



Comparative Analysis of Forecasting Models for Hourly Solar Irradiance in Southern Thailand

Mojtaba Safari¹, Suttipan Aksornniem¹, Tanawat Srirugsa^{1,2}, Wiwat Su-hren^{1,2}
and Supachai Kaewpoung^{1,2*}

¹ Faculty of Engineering, Thaksin University, Phatthalung 93210, Thailand

² Center of Excellence for Agricultural Innovation and Bioproducts of Thaksin University, Thaksin University, Phatthalung, 93210, Thailand

* Correspondence: supachai.ka@tsu.ac.th

Citation:

Safari, M., Aksornniem, S., Srirugsa, T., Su-hren, W., Kaewpoung, S., Comparative analysis of forecasting Models for hourly solar irradiance in southern Thailand. *ASEAN J. Sci. Tech. Report.* **2026**, 29(7), e262134. <https://doi.org/10.55164/ajstr.v29i7.262134>

Article history:

Received: October 30, 2025

Revised: April 11, 2026

Accepted: April 12, 2026

Available online: June 14, 2026

Publisher's Note:

This article is published and distributed under the terms of the Thaksin University.

Abstract: Accurate forecasting of short-term solar irradiance is essential to optimize energy supply, improve grid stability, and increase the economic viability of photovoltaic systems. This study compares several forecasting models for predicting hourly solar irradiance using real-world data from Southern Thailand. Eight models were evaluated, including six forecasting models (Prophet, LSTM, GRU, XGBoost, Random Forest, and SARIMA with walk-forward validation) and two benchmarks (Naive and Seasonal Naive), using RMSE, MAE, and MAPE metrics. The results demonstrate that ensemble-based machine learning models significantly outperformed traditional statistical and deep learning approaches. Specifically, XGBoost achieved the best performance with RMSE of 79.43 W/m², MAE of 59.63 W/m², and MAPE of 30.07%, followed closely by Random Forest with RMSE of 83.00 W/m² and MAPE of 29.36%. Compared to the best benchmark model, XGBoost reduced RMSE by 50.3% and MAPE by 47.1%. The GRU model outperformed LSTM, demonstrating greater efficiency in capturing short-term temporal dependencies. The findings confirm that ensemble methods effectively capture complex temporal patterns and nonlinear meteorological dependencies, making them highly suitable for real-time solar forecasting applications in tropical climates.

Keywords: Solar irradiance forecasting; southern Thailand; machine learning; deep learning

1. Introduction

Currently, solar energy is a critical renewable energy source driving environmental and economic progress worldwide. As global energy demand continues to rise, transitioning to solar energy has become essential for reducing greenhouse gas emissions, advancing sustainable development, and decreasing reliance on fossil fuels [1]. However, solar power generation is inherently variable and intermittent, depending on weather conditions, time of day, and seasonal factors. These fluctuations pose significant challenges for power generation and distribution management, potentially affecting electrical grid stability [2–3]. Consequently, developing accurate forecasting models for solar irradiance has become crucial for optimizing energy generation, storage, and management in modern renewable energy systems [4–5].

Accurate short-term solar irradiance forecasting, particularly at hourly resolution, is essential for real-time control of photovoltaic (PV) systems and for optimizing energy storage devices, such as batteries, and load control mechanisms [6]. Statistical models such as ARIMA and SARIMA have been

widely used in time-series analysis due to their effectiveness in capturing seasonal patterns and linear trends [7–8]. However, these models often struggle to capture nonlinear relationships between climatic variables [9]. The rapid development of machine learning (ML) and deep learning (DL) techniques has introduced new approaches for time-series forecasting [10–11]. Ensemble methods such as Random Forest (RF) and XGBoost have gained popularity due to their ability to handle large datasets and model complex interactions [12–13].

Deep neural network architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have demonstrated strong capabilities for retaining long-term temporal information and learning sequential dependencies, making them particularly suitable for analyzing complex, time-varying solar irradiance patterns [14–17]. Additionally, Facebook's Prophet model was created especially to handle data with changing temporal trends and varied seasonalities. For extremely variable environmental and energy datasets, it has shown especially favorable results [18–19]. Consequently, evaluating the effectiveness of statistical, machine learning, and deep learning models offers a crucial foundation for determining the best method for forecasting solar irradiance, especially in tropical areas like Thailand [20–21].

This study aims to develop and evaluate hourly solar irradiance forecasting models using real-world data from a solar monitoring station at the Faculty of Engineering, Thaksin University, Southern Thailand. The dataset includes solar irradiance and key meteorological variables that influence solar radiation on PV surfaces: ambient temperature, panel temperature, relative humidity, wind speed, and wind direction [22–23]. The proposed framework encompasses baseline models (Naive and Seasonal Naive (SNaive)) for performance benchmarking, statistical models (SARIMA and Prophet), machine learning models (XGBoost and Random Forest), and deep learning models (LSTM and GRU) [24–25].

The key contributions of this study include: (1) a comprehensive comparison of eight forecasting approaches under tropical conditions, (2) empirical evidence on the relative performance of traditional statistical methods versus modern machine learning and deep learning techniques for solar forecasting, and (3) practical guidance on computational efficiency and model selection for real-time applications. These findings are expected to support the development of intelligent solar energy management systems and enhance renewable energy integration in tropical regions with high solar potential [26].

The remainder of this paper is organized as follows: Section 2 presents the methodology, including data preprocessing, model descriptions, and evaluation metrics. Section 3 describes the exploratory data analysis, discusses the experimental setup, and compares model performance. Finally, Section 5 concludes with key findings, practical implications, and recommendations for future research.

2. Methodology

This study develops and compares multiple forecasting models to predict hourly solar irradiance. The methodology includes three types of techniques: statistical models (SARIMA and Prophet), ML models (XGBoost and RF), and DL models (LSTM and GRU). Each model is designed to capture different characteristics of solar irradiance patterns, including seasonal changes, nonlinear relationships, and the influence of weather variables. Furthermore, two simple benchmark models are used: the Naive method and the SNaive method, which provide a baseline for comparison. Finally, the performance of these models is evaluated using three key metrics: RMSE, MAE, and MAPE [27].

2.1 Pre-processing and normalization

Before building the forecasting models, the raw dataset was checked for missing or incorrect values that might have resulted from sensor failures. Missing hourly values were filled using a two-step method based on climatology patterns. In the second step, a Day-of-Year $\times$ Hour matrix was created and linearly interpolated across both days and hours to smooth any remaining gaps. This process produced a complete and physically consistent hourly dataset. All variables were then aligned to a uniform hourly timestamp format to ensure temporal consistency. After cleaning the data, all variables were normalized using Min-Max scaling, which maps values to the range [0, 1]. This normalization was applied separately to the training set (80% of the data) and the testing set (20% of the data) to prevent data leakage. The Min-Max scaling formula is:

$$Z = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

Where X is the original value of a feature, $X_{\min}$ is the minimum value of that feature in the training data, $X_{\max}$ is the maximum value of that feature in the training data, and Z is the normalized value.

2.2 Forecasting models

2.2.1 Walk-forward SARIMA model

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model extends the classical ARIMA framework by accounting for both regular patterns and seasonal patterns in time series data [28]. The general form of a SARIMA model is expressed as:

$$\text{SARIMA}(p, d, q)(P, D, Q)_S \quad (2)$$

where the parameters p , d , and q represents the non-seasonal autoregressive order, non-seasonal differencing, and non-seasonal moving average order, respectively, and the parameters P , D , Q , and S refer to the seasonal autoregressive order, seasonal differencing, seasonal moving average order, and length of the seasonal cycle, respectively. The mathematical formulation of the SARIAM model can be defined as [29]:

$$\phi_p(B)\phi_P(B^S)(1-B)^d(1-B^S)^D Y_t = \theta_q(B)\theta_Q(B^S)\varepsilon_t \quad (3)$$

where:

$$\begin{aligned} \phi_p(B) &= 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p, \\ \phi_P(B^S) &= 1 - \phi_1 B^S - \phi_2 B^{2S} - \dots - \phi_P B^{PS}, \\ \theta_q(B) &= 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q, \\ \theta_Q(B^S) &= 1 - \theta_1 B^S - \theta_2 B^{2S} - \dots - \theta_Q B^{QS}, \end{aligned}$$

where $\phi_1, \dots, \phi_p$, and $\phi_1, \dots, \phi_P$, are non-seasonal and seasonal autoregressive coefficients, while $\theta_1, \dots, \theta_q$, and $\theta_1, \dots, \theta_Q$ are the coefficients for the non-seasonal and seasonal moving-average terms. B is the backshift operator, whose effect on a time series Y_t , and ε_t is a white noise sequence.

