VOL. 19, NO. 22, NOVEMBER 2024 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

FORECASTING SOLAR POWER GENERATION WITH MACHINE LEARNING TECHNIQUES

K. Muthuvel[1], P. Muthukumar[2] and Thomas Thangam[3]
[1]Department of Electrical and Electronics Engineering, Noorul Islam Centre for Higher Education, Kumaracoil, India
[2]Department of Electrical and Electronics Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, India
[3]Department of Process Engineering, International Maritime College of Oman, National University of Science and Technology, Sohar, Sultanate of Oman
E-Mail: er.muthuvel@gmail.com

ABSTRACT

This study underscores the economic and environmental advantages of integrating solar energy into power systems. The unpredictable nature of solar power poses challenges to system operation and planning. To ensure the economic sustainability of newly constructed systems, precise forecasting of Photovoltaic (PV) system effectiveness and energy output is crucial. Addressing variations in solar power consumption, this work presents an enhanced Machine Learning (ML) model. Utilizing Python, the study explores Linear Regressor, Random Forest (RF) Regressor, XGBoost Regression, K-Nearest Neighbor (KNN) Regressor, and AdaBoost Regressor approaches, all proving effective in predicting electricity production. Results highlight the superior performance of ML algorithms over traditional time series methods and two baseline models, emphasizing their efficacy in solar power forecasting.

Keywords: PV system, time-series, machine learning, XGBoost regressor, adaboost regressor.

Manuscript Received 19 September 2024; Revised 24 November 2024; Published 31 December 2024

1. INTRODUCTION

The growth of Renewable Energy Sources (RES) has recently become the focus of international cooperation in combating global warming. Photovoltaic (PV) energy, along with wind energy, is one of the most well-liked forms of RES since it is sustainable, unrestricted, and affordable [1]. PV is one of the green energies that will be prioritized in the future and is expanding vigorously on a wide scale. To keep the power system stable, it is necessary to solve challenging situations including the intermittent nature of PV power and forecasting uncertainty. Power system security will be extremely difficult to maintain if PV output cannot be precisely predicted [2-5].

Precise prediction and forecasting production of solar electricity in PV systems are unquestionably required for better coordination and incorporation into energy storage. Particularly, the output power of a PV system, is strongly related to solar irradiation and temperature variations, which explains that PV system energy generation is intermittent [6]. The variability and intermittent nature of temperature and sun irradiation, in particular, have an impact on solar power output [7]. As a result, forecasting findings for the near-term horizon are utilized to estimate higher PV generation ramp rates. Long-term horizons and forecasting for medium aid in operation optimization and market participation [8]. As a result, solar energy forecasting is tailored to a specific operation, and an appropriate forecasting approach should be chosen [9]. Forecasting techniques effectively play a significant part in resolving the natural cyclicality and uncertainty.

An alternative method for predicting AC PV power directly from historical time series depends on the accuracy of the historical data and does not necessitate a deep understanding of the PV system or the climatic variables involved. Autoregressive models or ML approaches are commonly used to forecast time series PV power [10, 11]. Many forecasting models have been presented in an attempt to produce a practical technique for application in real-world circumstances [12]. These systems are heavily reliant on Artificial Intelligence (AI), complex statistics, and massive volumes of meteorological and topography data. By modelling or simulating future situations, these strategies aim to reduce the possibility of breakdowns within the energy system and predict its reliability [13]. The main disadvantage of statistical approaches is that they are based on the assumption of linear stationary structures, which are unsuitable for non-linear solar radiation. As an outcome, the statistical model's forecast accuracy is time-dependent and dependent on the accuracy of historical data. These techniques are sometimes recognized as black box methods [14].

Time series approaches are the most commonly used statistical models Auto-Regressive (AR) [15] model, Autoregressive Moving Average model, Moving Average, Autoregressive Integrated (ARIMA) [16] technique, Seasonal Autoregressive Integrated (SARIMA) method, and Auto-Regressive Exogenous Input and Exogenous Data Moving Averages (ARMAX) are some commonly used time series analysis models. These approaches perform well for static time series. As the temperature fluctuates often, the accuracy of solar irradiation predicted using the time-series technique drops vividly [17, 18]. As a result, ML models such as the Multi-Layer Perceptron network, Extreme Learning Machine, Support Vector Machine (SVM) [19], Long Short Term Memory (LSTM)

[20], and others are employed. Several variables do not require adjustment. Unfortunately, owing to simplistic network topology, creating a sophisticated mapping relation between affecting elements and sun irradiation is unfeasible. SVM achieves excellent accuracy in prediction with a limited number of samples; however, selecting kernel functions is more challenging. When the database is huge, the running speed falls dramatically. LSTM approach [21, 22] helps to capture long-distance relations between successive samples, nevertheless, parameter optimization is further difficult, inefficient, and heavily relies on knowledge. Convolutional Neural Networks (CNN) [23] and SVM [24] are used for detecting environmental faults in solar panels accurately by training thermal images of solar panels. However, the effectiveness of CNNs and SVMs trained on a specific dataset may not directly transfer to different environmental conditions. Changes in lighting, weather, or panel installations might impact the models' performance, requiring retraining or adaptation to new conditions.

In this work, implementation analysis of different ML techniques [25] like Linear Regression, KNN, RF, AdaBoost [26], and XGBoost 2] techniques are examined to feature their merits and demerits in forecasting solar power production issues. Additionally, the parameters for every model are analyzed and tuned. The originality of this article is important for adopting and replicating prediction approaches for power generation power based on meteorological information. As a result, the suggested methodology demonstrates how solar power generation is forecast using various types of weather data with maximum energy prediction efficiency.

The manuscript is structured as follows: Proposed research description in section II, Results and Discussion in section III, and Conclusion in section IV.

2. PROPOSED RESEARCH DESCRIPTION

Accurate modelling and prediction of solar power generation in PV systems are indeed necessary for better control and integration into energy storage. The output PV system power, in particular, is significantly associated with solar irradiance and climatic conditions, which explains the intermittency of PV system energy production. The fundamental reason for this is that the output differs concerning function, units, and range rate. Although the value varies owing to factors such as seasonality, location, air condition, and time of day, calculating the sun irradiation forecast is simple. Because solar radiation reaching the earth's surface is dignified in power per unit area, solar irradiance at different locations can be compared. Apart from major elements, PV power production is determined by the size and efficiency of solar panels. As a result, the solar panel size is defined during the evaluation of PV power forecasts to acquire the correct PV output information.

