Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 6, 2025

SOLAR RADIATION PREDICTION FOR RENEWABLE ENERGY: A MACHINE LEARNING PERSPECTIVE

Abdul Wasay[1], Bushra Raza[2], Zaryab Khan[3], Muhammad Amir[4], Bilal Ur Rehman[*5], Humayun Shahid[6], Kifayat Ullah Bangash[7]

*[1,2,3,4, *5,7]Department of Electrical Engineering, Faculty of Electrical and Computer Engineering, University of Engineering and Technology, Peshawar, KPK, Pakistan.*
[6]Department of Telecommunication Engineering, University of Engineering and Technology, Taxila, Pakistan.

DOI: https://doi.org/10.5281/zenodo.15761985

Keywords
Machine Learning (ML), Solar Radiation, Photovoltaic (PV)

Article History
Received on 21 May, 2025
Accepted on 21 June, 2025
Published on 28 June, 2025

Corresponding Author: *
Bilal Ur Rehman

Abstract
Accurate solar radiation forecasting plays a pivotal role in enhancing the efficiency, reliability, and integration of large-scale renewable energy systems. The performance of photovoltaic (PV) systems strongly depends on atmospheric and seasonal variability, necessitating precise short-term predictions to support optimal energy management and maintain grid stability. This study applies advanced machine learning (ML) techniques within a time-series forecasting framework to improve the accuracy of solar radiation prediction. Rigorous data preprocessing—encompassing cleaning, segmentation, and validation—ensures dataset integrity and prevents data leakage. A range of regression models, including Ridge, Lasso, XGBoost, Decision Tree, Random Forest, and Linear Regression, undergo evaluation using Root Mean Squared Error (RMSE) as the primary metric. K-fold cross-validation identifies Random Forest as the most effective model, demonstrating its superior performance in enhancing predictive accuracy and enabling more reliable integration of solar energy into modern power grids.

1. INTRODUCTION

Effective solar power forecasting is vital in optimizing renewable energy and maintaining grid stability, as observed in the global energy market with the mainstream transition from fossil fuels to clean sources. Using fossil fuel energy in power production leads to the emission of greenhouse gases and carbon dioxide, which increases climate change [1]. On the contrary, Photovoltaic (PV) systems use sunlight that is first converted to electrical energy, providing a clean solution to the renewable energy problem. Nevertheless, the daily production of PV is quite unpredictable because it largely depends on the geographical location, the time of the day, atmospheric conditions, seasonal weather, etc. The inherent variability makes highly-tuned short-term forecasts mandatory to optimize the management of the energy systems and dependent less on the backup systems.

The development of modern machine learning (ML) technology enables the current research to formulate a considerable framework to overcome the difficulties of sun energy forecasting. Machine learning models are particularly good at spotting complicated trends and connections with massive datasets, enabling better forecast accuracy [2]. When applied to solar power, ML algorithms exploit weather parameters, e.g., irradiance, ambient temperature, time-relevant parameters, and historical generation data, to estimate future power production in an electrical power system. These projection

models are important in alleviating the lack of conformity between energy supply and demand, improving grid stability, and lower operating costs.

Ensemble methods have proven to be promising for the different ML approaches. Combining the predictions of many decision trees using a Random Forest regression has become one of the most promising candidates for solar forecasting [3]. Random Forest effectively handles non-linear interactions through bootstrapping and random feature selection, compensating for any over-fitting and having robust predictions. The capacity to work with high dimensional data and capture non-linear patterns of interaction between the input variables qualifies it to be a good tool for solar energy prediction, where various environmental factors affect the prediction non-linearly.

This study will be modeling an optimized forecasting model incorporating Ranson (Random Forest) regression in precisely anticipating solar irradiance. This model uses strict data preprocessing, which consists of cleaning, splitting, and cross-validation to sustain the quality of the input data and guarantee reasonable performance scores. The proposed study will help improve the performance of the PV system and increase the overall acceptance of renewable energy technologies in the contemporary power grid due to the issues that the current approach to such forecasting can finally address.

To conclude, the combination of sophisticated machine learning practices and rich meteorological and PV performance data presents an optimistic way to deal with obstacles of solar power forecasting. Not only has this study indicated the viability of applying Random Forest regression to this end, but it also shows the promise to increase energy management practices, decrease the cost of operations, and aid renewable energy systems in growing sustainably.

2. LITERATURE REVIEW

Forecasting solar energy generation has also changed notably as it has been able to incorporate a lot of statistical aspects and even machine learning (ML) models. Recent studies have suggested hybrid models incorporating statistical models and ML and have proved more precise and cost-effective. Researchers have been investigated these hybrid structures are more effective than the original ML structures in the production of more accurate predictions concerning solar output [3]. As an illustration, an experimental configuration based on thin-film and polycrystalline photovoltaic solar panels reaching a 10 MW capacity indicates that low-bias methods of ML models could approximate near-real-time estimation of energy generation at a resolution of five minutes. It was also observed that predictions were considerably more accurate when it was evident that it was cloudy since it resulted in significant variance and fluctuations.

Random Forest (RF) is one of the (ensemble) techniques that has proved to be very helpful in predicting solar energy. A lot can be said about the comparison of Support Vector Regression (SVR), Linear Regression (LR), and RF since the latter demonstrated the best accuracy rate of up to 94.01%, especially when weather variables such as temperature and irradiance have been added [4]. Other models, such as Lasso, Ridge, Elastic Net, and baseline regressions, have also been tested with PV datasets that contained temporal and environmental features. In contrast, RF and deep learning models have performed consistently worse in predicting accuracy in these cases [5].

Solar forecasting has also benefited from deep learning methods like Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks, which enable the learning of long-term dependencies in sequential data. These models have performed significantly better than classical ML algorithms because they fit well with trends in time solar irradiance [6][7]. Despite their abilities, they have weaknesses due to overfitting, especially when their datasets are small, and thus, they require regularization methods like dropout layers.

Assimilation of meteorology data such as temperature, humidity, wind speed, and cloud cover into the forecasting systems has been revealed to enhance the success rates of the forecasts largely. Compared to the conventional methods of SVM, RF, and KNN, the deep learning models have shown reduced error metrics (MAE, MSE, RMSE) compared to their counterparts [8]. Also, various ensemble methods such as RF are highly successful at reflecting non-linear dependencies, which in turn help reduce variance in the forecast and, ultimately, increase the reliability [9].

The Artificial Neural network-LSTM (ANN-LSTM) frameworks have been built and tested to predict PV. They are more accurate in forecasting electrical and meteorological data [10,11]. Neural network parameters have also been optimized using genetic algorithms that produce better prediction accuracy by determining superior weight and bias settings [12]. Moreover, it has been observed that using meteorological data together with ML shows significant performance improvement, proving that detailed input features are required to ensure a proper prediction [13,14].

Accuracy in forecasting is also enhanced through LSTM architectures because they can address long-term temporal patterns in solar datasets. PV applications are particularly suitable for time-series predictions due to their past learning capabilities [15]. In the meantime, probabilistic forecasting capabilities have been provided by Bayesian neural networks that factor in uncertainty in the prediction, which is paramount in the risk management of grid functioning. Such networks do bring confidence intervals, besides point forecasts, which allow for making better decisions [16]-[18].