To evaluate the SARIMA model under realistic forecasting conditions, a walk-forward validation strategy was used. This approach is widely recommended in time series forecasting literature [29-30], because it provides a more reliable assessment of how well statistical models like SARIMA generalize to unseen data. Unlike a single train-test split, walk-forward validation continuously retrains the model as new data becomes available, ensuring that all forecasts are made on truly future data. This study used a fixed-length rolling window approach for the walk-forward validation. Specifically, at each forecasting step, the model was trained on the most recent observations within a window of approximately 720 hours. After making a forecast, the window moved forward by one hour: the oldest observation was removed, and the newest observation was added. The model was retrained at every step to capture short-term patterns and adapt to local changes in solar irradiance. The mathematical representation of the walk-forward validation process is as follows:

Let $y_1, y_2, \dots, y_t$ represent the observed irradiance series of length t . At each step t (where $t \geq W$), the model is trained on a fixed window of the most recent W observations as follows:

$$D_t = y_{t-W+1}, y_{t-W+2}, \dots, y_t \quad (4)$$

Using this training window, the model estimates its parameters θ_t and produces a one-step-ahead forecast:

$$\hat{y}_{t+1} = f(D_t; \theta_t) \quad (5)$$

After generating the forecast, the window is advanced by one time step, discarding the oldest observation and adding the newest actual value y_{t+1} :

$$D_{t+1} = y_{t-W+2}, y_{t-W+3}, \dots, y_{t+1} \quad (6)$$

This process repeats until the end of the testing period. The forecasting performance is evaluated by comparing each predicted value $\hat{y}_{t+1}$ to its corresponding actual value y_{t+1} .

2.2.2 LSTM model

The LSTM network is a specialized type of Recurrent Neural Network (RNN) designed to capture temporal patterns in sequential data. Traditional RNNs struggle to learn long-term dependencies due to vanishing or exploding gradients during training. LSTMs overcome this limitation by using memory cells and gating mechanisms that control which information to retain or discard over extended time periods. Each LSTM unit includes three gates that control information flow [31]. Which data from the prior cell state should be erased is decided by the forget gate. Which fresh data should be added to the cell state is controlled by the input gate. The output gate controls the amount of cell-state data sent to the next time step. These gates work together to selectively update and propagate information through the network, enabling the model to learn both short-term and long-term dependencies in the data. The operations within an LSTM unit at time step t are mathematically expressed as follows:

The forget gate f_t determines what information to discard from the previous cell state:

$$f_t = \sigma(W_{hf}h_{t-1} + W_{xf}X_t + b_{hf} + b_{xf}) \quad (7)$$

The input gate i_t decides which new information should be added to the cell state:

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + b_i) \quad (8)$$

The candidate cell state $\tilde{c}_t$ represents new information that could be added to the cell:

$$\tilde{c}_t = \tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \quad (9)$$

The cell state c_t is updated by combining the forgotten information and new information:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (10)$$

The output gate o_t controls what information from the cell state to output:

$$o_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + b_o) \quad (11)$$

The hidden state (h_t output of the LSTM unit) is computed as:

$$h_t = o_t \odot \tanh(c_t) \quad (12)$$

where X_t represents the input vector at the time step t , h_{t-1} is the previous hidden state, c_{t-1} refers to the previous cell state, $\tanh$ is the hyperbolic tangent activation function, W_{xf} , W_{hf} , W_{xi} , W_{hi} , W_{xc} , W_{hc} , W_{xo} , W_{ho} are weights for forget gate, input gate, candidate cell state, and output gate, respectively, and b_{xf} , b_i , b_c , b_o are bias values for forget gate, input gate, candidate cell state, and output gate, respectively. The symbol $\odot$ denotes element-wise multiplication.

2.2.3 GRU model

The Gated Recurrent Unit (GRU) is a simplified variant of the LSTM network, also designed to capture temporal dependencies in sequential data while addressing the vanishing gradient problem. GRUs achieve comparable performance to LSTMs but with a more simplified architecture that uses fewer parameters, making them computationally more efficient and faster to train [31]. Unlike LSTMs, which use separate memory cells and hidden states, GRUs combine these into a single hidden state. Each GRU unit contains two gates that control information flow:

- Reset gate: Determines how much of the previous hidden state should be discarded,
- Update gate: Controls how much of the previous hidden state to retain and how much new information to add.

These two gates enable the GRU to selectively retain relevant information from previous time steps while incorporating new inputs. The operations within a GRU unit at time step t are mathematically expressed as follows:

The reset gate r_t determines how much past information to forget:

$$r_t = \sigma(W_{xr}X_t + W_{hr}h_{t-1} + b_r) \quad (13)$$

The update gate z_t controls the balance between retaining old information and adding new information:

$$z_t = \sigma(W_{xz}X_t + W_{hz}h_{t-1} + b_z) \quad (14)$$

The candidate's hidden state $\tilde{h}_t$ represents new information based on the current input and resets the past state:

$$\tilde{h}_t = \tanh(W_{xh}X_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \quad (15)$$

The hidden state h_t is updated by combining the previous hidden state and the candidate hidden state:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (16)$$

where X_t is the input vector at time step t , h_{t-1} is the previous hidden state, σ is the sigmoid activation function, $\tanh$ is the hyperbolic tangent activation function, W_{xr} , W_{hr} , W_{xz} , W_{hz} , W_{xh} , W_{hh} are weight matrices for the reset gate, update gate, and candidate hidden state, respectively, and b_r , b_z , b_h are bias terms for the reset gate, update gate, and candidate hidden state, respectively.

2.2.4 Prophet model

Prophet is a time-series forecasting model developed by Facebook designed to handle data with strong seasonal patterns and multiple seasonality components [32]. Prophet uses an additive decomposition framework that automatically detects and models trends, seasonality, and holiday effects, making it particularly suitable for business and environmental forecasting applications. The Prophet model decomposes a time series into three main components: trend, seasonality, and holidays. The general formulation of the Prophet model is expressed as:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (17)$$

where $y(t)$ is the forecasted value at time t , $g(t)$ represents the trend component that captures long-term changes in the data, $s(t)$ represents the seasonal component that captures periodic patterns (daily, weekly, or yearly), $h(t)$ represents the holiday or special event effects, and ε_t is the error term representing random noise. The trend component $g(t)$ models non-periodic changes in the time series. Prophet uses a piecewise linear or logistic growth model to capture trends that may change over time:

$$g(t) = (k + a(t)^T \delta) \cdot t + (m + a(t)^T \gamma) \quad (18)$$

where k is the growth rate, δ contains rate adjustments at changepoints, m is the offset parameter, and $a(t)$ is an indicator function for the changepoint. The seasonality component $s(t)$ captures periodic patterns using the Fourier series:

$$s(t) = \sum_{n=1}^N (a_n \cos(\frac{2\pi nt}{P}) + b_n \sin(\frac{2\pi nt}{P})) \quad (19)$$

where P is the period, N is the number of Fourier terms, and a_n and b_n are the Fourier coefficients estimated from the data.

2.2.5 Random Forest

An ensemble machine learning approach called Random Forest (RF) combines multiple decision trees to produce more reliable and accurate predictions. The model averages predictions from numerous trees, each trained on a random subset of the data and features, to address the overfitting issue sometimes encountered with single decision trees. The RF algorithm operates through the following process [33]: (1) Bootstrap sampling creates multiple subsets of the training data through random sampling with replacement; (2) Tree construction builds a decision tree for each subset, where at each node, only a random subset of features is considered for splitting; and (3) Aggregation produces the final prediction by averaging the outputs of all individual trees for regression tasks. The prediction of a Random Forest model for a given input X is mathematically expressed as:

$$\hat{y} = \frac{1}{n_{trees}} \sum_{i=1}^{n_{trees}} T_i(X) \quad (20)$$

where $\hat{y}$ is the predicted value, n_{trees} is the total number of decision trees in the forest, and $T_i(X)$ is the prediction from the i -th decision tree. Each decision tree in the forest learns different patterns from the data due to the randomness in both data sampling and feature selection. This diversity among trees reduces the model's variance and improves its generalization ability. The RF model includes several key hyperparameters, including the number of trees (n_{trees}), the maximum depth of each tree (max_{depth}), and the minimum number of samples required to split a node ($min_{samples}$).

2.2.6 XGBoost model

XGBoost (Extreme Gradient Boosting) is an advanced ensemble machine learning algorithm that builds a series of decision trees sequentially, where each new tree attempts to correct the errors made by the previous trees. Unlike RF, which builds trees in parallel, XGBoost uses a boosting approach in which trees are added one at a time, with each tree learning from the mistakes of its predecessors. The XGBoost model optimizes a regularized objective function that balances prediction accuracy and model complexity [34]:

$$L = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (21)$$

where $l(y_i, \hat{y}_i)$ is the loss function measuring the difference between the actual value y_i and the predicted value $\hat{y}_i$, $\Omega(f_k)$ is the regularization term that penalizes model complexity to prevent overfitting, n is the number of samples, and K is the number of trees. The final prediction is obtained by summing the predictions from all trees:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad (22)$$

where $\hat{y}_i$ is the predicted value for the sample i , f_k represents the k -th tree, and x_i is the input feature vector. XGBoost employs gradient descent to iteratively minimize the loss function. At each iteration, a new tree is added that approximates the negative gradient of the loss function, effectively moving the predictions closer to the true values. The algorithm also incorporates several advanced techniques, including tree pruning, handling of missing values, and parallel processing, which contribute to its computational efficiency and superior performance. The key hyperparameters of the XGBoost model include the number of trees ($n_{estimators}$), the learning rate (η) that controls the contribution of each tree, the maximum tree depth (max_{depth}), the minimum loss reduction required to make a split (γ), and regularization parameters (λ and α) that control model complexity.

