A Report
On
STATISTICAL ANALYSIS AND FORECASTING OF SOLAR ENERGY

BY

Friedman Group

Neerav Krishna	2023A4PS0416P
Muralidhara Samarth	2023A4PS0418P
Yashwanth Varma Dandu	2023A4PS0458P
Amogh Aryan	2022B4A10735P
Abraham George	2023A1PS0222P
Pratham Galbale	2023A1PS0198P
Kritarth Gusain	2023AAPS0764P

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

Table of Contents

Introduction

India has really stepped up its game with renewable energy in the last 10 years. Energy groups say we've got over 170 GW of renewable power now, putting us up there with the best. But, we still use a lot of fossil fuels for electricity. So, we need to find better ways to get renewables onto the grid. Since we're using more and more energy, renewables need to be reliable if we want energy that lasts and keeps us secure.

When it comes to renewables, solar power is one of the best bets for India. We've got tons of sunlight, it's not too expensive, and it's good for the environment. However, weather like clouds, humidity, how hot it is, and the local climate can mess with solar's performance. One has to understand how these things change to guess how much energy obtained from the sun.

In this study, we look at hourly sunlight data from 2000 to 2014. We got the data from two solar plants in Rajasthan. Even though the plants are in the same state, they have different weather because of their geographical locations. This makes the data perfect for checking how sunlight changes over time and how reliable solar energy is in one area.

The aim of this study is to examine the sunlight data using time series analysis. This means checking how the data is spread out, spotting trends, and making models to predict the future.

The key parameters used in our study include:

- **Direct Normal Irradiance(DNI)**: Amount of solar radiation received per unit area by a surface that is always held perpendicular (or normal) to the rays that come in a straight line from the direction of the sun at its current position in the sky.

- **Diffuse Horizontal Irradiance(DHI)**: Solar radiation that does not arrive on a direct path from the sun, but has been scattered by clouds and particles in the atmosphere and comes equally from all directions.

- **The solar zenith angle(Z):** Angle between the sun's rays and the vertical.

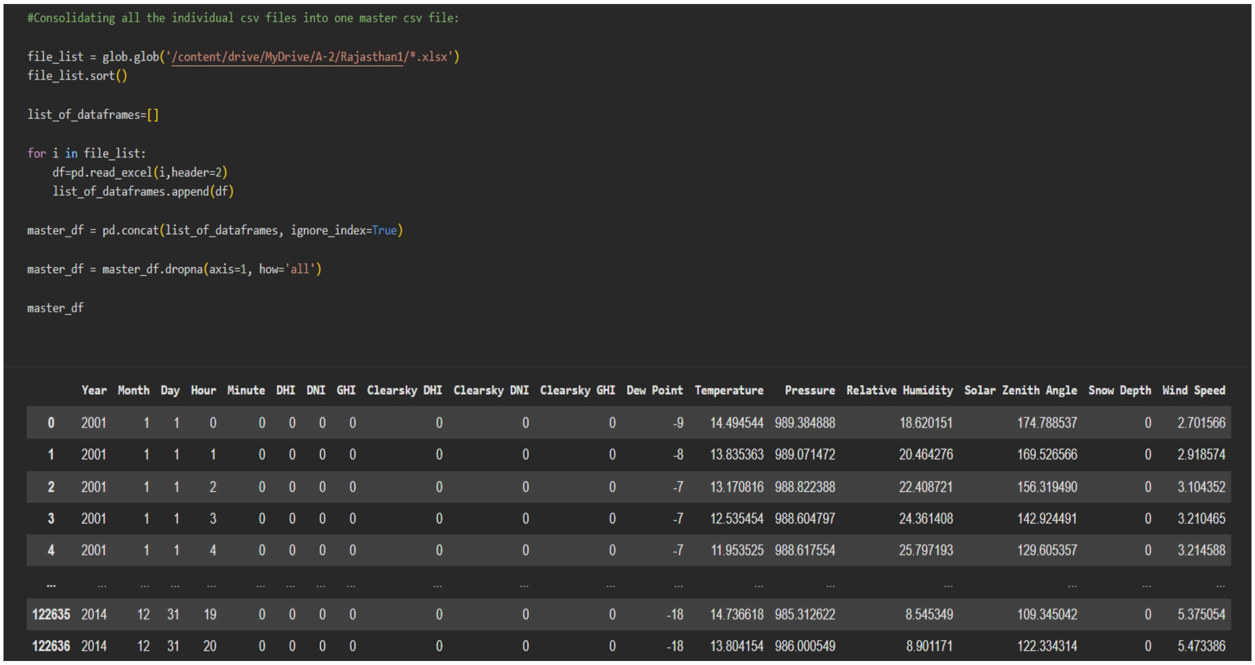
- **Global Horizontal Irradiance (GHI)**: Total amount of shortwave radiation received from above by a surface parallel to the ground. The following relation holds between GHI, DNI and DHI: $\mathbf{GHI = DNI * \cos Z + DHI}$

Descriptive Statistics

The dataset contains hourly data of two solar parks (Rajasthan 1 and Rajasthan 2)for the state of Rajasthan from 2000 to 2014.The following attributes were available in the dataset

- Date and Time of measurement.
- DHI and Clearsky DHI
- DNI and Clearsky DNI
- GHI and Clearsky GHI
- Dew Point
- Temperature
- Pressure
- Relative Humidity
- Solar Zenith Angle
- Wind Speed

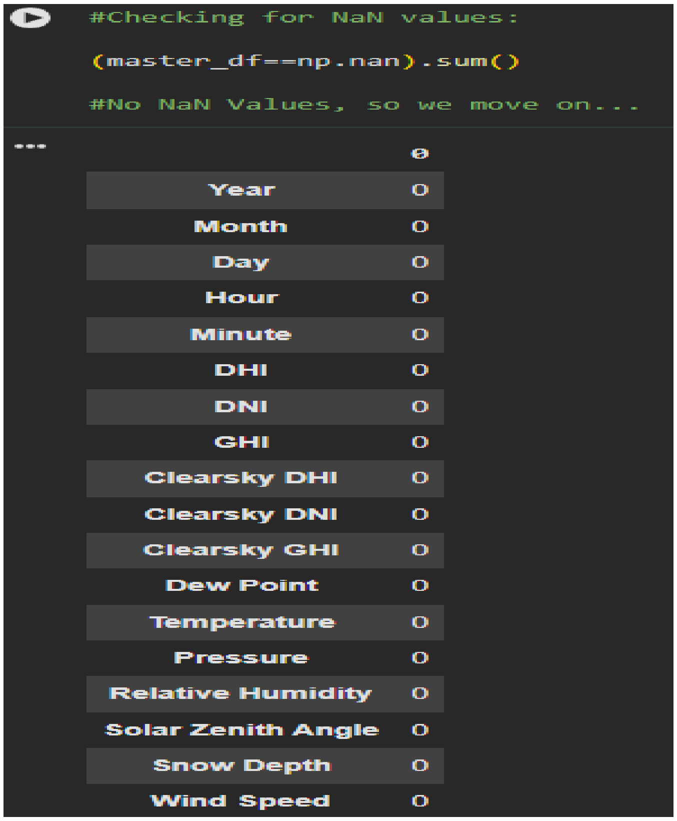
First we load the data in using python’s glob library:



```
#Consolidating all the individual csv files into one master csv file:

file_list = glob.glob('/content/drive/MyDrive/A-2/Rajasthan1/*.xlsx')
file_list.sort()

list_of_dataframes=[]

for i in file_list:
    df=pd.read_excel(i,header=2)
    list_of_dataframes.append(df)

master_df = pd.concat(list_of_dataframes, ignore_index=True)

master_df = master_df.dropna(axis=1, how='all')

master_df
```

	Year	Month	Day	Hour	Minute	DHI	DNI	GHI	Clearsky DHI	Clearsky DNI	Clearsky GHI	Dew Point	Temperature	Pressure	Relative Humidity	Solar Zenith Angle	Snow Depth	Wind Speed
0	2001	1	1	0	0	0	0	0	0	0	0	-9	14.494544	989.384888	18.620151	174.788537	0	2.701566
1	2001	1	1	1	0	0	0	0	0	0	0	-8	13.835363	989.071472	20.464276	169.526566	0	2.918574
2	2001	1	1	2	0	0	0	0	0	0	0	-7	13.170816	988.822388	22.408721	156.319490	0	3.104352
3	2001	1	1	3	0	0	0	0	0	0	0	-7	12.535454	988.604797	24.361408	142.924491	0	3.210465
4	2001	1	1	4	0	0	0	0	0	0	0	-7	11.953525	988.617554	25.797193	129.605357	0	3.214588
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
122635	2014	12	31	19	0	0	0	0	0	0	0	-18	14.736618	985.312622	8.545349	109.345042	0	5.375054
122636	2014	12	31	20	0	0	0	0	0	0	0	-18	13.804154	986.000549	8.901171	122.334314	0	5.473386

