Solar Photovoltaic Power Output Prediction Using Machine Learning-Based Regressors

Gregorius Satia Budhi[1] , Yusak Tanoto[2*], Dick Jovian[1]
Rudy Adipranata[1] , Clement Raphael[2]

[1]Informatics Department, Faculty of Industrial Technology,
Petra Christian University, Surabaya, 60236, Indonesia
[2]Electrical Engineering Department, Faculty of Industrial Technology,
Petra Christian University, Surabaya, 60236, Indonesia

Abstract

This study proposes a framework for predicting solar photovoltaic (solar PV) power output using Machine Learning-based regressors for short-, medium-, and long-term prediction horizons. To identify the most effective regressor, we propose a comparison framework to evaluate the performance of several types of regressor models. This evaluation will include Neural Networks, Boosting and Bagging Ensembles, and a baseline assessment using a linear regressor family. In this study, we implement the grid search method to improve model performance by fine-tuning hyperparameters, as does the K-fold shuffle split cross-validation method. We consider large spatial and long temporal historical datasets for the case study. A 5 km x 5 km gridded hourly temporal-based 1 MW modelled Solar PV dataset consisting of direct and diffuse irradiation, temperature, and power output during 2013-2022 in the Java-Bali region, Indonesia, is used as a case study. The grid search-optimized Neural Networks family, the Multilayer Perceptron model, can accurately predict power output from short-, medium-, and long-term horizons, with an average MAE of 0.248 kW and an average RMSE of 0.306 kW, followed by Random Forest, a grid search-optimized Bagging Ensemble and a grid search-optimized Histogram Gradient Boosting Ensemble model. All predictor models generally performed well under strong El-Nino-affected data but were sensitive to very strong El-Nino during 2015-2016. The method used and insights gained from this study also benefit other jurisdictions with similar contexts.

Keywords: machine learning, power output prediction, regressors, shuffle split cross-validation, solar photovoltaic

1. Introduction

Asia and other parts of the world are currently facing unprecedented rises in energy demand and environmental challenges, requiring every country to accelerate the energy transition [1].

*Corresponding author: tanyusak@petra.ac.id
Received: 6 January 2025 Accepted: 24 April 2025 Published: 28 April 2025
Journal of Asian Energy Studies (2025), Vol 9, 111-130, doi:10.24112/jaes.090007

Renewable energy (RE) technologies have emerged as viable, clean energy sources that facilitate the electricity industry transition from fossil fuels, including in Asian developing countries [2,3]. Nonetheless, numerous barriers to higher RE penetration are relevant factors that require deep attention and must be resolved by stakeholders [4]. RE technologies are the most likely anticipated strategies that countries have established and are implementing to meet a significant portion of total electricity demand by 2030, eventually replacing fossil fuels [5,6] and mitigating environmental impact [1]. Solar photovoltaic (solar PV) is a rapidly advancing, cost-competitive renewable energy technology [7,8]. The recent development of large energy storage systems enables a greater share of energy from solar PV during periods of insufficient solar radiation [9].

Global solar PV capacity is expected to increase to 2,840 GW by 2030 and 8,519 GW by 2050, up from 480 GW in 2018 [7]. In Southeast Asia, RE will account for over three-quarters of electricity over the long run. Solar PV will account for approximately 1,100 GW of this share, while fossil fuel sources will account for less than 10%. By 2050, solar PV will account for nearly 1,600 Terawatt-hours of the region's electricity generation [10].

The electricity generated by solar PV is primarily influenced by direct and diffuse irradiation and temperature [11,12]. The temperature significantly impacts the efficiency of solar PV panels. In full sunlight, the temperature is typically 40 °C higher than the ambient temperature [13]. Every ten degrees of temperature increase reduces the efficiency of crystalline silicon Solar PV by 6.5% to 10% [13,14].

This study addresses the gaps in spatially and temporally predicting solar PV power output. We aim to enhance the literature on machine learning (ML) applications for solar PV power output forecasting by introducing an ML-based framework that utilises gridded long-term hourly datasets encompassing direct radiation, diffuse radiation, temperature, and power output. This study uses the Java-Bali regions of Indonesia as a case study and particularly applies several types of ML, which are: a Neural Networks type, the Multilayer Perceptron (MLP) [15,16]; an ensemble boosting type, the Histogram Gradient Boosting (HGB) [17]; and a Bagging ensemble type, the Random Forest (RF) [18] as regressor model candidates and evaluates their performance. Besides that, we utilised Multiple Linear Regression (MLnR) [19] as a baseline assessment. Moreover, this study also applies the Grid Search (GS) method[1] to tune each regressor's hyperparameter to improve the models' performance, and the Shuffle Split Cross-validation (SSCV)[2], a technique to train and test the regressors. Their performance is measured using Mean Absolute Error (MAE), Mean Squared Error (MSE), root MSE (RMSE) and R^2.

Another significant research gap identified in prior studies is the lack of examination of the impact of climate occurrences, such as El Niño, on the analysis. This study, therefore, examines how El Niño influences the performance of the proposed models. This work thus contributes to relevant research areas of solar energy supply prediction towards a more sustainable energy future, particularly in the context of developing countries, while also considering the potential impact of complex weather pattern phenomena like El Niño on prediction accuracy. Accurate Solar PV power output prediction will provide insights into power sector investment, including selecting potential solar power plant locations and assisting the system planners and operators in managing Solar PV electricity generation planning and fleet operations.

The structure of this paper is as follows: In Section 2, we provide a comprehensive review of related work regarding the outputs of solar PV prediction. Section 3 elaborates on the dataset employed in this study and outlines the detailed design of the solar PV prediction system. The experimental results and corresponding discussions are presented in Section 4. Lastly, the Conclusion section summarizes our findings and identifies potential directions for future research.

[1]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[2]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ShuffleSplit.html

2. Related Work

The output prediction for solar PV systems is generally categorised according to the prediction horizon. This term refers to the timeframe into the future for which the photovoltaic power output is anticipated [5,20]. For this study, we will adhere to the category established by Iheanetu K.J [20]. The first category is centred on the very short-term prediction horizon, which encompasses a timeframe from a few seconds to less than one hour. This category plays a critical role in the management of power distribution [21,22]. The next prediction horizon is short-term, typically from hours to days. This timeframe is crucial for the effective commitment, scheduling, and dispatch of generated solar PV power. Recent studies have increasingly concentrated on enhancing the accuracy of short-term solar PV output predictions [23–28]. The third category is designated as medium-term, encompassing a timeframe of 1 week to 1 month. This category plays a crucial role in optimising the planning and maintenance schedule of the solar PV system.

Notably, research efforts have predominantly concentrated on longer timeframes, such as short- to medium-term analyses [29,30], medium- to long-term [31] or short- to long-term [2,32,33]. The final category identified is long-term, encompassing timeframes ranging from one month to over a year. Projections of solar PV output for the long term are critical for effective planning in electricity generation, transmission, and distribution. In addition to the previously mentioned studies on extended prediction horizons, numerous researchers have dedicated their efforts specifically to exploring the long-term category [34,35]. A concise overview of related work from the past five years (2020–2024) is provided in Table 1.

The input data are usually gathered from sensors and other measurement equipment. The attributes used in the studies for the input features are solar irradiation and temperature. Moreover, some studies used and added other attributes such as datetime and season [2,22,28]; weather conditions [23,29,35]; wind speed, air pressure, and humidity [2,23,27–30,35]; and tilt and azimuth angles of the solar PV devices [21,23]. Other studies have used time-series data to predict Solar PV output in the future [26,33] or predict solar irradiation to calculate the amount of Solar PV output [22,31]. Most studies keep their dataset in secret, except a few publish it to be used in other studies [31,32]. The challenge associated with private datasets is that they hinder others from replicating the research or advancing the study, which may prevent the achievement of improved outcomes. We obtained our datasets from the publicly available Renewables.Ninja website, ensuring that our study is easily replicable and can be enhanced by others in the field.

Recent studies mainly utilised ML regressors to predict the solar PV output with promising results [2,21,23,24,26,28–30,32,34,35]. The ML models include the MLP/Artificial Neural Network (ANN)/Backpropagation NN (BPNN)/Feed-forward NN (FFNN), Ridge Regression (RdR), Lasso Regression (LsR), Adaptive Boosting (AB), K-Nearest Neighbor (K-NN), Decision Trees (DT), RF, Extreme Gradient Boosting (XGB), Support Vector Machine Regressor (SVR), Principal Component Analysis (PCA), Long short-term memory (LSTM), Recurrent neural network (RNN), Gated Recurrent Unit (GRU) and Transformer, which were tested for Solar PV output prediction. While all the models performed well in predicting the output of Solar PV (see Table 1), most of these studies focused on specific private datasets and also a specific range of prediction horizons (i.e., short-range, short to medium, or long-range). Using GS to optimise the ML models, our study could identify the best model that could work in short-, medium- and long-range prediction horizons on a public dataset.

