Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 11, No. 1, February, Pp. 123-134

Integrating meteorological and PV data for short-term solar irradiance forecasting using BPNN

Ahmad Rizal Agustian*[1], Unit Three Kartini[1], Muhammad Miftahul Rizqi[1], Sa'adatud Daroini[1]
Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia[1]

Article Info
Keywords:
Forecasting, Solar Irradiance, BPNN, Neural Network, Solar Power Plant, Time-Series, MATLAB

Article history:
Received: July 30, 2025
Accepted: October 24, 2025
Published: February 01, 2026

Cite:
Ahmad Rizal Agustian, U. T. Kartini, M. M. Rizqi, and S. Daroini, "Integrating Meteorological and PV Data for Short-Term Solar Irradiance Forecasting Using BPNN", *KINETIK*, vol. 11, no. 1, Feb. 2026.
https://doi.org/10.22219/kinetik.v11i1.2449

*Corresponding author.
Unit Three Kartini
E-mail address:
unitthree@unesa.ac.id

Abstract
Solar power plants are highly dependent on solar radiation intensity, which fluctuates due to changes in atmospheric conditions. To maintain system stability and efficiency, an accurate short-term solar radiation prediction model is essential. This study developed a model for forecasting global solar radiation one hour ahead using the Backpropagation Neural Network (BPNN) method. The dataset was obtained from a photovoltaic (PV) system at Building A8 of Surabaya State University, recorded over four days (June 14-17, 2025) at two-minute intervals. Five input variables were used: clearness index, solar radiation, air temperature, air humidity, and PV output power, resulting in a total of 3,020 data samples. The model was trained through a trial-and-error process by varying the number of neurons, hidden layers, and epochs to determine the optimal configuration. The forecast capability of the model was assessed through four statistical indicators: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). The best performance was achieved with a network architecture of 15 input neurons representing input variables resulting from data transformation using the sliding window method, one hidden with 25 neurons, and a single unit in the output layer trained for 2000 epochs, resulting in R2 = 0.98, MAPE = 5.89%, and MSE = 0.00027. The novelty of this research lies in the integration of meteorological data with actual PV power output as model input, enabling the network to capture more realistic nonlinear temporal relationships. The proposed short-term forecasting model provides a practical approach to predicting solar radiation based on historical data and can support efficient energy management and photovoltaic system performance analysis.

1. Introduction

Solar radiation forecasting is an important aspect of solar energy and meteorology studies because solar radiation plays a key role in energy planning and maintaining the reliability of solar energy systems [1]. The fundamental challenge in solar radiation forecasting lies in its highly variable temporal and spatial nature due to the influence of atmospheric dynamics. Atmospheric variables such as cloud formation, relative humidity, air temperature, pressure, and aerosols cause fluctuations in radiation intensity that are difficult to predict consistently [2]. These conditions make solar radiation forecasting a complex problem [3].

This forecasting process requires the application of various methods designed to provide estimates of future solar radiation levels with varying degrees of accuracy. One widely used approach is data-based modelling. These data-based models apply machine learning techniques, such as artificial neural networks, to process large amounts of historical data related to solar radiation, while also considering relevant meteorological and environmental variables. This method is capable of producing forecasts that not only capture short-term fluctuations but also predict long-term trends in solar radiation patterns by revealing complex relationships in the data [4]. These forecasts are very useful for supporting load scheduling, energy distribution management, and solar power system control strategies to minimize the impact of radiation fluctuations [5].

Several researchers have conducted studies using various data-based forecasting methods. Lin et al. [6] integrated Long Short-Term Memory (LSTM) with sky image features, meteorological data (temperature, humidity, and wind speed), and solar geometry to capture spatial and temporal relationships in radiation data. This model is capable of representing radiation fluctuation patterns smoothly, but requires a large dataset and a long training time. Gupta et al. [7] proposed a Random Forest–Particle Swarm Optimization (RF–PSO) hybrid model using temperature, humidity, and wind speed data as input variables. This hybrid approach improves stability and accuracy compared to single machine learning models, but is still limited to meteorological variables without considering actual PV system output data.

In addition, Simanjuntak and Wibowo [8] used Backpropagation Neural Network (BPNN) to estimate solar radiation intensity in Jayapura City based on local observation data. Test results showed similarities between the prediction results and observation data with a Root Mean Square Error (RMSE) value of 1.970 and a correlation coefficient of 0.188, indicating that the model was able to recognize general patterns but was not yet optimal in capturing the complex relationships between atmospheric variables. Huang, Ma, and Chen [9] also applied BPNN using meteorological data such as rainfall, relative humidity, and clearness index to predict daily radiation in Guangzhou, China. The model showed an increase in error during periods of high rainfall with an RMSE of 0.4708, confirming that extreme atmospheric conditions are still difficult to model accurately.

Based on a review of previous studies, most solar radiation forecasting studies still focus on using meteorological data and sky images as the main variables. This approach is effective in representing atmospheric conditions, but it is not yet fully capable of capturing the direct response of photovoltaic systems to actual radiation changes. This approach shows that further research is needed to combine actual PV system output data with meteorological parameters to improve the model's ability to represent the actual conditions of solar power systems. The integration of these two types of data allows the model to not only study weather patterns, but also understand the dynamic relationship between radiation and PV output, which is essential for optimizing the operation of photovoltaic systems.

From a methodological perspective, various models have demonstrated good performance, such as LSTM, which excels at learning long-term temporal patterns, and RF–PSO, which is effective at improving prediction stability through a combination of optimization algorithms. However, both approaches have limitations: LSTM requires large datasets and long training times, while RF–PSO has difficulty capturing the subtle nonlinear relationships between continuous variables that often appear in solar radiation data. In this context, Backpropagation Neural Network (BPNN) offers a balance between complexity and efficiency. BPNN is capable of modelling nonlinear relationships between meteorological variables and PV power output with more moderate data requirements and relatively fast training time. Therefore, this study proposes a BPNN-based global solar irradiance forecasting model for the next hour that integrates meteorological data (temperature, humidity, clearness index) and actual PV output data. This approach is expected to produce an accurate, efficient, and applicable model to support optimal energy management and solar power system control.