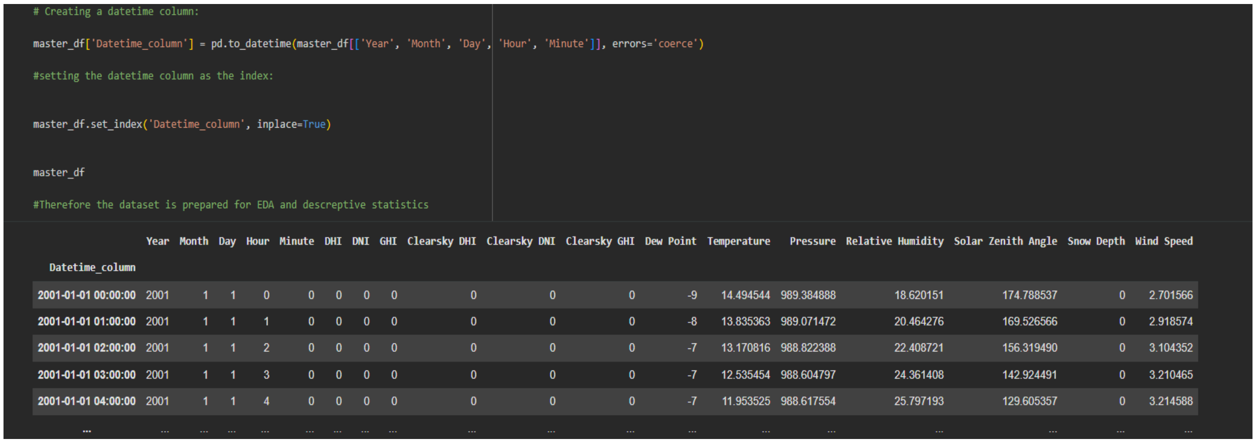
Then we check if the dataset has any null values in the form of NaN:



```
#Checking for NaN values:
(master_df==np.nan).sum()
#No NaN Values, so we move on...
```

```
                          0
               Year       0
              Month       0
                Day       0
               Hour       0
             Minute       0
                DHI       0
                DNI       0
                GHI       0
        Clearsky DHI      0
        Clearsky DNI      0
        Clearsky GHI      0
          Dew Point       0
        Temperature       0
           Pressure       0
   Relative Humidity      0
  Solar Zenith Angle      0
         Snow Depth       0
         Wind Speed       0
```

To prepare the dataset for time series analysis, we also create a column of the datetime datatype which is done by consolidating the data in the Year, Month, Day, Hour, Minute which is in the form of integers.

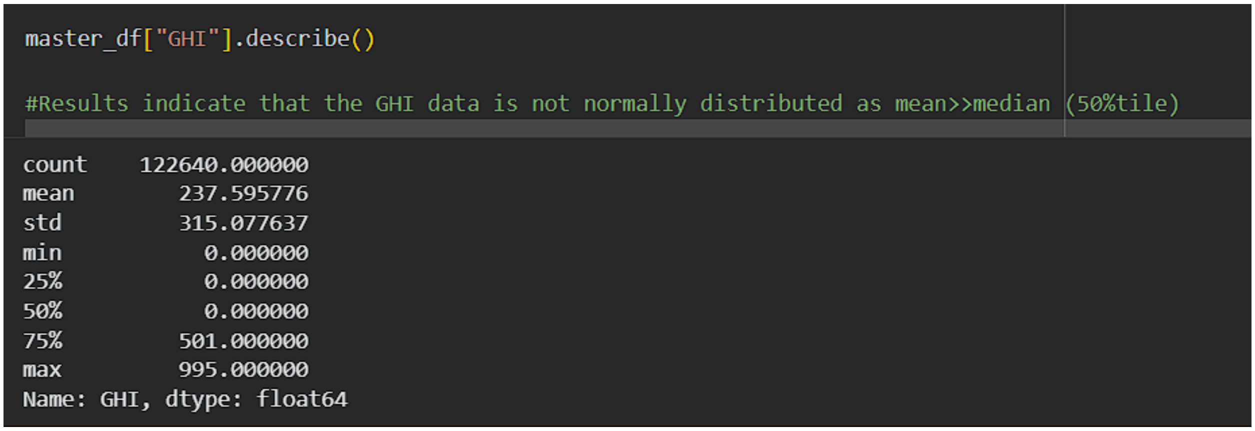


```
# Creating a datetime column:

master_df['Datetime_column'] = pd.to_datetime(master_df[['Year', 'Month', 'Day', 'Hour', 'Minute']], errors='coerce')

#setting the datetime column as the index:


master_df.set_index('Datetime_column', inplace=True)


master_df

#Therefore the dataset is prepared for EDA and descreptive statistics
```

```
                     Year  Month  Day  Hour  Minute  DHI  DNI  GHI  Clearsky DHI  Clearsky DNI  Clearsky GHI  Dew Point  Temperature    Pressure  Relative Humidity  Solar Zenith Angle  Snow Depth  Wind Speed
    Datetime_column
2001-01-01 00:00:00  2001      1    1     0       0    0    0    0             0             0             0         -9    14.494544  989.384888          18.620151          174.788537           0    2.701566
2001-01-01 01:00:00  2001      1    1     1       0    0    0    0             0             0             0         -8    13.835363  989.071472          20.464276          169.526566           0    2.918574
2001-01-01 02:00:00  2001      1    1     2       0    0    0    0             0             0             0         -7    13.170816  988.822388          22.408721          156.319490           0    3.104352
2001-01-01 03:00:00  2001      1    1     3       0    0    0    0             0             0             0         -7    12.535454  988.604797          24.361408          142.924491           0    3.210465
2001-01-01 04:00:00  2001      1    1     4       0    0    0    0             0             0             0         -7    11.953525  988.617554          25.797193          129.605357           0    3.214588
        ...           ...    ...  ...   ...     ...  ...  ...  ...           ...           ...           ...        ...          ...         ...                ...                 ...         ...         ...
```

We choose to analyse the GHI column as it represents the total energy available to a typical solar park, making it the most important metric for forecasting energy generation.

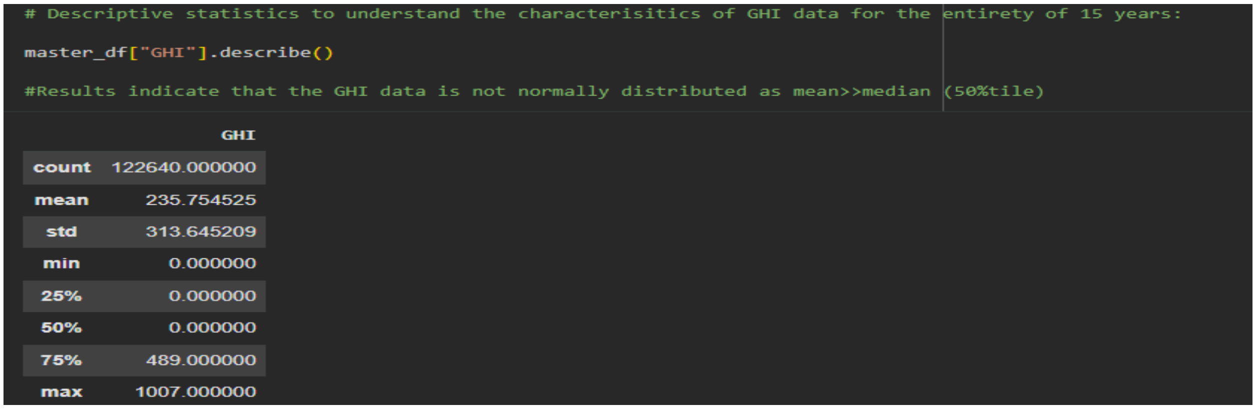
The descriptive statistics to understand the characteristics of GHI data for the entirety of 15 years for Rajasthan 1:



```
master_df["GHI"].describe()

#Results indicate that the GHI data is not normally distributed as mean>>median (50%tile)
```

```
count    122640.000000
mean        237.595776
std         315.077637
min           0.000000
25%           0.000000
50%           0.000000
75%         501.000000
max         995.000000
Name: GHI, dtype: float64
```

And the same for Rajasthan 2:



```
# Descriptive statistics to understand the characterisitics of GHI data for the entirety of 15 years:

master_df["GHI"].describe()

#Results indicate that the GHI data is not normally distributed as mean>>median (50%tile)
```

	GHI
count	122640.000000
mean	235.754525
std	313.645209
min	0.000000
25%	0.000000
50%	0.000000
75%	489.000000
max	1007.000000

