to overcome this by adapting weight updates based on the sign of the gradient, potentially improving convergence efficiency [24]. Meanwhile, the Elman Network is a Recurrent Neural Network (RNN) type with a context layer, allowing the network to remember information from the previous time so that it is suitable for time series data with short-term dependencies [25], [26]. Like Elman, Jordan Network is also an RNN, but its context layer receives input from the output layer, providing different capabilities in handling feedback from predictions to the network [27], [28].

Individually, these four architectures still have limitations in predictive performance. Backpropagation, for example, is prone to slow training processes and overfitting when the number of hidden neurons is too large. Conversely, the risk of underfitting becomes dominant if the configuration is too simple. Resilient Propagation (Rprop), although superior in convergence speed, still depends on the initial conditions of the weights and network structure. Meanwhile, Elman and Jordan, as RNN architectures, have advantages in capturing time dynamics but are also very sensitive to noise and the length of the input sequence. This imbalance can result in fluctuating or unstable prediction outputs, especially if the model is not trained with sufficiently representative validation[29], [30], [31], [32], [33].

One approach that can be used to improve the accuracy and stability of prediction results is to apply the ensemble forecasting method [34], [35]. Ensemble forecasting is a technique that combines predictions from several different models to produce a more reliable final estimate [35]. The basic principle is that the strengths of other models can cover the weaknesses of one model, so they collectively produce more accurate predictions and are robust to data variability[35], [36]. The ensemble strategy used in this study is the weighted average method, where the weight of each model's contribution is determined based on the accuracy achieved, in this case, using the Symmetric Mean Absolute Percentage Error (SMAPE) value [35], [37]. The smaller the SMAPE value of a model, the greater the weight given in the combination. This study uses the ensemble approach to combine the prediction results of two, three, or four neural network architectures: Backpropagation, Rprop, Elman, and Jordan. Rather than introducing a new ensemble algorithm, this study systematically evaluates different combinations of these architectures to investigate whether their distinct learning characteristics provide complementary predictive information. The combination is carried out using a weighted average approach based on the accuracy performance of each model. This evaluation aims to identify the ensemble configuration that achieves the most accurate maximum rainfall predictions at Juwata Station, Tarakan, based on quantitative comparisons of in-sample and out-of-sample performance. In relation to the global development agenda, this study directly supports Sustainable

Development Goal (SDG) number 13, ”Climate Action”. By evaluating alternative neural network configurations for maximum rainfall prediction at Juwata Station, Tarakan, this study contributes to the development of local rainfall forecasting methods. The resulting forecasts may provide preliminary information for future studies on climate adaptation and flood risk management, although their direct application in decision-making is beyond the scope of this study.

Although weighted averaging has been widely established in ensemble forecasting, the predictive benefits of combining different neural network architectures depend on their individual performance and the extent to which their prediction errors complement one another. In particular, combining feedforward neural networks, represented by Backpropagation and Rprop, with recurrent neural networks, represented by Elman and Jordan, raises an important empirical question regarding whether their different learning mechanisms can improve maximum rainfall prediction. Furthermore, the effectiveness of different ensemble configurations requires systematic evaluation to determine whether increasing the number of constituent models consistently improves predictive accuracy. These considerations motivate a comparative investigation of two-, three-, and four-model ensembles using performance-based weighting and out-of-sample evaluation.

Accordingly, this study investigates the predictive performance of different ensemble configurations involving Backpropagation, Rprop, Elman, and Jordan neural networks for maximum rainfall forecasting at Juwata Station, Tarakan. The main contribution of this study lies in the systematic comparison of ensemble configurations containing two, three, and four neural network models, with their contributions determined by SMAPE-based weights. Rather than claiming methodological novelty in the weighted averaging algorithm itself, this study provides an empirical assessment of how combining feedforward and recurrent neural network architectures affects predictive accuracy. Through comparisons between individual models and ensemble configurations, this research seeks to identify an effective modeling strategy for maximum rainfall prediction in a tropical region. The findings are expected to provide practical insights into neural network ensemble selection and contribute to the development of reliable rainfall forecasting applications. The structure of the article includes Materials and Methods, Results and Discussion, and Conclusion.

2. Materials and Methods

2.1. Rainfall at Juwata Station, Tarakan

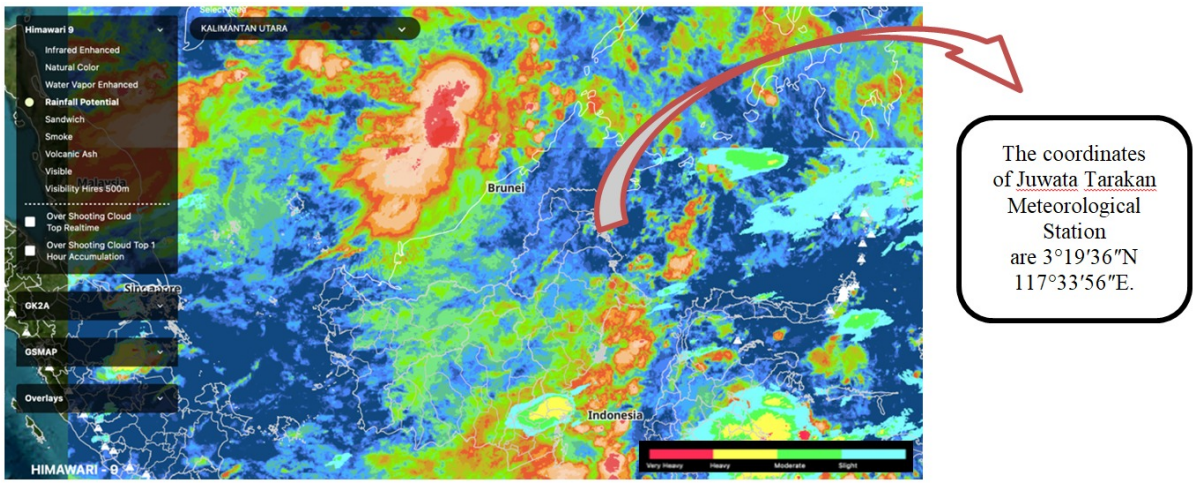
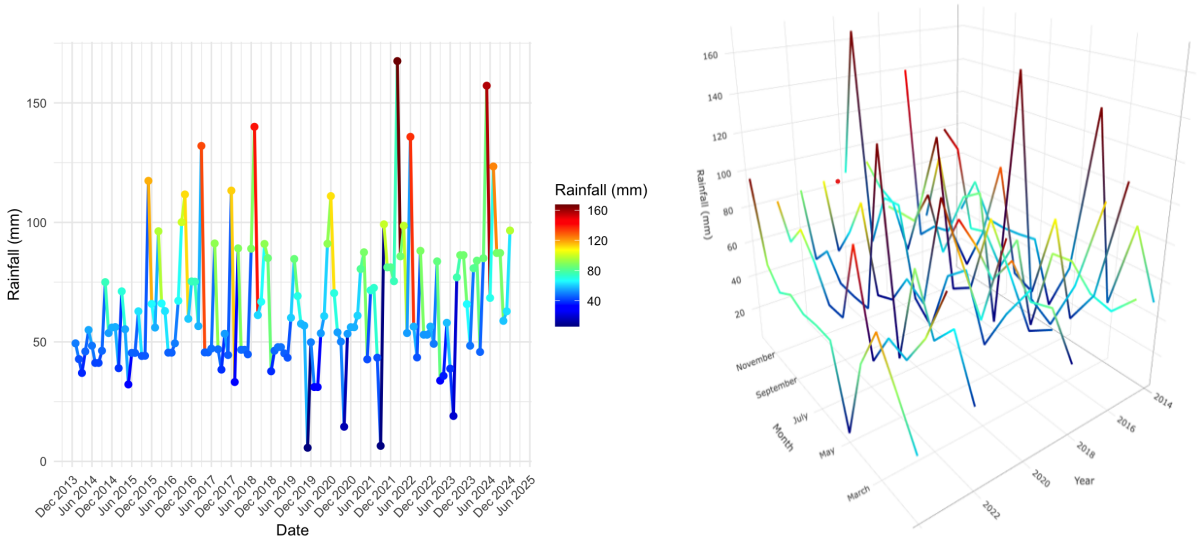


Figure 1. Himawari-9 satellites representation of data collection location (2024-12-01)

Figure 1 shows the spatial context of the study area using the Himawari-9 satellite image (captured on 2024-12-01), illustrating the geographical setting of the data collection point at Juwata Station, Tarakan, North Kalimantan.

The data used in this study were compiled by the Meteorology, Climatology, and Geophysics Agency (BMKG) from 2014 to 2024 on a monthly basis. The data collection point is located at the Juwata Station in Tarakan, North Kalimantan. The research variable is the monthly maximum rainfall during the observation period, denoted as y_t, and visualized in Figure 2.