To sum it up, significant studies support using ensemble learning and deep learning frameworks to improve the accuracy of solar power prediction. Random Forest, LSTM, GRU, and hybrid strategies, especially those with meteorological and historical data, generally work much better. Based on those, the current research focuses on rigorous preprocessing and multiple regression modeling on short-term solar radiations and recognizes Random Forest as the most appropriate method. The results can help improve solar energy integration and grid optimization in practical energy management systems.

3. DATA PREPROCESSING AND TRANSFORMATION

Good solar power forecasting must be based on structured, quality data. Consequently, preprocessing and transforming data is an important preliminary procedure that makes the data correct, standardized, and ready to be used in potential machine-learning processes. First, the dataset is analyzed to detect anomalies in the form of missing values, inaccurate entries, and outliers. It is fixed to ensure data integrity and not to cause bias in model outputs [9].

Preprocessing entails cleaning activities that include treating null values, eliminating duplicate records, identifying and adjusting outliers, and decreasing noises. Further, transformation methods of data (normalization, standardization, and encoding categorical variables) are used to fit data to achieve the best performance by the model. The processes assist in making similar scaling of features and compatibility with learning algorithms.

Data manipulation is the process of sorting and rearranging information using data filtering, sorting, aggregation, and merging of datasets. Such techniques enable more efficient feature engineering and dimensionality reduction, which enable the increased representation of underlying patterns in the data.

This part discusses the primary methodologies employed in preprocessing and transforming the dataset in this work. These steps will make data more valuable (regarding quality and consistency), improve model generalization, and result in more reliable and accurate solar power predictions.

3.1 HISTORICAL SOLAR POWER DATA

There will be monitoring on 22 inverters at a 15-minute interval through a 25-day data collection initiative, as illustrated in Table 1. Each inverter records important data, which includes the time when the reading was done, the ID number unique to that inverter, and the measurements concerning the power, the temperature, the DC input voltage or the AC output voltage, and the current of the inverter. A reliable system should be established to record this data regularly. Such information is stored in a database that can handle frequent updates and high-volume storage. Data retention and back-ups undertaken frequently need a plan to prevent data loss and maintain the quality of data.

The data must be cleansed and prepared first to overcome any missing value or inconsistency so that no data analysis can be started before it. Matplotlib and Seaborn visualize trends and detect patterns and/or anomalies. Performance is measured until the end of the 25 days, with inverter efficiency as one of the key metrics. Time-series analysis helps in predicting trends in the future. The results may be

reported continuously to track performance and displayed on live dashboards for continuous checking. This analysis can be used to improve solar energy systems and can be beneficial to meet regulatory requirements, predict maintenance, and optimize the inverter.

Table 1: Generation dataset

DATE_TIME	15-minute time stamp
PLANT _ID	Same for all
SOURCE KEY	Numbered from 1 to 22 for 22 inverters
DC_POWER	Power generated from each inverter
AC_POWER	AC power converted from DC TO AC after 15 min
DAILY YIELD	Total yield obtained for one day
TOTAL YIELD	Total power generated for some time

3.2 METEOROLOGICAL DATA

Weather data is gathered every 15 minutes over 25 days, recording both the plant's temperature and the ambient temperature via sensors affixed to the solar panels, as seen in Table 2. Additionally, solar irradiation data is recorded at the same intervals. This frequent data collection ensures a comprehensive understanding of the environmental conditions impacting the solar panels and the overall performance of the plant. We can learn more about how variations in solar irradiation and temperature impact the efficiency of energy production by examining these characteristics.

Table 2: Weather dataset

"DATE_TIME"	15-minute time stamp
"PLANT _ID"	Common for all file
"SOURCE KEY"	Numbered from 1 to 22 for 22 inverters
"AMBIENT_TEMPERATURE"	Plant's ambient temperature
"MODULE_TEMPERATURE"	Temperature reading for the solar panel module that is connected to the sensor panel
"IRRADIATION"	Radiation level over a 15-minute period

3.3 TIMESTAMPS MANAGEMENT

As of right now, each timestamp is repeated according to the number of inverters for which data is available for that specific time stamp; for example, if only 12 inverters' data is available for the first timestamp, 12 rows will be displayed for that single timestamp as shown in Figure 2.

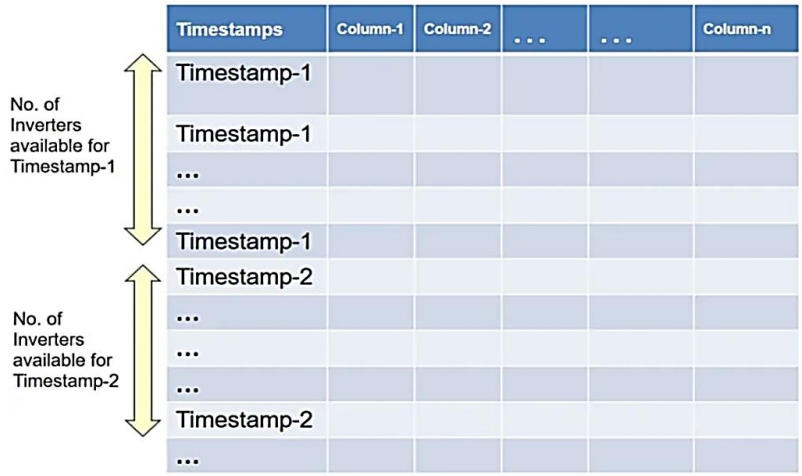

Figure 2: Timestamps.

However, we would like our forecasts to be broken down day-by-day into intervals of fifteen minutes at the plant level. Thus, from 00:00 to 24:00, the structure that is needed is day-wise (timestamp-wise) rows with columns that represent the total of all inverter values (for DC Power, AC Power, etc.) for that timestamp as displayed in Figure 3.

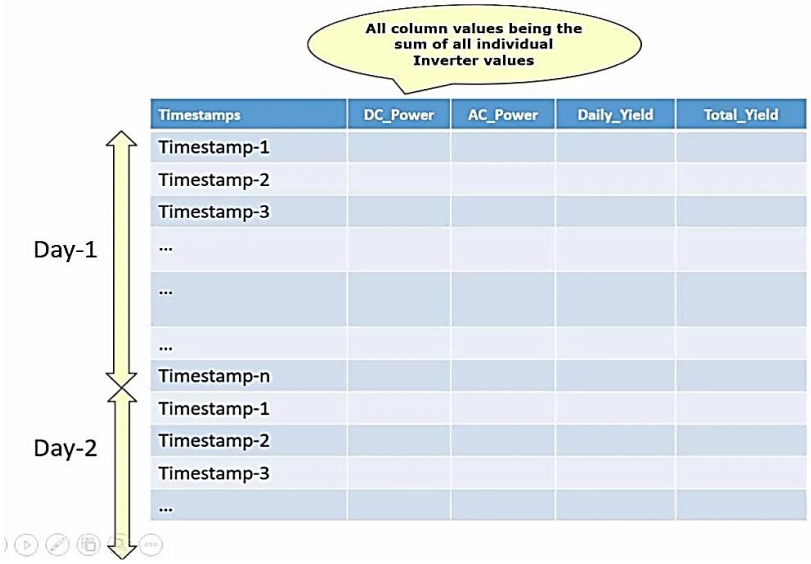

Figure 3: Timestamp Management.

