Statistical Analysis and Forecasting of Solar Energy (Intra-State)

Group 5 - Pearson Group

Raj Sanjay Shah	2017A7PS1181P
Nisarg Vora	2018A7PS0254P
Snigdh Sanjivani	2018A3PS0408P
Atith Gandhi	2017A7PS0062P
Reventh Sharma	2017A1PS0832P
Shray Mathur	2017A7PS1180P
Mustafa	2017B4A30894P

Note: Pranav Verma - 2017B4A40617P has withdrawn from the course

November 27,2020

Submitted in partial fulfillment to the course
Applied Statistical Methods

Submitted to
Dr. Sumanta Pasari
Mathematics Department
Birla Institute of Technology and Science, Pilani
Rajasthan, India

Table of Contents

1 Introduction

The demand for renewable energy is proliferating. The world is shifting from classical energy resources like thermal energy to environment-friendly ones like solar, wind, tidal, hydro and biomass. This change is being further encouraged by technological advancements taking place, making the shift efficient.

India is one of the largest renewable energy-producing nations in the world, amounting to 36% of the country's total electricity generated. In the 2015 Paris Agreement, India set a target of generating 40% of total electricity by non-fossil fuel sources by 2030. India is also aiming to produce more than 50% of electricity by renewable energy sources by 2027, which is an ambitious goal. A large number of wind and solar energy plants are being installed for this purpose.

Solar and wind energy together constitute 83% of the total renewable energy capacity. Solar energy is an essential renewable energy source and is increasingly being used for energy generation. It requires photovoltaic cells which convert sunlight into electricity. However, the environment-friendly nature of this energy source comes with some drawbacks. Integration of generation plants with inefficient transmission lines results in a large amount of power wastage. The unavailability of solar radiation at night is another problem as the demand for electricity is at its peak at that time.

Similar is the case with wind energy generation. Wind energy is generated through wind turbines which converts wind's mechanical energy into electricity. This too depends heavily on the wind speed and weather conditions.

Therefore, to tackle these issues, analysis and forecasting of renewable energy resources becomes essential. By estimating the amount of solar radiation received in an area along with other factors, a better and efficient decision can be taken on where to install the plants. Similarly, by estimating average wind speed throughout a period, a suitable target for wind-farms can be selected. In places where these are already installed, forecasting can help prevent a crisis by predicting cloudy days or less windy days.

In this report, we focus on statistical analysis and forecasting of solar energy at different locations in a given state. Since solar energy is dependent on the total amount of solar radiation received, we primarily use this variable for our analysis.

Several forecasting methods can be applied for solar irradiation like stochastic methods, autoregressive models, machine learning techniques. The pros and cons of these methods are discussed in later sections.

2 Data Description and Terminology

This report aims to analyse the solar energy for five study regions of the same state. For this purpose, we obtain hourly data of five solar parks for the state of Rajasthan from 2000-2014.

The solar radiation, which enters the earth's atmosphere, gets divided into three essential components. A part of it reaches directly to the surface of the earth and is called normal radiation. The remaining part gets scattered in the earth's atmosphere and then reaches the surface. Such radiation is called diffuse radiation. Another part of the radiation gets reflected from secondary surfaces and then reaches the desired surface and is called the albedo. The total amount of radiation received per unit area is a significant number for solar energy analysis.

Total Solar Irradiance (TSI) is the solar power (all wavelengths) per unit area falling upon the Earth's upper atmosphere. Direct Normal Irradiance(DNI), or beam irradiance, is the total radiation received per unit area by a surface which is held perpendicular to the rays coming from the Sun. Diffuse Horizontal

Irradiance (DHI) is the radiation that does not come directly from the Sun but which is scattered in the atmosphere. It comes equally from all directions, excluding the radiation coming from the sun disk and is measured on a horizontal surface. Global Horizontal Irradiance (GHI), is the total short-wave irradiance from the Sun received by a surface horizontal to the Earth. It is the sum of direct normal irradiance(DNI) (after accounting for the solar zenith angle of the sun z) and diffuse horizontal irradiance(DHI):

$$GHI = DHI + DNI*\cos(\theta)$$

Here, θ is the solar zenith angle, which is the angle between the rays of the sun and vertical normal. Our primary goal is to analyze and forecast the total solar radiation for the next day, week and month. The most important variable for this is the Global Horizontal Irradiance or GHI as it is the direct measure of the radiation, which corresponds to the energy generated by the solar cells. There are many other variables like dew, temperature, relative humidity, pressure which are not very important in this analysis and therefore have not been used. Further, the correlation matrix calculated from the data in the next section shows the same result.

3 Methodology and Results

3.1 Data Analysis

Given below is the methodology flow chart used in the time series analysis and forecasting of the solar park data.