2. Research Method

Research The research framework consists of several stages: data collection and preprocessing, model building with parameter adjustment, and model performance evaluation. A visual overview of the research stages is presented in Figure 1, which illustrates the logical flow and relationships between stages in the methodological framework.

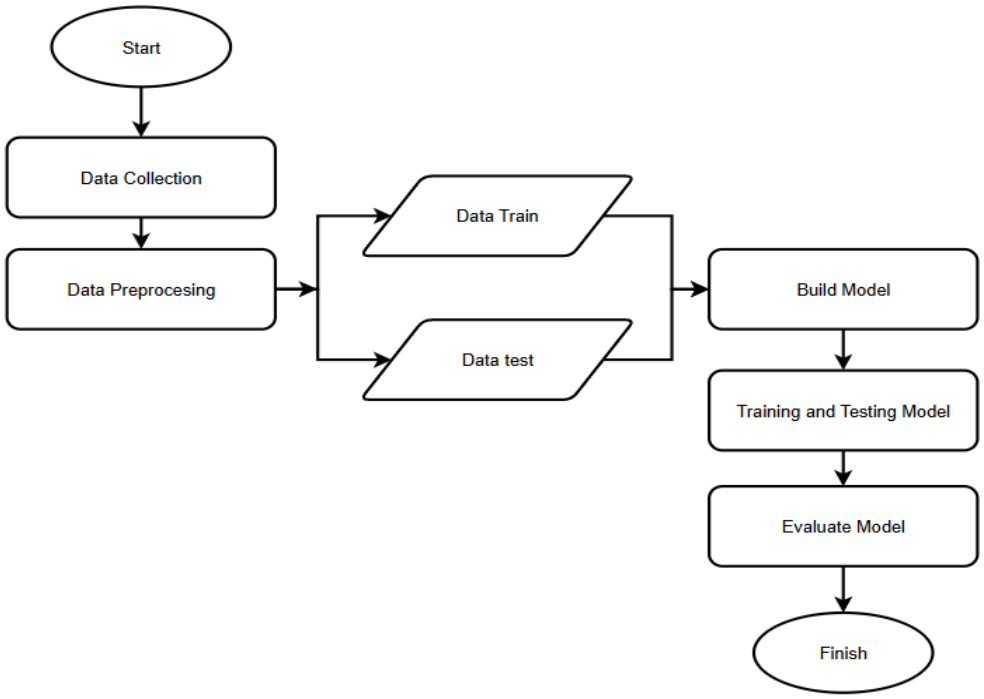

Figure 1. Research Stages

2.1 Data Acquisition Process

Data acquisition was carried out during a four-day observation period, from June 14 to June 17, 2025, at a photovoltaic (PV) prototype system installed in Building A8, Universitas Negeri Surabaya. Data Acquisition was conducted between 7:00 AM and 12:00 PM, capturing intervals with the highest solar radiation to ensure adequate data variability. Measurements were taken at two-minute intervals, resulting in a total of 3,020 data samples. The selected time window encompassed a variety of sky conditions—clear, cloudy, and transitional—reflecting the dynamic nature of the tropical climate and enabling modeling of solar radiation under various atmospheric conditions.

This study uses a dataset consisting of five main parameters: global solar radiation (W/m²), photovoltaic power output (W), ambient temperature (°C), relative humidity (%), and clearness index (Kt). These parameters were selected because they significantly affect the performance of photovoltaic systems. Global solar radiation represents the primary energy input, while PV power output indicates the system's energy conversion. Ambient temperature and relative humidity affect the electrical and thermal behavior of PV modules. The clearness index (Kt) serves as an indicator of atmospheric clearness and sky conditions. A sample of the collected data is presented in Table 1.

Table 1. Sample of PV system dataset

Date	Time	Solar Irradiance (W/m^2)	Ambient Temperature (°C)	Relative Humidity (Rh%)	PV Output Power (Watt)	Clearness Index
14/06/25	07:00	228,70	30,50	69,00	4,80	0,93
14/06/25	07:02	229,10	30,70	69,00	6,00	0,90
14/06/25	07:04	228,70	30,70	69,00	8,20	0,86
14/06/25	07:06	228,80	30,70	69,00	6,10	0,83
14/06/25	07:08	228,60	30,70	69,00	4,90	0,80
14/06/25	07:10	228,80	30,80	69,00	7,50	0,77

The clearness index (Kt) is defined as the ratio between the global solar radiation measured at the Earth's surface and the extraterrestrial solar radiation incident at the top of the atmosphere. This index reflects the degree to which the atmosphere is attenuated by clouds, aerosols, and other components. It serves as a metric for the transparency of the atmosphere to solar radiation. It captures both daily and seasonal variability in atmospheric clearness and is widely used in modeling the availability of solar energy. In theory, Kt values range from 0 to 1. however, in practice, it typically varies from approximately 0.8 under clear-sky conditions to values near 0 during overcast skies [10],[11]. Systematically, it is show in Equation 1.

$$Kt = \frac{G}{G_o} \tag{1}$$

Where G represents global solar radiation measured at the Earth's surface, and G_o represents extraterrestrial solar radiation estimated using solar constants and solar geometry. The calculations involved in estimating Go are detailed in the following equation:

$$G_o = I_{sc}\, x\, \left(1 + 0{,}033\, \cos\left(\frac{360n}{365} x \frac{\pi}{180}\right)\right) x\, (\cos\varphi \cos\delta \cos\omega + \sin\varphi \sin\delta) \tag{2}$$