2.2.7 Benchmark models

To evaluate the performance of the forecasting models, two simple benchmark methods were implemented: the Naive model and the Seasonal Naive (SNaive) model. These benchmarks provide baseline comparisons and help assess whether the more complex models offer meaningful improvements over straightforward forecasting approaches. These two benchmark models were selected because hourly solar irradiance typically exhibits both short-term continuity and daily cyclical behavior, making persistence-based and seasonal-repeat baselines appropriate reference models for performance evaluation. The Naive model is

the simplest forecasting method, which assumes that the next value will be equal to the most recent observed value. Mathematically, the Naive forecast at time t is expressed as:

$$\hat{y}_t = y_{t-1} \quad (23)$$

where $\hat{y}_t$ is the predicted value at time t , and y_{t-1} is the actual value at the previous time step. This model is particularly useful for data without strong trends or seasonal patterns and serves as a minimum performance threshold that any sophisticated model should exceed. SNaive model accounts for seasonal patterns in the data by using the value from the same period in the previous seasonal cycle as the forecast. For hourly solar irradiance data with a daily seasonal pattern, the SNaive forecast is expressed as:

$$\hat{y}_t = y_{t-S} \quad (24)$$

where S is the seasonal period. In this study, $S = 12$ hours, corresponding to the 12 daylight hours retained after preprocessing. This means the forecast for a given hour is based on the irradiance value from the same hour on the previous day. The SNaive model is particularly effective for data with strong and stable seasonal patterns, making it a more challenging benchmark than the simple Naive model for solar irradiance forecasting. Both benchmark models require minimal computational resources and no parameter tuning. Their performance serves as a reference point to determine whether the added complexity of statistical, machine learning, and deep learning models translates into tangible forecasting improvements.

2.3 Evaluation metrics

To assess and compare the forecasting performance of all models, three widely used evaluation metrics were employed: RMSE, MAE, and MAPE.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (25)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (26)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (27)$$

where n is the number of observations, y_i is the actual value at time i , and $\hat{y}_i$ is the predicted value at time i . Lower values of all three metrics indicate better forecasting performance.

3. Experiment and numerical results

This section presents the experimental framework and the numerical results obtained from evaluating the forecasting models. All experiments were conducted using Python 3.11 on a system equipped with an Intel Core i7 processor and 16 GB of RAM. The models were trained on 80% of the dataset and tested on the remaining 20% to assess their generalization performance. Model accuracy was evaluated using three performance metrics: MAE, RMSE, and MAPE, which enable direct comparison across all models.

3.1 Exploratory data analysis

Before developing the forecasting models, an exploratory data analysis was conducted to understand the dataset's temporal and statistical characteristics. This study uses data recorded by a solar photovoltaic monitoring station at the Faculty of Engineering, Thaksin University, Thailand. The monitoring system continuously measures key meteorological and operational parameters that influence solar irradiance and energy generation. The dataset spans the period from May 2024 to April 2025 and consists of 4,406 hourly records. The variables included in the dataset are:

- Irradiance (W/m^2): The target variable representing the solar power received per unit area on the panel surface,
- Environmental Temperature (Env Temp, $^{\circ}\text{C}$): The ambient air temperature measured near the panel installation site,
- Panel Temperature (Panel Temp, $^{\circ}\text{C}$): The surface temperature of the PV panel,
- Wind Speed (m/s): The average wind speed measured at sensor height,
- Wind Angle ($^{\circ}$): The direction of wind flow, which influences panel cooling,
- Environmental Humidity (Env Humidity, %): The ambient moisture content in the air.

These data were collected using several sensors connected to a data logger, which was synchronized to local time to ensure data accuracy and consistency. Figure 1(a) illustrates the temporal distribution of solar irradiance throughout the dataset, highlighting the daily and monthly patterns. The figure shows that solar irradiance has a strong daily cycle, with peak intensity concentrated around midday (11:00-13:00). This pattern is typical of tropical solar conditions [35-36]. The visualization also reveals seasonal variations in irradiance levels across different months of the year. Since the original dataset includes nighttime hours with zero irradiance values, these non-productive hours were excluded from the analysis to improve forecasting. Only daytime hours with meaningful solar activity were retained, ensuring that the models are trained on relevant data. Figure 1(b) shows the filtered dataset after removing nighttime observations, which consists of 12 daylight hours per day.

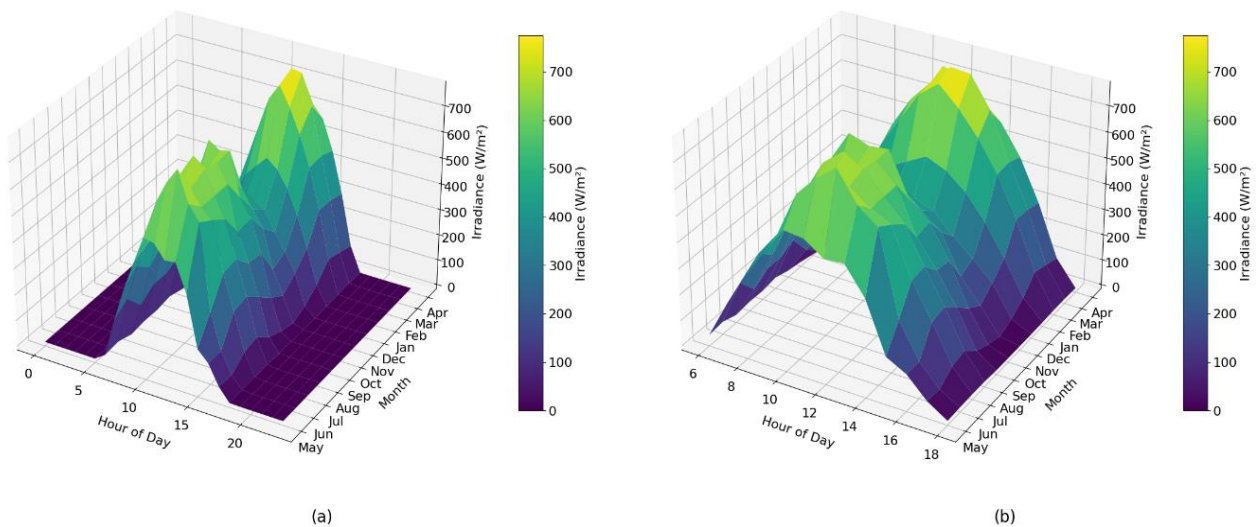


Figure 1. Temporal distribution of solar irradiance: (a) complete 24-hour dataset showing daily and seasonal patterns, (b) filtered daytime-only dataset (12 hours) used for model training and testing.

Figure 2 presents the monthly distributions of all measured variables, with boxplots and trend lines overlaid on the monthly averages. The analysis reveals clear seasonal patterns across all meteorological parameters. Irradiance and temperature variables (both panel and environmental) exhibit a positive correlation and show higher values from February to May, corresponding to the hot, dry season in Thailand. Conversely, these variables reach their lowest levels during the monsoon season from October to December. Relative humidity displays an inverse relationship with irradiance, reaching its highest values during the wet season (October to December) and decreasing as irradiance levels increase during the dry months. Wind speed gradually increases throughout the year, with higher values observed from December to April. Wind angle shows significant seasonal variation, likely influenced by local monsoon patterns that shift the direction of prevailing winds. These seasonal patterns demonstrate that meteorological and climatic factors strongly influence solar energy generation. The observed correlations between irradiance and weather variables justify

the inclusion of environmental predictors (temperature, humidity, and wind characteristics) in the forecasting models, as they contain valuable information that can improve prediction accuracy. Wind speed and wind direction were recorded using an anemometer positioned at approximately 12.5 meters above ground level, consistent with standard meteorological guidelines, since wind behavior changes markedly with height due to surface friction. Atmospheric wind conditions affect solar irradiance indirectly by influencing cloud development, cloud transport, and air turbulence. Stronger winds are commonly linked to faster cloud movement, which can produce abrupt variations in incoming solar radiation, while wind direction provides insight into dominant weather systems, particularly during tropical monsoon periods. For this reason, wind parameters were incorporated into the forecasting models as meteorological inputs to represent atmospheric processes that drive short-term fluctuations in solar irradiance.