Initially, the data from solar power plants and weather stations are collected and preprocessed to obtain suitable input data. After preprocessing cross-validation is performed to attain high performance parameters, which are utilized as input for ML technique. Finally, different ML techniques like linear, KNN, RF, AdaBoost, and XGBoost techniques are analyzed to predict the most suitable method for forecasting solar power.

A. Research Framework

The methodology and data used to assess the ML techniques explored in this study are defined in this subsection. The evaluation framework is graphically represented in Figure-1. Data collection, data preprocessing, and cross-validation (CV) are the three processes that make up the framework. Figure-2 depicts the proposed work's flow diagram.

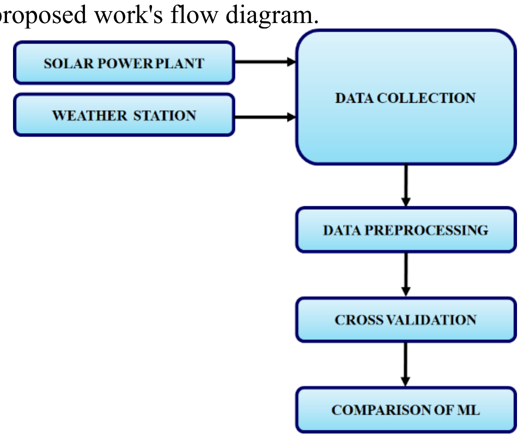

Figure-1. Proposed system for analyzing and comparing ML technique for PV prediction.

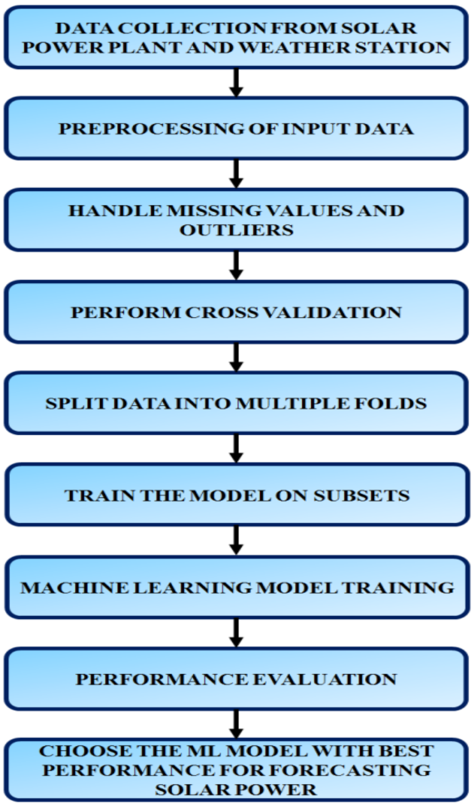

Figure-2. Flow diagram of the proposed work.

ARPN Journal of Engineering and Applied Sciences

B. Data Collection

The solar panel input data is collected from Vijayawada, Andhra Pradesh, India with a latitude position of 16.5075°N. In this study, polycrystalline solar panels are used. Two data sources are gathered during the data collection stage: data on solar power generation and data from weather forecasts. The data were gathered for 10 days that is Table-1 displays the solar power generation data and Table-2 lists the weather forecast data. From the gathered data, 80% is utilized for training and 20% is used for testing.

In addition to the data on solar energy production, weather forecast data is also gathered to use as inputs in ML models. After the collection process is over, preprocessing is carried out to make the data a suitable input.

Table-1. Numerical data of solar power generation.

Date	Generated power (W)
01-01-2021	121
02-01-2021	128
03-01-2021	142
04-01-2021	112
05-01-2021	139
06-01-2021	87
07-01-2021	103
08-01-2021	131
09-01-2021	134
10-01-2021	133

Table-2. Numerical weather forecast data.

Ambient Temp (°C)	Humidity (%)	Pressure (Pa)	Wind Speed (m/s)
28	68	1013	4
29	65	1012	4
28	66	1010	5
29	69	1011	4
29	76	1011	6
29	70	1009	4
29	70	1007	3
29	73	1008	6
29	74	1009	4
31	72	1009	2

C. Data Preprocessing

The pre-processing stage prepares the acquired data for use as inputs to the models. To begin, cases with no sunlight statistics are eliminated from the filtering operation. Following that, in the merging task, three-hour intervals of consecutive instances are combined into a single occurrence. In other circumstances, data from solar energy production and meteorological observation are provided in one-hour increments, and hourly data is combined every three hours to complement weather forecast data. Finally, in the qualitative variables task, all categorical variables are converted to numerous binary variables. Preprocessing ensures that the data is in a format that is suitable for specific requirements of analysis or modeling techniques being employed. It helps in addressing issues such as missing data, temporal misalignments, and incompatible variable types, ultimately leading to more accurate and meaningful results from the models.

D. Cross-Validation

The cross-validation stage is intended to obtain each model's best hyperparameter values. Several performance measures are employed to check the models in this step. First, all data is segmented by successively splitting it into training, validation, and test sets.

Second, the grid search approach is used over the training set and evaluated over the validation set to determine the optimum hyperparameters. The initial comparison of the created models is performed in a test set. The highest-performing models are then submitted as input for the final stage. After, this stage five different machine learning methods are examined to determine an optimal approach for PV prediction.

E. Forecasting Machine Learning Methods

a) Linear regression

The output solar power is a dependent variable in the domain of solar power prediction, and independent variables include elements like weather, time of day, and solar panel parameters. In linear regression, a linear equation is utilized to represent the connection between the predictor and the outcome variable. Finding the coefficient values in the equation that best fit the data is an objective.

A linear regression model utilizes one or more independent and dependent variables to model a linear relationship. When there is just one dependent variable and one independent variable, simple regression is utilized.

In Equation 1, the dependent variable Y is related to the independent variable X.

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

From the expression above, the regression coefficient is denoted asβ_1 andβ_0 as constant and ϵ as an error.

Multivariate-linear regression is used to model dependent variables utilizing several independent variables. Equation (2) determines the linear relationship between various independent factors and dependent variable Y:

$$Y = \beta_0 + \beta_n X_n + \epsilon \quad (2)$$

Here, independent variables are represented by X_n, regression coefficient as β_n, constant is denoted as β_0, and error as ϵ. Figure- 3 indicates the flowchart for Linear Regression.