Equation 2 is used to compute the extraterrestrial radiation G_o, based on the solar constant I_{sc}, whose standard value is 1367 W/m^2, the day of the year n, and solar geometry terms. The correction factor $1 + 0{,}033\, \cos\left(\frac{360n}{365} x \frac{\pi}{180}\right)$ adjusts for the variation in Earth–Sun distance throughout the year. The trigonometric terms account for the location's latitude φ, the solar declination δ, and the hour angle ω.

$$\delta = 23{,}45^o x \sin\left(\frac{360x(284+n)}{365} x \frac{\pi}{180}\right) \tag{3}$$

Equation 3 provides the solar declination angle δ, which varies daily due to the tilt of the Earth's axis. It is essential for determining the solar angle of incidence, a critical component in estimating the amount of radiation that reaches a specific point on Earth.

$$\omega = 15^o(t_{solar} - 12) \quad (4)$$

Equation 4 calculates the solar hour angle ω, representing the angular distance between the sun and the local meridian, based on solar time t_{solar}. This helps identify the sun's position in the sky at any given hour.

2.2 Data Preprocessing

Prior to the training phase, the dataset underwent a preprocessing stage to ensure that the input data were properly structured, clean, and suitable for model development. Several preprocessing steps were applied, including data segmentation, sliding window transformation, and normalization. These steps were essential to enhance data consistency and capture temporal dependencies between consecutive observations, which are critical for short-term forecasting tasks.

The dataset is divided into training and testing sets based on chronological order to maintain the temporal integrity of the data and prevent information leakage. The training data consists of the first three days (June 14-16), while the test data consists of the last day (June 17). To accommodate one-hour-ahead forecasting, and since actual observations are only available until 12:00 PM, dummy data are added to the test set to simulate predictions until 1:00 PM. These dummy data entries are filled with the maximum value from the training set to avoid forward-looking bias. After the dataset is split, a sliding window approach is applied to organize the time series data into a supervised learning format. In this study, a three-step time window size was selected, meaning that each input vector consists of data from six consecutive time intervals, while the corresponding target is the global solar radiation at the next time step. This strategy allows the model to learn short-term temporal dependencies in the meteorological power signal and output [12].

Subsequently, min-max normalization is applied to scale each input variable into the range [0, 1]. This normalization method ensures that all features contribute proportionally to the training process and helps accelerate convergence in the neural network [13],[14]. Normalization is performed using the minimum and maximum values of the training data.

$$x' = \frac{(x - a)}{b - a} \quad (5)$$

Equation 5 x is the original measurement, a and b represent the minimum and maximum values of the variable, and x' correspond to be the normalized result.

Importantly, the normalization parameters (min and max) are calculated only from the training set to prevent data leakage. These parameters are then applied to normalize the test set. All normalization and denormalization processes are implemented using automated MATLAB scripts to ensure consistency and reproducibility.

2.3 Backpropagation Model

Among various learning approaches, backpropagation is extensively adopted in neural network training, as it refines connection weights iteratively based on the computed error between predicted and target outputs. This algorithm consists of three main stages, namely (1) the feedforward stage, (2) the backward propagation stage, and (3) the weight and bias update stage. In stage (1), each neuron calculates the activation value based on the input from the previous layer and the connection weights that connect it to produce the network output. In stage (2), the difference between the actual output and the desired target is calculated to obtain the error gradient used in the process of adjusting the weights between units. Furthermore, in stage (3), the network weights and biases are updated based on the error gradient obtained in order to minimize the loss function and improve the accuracy of the model's predictions [15].

To build an artificial neural network model with the Backpropagation algorithm, a mathematical formulation is needed to describe the computational process at each stage of network training. This formulation includes calculating activation values at the feedforward stage, determining the error between the actual and target outputs, and updating the weights and biases based on the error gradient at the backpropagation stage. The mathematical relationship between these variables forms the basis for the prediction model in this study, as described in the following equation [16].

(1) Feedforward Stage

In this phase, every neuron within the hidden layer obtains signals from the preceding layer and determines its activation value based on Equations 6 and 7:

$$z_j = f(\sum_{i=1}^{n} v_{ji}x_i + v_{j0}) \quad (6)$$

$$y_k = f(\sum_{j=1}^{p} w_{kj}z_j + w_{k0}) \quad (7)$$

Equations 6 and 7 calculate the activation values in the hidden layer (z_j) and output layer (y_k) using the binary sigmoid activation function, as shown in Equation 8. Here, x_i is the i -th input, v_{ji} and w_{kj} are the connection weights between the input–hidden and hidden–output layers, respectively, while v_{j0} and w_{k0} are the bias values. Equation 8 shows the sigmoid activation function that converts input values into a range between 0 and 1, thereby producing continuous outputs that can be used in nonlinear learning processes.

$$f(x) = \frac{1}{1+e^{-x}} \quad (8)$$

(2) Backward Propagation Stage

This stage calculates the difference between the actual output value and the desired target to determine the amount of error that will be used in weight updating. Equations 9 and 10 show the process of calculating the error factor in the output layer and hidden layer:

$$\delta_k = (t_k - y_k)y_k(1-y_k) \quad (9)$$

$$\delta_j = z_j(1-z_j)\sum_{k=1}^{m}\delta_k w_{kj} \quad (10)$$

Equation 9 calculates the error factor (δ_k) in the output neuron determined by the error between the target value (t_k) and the network output (y_k), while Equation 10 describes the backward propagation of the error to the hidden layer to calculate the contribution of each neuron to the total error.