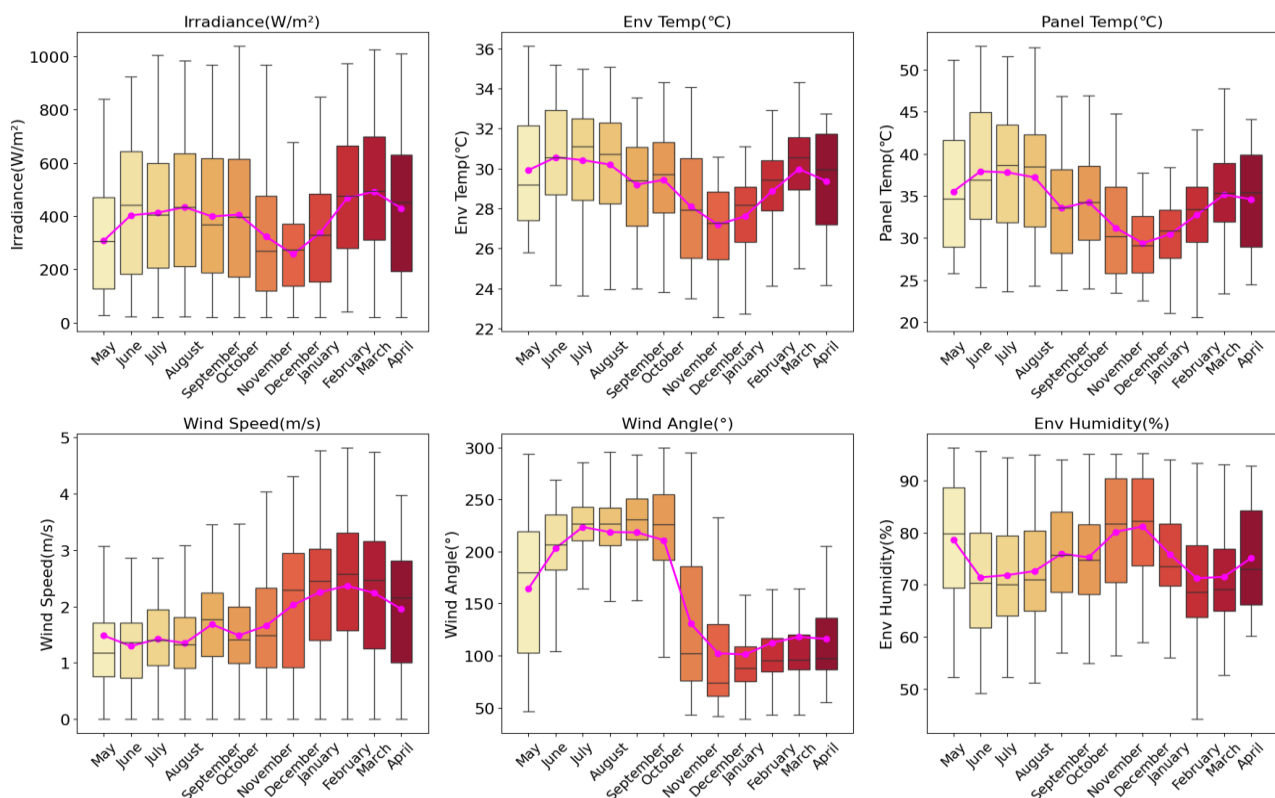


Figure 2. The monthly boxplots of irradiance and meteorological variables.

In this study, "sun irradiance" refers to the Global Horizontal Irradiance (GHI) recorded at an actual monitoring site. GHI, or the total amount of solar radiation incident on a horizontal surface, is composed of diffuse radiation scattered by the atmosphere and direct beam radiation from the sun. Because the irradiance sensor was positioned horizontally, the observed values are appropriate for GHI-based analysis and predictions.

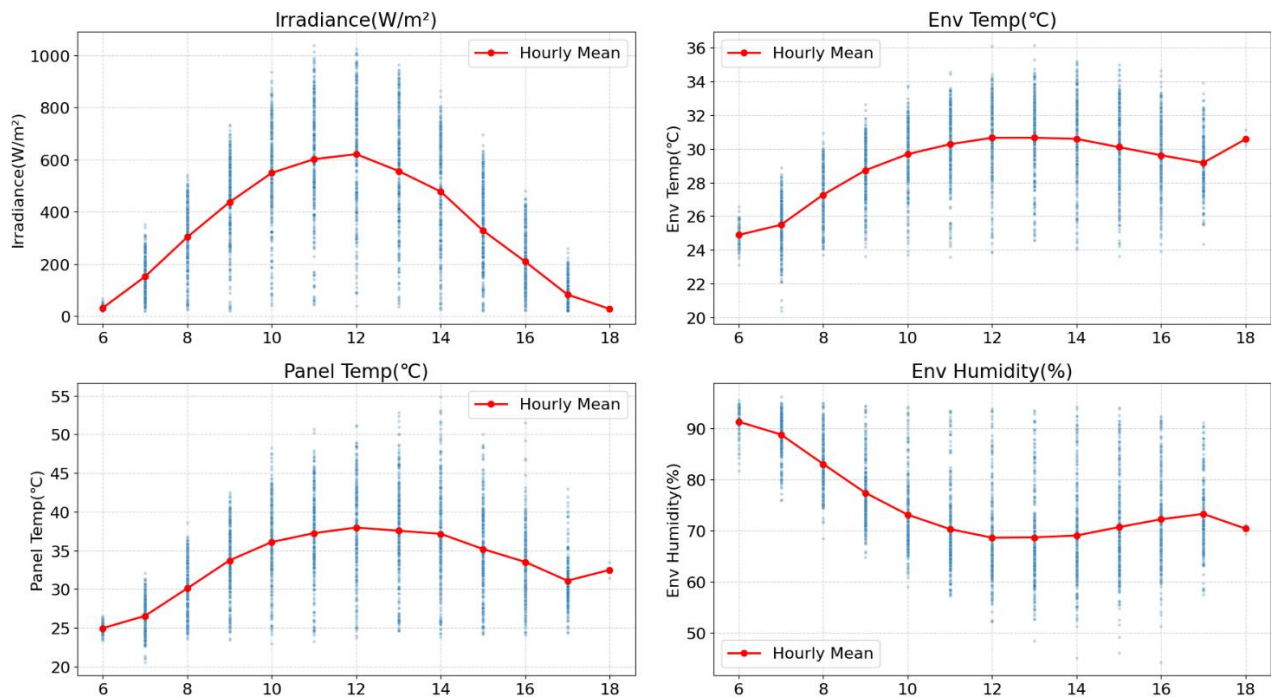


Figure 3. The hourly distribution of the main variables.

Panel temperature is closely related to solar irradiance. As solar radiation increases, the surface temperature of photovoltaic modules also rises. Therefore, under high irradiance, some of the incoming solar energy is converted into electricity, while the rest is converted into heat, which raises the panel's temperature. Moreover, the panel's temperature directly affects how well the photovoltaic (PV) system operates, as higher temperatures often reduce its energy conversion efficiency. Consequently, it serves as a useful indirect indicator of both solar radiation intensity and the system's operational condition. In forecasting applications, panel temperature is treated as an external (exogenous) variable to account for the PV system's thermal response to fluctuating atmospheric conditions. By providing supplementary information beyond irradiance data, panel temperature improves the precision of predictions.

From a daily perspective, Figure 3 presents the hourly distributions of the main variables using histograms, along with the trend line for the hourly mean. The Irradiance curve shows a bell-shaped pattern, with a peak between 11:00 and 13:00, corresponding to maximum solar exposure. Both environmental and panel temperatures increase during the morning, reach their highest values around midday, and then decline toward evening. In contrast, relative humidity decreases as irradiance and temperature rise, then increases in the late afternoon. These results confirm a strong diurnal relationship between solar radiation and atmospheric conditions, supporting the importance of these meteorological variables for enhancing short-term irradiance forecasting accuracy.

3.2 Model hyperparameter configuration

To ensure a fair comparison across all forecasting models, each model was systematically optimized via hyperparameter tuning on the training dataset. For the SARIMA model, a comprehensive grid search was conducted across multiple combinations of non-seasonal (p, d, q) and seasonal P, D, Q, S parameters. The search space included reasonable ranges derived from the observed autocorrelation and partial autocorrelation structures of the irradiance series. Specifically, the orders p, q, P , and Q were varied between 0 and 2, while the differencing orders d and D were restricted to 0 or 1 to maintain model simplicity and avoid over-differencing. The seasonal period was set to $S = 12$, corresponding to the 12 daylight hours retained after preprocessing. In total, 325 unique parameter combinations were evaluated. Each candidate model was trained on 80% of the data (the training set) and assessed using two criteria: the Akaike Information Criterion

(AIC) and the Bayesian Information Criterion (BIC). These metrics balance model accuracy with model complexity by penalizing models that have too many parameters. The model with the lowest AIC and BIC values was selected as the optimal configuration. Figure 4(a) shows the AIC and BIC values for the top 10 model combinations, with the best-performing model being SARIMA (1, 1, 2) (2, 0, 2, 12). Figure 4(b) analyzes the relationship between model complexity and performance. Model complexity is measured by the total number of parameters ($p + d + q + P + D + Q$). The results show that very simple models (2-4 parameters) perform worse than moderately complex models (6-8 parameters). Beyond 8 parameters, there is no significant improvement, and some models with 9-10 parameters actually perform worse.