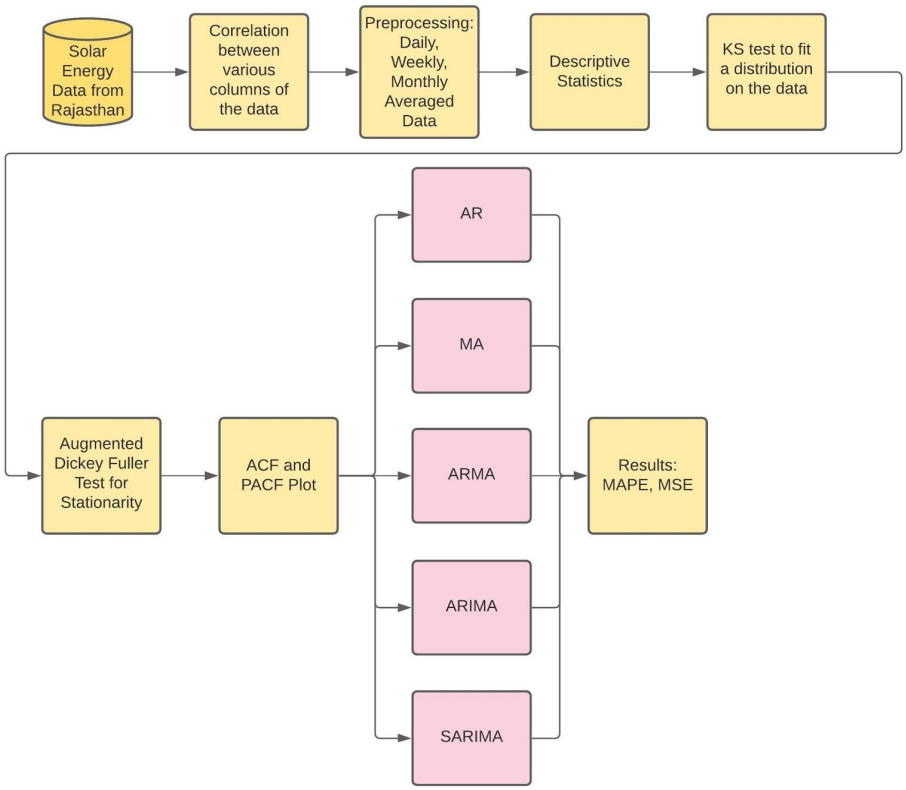

Figure 3.1 Methodology overview flowchart.

Solar energy data was obtained for 5 different locations in Rajasthan. After obtaining the data set, correlation between the various columns has been analysed which helped in identifying how the columns were linked to each other. The hourly data was then preprocessed to convert it into daily, weekly and monthly data by simply averaging the values over the respective time periods(since we are mainly concerned with the daily, weekly and monthly values, hourly fluctuations wouldn't affect the results). To perform a preliminary analysis of the data, various descriptive statistics of DHI, DNI and GHI values have been carefully analysed. An attempt was made to identify the underlying distribution on the values from the set of 10 different well-known distributions using K-S Goodness of fit test. Time series forecasting of the GHI data was done using Autoregressive, Moving Average, Autoregressive Moving Average, Autoregressive Integrated Moving Average and Seasonal Autoregressive Moving Average models. We find the parameters of the models from the first fourteen years of data of Rajasthan 1 location and we test our forecast on the year 2014 (fifteenth year) for all five regions of Rajasthan. The results were then compared with an LSTM (Long Short Term Memory) based machine learning model. Python libraries: Scikit Learn, pandas, numpy, scipy, keras, and statsmodels have been used for preprocessing and time series forecasting using ARIMA based models and machine learning models. MS Excel was used for descriptive analytics of the data.

3.1.1 Correlation Analysis.

Correlation analysis between various columns of the data set is the key to understanding how strongly the various columns of the solar energy data are linked to each other.

Given below is a 12x12 matrix denoting the correlation between all the pairs of columns:

	DHI	DNI	GHI	Clearsky DHI	Clearsky DNI	Clearsky GHI	Dew Point	Temperature	Pressure	Relative Humidity	Solar Zenith Angle	Wind Speed
DHI	1.000000	0.810126	0.927168	0.978921	0.897983	0.952093	0.136632	0.614919	-0.165429	-0.204736	-0.886254	-0.082755
DNI	0.810126	1.000000	0.940300	0.840339	0.947746	0.902440	-0.056640	0.466896	0.079642	-0.316748	-0.803984	-0.196673
GHI	0.927168	0.940300	1.000000	0.940370	0.938249	0.985022	0.063107	0.591803	-0.076907	-0.262982	-0.869622	-0.123800
Clearsky DHI	0.978921	0.840339	0.940370	1.000000	0.898374	0.959677	0.136944	0.639096	-0.177787	-0.219523	-0.899532	-0.075824
Clearsky DNI	0.897983	0.947746	0.938249	0.898374	1.000000	0.949993	-0.023263	0.484311	0.048106	-0.286224	-0.865811	-0.202176
Clearsky GHI	0.952093	0.902440	0.985022	0.959677	0.949993	1.000000	0.083277	0.596223	-0.098308	-0.241034	-0.888108	-0.121181
Dew Point	0.136632	-0.056640	0.063107	0.136944	-0.023263	0.083277	1.000000	0.495483	-0.778792	0.805823	-0.174782	0.289728
Temperature	0.614919	0.466896	0.591803	0.639096	0.484311	0.596223	0.495483	1.000000	-0.644223	-0.035681	-0.619844	0.108974
Pressure	-0.165429	0.079642	-0.076907	-0.177787	0.048106	-0.098308	-0.778792	-0.644223	1.000000	-0.504076	0.186726	-0.395237
Relative Humidity	-0.204736	-0.316748	-0.262982	-0.219523	-0.286224	-0.241034	0.805823	-0.035681	-0.504076	1.000000	0.156985	0.223948
Solar Zenith Angle	-0.886254	-0.803984	-0.869622	-0.899532	-0.865811	-0.888108	-0.174782	-0.619844	0.186726	0.156985	1.000000	0.091256
Wind Speed	-0.082755	-0.196673	-0.123800	-0.075824	-0.202176	-0.121181	0.289728	0.108974	-0.395237	0.223948	0.091256	1.000000

Table 3.1. Correlation matrix

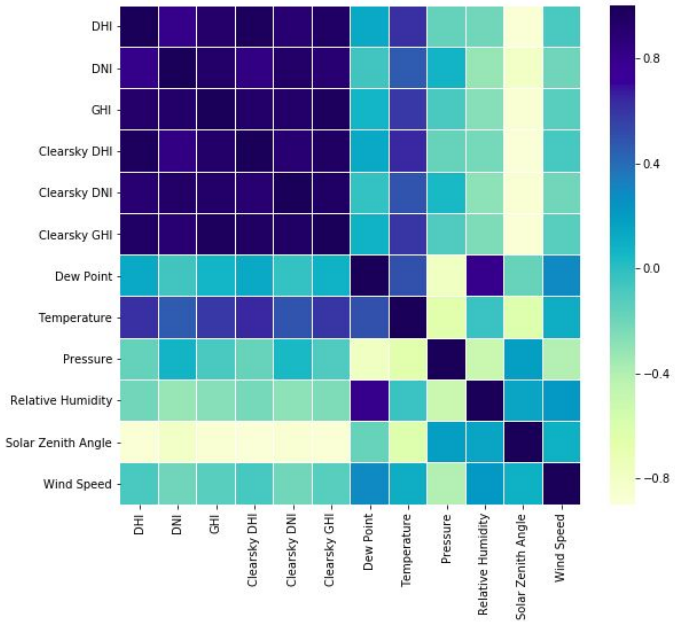

Figure 3.2. Correlation matrix visualization.

From the results above we can clearly observe strong positive correlation among GHI, DHI and DNI values and negative correlation between GHI, DHI, DNI with Solar Zenith angle. This negative correlation is because of the increase in atmospheric diffusion with the increase in solar zenith angle. We observe that there is no tangible correlation of GHI, DHI and DNI with pressure and wind speed. We also notice a positive correlation of GHI, DHI and DNI with temperature, this is justified by increasing temperatures during seasons with longer daylight hours.

3.1.2 Preprocessing

The hourly solar energy data had column values as 0 for more than 12 hours each day. For the purpose of identifying daily, weekly and monthly trends in solar energy, the hourly solar energy data was averaged for the respective time periods. Since, we were primarily interested in the daily, weekly and monthly forecast averaging the hourly data would suffice and would not affect the final results.

3.1.3 Graphical plots and Descriptive Statistics for DHI, DNI and GHI data.

We calculate various descriptive statistics to perform exploratory analysis of the data like the mean, median, skewness etc. Furthermore, we try to fit a distribution to the data. If we can fit a distribution on the data, we can obtain a confidence interval. While forecasting is difficult from a confidence interval, we can reject our time series forecasting models if they predict values outside of the obtained confidence interval.

3.1.3.1 DHI data

DHI	
Mean	96.77270167
Standard Error	0.378276294
Median	94.75
Mode	58.79166667
Standard Deviation	27.98988974
Sample Variance	783.4339275
Kurtosis	-1.344021114
Skewness	0.053379949
Range	158.4166667
Minimum	0
Maximum	158.4166667
Sum	529830.5417
Count	5475

Table 3.3 Daily DHI data descriptive statistics.