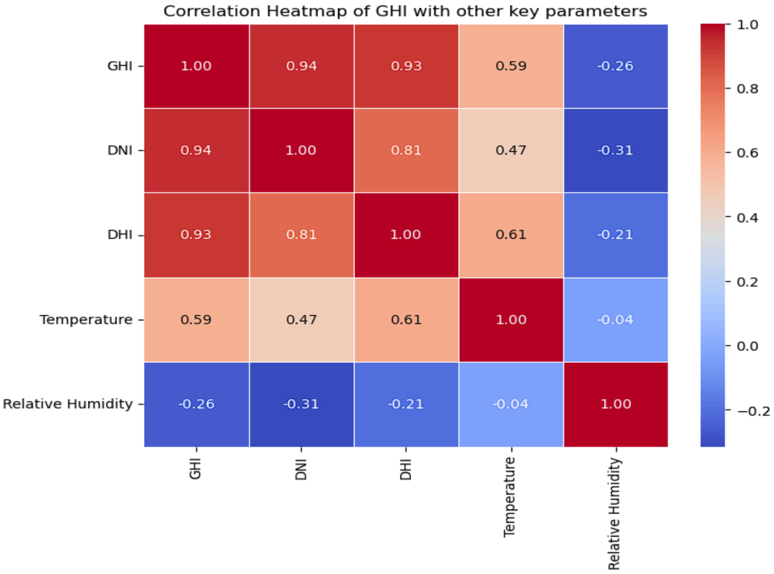
The following inferences can be made for both the locations:

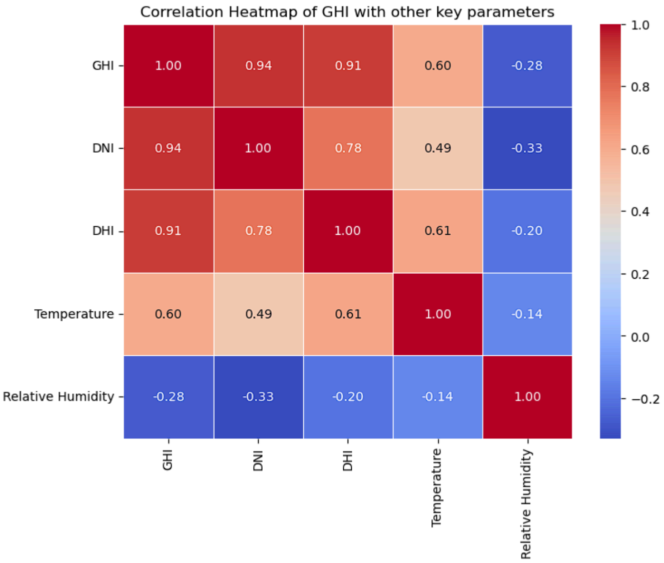
1. We observe that mean >> median (50%tile). This is the first indication that the data may not follow a Normal distribution.
2. The 25%tile = 50%tile indicates that the median of the lower half of the dataset is equal to the median itself which itself is zero. This makes sense because the GHI (measure of irradiance) is zero in the night, which comprises half the 15-year long hourly dataset given. Moreover, the GHI in either half of the night would also be zero, implying that 25%tile = 0.

Exploratory Data Analysis

Analysis of Correlation Matrix:

To examine the correlation of GHI with other parameters like DNI, DHI, temperature and humidity, we understand the significance of the correlation matrix that is visualized using a heatmap.





Rajasthan 1

Rajasthan 2

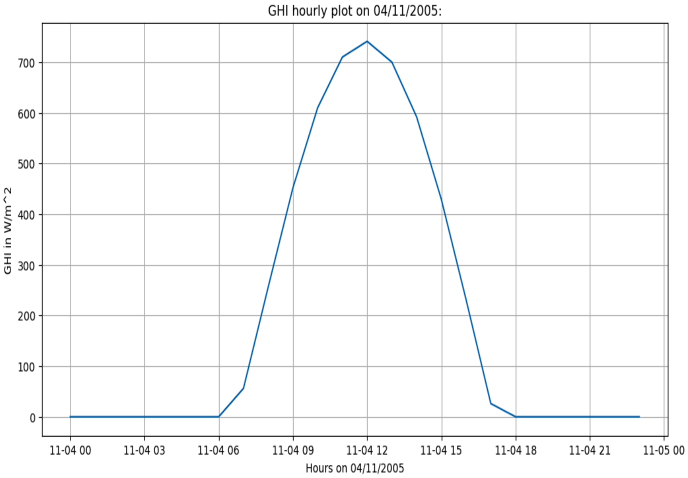
The following inferences can be made from the correlation parameters:

1. GHI is overwhelmingly determined by its two components, DNI and DHI, as evidenced by the correlation coefficients of 0.94 and 0.93. This is expected as GHI depends on DNI and DHI according to the equation: GHI = DNI ∗ cosZ + DHI
2. There is a significant, but less perfect, correlation between GHI and temperature (0.59), highlighting that sunlight is the key factor heating the air.
3. The negative correlation with Relative Humidity (-0.26) confirms that weather conditions with high moisture content (such as a cloudy day) generally lead to reduced solar irradiance (GHI).

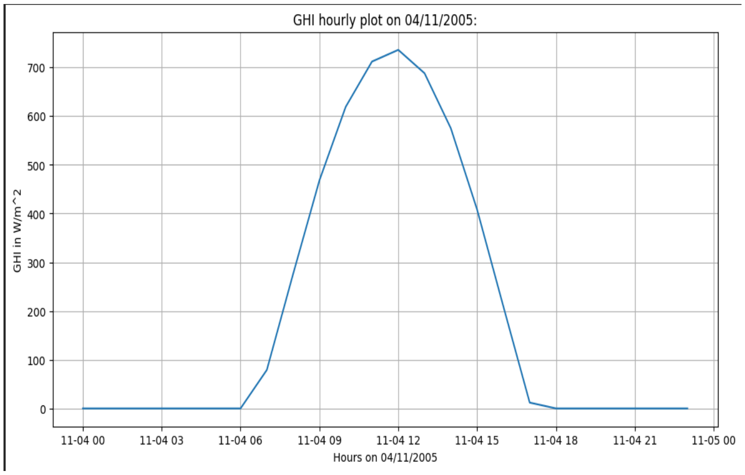
Hourly and daily variation of GHI

In this section, we plot the hourly variations of GHI data in a randomly selected day and daily variations of GHI in a randomly selected week and in an average year (by consolidating all the data available for 15 years).

A. Hourly Variation in on 04/11/2005



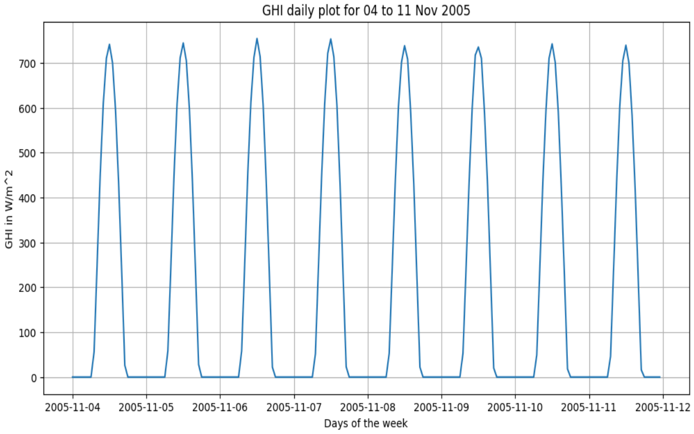
RJ1



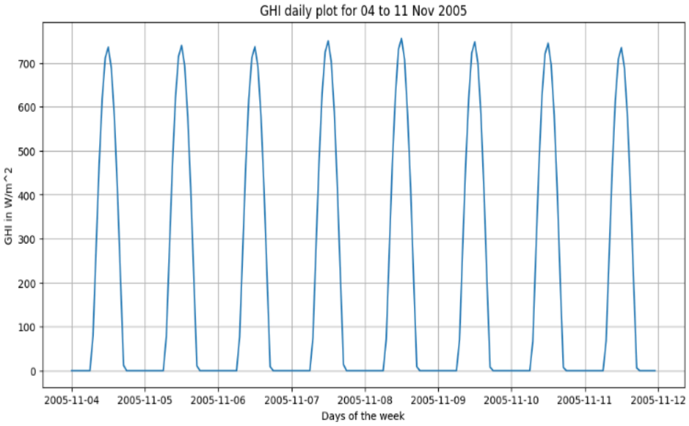
RJ2

Inference: We see that GHI peaks exactly at noon and is the least during the dusk and dawn. This is the expected behaviour of GHI data on an hourly basis.