Despite the success of ML/DL regressors, more traditional regressor methods, such as Linear Regression (LnR), MLnR, Auto-regressive integrated moving average (ARIMA), Seasonal-ARIMA (SARIMA) and ARIMA with exogenous variable (ARIMAX) were still tested to predict Solar PV output of the time series data [2,23,31–33]. Traditional regression methods frequently do

not achieve the predictive accuracy of ML models. Additionally, approaches such as ARIMA and SARIMA are limited to forecasting a variable based solely on its historical values. While the ARIMAX allows for the inclusion of one additional variable only in the prediction process. Therefore, to achieve better results, Fan et al. [36] combined ARIMA with ML methods such as BPNN and SVR.

Our study employs solar irradiation, encompassing both direct and diffuse components, as well as ambient temperature, as key input features. We have also included location data, specifying the relevant Regency or City, to enhance the predictive accuracy of our solar PV output model across diverse geographical contexts. For this research, we have sourced datasets from publicly available resources generated by MERRA-2 [37], which are also provided through the renewables.Ninja website. This methodology is designed to promote transparency and facilitate the replication of our study by other researchers.

As mentioned before, we evaluated three machine learning models, MLP, HGB, and RF, as potential predictor candidates. Additionally, we included a traditional regression model, MLnR, to serve as a baseline for comparison. Each of these models, along with MLnR, underwent optimization using the GS method. The performance of these models was assessed on a comparison platform that was designed based on our prior research [38,39]. The SSCV is employed to evaluate the performance of various model candidates. This validation process is essential for ensuring the reliability of systems developed for the accurate prediction of solar PV output.

3. Material and Methods

This study gathers solar irradiation (direct and diffuse), ambient temperature, and solar PV power output as input attributes from MERRA-2-based Solar PV model datasets in the Renewables.Ninja website [8,34]. In this study, these hourly temporal-based solar PV datasets are gridded with a spatial resolution of 0.05° x 0.05°, or every 0.5 km^2, collected from all locations in Indonesia's Java and Bali areas, from 2013 to 2022. This research also determines the geographical coordinates of all Regencies/cities across the Java-Bali region, Indonesia, for solar PV power output prediction at those locations, based on the best annual solar PV capacity factor. Figure 1 (above) shows the location coordinates of a spatial resolution of 0.05° x 0.05° within the Java-Bali region, Indonesia, and (below) the mapping of the 1-year PV capacity factor in all Java-Bali areas in 2015, which implicitly shows the solar PV output level of a modeled 1 MW solar PV plant in each spatial resolution [40].

As previously mentioned in the introduction section, this study assesses four regressor models: The MLP – an artificial neural networks method; The HGB, which is based on an ensemble boosting method; and RF, which is based on an ensemble bagging method; as the predictor candidates along with one traditional regressor, the MLnR, a linear regressor family that is commonly used as the baseline. In this study, all these models are built using the scikit-learn library [41].

The MLP model learns mainly using two phases: The 1st phase is Feed-forward, and the 2nd phase is backpropagation [42]. The Feed-forward phase will present input data $x_i(p)$ and propagate this data through the output to generate predicted output y_k for each output unit. The formula for this phase can be seen in equations 1 and 2.

$$y_j(p) = activation\_function\left(\sum_{i=1}^{n} x_i(p) \cdot w_{ij}(p)\right) \tag{1}$$

$$y_k(p) = activation\_function\left(\sum_{i=1}^{m} x_{jk}(p) \cdot w_{jk}(p)\right) \tag{2}$$

Table 1: *An overview of related work from 2020 to 2024*

Author	Year	Prediction Horizon	Dataset[1]	Method[2]	Best Result[3]
Lee et al. [29]	2024	Short- to medium-term	(Pr) Input: air pressure, temperature, humidity, wind speed, rainfall, solar irradiance. Output: Solar PV output	LSTM, MLP	nRMSE = 8.03%
Cui et al. [30]	2024	Short- and medium-term	(Pr) Input: solar irradiance, air pressure, wind speed, humidity. Output: Solar PV output	MLP	MAE = 2.36; MAPE = 13.95%; RMSE = 6.28 kW
Asiedu et al. [32]	2024	Short- to long-term	(Pu) Input: solar irradiance, module and ambient temperature. Output: Solar PV output	ANN, RdR, LnR, LsR, AB, XGB, K-NN, DT, RF, ANN-RF, XGB-RF, ANN-XGB-RF	R2 = 0.87; MAE = 0.3; RMSE = 0.75
Scott et al. [2]	2023	Short- to long-term	(Pr) Input: cloud coverage, humidity, rainfall, air pressure, temperature, wind speed, DateTime. Output: Solar PV output	MLP, SVM, RF, MLnR	RMSE = 1.76 kW
Visser et al. [23]	2023	Short-term	(Pr) Input: 26 variables (absolute/relative air mass, clear sky, direct and diffuse irradiance, etc.). Output: Solar PV output	RF, MLnR	RMSE = 0.13 kW; MAE = 0.65 kW
Rahman et al. [24]	2023	Short-term	(Pr) Input: solar irradiance, module temperature. Output: Solar PV output	LSTM	RMSE = 1 kW; MAE = 0.16 kW; MAPE = 1.93%; R2 = 1
Poti et al. [25]	2023	Short-term	(Pr) Input: solar irradiance, cell temperature. Output: Solar PV output	Proposed new predictor formula	RMSE = 0.43 kW; MAE = 0.25 kW; R2 = 1
Jeong [26]	2023	Short-term	(Pr) Input/Output: Time series Solar PV output	Transformer, RNN, GRU, LSTM	MSE = 0.083; MAE = 0.15
Dimd et al. [21]	2023	Very short-term	(Pr) Input: solar irradiance, temperature, tilt angle, azimuth angle. Output: Solar PV output	LSTM	RMSE = 2.24 kW; WAPE = 4.66%
Dhaked et al. [27]	2023	Short-term	(Pr) Input: solar irradiance, temperature, humidity. Output: Solar PV output	LSTM, MLP	RMSPE = 4.7%
Alrashidi & Rahman [28]	2023	Short-term	(Pr) Input: DateTime (Month, Date, hour), temperature, wind (direction, speed), solar irradiance (direct, global), pressure. Output: Solar PV output	BPNN, SVR	RMSE = 4.84 kW; nRMSE = 4.69%; MAE = 3.06 kW; nMAE = 2.96%
Chodakowska et al. [31]	2023	Medium- to long-term	(Pu) Input/Output: Time series solar irradiation	ARIMA	MSE = 183.18; RMSE = 13.53; MAPE = 2.79%; Std. Error = 14.14; R2 = 99.9%
Tanoto et al. [33]	2023	Medium- to long-term	(Pu) Input: solar irradiance (direct & diffuse), ambient temperature, PV power output. Output: Solar PV output	ARIMAX, ARIMA, SARIMA	RMSE = 9.21 kW; MAE = 2.52 kW; R2 = 0.41
Fan et al. [36]	2022	Short-term	(Pr) Input/Output: Time series Solar PV output	ARIMA-BPNN-SVR	MAE = 0.53; MSE = 0.41; RMSE = 0.64; MAPE = 0.84
Kazem et al. [34]	2022	Long-term	(Pr) Input: solar irradiance, temperature. Output: Solar PV power and current output	PCA, Full-RNN	MSE = 0.077; NMSE = 0.442; R2 = 0.762
Rodríguez et al. [22]	2021	Very short-term	(Pr) Input: season, time of day, solar irradiance. Output: (predicted) Solar irradiance	FFNN, RNN, SVM, FFNN spatiotemporal	RMSE = 6.08 W/m^2
Jung et al. [35]	2020	Long-term	(Pr) Input: solar irradiation, temperature, humidity, wind speed, precipitation, cloud amount, duration of sunshine. Output: Solar PV output	LSTM-RNN	nRMSE = 7.416%; RMSE = 14.003; MAPE = 10.81%