(3) Weight and Bias Update Stage

Based on the error values obtained, the weights and biases are updated iteratively to minimize the loss function. The update process follows the gradient descent rule, as shown in Equations 11 and 12:

$$\Delta w_{kj} = \alpha\delta_k z_j, \Delta v_{ji} = \alpha\delta_j x_i \quad (11)$$

$$w_{kj}^{new} = w_{kj}^{old} + \Delta w_{kj}, v_{ji}^{new} = v_{ji}^{old} + \Delta v_{ji} \quad (12)$$

Equations 11 and 12 show the process of adjusting weights and biases using the learning rate α. The value of α controls the size of the update step so that the network can learn stably without causing oscillations or slow convergence. This training process is repeated until the total error reaches a specified threshold or the maximum number of epochs is reached. To implement this optimization mechanism effectively, the network must be structured in a way that allows information to flow and be processed efficiently between its layers. Therefore, an appropriate architectural design plays a crucial role in determining how the model captures and transforms input information during the learning process.

A typical Backpropagation Neural Network (BPNN) architecture consists of three major components, namely an input layer, a hidden layer, and an output layer. The input layer functions as the medium through which external variables enter the model, while the hidden layer performs nonlinear information processing through interconnected neurons. The output layer then produces the final predicted value by applying the Sigmoid activation function [17]. Each layer is composed of several neuron units, depicted as circular nodes, and every neuron in one layer is connected to those in the subsequent layer via weighted connections, typically denoted as w. In addition, bias nodes with fixed weights may be present in the hidden layer to improve learning flexibility. The final result produced by the output layer represents the overall output of the neural network model [16], as illustrated in Figure 2.

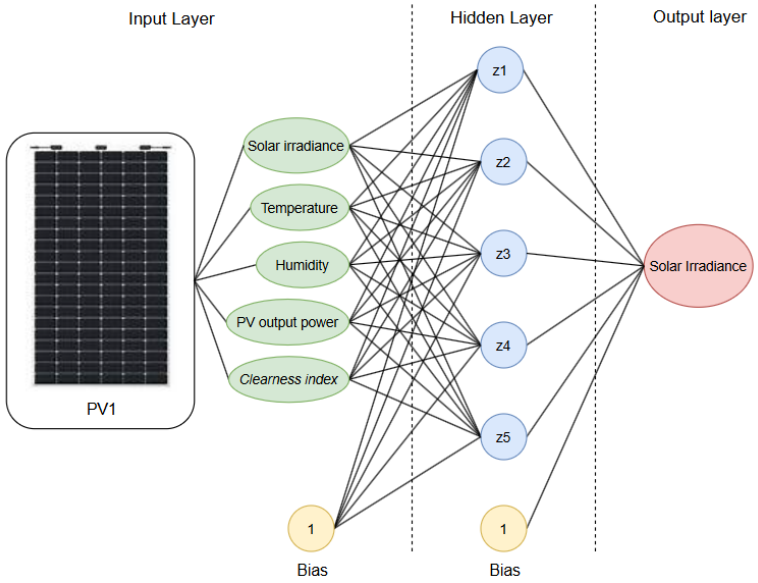

Figure 2. BPNN Architecture

Figure 2 shows an example of the Backpropagation Neural Network (BPNN) model architecture used in this study. The network model consists of one input layer, one hidden layer, and one output layer with a 5-5-1 architecture. The input layer consists of five input variables, namely:
X_1= *Solar irradiance*,
X_2= *Temperature*,
X_3= *Humidity*,
X_4= *PV output power*, dan
X_5= *Clearness index*.

The hidden layer consists of five neurons (z_1, z_2, z_3, z_4, z_5), each of which is connected to all neurons in the input layer through connection weights v_{ji}. Each neuron in the hidden layer and output layer also has a bias unit that functions to adjust the activation threshold value. The output layer has only one neuron that produces an output in the form of a predicted solar irradiance value.

In the network formation process, initial weights are randomly initialized for each inter-neuron connection, both between the input layer and the hidden layer (v_{ji}) and between the hidden layer and the output layer (w_{ji}). These weight values will be adjusted during the training process using the backpropagation algorithm until the model reaches the minimum error rate.

2.4 Model Hyperparameter

Hyperparameters are important variables that control the behavior and dynamics of an algorithm's learning process, and must be determined before the model training process begins [18]. Hyperparameter optimization is crucial for maximizing the performance of Backpropagation Neural Networks (BPNN) in short-term solar radiation forecasting. These parameters include several key components, such as training functions, performance metrics, learning rates, minimum gradients, and transfer functions.

In this study, the selection of hyperparameters—which includes the training function, learning rate, transfer function, performance metric, and minimum gradient—refers to the default configuration provided by MATLAB's Neural Network Toolbox. The training function and performance metric play an important role in determining the direction and goal of optimization during the model learning process. In the MATLAB default settings, the Levenberg–Marquardt training function (*trainlm*) and Mean Squared Error (MSE) performance metric are used, which are widely adopted in various function approximation studies, including solar radiation forecasting, due to their stability and convergence efficiency [19]. The complete hyperparameter configuration used in this study is presented in Table 2.

Table 2. Hyperparameter Model

Parameters	Value / Description
Network Type	Feedforwardbackprop
Training function	Trainlm
Performance function	MSE

Learning rate	0.001
Minimal gradient	1e-05
Transfer function hidden	LOGSIG
Transfer function output	PURELINE

The logsig (*log-sigmoid*) function is applied to the hidden layer, as it is well-suited for input data normalized within the (0-1) range and capable of identifying and approximating nonlinear interactions among variables [20]. For the output layer, the *purelin* (pure linear) function is used because the target variable—solar radiation—is continuous rather than categorical [21]. The Levenberg-Marquardt algorithm (*trainlm*) is selected as the training method due to its fast convergence properties, particularly effective for medium-sized neural networks.

To assess the impact of training duration on model convergence and generalization performance, this study explores three different epoch settings: 500, 1000, and 2000 epochs for each tested network architecture. This approach also helps in identifying the appropriate training depth to minimize the risk of underfitting or overfitting. By combining architecture testing with epoch variations, the optimal BPNN configuration can be systematically determined to achieve effective and reliable short-term solar radiation predictions.