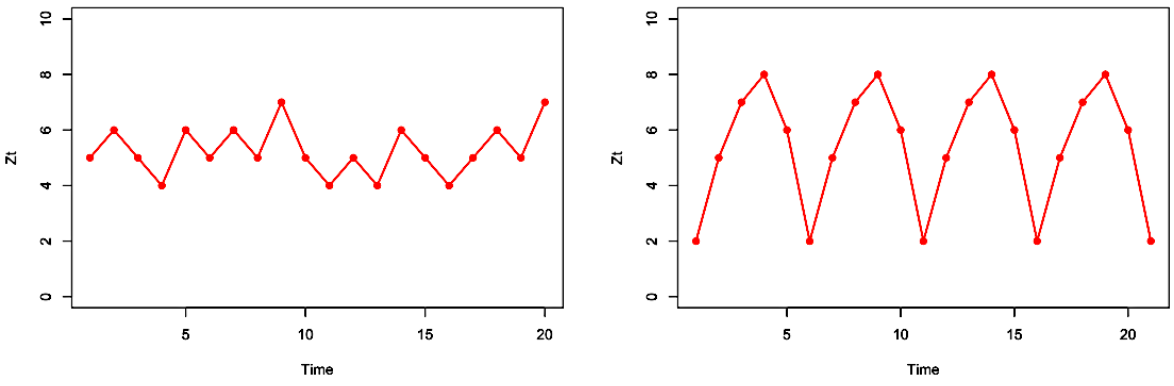


(a) (b)

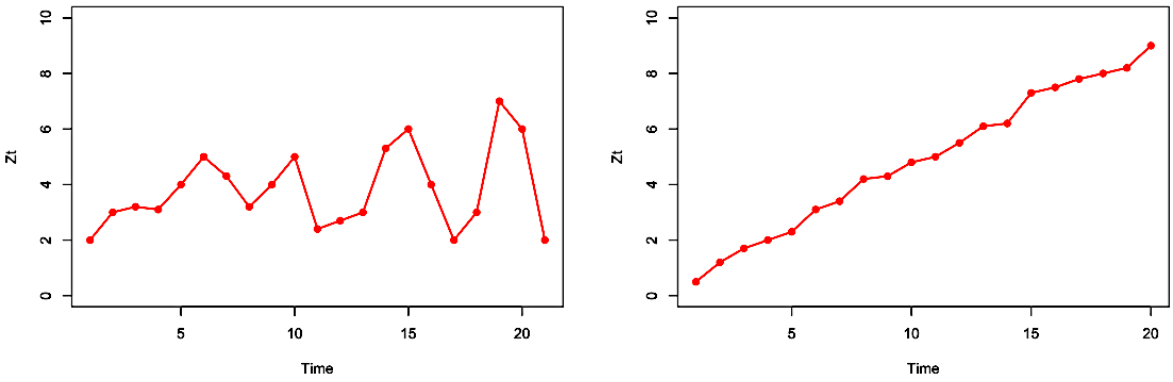
Figure 2. Rainfall data visualization (a) 2D time series plot; (b) 3D time series

2.2. Time Series Data

Time series analysis is a statistical method used to predict future events based on historical data [12], [38]. This analysis process is carried out by processing data from the current period (y_t) which is influenced by one or more observations from the previous period ($y_(t-k)$).



(a) (b)



(c) (d)

Figure 3. Time series data patterns (a) horizontal; (b) seasonal; (c) cyclical; (d) trend

Figure 3 illustrates four common patterns in time series data often encountered in hydrological analysis. The horizontal pattern (a) shows data fluctuating around a fixed mean without any upward or downward trend, indicating a long-term stationary condition. The seasonal pattern (b) is characterized by periodic fluctuations in fixed time intervals, such as monthly rainfall that follows the wet and dry seasons cycle. Meanwhile, the cyclical pattern (c) is like seasonality but occurs over a more extended period and is irregular, often influenced by external macro factors such as global climate. Finally, the trend pattern (d) shows a tendency for data values to increase or decrease consistently over a specific period, reflecting long-term systematic changes such as global warming. Understanding these four patterns is essential in building accurate and relevant forecasting models with data characteristics [39], [40].

2.3. Neural Network

Time series modeling aims to capture historical patterns from sequential data to predict future values. General characteristics of time series data, especially in hydrology, include non-stationary, non-linear properties and contain seasonal, cyclical, trend, and random noise components. This complexity challenges conventional statistical models that rely on the assumption of linearity and normal distribution. Therefore, more flexible approaches such as artificial neural networks (Neural Networks/NN) become relevant choices to handle dynamic time series problems [9], [11].

Neural Network is a machine learning system inspired by how the human brain works. It consists of interconnected processing units (neurons) through layers: an input layer, one or more hidden layers, and an output layer [19], [41]. In time series modeling, the data is usually first transformed into a supervised learning format, where one or more past values (lags), such as $y_(t-1)$, $y_(t-2)$,... are used as input to predict the current value y_t. The main advantage of NN is its ability to learn non-linear relationships between variables without requiring data distribution assumptions, as in ARIMA or other parametric models [16].

2.4. Feed Forward Neural Network

Feedforward Neural Network (FFNN) is one of the most widely used artificial neural network architectures in nonlinear modeling, including time series forecasting [19]. This network consists of three main layers: the input layer, one or more hidden layers, and the output layer [40], [43]. There is no feedback connection between layers, so information only flows in one direction from the input to the output. An illustration of the FFNN architecture is shown in Figure 4.

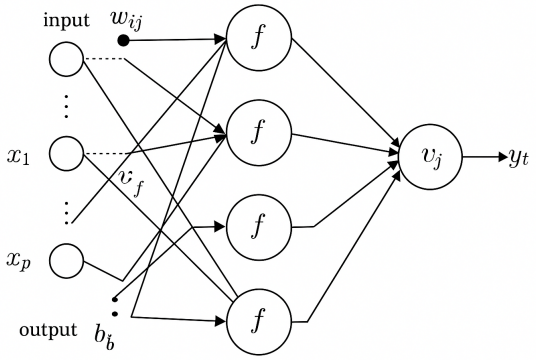


Figure 4. FFNN architecture

In univariate time series, FFNN models the relationship between current and previous values (lags). The FFNN (p, q) model uses p input units representing observation lags, q hidden units, and one output neuron. The mathematical Equation of FFNN is written in Equation (1).

$$\hat{y}_t = \sum_{j=1}^{q} v_j \cdot f\left(\sum_{i=1}^{p} w_{ij} \cdot x_i + b_j\right) + b_0 \tag{1}$$

where:
$\hat{y}_t$: estimated value at time t
x_i : input value at lag i
w_{ij} : weight from input neuron i to hidden neuron j
v_j : weight from hidden neuron j to output neuron
b_j, b_0 : bias at hidden and output neurons
$f(\cdot)$: non-linear activation function, such as sigmoid or tanh.

2.5. Recurrent Neural Network

Recurrent Neural Network (RNN) is one of the artificial neural network architectures specifically designed to process sequential data such as time series, text, or signals. Unlike FFNN, which processes input independently, RNN has a recurrent or looping mechanism that allows the network to store and remember information from previous inputs to maintain context or short-term memory in the training and prediction process [22], [28].

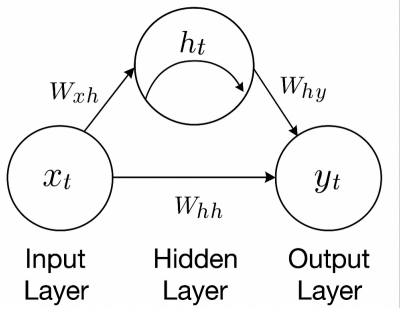


Figure 5. RNN architecture

Figure 5 illustrates the general architecture of a RNN. In its basic structure, RNN consists of an input, hidden, and output layer. In its basic structure, RNN consists of an input, hidden, and output layer. What distinguishes RNN from FFNN is the presence of feedback connections in the hidden layer so that the output of a neuron at time

will be used as additional input at time . This process creates a hidden state that evolves, allowing RNNs to capture temporal dependencies in the data [22], [35]. Mathematically, the simple representation of the RNN is written in Equations (2) and (3).

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} - b_h) \tag{2}$$

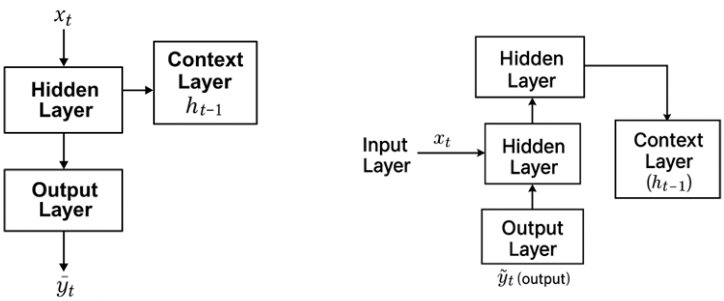
thus,

$$\hat{y}_t = \sigma(W_{hy}h_t + b_y) \tag{3}$$

where:

x_t : input at time t
h_t : hidden state at time t
$\hat{y}_t$: estimated value at time t
W_{xh}, W_{hh}, W_{hy} : weight matrices
b_h, b_y : bias
σ : non-linear activation function such as tanh or ReLU

Two commonly used RNN architectures are Elman and Jordan. Elman's architecture was introduced by Jeffrey Elman in 1990. In this architecture, the context unit that carries memory comes from the hidden state at the previous time h_{t-1}. The secret state is fed back to the hidden layer as part of the input the next time so the network can capture the dependency patterns in sequential data [27], [44]. Figure 6a illustrates the Elman RNN architecture, where the context units receive feedback from the hidden layer at the previous time step.