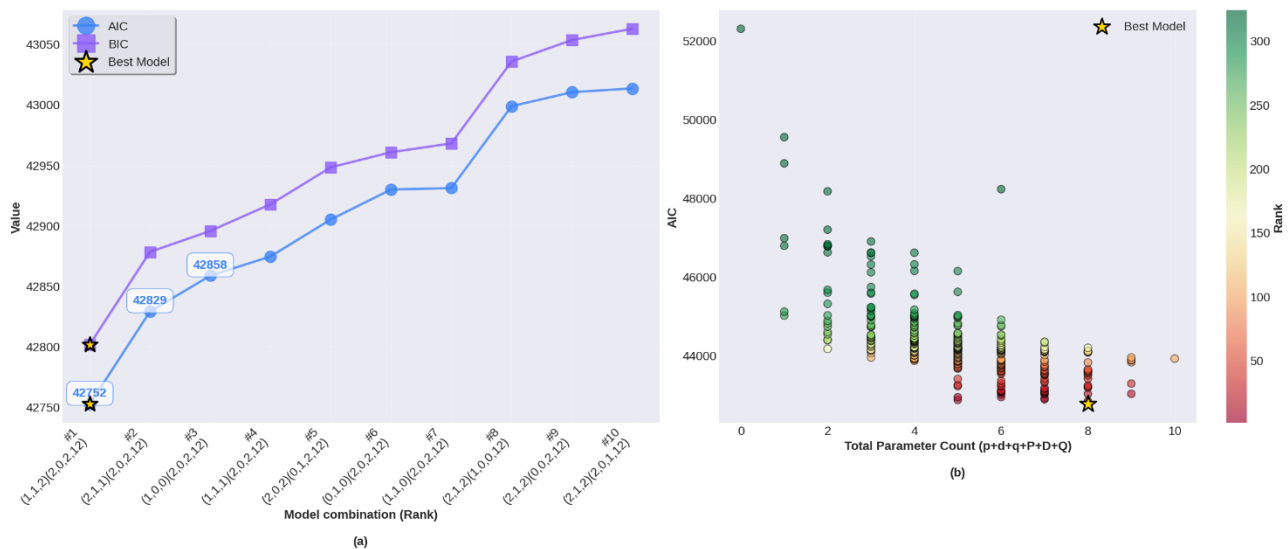


Figure 4. a) AIC and BIC across the top 10 model combinations, b) model complexity versus performance.

The Prophet model was configured to capture daily and weekly seasonality with four meteorological variables (environmental temperature, panel temperature, wind speed, and relative humidity) incorporated as exogenous regressors. Both XGBoost and Random Forest models utilized lag features at 1-hour, 24-hour, and 168-hour intervals, combined with meteorological variables, with all features standardized using a StandardScaler fitted to the training set. The LSTM and GRU networks used a 24-hour lookback window, with all features standardized, and their architectures included a recurrent layer, a dense layer with ReLU activation, and an output layer. Training used the Adam optimizer with early stopping (patience: 5-6 epochs) and learning rate reduction at plateaus. The final hyperparameter configurations for all models are summarized in Table 1.

Table 1. Summary of model hyperparameters and configurations.

Model	Key Hyperparameters	Configuration
SARIMA	$(p, d, q)(P, D, Q, s)$	(1,1,2) (2,0,2,12)
Prophet	Seasonality, Regressors	Daily and weekly, variables
XGBoost	Estimators, Learning rate, Max depth, Lags	500, 0.05, 6, (1, 24, 168)
RF	Estimators, Max depth, Min samples split, Lags	500, 15, 5, (1, 24, 168)
LSTM	Neurons, Batch size, Epochs, Lookback	64, 128, 50, 24
GRU	Neurons, Batch size, Epochs, Lookback	64, 128, 50, 24

3.3 Forecasting results

Six forecasting models, which are SARIMA, Prophet, LSTM, GRU, XGBoost, and RF, were trained on the training set spanning from May 6, 2024, 06:00 to February 1, 2025, 06:00 (3,524 hours, representing 80% of the dataset). The performance of these models was evaluated on the test set from February 1, 2025, 07:00 to April 9, 2025, 17:00 (885 hours, representing 20% of the dataset) and compared against two benchmark models:

the Naive and SNaive methods. Model performance was assessed using three evaluation metrics: RMSE, MAE, and MAPE.

The numerical results are presented in Table 2, which provides a comprehensive comparison of all forecasting models against the benchmark methods. Furthermore, Figure 5 presents a visual comparison of forecasted versus actual irradiance values for all models across the entire test period. The plots reveal that all forecasting models successfully capture the daily cyclic pattern of solar irradiance, with peaks around midday and troughs in the early morning and late afternoon. However, notable differences in tracking accuracy are evident across models. As shown in Table 2, all advanced forecasting models substantially outperformed both benchmark approaches, confirming their ability to capture the complex temporal patterns and meteorological dependencies inherent in solar irradiance data. Between the two benchmarks, the Naive model performed considerably better than the SNaive model, with RMSE improvements of 34.06 W/m² and MAPE reductions of 22.78%. This suggests that short-term persistence (hour-to-hour continuity) is a more reliable forecasting assumption than daily periodicity for this tropical solar irradiance dataset.

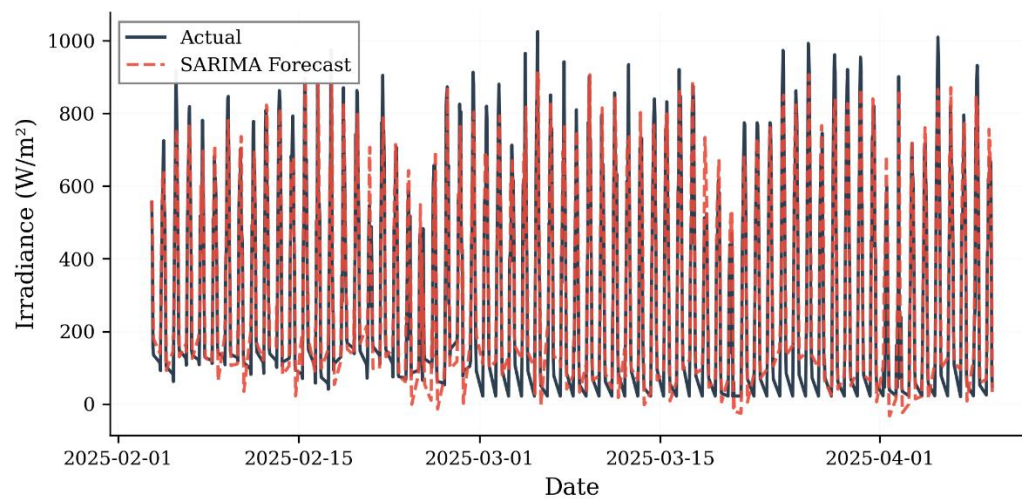
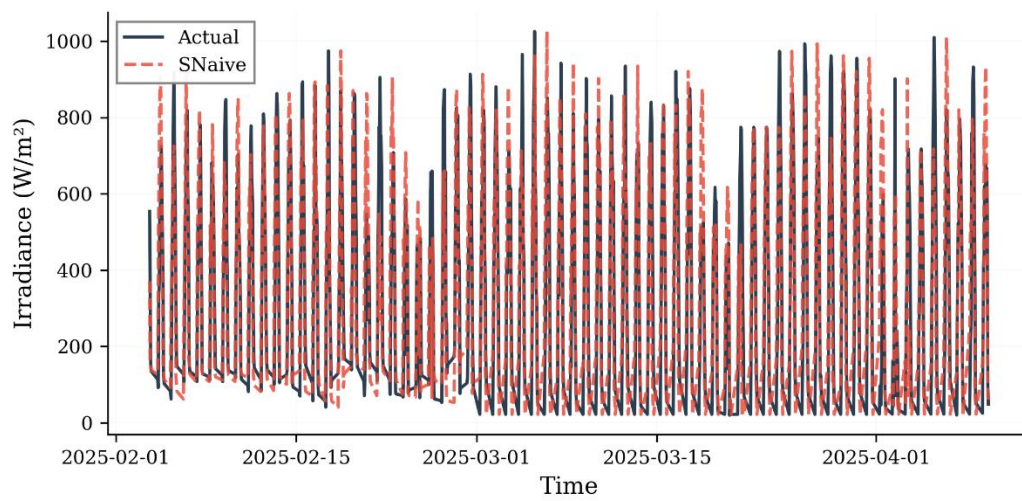
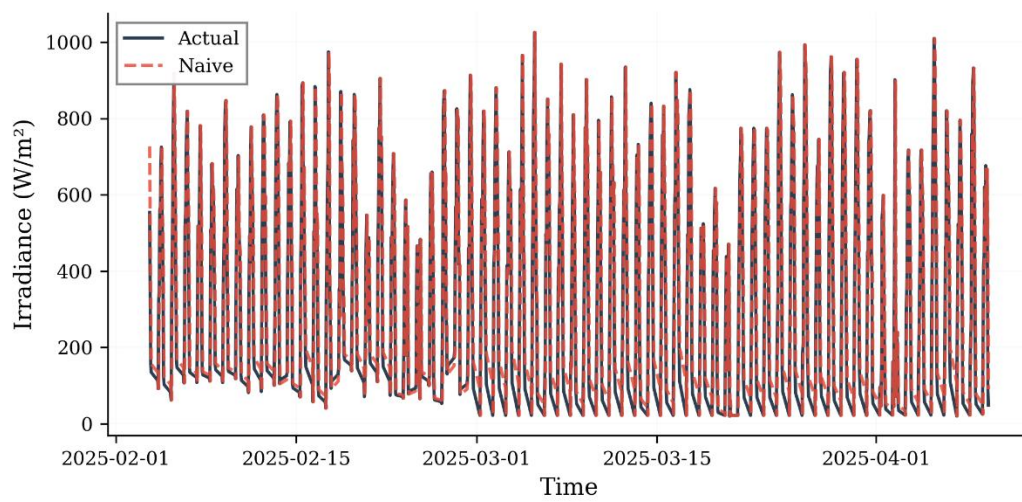
Table 2. Forecasting performance comparison across all models.