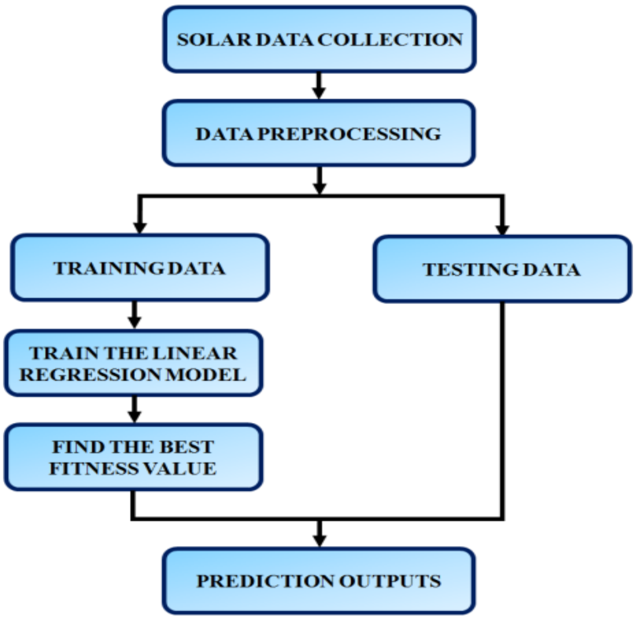

Figure-3. Flowchart for Linear Regression.

The following are some advantages of linear regression: The link concerning a continuous result variable and one or more predictor variables is modelled using linear regression, which is reasonably easy to comprehend and apply. Although it has some benefits, linear regression also has drawbacks. It's possible that the relation between predictor and response variable is not always linear, which is what linear regression assumes. It is susceptible to outliers, which impact the computed coefficients and model-predicted.

b) KNN regressor

A non-parametric technique called KNN shown in Figure-4 is developed as a result of pattern recognition investigation. Since the study of non-linear dynamics has advanced, numerous experts have used KNN to regularly forecast time series. Therefore, the algorithm has a remarkable capability to identify a time series' nonlinear characteristics. The fundamental principle of KNN is that the independent variable in historic remarks is derived based on the similarity between the independent variable of the predictors to obtain the best estimators for the predictor.

To find a set of k past nearest neighbors in past data sets for the current situation, KNN relates a metric to predictors. A kernel function is suggested as a solution to the regression issue, and regression yields the following results.

$$Y_i = f\left(Y_{i(1)}, Y_{i(2)}, \dots, Y_{i(k)}\right) \quad (3)$$

Here, the prediction value is specified as Y_i, the nearest neighbor magnitude of j as $Y_{i\,(j)}$. It has to be noted that i (j=1 to k) represents the arrangement of closest neighbours depending on the distance from the present condition. Accordingly, the similarity between the historical label and predictor depends on distance:

$$r_{i,j} = \sqrt{\sum_{t=1}^{q}\left(d_{jt} - d_{it}\right)^2} \quad (4)$$

Here tth independent variable of Y_i and Y_j is represented as d_{it} and d_{jt} and several independent variables as q. So, the primary problem for forecasting the time series is defining the kernel function of KNN.

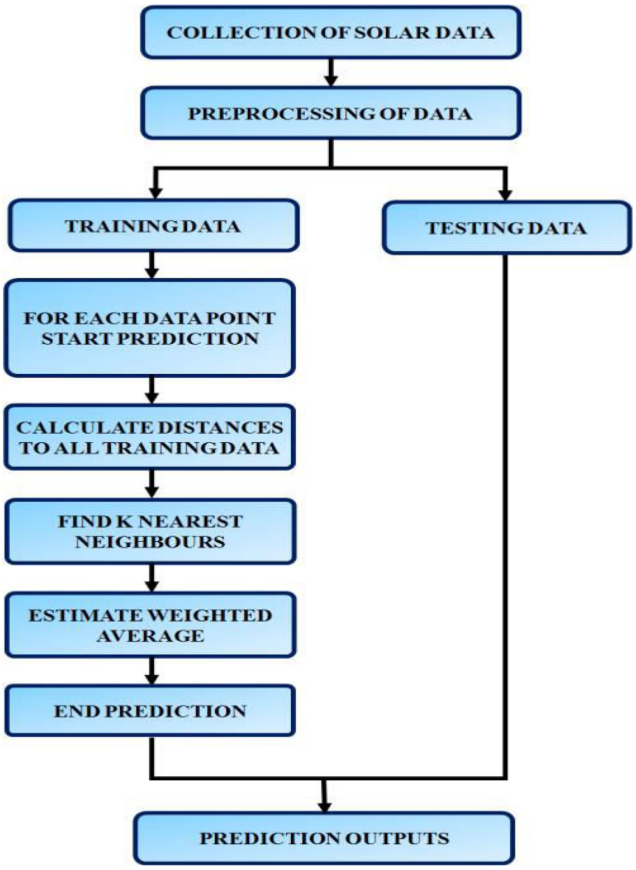

Figure-4. Flowchart for KNN.

c) Random forest regressor

To solve the drawbacks of the conventional Classification and Regression Tree (CART) technique, RF, a tree-based ensemble technique is used. RF is composed of many weak decision tree learners that are created in parallel, as shown in Figure-5, to simultaneously reduce model bias and variance. To train a RF N bootstrapped sample sets are utilized. An unpruned

 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

regression tree is then built by each bootstrapped sample as input. Rather than using every available predictor, only a limited and fixed number of random sample K predictors are selected as split candidates at this stage. The flowchart for RF is illustrated in Figure-6.

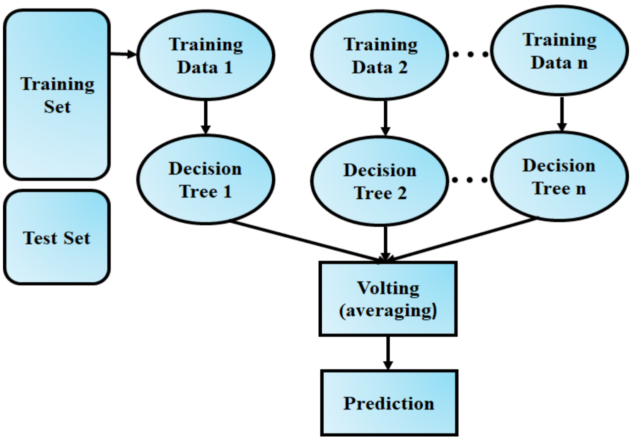

Figure-5. Random Forest Regressor.