(a) (b)

Figure 6. RNN architecture (a) Elman; (b) Jordan

Meanwhile, Michael Jordan introduced Jordan architecture in 1986, using context units derived from the network output at the previous time $\hat{y}_{t-1}$. Figure 6b illustrates the Jordan RNN architecture. This output is then fed back to the hidden layer the next time. The main difference between Elman architecture and Jordan architecture is the location of the network memory source; memory comes from the network output itself [27]. Mathematically, a simple representation of Jordan architecture is written in Equations (4) and (5).

$$h_t = \sigma(W_{xh}x_t + W_{hy}\hat{y}_{t-1} + b_h) \tag{4}$$

thus,

$$\hat{y}_t = \sigma(W_{hy}h_t + b_y) \tag{5}$$

2.6. Algorithm Backpropagation and Resilient Propagation

Backpropagation is a gradient-based learning algorithm used to train FFNN. Training is done by minimizing the error function, namely the Mean Squared Error (MSE), using the gradient descent method [33], [45]. The training process involves two main stages: a forward pass to calculate the network output and a backward pass to calculate the gradient and update the weights. The weights are updated with the formula in Equation (6).

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial E}{\partial w_{ij}} \tag{6}$$

where:
η : learning rate
$\frac{\partial E}{\partial w_{ij}}$: derivative or the error function with respect to the weight w_{ij}
E : total error function

To overcome the weaknesses of backpropagation, the Resilient Propagation (Rprop) algorithm is introduced as a gradient-based approach that does not depend on the magnitude of the gradient. Rprop only considers the direction of the derivative and adjusts the magnitude of the weight change using an adaptive value ($\Delta_i j$). The weight update is performed based on the rules in Equation (7).

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \text{sign}\left(\frac{\partial E^{(t)}}{\partial w_{ij}}\right) \Delta_{ij}^{(t)} \tag{7}$$

The value of $\Delta_i j$ will be increased if the derivative sign is the same as the previous iteration but will be decreased if there is a change in direction. Rprop can accelerate convergence and provide more stable training performance [33].

2.7. Ensemble Forecasting

Ensemble forecasting is an approach in machine learning combining several predictive models to obtain more accurate, stable, and robust results than using a single model [34], [46]. This approach was introduced by Hansen and Salamon (1990), who stated that combining several weak models (weak learners) can produce better prediction performance if the models are sufficiently diverse and independent. This technique has been proven effective in various domains, including classification, regression, and time series forecasting[37] . In the context of hydrological time series forecasting, ensembles can overcome the weaknesses of each model, such as overfitting, underfitting, or sensitivity to noise. One of the most used ensemble techniques is weighted averaging, where the weight of each model is determined based on performance criteria, such as the Symmetric Mean Absolute Percentage Error (SMAPE). Models with better performance are given higher weights to make the result more balanced [36], [37].

In this study, ensemble forecasting is carried out by combining four artificial neural network architectures, namely:

(a) Feedforward Neural Network (FFNN) with Backpropagation algorithm,

(b) Feedforward Neural Network (FFNN) with Resilient Propagation (Rprop) algorithm,

(c) Elman type Recurrent Neural Network (RNN), and

(d) Jordan-type Recurrent Neural Network (RNN).

Ensembling will be carried out in three schemes:

(a) Combination of two models (2-model ensemble): all combinations of the four models above.

(b) Combination of three models (3-model ensemble): all combinations of triplets.

(c) Overall Combination (4-model ensemble): combining all four models simultaneously.

Each ensemble will use a weighted average approach based on SMAPE in-sample performance to determine the weight of each model's contribution to the final prediction. This strategy is expected to improve the accuracy of maximum rainfall predictions and reduce dependence on a single model at risk of producing extreme results.

Mathematically, the prediction of the ensemble ($\hat{y}_{\text{ensemble}}$) from n base models can be written in Equation (8).

$$\hat{y}_{ensemble}(t) = \sum_{i=1}^{n} w_i \, \hat{y}_i(t) \tag{8}$$

where:
$\hat{y}_i(t)$: prediction result from the i-*th* model at time t
w_i : weight of the i-*th* model at time t
n : number of models being combined

The weight w_i in this study is obtained based on the in-sample performance of each model, particularly the inverse of the SMAPE value.

$$w_i = \frac{1/SMAPE_i}{\sum_{j=1}^{n} 1/SMAPE_j} \tag{9}$$

Thus, the model with higher accuracy (smaller SMAPE) will contribute more to the final prediction. This strategy ensures a balance between individual performance and the diversity of models in the ensemble system. Next, we conducted a Systematic Literature Review (SLR) by reviewing several studies that examined Ensemble Forecasting methods in Table 1.

Table 1. SLR metode Ensemble forecasting

No	Author & Years	Ensemble Method	Key Findings
1	Melin et al. (2020)	MNNF – Modular Neural Network with Fuzzy aggregation (combination of Nonlinear Autoregressive (NAR), Feedforward Neural Network (FITNET), and fuzzy logic)[47]	• MNNF is an ensemble model combining NAR and FITNET using fuzzy integration. • Improves COVID-19 prediction accuracy compared to single models.
2	Phyo & Byun (2021)	Hybrid Ensemble Deep Learning (combination of MLP, CNN, LSTM, CNN-LSTM) [48]	• This hybrid ensemble model is more accurate than individual deep learning models (MLP, CNN, LSTM). • Adaptive to seasonal patterns and high non-linearity.
3	Kim et al. (2023)	Stacked Ensemble (combination of Recurrent Neural Network (RNN), LSTM, Gated Recurrent Unit (GRU)) [46]	• Combines three time series methods: RNN, LSTM, and GRU. • Stacking ensemble proves to improve accuracy.
4	He et al. (2023)	ARMA-CNNLSTM (combination of Autoregressive Moving Average (ARMA), CNN, LSTM) [49]	• ARMA-CNNLSTM outperforms all evaluation metrics. • Effectively combines the strengths of linear models (ARMA) and non-linear models (CNN, LSTM).
5	Xiao et al. (2023)	Hybrid Ensemble NP–CNN–LSTM (NeuralProphet + CNN + LSTM)[50]	• NP for seasonal trends, CNN for local features, LSTM for long-term dependencies. • This two-stage model excels in daily photovoltaic power prediction.

No	Author & Years	Ensemble Method	Key Findings
			• Outperforms single and other hybrid models in RMSE, MAE, and R^2.
6	Ciechulski & Osowski (2024)	Ensemble MLP–Radial Basis Function (RBF)–LSTM with integration of MLP and Principal Component Analysis (PCA) [10]	• Combination of MLP, RBF, and LSTM shows improved accuracy over single models. • Integration of MLP combiner and PCA effectively reduces MAE and RMSE.
7	Huang et al. (2024)	Ensemble Feature Selection (combination of Pearson Correlation, Recursive Feature Elimination (RFE), Random Forest (RF), Gradient Boosted Decision Tree (GBDT), and eXtreme Gradient Boosting (XGBoost)) [37]	• Ensemble feature selection improves accuracy and robustness. • Outperforms single feature selection in terms of MAE and MSE.
8	Jumadi et al. (2025)	Ensemble stacking combining RF, XGB, SVR, MLP, LGBM, LSTM, GRU, TCN, CNN, and Transformer, with a linear regression meta-learner [51]	• The stacking ensemble model outperforms all individual machine learning and deep learning models (RF, XGB, SVR, MLP, LGBM, LSTM, GRU, TCN, CNN, Transformer). • Provides reliable rainfall projections for 2025–2030 using robust ensemble learning.
9	Song et al. (2025)	Flexible Hybrid Ensemble (FHE) with dynamic model selection using a meta-model [52]	• Dynamically selects the best base model for each prediction instance. • Improves MAPE by 30% compared to SVR. • Maintains high accuracy across diverse weather conditions. • Robust and scalable for distributed solar power systems.
10	Ahn & Lee (2025)	Stacked ensemble model combining LightGBM + XGBoost, compared against single models (LightGBM, XGBoost, LSTM) [53]	• Stacked ensemble achieves the lowest RMSE and MAE across all datasets. • Outperforms all individual models (LightGBM, XGBoost, LSTM).

The studies summarized in Table 1 demonstrate the growing application of ensemble forecasting across different domains, including infectious disease prediction, renewable energy forecasting, and rainfall prediction. Previous studies have explored various ensemble strategies, such as fuzzy aggregation, stacking, hybrid architectures, and dynamic model selection, with many reporting improvements over individual models. These findings highlight the potential of ensemble learning to combine the predictive capabilities of different models. However, the reviewed studies primarily emphasize the predictive advantages of particular ensemble methods or hybrid architectures. Less

attention is given to systematically examining whether increasing the number of constituent models consistently improves out-of-sample forecasting performance, particularly when different combinations of neural network architectures are evaluated under the same experimental framework. Moreover, the effectiveness of an ensemble may depend not only on the number of models included but also on the characteristics and predictive performance of the constituent models.