3.4 DATA CLEANING

In the solar power forecast model, the positive attributes of information should be incorporated into the predictive model, which makes data cleansing important. This method involves determining missing values by counting the missing rows and non-null cells or removing rows or columns with missing values that can distort forecasts. The direct process of removing extraneous information, the column of PLANT_ID and Yields in the two generating and weather data sets, is carried out without any new data frames. The data in the field labeled PLANT_ID can be considered redundant, as it is the same throughout all the entries and is not related to current forecasting or analysis tasks being carried out. Analysts can use data cleansing to increase the quality of the inputs to forecasting models and boost the accuracy of the prediction, aided by data cleansing and grounded by informed decisions in energy management. Data integrity is a required process; therefore, verifying that there are no null values is important, as the generation data and the meteorological data values are the key elements to accurate forecasting results. Such practices help to maximize models so that the forecast of solar power generation is more accurate.

3.5 TRANSFORMATION OF DATA

To reduce the effects of differences in input scale, algorithms that are sensitive to scale, e.g., algorithms used in solar irradiance prediction models, must normalize data (usually between 0 and 1). The data is scaled with a mean value of 0 and a standard deviation of 1, making it easy to compare with other meteorological and solar radiation databases. The log transformation is applied to level the variance, normalize the solar irradiance data distribution, and make it more apt for modeling. Binning converts unsequential variables, i.e., the level of solar radiation, to categorical ranges and makes it easy to study data structure in a simplified manner. Categorical data, e.g., meteorological conditions or geographic locations that impact sun exposure, can be transformed into numerical form to use them be used most effectively in predictive mechanisms to enhance the quality of data, which makes forecasts and planning of solar energy generation and management a lot facilitated and more reliable.

3.6 GENERATION DATA MERGING

Solar power forecasting would need to integrate information on numerous inverters to create an aggregate one, enabling them to capture the change in power generation in different locations or units. The totaled values in Table 3 allow the analysts to study the trends, correlations, and dependencies at different sources of solar power generating data. Assimilation of detailed knowledge of solar power generation parameters (e.g., irradiance, temperature, equipment performance in various sites or units) increases the predictability of forecasting models. Data should be combined and reformatted to make the best out of predictive models and increase the efficiency and reliability of solar energy production systems.

Table 3: Merged Generation dataset

	DATE_TIME	DC_POWER_1	AC_POWER_1	Inverter_No_1	DC_POWER_2	AC_POWER_2	Inverter_No_2	DC_POWER_3	AC_POWER_3	Inverter_No_3	...
0	2020-05-15 00:00:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
1	2020-05-15 00:15:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
2	2020-05-15 00:30:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
3	2020-05-15 00:45:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
4	2020-05-15 01:00:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
...	...	...	...	...	...	...	...	...	...	...	...
3254	2020-06-17 22:45:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
3255	2020-06-17 23:00:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
3256	2020-06-17 23:15:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
3257	2020-06-17 23:30:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...
3258	2020-06-17 23:45:00	0.0	0.0	1.0	0.0	0.0	2	0.0	0.0	3	...

3259 rows × 67 columns

3.7 DATA INTEGRATION AND TIME BLOCK GENERATION

This approach entails amalgamating weather and solar generation datasets to create a cohesive dataset for analysis and forecasting purposes. Integrating these datasets according to timestamps allows for examining the relationships between weather conditions and solar power output. Table 4 presents the integrated solar generation and weather data set.

Table 4: Integrated Solar Generation and Weather dataset

	BLOCK	DATE	TIME	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	DC_POWER_1	AC_POWER_1	Inverter_No_1	DC_POWER_2	...
0	1	2020-05-15	00:00	27.004764	25.060789	0.0	0.0	0.0	1.0	0.0	...
1	2	2020-05-15	00:15	26.880811	24.421869	0.0	0.0	0.0	1.0	0.0	...
2	3	2020-05-15	00:30	26.682055	24.427290	0.0	0.0	0.0	1.0	0.0	...
3	4	2020-05-15	00:45	26.500589	24.420678	0.0	0.0	0.0	1.0	0.0	...
4	5	2020-05-15	01:00	26.596148	25.088210	0.0	0.0	0.0	1.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
3254	92	2020-06-17	22:45	23.511703	22.856201	0.0	0.0	0.0	1.0	0.0	...
3255	93	2020-06-17	23:00	23.482282	22.744190	0.0	0.0	0.0	1.0	0.0	...
3256	94	2020-06-17	23:15	23.354743	22.492245	0.0	0.0	0.0	1.0	0.0	...
3257	95	2020-06-17	23:30	23.291048	22.373909	0.0	0.0	0.0	1.0	0.0	...
3258	96	2020-06-17	23:45	23.202871	22.535908	0.0	0.0	0.0	1.0	0.0	...

3259 rows × 72 columns

5 DATA SPLITTING

Splitting data is crucial for practical model training and assessment in machine learning for solar power generation prediction. Due to the temporal correlation between solar and meteorological data, it is essential to partition the dataset meticulously to preserve the sequence of data points. This chapter outlines a method for partitioning a dataset of 25 days of meteorological and solar power generation data, focusing on continuity. The last three days are allocated for testing, whilst the preceding twenty-two days are designated for training. This approach is essential for capturing temporal relationships in the data and implementing effective imputation methods.

Data splitting is crucial to test machine learning models' performance using data not seen during training. This approach is essential for time series data, such as weather and solar power generation data, because maintaining chronological order is vital for accurately capturing temporal trends that influence forecasting precision.

We may evaluate the model using data that simulates real forecasting situations by allocating the final three days for testing. Further, it enables the objective assessment of the model's predictive capability. This strategy prevents overfitting by ensuring the model learns underlying patterns from new data rather than only relying on the training set. Moreover, separating chronological data preserves the temporal sequence, crucial for imputation methods reliant on data continuity.

5.1 DATA OVERVIEW

The 25-day dataset includes data on temperature, humidity, solar irradiance, and AC/DC power production. For consistency and comprehensive model evaluation, the data is divided into two sets: a training set including the initial 22 days and a testing set comprising the last 3 days.

5.2 SPLITTING STRATEGY

The subsequent approaches are employed to partition the dataset into training and testing subsets:

Identify the Time Span: The dataset covers 25 days, from day 1 to day 25.

Determine the Split Point:
The training set includes data from day 1 to day 22, and the testing set includes data from day 23 to day 25.

Allocate Data to Sets:
Based on the split point that has been found, data is divided into training and testing sets so that the training set includes the first 22 days and the testing set includes the final three days.

Our training and testing datasets are comprised of 2971 and 288 rows respectively. Missing values from training dataset are assigned in next segment.

```
Train: (2971, 73)
Test: (288, 73)
```

5.3 MANAGING "MISSING DATA"

Plotting different features prior to imputation is essential because it clarifies their correlations, which is necessary to determine the best imputation technique.

The columns for DC power and ambient temperature are ordered, continuous, and have sequential qualities, according to visual analysis. This suggests that patterns found in one feature can direct the imputation of another. The idea behind not dividing the dataset into train and test groups at random is to maintain sequences and use time stamp values from prior and subsequent iterations to fill in missing data. Visualization aids in confirming that the imputation process preserves the inherent patterns and trends in the data. Plotting also helps in finding anomalies or missing numbers because abrupt drops or gaps in the plotted lines can be signs of these problems as shown in Figure 4.