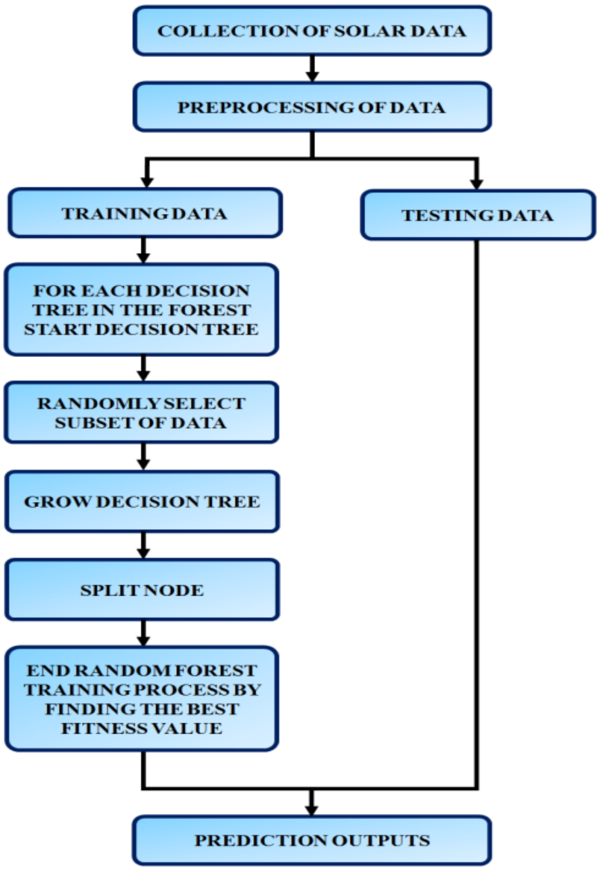

Figure-6. Flowchart for RF.

By repeating these two procedures a total of *C* times, a new set of data is predictable by relating predictions of all *C* trees. RF usage bagging to boost a variety of trees and minimize the overall variance of the model by nurturing them from many training datasets.

$$\hat{f}_{RF}^{C}(X) = \frac{1}{C}\sum_{i=1}^{C} T_i(X) \tag{5}$$

Where *X* denotes the input vector variable, the number of trees is *C* and a single regression tree is built utilizing bootstrapped samples and a subset of input variables as $T_i(X)$.

d) AdaBoost regressor

An ensemble of multiple weaker learning decision trees, referred to as adaptive boosting, outperforms random guessing by a small margin. But quite, due to the adaptive nature of the AdaBoost approach, the inaccuracy of the previous tree is reduced by transmitting the gradient of previous trees to succeeding trees. As a consequence, a strong learner is developed as a result of continuous learning of trees at every stage. The flowchart for the AdaBoost regressor is indicated in Figure-7.

The ultimate prediction, as depicted in Figure-8, is the weighted average of those produced by each tree. Ada Boost is more adaptable to outliers and irrelevant data because of its high degree of flexibility. The strategy is such that the information acquired by earlier trees is fed to subsequent trees, enabling them to concentrate exclusively on training samples that are difficult to predict.

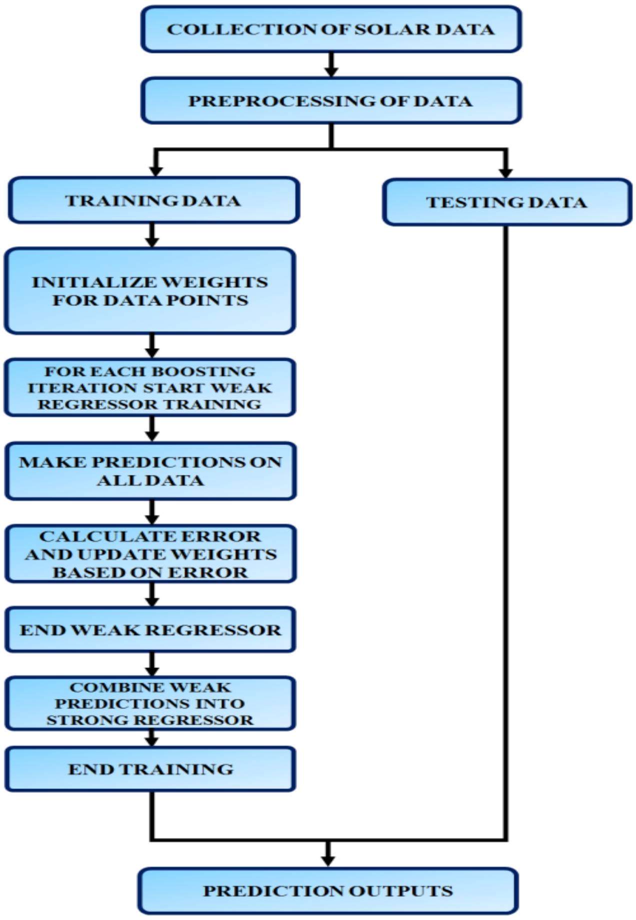

Figure-7. Flowchart for AdaBoost Regressor.

ARPN Journal of Engineering and Applied Sciences

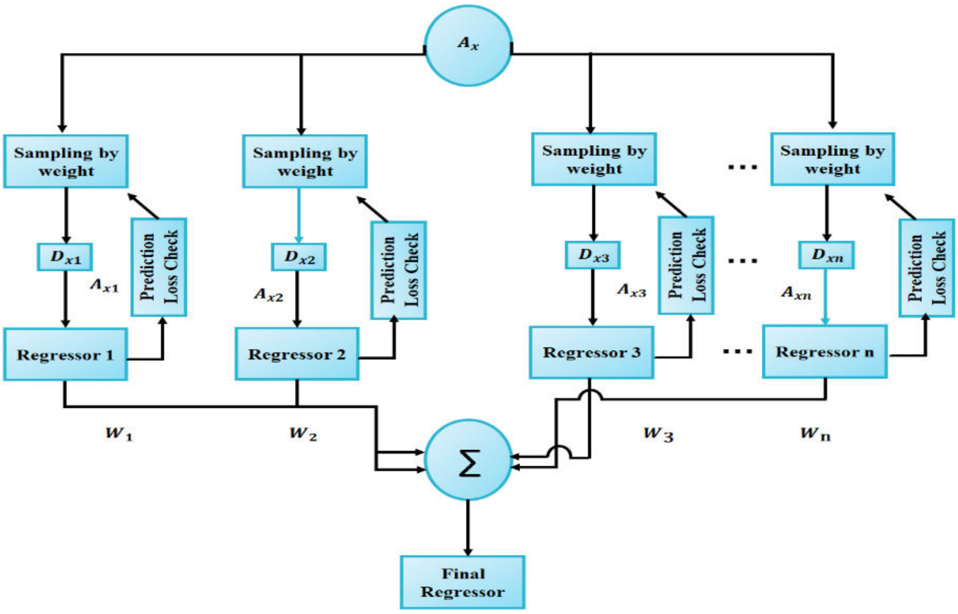

Figure-8. AdaBoost regression model.

e) XGBoost regressor

Gradient Boosting Decision Tree (GBDT) is the foundation of the classification algorithm and regression known as XGBoost. It generates a set number of weak learners to train, the majority of which are categories of regressed trees. Weighted summation is done after training to generate a final regression model. Depending on any residual errors found during the weak learner's preceding iteration, an advanced learner will always be included during the construction procedures.