To address this gap, the present study comparatively evaluates two-, three-, and four-model ensemble configurations involving FFNN Backpropagation, FFNN Rprop, Elman RNN, and Jordan RNN for maximum rainfall prediction at Juwata Station, Tarakan. The models are combined using inverse-SMAPE-based weighted averaging, and their predictive performance is assessed using SMAPE, RMSE, and MSE on both training and testing datasets. Rather than proposing a fundamentally new ensemble algorithm, this study provides empirical evidence on whether increasing ensemble size necessarily leads to improved out-of-sample predictive accuracy.

2.8. Accuracy Measurement

To evaluate the performance of the predictive models, several accuracy metrics are used, including Symmetric Mean Absolute Percentage Error (SMAPE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These three metrics are commonly used in time series forecasting studies to assess how well the predicted values match the actual data [40], [54].

SMAPE measures the absolute error of the model in a symmetric manner, providing a more balanced percentage error, especially when dealing with values close to zero. The formula for SMAPE is written in Equation (10).

$$SMAPE = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|y_t - \hat{y}_t|}{\left(\frac{|\hat{y}_t| + |y_t|}{2}\right)} \tag{10}$$

Where y_t is the actual value at time t and $(\hat{y}_t)$ is the predicted value at time t.

MSE measures the average of the squared differences between the actual and predicted values. MSE gives a larger penalty to extreme prediction errors (outliers), so it is more sensitive to significant errors. The MSE formula is written in Equation (11).

$$MSE = \frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2 \tag{11}$$

RMSE is the root of MSE and returns the error units to the same units as the original data. The RMSE formula is written in Equation (12).

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2} \tag{12}$$

2.9. Research Stages

The research stages of Hydrological Ensemble Forecasting of Juwata Station Maximum Rainfall using Feed Forward Neural Network Backpropagation & Rprop and Recurrent Neural Network Elman & Jordan are visualized in Figure 7.

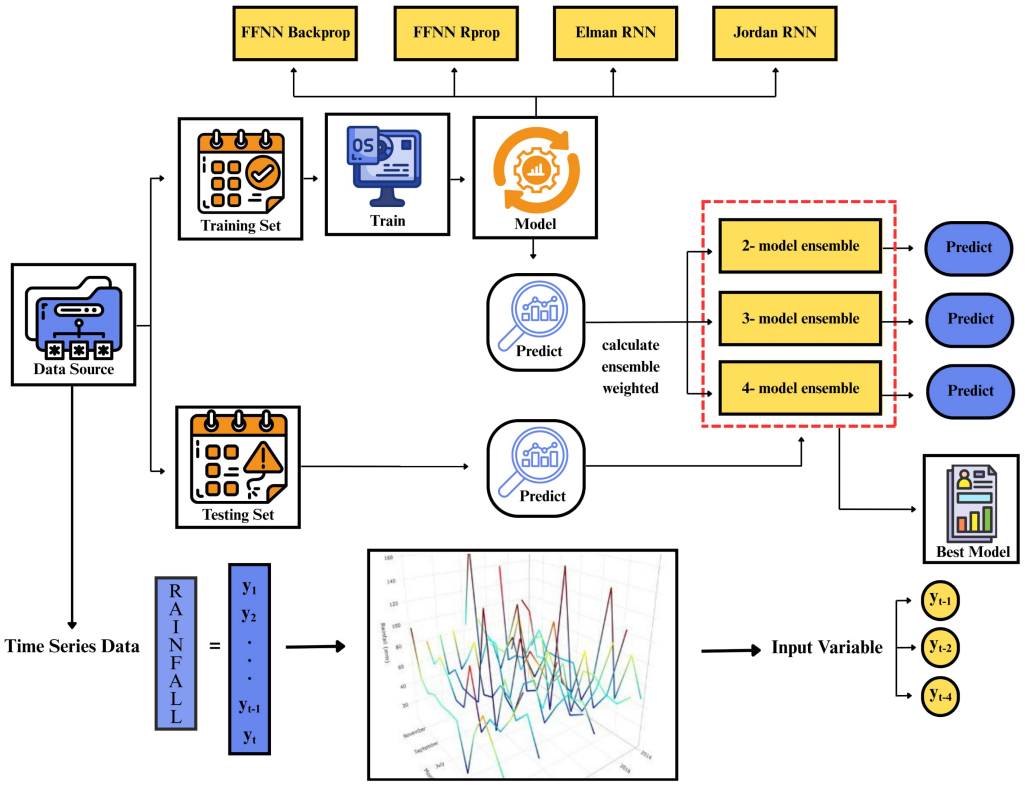


Figure 7. Workflow ensemble forecasting

As shown in Figure 7, the pseudo code for the different methods: FFNN Backpropagation, Rprop, Elman RNN, and Jordan RNN is presented below.

Pseudo code for FFNN:

```
Input: Hydrological time series dataset
1:  Split the dataset chronologically into training (95%) and testing (5%) sets
2:  Initialize best_SMAPE ← ∞
3:  Initialize best_model ← NULL
4:  Initialize optimal_architecture ← NULL
5:  For each configuration (L1, L2, learn_rate, algorithm) do
6:      Train the FFNN model using the training data
7:      Generate predictions for the testing data
8:      Calculate SMAPE, RMSE, and MSE
9:      If SMAPE < best_SMAPE then
10:         best_SMAPE ← SMAPE
11:         best_model ← trained_model
12:         optimal_architecture ← (L1, L2, learn_rate, algorithm)
13:     End if
14: End for
15: Return best_model, optimal_architecture
```

Pseudo code for RNN:

```
Input: Hydrological time series dataset
1:  Split the dataset chronologically into training (95%) and testing (5%) sets
2:  Initialize best_SMAPE ← ∞
3:  Initialize best_model ← NULL
```

```
Initialize optimal_architecture ← NULL
For each configuration (L1, L2, learn_rate, algorithm) do
    Train the RNN model using the training data
    Generate predictions for the testing data
    Calculate SMAPE, RMSE, and MSE
    If SMAPE < best_SMAPE then
        best_SMAPE ← SMAPE
        best_model ← trained_model
        optimal_architecture ← (L1, L2, learn_rate, algorithm)
    End if
End for
Return best_model, optimal_architecture
```

Pseudo code for Ensemble Forecasting:

```
# Load best models from grid search
Load best models: Rprop, Backprop, Elman, Jordan
# Predict using each model (in-sample)
preds_rprop = predict(Rprop, input)
preds_backprop = predict(Backprop, input)
preds_elman = predict(Elman, input)
preds_jordan = predict(Jordan, input)
# Unscale predictions to original scale
unscale_predictions()
# Calculate SMAPE for each model (in-sample)
smape_rprop = calculate_SMAPE(actual, preds_rprop)
smape_backprop = calculate_SMAPE(actual, preds_backprop)
smape_elman = calculate_SMAPE(actual, preds_elman)
smape_jordan = calculate_SMAPE(actual, preds_jordan)
# Calculate weights (inverse of SMAPE)
weight_rprop = 1 / smape_rprop
weight_backprop = 1 / smape_backprop
weight_elman = 1 / smape_elman
weight_jordan = 1 / smape_jordan
# Normalize weights
normalize_weights()
# Create ensemble predictions (weighted average)
ensemble_preds = weighted_average(preds_rprop, preds_backprop, preds_elman, preds_jordan)
# Evaluate ensemble (SMAPE, RMSE, MSE)
smape_ensemble = calculate_SMAPE(actual, ensemble_preds)
rmse_ensemble = calculate_RMSE(actual, ensemble_preds)
mse_ensemble = calculate_MSE(actual, ensemble_preds)
# Visualize the results
plot_results(actual, ensemble_preds)
# Return ensemble predictions and metrics
return ensemble_preds, smape_ensemble, rmse_ensemble, mse_ensemble
```

The pseudo code presented in the preceding sections serves as the conceptual foundation for model construction and analysis within the R programming environment. To facilitate the implementation of these methodologies, several key libraries are utilized. The **ggplot2** package is employed for data visualization, enabling the generation of informative plots that are essential for the evaluation and presentation of model predictions. The **RSNNS** library is integral to the development and training of neural networks, including Feed-Forward Neural Networks (FFNN), and facilitates the application of optimization techniques such as Backpropagation and Rprop. The **Metrics** package provides a range of functions for the calculation of performance metrics, including SMAPE, RMSE, and MSE, all of which are critical for assessing model accuracy and predictive power. Additionally, the **forecast** library is used for time series forecasting, enabling model prediction and validation processes. Collectively, these libraries provide a robust framework for the implementation, evaluation, and optimization of models such as FFNN Backpropagation, Rprop, Elman RNN, Jordan RNN, and ensemble forecasting techniques, specifically tailored for hydrological time series analysis.

In addition to these libraries, a data preprocessing stage was implemented to improve training consistency and model performance. Specifically, all input variables were normalized using the Min–Max scaling method, transforming the data into a 0–1 range. This normalization is essential for neural network models, as it prevents

variables with larger magnitudes from dominating the learning process and helps stabilize gradient updates. Min–Max scaling also accelerates convergence during training and reduces numerical instability in architectures such as FFNN Backpropagation, Rprop, Elman RNN, and Jordan RNN. By standardizing the feature scales prior to model fitting, this preprocessing step contributes to more stable learning dynamics and enhances the overall robustness of the proposed ensemble modeling framework.