B. Daily Variation from 04/11/2005-11/11/2005:



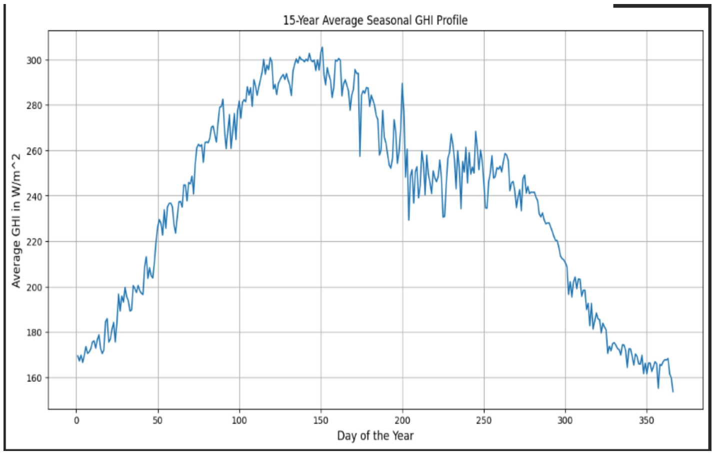
RJ1

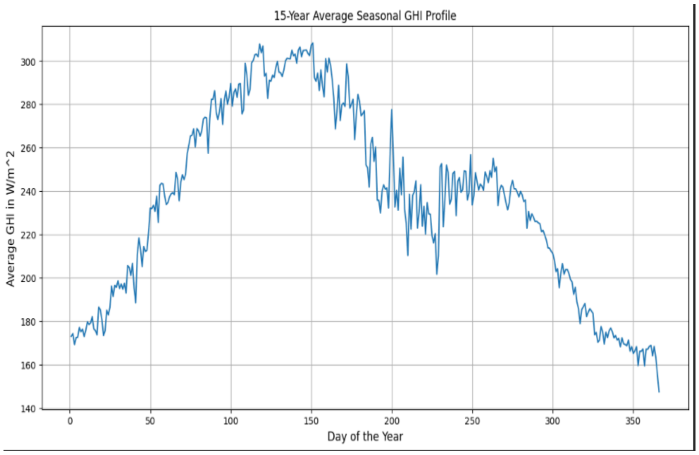


RJ2

Inference:We further observe a seasonal pattern of peaks that seem to occur everyday of the week around 12pm.

C. Daily GHI variation in an average year:





RJ1

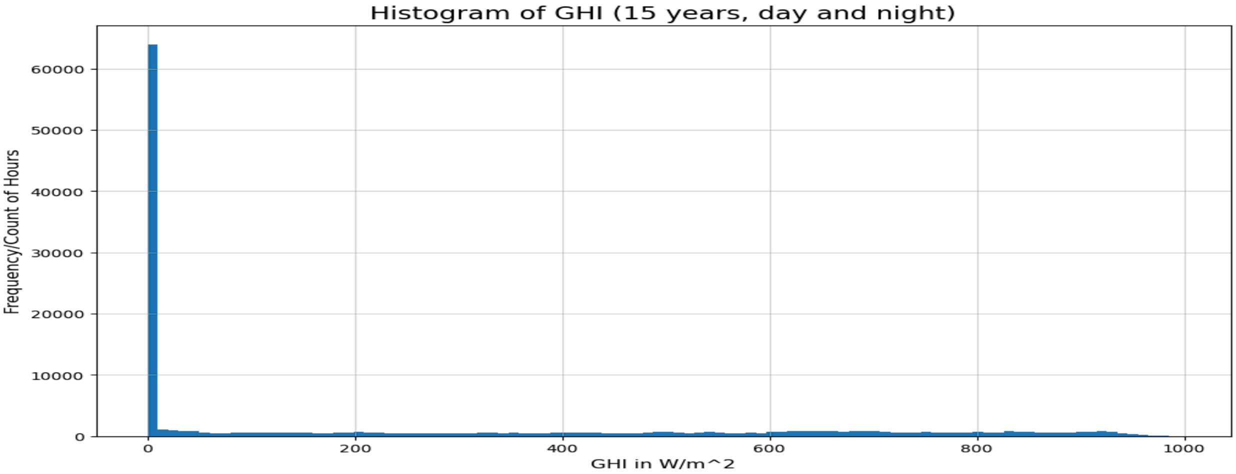
RJ2

Inference: From the 15 year seasonal GHI profile it is evident that GHI peaks during the summer days (April to June) and drastically reduces on the onset of monsoon.

To test for normality of GHI data:

We use visual methods and the Shapiro Wilk test to check if the GHI data follows a normal distribution or not.

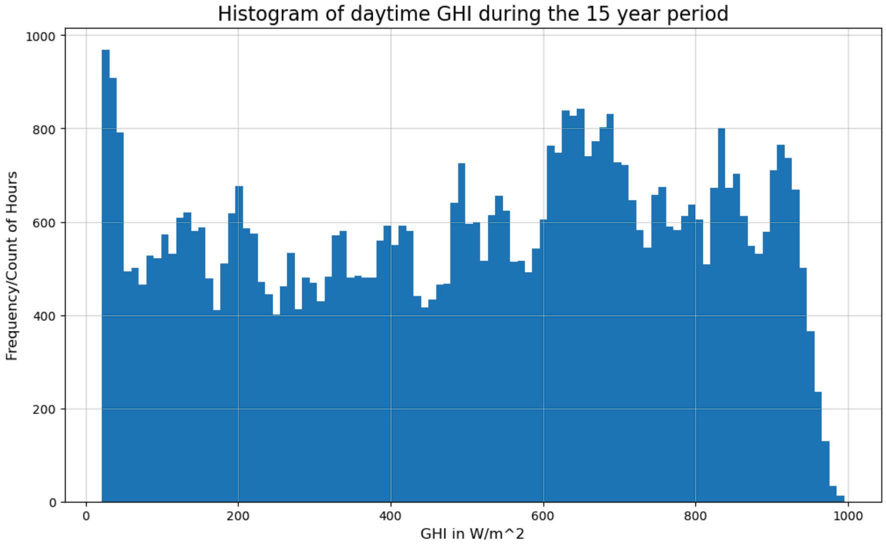
A. VIsual Method using histogram: We first plot the histogram of the entire GHI dataset for Rajasthan1:



The histogram reveals a massive spike at GHI = 0.This zero-value frequency is due to GHI measurements taken during the night time measurements. These datapoints with GHI = 0 must be removed to properly analyze the GHI distribution.

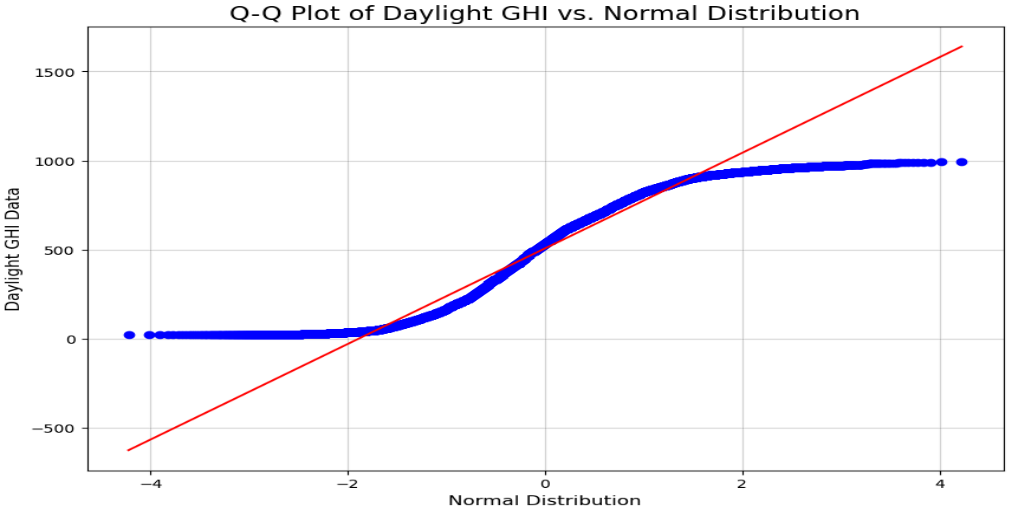
Once nighttime data is excluded, from the below plot it can be inferred that the remaining daylight GHI data exhibits strong positive (right) skewness. This suggests that moderate-to-low GHI values are more common than peak, high-irradiance events. This suggests that the data does not follow a normal distribution.

Plotting histogram for only daytime with a cutoff for GHI>20 (avoid taking in noise and to take into account only the meaningful sunlight in the non transition periods like noon (not dawn and dusk)



A similar histogram is obtained for Rajasthan2, indicating non-normality of the GHI data.

B. Visual Method using Q-Q plot: Since the GHI data for RJ1 does not follow the red line which is the indicator line for a normal distribution, we further conclude that the GHI data does not follow a normal distribution. A similar behaviour is seen for RJ2.