A gradient-based learner is used to minimize overall model error. The final step is the development of a regression prediction model with a greater degree of significance. By introducing L1 and L2 regularization terms, XGBoost offers its users a range of optimization options. GBDT is used only to optimize the model using the first gradient. XGBoost expands the loss function using a second-order Taylor method. To minimize computation and avoid overfitting, XGBoost supports column sampling. As each iteration progresses, XGBoost distributes learning speed and creates a better learning environment for each tree.

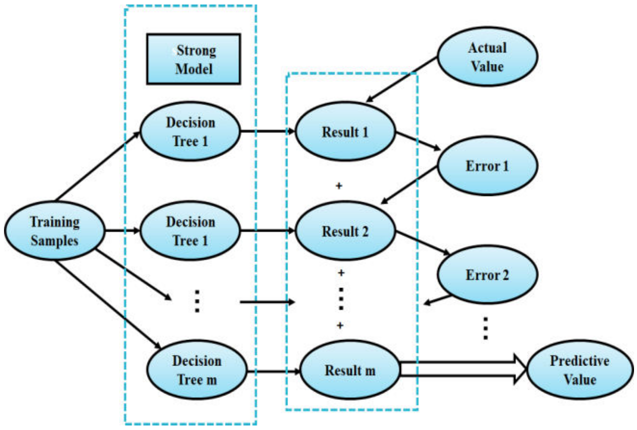

Figure-9. Principle of XGBoost Lifting.

It is assumed that sample x_i and class label y_i in training sample data set D are expressed as follows:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad (6)$$

The ith sample's prediction function is thus written as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)\,, f_k \in F \quad (7)$$

Where F denotes robust integration of the k decision tree, and $f_k(x_i)$ represents the discriminant function of the kth tree of ith data. XGBoost is a boosted tree model, as demonstrated in Figure-9.

ARPN Journal of Engineering and Applied Sciences

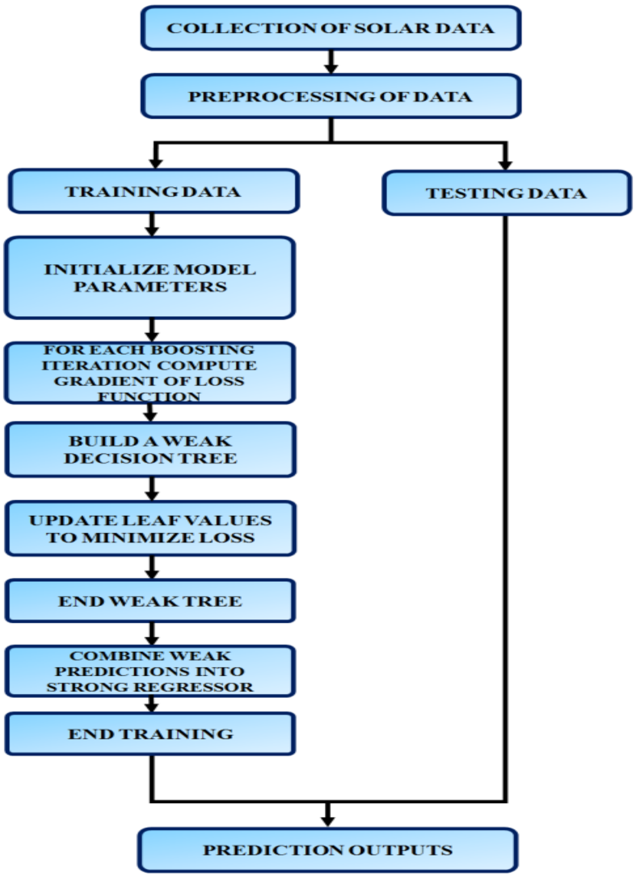

Figure-10. Flowchart for XGBoost Regressor.

This implies that a second tree is injected to learn the difference between predicted and actual values after the first tree has been trained to predict a value. In case when $k = 0, \hat{y}_i^{(0)} = 0$, similarly, when $k = 1$, $\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)$. While adding t trees the expression becomes,

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + \eta f_t(x_i), 0 < \eta < 1 \quad (8)$$

Here, the rate of learning is denoted as η. The flowchart for the XGBoost Regressor is presented in Figure-10.

3. RESULTS AND DISCUSSIONS

The outcomes of ML algorithms discussed in the preceding part are presented in this section. Machine Learning models are discussed in this article for predicting solar irradiance and PV power generation. There are several advantages over conventional ML models, particularly when it comes to forecasting time series data. Every model has strengths and limits in predicting irradiance and PV power; hence, determining the best among all of them is difficult. Python Software is utilized in analyzing the energy prediction efficiency of the proposed work.

A. Performance Metrics

To compare algorithms based on results, it is essential to assess the performance of the predictive model in practice using multiple performance indicators. By squaring and averaging across *N*, Root Mean Square Error (RMSE) restricts regression models for the distance that $\hat{y}_i$ is from y_i.

$$\text{RMSE} = (1/N \times \sum((y_i - \hat{y}_i)^2)^{\frac{1}{2}} \quad (9)$$

The average distance among $\hat{y}_i$ and y_i is measured by Mean Absolute Error (MAE).

$$\text{MAE} = 1/N \times \sum|y_i - \hat{y}_i| \quad (10)$$

Parameter R^2 Score indicates the strength of the relationship between $\hat{y}_i$ and y_i.

$$R^2 = 1 - (\sum(y_i - \hat{y}_i)^2 / \sum(y_i - \bar{y}_i)^2) \quad (11)$$

The above-mentioned parameters are considered as; predicting energy power is a regression problem. RMSE restricts the higher discrepancy between predicted and real solar power, whereas MAE just averages offsets. The comprehensive analysis of approaches like Linear Regression, RF Regressor, AdaBoost, KNN Regressor, and XGBoost techniques along with their evaluation metrics utilized for predicting solar power generation by adopting Python Software.