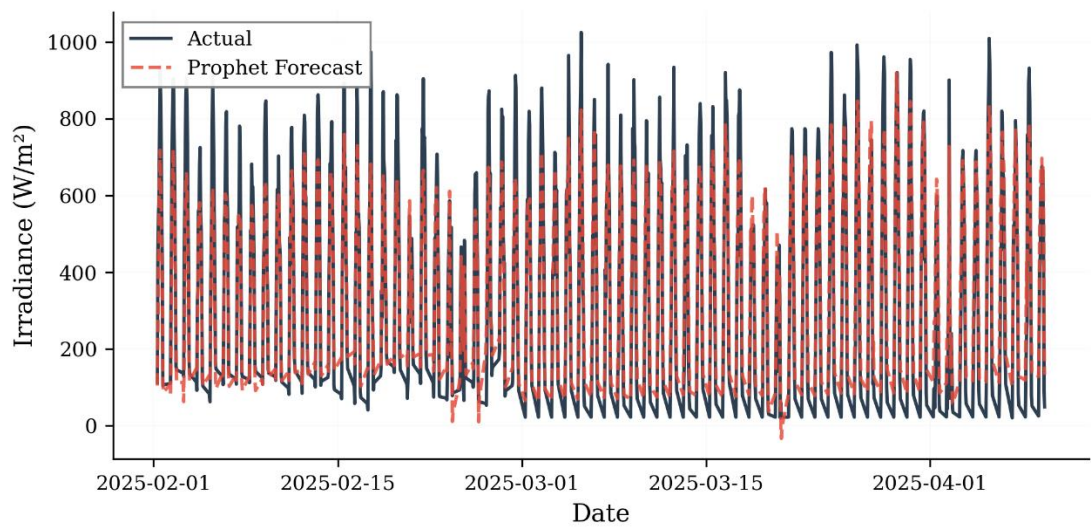
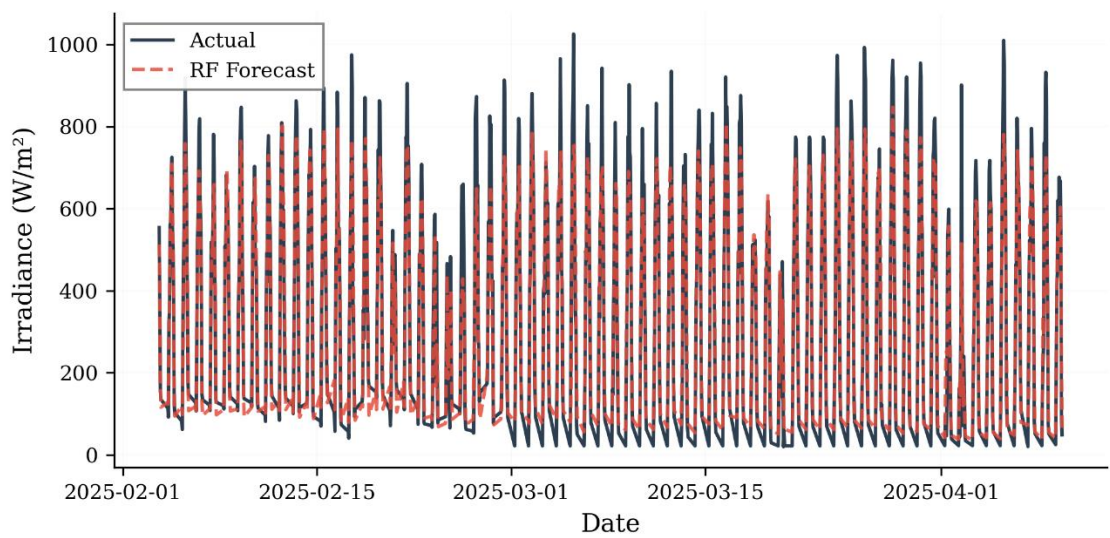
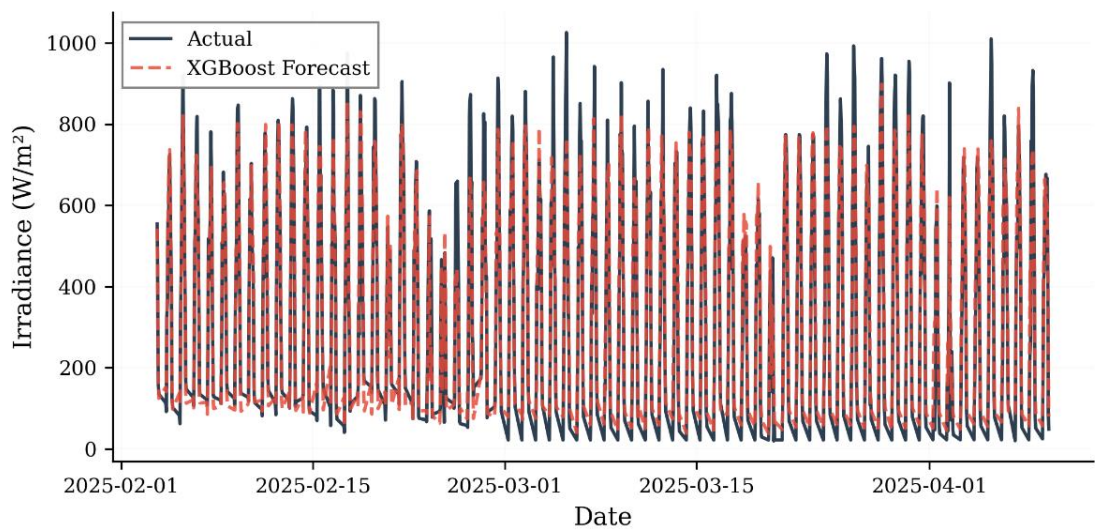
Model type	Models	Evaluation metrics		
		RMSE	MAE	MAPE %
Forecasting models	SARIMA	110.79	83.78	49.29
	Prophet	94.31	74.59	41.85
	LSTM	140.36	105.86	43.79
	GRU	106.94	78.24	35.98
	XGboost	79.43	59.63	30.07
	RF	83.00	61.98	29.36
Benchmark models	Naive	159.80	129.48	56.89
	SNaive	193.86	153.06	79.67

Among the forecasting models, the ensemble-based machine learning approaches demonstrated superior performance. XGBoost achieved the best overall results across all metrics, yielding the lowest RMSE (79.43 W/m²) and MAE (59.63 W/m²), with a MAPE of 30.07%. RF produced comparable results, achieving the lowest MAPE (29.36%) while maintaining competitive RMSE (83.00 W/m²) and MAE (61.98 W/m²) values. The strong performance of both ensemble methods can be attributed to their ability to effectively model nonlinear feature-target relationships and capture complex interactions between meteorological variables and solar irradiance. Compared to the best benchmark (Naive), XGBoost reduced RMSE by 50.3%, MAE by 53.9%, and MAPE by 47.1%, demonstrating substantial improvements in forecasting accuracy.