C. Statistical Verification of GHI Distribution Normality: We use the Shapiro Wilk test to formally verify the non-normal distribution observed graphically for the filtered daylight GHI data. This test is a highly effective goodness-of-fit measure for assessing whether a sample comes from a normally distributed population. The hypotheses for the SW test:

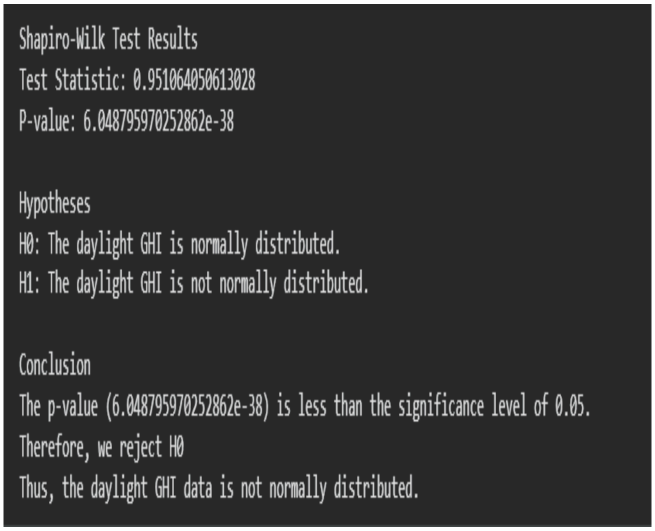
H0: The daylight GHI is normally distributed
H1: The daylight GHI is not normally distributed

The test statistic in SW test:

$$W = \frac{\left(\sum_{t=2}^{n} a_t y_t\right)^2}{\sum_{t=1}^{n} (x_t - \bar{y})^2}$$

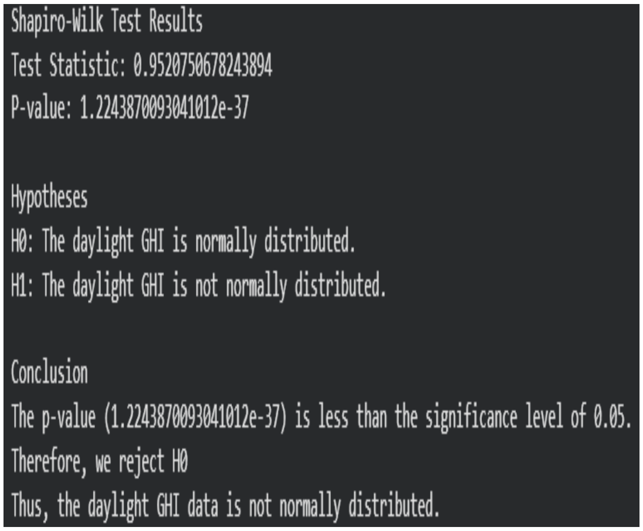
n ... number of observations
y_t... values of the ordered sample
a_t... tabulated coefficients

The results obtained:



```
Shapiro-Wilk Test Results
Test Statistic: 0.951064050613028
P-value: 6.048795970252862e-38


Hypotheses
H0: The daylight GHI is normally distributed.
H1: The daylight GHI is not normally distributed.


Conclusion
The p-value (6.048795970252862e-38) is less than the significance level of 0.05.
Therefore, we reject H0
Thus, the daylight GHI data is not normally distributed.
```

RJ1



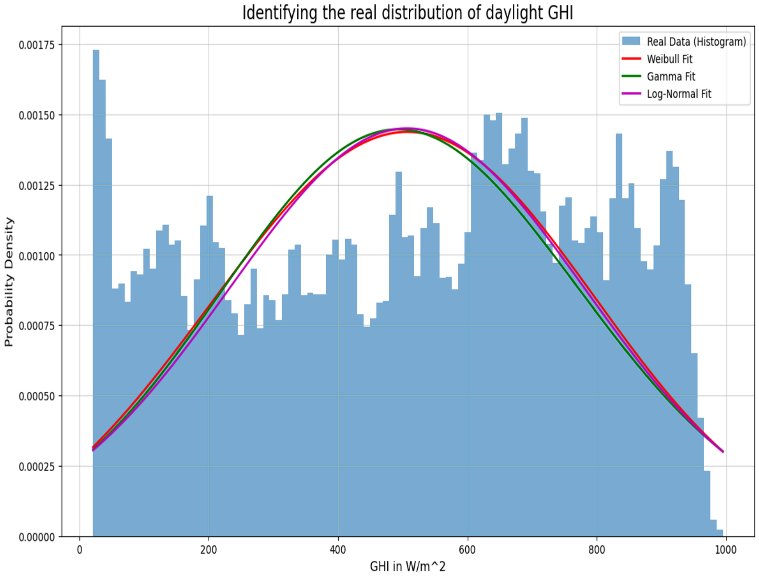
```
Shapiro-Wilk Test Results
Test Statistic: 0.9520750678243894
P-value: 1.2243870093041012e-37


Hypotheses
H0: The daylight GHI is normally distributed.
H1: The daylight GHI is not normally distributed.


Conclusion
The p-value (1.2243870093041012e-37) is less than the significance level of 0.05.
Therefore, we reject H0
Thus, the daylight GHI data is not normally distributed.
```

RJ2

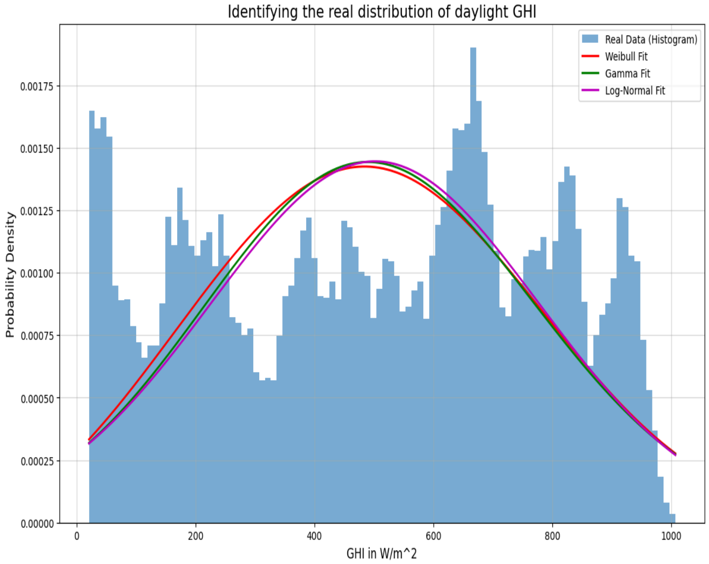
This statistically confirms the initial visual assessment: The daylight GHI data is NOT normally distributed. This conclusion is consistent with initial expectations for irradiance data, which is typically constrained to be positive and often exhibits positive skewness.

To identify the distribution of daylight GHI data

We try to visually see which distribution fits the histogram of daytime GHI. We choose three distributions (Weibull, Gamma and Log-normal) for this visual analysis.



RJ1



RJ2

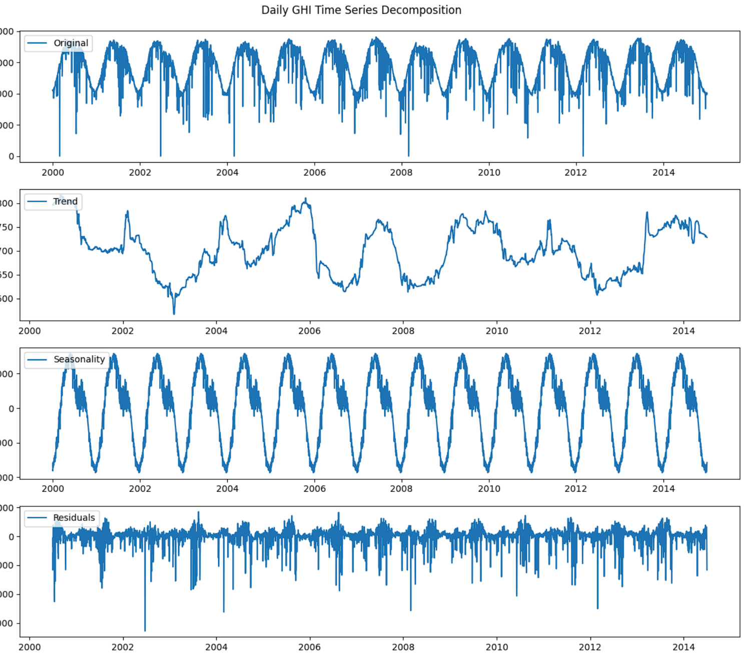
From the above plot, the daylight GHI does not seem to follow any of the standard distributions but rather seems to be a mixture of several different distributions. This further indicates that none of the standard distributions seem to fit the complex real world daylight GHI data.