The energy prediction outcomes of different machine learning regressors are listed in Table-3 obtained using Python software. It is crucial to project a predictive model's performance depending on a variety of enactment criteria to compare techniques based on the results produced. In this work, results are provided for performance measures like training and testing accuracy, R^2 Score, and error measures like RMSE, and MAE.

Table-3. Experimental outcomes.

Algorithm	Training Accuracy (%)	Testing Accuracy (%)	RMS Score	MAE Score	R^2 Score	CV R^2 Score
Linear Regression	20.707648	28.296171	29.168202	21.678412	28.296171	28.296171
Random Forest Regressor	88.579307	24.094348	30.010660	21.363508	24.094348	22.397013

 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

AdaBoost Regressor	93.547785	21.1489009	30.587384	22.371350	21.148909	9.831572
K-Nearest Neighbours Regressor	70.118629	8.687490	32.915740	23.675676	8.687490	25.821828
XGBoost Regressor	97.885245	4.826993	33.604341	23.759979	4.826993	4.826993

The depiction of energy prediction utilizing a linear regressor is showcased in Figure-11. The outcomes reveal that the model achieves a testing accuracy of 28.296%, accompanied by a training accuracy of 20.707%. In terms of precision assessment, the RMSE is computed to be 29.168, while MAE stands at 21.678. Additionally, a coefficient of determination R^2 score of 28.296% is obtained. R^2 score is a statistical measure that denotes the proportion of the variance in the dependent variable (target) that is explained by independent variables (features) in the model. Higher R^2 score values signify a better fit of the model data. This signifies that the performance of the regression line closely approximates the actual data values. In essence, the regression line effectively captures the underlying trends within the dataset. To assess the performance of a model on multiple subsets of the data across validation technique is used. CV R^2 is the R-squared score calculated using cross-validation, providing a more robust estimate of model performance. CV R^2 helps mitigate the risk of overfitting or underfitting, offering a more reliable evaluation of the model's generalization ability. By using cross-validation, the CV R^2 score offers a more realistic estimate of the model to perform on unseen data. These results emphasize the model's ability to create reasonably accurate predictions, as reflected by low RMSE and MAE values, and the relatively moderate R^2 score. This suggests that the linear regressor can provide meaningful insights into energy prediction based on the given data.

The Prediction of energy using the KNN regressor is illustrated in Figure-12. It is observed that with a test accuracy of 8.687%, a training accuracy value of 70.118% is obtained. Consequently, the R^2 score obtained is the same as that for testing accuracy, ensuring that the regression line is similar to the actual data value. Also, an RMSE error value of 32.915 is attained with an MAE value of 23.675, respectively.

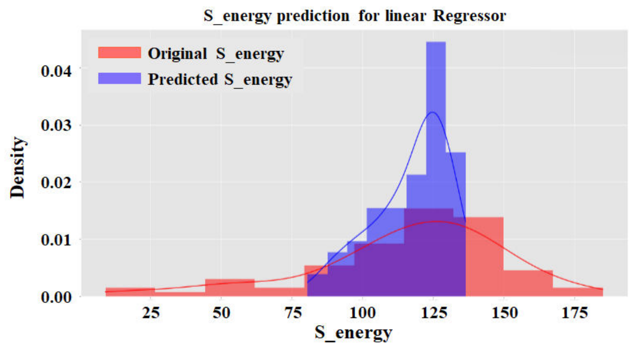

Figure-11. Energy prediction using linear regression model.

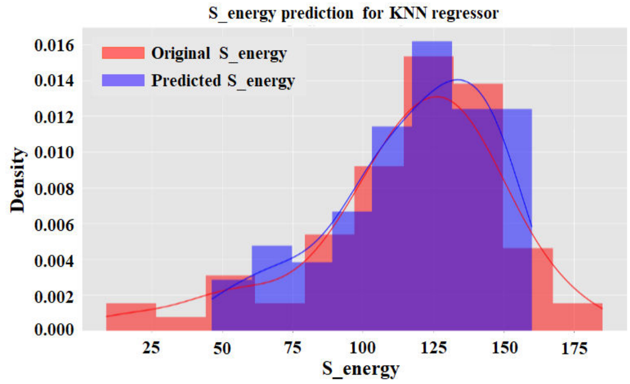

Figure-12. Energy prediction using KNN regressor.

The outcomes of energy prediction by adopting an RF Regressor are depicted in Figure-13. It is noticed that with a test accuracy of 24.094%, a training accuracy value of 88.579% is achieved, where the corresponding R^2 score value also equals the test accuracy. The RMSE and MAE achieved using the RF Regressor are 30.205 and 21.49, respectively. The KNN regressor seems to struggle with generalization, while the RF regressor provides more consistent and accurate predictions. The RF regressor's ensemble nature likely contributes to its ability to capture complex patterns in energy consumption data.

The results in terms of accuracy and error measures obtained using the AdaBoost Regressor are illustrated in Figure-14. From Figure-14 it is observed that with test accuracy of 21.148% an improved training accuracy of 93.547% is attained, along with RMSE value of 32.016 and MAE value of 22.427, respectively. Also, it is noticed that R^2score value is also similar to the actual data value of 13.608.

The prediction of energy using the XGBoost Regressor is depicted in Figure-15. It is noticed that, with a minimized testing accuracy of 4.826%, an enhanced training accuracy value of 97.885% is achieved. Like other approaches, the RMSE and MAE values of the XGBoost Regressor are 33.604 and 23.759. Moreover, the value of the R^2score is also the same as of test accuracy.

VOL. 19, NO. 22, NOVEMBER 2024 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

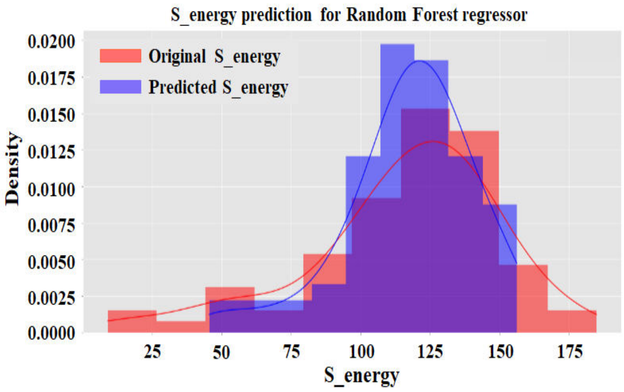

Figure-13. Energy prediction using RF regressor.

