Predicting Solar Energy Generation using Time Series and Machine Learning Models

by

Ayse B Sengul

July 9, 2024

Table of Contents

1 Introduction

Solar energy is a renewable and abundant source of energy that offers a promising alternative to fossil fuels by significantly reducing greenhouse gas emissions [1]. Incorporating solar energy in energy production is estimated to cut carbon dioxide emissions by approximately 32%. Photovoltaic (PV) cells, which convert sunlight into electricity, are central to this process and have the advantage of generating electricity without emitting harmful pollutants [2]. This makes solar energy a clean and environmentally friendly option, crucial for improving air quality and combating climate change.

The construction and operation of solar energy systems are influenced by various environmental factors such as solar irradiance, temperature, humidity, and wind speed. These factors' variability poses challenges for accurately predicting power production, impacting the electric grid's reliability, stability, planning, and operations [4-8]. Therefore, developing accurate forecasting models for PV energy generation is essential to mitigate these challenges and ensure efficient grid operation. Even minor improvements in prediction accuracy can lead to significant cost savings, highlighting the financial benefits of enhanced forecasting models. For instance, a 25% improvement in accuracy could result in a 1.56% reduction in generation cost, translating to approximately US$ 46.5 million [9]. This highlights the substantial financial impact of enhancing the accuracy of solar power generation predictions

Various methods have been documented in the literature for forecasting PV energy, which can be categorized into four classes: (i) physical, (ii) empirical, (iii) statistical and (iv) machine learning models. Physical models are based on numerical weather prediction and satellite imagery. Empirical models develop linear or nonlinear regression equations. Statistical models, like autoregressive moving average (ARMA), the autoregressive integrated moving average (ARIMA), are developed based on statistical correlations. Machine learning models, like artificial neural networks (ANNs) and support vector machines (SVM), based on machine learning approaches [6, 10, 11].

This project aims to predict solar energy generation using advanced analytical methods. The specific objectives include:

1. Analyzing historical data on solar energy production and relevant weather conditions.
2. Developing forecasting models using time series analysis techniques like ARIMA and SARIMA.
3. Incorporating machine learning algorithms such as random forests, gradient boosting.

4. Evaluating the models' performance using metrics like root mean squared error and mean absolute percentage error.

The dataset that is used in this study taken from Solar Power plant Dataset, which contains data from two separate solar power plants located in India, with a particular focus on Plant 2. Solar power generation data gathered at 15 minutes intervals over a 34-day period.

2 Data Wrangling

The solar power generation and weather sensor data were loaded from CSV files into Pandas DataFrames named 'generation_df' and 'weather_df'. The generation data consisted of 67,698 rows and 7 columns, while the weather data comprised 3,259 rows and 6 columns. Initial exploration of the generation data revealed columns such as 'DATE_TIME', 'PLANT_ID', 'SOURCE_KEY', 'DC_POWER', 'AC_POWER', 'DAILY_YIELD', and 'TOTAL_YIELD'. Notably, rows with zero values for 'DC_POWER' and 'AC_POWER' indicated nighttime or no power generation. Similarly, the weather data included columns like 'DATE_TIME', 'PLANT_ID', 'SOURCE_KEY', 'AMBIENT_TEMPERATURE', 'MODULE_TEMPERATURE', and 'IRRADIATION', with zero 'IRRADIATION' values representing nighttime or lack of solar irradiance.

For the generation data, the 'DC_POWER' and 'AC_POWER' columns had mean values of approximately 246.7 kW and 241.3 kW, respectively, with significant variability. 'DAILY_YIELD' averaged 3,294.9 kWh, peaking at 9,873 kWh, and 'TOTAL_YIELD' represented cumulative energy production up to 2.25 billion kWh. In the weather data, 'AMBIENT_TEMPERATURE' averaged 28.07°C, ranging from 20.94°C to 39.18°C, while 'MODULE_TEMPERATURE' had a mean of 32.77°C, spanning from 20.27°C to 66.64°C. 'IRRADIATION' averaged 0.233 kW/m², with a maximum value of 1.099 kW/m².

Both datasets were free of missing values and duplicates, with only one unique 'PLANT_ID' value confirming that all data pertained to Plant 2. The generation data was collected from 22 inverters, while the weather data came from a single weather sensor array. The 'DATE_TIME' column was converted to datetime format, and additional columns such as 'DATE', 'TIME', 'MONTH', 'HOURS', and 'MINUTES' were extracted for detailed time series analysis. Redundant columns such as 'PLANT_ID' in both datasets and 'SOURCE_KEY' in the weather data were dropped. The generation and weather datasets were merged based on the 'DATE_TIME' column. The complex 'SOURCE_KEY' categorical labels in the merged data are converted into numerical values named 'INVERTER' to simplify analysis and visualization. The original 'SOURCE_KEY' column was then dropped.

3 Exploratory Data Analysis (EDA)

During the EDA process, histograms were created for numerical columns to visualize the distribution of values, showing typical power generation patterns with daytime peaks and nighttime zero values. Kernel Density Estimation (KDE) plots were then used to smooth these distributions, revealing underlying patterns more clearly. Figure 1 show the distribution of numerical columns before filtering out zero values. DC_POWER and AC_POWER variables displayed significant right-skewness, indicating low power generation during nighttime or low irradiation conditions. DAILY_YIELD and TOTAL_YIELD showed similar right-skewed patterns. AMBIENT_TEMPERATURE had a normal distribution peaking around 27.5°C, while MODULE_TEMPERATURE and IRRADIATION were right-skewed, with most values between 20°C to 40°C and 0 to 1 kW/m², respectively. Data collection was consistent across days, weeks, and hours, suggesting uniform sampling.

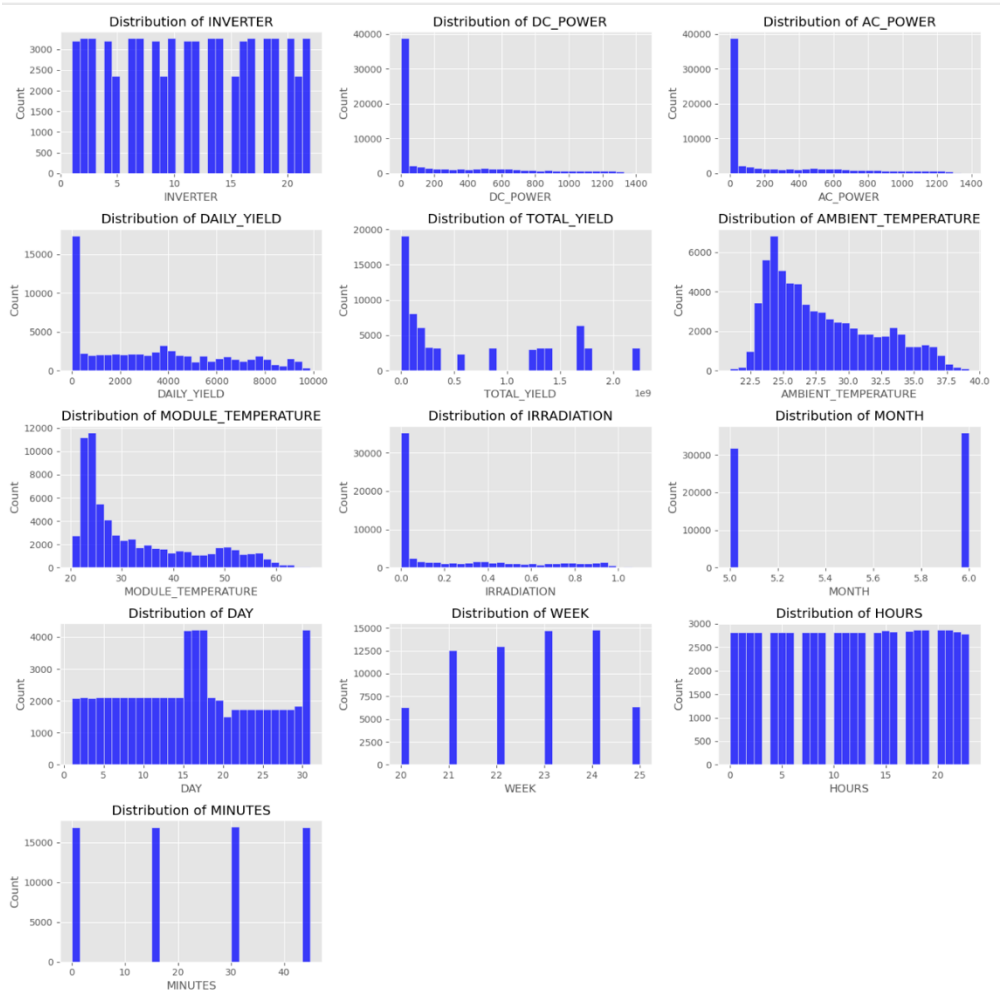

Figure 1. Distribution of numerical columns before filtering out zero values.