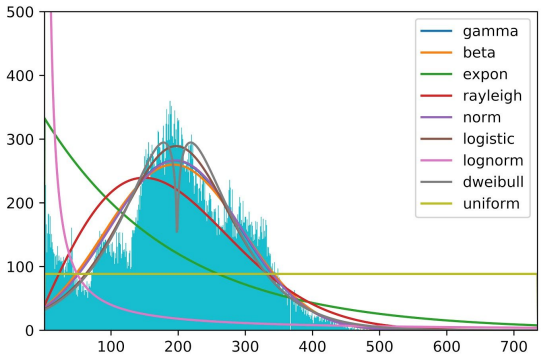

Figure 3.3 Distribution fitting on Hourly DHI data graph.

For the purpose of identifying the distribution of the DHI data we perform Goodness-of-Fit Test for 10 different distributions using **K-S Test Statistic**. We also use the distfit library to check the best fitting distribution using the residual sum of squares method. Model parameters of the distributions are estimated using the maximum likelihood estimation (MLE) method. The table for the same is given below:

	Statistic	p-value
Logistic:	0.030582887632595132,	3.882416059981373e-53
Norm:	0.032840031493179454,	3.320931948078452e-61
Gamma:	0.03347337156211639,	1.426605442167479e-63
Beta:	0.034254927963903725,	1.482871578142235e-66
Dweibull:	0.0343522573053158,	6.234140719501517e-67
Expon:	0.25307019949878806,	0.0
Rayleigh:	0.10638123747551981,	0.0
Lognorm:	0.6038148433539875,	0.0
Uniform:	0.4773310810640561 4,	0.0

Table 3.3 K-S Goodness of fit test statistic for fitting different distributions on the data.

From the above results we reject the null hypothesis at a significance level of 1% and hence DHI data does not fit any of the 10 distributions.

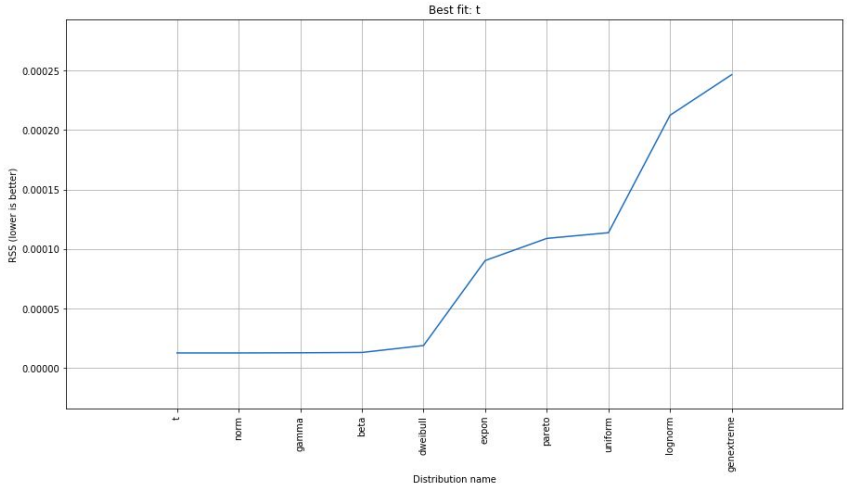

Figure 3.4 Underlying Distribution of hourly DHI data based on residual sum of squares

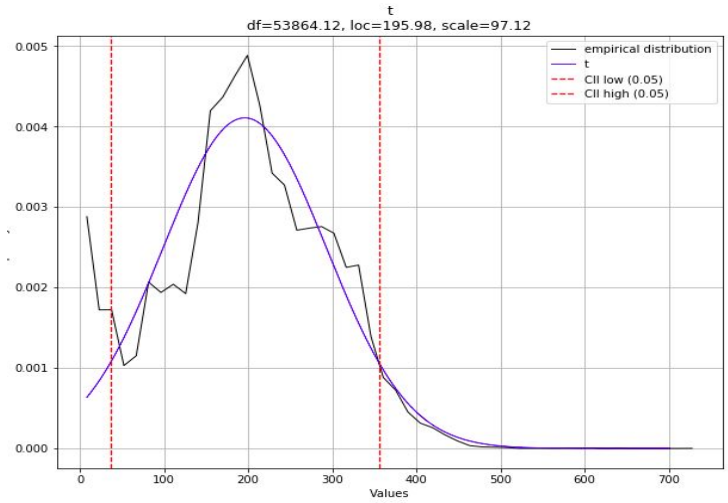

Figure 3.5 Underlying Distribution of hourly DHI data based on residual sum of squares