[1] Pr = Private dataset; Pu = Published dataset

[2] The first method in bold is the best method

[3] nRMSE: normalised RMSE; MAPE: Mean Absolute Percentage Error; WAPE: Weighted Absolute Percentage Error; RMSPE: Root Mean Squared Percentage Error; nMAE = normalised MAE; MSE: Mean squared error

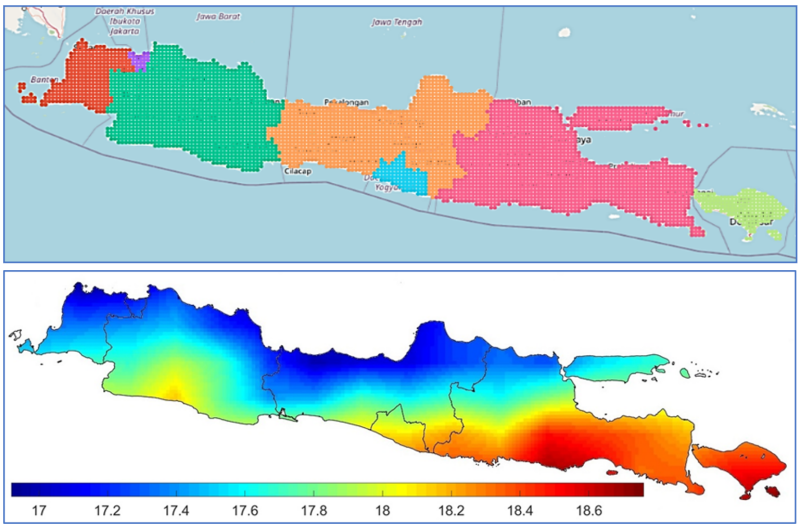

Figure 1: *(Above) Location coordinates of a spatial resolution of 0.05° x 0.05° across the Java-Bali region, Indonesia, and (Below) mapping of 1-year 1 MW modelled solar PV capacity factor in all Java-Bali areas in 2015*

Where n is the number of inputs of the hidden layer's neuron j; w_{ij} is the weight of input i to the hidden layer's neuron j; y_j is the output of the neuron j in the hidden layer; x_{jk} is the input of the neuron k of the output layer from output y_j; w_{jk} is the weight of the hidden layer's neuron j to the output layer's neuron k; m is the inputs number of neuron k in the output layer. The classical activation function of MLP is the Sigmoid function or Tanh, but lately, Rectified Linear Unit (ReLU) and Softmax functions are commonly used.

The backpropagation phase begins directly after the Feed-forward finishes. Firstly, this phase calculates the gradient error δ_k of the output layer's neuron k, then uses the gradient error to update the weights of the output layer and hidden layer neurons. The formulas for the Backpropagation phase can be seen in equations 3 to 6.

$$\delta_k(p) = y_k(p) \cdot [1 - y_k(p)] \cdot (y_{d,k} - y_k(p)) \tag{3}$$

$$w_{jk}(p+1) = w_{jk}(p) \cdot \alpha \cdot y_j(p) \cdot \delta_k(p) \tag{4}$$

$$\delta_j(p) = y_j(p) \cdot \left[1 - y_j(p)\right] \cdot \sum_{k=1}^{l} \delta_k(p) \cdot w_{jk}(p) \tag{5}$$

$$w_{ij}(p+1) = w_{ij}(p) \cdot \alpha \cdot x_i(p) \cdot \delta_j(p) \tag{6}$$

Where $y_{d,k}$ is the target/real output from the dataset; α is the learning rate, a small number from 0 to 1; δ_j is the gradient error of the hidden layer's neuron j; and l is the number of output layer's neurons that get input from the hidden layer's neurons. These two phases are iterated alternately for all the data in the training set until the selected error criterion is satisfied.

The RF ensembles multiple Decision Tree Regressors (DTR) and merges their results to improve accuracy and reduce overfitting. The model implements Bootstrap Sampling, where it randomly selects subsets of data with replacement. Each data subset is used to build a DTR using a random subset of features at each split. The common split of DTR uses the MSE, with the formula in

equation 7. The result of RF regression prediction y is the average of outputs from all DTR [18] (see equation 8 for the formula).

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

$$y = \frac{1}{B}\sum_{b=1}^{B} h_b(x) \tag{8}$$

Where y_i is the i-th observed/target value; $\hat{y}_l$ is the i-th predicted value; n is the number of data points; $h_b(x)$ is the prediction from the i-th DTR for input x; B is the number of DTR.

The HGB is an advanced and efficient implementation of Gradient-boosted Decision Trees (GBDT)[3], designed to handle large datasets more quickly and with lower memory usage. It works by discretising continuous input features into a fixed number of bins, essentially converting them into histograms. This binning significantly reduces the number of split points the algorithm needs to evaluate during training, which results in a major speed-up compared to traditional GBDT methods. In HGB, each iteration adds a new DT that tries to correct the errors made by the previous ensemble of DTs. To do this, gradient descent is used, where the new DT is trained to predict the negative gradients (residuals) of the loss function with respect to the model's current predictions. The GBTD aims to minimize a loss function $L(y, F(x))$, where y is the true target and $F(x)$ is the predicted value. The model $F_M(x)$ comprises M additive functions [43], as seen in equation 9.

$$F_M(x) = \sum_{m=1}^{M} \alpha h_m(x) \tag{1}$$

Where $h_m(x)$ is the m-th base learner (e.g., DT), and α is the learning rate.

The MLnR is a fundamental statistical technique that models the relationship between one dependent and two independent variables. It extends simple linear regression, which involves only one predictor, by allowing for multiple predictors. The formula to predict output y can be seen in equation 10 [44].

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon \tag{2}$$

Where $x_1, x_2, ..., x_n$ are independent variables/features; β_0 is the intercept (constant term); $\beta_1, \beta_2, ..., \beta_n$ are the coefficients of the predictors; ϵ is the error term (captures noise or unexplained variation).

The GS method is used to optimise all ML and MLnR and tested on a comparison framework modified from previous research [26, 35]. The GS technique thoroughly searches a manually specified subset of hyperparameter values, testing each combination to determine the best settings for the model's performance. The SSCV method is used to assess the performance of model candidates, as it offers flexibility by allowing random shuffling of data and customizable numbers of training and testing splits. All models are trained and tested with K-fold SSCV from scikit-learn to avoid overfitting.

The SSCV, also known as Monte Carlo cross-validation, randomly splits the dataset into several training and validation sets. Unlike k-fold cross-validation, which splits the dataset into fixed K-fold, SSCV makes K random splits. The number of iterations, K, can vary based on the analysis being conducted. The results of each split are then averaged. Additionally, the proportion of

[3]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html

training and validation splits is not determined by the number of partitions. The visualisation of SSCV can be seen in Figure 2. Because the split process is combined with data shuffle, the SSCV is regarded as more equitable than the traditional K-fold cross-validation (CV). As a result, K-fold SSCV could reduce overfitting more than K-fold CV and provide more accurate measurements. The chosen trained model is saved for use in the subsequent section after the comparison.

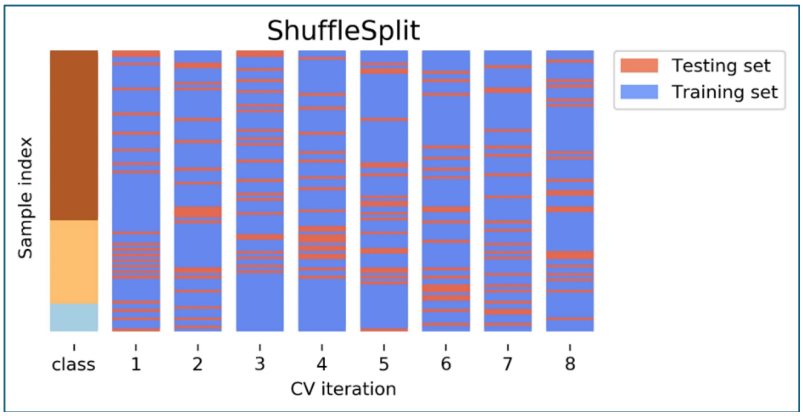

Figure 2: *Example visualisation of SSCV (8-fold)*

This study develops the Solar PV power output prediction model – inspired by the previous research [6] – which consists of two sections. The first section is named Model Comparison and Selection, and the second is Deployment. The first section is a comparison platform for training and testing all considered regressors as potential Solar PV power output predictor candidates. The flow diagrams of the Model Comparison and Selection section and the Deployment section are presented in Figure 3 and Figure 4, respectively.