After filtering out zero values, the distributions of DC_POWER and AC_POWER improved but remained right skewed. DAILY_YIELD showed a more continuous range. Other variables retained their distributions, and data collection remained balanced throughout the day (Figure 2).

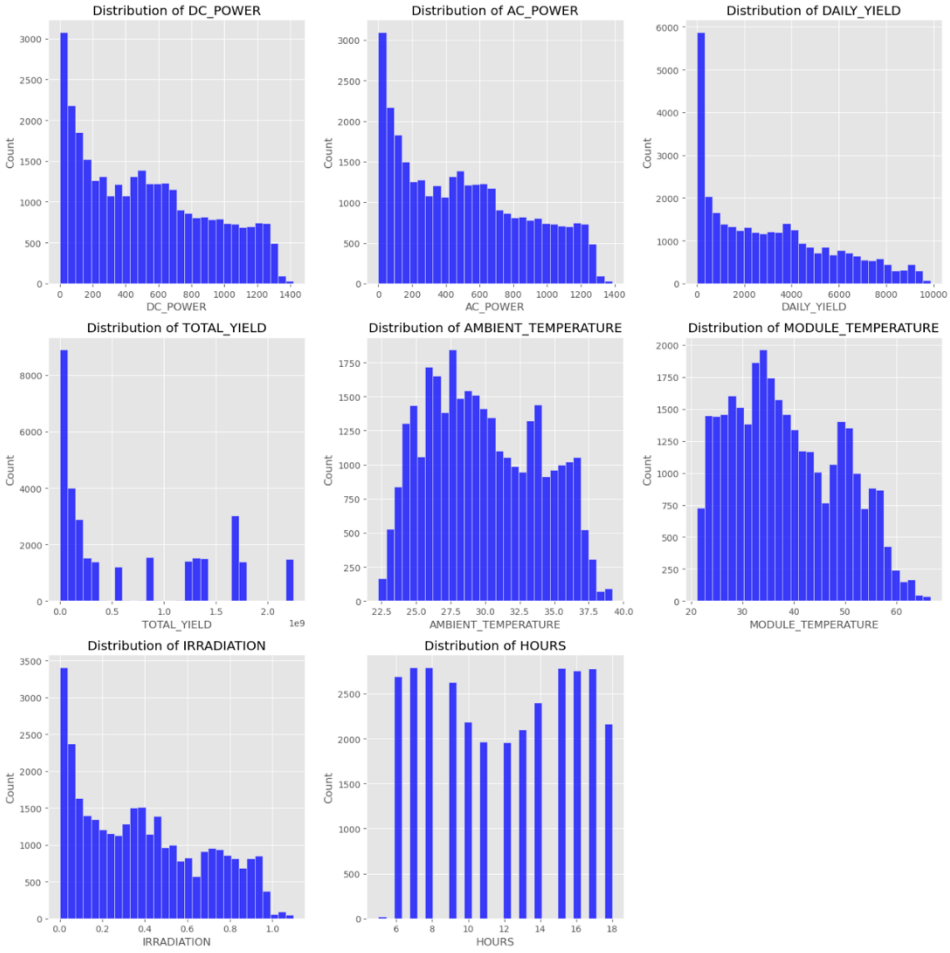

Figure 2. Distribution of numerical columns after filtering out zero values.

Box plot analysis identified outliers in DC_POWER, AC_POWER, MODULE_TEMPERATURE, and IRRADIATION, suggesting occasional extreme conditions in power generation and environmental factors.

Pair plots explored relationships between numerical features, while a correlation matrix heatmap (Figure 3) and detailed scatter plots (Figure 4) provided deeper insights. Key findings included a perfect correlation between DC_POWER and AC_POWER, indicating a direct relationship, and high correlations between AC_POWER and IRRADIATION, AMBIENT_TEMPERATURE and MODULE_TEMPERATURE, and MODULE_TEMPERATURE and

IRRADIATION. Additionally, a moderate correlation between HOURS and DAILY_YIELD reflected the daily cycle of solar radiation and power generation.

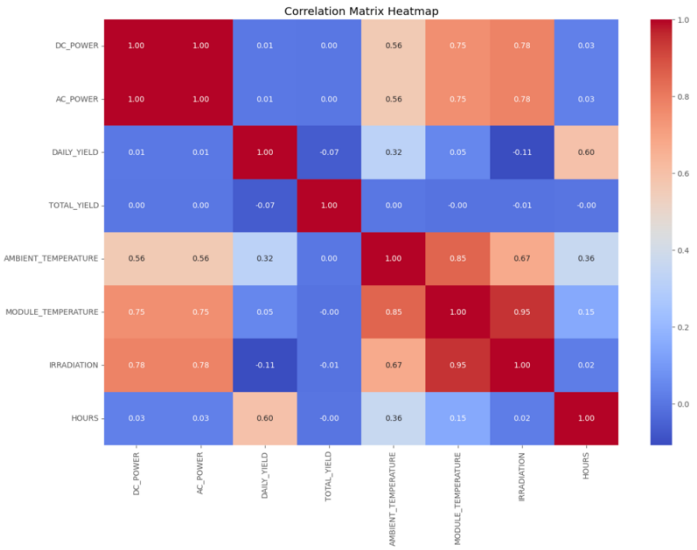

Figure 3. Correlation matrix heatmap.

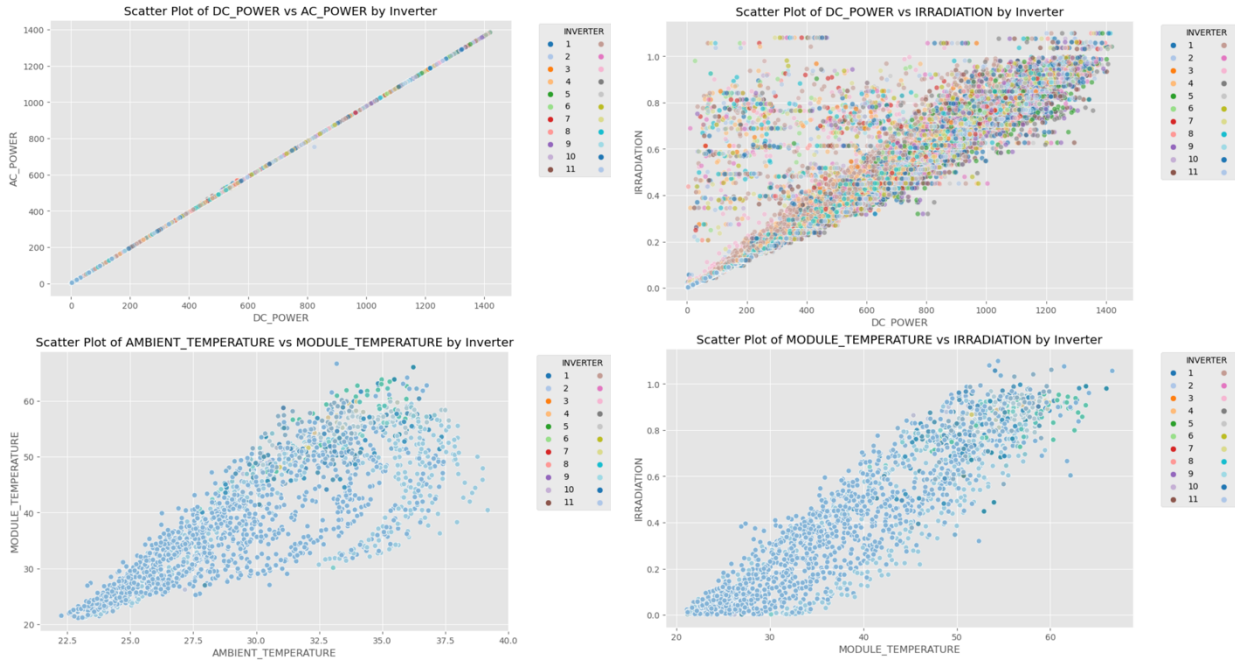

Figure 4. Scatter plot of strongly correlated features.

To visualize changes over time, time series plots for AC_POWER, IRRADIATION, and MODULE_TEMPERATURE were created (Figure 5). These plots revealed daily cyclical patterns, peaking during the day and dropping to zero at night, with a slight overall decrease over the months due to seasonal effects. AC_POWER and IRRADIATION exhibited strong diurnal

patterns, while MODULE_TEMPERATURE followed a similar pattern, influenced by sunlight exposure. Variability in daily values was likely due to changing weather conditions.