3. Result and Discussion

This section presents the results of hydrological ensemble modeling for predicting the maximum rainfall at Juwata Station using four neural network models: FFNN with Backpropagation and Rprop and RNNs (Elman and Jordan). Each model's performance and ensemble are evaluated based on key metrics like SMAPE, MSE, and RMSE to assess prediction accuracy. The following discussion will compare individual models and ensemble configurations, highlighting their strengths and weaknesses in forecasting maximum rainfall.

3.1. Exploration Data

Based on Figure 2, it is known that the maximum rainfall data from Juwata Tarakan Station during the period from January 2014 to December 2024 shows a very consistent seasonal pattern. Rainfall tends to peak at the end of the year, especially between September and December, and reaches a minimum in the middle of the year, especially around June to August. This pattern reflects the characteristics of the wet tropical climate in the North Kalimantan region, where the rainy season occurs at the end of the year and the dry season occurs in the middle of the year. The color visualization in the graph, from blue for low rainfall to red for high rainfall, clarifies the seasonal cycle. Outside the seasonal pattern, there is significant inter-annual variability in the magnitude of maximum rainfall. Several years, such as 2017, 2020, and especially 2023, showed a spike in extreme rain, with the peak in 2023 exceeding 160 mm being the highest throughout the decade. In contrast, 2018 recorded a very low minimum value, indicating an arid season. This phenomenon shows that in addition to seasonal factors, global climate conditions such as El Niño or La Niña are very likely to influence the fluctuation of maximum rainfall in Tarakan from year to year.

There is no clear long-term trend in maximum rainfall intensity over the past 11 years. There is no consistent tendency to increase or decrease, but annual climate dynamics influence fluctuations. Therefore, continuous monitoring of rainfall is critical for preparedness for flood risks during extreme rainy seasons such as 2023 and for anticipating droughts such as those that occurred in mid-2018. This information is essential for spatial planning, agriculture, and infrastructure in the Tarakan area, which is sensitive to changes in extreme rainfall patterns.

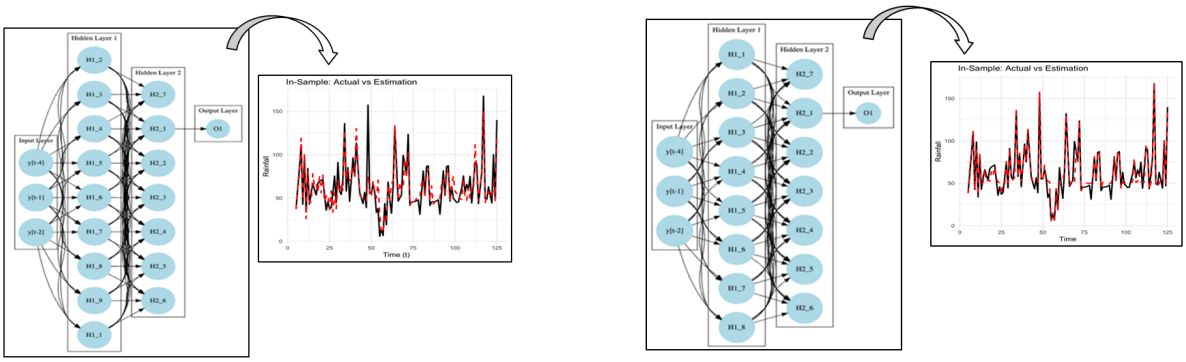
3.2. Modeling with FFNN

The FFNN modeling process on the maximum rainfall data of Juwata Tarakan station begins by dividing the data into training data and testing data with a proportion of 95:5. The training process is carried out on training data with several predetermined scenarios. This study used two hidden layers, each with one to nine neurons. Determining neurons in each hidden layer is carried out in a combination that minimizes the SMAPE value. The activation function used in the hidden layer is Sigmoid, while the output layer uses the Linear function. The process of determining input variables through the identification of the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) so that the input variables y_{t-1}, y_{t-2}, and y_{t-4} are obtained. The results of the FFNN modeling are shown in Table 2.

Table 2. FFNN modeling results

No	Architecture	Dataset	SMAPE	MSE	RMSE
1	FFNN Backpropagation	Training	16.39	200.63	14.16
		Testing	91.98	3055.3	55.27
2	FFNN Rprop	Training	8.63	67.65	8.23
		Testing	51.12	1317.33	36.29

The results of the FFNN modeling, as shown in Table 2, are visualized in Figure 8, which represents the optimal FFNN architecture.



(a) (b)

Figure 8. Optimal FFNN architecture (a) Backpropagation (b) Rprop

The first architecture (Figure 8a) uses standard backpropagation (gradient descent) with linear weight optimization in the output layer. A more complex configuration is required to achieve predictive performance, namely H1 = 9 and H2 = 7 (3–9–7–1 network). Backpropagation requires more neurons and longer iterations to overcome the vanishing gradient problem and adjust one global learning rate. Meanwhile, in the first architecture (Figure 8b), weight updates are performed using the Resilient Propagation (Rprop) algorithm. The grid search results show the optimal configuration of H1 = 8 and H2 = 7, forming a 3–8–7–1 network. Rprop accelerates the convergence process by separating the connection learning rate and eliminating gradient sign fluctuations, thus achieving the lowest RMSE and SMAPE in relatively few iterations. A comparison of the two approaches reveals that Rprop not only reduces the complexity of the architecture by relying on eight units in the first layer but also shows higher convergence stability.

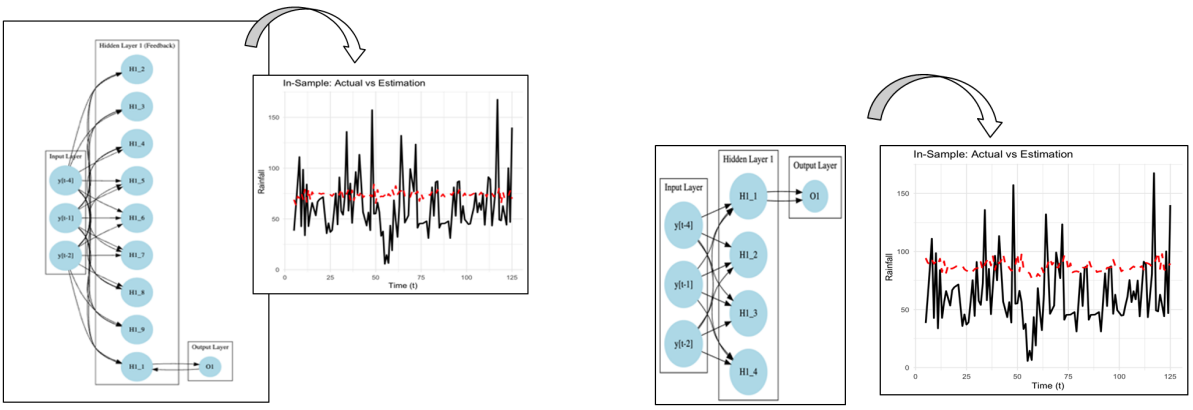
3.3. Modeling with RNN

The RNN modeling process on the maximum rainfall data of Juwata Tarakan station begins by dividing the data into training data and testing data with a proportion of 95:5. The training process is carried out on training data with several predetermined scenarios. The RNN architecture is limited to only one hidden layer with a grid that is tried from 1 to 30. Determining neurons in each hidden layer are selected based on the one that minimizes the SMAPE value. The method of deciding input variables through the identification of the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) so that the input variables y_{t-1}, y_{t-2}, and y_{t-4} are obtained. The results of the RNN modeling are shown in Table 3.

Table 3. FFNN modeling results

No	Architecture	Dataset	SMAPE	MSE	RMSE
1	Elman RNN	Training	38.03	889.07	29.82
		Testing	20.61	303.27	17.41
2	Jordan RNN	Training	46.28	1325.56	36.41
		Testing	14.44	221.76	14.89

The results of the RNN modeling, as shown in Table 3, are visualized in Figure 9, which represents the optimal RNN architecture.



(a) (b)

Figure 9. Optimal RNN architecture (a) Elman (b) Jordan

The Elman RNN and Jordan RNN models are both Recurrent Neural Network (RNN) types that use a single hidden layer, but they have significant differences in feedback and weight updates. In the Elman RNN, feedback occurs in the hidden layer, where the nodes in the hidden layer receive input from the previous output. The weight update process uses the Resilient Propagation (Rprop) algorithm, which separates the learning rate for each connection and eliminates gradient sign fluctuations. This allows the Elman RNN to achieve faster, more stable convergence with fewer iterations than other models. In the H1=9 configuration in the hidden layer, Elman forms a 3–9–1 network, which allows this model to achieve the lowest RMSE and MAPE in a relatively short training time.

In contrast, in Jordan RNN, feedback occurs in the output layer, where previous outputs are used as feedback to influence the current outcome. The weight updates in Jordan models are done using standard backpropagation (gradient descent), which is more commonly used but requires more iterations and time to reach stable convergence. Jordan RNN has an H1=4 configuration, which forms a 3–4–1 network with fewer neurons in the hidden layer. Although this model has a simpler configuration, Jordan RNN requires more neurons and iterations to overcome the vanishing gradient problem, which slows down the convergence process.