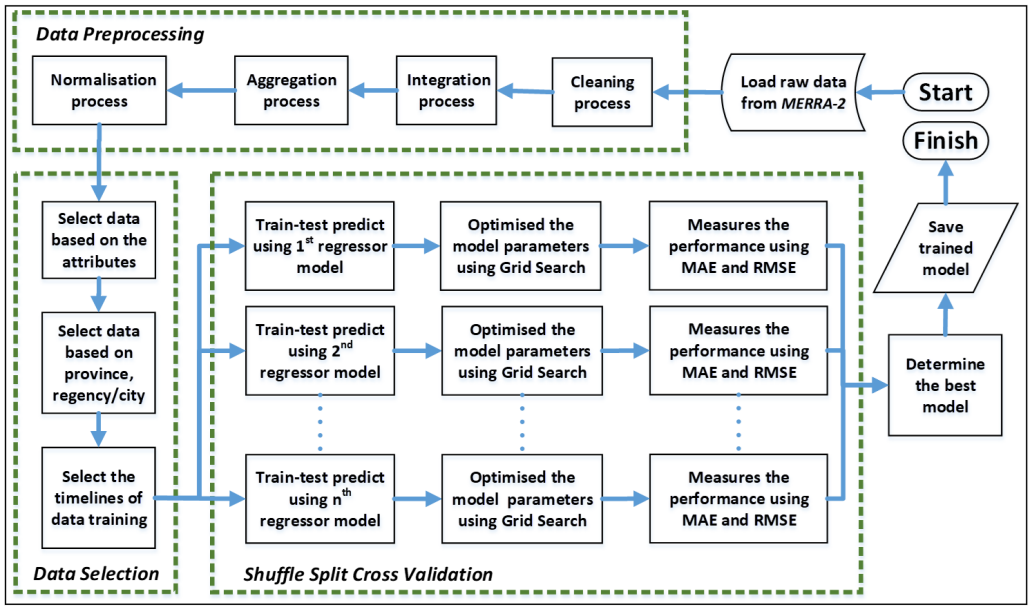

Figure 3: *Model Comparison and Selection section*

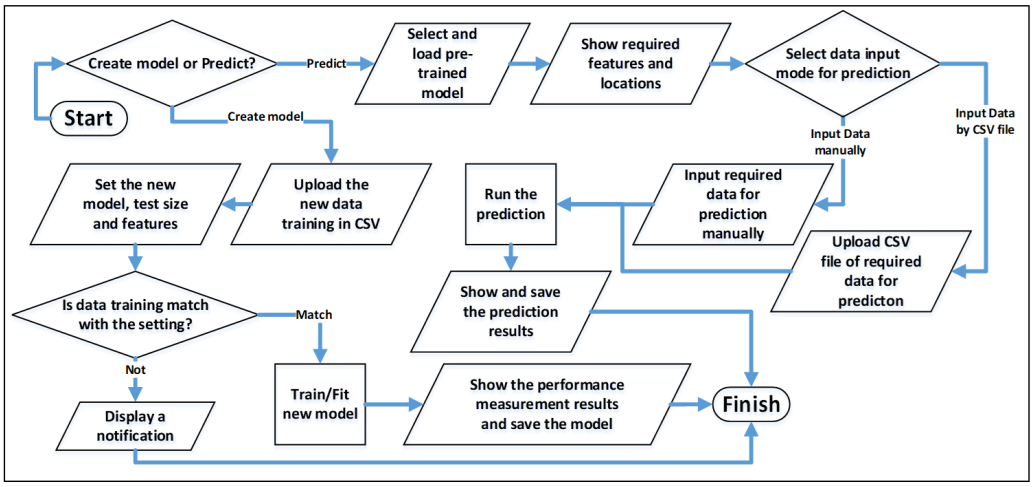

Figure 4: *The deployment section*

The subsequent phase in this first section, Data Selection, minimises the volume of processed data to facilitate processing with constrained computer resources. Consequently, data training concentrates on a certain province or city to ensure that the model addresses the requirements of distinct features and locales. Consequently, the initial task in this phase is to choose the qualities for input: Direct, Diffuse, Temperature, or a mix of two or all three features. Subsequently, we select the dataset according to province, regency, and city. The concluding stage is to choose the dataset according to time intervals (in years).

The Deployment section (flow diagram shown in Figure 4) is divided into two parts, each directed by a condition. The first step involves creating a new model with updated data in CSV format. The new model can be specified here, along with the test size and input features/attributes used in the model training process. If the new data attributes match the input feature settings, the model will start the training. On the other hand, if the new data attributes do not match, the model will generate a notification and terminate. Once the training process is completed, the trained model and its performance measurements for MAE, MSE, RMSE and R^2 formulas[4] will be saved. The formulas of these measurements can be seen in equations 11 to 13.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{5}$$

Where y_i is the i-th observed/target value; $\hat{y}_l$ is the i-th predicted value; $\bar{y}_i$ is the average of all y observed/target values; n is the number of data points.

In the second part of the Deployment section, the new solar PV data can be entered for prediction. The first step of this particular part is to select and load the desired model. After the model has been loaded, its information is displayed, including whether it is only for specific

[4]https://scikit-learn.org/stable/api/sklearn.metrics.html

features (e.g., Diffuse only or Direct-Diffuse only) and locations, e.g., Bali province only and East Java provinces. This information is critical when selecting input data by CSV file mode because the CSV file with the data structure that the model accepts must be synchronised. The solar PV power output prediction model also accommodates a manual mode of inputting data, which is manually entered and recorded directly in the system.

All records with null/zero attributes on the Direct, Diffuse, and Output tables are removed during the raw data cleaning process. Zero/null values are typically present because it was nighttime (no solar radiation) or due to an error in equipment. The raw data tables, Direct, Diffuse, Temperature, and solar PV Output tables, are then integrated using date (rows) and locations (columns). While being integrated, each record is aggregated and written to a new Table, the solar PV dataset, which has the structure shown in Table 2. For this record, this study uses the Reverse Geocoding API to extract information about the province and city/regency from the location data (Latitude-Longitude). The final step in pre-processing is the Normalization Step. We use the Min-Max Scaler method by Scikit-learn to normalize the Direct, Diffuse, and Temperature attribute values.

Table 2: *Solar PV dataset structure*

Attribute	Data type	Description
Date (GMT+7)	DateTime	Converted from the Date attribute of the raw data to GMT+7.
Latitude & Longitude	Spatial	The representation of a location on the earth. This attribute is from the Latitude-Longitude attribute in all raw datasets.
Regency/city	Text	City or regency of a particular Latitude-Longitude that is converted using Reverse Geocoding API.
Province	Text	City or regency of a particular Latitude-Longitude that is converted using Reverse Geocoding API.
Direct (W/m^2)	Number	A value from the "Direct" raw data table associated with a particular date and Latitude-Longitude.
Diffuse (W/m^2)	Number	A value from the "Diffuse" raw data table associated with a particular date and Latitude-Longitude.
Temperature (°C)	Number	A value from the "Temperature" raw data table associated with a particular date and Latitude-Longitude.
Output (kW)	Number	A value from the "Solar PV_Output" raw data table associated.

4. Experimental Results and Discussions

4.1. Is grid search useful?

Experiments in this subsection are designed to evaluate how effective GS is at improving the performance of regressor models. This study applies 410,260 records from the Central Java region's solar PV dataset in 2022 as a case study. The structure of this data can be seen in Table 2. Here, we used "Regency/City", "Province", "Direct", "Diffuse", and "Temperature" attributes as the input and "Output" attribute as labels/targets. All non-numerical attributes will be transformed into numeric values. After that, all the used attributes will be normalised using a MinMax Scaler[5] to be 0 to 1 and considered as input vectors to evaluate the model candidates. The formula of MinMax Scaler can be seen in equation 14.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{6}$$

Where x' is the scaled feature, x is the data, $min(x)$ and $max(x)$ are the range of the feature.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

For analysis purposes, this study aggregates the hourly temporal-based data to obtain daily averaged data and assigns a location with the highest capacity factor to represent each city or regency in the province. The RMSE is measured using 5-fold SSCV. This means that the SSCV will be iterated five times, and for each iteration, 20% of the dataset will be randomly selected for the testing set, while the remaining portion will be used to train the model.