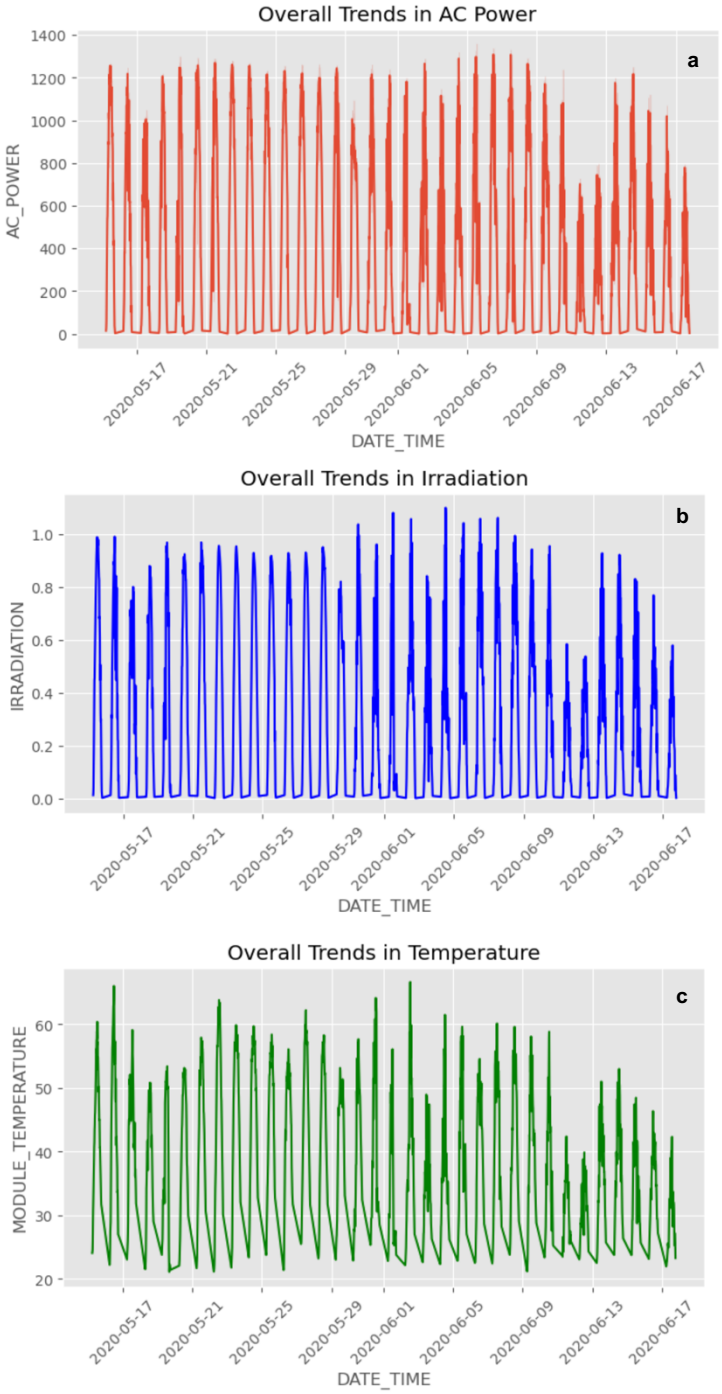

Figure 5. Time trend in a) AC_POWER, b) IRRADIATION, and c) MODULE_TEMPERATURE.

Further analysis evaluated AC_POWER generation for various days, revealing peaks between 10 AM and 2 PM, corresponding to optimal sunlight. Clear, sunny days such as May 15 and 20 show smooth single-peaked curves, while days like June 1 and 11 display multiple peaks, suggesting variable weather. May 15 recorded the highest AC_POWER generation, while June 11 had the lowest average (Figure 6).

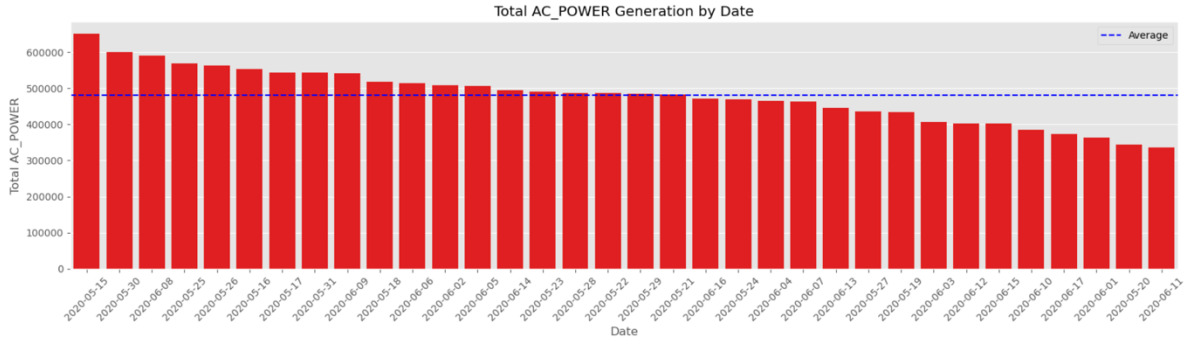

Figure 6. Total AC_POWER generation by date.

Solar irradiation data for different days showed peaks around 10 AM to 2 PM, with smooth curves on clear days and multiple peaks on variable weather days. The highest irradiation occurred on May 15, 2020, and the lowest on June 11, 2020. These patterns closely mirrored AC_POWER generation trends (Figure 7).

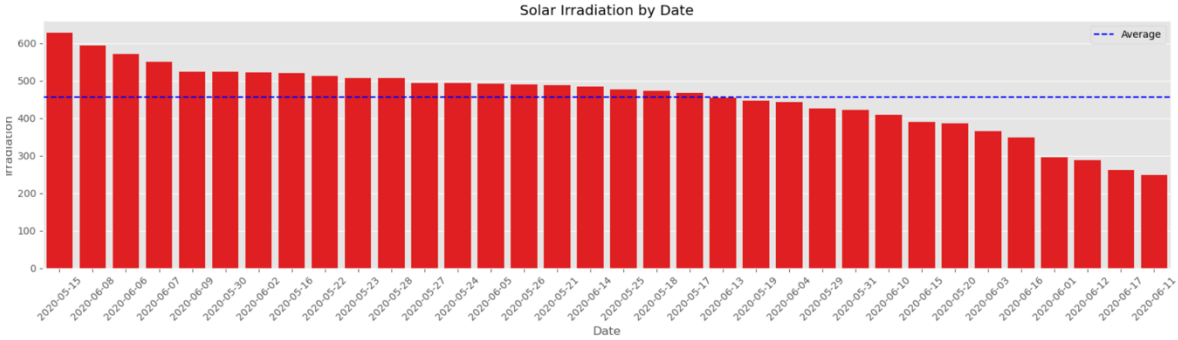

Figure 7. Solar irradiation by date.

Module temperatures typically peak between 10 AM and 2 PM, reflecting sunlight intensity. Days like May 15, 20, and 24 show smooth temperature curves, indicating stable weather. In contrast, days such as May 19 and June 3 display variable patterns due to clouds. The highest average daily module temperature was on May 24, indicating high solar irradiance and ambient temperatures, which could negatively affect solar panel efficiency due to overheating. The lowest average daily module temperature was on June 17, suggesting lower solar irradiance (Figure 8).

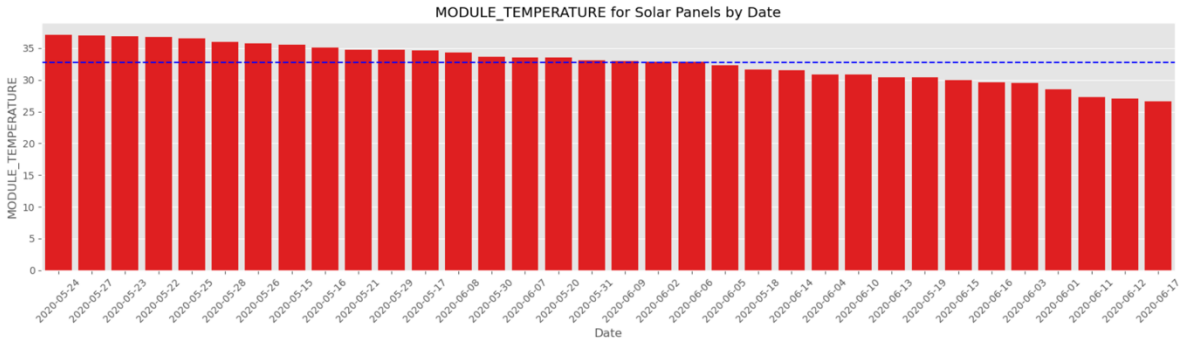

Figure 8. Module temperature for solar panels by date.

Lastly, the average efficiency for each inverter was calculated and analyzed. The efficiency values across all inverters are consistently around 97%, indicating reliable performance in converting DC to AC power. The highest recorded efficiency is approximately 97.7%, while the lowest is around 97.6%. These minor variations may result from factors such as inverter locations, installation differences, age, or environmental conditions.

Figure 9 shows periodic fluctuations in efficiency for all inverters, influenced by factors such as changes in solar irradiation, temperature variations, and other environmental conditions. Some inverters exhibit sudden decreases or increases in efficiency, which then quickly return to normal levels. These anomalies could indicate temporary issues like partial shading, temporary faults, or data recording anomalies. (Figure 9).