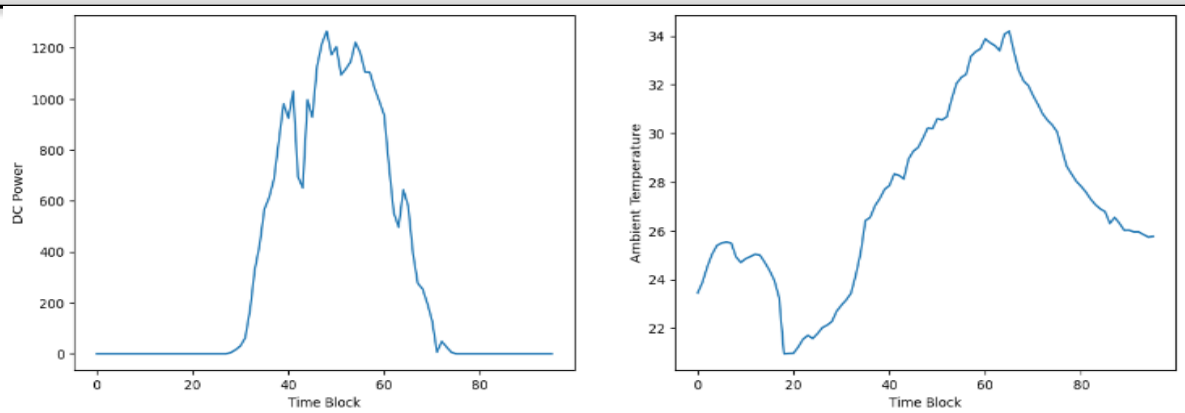

Figure 4: Visual Analysis of DC power and Ambient Temperature.

5.4 IMPUTATION STRATEGY

5.4.1 BLOCK-WISE IMPUTATION: In order to impute missing data in the testing set, we compute the mean, median, or mode for each time block from the training data. Instead of employing a global statistic, this approach makes sure that the imputation accurately represents the data patterns within particular time blocks.

5.4.2 MODEL-BASED IMPUTATION: It is possible to anticipate missing values by training a model only on non-missing values. However, this method can be laborious and costly to compute, which makes it less appropriate for usage in production.

5.5 SPLINE INTERPOLATION

Spline interpolation is a dependable method for imputing missing values by constructing a polynomial that aligns with the two nearest non-zero values. This approach is efficient for time series data, where preserving continuity and smoothness is essential.

Definition: Piecewise polynomials are used in spline interpolation to estimate missing values, preserving the data's general trend and pattern throughout the interpolation process.

Polynomial Order: To better suit the type of data, the polynomial's order might be changed. For instance, a quadratic polynomial (degree 2) is better at capturing more complex patterns, as those found in the generation of solar electricity, whereas a linear polynomial (degree 1) may be appropriate for simple trends as depicted in Figure 5.

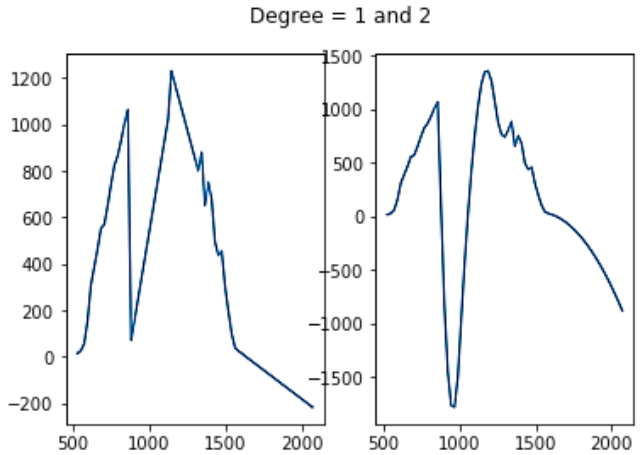

Figure 5: Comparison between degree 1 and degree 2 splines

Second-degree spline Imputation is significantly more parabolic in nature and much smoother. We currently have data compiled at the plant level, meaning that the 22 inverter values have all been added up and converted from kW to MW. Our training and test data are now prepared for further use.

5.6 ZERO IMPUTATION FOR NON-SOLAR HOURS: The values of variables like "solar irradiance" and "AC/DC" power generation are zero from 00:00 to 06:00 and 18:00 to 24:00 when there is no sun. Imputation for these periods should respect these zero values to avoid introducing inaccuracies. Spline interpolation should not be used for non-solar hours to prevent inappropriate non-zero values from being introduced as mentioned in Table 5. If all data points between 17:00 and 07:00 the next day are missing, spline interpolation might incorrectly fill non-solar hours with non-zero values due to the influence of adjacent solar hour data points. Instead, zero imputation is applied during non-solar hours to maintain data integrity.

Table 5: Imputation Methods by Data Type and Time

Data Type	Solar Hours (06:00 - 18:00)	Non-Solar Hours (00:00 - 06:00, 18:00 - 24:00)
Solar Irradiance	Linear Spline	Zero
Module Temperature	Linear Spline	Linear Spline
AC/DC Power	Polynomial Spline (Degree 2)	Zero

5.7 TRAINING AND TESTING SET CHARACTERISTICS

5.7.1 TRAINING SET (DAY 1 - DAY 22)

The training set comprises the initial 22 days of data, including critical temporal dynamics and weather variability, which are important for proficient model training.

Seasonal Trends: The training dataset is long enough to identify repetitive weekly patterns and trends in solar power generation.

Weather Variability: It covers a variety of weather scenarios, giving the model a thorough foundation for training.

5.7.2 KEY POINTS

Data Range: The training set encompasses an extensive timeframe, guaranteeing the inclusion of critical trends and patterns for model training as specified in Table 6.

Feature Engineering: To improve model efficacy, attributes such as moving averages, lagged variables, and seasonal indicators can be derived from the training data.

Table 6: Training dataset

	BLOCK	DATE	TIME	Unnamed: 0	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	DC_POWER_1	AC_POWER_1	Inverter_No_1	...
0	1	2020-05-15	00:00	0	27.004764	25.060789	0.0	0.0	0.0	1.0	...
1	2	2020-05-15	00:15	1	26.880811	24.421869	0.0	0.0	0.0	1.0	...
2	3	2020-05-15	00:30	2	26.682055	24.427290	0.0	0.0	0.0	1.0	...
3	4	2020-05-15	00:45	3	26.500589	24.420678	0.0	0.0	0.0	1.0	...
4	5	2020-05-15	01:00	4	26.596148	25.088210	0.0	0.0	0.0	1.0	...
...	...	...	...	...	...	...	...	...	...	...	...
2966	92	2020-06-14	22:45	2966	24.185657	22.922953	0.0	0.0	0.0	1.0	...
2967	93	2020-06-14	23:00	2967	24.412542	23.356136	0.0	0.0	0.0	1.0	...
2968	94	2020-06-14	23:15	2968	24.652915	23.913763	0.0	0.0	0.0	1.0	...
2969	95	2020-06-14	23:30	2969	24.702391	24.185130	0.0	0.0	0.0	1.0	...
2970	96	2020-06-14	23:45	2970	24.534757	23.921971	0.0	0.0	0.0	1.0	...

2971 rows × 73 columns

5.7.3 TESTING SET (DAY 23 - DAY 25)

The last three days' worth of data comprise the testing set, which is used to assess the model's performance on fresh data and make sure it can generalize well to new circumstances.

Prediction Horizon: The testing set represents a realistic prediction horizon, immediately following the training period, providing a stringent evaluation of the model's forecasting capabilities as illustrated in Table 7.

Continuity with Training Set: The testing set maintains continuity with the training data, ensuring that temporal dependencies are preserved.