Figures 4 and 5 show the performance comparison between the default settings of the regressor candidates, as specified by the Scikit-Learn library [41] and their performance after optimisation via the GS, and a comparison of processing times, respectively. As shown in Figure 5, GS significantly improved the HGB's performance while slightly improving the MLPs (the RMSE is reduced by 0.13 kW). In the MLnR, the GS result is identical to the default parameters. However, the default parameter setting remains the best for the RF. Meanwhile, Table 3 shows the GS-optimised parameter results for regressor model candidates.

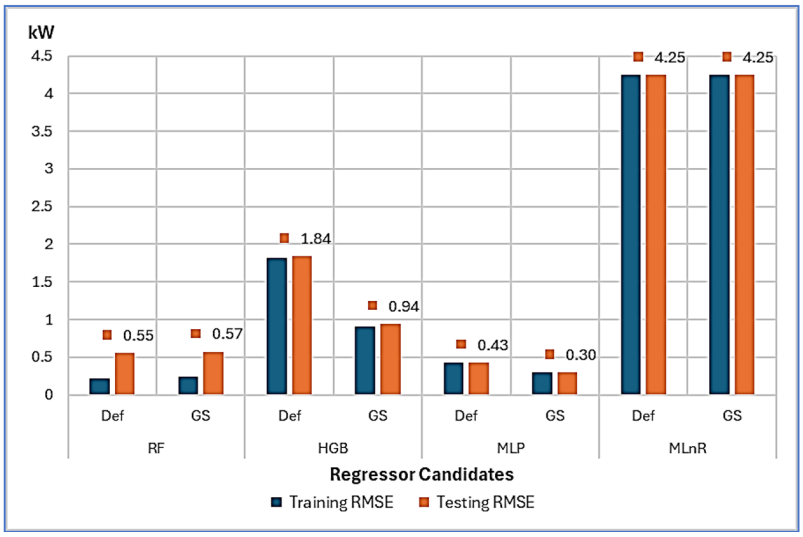

Figure 5: *Performances (RMSE in kW) of regressor models in default vs GS-optimized parameters*

Table 3: *The GS-optimized parameters of regressor model candidates*

Model	GS-optimized parameters
GS(RF)	N_estimator = 40; max_depth = 20; max_features = auto; min_samples_leaf = 1; and min_samples_split = 2.
GS(HGB)	Max_depth = 10; max_iter = 1000; learning_rate = 0.1; min_samples_leaf = 20; loss = 'squared_error'.
GS(MLP)	Max_iter = 200; activation = 'tanh'; solver = 'adam'; learning_rate = 'invscaling'; hidden_layer_sizes = (100,) ⇒ one hidden layer with 100 neurons.
GS(MLnR)	Fit_intercept = True; positive = False (these parameters are the same as the default parameters of Scikit-learn's MLnR).

This study incorporates the second-best configuration identified by the GS process due to computational memory constraints. The GS-optimised RF parameters yielded a marginally higher RMSE, increasing by 0.02 kW. Nonetheless, as illustrated in Figure 6, GS could markedly decrease the processing time in RF, achieving a reduction of 474.41 seconds. The processing time of MLP could potentially be diminished to 1,360.02 seconds. Conversely, the GS-optimised HGB necessitated a longer processing duration than the default version (81.29 seconds). The MLnR required a minimal processing time of 2.1 seconds. A thorough examination of the performance of

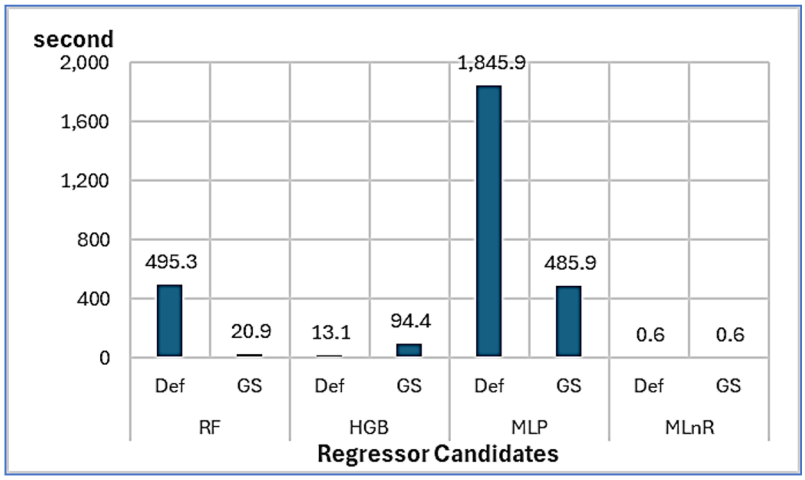

Figure 6: *Processing time (in second) of regressor models in default vs GS-optimized parameters*

regressor model candidates shows that, except for the MLnR, regressor models perform marginally better on training data than the MLnR, and their performance on training data is slightly better than on testing data, as illustrated in Figure 5.

Training data has been utilised to develop the models, while testing data has not. Nevertheless, due to the negligible differences (under 0.5 kW), we determined that none of the models exhibited overfitting. Moreover, the GS-optimised MLP surpassed the others in the testing data, achieving an RMSE of 0.3 kW. The default RF parameters for testing data surpassed the GS-optimized parameters in RMSE, recording values of 0.552 kW and 0.573 kW, respectively. The GS-optimized HGB RMSE was 0.944 kW, whereas the MLnR RMSE was 4.245 kW. Moreover, the GS-optimized MLP surpassed the others in the testing data, achieving an RMSE of 0.3 kW. The default configuration of the MLP regressor surpasses other regressors, even following optimization through the GS process. The model produced an RMSE of 0.43 kW. The R^2 of all models is 0.99, which means all the models are good for use in solar PV output prediction.

4.2. Training and testing for the whole big dataset

The performance of GS experiments is evaluated over various prediction horizons, such as short-, medium-, and long-term, by utilizing a daily Solar PV dataset from 2013 to 2022, as outlined in [20]. Two sets of experiments were implemented for each prediction horizon. The first set is situated in the middle of the prediction horizon range. For instance, if the short-term range is from hours to days, one day is approximately central to this range. The second set is located at the upper end of the range (six days) for the short term, as the medium term commences after one week (7 days). The solar PV dataset range utilised in the experiments is presented in Table 4.

Table 4: *Solar PV dataset range for experimenting on each prediction horizon*

Prediction Horizon	Duration of Prediction (daily)	Data Training/Testing Range for 10-Fold SSCV
Short-term	1 day	22 December 2022 – 31 December 2022
	6 days	2 November 2022 – 31 December 2022
	15 days	1 August 2022 – 28 December 2022
Medium-term	30/31 days (1 month)	1 March 2022 – 31 December 2022
	182/183 days (6 months)	1 January 2022 – 31 December 2022
Long-term	365 days (1 year)	1 January 2022 – 31 December 2022

To evaluate the performance of the GS-optimized results in Table 3 on this large dataset, this study trains the model candidates using 10-fold SSCV on the Solar PV dataset, as 10-fold is considered a better measurement than 5-fold for big data. This study uses two measurements: MAE and RMSE. This study includes default settings whenever possible, especially for the RF, but if a memory error occurs during the process, this study only provides the GS(RF) results. The memory error may occur due to the default RF configuration using 100 decision trees with a maximum depth. Each decision tree will be grown until no more leaves can be split (minimum sample split < 2). When the dataset is large, this setting requires a lot of memory to build the decision trees inside.