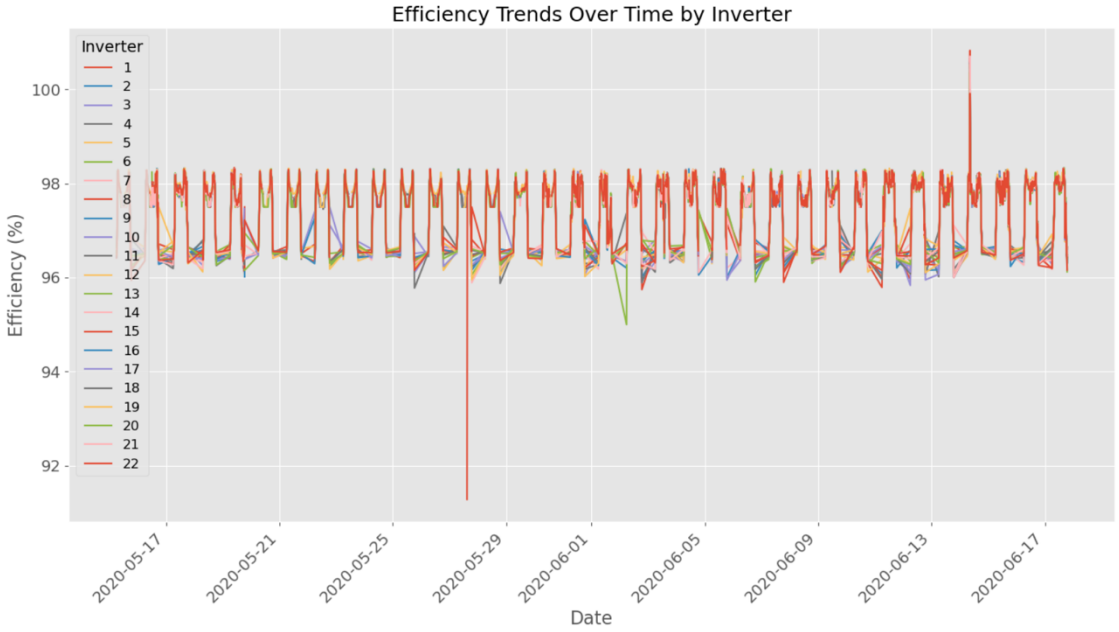

Figure 9. Efficiency Trends over Time.

4 Pre-processing

In the preprocessing phase, the dataset was initially loaded from a CSV file, and its dimensions were examined to confirm the number of rows and columns. The filtered data has 32,022 rows and 16 columns. Additionally, the data types of each column were reviewed, and it was verified that there were no missing values and duplicate rows. Histograms were created for numerical features to understand their distributions.

Subsequently, several preprocessing steps were applied to prepare the data for analysis. The 'DATE_TIME' column was converted to datetime object to facilitate time-based analysis and set as the index of the DataFrame. To capture the time-based patterns and seasonality in the data, sine and cosine transformations were applied to the 'HOURS' and 'MINUTES' columns. These transformations help in preserving the cyclical characteristics of time, which is important for models to recognize daily and monthly patterns. After creating the cyclical features, redundant columns such as 'DATE', 'TIME', 'HOURS', and 'MINUTES' were dropped. The data was then checked for duplicate timestamps to ensure that each time point was unique and consistent. The duplicates in the dataset were likely due to simultaneous recordings from multiple inverters. The data was aggregated by taking the mean of selected columns for each timestamp, helping to reduce noise and summarize the data at each time point. Following aggregation, the data was resampled to regular 15-minute intervals to ensure that the time series data was uniformly spaced. Any missing values were interpolated using a linear interpolation method to ensure the dataset was complete and ready for modeling without any gaps.

The time series data was decomposed into its trend, seasonal, and residual components to understand the underlying patterns and structures (Figure 10). There is a noticeable decreasing trend in AC power output from May 15 to June 17, 2020, which may be attributed to seasonal changes or other external factors that impact efficiency. The seasonal plot displays a consistent pattern that matches the daily solar cycle, with peaks in midday solar generation and troughs at night when there is no solar activity. The residuals plot shows random fluctuations, suggesting that there are additional factors influencing power generation that are not captured by seasonal or trend components. These fluctuations could be due to weather conditions or operational issues.

Figure 10. Decomposition of time series data.

Statistical testing for stationarity was conducted using the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. The ADF test indicated that the series was stationary with a low p-value, suggesting that the null hypothesis of a unit root could be rejected. The KPSS test confirmed this with a high p-value, indicating that the null hypothesis of stationarity could not be rejected. Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots were generated to help identify the appropriate parameters for modeling (Figure 11). The ACF plot showed a wave-like pattern that gradually diminished, suggesting seasonality in the data. The PACF plot had a significant spike at lag 1 and then quickly dropped off, indicating a potential AR(1) component.

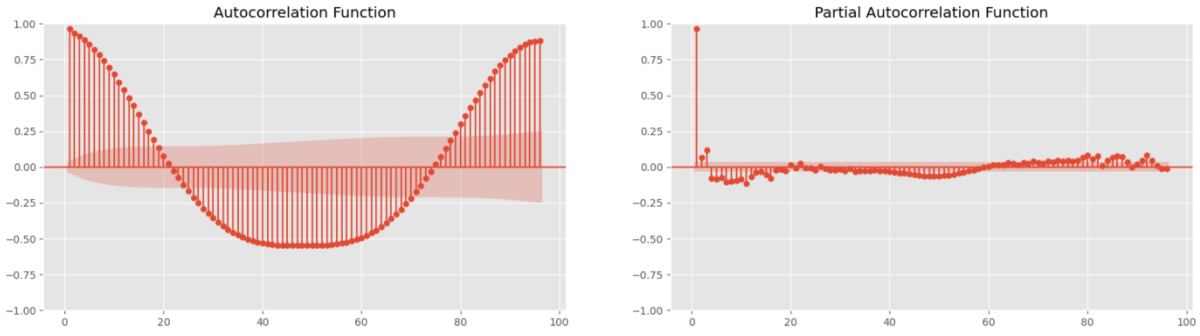

Figure 11. ACF and PACF plot.

5 Modeling

In the modeling process, three different models— The Seasonal AutoRegressive Integrated Moving Average (SARIMA), Random Forest, and XGBoost—were evaluated to predict solar energy generation. Each model was trained using the training data and evaluated using the testing data to assess its performance. Evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) were used to quantify the performance of each model.

The modeling process began with data splitting into training and testing sets with an 80-20 ratio, defining the features ('X') and target variable ('y'). Subsequent steps focused on feature scaling and transformation to ensure consistency and reduce skewness or outliers in the numerical features. Log Transformation was applied to address skewness and outliers, while StandardScaler was used to scale all numerical features to have a mean of 0 and a standard deviation of 1.

To optimize the performance of SARIMA, Random Forest and XGBoost, hyperparameter tuning was performed using grid search. This involved systematically searching through a range of hyperparameters to find the combination that yielded the best performance for each model. For SARIMA, parameters p, d, and q were tuned within the range of 0 to 2. For Random Forest, the hyperparameters tuned included 'n_estimators' (100, 200, 300), 'max_depth' (None, 10, 20), 'min_samples_split' (2, 5, 10), 'min_samples_leaf' (1, 2, 4), and 'max_features' ('sqrt', 'log2'). For XGBoost, the hyperparameters tuned included 'n_estimators' (100, 200, 300), 'max_depth' (3, 5, 7), and 'learning_rate' (0.01, 0.05, 0.1).

Cross-validation was employed to assess the models' performance of Random Forest and XGBoost across different splits of the data. This technique helps estimate how the models will perform on unseen data and provides insights into their stability and generalization ability. Various folds (e.g., 3-fold, 5-fold, 7-fold, 10-fold) were used to evaluate the models' performance under different scenarios.

Diagnostic plots were generated to evaluate the model’s performance, and line plots were used to compare the predicted vs. actual values, aiding in understanding the model's performance.

5.1 SARIMA

Based on the ACF and PACF plots, the **initial SARIMA model parameters** were defined as **order=(1, 0, 0)** and **seasonal_order=(1, 1, 0, 96)**. The diagnostic plots with these parameters indicated that the residual diagnostics reveal some noticeable spikes and patterns, suggesting the model might not have captured all the data patterns. The histogram shows that residuals peak around zero but are slightly skewed and do not perfectly match the normal distribution. The Q-Q plot demonstrates deviations from the reference line, especially in the tails, indicating that the residuals are not perfectly normal. Additionally, the correlogram indicates that most autocorrelation values for non-zero lags fall within the confidence intervals, suggesting no significant autocorrelation at these lags (Figure 12).