Residual sum of squares of different distributions:


```
{'distr': <scipy.stats._continuous_distns.t_gen object at 0x7fca615ec748>, 'params':
(53864.1207759269, 195.97997448300316, 97.11518242029648), 'name': 't', 'RSS':
1.2515499942265105e-05, 'loc': 195.97997448300316, 'scale': 97.11518242029648, 'arg':
(53864.1207759269,), 'CII_min_alpha': 36.236967095306994, 'CII_max_alpha': 355.7229818706993}
```


	distr	RSS	LLE	loc	scale
0	t	1.25155e-05	NaN	195.98	97.1152
1	norm	1.25156e-05	NaN	195.979	97.1165
2	gamma	1.26651e-05	NaN	-6797.99	1.35007
3	beta	1.29204e-05	NaN	-296.183	1060.34
4	dweibull	1.8751e-05	NaN	198.467	82.3959
5	expon	9.02374e-05	NaN	1	194.979
6	pareto	0.000108864	NaN	-4.33462e+06	4.33462e+06
7	uniform	0.000113647	NaN	1	734
8	lognorm	0.000212433	NaN	1	3.13745
9	genextreme	0.000246582	NaN	733.804	9.10309

Observations: The K-S Goodness of fit test rejects all the distributions at 1% confidence levels for DHI data. Through the residual sum of squares method we observe that t distribution is the best fit among all the other distributions (Best fitting among all distributions, but not a good fit by K-S test).

3.1.3.2 DNI data

DNI	
Mean	207.3129
Standard Error	0.634119
Median	216.6667
Mode	220.4167
Standard Deviation	46.92052
Sample Variance	2201.535
Kurtosis	3.133871
Skewness	-1.53958
Range	301.25
Minimum	0
Maximum	301.25
Sum	1135038
Count	5475

Table 3.4 Daily DNI data descriptive statistics.

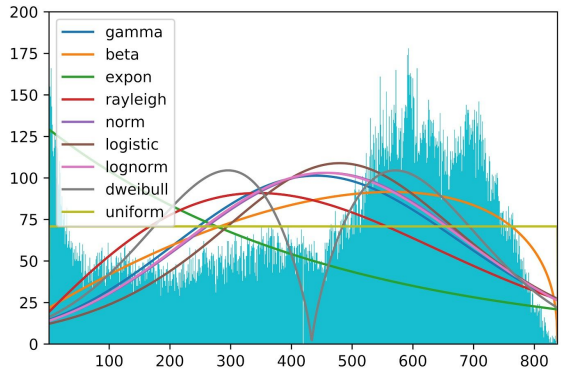

Figure 3.6 Distribution fitting on Hourly DNI data graph.

For the purpose of identifying the distribution of the DNI data we perform Goodness-of-Fit Test for 10 different distributions using **K-S Test Statistic**. We also use the distfit library to check the best fitting distribution using the residual sum of squares method. Model parameters of the distributions are estimated using the maximum likelihood estimation (MLE) method. The table for the same is given below:

	Statistic	p-value
Gamma:	0.1303817229867989,	0.0
Beta:	0.09343795258789994,	0.0
Expon:	0.22458030942625107,	0.0
Rayleigh:	0.1632846867626736,	0.0
Norm:	0.11858502546932875,	0.0
Logistic:	0.08932556718339024,	0.0
Lognorm:	0.12226814002278935,	0.0
Dweibull:	0.12544514981362254,	0.0
Uniform:	0.13936134852543042,	0.0

Table 3.5 K-S Goodness of fit test statistic for fitting different distributions on the data.

From the above results we reject the null hypothesis at a significance level of 1% and hence GHI data does not fit any of the 10 distributions.

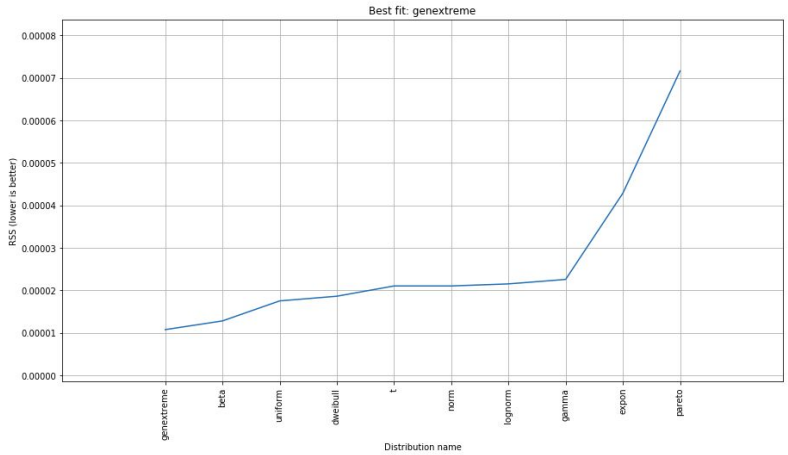

Figure 3.7 Underlying Distribution of hourly DNI data based on residual sum of squares