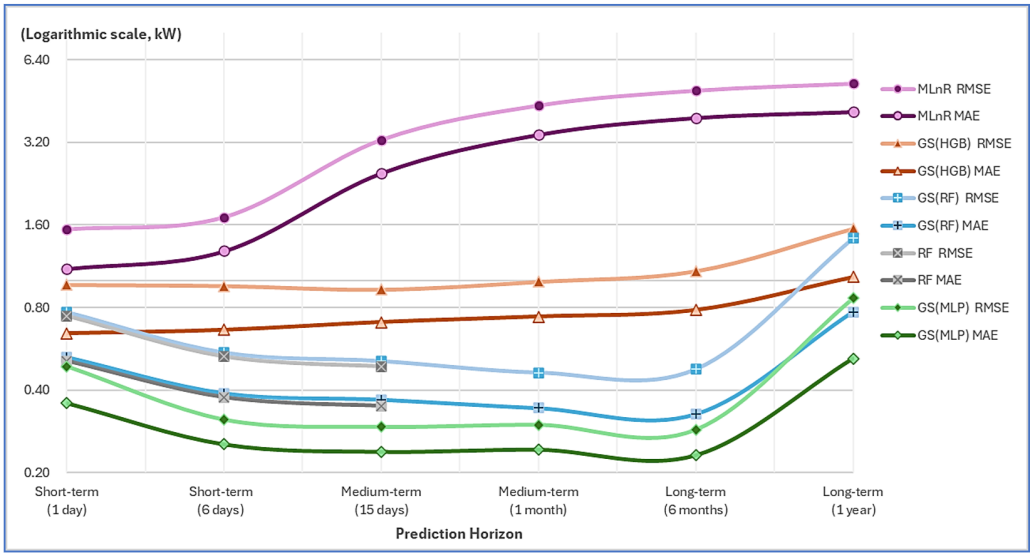

Figure 7: *RMSE and MAE of the regressor candidates across the short-, medium-, and long-term prediction horizons (data range 2013 to 2022)*

Figure 7 illustrates that GS(MLP) achieves the lowest errors for short-term (6 days), medium-term (6 weeks), and long-term (6 months) prediction horizons, with an RMSE of 0.3 kW and an MAE of 0.24 kW. The MAE of GS(RF) decreases from 0.39 kW to 0.33 kW, while the RMSE ranges from 0.55 kW in the short-term (6 days) to 0.48 kW in the long-term (6 months). Nevertheless, the MLnR and GS(HGB) errors increased in tandem with the extent of data training. Across all prediction horizons, the MLnR exhibited the highest (worst) MAE and RMSE. The MLnR regressor is regarded as weak due to its dependence on a linear equation.

Another drawback is that the MLnR generates a greater number of errors as the total volume of data trained increases. For instance, the RMSEs of MLnR are less than 2 kW in the short term, over 3 kW in the medium term, and approximately 5 kW in the long term. The results of these studies indicate that MLnR is a superior method for data training compared to medium- or long-term predictions, which typically necessitate a greater amount of data to train the model. Nevertheless, the MLnR continues to be the most unfavourable option in all instances.

The other three regressors in the ML method family have more intricate equations and can learn from complex patterns more effectively. The implication is that the RF, HGB, and MLP results outperform MLnR, with almost all MAEs and RMSEs less than 1 kW, except for GS(HGB), over the long-term prediction horizon of approximately 1 kW. Nevertheless, the MAE and RMSE of RF, GS(RF), and GS(MLP) improve as data training increases, in contrast to the MLnR. ML

models are trained in a broader range of data, resulting in more generalised models and improved prediction results, as a result of the increased data training. Nevertheless, the models' performance improves until they reach a specific threshold, at which point they reach a plateau [38].

The errors of RF, GS(RF), and GS(MLP) are greater than those of other prediction horizons when the short-term (1 day) prediction horizon is considered. The absence of data is the reason for the initial hypothesis. Additionally, experiments are implemented to verify the hypothesis and observe the short-term (1-day) prediction horizon. Aside from the short-term (1 day) issue with small data training, as illustrated in Figure 7, a second anomaly occurred in the long-term (1 year) when errors for all model candidates abruptly increased. Regarding technicality, only MLnR is unsuitable for big data processing; therefore, the problem is most likely with the data rather than the models. Consequently, further experiments are implemented to investigate this anomaly. The following experiments employ only the lighter GS(RF), which did not induce computational memory errors, due to the slight difference between RF and GS(RF) (± 0.02 kW).

4.3. Small data training problem in short-term (1 day) prediction horizon

The short-term (1 day) variety of data training for a location is only nine days because this study uses 10-fold SSCV. This results in slightly worse prediction performance for GS(RF) and GS(MLP) than in the other cases. The initial hypothesis is that GS(RF) and GS(MLP) require additional data training. Based on this hypothesis, this study investigated whether total data training can be achieved by conducting experiments with small amounts of data ranging from 3 to 40 days and running them using 3-fold SSCV to 40-fold SSCV. These settings ensure the testing data is always one day old, while the rest is training data. For example, in 3-fold SSCV, the training data is two days; in 40-fold SSCV, the training data is 39 days. Table 5 shows the detailed data ranges for each n-fold SSCV in these experiments. Meanwhile, the results are shown in Figure 8.

Table 5: *The data range of each fold setting for short-term (1 day) prediction horizon*

Fold	Data Time Range	Total Data Training/Testing (in days)
3	29 December – 31 December 2022	2/1
5	27 December – 31 December 2022	4/1
7	25 December – 31 December 2022	6/1
10	22 December – 31 December 2022	9/1
15	17 December – 31 December 2022	14/1
20	12 December – 31 December 2022	19/1
30	2 December – 31 December 2022	29/1
40	22 December – 31 December 2022	39/1

Figure 8 shows that with sufficient data training (5-days), GS(RF), GS(HGB), and GS(MLP) perform better than MLnR, with RMSE and MAE plateauing at ± 1.5 kW and ± 1 kW, respectively. Furthermore, the MAE of GS(RF) and GS(HGB) is already lower than MLnR in the first experiment, where data training lasts two days. It means that, after two days of data training, GS(RF) and GS(HGB) produce fewer errors than MLnR (lower MAE), but they also produce a few significant errors, resulting in a higher RMSE.

The GS(MLP) underfitted after two days of data training, a situation in which the model's performance suffers due to insufficient data training or training epochs (repetitions). As a result, this study includes GS(MLP) experiments with two days of data training, increasing the number of training epochs from 300 to 2,000. Figure 9 shows the results of MAE, RMSE, and processing times of GS(MLP) with training epoch 200 to 2000 for a short-term (1-day) prediction horizon, 3-fold SSCV.

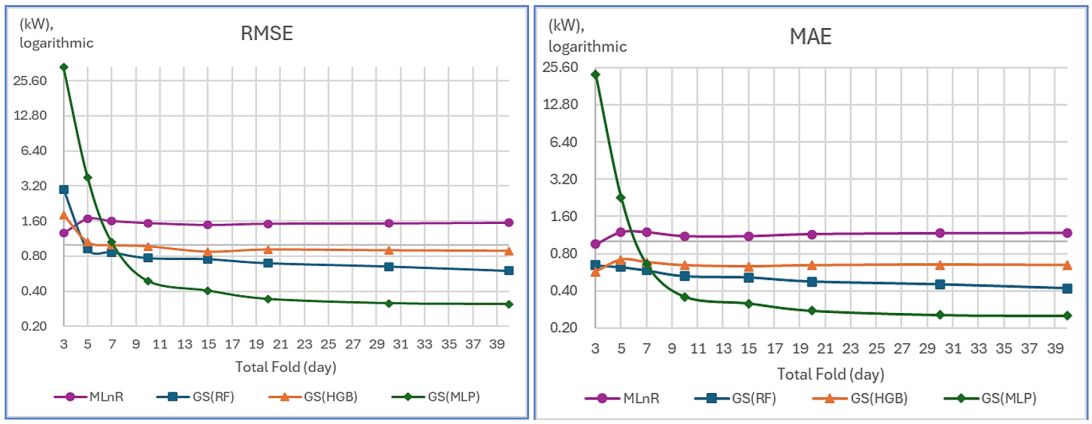

Figure 8: *RMSE (left) and MAE (right) of 3-fold to 40-fold SSCV for short-term (1-day) prediction horizon*

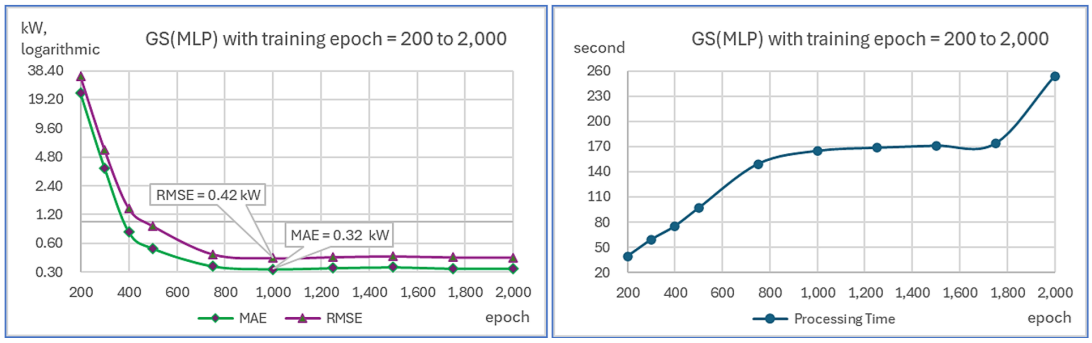

Figure 9: *The results of MAE, RMSE (left) and processing times (right) of GS(MLP) with training epoch 200 to 2000 for short-term (1 day) prediction horizon, 3-fold SSCV*

Figure 9 also shows that adding more training epochs without more data significantly reduced GS(MLP)'s RMSE and MAE. After 1,000 epochs, the GS(MLP) achieved the lowest MAE and RMSE before plateauing. As a result, a maximum of 1,000 epochs is recommended for small data training (i.e., two days) with a short prediction horizon of one day. However, as expected, processing times would increase with each additional epoch. GS(MLP) with 1,000 training epochs produces the lowest error among the model candidates based on 3-fold SSCV (see Figure 9). Given enough epochs to train the model, the GS(MLP) may be the best candidate for short-term (1-day) prediction. However, once the data training is large enough, i.e., ten days, 200 epochs are sufficient and do not cause an underfitting problem.