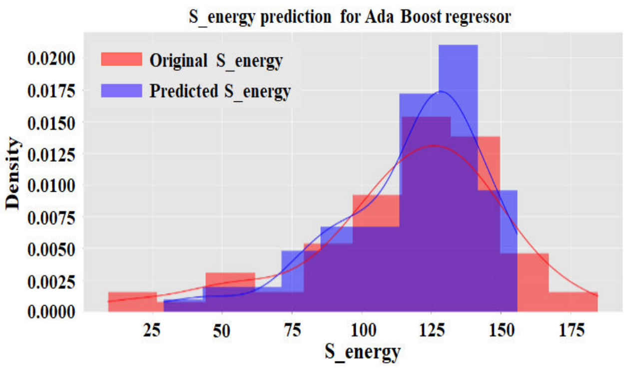

Figure-14. Energy prediction using AdaBoost regressor.

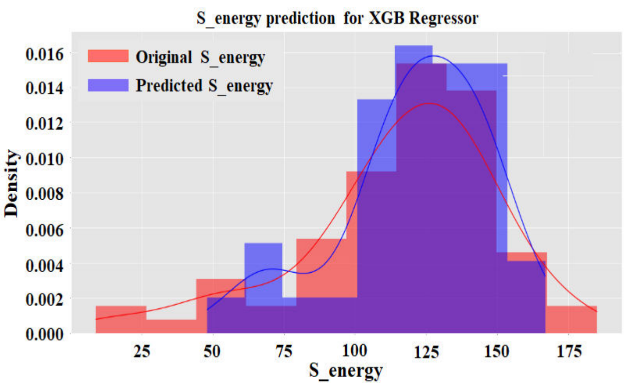

Figure-15. Energy prediction using XGBoost regressor.

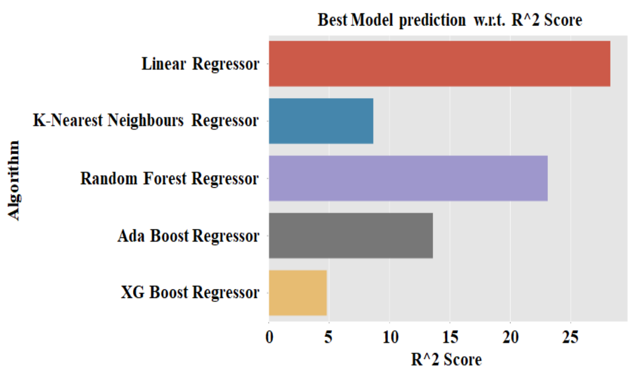

Figure-16. Comparison of regressors vs R^2 score.

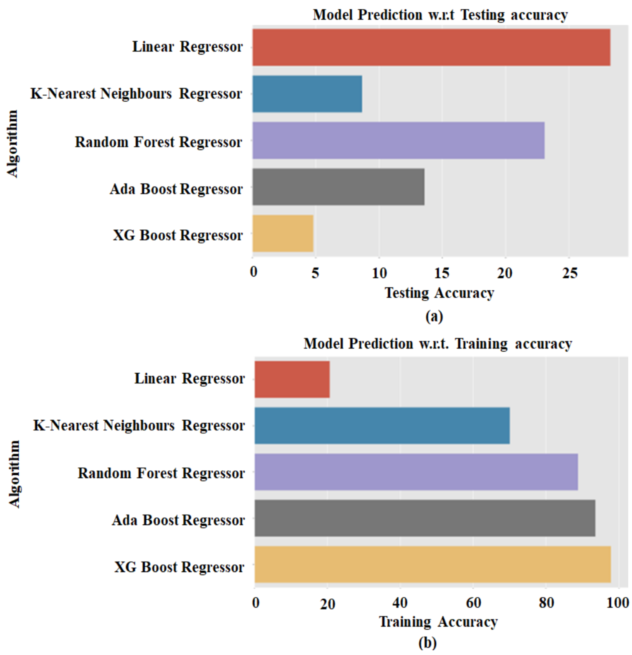

Figure-17. Comparison of (a) testing accuracy vs (b) training accuracy.

The AdaBoost Regressor appears to strike a better balance between training and test accuracy, offering a more reliable prediction. On the other hand, the XGBoost Regressor achieves remarkable training accuracy but falters in generalization, leading to lower testing accuracy. The distinctive features lie in the trade-off between training and test performance, as well as the ability to capture underlying energy consumption patterns accurately.

A comparative analysis of the R^2score, which is an important metric for evaluating regression techniques in machine learning is illustrated in Figure-16. It is noticed that, the attained results of different regressors provide the best fit and do not differ from the test value.

A comparative analysis of various regressors with respect to training and testing accuracy is represented in Figure-17. It is observed that AdaBoost and XGBoost regressors show enhanced accuracy of above 90% and result in better performance when contrasted with approaches like linear regressor, KNN regressor, and RF Regressor. It is observed that the AdaBoost regressor results in 93.54% and the XGBoost regressor results in 97.88% respectively.

Table-4. Comparison with Existing Works.

Techniques	RMS score
[28]	118.95
[28]	89.67
[29]	91.79
[30]	57.8
Linear Regression	29.16
Random Forest Regressor	30.01
AdaBoost Regressor	30.58
K-Nearest Neighbours Regressor	32.91
XGBoost Regressor	33.60

Table-4 provides a comparison of the proposed machine learning approaches with existing works in terms of RMS score. In comparison with the state-of-the-art approaches, the ML techniques exhibit reduced RMS scores indicating improved prediction accuracy.

4. CONCLUSION

The variability of PV power, dependent on environmental conditions, necessitates accurate forecasting for effective energy management. Machine Learning (ML) models offer a promising alternative due to their precision. This study explores ML applications for time-series data in predicting PV power generation. Regressors including Linear, KNN, RF, AdaBoost, and XGBoost are evaluated for optimal forecasting. Notably, AdaBoost and XGBoost demonstrate superior accuracy, achieving 93.54% and 97.88%, respectively, in PV power forecasting. The study concludes that AdaBoost and XGBoost's capacity to learn higher-level features contributes to their dominance in achieving accurate predictions.

REFERENCES

[1] Dairi A., Harrou F., Sun Y., Khadraoui S. 2022. Short-term forecasting of photovoltaic solar power production using variational auto-encoder driven deep learning approach. Applied Sciences. 10(23): 8400.

[2] Park S., Kim Y., Ferrier N. J., Collis S. M., Sankaran R., Beckman P. H. 2021. Prediction of solar irradiance and photovoltaic solar energy product based on cloud coverage estimation using machine learning methods. Atmosphere. 12(3): 395.