3.4. Modeling with Ensemble Forecasting

In this study, the application of the ensembling technique was carried out in three different schemes to test the effectiveness of model combinations in improving prediction accuracy. The first scheme is a combination of two models (2-model ensemble), which includes all combinations of pairs of the four models used in this study. This scheme aims to see how two models can complement each other and produce more accurate predictions than a single model.

The second scheme is a combination of three models (3-model ensemble), which combines all triplet combinations of the four models. With this scheme, better synergy is expected from the three models working together, thereby correcting the shortcomings of each model. The last scheme is an overall combination (4-model ensemble), where the four models are combined into one unit. This scheme aims to maximize the potential of all existing models by utilizing the advantages of each neural network architecture so that it can produce more stable and reliable predictions, especially when handling complex data and high variability. The modeling results with ensemble forecasting are visualized in Figure 10.

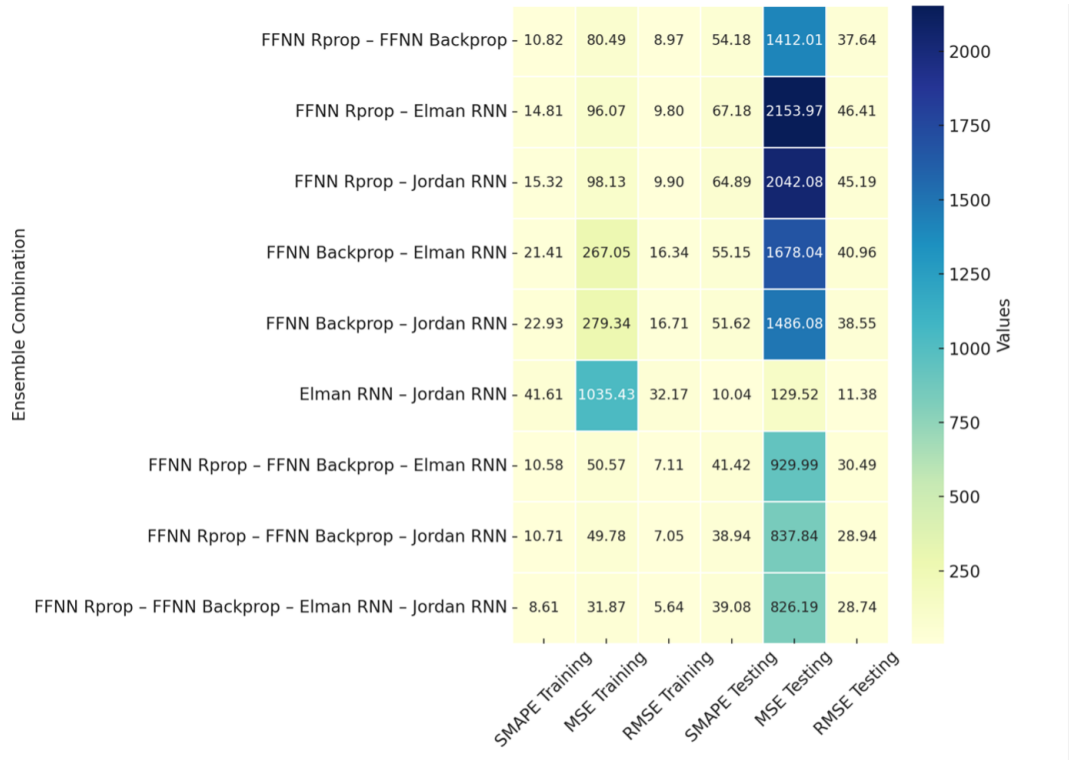


Figure 10. Heatmap of SMAPE, MSE, and RMSE for ensemble combinations

This study evaluated different ensemble configurations involving four neural network architectures: FFNN Backpropagation, FFNN Rprop, Elman RNN, and Jordan RNN. 10 presents the performance of these configurations based on SMAPE, MSE, and RMSE for both the training and testing datasets. The results reveal that the best-performing ensemble differs between the two datasets. Although the four-model ensemble achieves the best performance on the training dataset, the two-model ensemble combining Elman RNN and Jordan RNN yields lower SMAPE, MSE, and RMSE values on the testing dataset. This finding indicates that the best fit to the training data does not necessarily translate into the most accurate predictions for unseen observations.

The testing results further demonstrate that predictive performance varies depending on the combination of neural network architectures. For example, the FFNN Rprop–FFNN Backpropagation ensemble produces a testing SMAPE of 54.18, MSE of 1412.01, and RMSE of 37.64. These results illustrate that combining two models does not automatically guarantee accurate predictions; performance depends on which architectures are combined. In particular, the Elman–Jordan ensemble achieves lower testing errors than the four-model ensemble, making it the most promising configuration among the evaluated ensembles for predicting maximum rainfall at Juwata Station, Tarakan.

The difference between training and testing performance also shows that increasing the number of constituent models does not necessarily improve predictive accuracy. Although the four-model ensemble provides a better

fit to the training data, adding FFNN Backpropagation and FFNN Rprop to the Elman–Jordan combination does not produce better results on the testing dataset. Therefore, the specific combination of neural network architectures appears to be more important than the number of models alone. These findings highlight the importance of evaluating ensemble configurations using out-of-sample performance. Based on the testing results, the Elman–Jordan ensemble is the preferred configuration for maximum rainfall prediction at Juwata Station, Tarakan, among the configurations evaluated in this study.

3.5. Discussion

This study evaluated ensemble configurations combining FFNN Backpropagation, FFNN Rprop, Elman RNN, and Jordan RNN for maximum rainfall prediction at Juwata Station, Tarakan. By comparing combinations of two, three, and four architectures, the analysis examined whether combining different neural network structures could improve predictive performance. The effectiveness of each configuration was assessed using SMAPE, MSE, and RMSE on both the training and testing datasets. The results show that ensemble performance depends on both the constituent architectures and the evaluation dataset, indicating that combining more models does not necessarily lead to more accurate predictions.

FFNN Backpropagation is a widely used learning approach for feedforward neural networks. Its training process adjusts network weights by minimizing a loss function, such as the mean squared error (MSE), through gradient-based optimization. However, its convergence and predictive performance depend on factors such as network architecture, weight initialization, and training settings [55]. FFNN Rprop (Resilient Propagation), in contrast, adjusts weight updates primarily according to the sign of the gradient rather than its magnitude. This mechanism can improve convergence efficiency under certain training conditions, although its performance also depends on the network configuration and learning parameters [56].

Elman RNN is designed to process sequential data through context units that retain information from previous time steps, enabling the model to represent temporal dependencies. Jordan RNN also incorporates recurrent feedback, but its context units receive information from the network output, distinguishing its feedback structure from that of the Elman architecture [25]. These differences provide a basis for investigating whether feedforward and recurrent architectures offer complementary predictive information when combined for maximum rainfall forecasting. However, the effectiveness of a particular combination must be established through empirical evaluation rather than inferred solely from the theoretical characteristics of its constituent models.

The comparative results reveal an important difference between training and testing performance. The four-model ensemble, which combines FFNN Backpropagation, FFNN Rprop, Elman RNN, and Jordan RNN, achieves the best performance on the training dataset. However, the two-model ensemble combining Elman RNN and Jordan RNN produces lower SMAPE, MSE, and RMSE values on the testing dataset than the four-model ensemble. This finding identifies the Elman–Jordan ensemble as the best-performing configuration among those evaluated for predicting unseen observations. In contrast, the FFNN Rprop–FFNN Backpropagation ensemble yields testing errors of 54.18 for SMAPE, 1412.01 for MSE, and 37.64 for RMSE, illustrating that the predictive performance of two-model ensembles varies according to the architectures being combined.

The difference between training and testing performance suggests that increasing the number of constituent models does not necessarily improve the generalization ability of an ensemble. Although the four-model ensemble achieves better performance on the training dataset, the additional constituent models may improve the fit to the training observations without providing complementary predictive information that consistently benefits forecasts on unseen data. This outcome may also be related to the use of in-sample SMAPE to determine ensemble weights, which can favor models with better training performance but does not guarantee optimal performance on the testing dataset. In contrast, the Elman–Jordan ensemble achieves lower SMAPE, MSE, and RMSE on the testing dataset, suggesting that the choice of constituent architectures may be more important than ensemble size alone. However, these explanations remain plausible interpretations, as the underlying mechanisms were not directly investigated in this study.

These findings demonstrate that increasing the number of constituent models does not necessarily improve out-of-sample predictive accuracy. Although the four-model ensemble fits the training data more effectively, its testing performance is lower than that of the Elman–Jordan ensemble. The results suggest that the specific combination

of architectures is an important consideration when developing an ensemble forecasting model. Therefore, model selection should prioritize performance on the testing dataset rather than relying exclusively on training results. For maximum rainfall prediction at Juwata Station, Tarakan, the Elman–Jordan ensemble is the preferred configuration among the evaluated combinations based on the three testing metrics.

The contribution of this study lies in the comparative evaluation of ensemble configurations involving FFNN Backpropagation, FFNN Rprop, Elman RNN, and Jordan RNN for maximum rainfall prediction. The configurations combine two, three, or four architectures using performance-based weighted averaging, with weights determined by the SMAPE values of the constituent models. Rather than proposing a fundamentally new ensemble algorithm, this study provides empirical evidence of how different combinations perform on the available rainfall dataset. In particular, the results show that the configuration with the largest number of constituent models does not necessarily provide the most accurate testing predictions. Nevertheless, these findings are limited to the study period, station, and model configurations examined.