4.4. What happened in the long term (1 year)?

An anomaly occurs during the long-term (1 year) experiments using the solar PV dataset from 2013 to 2022 (see Figure 8). In these experiments, both MAE and RMSE of GS(HGB), GS(RF), and GS(MLP) deteriorated and increased sharply, outperforming the short-term results (1 day). Investigation of the Solar PV dataset turned up anomalies in the 2015-2016 data. Because weather conditions influence our data, climate change is a plausible explanation for these anomalies. Indonesia's climate is heavily influenced by Indo-Pacific climate modes [45].

After analyzing Indonesian climates from 2005 to 2022 using the Oceanic Nino Index (ONI),

this study found that a strong El Niño occurred between 2015 and 2016, affecting weather in Pacific areas such as Java and Bali. Figure 10 shows the Oceanic Nino Index (ONI) from 2005 to 2022. To conduct a thorough investigation, this study runs experiments for a long-term (1-year) prediction horizon using data from a 10-fold SSCV range from 2011 to 2022 but excludes data from 2015 and 2016. Figure 11 shows RMSE and MAE of the regressor candidates across the short-, medium-, and long-term prediction horizons (data range 2013 to 2022), with long-range data (1 year) without 2015-2016.

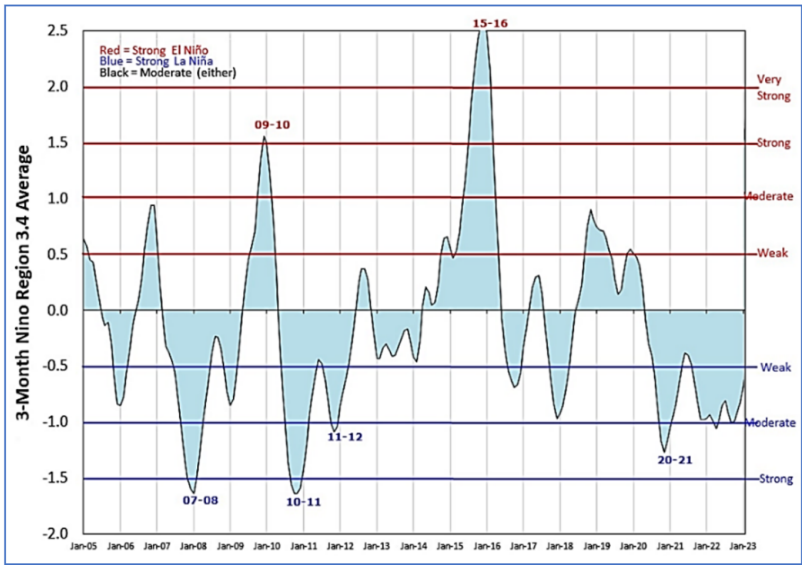

Figure 10: *Oceanic Nino Index (ONI), 2005 to 2022*

Figure 11 shows that without the data affected by a strong El Nino, the MAE and RMSE of GS(HGB) and GS(MLP) do not increase but plateaued as prediction horizons shrank, whereas GS(RF) errors decreased. Only MLnR is unaffected by the anomalies, but its errors are still higher than those of other model candidates trained using anomaly data. As previously stated, the MLnR model is not suitable for training on large datasets.

The best model is GS(MLP), which has an MAE of 0.258 kW and an RMSE of 0.318 kW while being unaffected by robust El Nino data. The GS(RF) is marginally worse, with MAE equal to 0.283 kW and RMSE equal to 0.361 kW. Following that, the GS(HGB) MAE and RMSE were 0.768 kW and 1.017 kW, respectively. Figures 6 and 10 show a comparison of long-term (1 year) with and without strong El Nino-affected data (2015-2016), demonstrating that ML predictor models (RF, HGB, and MLP) are sensitive to robust (very strong) El Nino data.

5. Conclusion and Future Work

Using the Java-Bali region as a case study and several ML techniques, this study shows that the GS-optimised MLP model can accurately predict the solar PV power output across all prediction horizons from short-term (1 day) to long-term (1 year). The Average MAE of GS(MLP) across all prediction horizons is 0.248 kW with a standard deviation of 0.011, while the average RMSE is 0.306 kW with a standard deviation of 0.013. However, when total data training is small, i.e., in a short-term (1 day) prediction horizon, GS(MLP) requires many epochs to train the model, precisely

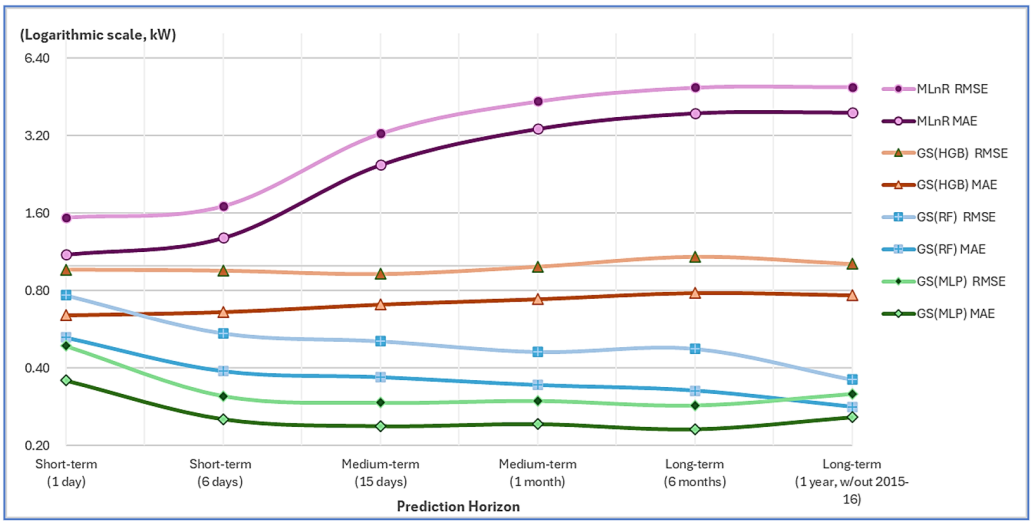

Figure 11: *RMSE and MAE of the regressor candidates across the short, medium, and long-term prediction horizons (data range 2013 to 2022), with long-range data (1 year) without 2015-2016*

1,000 epochs. When data training is sufficient, such as in short-term (6 days) to long-term (1 year) prediction horizons, the GS(MLP) can be trained with only 200 epochs and perform well. GS(RF) is the second-best model, with an average MAE of 0.373 kW, a standard deviation of 0.041, and an average RMSE of 0.521 with a standard deviation of 0.07. The average MAE for the GS(HGB) is 0.718 kW with a standard deviation of 0.049, and the RMSE is 0.992 kW with a standard deviation of 0.059. The MLnR performs poorly, with errors on all prediction horizons greater than 1 kW.

The analytical findings indicate that the machine learning family predictor models (MLP, RF, and HGB) may be susceptible to robust El Niño-induced training data. Future research should focus on identifying alternative prediction models that are resilient to data influenced by severe El Niño events and evaluating the performance of deep learning-based models. Additional analysis of the solar PV power output predictions, which integrate socioeconomic and electrical demand data specific to the region, is also interesting.

Acknowledgements: This work was supported by the Competitive Fundamental Research Scheme 2024 provided by The Directorate General of Higher Education, Research, and Technology (DGHERT) of the Ministry of Education, Culture, Research, and Technology (MOECRT) of the Republic of Indonesia, under contract No. 109/E5/PG.02.00.PL/2024 (25/SP2H/PT/LPPM-UKP/2024).