Time Series Analysis

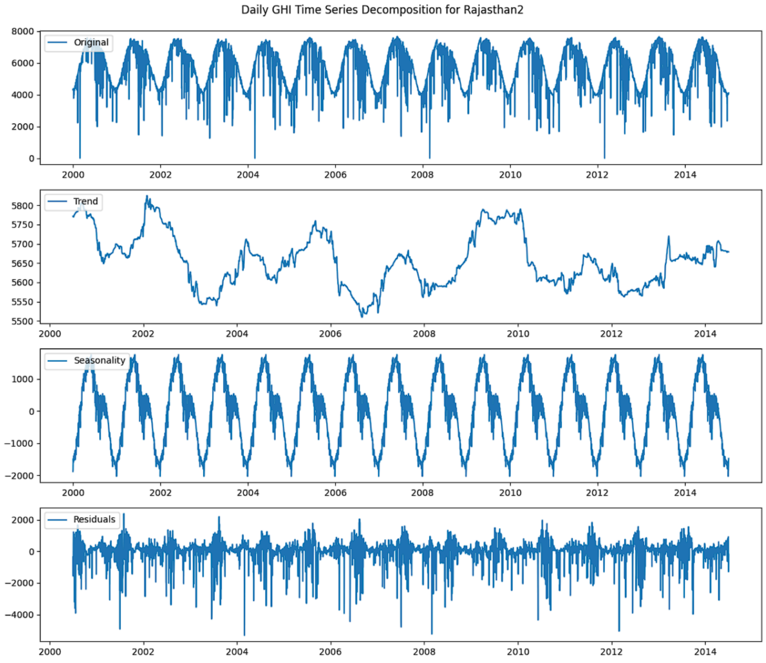
Time Series Decomposition

The time series analysis serves as the foundation for selecting and tuning appropriate forecasting models. This analysis focuses on the Daily Average Global Horizontal Irradiance (GHI) data from 2000 to 2014, obtained by resampling the original hourly data.

The series was decomposed into its three primary components—Trend, Seasonality, and Residuals—using an additive model with an annual period of 365 days.



Rajasthan 1



Rajasthan 2

Autoregressive (AR) Model

Mathematical Formulation:

The autoregressive process of order p, denoted AR(p), is defined as:

$$X_t = C + \sum_{i=1}^{p} \phi_t X_{t-i} + \epsilon_t$$

where:

- X_t is the value of the time series at time t
- c is a constant term (mean adjustment)
- φ_j are the autoregressive coefficients for j = 1, 2, …, p
- $\omega_t \sim N(0, \sigma^2)$ is white noise (Gaussian error term with zero mean and constant variance)
- p is the order of the AR process

In lag operator notation (using the backshift operator B), where $BX_t = X_{t-1}$:

$$\Phi(B)X_t = c + \omega_t$$

where the characteristic polynomial is $\Phi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p$

Autocorrelation Function (ACF):

The autocorrelation function for an AR(p) process satisfies the Yule-Walker equations:

$$\rho_k = \varphi_1 \rho_{k-1} + \varphi_2 \rho_{k-2} + \dots + \varphi_p * \rho_{k-p}, \text{ for } k > p$$

The ACF of AR processes exhibits exponential decay (or damped oscillations for complex roots). Critically, the ACF never cuts off but decays gradually, making AR processes identifiable through this behavior.

Partial Autocorrelation Function (PACF):

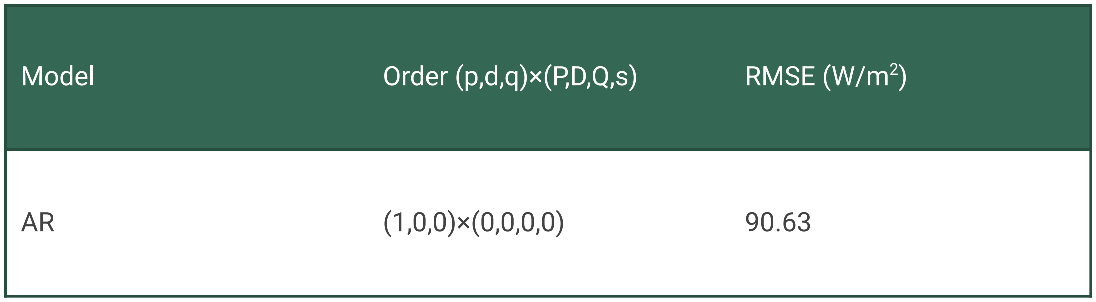
The PACF of an AR(p) process cuts off after lag p—this is the key identifying characteristic. The partial autocorrelation φ_{kk} at lag k can be computed using Cramer's rule from the Yule-Walker equations.

Key limitations

- Require stationarity : AR models assume a constant mean and variance. Non-stationary data must be differenced.
- Poor performance on data with strong moving-average components: If the underlying process is mainly MA, AR models do not perform well even with a large AR order.
- High AR order can lead to overfitting : Choosing a large p makes the model complex and unstable.
- Sensitive to outliers : Since AR uses past observations directly, outliers affect predictions.
- Assumes linear relationships: It cannot capture nonlinear patterns unless transformed.
- Needs long data history: A higher AR order requires many past data points.

Implementation:

The AR forecast quickly flattens out due to the lack of differencing and seasonal terms. Its high RMSE of 90.63 W/m^2 shows it fails to predict any of the GHI's dynamic movements across the year. The slight downward drift observed is an artifact of the training data's characteristics near the test set boundary.



Model	Order (p,d,q)×(P,D,Q,s)	RMSE (W/m^2)
AR	(1,0,0)×(0,0,0,0)	90.63

Moving Average (MA) Model

Mathematical Formulation

The moving average process of order q, denoted MA(q), is defined as:

$X_t = c + \omega_t + \sum_{i=1}^{q} \theta_i * \omega_{t-i}$

where:

- X_t is the value of the time series at time t
- c is the constant (mean) term
- θ_j are the moving average coefficients for j = 1, 2, …, q
- $\omega_t \sim N(0, \sigma^2)$ is white noise
- ω_{t-j} are past error terms (not past observations)
- q is the order of the MA process

In lag operator notation:

$X_t = c + \Theta(B)\omega_t$

where $\Theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q * B^q$

Stationarity and Invertibility :

Unlike AR models, MA processes are always stationary, regardless of parameter values, because the error terms form a finite sum. However, invertibility is a critical condition: an MA(q) process is invertible if all roots of Θ(B) = 0 lie outside the unit circle. Invertibility ensures a unique AR(∞) representation and is essential for parameter estimation.

Autocorrelation Function (ACF) :

The ACF of an MA(q) process has a distinctive property it cuts off after lag q:

$$\rho_k = \begin{cases} 1 & k = 0 \\ \dfrac{\theta_k + \theta_1\theta_{k+1} + \cdots + \theta_{q-k}\theta_q}{\sigma_\omega^2(1 + \theta_1^2 + \cdots + \theta_q^2)} & 0 < k \le q \\ 0 & k > q \end{cases}$$

This cutoff behavior is the identifying characteristic of MA processes.

Partial Autocorrelation Function (PACF):

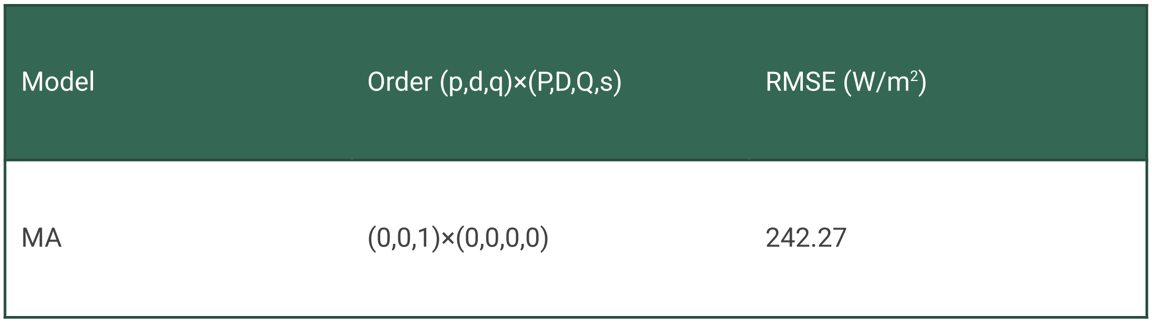
The PACF of an MA(q) process exhibits exponential decay (or damped oscillations), theoretically extending to infinity. The PACF decays gradually but never cuts off, making it complementary to the ACF for identification.

Key limitations

- Error terms are unobservable: They must be estimated, which increases uncertainty and makes the model sensitive to mis-specification.
- Model identification is harder: Finding the correct MA(q) order is more difficult than AR(p).
- It fails for data with a strong autoregressive structure: MA alone cannot capture long-term dependencies.
- It is strongly affected by initial estimation errors because residuals must be estimated recursively.
- It assumes stationarity: Like AR, non-stationary series must be differenced first.