Furthermore, the models primarily use lagged rainfall values identified through autocorrelation function (ACF) and partial autocorrelation function (PACF) analysis. Consequently, the current framework does not explicitly incorporate large-scale climate drivers or other external meteorological predictors. Another limitation concerns the use of in-sample SMAPE to determine the ensemble weights. Although this approach provides a consistent weighting criterion based on training performance, it may favor models that fit the training data well but generalize less effectively to unseen observations. Future research could investigate validation-based weighting strategies to improve the reliability of ensemble forecasts. In addition, incorporating exogenous variables, such as relevant climate indices and meteorological predictors, may help improve maximum rainfall prediction. Comparisons with conventional time-series models would also provide a more comprehensive assessment of predictive performance. Future studies should also consider time-series cross-validation and rolling-origin evaluation to assess predictive performance across multiple forecast origins. These approaches could provide more comprehensive performance estimates than a single training-to-testing split, particularly when the available testing observations are limited. Future research should also investigate the sensitivity of ensemble performance to different chronological training-to-testing split proportions to determine whether the relative performance of the evaluated configurations remains consistent across different forecast periods. Given the relatively small dataset, future research should also investigate the potential risk of overfitting by evaluating regularization techniques, such as dropout, weight decay, and early stopping, where appropriate for the selected architectures. These approaches could help assess whether model complexity can be controlled and predictive performance on unseen observations can be improved. Finally, evaluating the selected ensemble across different stations and forecasting periods would help establish the robustness and broader applicability of the findings.

4. Conclusion

This study evaluated ensemble configurations combining four neural network architectures—FFNN Backpropagation, FFNN Rprop, Elman RNN, and Jordan RNN for maximum rainfall prediction at Juwata Station, Tarakan. The models were combined in two-, three-, and four-component configurations using performance-based weighted averaging. Their predictive performance was evaluated using SMAPE, MSE, and RMSE on the training and testing datasets. The results demonstrate that the best-performing ensemble configuration differs between the training and testing datasets. The four-model ensemble achieves the best performance on the training dataset, whereas the two-model ensemble combining Elman RNN and Jordan RNN produces lower SMAPE, MSE, and RMSE values on the testing dataset. This finding indicates that incorporating more neural network architectures does not necessarily improve predictive accuracy for unseen observations. Instead, the selection of constituent models plays an important role in determining ensemble performance. Among the evaluated configurations, the Elman–Jordan ensemble provides the most accurate maximum rainfall predictions on the testing dataset based on the three evaluation metrics.

The findings contribute empirical evidence regarding the performance of different neural network ensemble configurations for maximum rainfall prediction. However, the results are limited to the observed rainfall data

from Juwata Station and the model configurations examined in this study. Further research is needed to assess the generalizability of these findings across different locations and forecasting periods. Future studies could also incorporate exogenous variables, such as relevant climate indices and meteorological predictors, and compare the ensemble configurations with conventional time-series models. In addition, investigating advanced deep learning architectures and probabilistic forecasting approaches could provide further insights into predictive performance and uncertainty in maximum rainfall forecasting.

Acknowledgments

The authors would like to express sincere gratitude to the Meteorological, Climatological, and Geophysical Agency (BMKG) Juwata Station, Tarakan, North Kalimantan, for their support in providing the data on the number of rainy days and maximum rainfall from 2014 to 2024. The data provided was invaluable and made a significant contribution to the success of this research.

The Declaration of Conflict of Interest/Common Interest

The authors declare that there are no conflicts of interest concerning the publication of this manuscript. Furthermore, all ethical considerations, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies havebeen completely observed by the authors.

REFERENCES

1. Marjenah, Ramadani, W. L. Farahdita, and Kiswanto, "Species Diversity and Carbon Storage of Tanah Grogot Urban Forest in Paser Regency of East Kalimantan", in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics, 2025. doi: 10.1088/1755-1315/1447/1/012012.
2. F. Lestari *et al.*, "Estimating the Flood, Landslide, and Heavy Rainfall Susceptibility of Vaccine Transportation after 2021 Flooding in South Kalimantan Province, Indonesia,", *Sustainability (Switzerland)*, vol. 16, no. 4, Feb. 2024, doi: 10.3390/su16041554.
3. J. Lala, J. Bazo, V. Anand, and P. Block, "Optimizing forecast-based actions for extreme rainfall events," *Clim Risk Manag*, vol. 34, Jan. 2021, doi: 10.1016/j.crm.2021.100374.
4. I. F. Amri *et al.*, "Prediction of extreme weather using nonparametric regression approach with Fourier series estimators," *Data and Metadata*, vol. 4, pp. 1–12, 2024, doi: 10.56294/dm2024319.
5. S. Manandhar, S. Dev, Y. H. Lee, S. Winkler, and Y. S. Meng, "Systematic Study of Weather Variables for Rainfall Detection," in *International Geoscience and Remote Sensing Symposium*, IEEE, 2018, pp. 3027–3030.
6. A. Susandi *et al.*, "Spatiotemporal Evaluation of Water Resources in Citarum Watershed during Weak La Nina and Weak El Nino," *Hydrology*, vol. 11, no. 6, Jun. 2024, doi: 10.3390/hydrology11060073.
7. P. K. Das, R. L. Sahu, and P. C. Swain, "Comparative analysis of machine learning models for rainfall prediction," *J Atmos Sol Terr Phys*, vol. 264, p. 106340, 2024, doi: https://doi.org/10.1016/j.jastp.2024.106340.
8. A. Sudradjat, B. S. Muntalif, N. Marasabessy, F. Mulyadi, and M. I. Firdaus, "Relationship between chlorophyll-a, rainfall, and climate phenomena in tropical archipelagic estuarine waters," *Heliyon*, vol. 10, no. 4, pp. 1–12, Feb. 2024, doi: 10.1016/j.heliyon.2024.e25812.
9. A. Kurniadi, E. Weller, J. Salmond, and E. Aldrian, "Future projections of extreme rainfall events in Indonesia," *International Journal of Climatology*, vol. 44, no. 1, pp. 160–182, Jan. 2024, doi: 10.1002/joc.8321.
10. T. Ciechulski and S. Osowski, "Wind Power Short-Term Time-Series Prediction Using an Ensemble of Neural Networks," *Energies (Basel)*, vol. 17, no. 1, Jan. 2024, doi: 10.3390/en17010264.
11. A. Y. Barrera-Animas, L. O. Oyedele, M. Bilal, T. D. Akinosho, J. M. D. Delgado, and L. A. Akanbi, "Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting," *Machine Learning with Applications*, vol. 7, p. 100204, Mar. 2022, doi: 10.1016/j.mlwa.2021.100204.
12. W. W. S. Wei, "Time Series Analysis: Univariate and Multivariate Methods," 2nd ed. New York: Pearson Education, Inc, 2006.
13. I. F. Amri, N. Chamidah, T. Saifudin, B. Lestari, D. Aydin, and F. H. N. Rohim, "A Penalized Least Squares Estimation of Fourier Series Semiparametric Regression: Theory, Simulation, and Application", in *Statistics, Optimization and Information Computing*, vol. 15, no. 4, pp. 3255–3281, Mar. 2026. doi: 10.19139/soic-2310-5070-3139.
14. P. Unnikrishnan and V. Jothiprakash, "Hybrid SSA-ARIMA-ANN Model for Forecasting Daily Rainfall," *Water Resources Management*, vol. 34, no. 11, pp. 3609–3623, Sep. 2020, doi: 10.1007/s11269-020-02638-w.