2.5 Model Training

The model training process in this study was conducted using a trial-and-error approach to find the most optimal network architecture configuration, taking into account the number of neurons in the hidden layer. In this study, the number of neurons in the input layer was set at 15 neurons, which was obtained from the application of the sliding window technique with three backward steps on five main input variables. The output layer consisted of one neuron representing the global irradiance value one hour ahead. The network architecture was tested with various combinations of the number of neurons in the hidden layer, as shown in Table 3.

Table 3. Configuring Multiple Network Architectures

Number	Input layer and neurons	Hidden layer and neurons	Output layer and neuron
1	15	15	1
2	15	20	1
3	15	25	1
4	15	40	1
5	15	20-10	1
6	15	30-15	1
7	15	15-5	1
8	15	30-15-5	1
9	15	15-30-15	1
10	15	60-30-15	1
11	15	15-10-5-3	1
12	15	25-15-10-5	1
13	15	15-30-15-5	1

As stated by Uzair and Jamil [22], a good network topology cannot be determined solely by the number of input and output neurons. In fact, selecting the appropriate number of hidden layers can significantly reduce training time while maintaining high accuracy. Although various heuristics have been proposed to estimate the optimal number of hidden layers, the most effective configuration typically depends on the specific characteristics of the dataset. Using too many hidden layers can lead to increased training time without substantial performance improvements. Therefore, when accuracy is the primary goal, deeper networks may be beneficial, but when computational efficiency is a concern, simpler architectures are preferred.

Additionally, unnecessarily increasing the number of neurons or hidden layers can lead to overfitting, where the model performs well on training data but poorly on unseen data. According to Deng [23], the accuracy of neural networks tends to increase with the addition of more neurons to the hidden layer; however, after reaching a certain threshold, additional neurons result in a decrease in accuracy. In other words, there is a saturation point where further increases in complexity no longer significantly improve model performance.

2.6 Model Evaluation

To evaluate the performance of the Backpropagation Artificial Neural Network (BPNN) model in predicting very short-term solar radiation, a series of quantitative metrics were used, namely Mean Absolute Percentage Error (MAPE),

Mean Square Error (MSE), Root Mean Square Error (RMSE), and coefficient of determination (R^2) [24],[25]. These metrics aim to measure how well the model can accurately represent actual data. The accuracy of the prediction can be seen from the error values produced. The difference between the actual value and the predicted result is referred to as the residual, as shown in Equations 13, 14, 15, and 16.

The evaluation focuses on the test data to capture the model's ability to generalize beyond the data on which it was trained. However, the prediction results at 12:02 and 13:00 are not included in the evaluation process to prevent the performance assessment from being distorted by the presence of dummy inputs. The evaluation is only conducted up to 12:00, which is the last time interval with actual data available.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2} \tag{13}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{|yi - \hat{y}i|}{yi}\right| \times 100\% \tag{14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2 \tag{15}$$

$$R\ Square = 1 - \frac{\sum_{i=1}^{n}(yi - \hat{y}i)^2}{\sum_{i=1}^{n}(yi - \bar{y}i)^2} \tag{16}$$

Within these equations, $y_i$refers to the true value recorded at time step i, $\hat{y}_i$is the estimated output generated by the model, and $\bar{y}_i$stands for the average of the observed data. The parameter $n$indicates the total number of observations considered in the evaluation. These metrics serve to quantify both the absolute and proportional deviations of the model predictions, offering an overall measure of how well the model reproduces the real data pattern. The next section elaborates on the results and performance comparison using these metrics.

3. Results and Discussion

The following section presents the performance evaluation of the proposed Backpropagation Neural Network (BPNN) model applied to short-term forecasting of global solar irradiance in photovoltaic energy systems. The testing phase involved training and evaluating the model under different configurations, particularly by adjusting the quantity of neurons in the hidden layers and altering the number of learning iterations. The purpose of this testing was to determine the best architecture configuration that produces the most accurate prediction performance based on actual data. The testing results show significant differences in model performance based on variations in the number of neurons and training epochs, highlighting the sensitivity of the BPNN model to hyperparameter selection and its direct impact on forecasting accuracy and convergence stability.

3.1 Comparative Performance of BPNN Configurations

The performance of the Backpropagation Artificial Neural Network (BPNN) model on various architectural configurations is summarized in Tables 4, 5, and 6, which present the prediction accuracy for each configuration using training epochs of 500, 1000, and 2000, respectively, as assessed by four metrics: Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and coefficient of determination (R^2). These evaluation metrics were selected to comprehensively assess the model's prediction capability from different perspectives—accuracy, error magnitude, and correlation between predicted and actual values. The comparison among different epoch settings also provides insight into the model's convergence behavior, helping to identify the optimal training duration that balances computational efficiency and predictive accuracy.

Table 4. Performance Comparison of BPNN Models with Different Architectures Using 500 Epochs

Network architecture configuration	EPOCH 500			
	MAPE	MSE	RMSE	R SQUARE
15-15-1	12,80%	0,00191	0,33	0,961237
15-20-1	12,60%	0,00188	0,33	0,962174
15-25-1	16,50%	0,00215	0,34	0,950127
15-40-1	5,90%	0,00168	0,25	0,972702

15-20-10-1	22,60%	0,00285	0,38	0,945521
15-30-15-1	15,90%	0,00205	0,35	0,956882
15-15-5-1	8,80%	0,00174	0,29	0,964863
15-30-15-5-1	11,70%	0,00179	0,3	0,963352
15-15-30-15-1	14,40%	0,00199	0,34	0,960437
15-60-30-15-1	12,30%	0,00186	0,32	0,965045
15-15-10-5-3-1	18,60%	0,00225	0,36	0,952645
15-25-15-10-5-1	14,50%	0,002	0,35	0,958237
15-15-30-15-5-1	11,20%	0,00177	0,3	0,963774