5.7.4 KEY POINTS

Evaluation Metrics: The model's performance on the testing set is evaluated using metrics like mean absolute error (MAPE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

Scenario Testing: The testing set enables the model to be assessed in a variety of weather and power generating scenarios.

Table 7: Testing dataset

	BLOCK	DATE	TIME	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	AC_POWER
2971	1	2020-06-15	00:00	24.486876	23.846251	0.0	0.0
2972	2	2020-06-15	00:15	24.509378	23.902851	0.0	0.0
2973	3	2020-06-15	00:30	24.605338	24.172737	0.0	0.0
2974	4	2020-06-15	00:45	24.679791	24.459142	0.0	0.0
2975	5	2020-06-15	01:00	24.636373	24.380419	0.0	0.0
...	...	...	...	...	...	...	...
3254	92	2020-06-17	22:45	23.511703	22.856201	0.0	0.0
3255	93	2020-06-17	23:00	23.482282	22.744190	0.0	0.0
3256	94	2020-06-17	23:15	23.354743	22.492245	0.0	0.0
3257	95	2020-06-17	23:30	23.291048	22.373909	0.0	0.0
3258	96	2020-06-17	23:45	23.202871	22.535908	0.0	0.0

288 rows × 7 columns

6. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is a dataset evaluation method that highlights its key features, which usually involves visual methods. The step is crucial in data analysis because it uses visual and quantitative models to unearth anomalies, test hypotheses, and prove the assumptions. After exploratory data analysis (EDA) comes formal modeling or hypothesis testing. The analysis of different types of graphs, such as scatter plots or bar graphs, is done to acquire the final results that determine correlations, trends, and relationships between variables in the same way that assumptions of statistical models are vindicated (e.g., normality, linearity). They are performed using a training dataset.

6.1 PAIR PLOTS

Visualizing Relationships: Different environmental factors (like temperature and irradiation) impact the solar power output as reflected in Figure 6.

Identifying Trends and Patterns: Recognizing patterns that can help in developing predictive models.

Detecting Anomalies: Spotting outliers and anomalies in the data which might affect model performance.

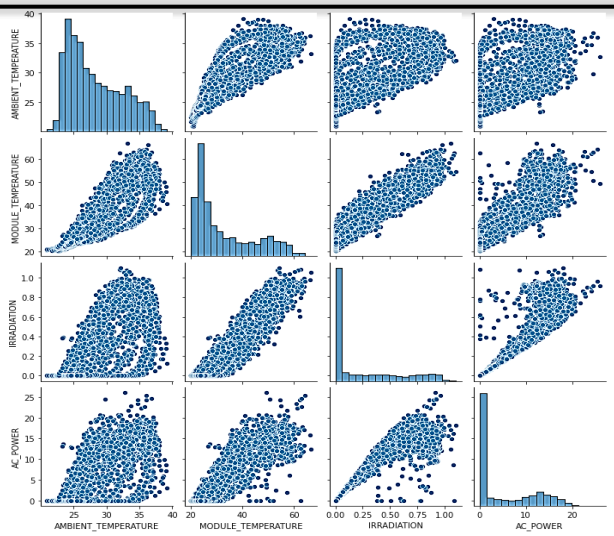

Figure 6: Pair plots

6.2 OBSERVATIONS

Feature Distributions in Histograms: The histograms reveal the distribution of each feature, such as irradiation, which tends to peak around midday, indicating higher solar intensity during those hours.

Strong Linear Relationship: There is a strong linear correlation between module temperature and irradiation, AC power and irradiation.

Skewed Distributions: Due to 0 values during non-generating hours (6 pm to 6 am), the distributions of AC/DC power and irradiation are severely right-skewed. Module and ambient temperatures are less distorted.

Growing Variability with Ambient Temperature: The temperature of the module varies more with an increase in the surrounding air temperature. This variability indicates that additional meteorological variables that are not included in the dataset, such as humidity, wind speed, and precipitation, may have an impact.

Anomalies in AC Power: There are anomalies present in the AC power data.

6.3 BOX PLOTS

After creating pair plots for Exploratory Data Analysis (EDA), using boxplots can provide additional insights into the data by allowing us to:

Find and visualize outliers: Boxplots are the most natural way to find an outlier in data. They clearly show data beyond the end of what's called whiskers, whereas pair plots simply allude to the existence of outliers (generally 1.5 times the interquartile range above the third quartile or below the first quartile).

Learn the Spread and Skewness: Boxplots conveniently show the skewness, dispersion, and middle of the data. Hence, they help understand data distributions more than histograms do, even with skewed data.

Compare Grouped Distributions: Boxplots are easily used to compare the distributions of several groups or categories. For example, you can compare the AC power output of different irradiation or ambient temperature ranges.

Identify Median, Quartiles, and Range: Boxplots show the median, quartiles, and overall range of the data, providing a clear summary of these statistical measures. For example, the box plot of ambient temperature, module temperature are shown in Figure 7 and Figure 8.

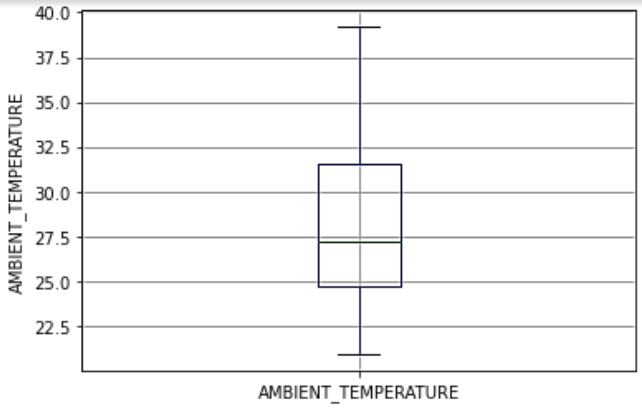

Figure 7: Ambient Temperature Box Plot

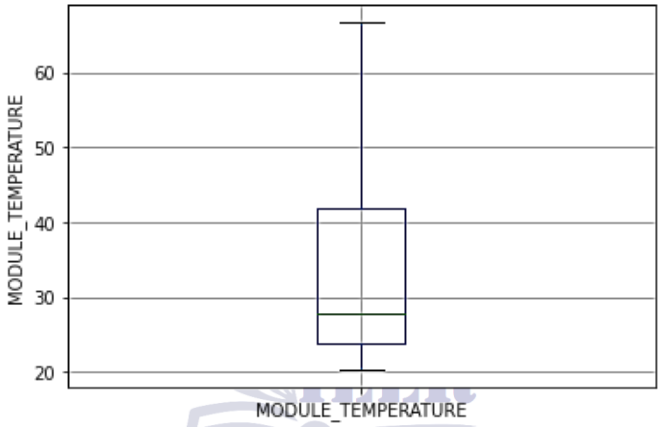

Figure 8: Module Temperature Box Plot

6.4 OBSERVATIONS

Outliers: Identify specific data points that are outliers, which can be further investigated or potentially removed if they are errors. **Spread and Central Tendency:** Observe the median and how data is spread around it. For example, in the AC power boxplot, you can see if most values are concentrated near the median or if they are spread out.

Skewness: Determine if the data is skewed. Right-skewed data will have a longer whisker on the right side and more outliers on the high end.

Comparison Across Features: Compare the spread and central tendency of different features. For instance, you might find that module temperature has a wider interquartile range compared to ambient temperature, indicating greater variability.