15. M. Mislan and A. T. R. Dani, "Forecasting Maximum Water Level Data for Post Sangkuliman using An Artificial Neural Network Backpropagation Algorithm," *JTAM (Jurnal Teori dan Aplikasi Matematika)*, vol. 8, no. 2, p. 465, Mar. 2024, doi: 10.31764/jtam.v8i2.20112.
16. T. T. K. Tran, S. M. Bateni, S. J. Ki, and H. Vosoughifar, "A review of neural networks for air temperature forecasting," May 01, 2021, *MDPI AG*. doi: 10.3390/w13091294.
17. A. T. Fitriyah, N. Chamidah, T. Saifudin, B. Lestari, and D. Aydin, "Least Squares Spline Estimation Method in Semiparametric Time Series Regression for Predicting Indonesia Composite Index", in *Statistics, Optimization and Information Computing*, vol. 14, no. 6, pp. 3776–3803, Nov. 2025. doi: 10.19139/soic-2310-5070-2704.
18. A. F. Atiya, S. M. El-Shoura, S. I. Shaheen, and M. S. El-Sherif, "A Comparison Between Neural-Network Forecasting Techniques-Case Study: River Flow Forecasting," 1999.
19. G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," in *European Journal of Operational Research*, Jan. 2005, pp. 501–514. doi: 10.1016/j.ejor.2003.08.037.
20. Suhartono, D. E. Ashari, D. D. Prastyo, H. Kuswanto, and M. H. Lee, "Deep neural network for forecasting inflow and outflow in Indonesia," *Sains Malays*, vol. 48, no. 8, pp. 1787–1798, 2019, doi: 10.17576/jsm-2019-4808-26.
21. M. Narvekar and P. Fargose, "Daily Weather Forecasting using Artificial Neural Network," 2015.
22. P. Escudero, W. Alcocer, and J. Paredes, "Recurrent neural networks and ARIMA models for euro/dollar exchange rate forecasting," *Applied Sciences (Switzerland)*, vol. 11, no. 12, Jun. 2021, doi: 10.3390/app11125658.
23. M. Putra, M. S. Rosid, and D. Handoko, "A Review of Rainfall Estimation in Indonesia: Data Sources, Techniques, and Methods," *Signals*, vol. 5, no. 3, pp. 542–561, Aug. 2024, doi: 10.3390/signals5030030.
24. L. Yu *et al.*, "Application of a New Hybrid Model with Seasonal Auto-Regressive Integrated Moving Average (ARIMA) and Nonlinear Auto-Regressive Neural Network (NARNN) in Forecasting Incidence Cases of HFMD in Shenzhen, China," *PLoS One*, vol. 9, no. 6, pp. 1–9, 2014, doi: 10.1371/journal.pone.0098241.t001.
25. G. Drałus, D. Mazur, J. Kusznier, and J. Drałus, "Application of Artificial Intelligence Algorithms in Multilayer Perceptron and Elman Networks to Predict Photovoltaic Power Plant Generation," *Energies (Basel)*, vol. 16, no. 18, Sep. 2023, doi: 10.3390/en16186697.
26. X. Zhang, Q. Zhang, G. Zhang, Z. Nie, and Z. Gui, "A hybrid model for annual runoff time series forecasting using elman neural network with ensemble empirical mode decomposition," *Water (Switzerland)*, vol. 10, no. 4, Apr. 2018, doi: 10.3390/w10040416.
27. H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K. W. Chau, "Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting," *Water (Switzerland)*, vol. 12, no. 5, May 2020, doi: 10.3390/w12051500.
28. I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," *Information*, vol. 15, no. 9, p. 517, Aug. 2024, doi: 10.3390/info15090517.
29. W. Ge, P. Lalbakhsh, L. Isai, A. Lenskiy, and H. Suominen, "Neural Network-Based Financial Volatility Forecasting: A Systematic Review," Jan. 17, 2022, *Association for Computing Machinery*. doi: 10.1145/3483596.
30. M. Adya and F. Collopy, "How Effective are Neural Networks at Forecasting and Prediction? A Review and Evaluation," *J Forecast*, vol. 17, pp. 481–495, 1998.
31. N. H. A. Rahman, M. H. Lee, Suhartono, and M. T. Latif, "Artificial neural networks and fuzzy time series forecasting: an application to air quality," *Qual Quant*, vol. 49, no. 6, pp. 2633–2647, Nov. 2015, doi: 10.1007/s11135-014-0132-6.
32. Y. Wang, R. Zou, F. Liu, L. Zhang, and Q. Liu, "A review of wind speed and wind power forecasting with deep neural networks," *Appl Energy*, vol. 304, Dec. 2021, doi: 10.1016/j.apenergy.2021.117766.
33. A. E. Muñoz-Zavala, J. E. Macías-Díaz, D. Alba-Cuéllar, and J. A. Guerrero-Díaz-de-León, "A Literature Review on Some Trends in Artificial Neural Networks for Modeling and Simulation with Time Series," Feb. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/a17020076.
34. I. E. Livieris, E. Pintelas, S. Stavroyiannis, and P. Pintelas, "Ensemble Deep learning models for forecasting cryptocurrency time-series," *Algorithms*, vol. 13, no. 5, May 2020, doi: 10.3390/A13050121.
35. A. Bhambu, R. Gao, and P. N. Suganthan, "Recurrent ensemble random vector functional link neural network for financial time series forecasting," *Appl Soft Comput*, vol. 161, Aug. 2024, doi: 10.1016/j.asoc.2024.111759.
36. A. Galicia, R. Talavera-Llames, A. Troncoso, I. Koprinska, and F. Martínez-Álvarez, "Multi-step forecasting for big data time series based on ensemble learning," *Knowl Based Syst*, vol. 163, pp. 830–841, Jan. 2019, doi: 10.1016/j.knosys.2018.10.009.
37. D. Huang, Z. Liu, and D. Wu, "Research on Ensemble Learning-Based Feature Selection Method for Time-Series Prediction," *Applied Sciences (Switzerland)*, vol. 14, no. 1, Jan. 2024, doi: 10.3390/app14010040.
38. N. Mohamed, M. H. Ahmad, Z. Ismail, and S. Suhartono, "Short Term Load Forecasting Using Double Seasonal ARIMA Model," in *Proceedings of the Regional Conference on Statistical Sciences 2010*, 2010, pp. 57–73.
39. S. Makridakis, R. J. Hyndman, and F. Petropoulos, "Forecasting in social settings: The state of the art," *Int J Forecast*, vol. 36, no. 1, pp. 15–28, Jan. 2020, doi: 10.1016/j.ijforecast.2019.05.011.
40. S. Makridakis, E. Spiliotis, V. Assimakopoulos, and V. Assimakopoulos, "The Accuracy of Machine Learning (ML) Forecasting Methods versus Statistical Ones: Extending the Results of the M3-Competition," 2017. [Online]. Available: https://www.researchgate.net/publication/320298859
41. G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," 2003. [Online]. Available: www.elsevier.com/locate/neucom
42. L. Safitri, S. Mardiyati, and H. Rahim, "Forecasting the mortality rates of Indonesian population by using neural network," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2018. doi: 10.1088/1742-6596/974/1/012030.
43. M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Syst Appl*, vol. 37, no. 1, pp. 479–489, Jan. 2010, doi: 10.1016/j.eswa.2009.05.044.
44. M. Pujara and N. Paudel, "Rainfall Prediction using Long Short-Term Memory and Gated Recurrent Unit with Various Meteorological Parameters," *Nepalese Journal of Statistics*, vol. 8, pp. 47–60, Dec. 2024, doi: 10.3126/njs.v8i1.73165.
45. A. Wanto *et al.*, "Epoch Analysis and Accuracy 3 ANN Algorithm using Consumer Price Index Data in Indonesia," *Scitepress*, May 2021, pp. 35–41. doi: 10.5220/0010037400350041.

46. M. Kim, S. Lee, and T. Jeong, "Time Series Prediction Methodology and Ensemble Model Using Real-World Data," *Electronics (Switzerland)*, vol. 12, no. 13, Jul. 2023, doi: 10.3390/electronics12132811.
47. P. Melin, J. C. Monica, D. Sanchez, and O. Castillo, "Multiple ensemble neural network models with fuzzy response aggregation for predicting covid-19 time series: The case of mexico," *Healthcare (Switzerland)*, vol. 8, no. 2, 2020, doi: 10.3390/healthcare8020181.
48. P. P. Phyo and Y. C. Byun, "Hybrid ensemble deep learning-based approach for time series energy prediction," *Symmetry (Basel)*, vol. 13, no. 10, Oct. 2021, doi: 10.3390/sym13101942.
49. K. He, Q. Yang, L. Ji, J. Pan, and Y. Zou, "Financial Time Series Forecasting with the Deep Learning Ensemble Model," *Mathematics*, vol. 11, no. 4, Feb. 2023, doi: 10.3390/math11041054.
50. Z. Xiao, X. Huang, J. Liu, C. Li, and Y. Tai, "A novel method based on time series ensemble model for hourly photovoltaic power prediction," *Energy*, vol. 276, Aug. 2023, doi: 10.1016/j.energy.2023.127542.
51. J. Jumadi *et al.*, "AI-Driven Ensemble Learning for Spatio-Temporal Rainfall Prediction in the Bengawan Solo River Watershed, Indonesia," *Sustainability (Switzerland)*, vol. 17, no. 20, Oct. 2025, doi: 10.3390/su17209281.
52. N. Song, R. Chang-Silva, K. Lee, and S. Park, "Dynamic Model Selection in a Hybrid Ensemble Framework for Robust Photovoltaic Power Forecasting," *Sensors*, vol. 25, no. 14, Jul. 2025, doi: 10.3390/s25144489.
53. J. Ahn and J. Lee, "Impact on Predictive Performance of Air Pollutants in PV Forecasting Using Multi-Model Ensemble Learning: Evidence from the Port Logistics Hinterland Area," *Systems*, vol. 13, no. 11, p. 943, Oct. 2025, doi: 10.3390/systems13110943.
54. H. Alabdulrazzaq *et al.*, "On the accuracy of ARIMA based prediction of COVID-19 spread," *Results Phys*, vol. 27, p. 104509, 2021, doi: 10.1016/j.rinp.2021.104509.
55. A. Atesongun and M. Gulsen, "A Hybrid Forecasting Structure Based on Arima and Artificial Neural Network Models," *Applied Sciences (Switzerland)*, vol. 14, no. 16, Aug. 2024, doi: 10.3390/app14167122.
56. N. H. A. Rahman *et al.*, "Artificial neural networks and fuzzy time series forecasting: an application to air quality," *Qual Quant*, vol. 49, no. 6, pp. 2633–2647, Nov. 2015, doi: 10.1007/s11135-014-0132-6.