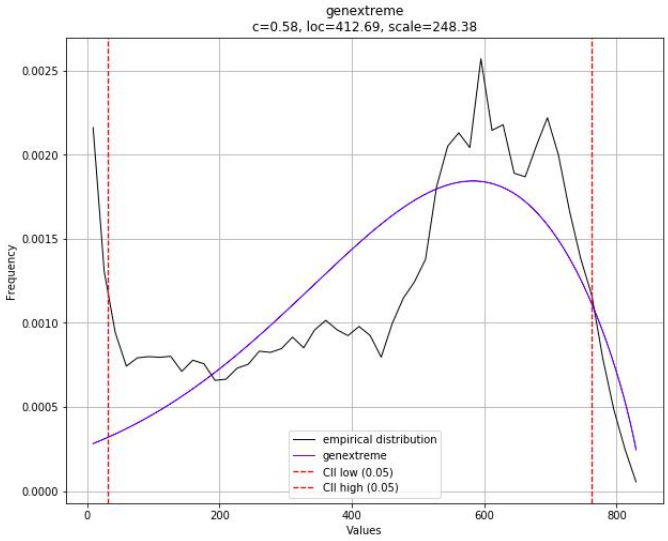

Figure 3.8 Underlying Distribution of hourly DNI data based on residual sum of squares

Residual sum of squares of different distributions:

{'distr': <scipy.stats._continuous_distns.genextreme_gen object at 0x7fca6161afd0>, 'params': (0.5839553812139646, 412.68895323837467, 248.38130717143812), 'name': 'genextreme', 'RSS': 1.078666292014243e-05, 'loc': 412.68895323837467, 'scale': 248.38130717143812, 'arg': (0.5839553812139646,), 'CII_min_alpha': 30.804807899642697, 'CII_max_alpha': 762.9609104867719}

	distr	RSS	LLE	loc	scale
0	genextreme	1.07867e-05	NaN	412.689	248.381
1	beta	1.28347e-05	NaN	-90.6243	928.633
2	uniform	1.75411e-05	NaN	1	837
3	dweibull	1.86444e-05	NaN	433.535	224.402
4	t	2.10579e-05	NaN	459.701	229.522
5	norm	2.10589e-05	NaN	459.7	229.514
6	lognorm	2.15453e-05	NaN	-22494.8	22952.3
7	gamma	2.25961e-05	NaN	-3068.59	15.5
8	expon	4.28584e-05	NaN	1	458.7
9	pareto	7.16174e-05	NaN	-22348.6	22348.8

Observations: The K-S Goodness of fit test rejects all the distributions at 1% confidence levels for DNI data. Through the residual sum of squares method we observe that the genextreme distribution is the best fit among all the other distributions (Best fitting among all distributions, but not a good fit by K-S test).

3.1.3.3 GHI data

We primarily deal with the GHI data as major companies like Homer Pro use Solar GHI to compute flat-panel PV output. This makes GHI of more interest for the calculation of solar energy obtained from a solar park.

3.1.3.3.1 GHI Hourly Data

GHI	
Mean	237.8900685
Standard Error	0.670737373
Median	244.3333333
Mode	302.5
Standard Deviation	49.6300334
Sample Variance	2463.140215
Kurtosis	-0.766381511
Skewness	-0.381571748
Range	317.5833333
Minimum	0
Maximum	317.5833333
Sum	1302448.125
Count	5475

Table 3.6 Daily GHI data descriptive statistics.

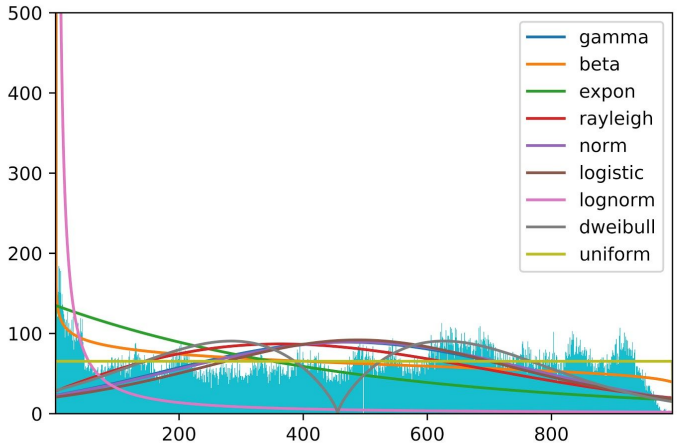

Figure 3.9 Distribution fitting on Hourly GHI data graph.