The deep learning models exhibited mixed results. The GRU model outperformed the LSTM across all metrics, achieving an RMSE of 106.94 W/m² compared to the LSTM's 140.36 W/m², and reducing MAPE from 43.79% to 35.98%. This performance difference highlights GRU's greater efficiency in modelling short-term temporal dependencies with fewer parameters and its superior stability during training. However, both recurrent neural network architectures were outperformed by the machine learning ensemble methods, suggesting that the added complexity of deep learning may not be necessary for this particular forecasting task, given the dataset size and temporal structure.

Among the statistical and hybrid models, Prophet demonstrated competitive performance with an RMSE of 94.31 W/m² and an MAPE of 41.85%, substantially outperforming SARIMA (RMSE: 110.79 W/m², MAPE: 49.29%). Prophet's strength lies in its ability to decompose automatically and model trend and seasonal components while incorporating exogenous meteorological variables with minimal parameter tuning. In contrast, despite the comprehensive grid search optimization and rigorous walk-forward validation employed for SARIMA, the model achieved the second-worst performance among all forecasting methods, only surpassing LSTM.





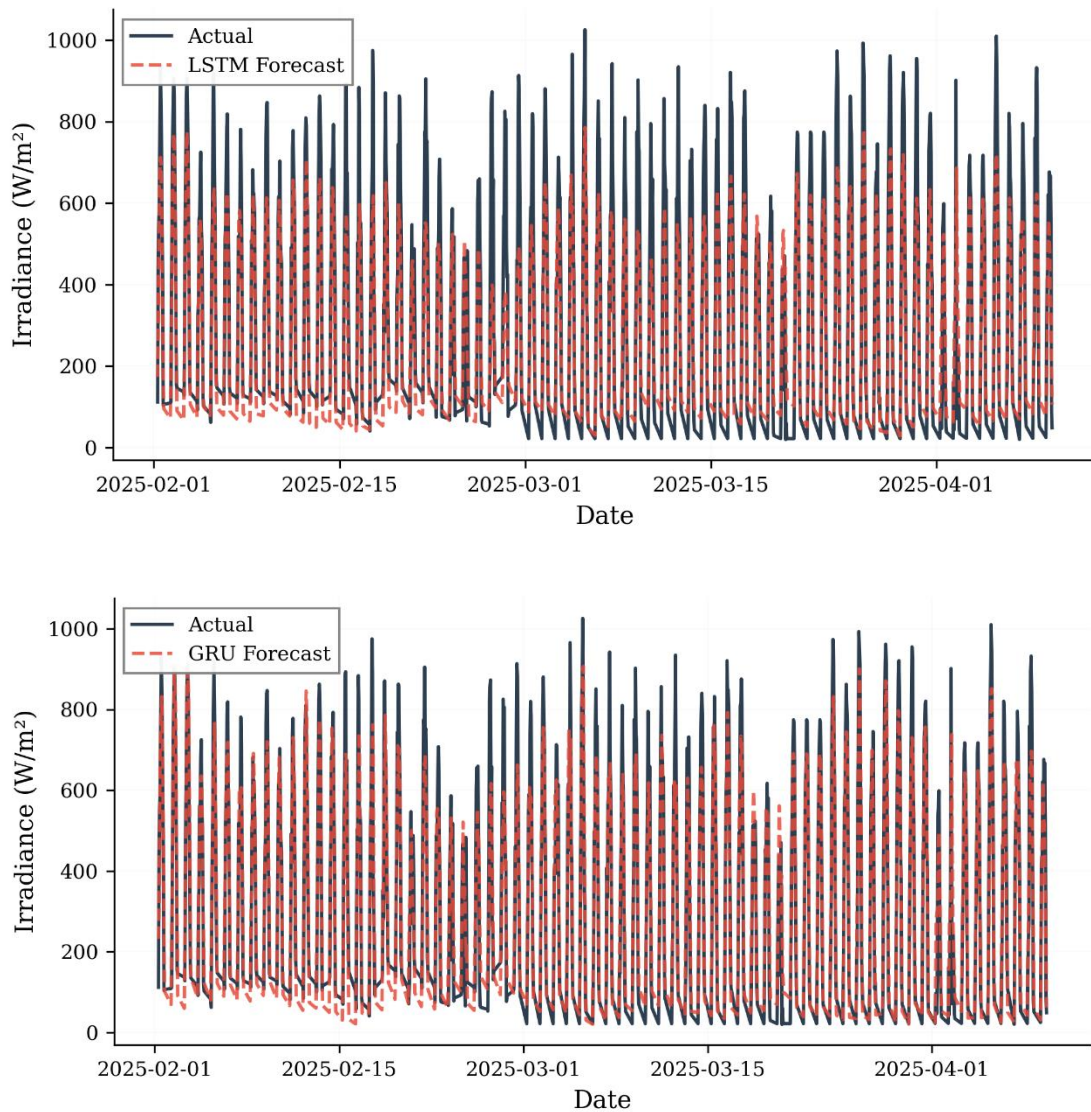


Figure 5. Time series comparison of actual versus forecasted irradiance for all models over the test period. Black solid lines represent actual values, while red dashed lines show model predictions.

In addition to forecasting accuracy, computational efficiency was evaluated for all models. The machine learning and deep learning models (XGBoost, RF, LSTM, and GRU) required 5-10 minutes each for the complete training and prediction cycle, making them highly suitable for real-time or near-real-time forecasting applications. In contrast, the walk-forward SARIMA model required approximately 170 minutes (2 hours and 50 minutes) to complete its training and walk-forward forecasting, due to iterative retraining at each time step—a computationally intensive but methodologically rigorous approach recommended in the time series forecasting literature to ensure temporal robustness and prevent data leakage.

In conclusion, the results demonstrate that ensemble-based machine learning models (XGBoost and Random Forest) provide the most accurate and computationally efficient forecasts for hourly solar irradiance prediction. These models effectively exploit both temporal persistence and nonlinear meteorological effects, making them well-suited for short-term solar forecasting in tropical climates. The GRU model offers a viable alternative when sequential modelling is preferred, outperforming LSTM while maintaining reasonable computational efficiency. Prophet provides a practical compromise between statistical rigor and forecasting accuracy, particularly valuable when model interpretability and automatic seasonality detection are priorities. Overall, the findings confirm that modern machine learning approaches significantly outperform traditional statistical methods for this application, both in terms of accuracy and computational efficiency.

3.4 Practical implications for intelligent solar energy management

Accurate short-term solar irradiance forecasting is crucial for advanced energy management systems, especially in regions with substantial renewable energy integration. The models developed in this research support real-time operational decision-making processes, including load management strategies, maintaining grid stability, and optimal coordination of energy storage units. This capacity is especially vital in tropical areas, such as Southern Thailand, where the monsoon's influence on cloud dynamics and seasonal shifts substantially affects solar irradiance. Accurate predictions empower operators to anticipate fluctuations in power generation, thereby facilitating proactive strategies, such as activating backup generators or adjusting battery charging and discharging protocols. Consequently, the strong efficacy demonstrated by ensemble machine learning methods highlights their applicability in automated energy management systems, thereby optimizing the utilization of renewable energy sources, diminishing dependence on fossil-fuel-based backup systems, and enhancing overall system functionality. In accordance with this premise, the research outcomes provide substantial support for the advancement of advanced solar energy management systems, specifically designed to promote extensive renewable integration within tropical environments [37].

4. Conclusions

This study developed and evaluated multiple forecasting models for predicting hourly solar irradiance using data collected from a photovoltaic monitoring station at Thaksin University, Thailand. The research compared six forecasting approaches, namely SARIMA, Prophet, LSTM, GRU, XGBoost, and Random Forest, against two benchmark models (Naive and SNaive). The results showed that ensemble machine learning models performed best, with XGBoost achieving the highest overall accuracy (RMSE = 79.43 W/m², MAE = 59.63 W/m², and MAPE = 30.07%), followed by Random Forest. In contrast, the statistical and deep learning models showed lower predictive performance under the conditions of this dataset. These findings indicate that ensemble methods are highly effective for short-term solar irradiance forecasting in tropical environments and can support more reliable solar energy management. These findings align with previous research in similar climatic regions. A study conducted in Rajasthan, India (another tropical region with high solar potential) found that Random Forest achieved the best performance with an average error below 20%, outperforming linear regression by 19.47%. This consistency across different tropical locations suggests that ensemble methods are particularly well-suited for solar irradiance forecasting in regions with complex meteorological dynamics and strong seasonal variations. Future work should evaluate hybrid models and longer, multi-year datasets to improve generalizability and forecasting robustness further.

5. Acknowledgments

We would like to sincerely thank Thaksin University's Faculty of Engineering for providing important facilities, necessary equipment, and useful data that made this research much easier to complete. Additionally, we would like to express our gratitude to Thaksin University for making academic resources accessible and for their outstanding assistance throughout this research.

Author Contributions: The simulation strategy and techniques were conceptualized and designed by M.S., S.A., T.S., and S.K. M.S. and S.K. implemented the simulation, collected the data, analyzed the results, and reached the conclusion. The authors M.S., S.A., T.S., W.S., and S.K. contributed to the writing and the general content modification.