Declaration of interest: The authors declare no conflicts of interest.

References

[1] Lo K. Asian energy challenges in the Asian century. *Journal of Asian Energy Studies* 2017:1:1–6.
[2] Scott C, Ahsan M, Albarbar A. Machine learning for forecasting a photovoltaic (PV) generation system. *Energy* 2023:278:127806.
[3] Ahmed R, Sreeram V, Mishra Y, Arif MD. A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization. *Renewable and Sustainable Energy*

Reviews 2020:124:109792.

[4] Obuseh E, Eyenubo J, Alele J, Okpare A, Oghogho I. A systematic review of barriers to renewable energy integration and adoption. *Journal of Asian Energy Studies* 2025:9:26-45.

[5] Nguyen TN, Müsgens F. What drives the accuracy of PV output forecasts? *Applied Energy* 2022:323:119603.

[6] Tanoto Y, Budhi GS, Mingardi SF. Clustering-based assessment of solar irradiation and temperature attributes for PV power generation site selection: A case of Indonesia's Java-Bali region. *International Journal of Renewable Energy Development* 2024:13(2):351-361.

[7] IRENA. Future of Solar Photovoltaic: Deployment, investment, technology, grid integration and socio-economic aspects (A Global Energy Transformation: paper). Abu Dhabi, International Renewable Energy Agency, 2019, p. 1-73.

[8] Andrews-Speed P, Zhang S. China as a low-carbon energy leader: Successes and limitations. *Journal of Asian Energy Studies* 2018:2(1):1-9.

[9] Ledmaoui Y, El Maghraoui A, El Aroussi M, Saadane R, Chebak A, Chehri A. Forecasting solar energy production: A comparative study of machine learning algorithms. *Energy Reports* 2023:10:1004-1012.

[10] IRENA-ACE. Renewable energy outlook for ASEAN: Towards a regional energy transition. International Renewable Energy Agency, Abu Dhabi; and ASEAN Centre for Energy, Jakarta, 2022.

[11] Pfenninger S, Staffell I. Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data. *Energy* 2016:114:1251-1265.

[12] Scarpa F, Marchitto A, Tagliafico L. Splitting the solar radiation in direct and diffuse components; insights and constrains on the clearness-diffuse fraction representation. *International Journal of Heat and Technology* 2017:35(2):325-329.

[13] Huang M. Two phase change material with different closed shape fins in building integrated photovoltaic system temperature regulation. World Renewable Energy Congress-Sweden. 2011.

[14] Zhao J, Li Z, Ma T. Performance analysis of a photovoltaic panel integrated with phase change material. *Energy Procedia* 2019:158:1093-1098.

[15] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, 1986, p. 318-362.

[16] Kingma DP, Ba J. Adam: A method for stochastic optimization. International Conference on Learning Representations. San Diego, US. 2015.

[17] Ke G, Meng Q, Finley T, Wang T, Chen W, et al. LightGBM: A highly efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30 (NIPS 2017). Long Beach, CA, USA. 2017.

[18] Breiman L. Random forests. *Machine Learning* 2001:45(1):5-32.

[19] Uyanık GK, Güler N. A study on multiple linear regression analysis. *Procedia - Social and Behavioral Sciences* 2013:106:234-240.

[20] Iheanetu KJ. Solar photovoltaic power forecasting: A review. *Sustainability* 2022:14(24):17005.

[21] Dimd BD, Völler S, Midtgård O-M, Sevault A. The effect of mixed orientation on the accuracy of a forecast model for building integrated photovoltaic systems. *Energy Reports* 2023:9:202-207.

[22] Rodríguez F, Martín F, Fontán L, Galarza A. Ensemble of machine learning and spatiotemporal parameters to forecast very short-term solar irradiation to compute photovoltaic generators' output power. *Energy* 2021:229:120647.

[23] Visser L, AlSkaif T, Hu J, Louwen A, van Sark W. On the value of expert knowledge in estimation and forecasting of solar photovoltaic power generation. *Solar Energy* 2023:251:86-105.
[24] Rahman NHA, Hussin MZ, Sulaiman SI, Hairuddin MA, Saat EHM. Univariate and multivariate short-term solar power forecasting of 25MWac Pasir Gudang utility-scale photovoltaic system using LSTM approach. *Energy Reports* 2023:9:387-393.
[25] Poti KD, Naidoo RM, Mbungu NT, Bansal RC. Intelligent solar photovoltaic power forecasting. *Energy Reports* 2023:9:343-352.
[26] Jeong H. Predicting the Output of Solar Photovoltaic Panels in the Absence of Weather Data Using Only the Power Output of the Neighbouring Sites. *Sensors* 2023:23(7):3399.
[27] Dhaked DK, Dadhich S, Birla D. Power output forecasting of solar photovoltaic plant using LSTM. *Green Energy and Intelligent Transportation* 2023:2(5):100113.
[28] Alrashidi M, Rahman S. Short-term photovoltaic power production forecasting based on novel hybrid data-driven models. *Journal of Big Data* 2023:10(1):26.
[29] Lee DS, Lai CW, Fu SK. A short- and medium-term forecasting model for roof PV systems with data pre-processing. *Heliyon* 2024:10(6):e27752.
[30] Cui C, Wu H, Jiang X, Jing L. Short- and medium-term forecasting of distributed PV output in plateau regions based on a hybrid MLP-FGWO-PSO approach. *Energy Reports* 2024:11:2685-2691.
[31] Chodakowska E, Nazarko J, Nazarko Ł, Rabayah HS, Abendeh RM, Alawneh R. ARIMA models in solar radiation forecasting in different geographic locations. *Energies* 2023:16(13):5029.
[32] Asiedu ST, Nyarko FKA, Boahen S, Effah FB, Asaaga BA. Machine learning forecasting of solar PV production using single and hybrid models over different time horizons. *Heliyon* 2024:10(7):e28898.
[33] Tanoto Y, Budhi GS, Widjaya JC. Time Series Forecasting for Daily to Monthly Temporal Hourly-based Solar PV Output Power. 2023 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). 2023.
[34] Kazem HA, Yousif JH, Chaichan MT, Al-Waeli AHA, Sopian K. Long-term power forecasting using FRNN and PCA models for calculating output parameters in solar photovoltaic generation. *Heliyon* 2022:8(1):e08803.
[35] Jung Y, Jung J, Kim B, Han S. Long short-term memory recurrent neural network for modeling temporal patterns in long-term power forecasting for solar PV facilities: Case study of South Korea. *Journal of Cleaner Production* 2020:250:119476.
[36] Fan G-F, Wei H-Z, Chen M-Y, Hong W-C. Photovoltaic Power Generation Forecasting Based on the ARIMA-BPNN-SVR Model. *Global Journal of Energy Technology Research Updates* 2022:9:18-38.
[37] Gelaro R, McCarty W, Suárez MJ, Todling R, Molod A, Takacs L, Randles CA, Darmenov A, Bosilovich MG, Reichle R, Wargan K. The modern-era retrospective analysis for research and applications, version 2 (MERRA-2). *Journal of climate* 2017:30(14):5419-5454.
[38] Budhi GS, Chiong R, Pranata I, Hu Z. Using machine learning to predict the sentiment of online reviews: A new framework for comparative analysis. *Archives of Computational Methods in Engineering* 2021:28:2543–2566.
[39] Budhi GS, Chiong R, Wang Z, Dhakal S. Using a hybrid content-based and behaviour-based featuring approach in a parallel environment to detect fake reviews. *Electronic Commerce Research and Applications* 2021:47:101048.
[40] Tanoto Y, Macgill I, Bruce A, Haghdadi N. Photovoltaic Deployment Experience and Technical Potential in Indonesia's Java-Madura-Bali Electricity Grid. The 2017 Asia Pacific Solar Research Conference (APSRC). Melbourne, Australia. 2017.

[41] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 2011:12:2825-2830.
[42] Negnevitsky M. Artificial neural networks. Artificial Intelligence: A Guide to Intelligent Systems (2nd Edition). England, Addison-Wesley, 2005.
[43] Friedman JH. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 2001:29(5):1189-1232.
[44] Montgomery DC, Peck EA, Vinin GG. Multiple Regression Models. Introduction To Linear Regression Analysis 5th edition. New Jersey, US, John Wiley & Sons, 2012.
[45] Iskandar I, Lestrai DO, Nur M. Impact of El Niño and El Niño Modoki Events on Indonesian Rainfall. *Makara Journal of Science* 2019:23:217-222.