Table 5. Performance of BPNN Models with Various Network Architectures Using 1000 Epochs

Network architecture configuration	EPOCH 1000			
	MAPE	MSE	RMSE	R SQUARE
15-15-1	13,10%	0,00159	0,097	0,964583
15-20-1	10%	0,00123	0,092	0,969163
15-25-1	10,90%	0,00129	0,093	0,971002
15-40-1	9,40%	0,00114	0,091	0,973057
15-20-10-1	8,40%	0,0006	0,09	0,973488
15-30-15-1	12,30%	0,00152	0,096	0,964451
15-15-5-1	10%	0,00125	0,092	0,967837
15-30-15-5-1	13,10%	0,0016	0,097	0,962011
15-15-30-15-1	13,60%	0,00166	0,098	0,961027
15-60-30-15-1	9,80%	0,00118	0,091	0,970437
15-15-10-5-3-1	18,60%	0,0023	0,105	0,951943
15-25-15-10-5-1	8,70%	0,00091	0,091	0,974137
15-15-30-15-5-1	19,30%	0,00247	0,106	0,950063

Table 6. Performance of BPNN Models with Various Network Architectures Using 2000 Epochs

Network architecture configuration	EPOCH 2000			
	MAPE	MSE	RMSE	R SQUARE
15-15-1	15%	0,00174	0,095	0,962319
15-20-1	19%	0,00239	0,11	0,955829
15-25-1	5,89%	0,00027	0,04	0,981874
15-40-1	11,50%	0,00135	0,075	0,968345
15-20-10-1	10,50%	0,00126	0,071	0,969437
15-30-15-1	17,70%	0,00221	0,106	0,949932
15-15-5-1	9,10%	0,00112	0,067	0,971128
15-30-15-5-1	14,60%	0,00169	0,09	0,963523
15-15-30-15-1	12%	0,00141	0,078	0,965618
15-60-30-15-1	11,20%	0,00131	0,074	0,967018
15-15-10-5-3-1	14,4	0,00166	0,089	0,966873
15-25-15-10-5-1	6,40%	0,00071	0,055	0,976045
15-15-30-15-5-1	12,30%	0,00145	0,079	0,965919

At all epoch levels, network architectures with deeper or wider configurations do not always produce better performance, highlighting the importance of balancing model complexity and generalization. At the 500-epoch training setting (Table 4), the architecture with 15 input neurons, one hidden layer consisting of 40 neurons, and a single unit in the output layer produced the best results with a MAPE of 5.90% and an R^2 value of 0.9727, indicating that this architecture is more effective at capturing the underlying patterns in the data compared to other configurations.

By increasing the training duration to 1000 epochs (Table 5), the architecture with 15 input neurons, two hidden layers with 20-10 neurons, and one output neuron outperformed other architectures with a MAPE of 8.40% and an R^2 of 0.9735, indicating improved generalization over time. Interestingly, while error rates decreased for some models, overfitting began to emerge in deeper networks such as the 15 input neurons, four hidden layer architecture with 15-30-15-5 neurons and a single unit in the output layer, with MAPE exceeding 19%.

The 2000 epoch configuration (Table 6) yields further improvements for certain architectures. Specifically, the architecture with 15 input neurons, one hidden layer with 25 neurons, and a single unit in the output layer achieves the lowest MAPE of 5.89% and the highest R^2 of 0.9819, outperforming even deeper configurations. However, some deeper

architectures, such as the 15 input neurons with two hidden layers (30-15 neurons) and a single unit in the output layer, showed a decline in performance, indicating that longer training without careful regularization can lead to diminishing returns or overfitting.

These results indicate that while increasing the number of epochs generally improves model accuracy, optimal performance does not solely depend on deeper networks but rather on a balanced architecture with appropriate training duration. The combination of 15 input neurons, one hidden layer with 25 neurons, and 2000 training epochs proved to be the most effective configuration in this study for very short-term solar radiation prediction.

3.2 Global Solar irradiance Forecasting Results

Global solar radiation forecasts are generated using the BPNN architecture with the best performance based on quantitative evaluation metrics. The network architecture configuration with 15 input neurons, one hidden layer with 25 neurons, and a single unit in the output layer, trained for 2000 epochs, was identified as the most accurate and stable model across all MAPE, MSE, RMSE, and R^2 indicators. To complement the performance evaluation, a visual analysis was conducted to test how closely the best-performing BPNN model could replicate the actual solar radiation data patterns. This visualization helps highlight the model's ability to follow temporal trends, capture fluctuations, and maintain prediction stability across the observed timeframe.

Figure 3 illustrates the comparison between actual solar radiation values and the predictions generated by the BPNN model with 15 input neurons representing input variables resulting from data transformation using the sliding window method, one hidden layer consisting of 25 neurons, and a single unit in the output layer, trained over 2000 epochs. This model was selected based on its superior performance across all evaluation metrics, as presented in Table 6. The predicted curve closely aligns with the actual data, demonstrating the model's ability to capture peak variations and slopes, which are crucial for real-time solar forecasting. The visualization is based on the test dataset from June 17, 2025, which was not used during the training process. By applying the model to this unseen dataset, the generalization capability of the BPNN is effectively evaluated. The comparison with the actual test data provides a reliable benchmark to assess how well the model performs in real-world conditions.