Implementation:

The MA model performed the worst, with a substantial RMSE of 242.27 W/m2. The forecast line immediately drops to near zero and stays there for most of the year. In a simple MA(1) model, the long-term forecast tends toward the series mean, but for GHI, this mean is far from the seasonal peaks, leading to catastrophic errors.



Model	Order (p,d,q)×(P,D,Q,s)	RMSE (W/m^2)
MA	(0,0,1)×(0,0,0,0)	242.27

ARMA(p,q) Model

The combined autoregressive moving average model incorporates both components:

$$X_t = c + \sum_{j=1}^{p} \varphi_j X_{t-j} + \omega_t + \sum_{j=1}^{q} \theta_j \omega_{t-j}$$

or in lag operator form: $\Phi(B)X_t = c + \Theta(B)\omega_t$

ARMA models provide flexibility: use the ACF/PACF diagnostic plots to identify appropriate orders. If PACF cuts off at lag p and ACF gradually decays, use AR(p); if ACF cuts off at lag q and PACF gradually decays, use MA(q); if both decay gradually, use ARMA(p,q).

Key limitations

- Requires strict stationarity: ARMA cannot handle trends or seasonality unless you preprocess the data through differencing or decomposition.
- Cannot model seasonal patterns: You need to use SARIMA to work with seasonal data.
- Parameter estimation can be complex: Estimating AR and MA terms together is computationally heavy and often gets stuck in local minima.
- Model selection (p, q) is difficult: You need to perform AIC/BIC analysis; making an incorrect choice can lead to unstable or divergent models.
- Assumes linearity: It cannot capture nonlinear dynamics, volatility clustering, or structural breaks.

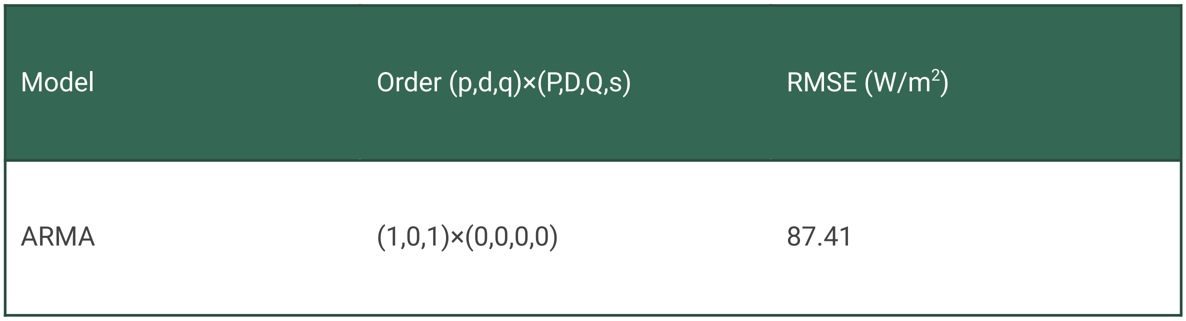
Practical Implications

AR models are particularly useful for capturing momentum and mean-reversion patterns (common in financial returns and physical processes), while MA models excel at smoothing shocks and correcting for measurement errors. The choice between AR and MA depends on the data's autocorrelation structure and the physical mechanism generating the series. Stationary time series data (verified via Augmented Dickey-Fuller test) is a prerequisite for reliable parameter estimation in both model classes.

AR, MA, ARMA models just predict a straight line as they use only previous data to predict the next day. Therefore, when predicting over a year, they just predict an average without any fluctuations due to seasonality

Implementation:

With an RMSE of 87.41 W/m^2, the ARMA forecast is effectively a flat line. Since the GHI series is non-stationary and highly seasonal, the ARMA model immediately stabilizes to a constant value close to the mean of the training data or the start of the test period. This demonstrates its complete inadequacy for modeling data with strong annual cycles.



Model	Order (p,d,q)×(P,D,Q,s)	RMSE (W/m^2)
ARMA	(1,0,1)×(0,0,0,0)	87.41

ARIMA (Autoregressive Integrated Moving Average) Model

Mathematical Formulation

The ARIMA(p,d,q) model generalizes the ARMA model by adding an initial "integration" step (differencing) to handle non-stationarity. It is defined as:

$$\phi(B)(1 - B)^d X_t = c + \Theta(B)\omega_t$$

where:

- X_t is the value of the time series at time $t$
- c is a constant term
- $(1 - B)^d$ is the differencing operator of order d (where d is usually 1 or 2)
- $\phi(B)$ is the autoregressive polynomial of order p
- $\Theta(B)$ is the moving average polynomial of order q
- $\omega_t \sim N(0, \sigma^2)$ is white noise

In simple terms, an ARIMA model is an ARMA model running on data that has been differenced d times to achieve stationarity.

- If $d = 0$: $X_t = X_t$ (The model becomes ARMA)
- If $d = 1$: The model analyzes the change $\Delta X_t = X_t - X_{t-1}$
- If $d = 2$: The model analyzes the change in the changes (acceleration).

Stationarity and Differencing

The "Integrated" (I) component is the defining feature of ARIMA. Many real-world time series (like stock prices or GDP) have a trend and are non-stationary (the mean changes over time). AR and ARMA models fail on such data.

ARIMA solves this by applying differencing until the data becomes stationary (constant mean and variance). Once stationary, the standard AR and MA terms can model the dynamics.

Key Limitations

- **Cannot model seasonality:** While ARIMA handles trends via differencing, it fails to capture repeating cycles (e.g., monthly sales, daily solar irradiance). It treats seasonal waves as simple autocorrelation, often leading to poor long-term forecasts.
- **Parameter selection is complex:** Choosing the correct combination of (p, d, q) is difficult. Over-differencing (d is too high) introduces artificial memory, while under-differencing leaves the model non-stationary.
- **Sensitive to outliers:** Like AR and MA, ARIMA assumes errors are normally distributed. A single extreme event can skew the coefficients and ruin future predictions.
- **Linearity assumption:** It assumes the future is a linear combination of the past. It cannot capture non-linear structural breaks or volatility changes

Implementation:

The RMSE of 81.44 W/m^2 is significantly higher than that of the SARIMA model. The forecast shows a slight downward slope but fails to capture the rising GHI through spring, the summer peak, or the steep decline in autumn. It acts primarily as a persistence forecast based on the last observed value after differencing, quickly missing the dominant annual pattern.

Model	Order (p,d,q)×(P,D,Q,s)	RMSE (W/m^2)
ARIMA	(1,1,1)×(0,0,0,0)	81.44

SARIMA (Seasonal ARIMA) Model

Mathematical Formulation

The Seasonal ARIMA model, denoted as SARIMA(p,d,q)(P,D,Q)s, extends ARIMA by adding seasonal counterparts to the autoregressive, integrated, and moving average terms.

The full multiplicative model is written as:

$$\phi_P(B^s)\phi_p(B)(1 - B^s)^D(1 - B)^D X_t = c \; + \phi_Q(B^s)\phi_q(B)\omega_t$$

where:

- **Non-Seasonal Terms:** (p, d, q) describe the immediate relationships (e.g., today depends on yesterday).
- **Seasonal Terms:** (P, D, Q) describe the periodic relationships.
- s **(Seasonality):** The number of time steps in a single cycle (e.g., $s = 12$ for monthly data, $s = 52$ for weekly, $s = 24$ for hourly).
- $\phi_P(B^s)$**:** Seasonal Autoregressive component (looks back s steps).
- $(1 - B^s)^D$ **:** Seasonal Differencing (subtracting this year from last year).

Seasonality and Identification

SARIMA is explicitly designed for data with repeating cycles.

- **Seasonal AR (P):** Captures the correlation between X_t and X_{t-s} (e.g., predicting this January based on last January).

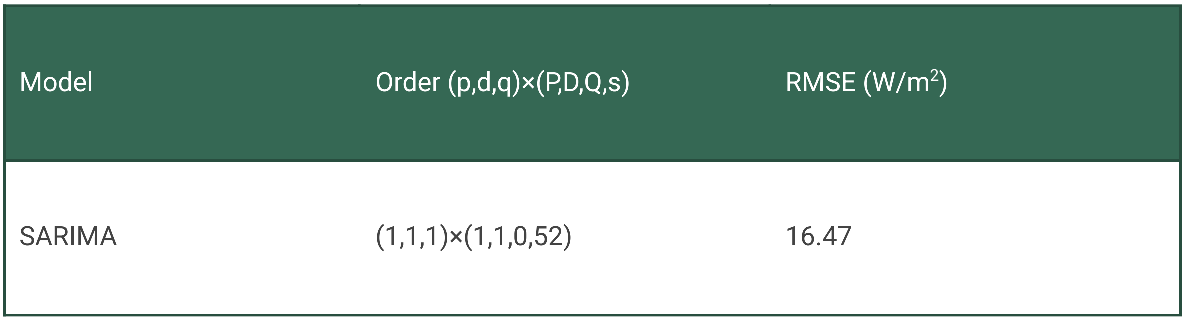
- **Seasonal Differencing (D):** Removes the seasonal pattern to make the series stationary. For example, if solar energy is always high in June and low in December, seasonal differencing $D = 1$removes this "wave" so the model can focus on the anomalies.