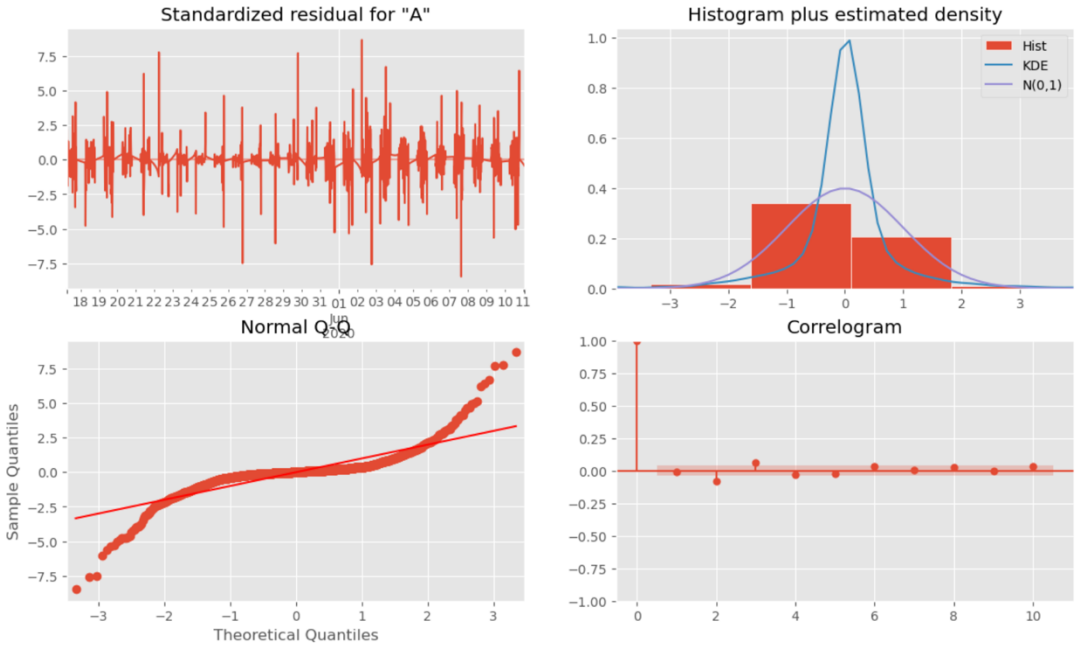

Figure 12. The diagnostic plots for the initial SARIMA model.

The **RMSE** of the predictions was **0.70.** This means that the model's predictions deviate from the actual values by approximately 0.70 units of AC_POWER on average. The model's predictions had an **MAPE** of about **23.38%.** This indicates that the model's predictions were, on average, 23.38% away from the actual values, which is relatively high. This suggests that the basic SARIMA model may not be capturing the underlying patterns in the data as effectively as desired.

The plot shows comparison of predicted and actual values for SARIMA model (Figure 13). The predicted values generally followed the trend of the actual values, indicating that the model is captured the overall pattern and seasonal components. However, there were noticeable

discrepancies between the actual and predicted values, especially during periods of rapid change or at the peaks and troughs. These discrepancies contribute to the relatively high RMSE and MAPE.

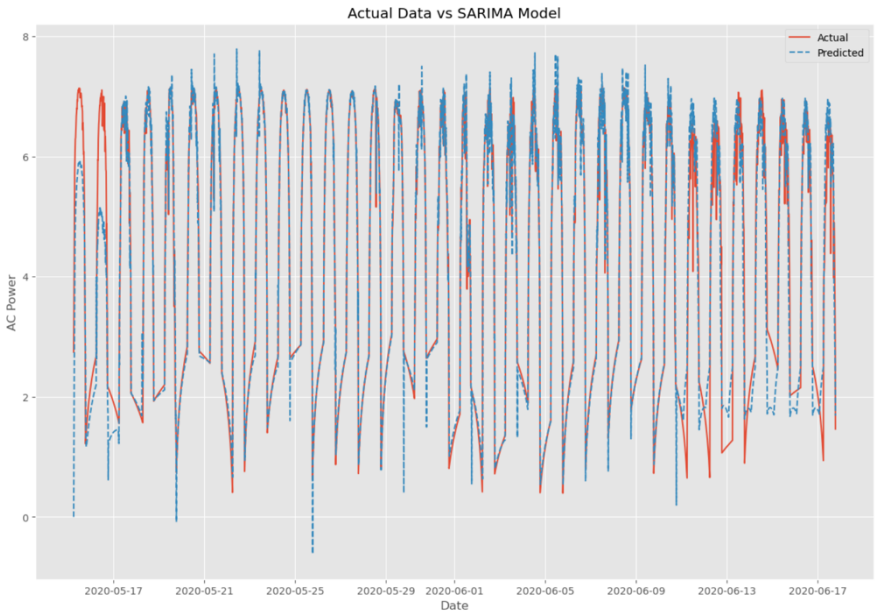

Figure 13. Actual vs. predicted values for the SARIMA model.

After performing a systematic grid search to find the optimal parameters, the **best SARIMA model parameters** were identified as **order=(1, 1, 1)** and **seasonal_order=(1, 1, 0, 96)**. The tuned SARIMA model was then fitted to the training data and used to forecast the testing data. The evaluation metrics for the tuned SARIMA model improved as follows: **RMSE 0.61** and **MAPE: 20.57.** These metrics indicate that the optimized model has better accuracy and lower error compared to the initial model.

The diagnostic plots for the tuned SARIMA model indicated that the residuals were more randomly distributed with no significant autocorrelation, suggesting that the model adequately captured the underlying patterns in the data. However, the line plots comparing the predicted values to the actual values still showed that the tuned SARIMA model struggled to capture some of the rapid fluctuations in solar power output. This could be due to the inherent limitations of the SARIMA model in handling sudden changes or outliers in the data (Figure 14). The tuned model follows the actual data more closely compared to initial one, capturing the overall trend and seasonal patterns with greater accuracy as shown in the Figure 15.

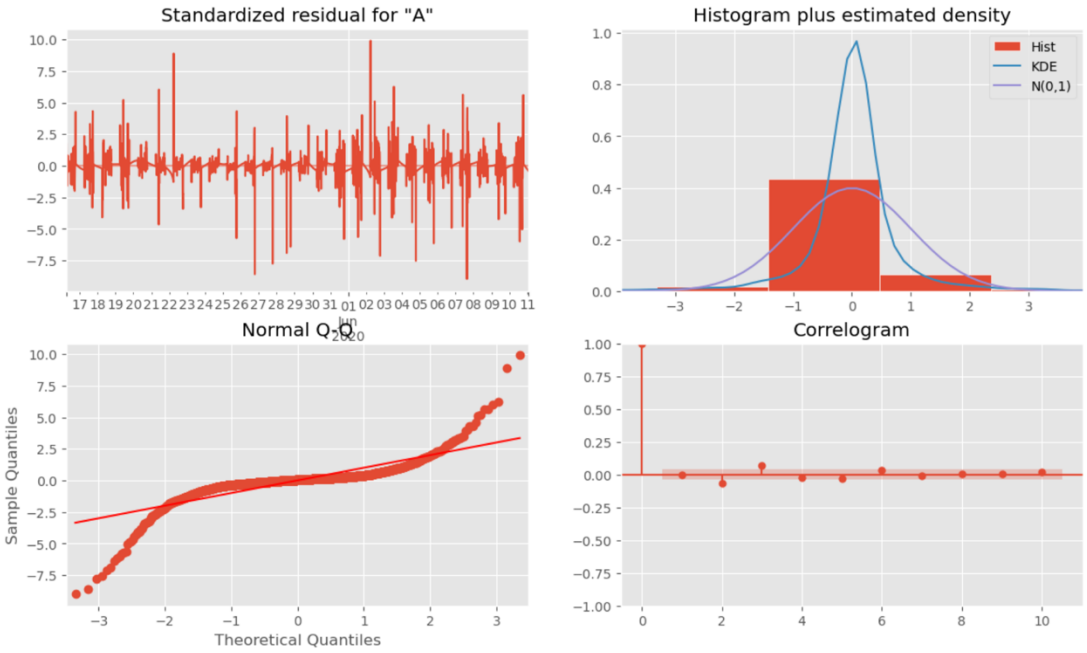

Figure 14. The diagnostic plots for the tuned SARIMA model.