[3] Li Z., Rahman S. M. M., Vega R., Dong B. 2016. A hierarchical approach using machine learning methods in solar photovoltaic energy production forecasting. Energies. 9(1): 55.

[4] Bosman L. B., Leon-Salas W. D., Hutzel W., Soto E. A. 2020. PV system predictive maintenance: Challenges, current approaches, and opportunities. Energies. 13(6): 1398.

[5] Al-Shetwi A. Q., Hannan M. A., Jern K. P., Alkahtani A. A., Abas A. E. P. G. 2020. Power quality assessment of grid-connected PV system in compliance with the recent integration requirements. Electronics. 9(2): 366.

[6] Dairi A., Harrou F., Sun Y., Khadraoui S. 2020. Short-term forecasting of photovoltaic solar power production using variational auto-encoder driven deep learning approach. Applied Sciences. 10(23): 8400.

[7] Sorkun M. C., Incel O. D., Paoli C. 2020. Time series forecasting on multivariate solar radiation data using deep learning (LSTM). Turkish Journal of Electrical Engineering and Computer Sciences. 28(1): 211-223.

[8] Lee W., Kim K., Park J., Kim J., Kim Y. 2018. Forecasting solar power using long-short term memory and convolutional neural networks. IEEE Access. 6, pp. 73068-73080.

[9] Suresh V., Janik P., Rezmer J., Leonowicz Z. 2020. Forecasting solar PV output using convolutional neural networks with a sliding window algorithm. Energies. 13(3): 723.

[10] Polo J., Martín-Chivelet N., Alonso-Abella M., Sanz-Saiz C., Cuenca J., de la Cruz M. 2023. Exploring the PV Power forecasting at building façades using gradient boosting methods. Energies. 16(3): 1495.

[11] Huang C. J., Kuo P. H. 2019. Multiple-input deep convolutional neural network model for short-term photovoltaic power forecasting. IEEE Access. 7, pp. 74822-74834.

[12] Alsharif M. H., Younes M. K., Kim J. 2019. Time series ARIMA model for prediction of daily and monthly average global solar radiation: The case study of Seoul, South Korea. Symmetry. 11(2): 240.

[13] Sharadga H., Hajimirza S., Balog R. S. 2020. Time series forecasting of solar power generation for large-scale photovoltaic plants. Renewable Energy. 150, pp. 797-807.

[14] Belmahdi B., Louzazni M., Bouardi A. E. 2020. A hybrid ARIMA-ANN method to forecast daily global solar radiation in three different cities in Morocco. The European Physical Journal Plus. 135, pp. 1-23.

[15] Louzazni M., Mosalam H., Cotfas D. T. 2021. Forecasting of photovoltaic power using non-linear auto-regressive exogenous artificial neural network and time series analysis. Electronics. 10(16): 1953.

[16] Wan C., Zhao J., Song Y., Xu Z., Lin J., Hu Z. 2015. Photovoltaic and solar power forecasting for smart grid energy management. CSEE Journal of Power and Energy Systems. 1(4): 38-46.

[17] Khodayar M., Mohammadi S., Khodayar M. E., Wang J., Liu G. 2019. Convolutional graph autoencoder: A generative deep neural network for probabilistic spatiotemporal solar irradiance forecasting. IEEE Trans. on Sustainable Energy, 11(2): 571-583.

[18] de Santos D. S., de Mattos Neto P. S, de Oliveira J. F., Siqueira H. V., Barchi T. M., Lima A. R., Madeiro F., Dantas D. A., Converti A., Pereira A. C., de Melo Filho J. B. 2022. Solar irradiance forecasting using dynamic ensemble selection. Applied Sciences. 12(7): 3510.

[19] Huang H., Karami H., Ehteram M., Chau K. W., Zhang Q. 2022. Solar radiation prediction using improved soft computing models for semi-arid, slightly arid, and humid climates. Alexandria Engineering Journal. 61(12): 10631-10657.

[20] Alharbi F. R., Csala D. 2021. Wind speed and solar irradiance prediction using a bidirectional long short-term memory model based on neural networks. Energies. 14(20): 6501.

[21] Zafar R., Vu B. H., Husein M., Chung I. Y. 2021. Day-ahead solar irradiance forecasting using a hybrid recurrent neural network with weather classification for power system scheduling. Applied Sciences. 11(15): 6738.

[22] Hosseini M., Katragadda S., Wojtkiewicz J., Gottumukkala R., Maida A., Chambers T. L. 2020. Direct normal irradiance forecasting using multivariate gated recurrent units. Energies. 13(15): 3914.

[23] Selvaraj T., Rengaraj R., Venkatakrishnan G., Soundararajan S. G., Natarajan K., Balachandran P. K., David P. W., Selvarajan S. 2022. Environmental Fault Diagnosis of Solar Panels Using Solar Thermal Images in Multiple Convolutional Neural Networks. International Transactions on Electrical Energy Systems.

[24] Natarajan K., Bala P. K., Sampath V. 2020. Fault detection of solar PV system using SVM and thermal image processing. International Journal of Renewable Energy Research (IJRER). 10(2): 967-977.

[25] Abdelhag M. E., Elmarud Ali S. E., Amin S. T., Ali A., AbdElwakil Metwally Khan A. 2022. Machine Learning Based Model for Solar Radiation Prediction. Machine Learning. 54(02).

[26] Babbar S. M., Lau C. Y., Thang K. F. 2021. Long-Term Solar Power Generation Prediction using Adaboost as a Hybrid of Linear and Non-linear Machine Learning Model. International Journal of Advanced Computer Science and Applications. 12(11).

[27] Li X., Ma L., Chen P., Xu H., Xing Q., Yan J., Lu S., Fan H., Yang L., Cheng Y. 2022. Probabilistic solar irradiance forecasting based on XGBoost. Energy Reports. 8, pp. 1087-1095.

[28] Garniwa P. M. P., Ramadhan R. A. A., Lee H. J. 2021. Application of semi-empirical models based on satellite images for estimating solar irradiance in Korea. Applied Sciences. 11(8): 3445.

[29] Yeom J. M., Park S., Chae T., Kim J. Y. and Lee C. S. 2019. Spatial assessment of solar radiation by machine learning and deep neural network models using data provided by the COMS MI geostationary satellite: A case study in South Korea. Sensors. 19(9): 2082.

[30] Pereira S., Canhoto P., Salgado R., Costa M. J. 2019. Development of an ANN-based corrective algorithm of the operational ECMWF global horizontal irradiation forecasts. Solar Energy. 185, pp. 387-405.