6.5 HEAT MAP

A heat map is a type of data visualization that shows a phenomenon's magnitude in two dimensions as color. A heat map is commonly used in data analysis to display the correlation between several variables in a dataset as depicted in Figure 9. The correlation strength is indicated by the color's intensity.

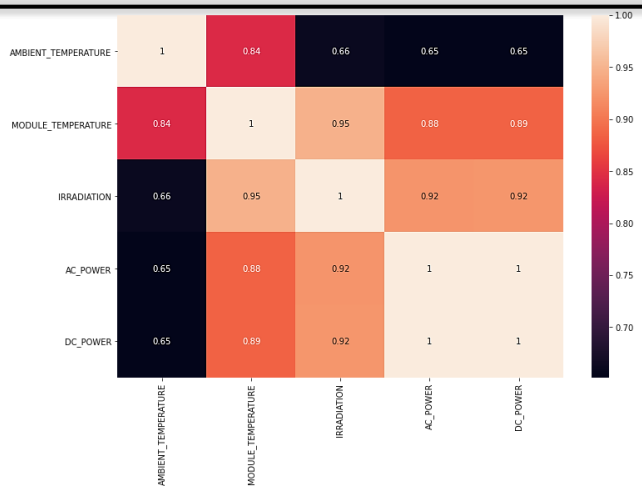

Figure 9: Heat map

6.6 OBSERVATIONS

Correlation Heat map Validation: Our previous discovery of a high link between the variables is supported by the correlation heat map. DC and AC have a perfect correlation.

Irradiation and Power Correlation: There may have been some power loss during the conversion process because the correlation between radiation and DC power is marginally larger than the correlation between radiation and AC power. To further examine this, inverter-wise conversion analysis may be performed using the previously mentioned df_train dataset.

Ambient Temperature Influence: The relationship between ambient temperature and (AC power or irradiation) is comparatively smaller, suggesting that ambient temperature has less of an impact on power generation predictions.

Even though the analysis and popular consensus both indicate that irradiation is the most important component in solar power generation, we will analyze all other parameters except DC power in more detail. When predicting future time blocks in real-world circumstances, very accurate anticipated meteorological data might not always be available. Over-reliance on a small number of features could have a big effect on how well the model performs.

7. HANDLING OUTLIERS

Data points that significantly diverge from the other observations in a dataset are termed outliers. They may stem from inaccuracies or data variability, as they significantly exceed the typical range of the dataset. Various causes can be referred to as extreme, such as measurement inaccuracies, errors in data input, or inherent randomness in the data.

It is necessary to remove outliers as a preprocessing method. The existence of outliers affects the performance of models. Research has shown that it has a considerable effect in distance-based distance-based models; some of these models include k-means clustering and linear regression.

7.1 METHODS FOR HANDLING OUTLIERS

7.1.1 DATA DISTRIBUTION

Percentiles Analysis: By examining the percentiles of each feature, we can understand the distribution and spread of the data. This is crucial for features like ambient temperature, module temperature, and irradiation, which directly influence solar power generation as illustrated in Table 8.

Tail Behavior: The 1st and 99th percentiles help us understand the behavior of the data in the tails, which might include extreme values or outliers. This is important for efficient model training.

7.1.2 DETECTING OUTLIERS

Extreme Values: Extreme percentile values may distort the analysis and influence the model's

performance. They can be detected by identifying extreme percentile values. Knowing the location of these outliers enables better preprocessing, elimination, or correction.

The percentile values for all features are given as:

Table 8: Percentile Values for all Features

	Percentiles	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	AC_POWER
0	1	22.55	21.10	0.00	0.00
1	10	23.66	22.40	0.00	0.00
2	25	24.73	23.76	0.00	0.00
3	50	27.24	27.79	0.02	0.46
4	75	31.55	41.75	0.46	11.23
5	90	34.61	52.28	0.80	15.28
6	99	37.40	60.32	0.96	19.54
7	100	39.18	66.64	1.10	25.98

7.1.3 IMPUTING OUTLIERS

Any data point that has exceeded the 99th percentile and occurs lower than the 1st percentile will be replaced by the 99th percentile data point and the 1st percentile, respectively.

Values below the 1st percentile are considered extremely low outliers.

Values above the 99th percentile are considered extremely high outlier

From the given table:

Any **ambient temperature** below 22.55 or above 37.40 is an extreme outlier.

Any **module temperature** below 21.10 or above 60.32 is an extreme outlier.

Any **irradiation** value above 0.96 is an extreme outlier.

Any **AC power** value above 19.54 is an extreme outlier.

To help with the replacement of outliers in the training and test datasets, we'll construct a dictionary including these percentile values from the training dataset. It is possible to preserve and utilize this dictionary again for upcoming uses.

The dictionaries store the 1st and 99th percentile values for each feature, which serve as thresholds to identify outliers. The final values are shown in Figure 10.

```
AMBIENT_TEMPERATURE {'99th': 37.39861080172413, '1st': 22.546186481034486}
MODULE_TEMPERATURE {'99th': 60.32299286206897, '1st': 21.099540196666666}
IRRADIATION {'99th': 0.9607427385333338, '1st': 0.0}
AC_POWER {'99th': 19.544968952380955, '1st': 0.0}
DC_POWER {'99th': 19.993737333333335, '1st': 0.0}
```

Figure 10: Final Percentile Values.

At last, the dataset is clean and fully prepared for model creation.

8. MODEL CONSTRUCTION

8.1 DATA PREPARATION

The initial data preparation phase for model training is partitioning it to provide an equitable representation of various time intervals (bins). The

data are divided into testing and training datasets. It is essential for developing a resilient model that excels across all temporal intervals. Classify the data into intervals according to periods to guarantee that each training fold possesses a balanced representation. Bins will be allocated to each row based on the corresponding Block No. Each Block serves as a timestamp; hence, each bin will correspond to a specific time of day. Blocks 0–12 correspond to BIN1, 13–24 correspond to BIN2, and Blocks 85–96 correspond to BIN8, constituting the two blocks. There are a total of eight bins. Having categorized our data into bins, we are now ready for training.

8.2 DATA TRAINING

To improve code readability, maintainability, and ensure consistency, we have organized the datasets into a clear sequence of steps using Scikit-learn's convenient 'Pipeline' functionality. This allows us to easily wrap and train different models. We have utilized 7 different regression algorithms with default parameters. Start with simple baseline models to establish a benchmark for performance. To compare their performance and choose the best one with the least amount of error, implement a variety of regression algorithms, including Multi-layer Perceptron (MLP) Regressor, Decision Tree Regressor, Random Forest Regressor, Support Vector Regressor, Gradient Boosting Regressor, XG Boost Regressor, and (3-layer Neural Network).

8.3 SELECTION OF PERFORMANCE MATRIC

Selecting the appropriate performance metric is essential. The following metrics are frequently used for regression problems: (a) Mean Absolute Error (MAE), (b) Mean Squared Error (MSE), and (c) Root Mean Squared Error (RMSE).

Because RMSE penalizes greater errors more than smaller ones, we shall utilize it instead of MAE and MSE. Larger errors are amplified by the squared error terms in both MSE and RMSE. This motivates the model to more efficiently remove major errors.

8.4 ASSESSING AND CHOOSING MODELS WITH STRATIFIED K-FOLD CROSS VALIDATION

A useful method for selecting and evaluating models is k-fold cross validation. To take things a step further, we'll employ stratified k-folds that are dependent on the BINS column. This will assist us in obtaining the same bin distribution for each fold.