Key Limitations

- **High Computational Cost:** SARIMA models are computationally intensive, especially with large seasonal periods (e.g., daily data with $s = 365$). The optimization algorithm must calculate correlations over long lag periods.
- **Requires Extensive Data History:** To estimate seasonal parameters effectively, the dataset must contain multiple full cycles. For example, to model annual seasonality($s = 12$), you generally need at least 3-4 years of data.
- **Complexity:** With 7 parameters $(p, d, q)(P, D, Q)s$ the risk of overfitting is high. A model might fit historical data perfectly but fail to predict the future (low generalization).
- **Assumption of Constant Seasonality:** SARIMA assumes the seasonal pattern is fixed over time. It struggles if the seasonality evolves (e.g., climate change shifting the onset of the monsoon).

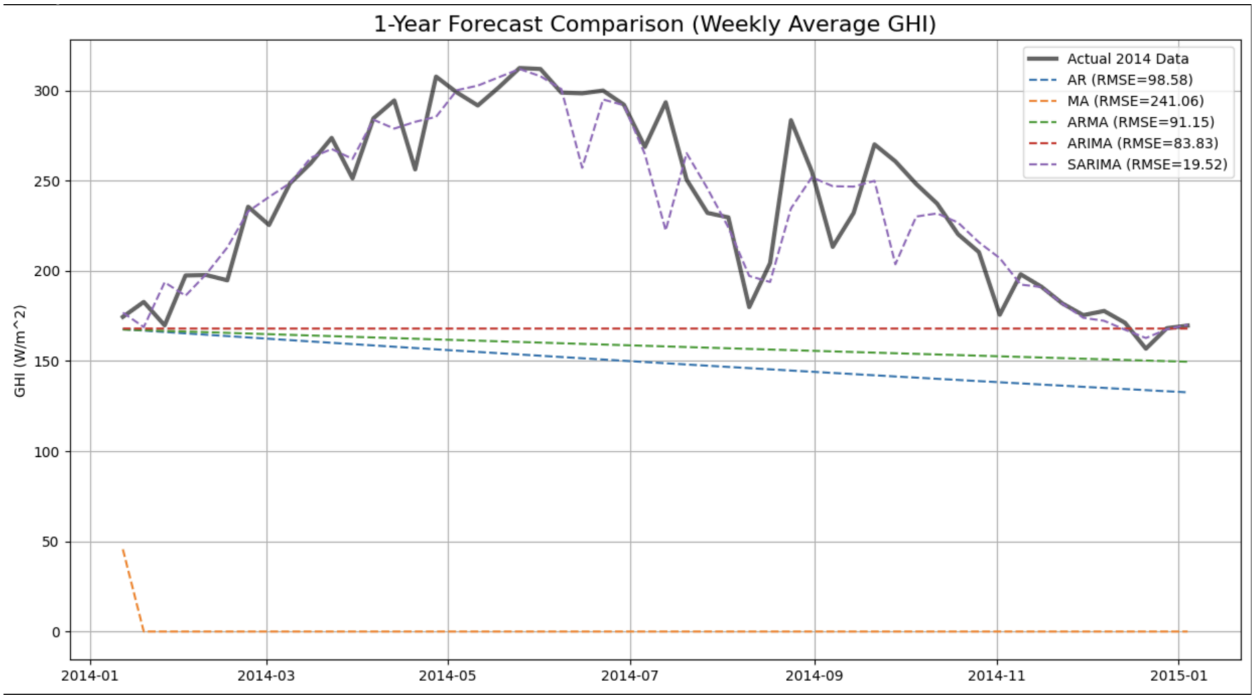
Implementation:

The SARIMA model's forecast successfully captures both the overall trend (the annual solar cycle) and the week-to-week variations. Visually, the SARIMA forecast line closely tracks the actual GHI data, confirming the importance of the seasonal parameters (P=1, D=1, s=52) for this type of data.



Model	Order (p,d,q)×(P,D,Q,s)	RMSE (W/m^2)
SARIMA	(1,1,1)×(1,1,0,52)	16.47

Summary



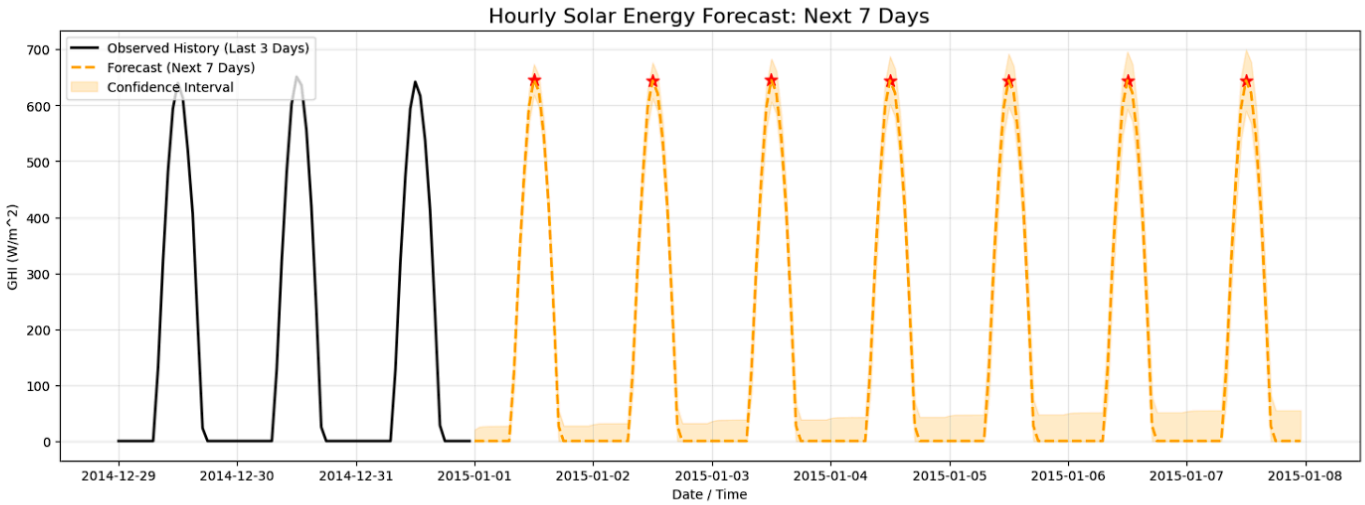
Model Type	Best Performer
Seasonal	SARIMA (16.47 W/m2)
Non-Seasonal	ARIMA (81.44 W/m2)

The **SARIMA** model provided a prediction that is **4.9 times more accurate** (based on the ratio of RMSEs: 81.44 / 16.47 = 4.9) than the best non-seasonal model (ARIMA).

Daily Forcasting

To facilitate high-precision operational forecasting for the upcoming 168-hour (7-day) horizon, the SARIMA model was recalibrated using a **sliding window approach**. Specifically, the model was retrained exclusively on the most recent 60 days (1,440 hours) of observed data. This strategy prioritizes temporal relevance, ensuring the model captures immediate meteorological trends and the current solar geometry while discarding outdated historical noise.

For the short-term operational forecast, we restricted the training dataset to the trailing 60 days. This focused training window allowed the model to adapt to the specific seasonal characteristics of the current month, resulting in a highly granular hourly forecast for the next week that accounts for recent shifts in sunrise/sunset times and prevailing cloud cover patterns.



Conclusion

- This study successfully analyzed 15 years of solar data for two regions in Rajasthan. The analysis confirmed that while Rajasthan has immense solar potential, it is subject to significant seasonal variance, particularly during the Monsoon.

- Weekly forecasting was found to be more accurate (with any model) as compared to daily forecasting. This is because daily data tends to have much more random variation as compared to weekly data.

- The SARIMA model proved to be the most effective tool for forecasting, capable of capturing both the daily "bell curve" and the annual seasonal shifts. The generated hourly forecasts for the upcoming week provide actionable insights for grid management and energy dispatch planning.

References

- Clay Ford, Understanding Q-Q Plots, University of Virginia Library, August 26, 2015, data.library.virginia.edu/understanding-q-q-plots

- Engineering Statistics Handbook, Introduction to Time Series Analysis, www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm

- Stationary data checking: https://machinelearningmastery.com/time-series-data-stationary-python/

- Augmented Dickey fuller test: https://en.wikipedia.org/wiki/Augmented_Dickey%E2%80%93Fuller_test

- Quantile quantile plot: https://en.wikipedia.org/wiki/Q%E2%80%93Q_plot