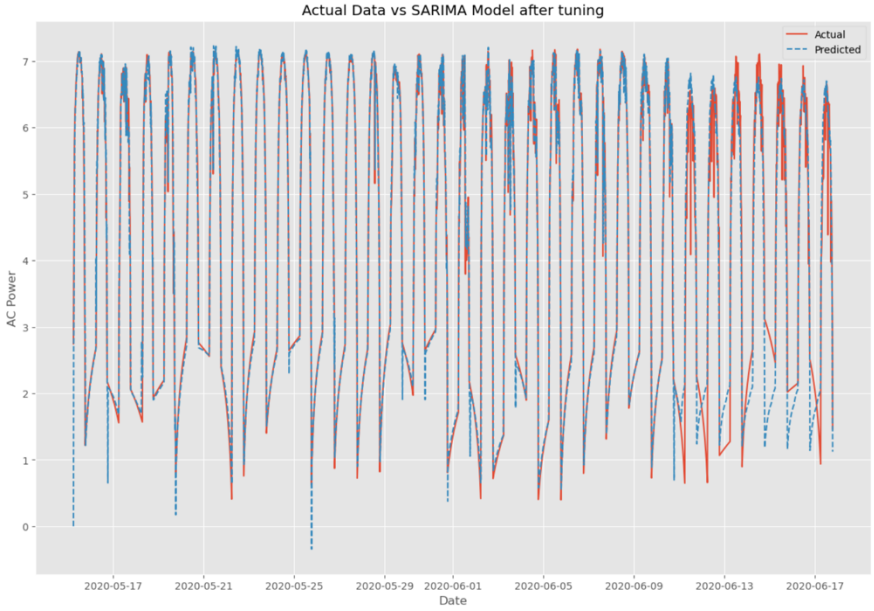

Figure 15. Actual vs. predicted values for the SARIMA model after tuning.

5.2 Random Forest

The **Random Forest** model achieved an impressive **RMSE** value of approximately **0.143** on the testing set. This indicates that, on average, the model's predictions are off by approximately 0.143 units in the log-transformed scale. This is a very low error, suggesting high accuracy, which is significantly higher than the RMSE values obtained from SARIMA

model. Additionally, the Random Forest model yielded a **MAPE** of approximately **4.387%** on the testing set. This indicates that the model's predictions are, on average, about 4.387% off from the actual values. This is an excellent result, as a MAPE below 10% is generally considered very good.

The plot comparing predicted and actual values for the Random Forest model (Figure 16) shows that the model was able to follow the general trend of the actual AC power values. However, there were some discrepancies, especially during periods of rapid change. This was reflected in the RMSE and MAPE values, indicating that while the model captured the overall trend, it struggled with finer details.

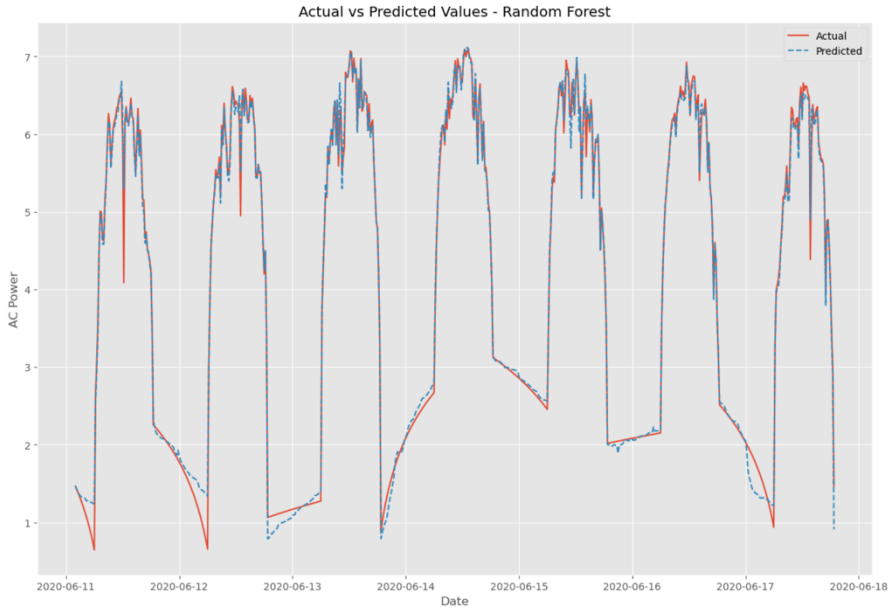

Figure 16. Actual vs predicted values for the Random Forest model.

Table 1 provides an overview of the Random Forest cross-validation results, including the maximum depth, maximum features, number of estimators, the best RMSE score achieved, cross-validation scores on training and testing sets, and the standard deviation in cross-validation scores on the testing set.

Table 1. Random Forest Model Cross-Validation Results.

	Cross-Validation	Max Depth	Max Features	N_Estimators	Best RMSE Score	Mean CV on Training Set	Mean CV on Testing Set	Standard Deviation on Testing Set
0	3-fold	20.0	sqrt	200	0.1114	0.1131	0.1481	0.0308
1	5-fold	NaN	sqrt	300	0.1078	0.1088	0.1453	0.0342
2	7-fold	20.0	log2	300	0.1048	0.1055	0.1384	0.0433
3	10-fold	NaN	sqrt	300	0.1035	0.1034	0.1326	0.0405

The Random Forest model consistently perform well across different cross-validation splits. The best RMSE score improves as the number of splits increases, indicating that using more folds provides a better estimate of the model's performance. The **10-fold** cross-validation achieves the best RMSE score with **max_features=sqrt,** and **n_estimators=300**, suggesting it provides the most reliable estimate among the tested configurations. The RMSE on the training set decreases slightly with increasing number of splits, indicating the model is learning effectively. The RMSE on the testing set also decreases with increasing number of splits, indicating improved model performance and generalization. The standard deviation of RMSE on the testing set indicates variability in model performance across different splits. Lower standard deviation (0.0308) suggests more consistent performance.

After using best parameters, the **RMSE** increased from 0.143 to **0.165**, suggesting that the tuning process might have introduced more complexity. This indicates that the model's predictions are slightly less accurate after tuning. The **MAPE** increased from 4.387% to **4.653%.** The percentage error in predictions has also increased slightly. This further suggests that the model's accuracy has decreased marginally after tuning. This could be attributed to the model overfitting during the tuning process, despite the cross-validation efforts. Actual vs predicted values for the Random Forest model after tuning. Slight variations can be seen in Figure 17 when compared to the initial plot.

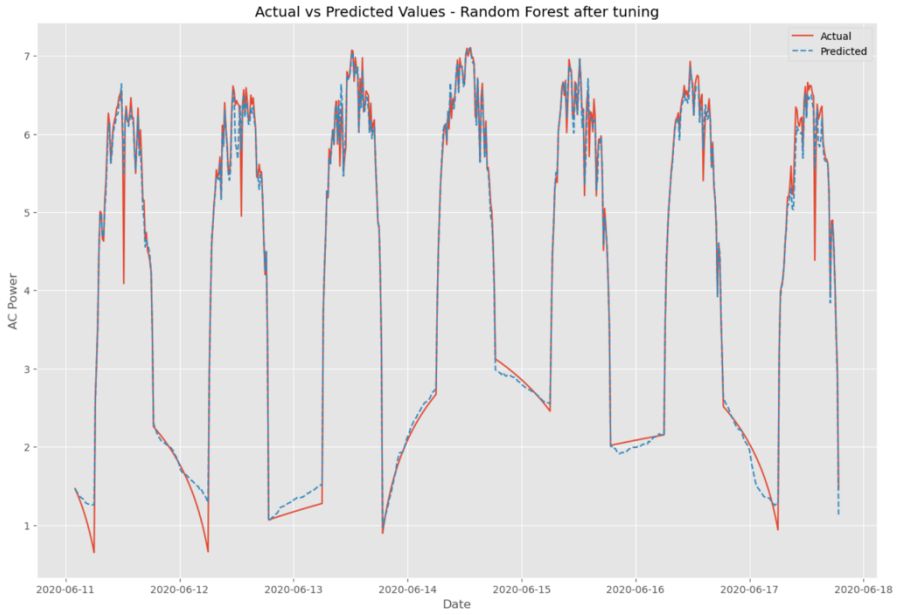

Figure 17. Actual vs predicted values for the Random Forest model after tuning.

The feature importance plot for the Random Forest shown in the Figure 18. Model indicates that IRRADIATION is the most critical feature, accounting for approximately 60% of the importance. This suggests that the amount of irradiation is the primary driver of the model's predictions. MODULE_TEMPERATURE is the second most important feature with around 35%, indicating that the efficiency of solar panels is strongly affected by their temperature. AMBIENT_TEMPERATURE also plays a role, contributing about 7.5%, as it can impact the overall environment and performance of the solar panels. DAILY_YIELD (~ 2.5%) has a lower importance but still contributes to the model. TOTAL_YIELD (~1%) has the least importance among the features.

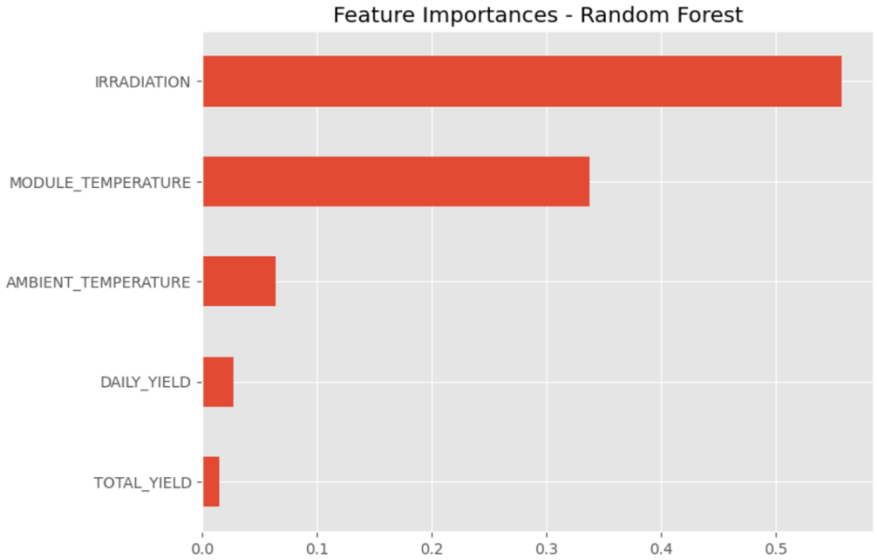

Figure 18. Feature importance of the Random Forest model.