1. Split df_train into into 8 folds.
2. Use 7 folds for training (xtrain, ytrain), 8th fold for validation (xvalid, yvalid).
3. Standardize the xtrain & xvalid generated in step 2.
4. Fit xtrain, ytrain on the model.
5. Predict on xvalid, find the RMSE value and store in a list.

To obtain the RMSE for eight iterations, repeat steps 1 through 5 eight times. Then, determine the mean of the list containing RMSE scores. For every model in the Pipeline list, repeat steps 1-6, and compare the outcomes as indicated in Figure 11.

```
Mean Validation RMSE for Linear Regression: 2.3738425616651613
Mean Validation RMSE for Decision Tree Regressor: 2.3598920923506563
Mean Validation RMSE for Random Forest Regressor: 1.7386875230088275
Mean Validation RMSE for Ridge Regressor: 2.3774000818909666
Mean Validation RMSE for Lasso Regressor: 2.792532888972481
Mean Validation RMSE for XG Boost Regressor: 1.8680211963786664
Mean Validation RMSE for ANN Regressor: 3.376760114293983


Regressor with least RMSE:  Random Forest Regressor
Pipeline(steps=[('rf_regression', RandomForestRegressor(random_state=0))])
```

Figure 11: Mean Validation RMSE

8.5 HYPERPARAMETER OPTIMIZATION

The methods involved in this hyperparameter optimization process include data splitting, defining a hyperparameter grid, using 'Randomized Search CV' for hyperparameter tuning, and then training a Random Forest Regressor model using the best

parameters found from the search. Using 'Randomized Search CV', the algorithm explores random combinations of these parameters, evaluating each through cross-validation to identify the set that maximizes predictive performance. The final model (rf_model') is expected to offer improved accuracy and robustness when applied to unseen data, as its hyperparameters have been fine-tuned to extract meaningful patterns from the training data. Ultimately, this methodological approach helps in achieving better predictive outcomes and mitigating over-fitting, thus enhancing the model's effectiveness in real-world applications. Once the best parameter is selected, train the model on that.

8.6 MAKING PREDICTIONS

We have 3 days of datasets for testing. Imputation of missing data and removal of outliers has been done earlier. Now prediction is performed.

Here, x_test includes the features 'AMBIENT TEMPERATURE', 'MODULE TEMPERATURE', and 'IRRADIATION') from the test dataset and y_test includes the target variable ('AC_POWER') from the same dataset. The model has been trained and is used to make predictions on test data.

We utilize Root Mean Squared Error (RMSE), a popular regression task statistic, to evaluate the model's performance on the test data. The prediction errors' average magnitude is measured by RMSE, which shows how near the actual values are to the expected values as illustrated in Figure 12.

```
Root Mean Squared Error for Test Data: 1.86830155234851
```

Figure 12: Testing Data RMSE

Although the RMSE for the test set is marginally higher (1.7386) than it was for the training data, this is still acceptable given the smaller volume of training data. Better model performance is indicated by a smaller root mean square error (RMSE), which shows that the predicted and actual values are more similar. We may assess the model's capacity to generalize and make accurate predictions about future data by comparing it to test data.

9. SIMULATION RESULTS

For testing, we saved the latest three days' worth of data. The model did not view this data at all, and there was no "data leakage" of any type. On this, outlier removal and missing value imputation have been completed independently.

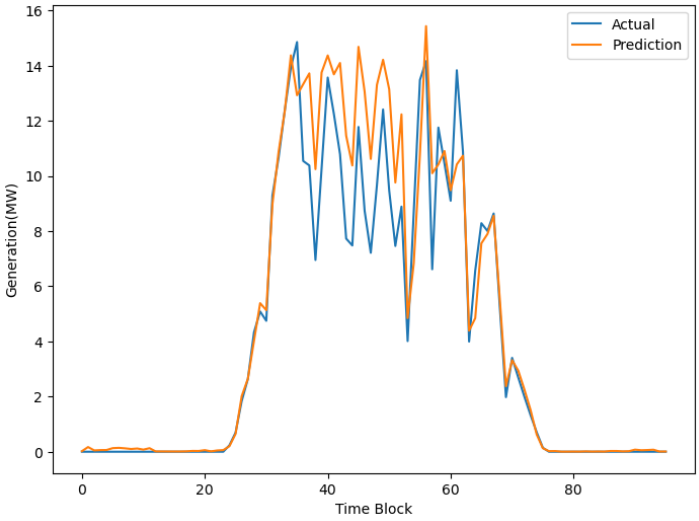

Figure 13: Actual Vs Predicted Output for Day 1

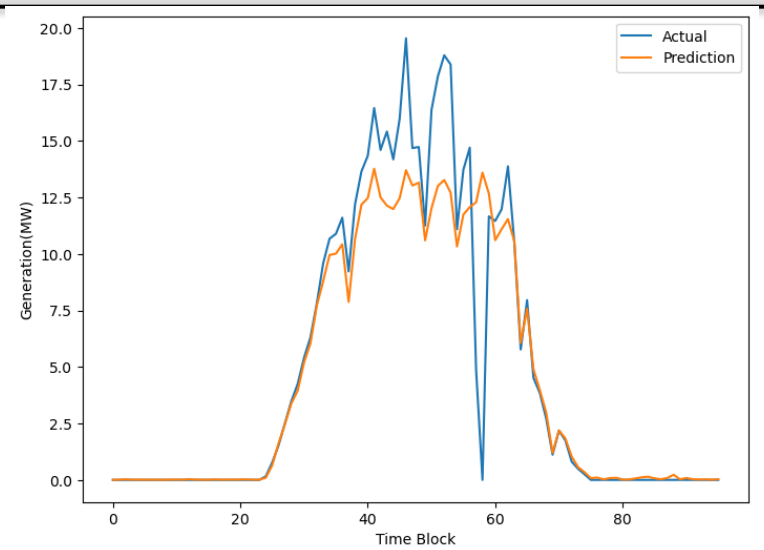

Figure 14: Actual Vs Predicted Output for Day 2

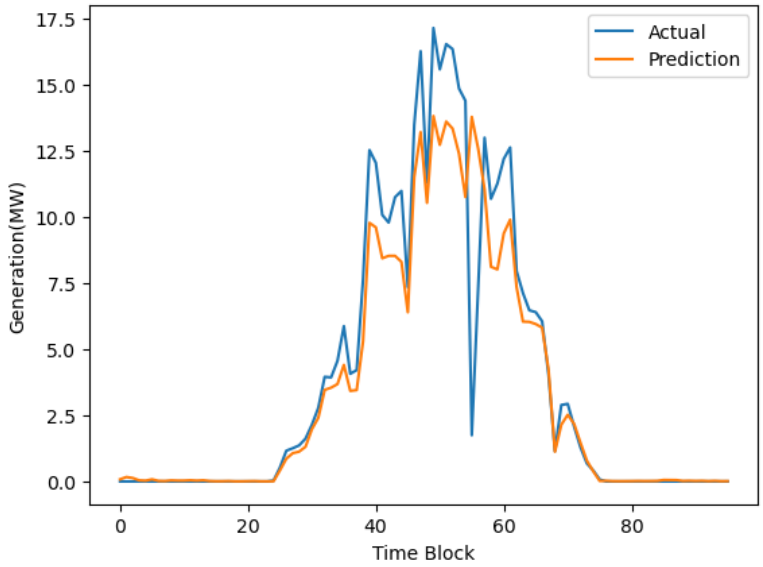

Figure 15: Actual Vs Predicted Output for Day 3

Figures 13, 14, and 15 show the power generation forecasts (MW) within three days. This indicates a high variability of data, with significant changes in the actuals across the three days. Figure 13 shows that the actual values closely coincide with the other forecasts, but the predictions (orange marks) fall higher than the actual values in the central part of the graph. Hence, the model might be good at predicting generally but sometimes may overpredict.