For the purpose of identifying the distribution of the GHI data we perform Goodness-of-Fit Test for 10 different distributions using **K-S Test Statistic**. We also use the distfit library to check the best fitting distribution using the residual sum of squares method. Model parameters of the distributions are estimated using the maximum likelihood estimation (MLE) method. The table for the same is given below:

	Statistic	p-value
Uniform:	0.04601010911036778,	9.91023327256035e-120
Dweibull:	0.07084082204684172,	2.9865938048258525e-283
Norm:	0.07204284047698639,	6.249408062327285e-293
Logistic:	0.07378314543180545,	3.1140877302525494e-307
Gamma:	0.0762687370501175	0.0
Beta:	0.09159433735281952,	0.0
Expon:	0.1641034334100575,	0.0
Rayleigh:	0.09876302494214229,	0.0
lognorm:	0.7012957592632292,	0.0

Table 3.7 K-S Goodness of fit test statistic for fitting different distributions on the data.

From the above results we reject the null hypothesis at a significance level of 1% and hence GHI data does not fit any of the 10 distributions.

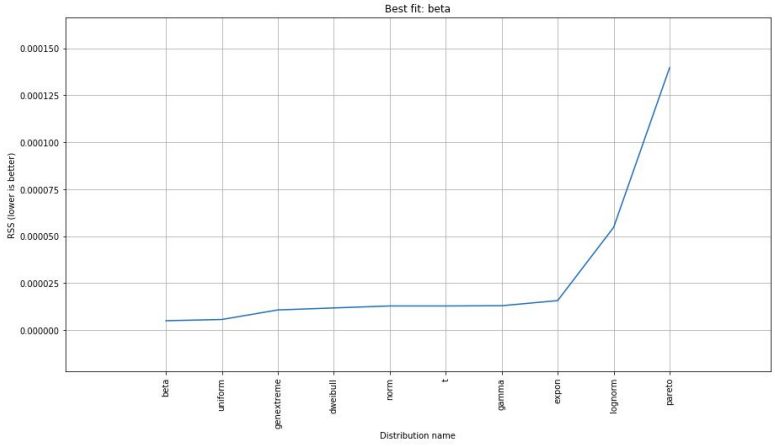

Figure 3.10 Underlying Distribution of hourly GHI data based on residual sum of squares

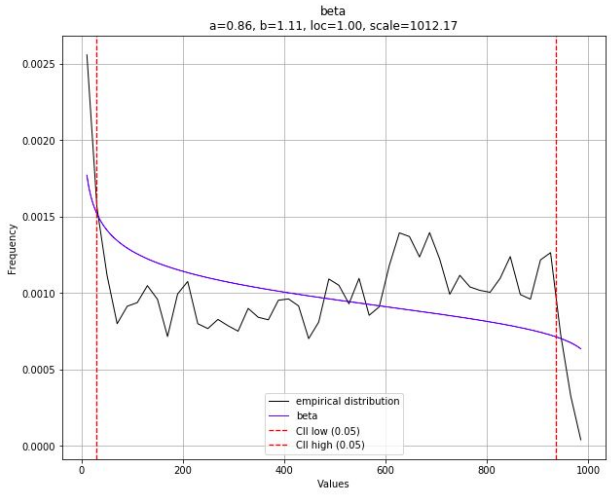

Figure 3.11 Underlying Distribution of hourly GHI data based on residual sum of squares

Residual sum of squares of different distributions:

{'distr': <scipy.stats._continuous_distns.beta_gen object at 0x7fca618df908>, 'params': (0.8613503611352078, 1.1083463243019511, 0.9999999999997742, 1012.1666929670664), 'name': 'beta', 'RSS': 4.954694852282682e-06, 'loc': 0.9999999999997742, 'scale': 1012.1666929670664, 'arg': (0.8613503611352078, 1.1083463243019511), 'CII_min_alpha': 29.091558660510948, 'CII_max_alpha': 935.2796936109214}

	distr	RSS	LLE	loc	scale
0	beta	4.95469e-06	NaN	1	1012.17
1	uniform	5.60773e-06	NaN	1	994
2	genextreme	1.06965e-05	NaN	415.377	317.105
3	dweibull	1.17641e-05	NaN	455.553	282.408
4	norm	1.28112e-05	NaN	481.764	289.647
5	t	1.28112e-05	NaN	481.765	289.646
6	gamma	1.29442e-05	NaN	-6342.58	12.3773
7	expon	1.56774e-05	NaN	1	480.764
8	lognorm	5.46903e-05	NaN	1	0.673749
9	pareto	0.000139736	NaN	-4.16653	5.16653

Observations: The K-S Goodness of fit test rejects all the distributions at 1% confidence levels for GHI data. Through the residual sum of squares method we observe that the beta distribution is the best fit among all the other distributions (Best fitting among all distributions, but not a good fit by K-S test).