Funding: This study was funded by the Researcher and Research for Industry Development Program (RRID), which is managed by the National Research Council of Thailand (NRCT), Ministry of Higher Education, Science, Research, and Innovation.

Conflicts of Interest: The study's authors hereby declare that they have no competing interests.

References

- [1] Antonanzas, J.; Osorio, N.; Escobar, R.; Urraca, R.; Martinez-de-Pison, F. J.; Antonanzas-Torres, F. Review of photovoltaic power forecasting. *Sol. Energy* **2016**, *136*, 78–111. <https://doi.org/10.1016/j.solener.2016.06.069>
- [2] Armstrong, B.; Collopy, F. Error measures for generalizing about forecasting methods. *Int. J. Forecast.* **1992**, *8* (1), 69–80. [https://doi.org/10.1016/0169-2070\(92\)90008-W](https://doi.org/10.1016/0169-2070(92)90008-W)
- [3] Hussain, D.; Hossain, S.; Talukder, J.; Mia, A.; Shamsuzzaman, H. M. Solar energy integration into smart grids: Challenges and opportunities. *Lett. High Energy Phys.* **2024**, *4*, 2313–2324
- [4] Sammar, M. J.; Saeed, M. A.; Mohsin, S. M.; Akber, S. M.; Bukhsh, R.; Abazeed, M.; Ali, M. Illuminating the future: A comprehensive review of AI-based solar irradiance prediction models. *IEEE Access* **2024**, *12*, 114394–114415. <https://doi.org/10.1109/ACCESS.2024.3402096>
- [5] Teixeira, R.; Cerveira, A.; Pires, E. J. S.; Baptista, J. Advancing renewable energy forecasting: A comprehensive review of renewable energy forecasting methods. *Energies* **2024**, *17*(14), 3480. <https://doi.org/10.3390/en17143480>
- [6] Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Doha, Qatar, 2014; pp 1724–1734. <https://doi.org/10.3115/v1/D14-1179>
- [7] Crone, A.; Goel, S.; Billings, S. Forecasting with computational intelligence: An overview. In *Computational Intelligence in Forecasting*; Springer: Cham, Switzerland, 2020; pp 1–24.
- [8] Diagne, M.; David, M.; Lauret, P.; Boland, J.; Schmutz, N. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renew. Sustain. Energy Rev.* **2013**, *27*, 65–76. <https://doi.org/10.1016/j.rser.2013.06.042>
- [9] Assaf, A. M.; Haron, H.; Abdull Hamed, H. N.; Ghaleb, F. A.; Qasem, S. N.; Albarrak, A. M. A review on neural network based models for short-term solar irradiance forecasting. *Appl. Sci.* **2023**, *13*(14), 8332. <https://doi.org/10.3390/app13148332>
- [10] Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- [11] Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, 2016.
- [12] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [13] Hyndman, R. J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 2nd ed.; OTexts: Melbourne, Australia, 2018.
- [14] International Energy Agency (IEA). *Renewable Energy Market Update*; IEA: Paris, France, 2023.
- [15] Kumari, P.; Toshniwal, D. Deep learning models for solar irradiance forecasting: A comprehensive review. *J. Clean. Prod.* **2021**, *318*, 128566. <https://doi.org/10.1016/j.jclepro.2021.128566>
- [16] Hossain, M. A.; Rahman, M. M.; Hasan, S. S.; Mahmud, A.; Bai, L. Analysis and forecasting of meteorological drought using PROPHET and SARIMA models deploying machine learning technique for southwestern region of Bangladesh. *Environ. Sustain. Indic.* **2025**, *25*, 100761. <https://doi.org/10.1016/j.indic.2025.100761>
- [17] Tsai, W. C.; Tu, C. S.; Hong, C. M.; Lin, W. M. A review of state-of-the-art and short-term forecasting models for solar PV power generation. *Energies* **2023**, *16* (14), 5436. <https://doi.org/10.3390/en16145436>
- [18] Mellit, A.; Kalogirou, S. A. Artificial intelligence techniques for photovoltaic applications: A review. *Prog. Energy Combust. Sci.* **2008**, *34* (5), 574–632. <https://doi.org/10.1016/j.pecs.2008.01.001>
- [19] Ministry of Energy, Department of Alternative Energy Development and Efficiency. *Thailand Solar Energy Potential Report*; Ministry of Energy: Bangkok, Thailand, 2022.

- [20] Baloch, M.; Honnurvali, M. S.; Kabbani, A.; Ahmed, T.; Chauhdary, S. T.; Saeed, M. S. Solar energy forecasting framework using Prophet-based machine learning model: An opportunity to explore solar energy potential in Muscat, Oman. *Energies* **2025**, *18*(1), 205. <https://doi.org/10.3390/en18010205>
- [21] Reikard, G. Predicting solar radiation at high resolutions: A comparison of time series forecasts. *Sol. Energy* **2009**, *83*(3), 342–349. <https://doi.org/10.1016/j.solener.2008.08.007>
- [22] Raschka, S.; Liu, Y. H.; Mirjalili, V. *Machine Learning with PyTorch and Scikit-Learn: Develop Machine Learning and Deep Learning Models with Python*, 2nd ed.; Packt Publishing: Birmingham, U.K., 2022.
- [23] Obando, E. D.; Carvajal, S. X.; Agudelo, J. P. Solar radiation prediction using machine learning techniques: A review. *IEEE Lat. Am. Trans.* **2019**, *17*(4), 684–697. <https://doi.org/10.1109/TLA.2019.8891934>
- [24] Siami-Namini, H.; Tavakoli, N.; Namin, A. S. A comparison of ARIMA and LSTM in forecasting time series. In *Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA 2018)*; IEEE: Orlando, FL, USA, 2018; pp 1394–1401. <https://doi.org/10.1109/ICMLA.2018.00227>
- [25] Chaves, R. R.; Oliveira, A. F.; Rubinger, R. M.; Silva, A. J. Recurrent neural networks (LSTM and GRU) in the prediction of current–voltage characteristics curves of polycrystalline solar cells. *Electronics* **2025**, *14*(17), 3342. <https://doi.org/10.3390/electronics14173342>
- [26] Voyant, C.; Notton, G.; Kalogirou, S.; Nivet, M. L.; Paoli, C.; Motte, F.; Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renew. Energy* **2017**, *105*, 569–582. <https://doi.org/10.1016/j.renene.2016.12.095>
- [27] Willmott, C. J.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* **2005**, *30*(1), 79–82. <https://doi.org/10.3354/cr030079>
- [28] Box, G. E. P.; Jenkins, G. M.; Reinsel, G. C.; Ljung, G. M. *Time Series Analysis: Forecasting and Control*, 5th ed.; John Wiley & Sons: Hoboken, NJ, 2015.
- [29] Safari, M.; Nakharutai, N.; Awiphan, S. Beyond traditional models: Predicting cryptocurrency prices with SARIMA, RNNs, and hybrid SARIMA–RNNs. *J. Financ. Data Sci.* **2025**, *7*(3). <https://doi.org/10.3905/jfds.2025.1.193>
- [30] Hyndman, R. J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 2nd ed.; OTexts: Melbourne, Australia, 2018.
- [31] Safari, M.; Nakharutai, N.; Chiawkhun, P.; Phetpradap, P. DEA-RNNs: An ensemble approach for portfolio selection in the Thailand stock market. In *Proceedings of ECONVN2024*; Ho Chi Minh City, Vietnam, January 9–11, 2024; Springer Nature: Singapore, 2025. https://doi.org/10.1007/978-3-031-59110-5_30
- [32] Taylor, S. J.; Letham, B. Forecasting at scale. *Am. Stat.* **2018**, *72*(1), 37–45. <https://doi.org/10.1080/00031305.2017.1380080>
- [33] Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [34] Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*; ACM: San Francisco, CA, USA, August 13–17, 2016; pp 785–794. <https://doi.org/10.1145/2939672.2939785>
- [35] Tantasiriwong, R.; Tantasiriwong, S. The solar radiation climate of Thailand. *Sol. Energy* **1976**, *18*(1), 43–49. [https://doi.org/10.1016/0038-092X\(76\)90062-1](https://doi.org/10.1016/0038-092X(76)90062-1)
- [36] Wang, X.; Zhang, Y.; Tan, L. Solar radiation and daily light integral characteristics of photovoltaic-green roof system in tropical area: A case study in Singapore. *Sol. Energy* **2024**, *274*, 117–129. [https://doi.org/10.1016/S0038-092X\(24\)00502-4](https://doi.org/10.1016/S0038-092X(24)00502-4)
- [37] Tandon, A.; Awasthi, A.; Pattnayak, K. C.; Tandon, A.; Choudhury, T.; Kotecha, K. Machine learning-driven solar irradiance prediction: Advancing renewable energy in Rajasthan. *Discover Appl. Sci.* **2025**, *7*(2), 107. <https://doi.org/10.1007/s42452-025-06490-8>