In Figure 14 and Figure 15, the predicted curves are somewhat underfitting the real curves. However, there is also misfitting that does not exceed the first band of deviation without penalty, which in this case is in the range of 7.4, which is acceptable. Thus, the model performance is somewhere in an acceptable range, but it may be improved to increase accuracy.

Figure 14 and Figure 15 show two anomalous dips in the actual value (in blue). Such dips have been explained to be caused by faulty data, where very weak power production occurs, even with high irradiation levels. This implies that data quality concerns influence the accuracy of the model, and it ought to be improved to enhance predictive accuracy.

Various shortcomings make forecasting solar power generation in real time a challenge. The forecasts are based on already inaccurate weather predictions, and the unpredictability of weather only adds to the problem. This is because flowing clouds may be difficult to detect in rainy weather, leading to a significant deviation in weather. Even more, when real-time data are received in the plant rather late, precise forecasting becomes more complicated.

The solar and wind forecasts are also limited to several revisions per day, which are 8 and 14, respectively, and any revision is effective after 45 minutes of submission and then locked again in 1.5 hours in certain states. The time between such moments also differs in different regions; in some regions, it takes 45 minutes; in other areas, it takes 30 minutes; and in other regions, it could take 60 minutes, depending on the region. Hence, it may also create an obstacle in our operation, whereby all the regions will not experience the same interval to make the revisions. Such constraints also drive the need for strong models because over-fitting models would collapse in a dynamic environment. The techniques adopted to make the model more resistant and more accurate in dealing with these issues include moving averages, exponentially weighted moving averages, bins-specific adjustments, and plant-specific feature engineering. Such issues show a dire necessity for conducting extensive research in AI and ML in weather prediction and renewable energy applications.

Forecast accuracy can be enhanced in future design modeling by increasing details and comprehensiveness of data by providing more training datasets and time stamps (15 minutes) to allow the model to pick up finer details and trends in generating solar power. Finally, the findings reveal that notwithstanding the above-mentioned problems and proposed implementations in the future, the case study considered in this paper suggests that when the model is comprehensively tested and K-fold cross-validation is used, the performance of the random forest algorithm outperformed all the other ones by attaining higher accuracy in prediction of solar energy.

14. CONCLUSION

This research addresses significant issues concerning inadequate geographical generalization, real-time data integration, computational efficiency, and model performance by proposing an efficient machine-learning model to predict solar radiation. The model provided better prediction and simplified itself using Recursive Feature Elimination (RFE), Random Forest Regression (RFR), and real-time weather data. Due to real-time data integration, an increase in adjustments to changing weather conditions was achieved, and, as Redmine portrayed, RFR proved to be more efficient than a historical means of analysis, such as a traditional regression. Future developments should target an improvement of the dataset diversity, the implementation of deep learning structures, the improvement of real-time deployment, and uncertainty quantification.

REFERENCES

[1] A. Al-Sarraj and F. Yigit, "Modelling the use of PVSYST software for a stand-alone PV solar system 'off grid' with batteries by utilizing silicon hetero-junction technology (HJT) panels in Iraq/Basra," pp. 32-42, 2024.

[2] J. Gaboitaolelwe, A. M. Zungeru, A. Yahya, C. K. Lebekwe, D. N. Vinod and A. O. Salau, "Machine Learning Based Solar Photovoltaic Power Forecasting: A Review and Comparison," in IEEE Access, vol. 11, pp. 40820-40845, 2023

[3] C. Vennila, K. Gokulakrishnan, A. Kandasamy, S. P. Pandey, and P. T. K. Reddy, "Forecasting Solar Energy Production Using Machine Learning," International Journal of Photoenergy, vol. 2022, 7 pages, 2022.

[4] K. Anuradha, D. Erlapally, G. Karuna, V. Srilakshmi, and K. Adilakshmi, "Analysis of Solar Power Generation Forecasting Using Machine Learning Techniques," E3S Web of Conferences, vol. 309, no. 01163, 7 pages, 2021.

[5] A. Sharma, R. C. Bansal, and N. Kumar, "Solar Energy Forecasting Using Deep Learning Techniques," Springer Nature Singapore Pte Ltd., 2021.

[6] S. Ungureanu, V. Ţopa, and A. Cziker, "Industrial Load Forecasting Using Machine Learning in the Context of Smart Grid," in 54th International Universities Power Engineering Conference (UPEC), 6 pages, 2019.

[7] M. Ali, M. H. Mohamed, A. Alashwali, M. Alfarraj, and M. Khalid, "Machine Learning Based Solar Power Forecasting Techniques: Analysis and Comparison," in IEEE PES 14th Asia-Pacific Power and Energy Engineering Conference (APPEEC), 6 pages, 2022.

[8] P. Singh, N. K. Singh, and A. K. Singh, "Solar Photovoltaic Energy Forecasting Using Machine Learning and Deep Learning Technique," in 9th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 7 pages, 2022.

[9] M. S. Nikitha, K. C. R. Nisha, M. S. Gowda, P. Aithal, and N. M. Mudakkayil, "Solar PV Forecasting Using Machine Learning Models," in Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), 6 pages, 2022.

[10] E. Subramanian, M. Karthik, G. P. Krishna, D. V. Prasath, and V. S. Kumar, "Solar Power Prediction Using Machine Learning," 7 pages, 2023.

[11] H. A. Khan, M. Alam, H. A. Rizvi, and A. Munir, "Solar Irradiance Forecasting Using Deep Learning Techniques," in *IEEC*, 6 pages, 2023.

[12] A. Alzahrani, P. Shamsi, C. Dagli, and M. Ferdowsi, "Solar Irradiance Forecasting Using Deep Neural Networks," in *Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems (CAS)*, 10 pages, 2017.

[13] Y. Zahraoui, T. Korõtko, S. Mekhilef, and A. Rosin, "ANN-LSTM Based Tool for Photovoltaic Power Forecasting," in *4th International Conference on Smart Grid and Renewable Energy (SGRE)*, 7 pages, 2024.

[14] S. Rana and P. Kumar, "Ensemble Methods for Improving Solar Power Forecasting Accuracy," *IEEE Transactions on Renewable Energy*, vol. 12, no. 4, pp. 1103-1110, Oct. 2017.

[15] L. Wang, F. Zhou, and J. Liu, "Optimizing Neural Networks with Genetic Algorithms for Solar Power Forecasting," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 988-995, Mar. 2017.

[16] A. Patel and M. Shah, "Improving Solar Power Forecasting with Meteorological Data Integration," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 1, pp. 17-26, Jan. 2018.

[17] B. Li, C. Chen, and Y. Gao, "Solar Power Forecasting Using LSTM Networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1995-2003, May 2018.

[18] J. Zhang and K. Sun, "Probabilistic Solar Power Forecasting with Bayesian Neural Networks," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3219-3227, Jul. 2018.