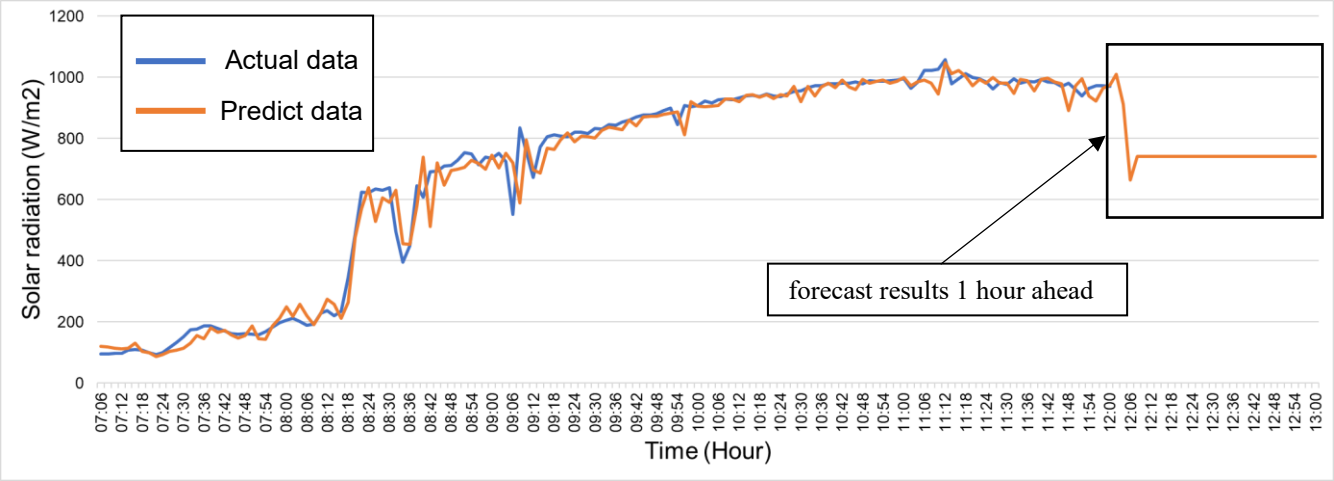

Figure 3. Visualization of Actual vs. Predicted Solar Irradiance and Extended One-hour Ahead Forecast Using the Optimal BPNN Model

In addition to predictions within the observation period (up to 12:00), Figure 3 also presents short-term forecast results for the next hour (from 12:02 to 13:00). Although this future interval is supported by dummy input (simulated data), the model is able to generate a smooth and reasonable continuation of the solar radiation trend, consistent with the last known actual data point. These results confirm that the BPNN configuration performed best in this study for short-term solar radiation forecasting. The balance between network complexity and training time proved to be crucial in achieving optimal prediction accuracy, both numerically and visually.

4. Conclusion

The Backpropagation Neural Network (BPNN) model developed in this study has proven to be effective for short-term forecasting of global solar radiation in photovoltaic (PV)-based solar power generation systems. Through a series of tests on 13 different network configurations and three variations in the number of training epochs (500, 1000, and 2000), the optimal configuration was obtained with 15 input neurons representing input variables resulting from data transformation using the sliding window method, one hidden layer consisting of 25 neurons, and a single unit in the

output layer, trained for 2000 epochs. This configuration consistently produced the best performance across all evaluation metrics. Test results showed that this configuration yielded the lowest error values, with a MAPE of 5.89%, MSE of 0.00027, RMSE of 0.04, and a coefficient of determination (R^2) of 0.981874. These findings indicate that a relatively simple network architecture, when optimized through training, can produce highly accurate prediction results. The model successfully captures temporal patterns and fluctuations in solar radiation, as evidenced by numerical evaluations and visualizations that align with actual data on the test date of June 17, 2025.

Additionally, visual analysis confirms the model's ability to generate smooth and realistic predictions for future time intervals, even when using synthetic (simulated) input data. Although this model is not designed for real-time operation, it functions as a short-term forecasting system that uses historical data to estimate radiation one hour ahead, similar to a weather prediction model. In practical applications, forecast results can be integrated into PV system management processes—such as energy scheduling, load planning, or monitoring dashboards—to support decision-making and improve overall system efficiency.

References

[1] S. Garg, A. Agrawal, S. Goyal, and K. Verma, "Day Ahead Solar Irradiance Forecasting using Different Statistical Techniques," in *2020 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, IEEE, Dec. 2020, pp. 1–4. https://doi.org/10.1109/PEDES49360.2020.9379907.

[2] V. A. G. Raju, J. Nayak, P. B. Dash, and M. Mishra, "Short-Term Solar Irradiance Forecasting Model Based on Hyper-parameter Tuned LSTM via Chaotic Particle Swarm Optimization Algorithm," *Case Studies in Thermal Engineering*, p. 105999, Mar. 2025, https://doi.org/10.1016/j.csite.2025.105999.

[3] B. H. Mahdi, K. M. Yousif, and L. MS. Dosky, "Using Artificial Neural Networks to Predict Solar Radiation for Duhok City, Iraq," in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, IEEE, Apr. 2020, pp. 1–6. https://doi.org/10.1109/CSASE48920.2020.9142119.

[4] I. M. Galván, J. Huertas-Tato, F. J. Rodríguez-Benítez, C. Arbizu-Barrena, D. Pozo-Vázquez, and R. Aler, "Evolutionary-based prediction interval estimation by blending solar radiation forecasting models using meteorological weather types," *Appl Soft Comput*, vol. 109, Sep. 2021, https://doi.org/10.1016/j.asoc.2021.107531.

[5] R. J. Jacques Molu *et al.*, "Advancing short-term solar irradiance forecasting accuracy through a hybrid deep learning approach with Bayesian optimization," *Results in Engineering*, vol. 23, p. 102461, Sep. 2024, https://doi.org/10.1016/j.rineng.2024.102461.

[6] Y. Lin, D. Duan, X. Hong, X. Cheng, L. Yang, and S. Cui, "Very-Short-Term Solar Forecasting with Long Short-Term Memory (LSTM) Network," in *2020 Asia Energy and Electrical Engineering Symposium (AEEES)*, IEEE, May 2020, pp. 963–967. https://doi.org/10.1109/AEEES48850.2020.9121512.