5.3 XGBoost Regression

The XGBoost model achieved an **RMSE** value of approximately **0.135** on the testing set. This indicates that, on average, the model's predictions are off by approximately 0.135 units in the log-transformed scale. This is a very low error, suggesting high accuracy and slightly better performance than the Random Forest model. Additionally, the XGBoost model yielded a **MAPE** of approximately **4.455%** on the testing set. This indicates that the model's predictions are, on average, about 4.455% off from the actual values. This MAPE is similar to the Random Forest model's performance and is generally considered very good, indicating that the model is performing well.

Comparison of predicted and actual values for the XGBoost model is shown in figure 19. The model's predictions align very well with the actual values, particularly during peak and trough periods. This demonstrates the model's ability to handle the daily cyclical patterns effectively.

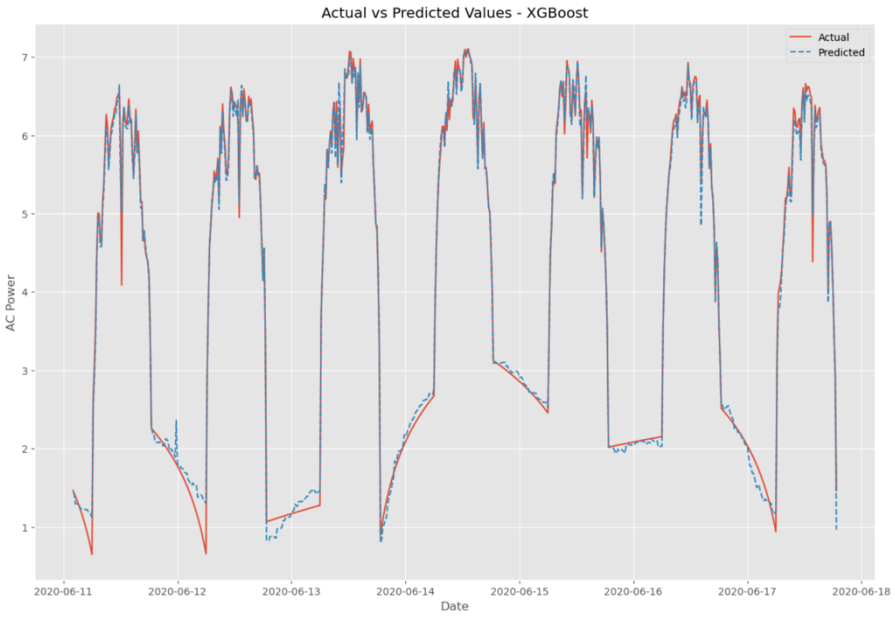

Figure 19. Actual vs. Predicted values for XGBoost model.

Table 2 provides an overview of the XGBoost cross-validation results, including the maximum depth, number of estimator, learning rate, the best score achieved, mean cross-validation scores on the training and testing sets, and the standard deviation in cross-validation scores on the testing set.

Table 2. XGBoost Model Cross-Validation Results.

	Cross-Validation	Max Depth	N Estimators	Learning Rate	Best RMSE Score	CV Score on Training Set	CV Score on Testing Set	Standard Deviation
0	3-fold	5	300	0.1	0.0870	0.08690	0.1021	0.0041
1	5-fold	5	300	0.1	0.0803	0.08031	0.0968	0.0227
2	7-fold	5	300	0.1	0.0854	0.08540	0.0887	0.0221
3	10-fold	5	300	0.1	0.0789	0.07890	0.0899	0.0318

Across all cross-validation folds, the best estimator consistently uses the parameters **learning_rate=0.1, max_depth=5**, and **n_estimators=300**. The RMSE scores improve as the number of folds increases, with the **best score** being **0.0789** for the **10-fold** cross-validation. This suggests that using more folds provides a better estimate of the model's performance and might help in achieving lower error rates. The training set RMSE consistently shows lower error rates compared to the testing set, which is expected. The difference between training and testing RMSE indicates the model's generalization performance. The smallest difference

between training and testing RMSE is observed in the 7-fold cross-validation, indicating good generalization. The 3-fold cross-validation has the lowest standard deviation, indicating the most consistent performance across folds, while the 10-fold cross-validation has the highest standard deviation.

After hyperparameter tuning, the model's performance improved, achieving an **RMSE** of **0.125** and a **MAPE** of **4.065%.** These results indicate that the XGBoost model effectively captures the underlying patterns in the data and provides accurate predictions. The cross-validation results further highlight the model's robustness and generalization capabilities as shown in the figure 20.

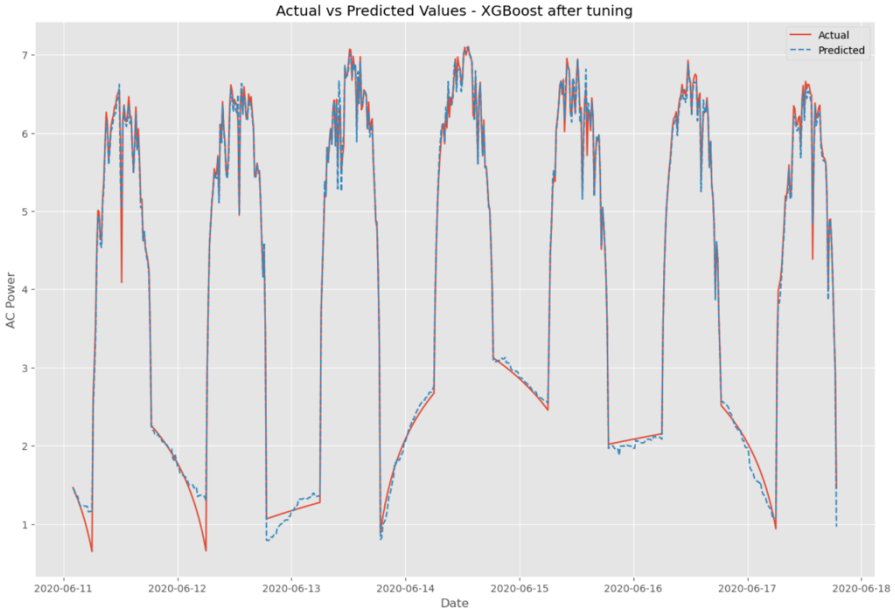

Figure 20. Actual vs. Predicted values for XGBoost model after tuning.

The feature importance plot for the XGBoost model (Figure 21) indicates that 'IRRADIATION' is the most critical feature by far, suggesting it has the highest impact on the model's predictions (~ 100%). This makes sense as solar power generation is directly influenced by the amount of solar irradiation. Other feature like 'TOTAL_YIELD' has minimal impact on the model's predictions. This might be because 'TOTAL_YIELD' is a cumulative measure and does not provide immediate information about the current state of power generation.

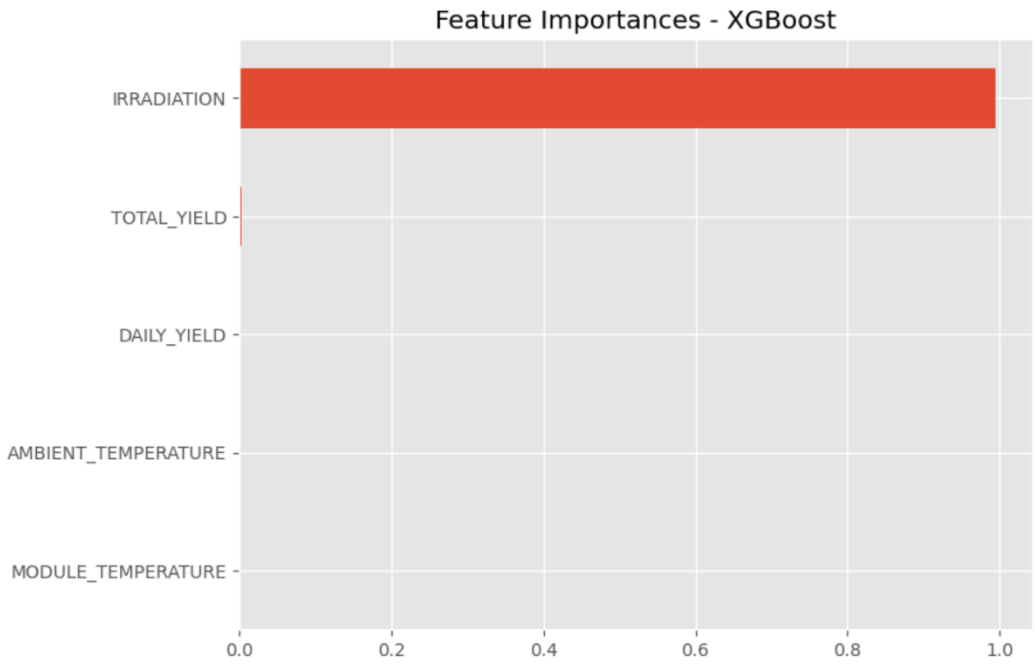

Figure 21. Feature importance of the XGBoost model.