[7] S. Gupta, A. R. Katta, Y. Baldaniya, and R. Kumar, "Hybrid Random Forest and Particle Swarm Optimization Algorithm for Solar Radiation Prediction," in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, IEEE, Oct. 2020, pp. 302–307. https://doi.org/10.1109/ICCCA49541.2020.9250715.

[8] P. P. Simanjuntak and K. P. Wibowo, "Estimasi Intensitas Radiasi Matahari Dengan Menggunakan Jaringan Syaraf Tiruan Backprpagation Di Kota Jayapura," *Jurnal Fisika : Fisika Sains dan Aplikasinya*, vol. 8, no. 1, pp. 44–49, Apr. 2023, https://doi.org/10.35508/fisa.v8i1.11823.

[9] C.-J. Huang, Y. Ma, and Y.-H. Chen, "Solar Radiation Forecasting based on Neural Network in Guangzhou," in *2020 International Automatic Control Conference (CACS)*, IEEE, Nov. 2020, pp. 1–5. https://doi.org/10.1109/CACS50047.2020.9289830.

[10] O. O. Soneye, "Evaluation of clearness index and cloudiness index using measured global solar radiation data: A case study for a tropical climatic region of Nigeria," *Atmosfera*, vol. 34, no. 1, pp. 25–39, 2021, https://doi.org/10.20937/ATM.52796 .

[11] O. O. Apeh, O. K. Overen, and E. L. Meyer, "Monthly, seasonal and yearly assessments of global solar radiation, clearness index and diffuse fractions in alice, south africa," *Sustainability (Switzerland)*, vol. 13, no. 4, pp. 1–15, Feb. 2021, https://doi.org/10.3390/su13042135.

[12] A. E. Radho, P. Sugiartawan, and G. A. Santiago, "Prediksi Jumlah Kasus COVID-19 Menggunakan Teknik Sliding Wondows dengan Metode BPNN," *Jurnal Sistem Informasi dan Komputer Terapan Indonesia (JSIKTI)*, vol. 4, no. 1, pp. 12–23, Sep. 2021, https://doi.org/10.22146/jsikti.xxxx.

[13] R. Aprianto, S. Fitriyanto, S. N. Walidain, and Hermansyah, "Artificial Neural Network Backpropagation for Predicting Rainfall (Case Study in Sultan Muhammad Kaharuddin Meteorological Station)," *Titian Ilmu: Jurnal Ilmiah Multi Sciences*, vol. 15, no. 1, pp. 63–70, Jan. 2023, https://doi.org/10.30599/jti.v15i1.2110.

[14] M. F. Mubarokh, M. Nasir, and D. Komalasari, "Jaringan Syaraf Tiruan Untuk Memprediksi Penjualan Pakaian Menggunakan Algoritma Backpropagation," *Journal of Computer and Information Systems Ampera*, vol. 1, no. 1, pp. 2775–2496, Jan. 2020, https://doi.org/10.51519/journalcisa.v1i1.3.

[15] K. Shihab, "A Backpropagation Neural Network for Computer Network Security," *Journal of Computer Science*, vol. 2, no. 9, pp. 710–715, 2006.

[16] A. P. Iskandar, "Efektifitas Jaringan Syaraf Tiruan Metode Backpropagation dalam Memprediksi Potensi Banjir (Studi Kasus : Kecamatan Sungai Serut Bengkulu)," *JTIS*, vol. 3, no. 2, pp. 79–85, Jul. 2020, https://doi.org/10.36085/jtis.v3i2.159.

[17] D. Kuncoro, A. M. H. Pardede, and S. Syahputra, "Jaringan Syaraf Tiruan Memprediksi Penyakit Gerd menggunakan Metode Backpropagation," *Bridge : Jurnal publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 4, pp. 269–287, Sep. 2024, https://doi.org/10.62951/bridge.v2i4.251.

[18] G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, and H. Samulowitz, "An effective algorithm for hyperparameter optimization of neural networks," *IBM J Res Dev*, vol. 61, no. 4, Jul. 2017, https://doi.org/10.1147/JRD.2017.2709578 .

[19] M. T. Hagan and M. B. Menhaj, "Brief Papers Training Feedforward Networks with the Marquardt Algorithm," 1994.

[20] I. Pamungkas, Sumadi, and S. Alam, "Studi Komparasi Fungsi Aktivasi Sigmoid Biner, Sigmoid Bipolar dan Linear pada Jaringan Saraf Tiruan dalam Menentukan Warna RGB Menggunakan Matlab," *Serambi Engineering*, vol. VII, no. 4, pp. 3749–3758, Oct. 2022, Accessed: Mar. 25, 2025.

[21] S. Sharma, S. Sharma, and A. Athaiya, "Activation Functions In Neural Networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 12, pp. 310–316, Apr. 2020, https://doi.org/DOI:10.33564/ijeast.2020.v04i12.054.

[22] M. Uzair and N. Jamil, "Effects of Hidden Layers on the Efficiency of Neural networks," in *Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. https://doi.org/10.1109/INMIC50486.2020.9318195.

[23] T. Deng, "Effect of the Number of Hidden Layer Neurons on the Accuracy of the Back Propagation Neural Network," *Highlights in Science, Engineering and Technology MISBP*, vol. 74, 2023. https://doi.org/doi.org/10.54097/nbra6h45

[24] D. Ardiansyah and D. N. Astuti, “Perbandingan Model Prediksi Radiasi Matahari Berbasis Mesin Pembelajaran Pada Stasiun Meterorologi Fatmawati Soekarno Bengkulu,” *Megasains*, vol. 14, no. 1, pp. 26–32, Sep. 2023, https://doi.org/10.46824/megasains.v14i1.129.
[25] Moch. N. Adiwana, “Desain Photovoltaic Dan Peramalan Jangka Pendek Radiasi Sinar Matahari Menggunakan Metode Feed-Forward Neural Network,” *Jurnal Teknik Elektro*, vol. 9, no. 1, pp. 757–764, Jan. 2020.