6 Results

The performance of each model was analyzed and compared based on evaluation metrics such as RMSE and MAPE. The strengths and weaknesses of each model was identified and determined which model performed best for predicting the solar power generation. Table 3 provides the comparison of regression models and figure 22 illustrates the comparison of MAPE values for different models. The analysis and comparison of the SARIMA, Random Forest, and XGBoost models for predicting solar energy generation revealed distinct performance differences, as summarized in Table 3 and Figure 20.

Table 3. Comparison of models.

	Model	RMSE	MAPE	CV RMSE Score on Testing Set	Training Time (minutes)	Best Parameters	CV Standard Deviation
0	SARIMA	0.608	20.570	-	143	order=(1,1,1), seasonal_order=(0,1,1,96)	-
1	Random Forest	0.165	4.665	10-fold: 0.1035	6	max_features=sqrt, n_estimators=300	0.0405
2	XGBoost	0.125	4.065	10-fold: 0.0789	1	max_depth=5, learning_rate=0.1, n_estimators=300	0.0318

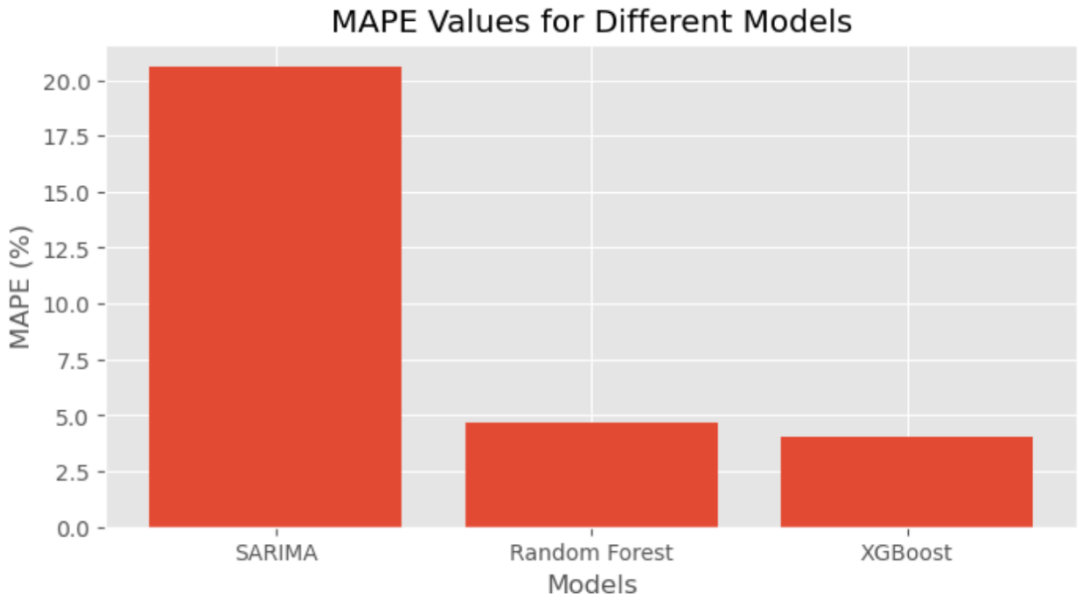

Figure 22. Comparison of MAPE values for different models.

- The XGBoost model outperformed the other models, achieving the lowest RMSE (0.125) and MAPE (4.065%). This indicates that the XGBoost model provides the most accurate predictions with the least error.
- The XGBoost model's cross-validation performance further confirmed its consistency and reliability, with the lowest mean RMSE score of 0.0789 and the smallest standard deviation of 0.0318 across different data splits.
- The XGBoost model's training time was also the shortest at 1 minute, demonstrating its efficiency.
- The Random Forest model, despite also showing good performance, had higher error rates with an RMSE of 0.165 and a MAPE of 4.653% after tuning. Its cross-validation RMSE score was 0.1035 with a standard deviation of 0.0405, indicating slightly less consistent performance compared to XGBoost.
- The training time for Random Forest was 6 minutes, which is longer than XGBoost but still reasonable.
- The SARIMA model showed the least favorable performance with an RMSE of 0.608 and a MAPE of 20.570%, indicating significant deviations between the predicted and actual values.
- The SARIMA model also required the longest training time at 143 minutes, highlighting its inefficiency for this dataset. The high error rates and extended training time make SARIMA less suitable for practical applications in solar energy prediction.

- The comparative analysis underscores the superiority of the XGBoost model in terms of accuracy, efficiency, and stability. Therefore, XGBoost is recommended as the optimal model for predicting solar energy generation in this dataset.

7 Conclusion

In this project, three models—SARIMA, Random Forest, and XGBoost—were evaluated to predict solar energy generation. Each model underwent a rigorous training and evaluation process, including hyperparameter tuning and cross-validation. The XGBoost model demonstrated the best performance with the lowest RMSE and MAPE, indicating its high accuracy and robustness in capturing the patterns in the data. The Random Forest model also performed well but showed slightly higher error rates after tuning. The SARIMA model, while useful in capturing seasonal trends, exhibited the highest error rates and longer training times, making it less favorable for this application.

8 Recommendations

Based on our findings, there are several promising directions for future research. Firstly, incorporating additional environmental factors such as humidity, wind speed, and cloud cover could provide more comprehensive inputs for the model, potentially improving its predictive accuracy. These additional features could help the model better understand the conditions affecting solar power generation. Secondly, incorporating cyclical features, such as sine and cosine transformations of time variables, can help capture daily and seasonal patterns more effectively. Thirdly, advanced feature engineering techniques should be implemented. Creating lag features, rolling statistics, and interaction terms can help capture complex patterns and dependencies in the data, enhancing the model's ability to make accurate predictions. Additionally, applying alternative time series models instead of SARIMA, such as Prophet, could capture different aspects of the data and potentially offer better performance for certain types of time series data. Lastly, exploring deep learning methods, such as neural networks, could yield significant improvements in predictive accuracy. Techniques like Long Short-Term Memory (LSTM) networks are particularly suited for time series forecasting and could be investigated for their potential benefits. By implementing these strategies, the predictive accuracy and robustness of the models can be further improved, leading to more reliable solar energy generation forecasts.

9 References

1. Rockström, J., et al., *Water resilience for human prosperity*. 2014: Cambridge University Press.
2. UNESCO, *The United Nations World Water Development Report 2021: Valuing Water*. 2021: United Nations.
3. Bui, D.T., et al., *Improving prediction of water quality indices using novel hybrid machine-learning algorithms.* Science of the Total Environment, 2020. **721**: p. 137612.
4. Ahmed, M., R. Mumtaz, and S.M. Hassan Zaidi, *Analysis of water quality indices and machine learning techniques for rating water pollution: A case study of Rawal Dam, Pakistan.* Water Supply, 2021. **21**(6): p. 3225-3250.
5. Lap, B.Q., et al., *Predicting water quality index (WQI) by feature selection and machine learning: a case study of An Kim Hai irrigation system.* Ecological Informatics, 2023. **74**: p. 101991.
6. Ahmed, M., R. Mumtaz, and Z. Anwar, *An Enhanced Water Quality Index for Water Quality Monitoring Using Remote Sensing and Machine Learning.* Applied Sciences, 2022. **12**(24): p. 12787.
7. Uddin, M.G., S. Nash, and A.I. Olbert, *A review of water quality index models and their use for assessing surface water quality.* Ecological Indicators, 2021. **122**: p. 107218.
8. Uddin, M.G., et al., *A comprehensive method for improvement of water quality index (WQI) models for coastal water quality assessment.* Water Research, 2022. **219**: p. 118532.
9. Uddin, M.G., et al., *A novel approach for estimating and predicting uncertainty in water quality index model using machine learning approaches.* Water Research, 2023. **229**: p. 119422.
10. Khoi, D.N., et al., *Using machine learning models for predicting the water quality index in the La Buong River, Vietnam.* Water, 2022. **14**(10): p. 1552.
11. Hassan, M.M., et al., *Efficient prediction of water quality index (WQI) using machine learning algorithms.* Human-Centric Intelligent Systems, 2021. **1**(3-4): p. 86-97.