3.1.3.3.2 GHI Daily Data

GHI	
Mean	237.8900685
Standard Error	0.670737373
Median	244.3333333
Mode	302.5
Standard Deviation	49.6300334
Sample Variance	2463.140215
Kurtosis	-0.766381511
Skewness	-0.381571748
Range	317.5833333
Minimum	0
Maximum	317.5833333
Sum	1302448.125
Count	5475

Table 3.8 Daily GHI Data descriptive stats

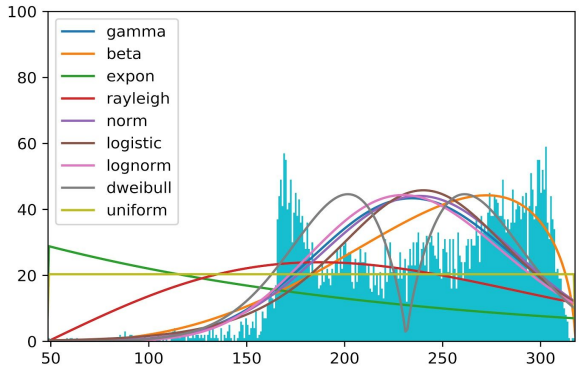

Figure 3.12 Distribution fit on Daily GHI Data graph.

	Statistic	pvalue
Beta:	0.06301271458291366,	2.6435192313260164e-19
Logistic:	0.08301534999624827,	3.4204908965871734e-33
Norm:	0.0939165781874387,	2.3090072580436875e-42
Dweibull:	0.09642725961400034,	1.2337004657283205e-44
Gamma:	0.09919858235746848,	3.2621909715683014e-47
Lognorm:	0.11121519325210905,	3.101342105287835e-59
Rayleigh:	0.2518291956862485,	5.895542415415657e-302
Expon:	0.41292537789661893,	0.0
Uniform:	0.3864460767335865,	0.0

Table 3.9 K-S Goodness of fit test statistic for fitting different distributions on the data.

3.2 Time Series Analysis of GHI data.

We now perform the time series analysis of the GHI data. We build models by getting optimal parameters using the data of the Rajasthan 1 location. We test the model forecasts on all five regions in Rajasthan. We build AR, MA, ARMA and ARIMA models on daily, weekly and monthly data, while SARIMA models only on the monthly data (due to hardware limitations). We do not forecast for the hourly data because there is very high variability in the hourly data. Additionally, around twelve hours in a day have near zero GHI values. Forecasting for hourly data may give information about technical details required to set up a solar park, but forecasting for daily, weekly and monthly data helps us choose a location for solar parks.

We use the auto_arima function of python and apply it on the dataset of Rajasthan1 to get the optimal values of parameters p, d and q and (P,D,Q). Thereafter, we test the model over all the 5 datasets of Rajasthan. We use measures such as MAPE and R-squared to validate our results of the forecast model.

To get an idea of our data, we plot the GHI data at various granular levels. We also decompose the data into trend, seasonality, and residual errors. This gives us two observations:

1. There is one outlier in the daily data whose average GHI value in the day is zero. (See figure 3.12).
2. There is no observed trend but we do observe seasonal behaviour in the daily data and the weekly data. This gives us some idea that SARIMA might perform better than ARIMA. Also, we get an idea that the data will be trend stationary and thus, the value of 'd' in our ARIMA model can be 0. The seasonal amplitude variation is also constant in nature, so through intuition we can understand that the model is additive in nature. (even though there is no trend)

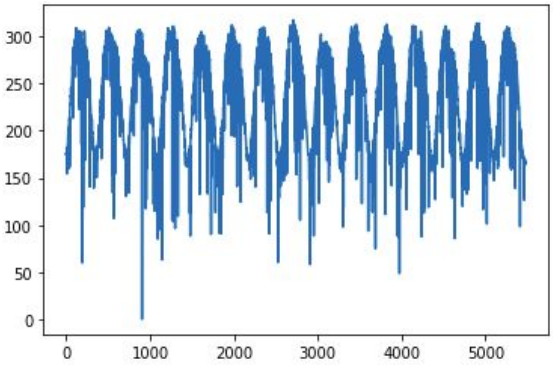

Figure 3.13 Daily GHI plot

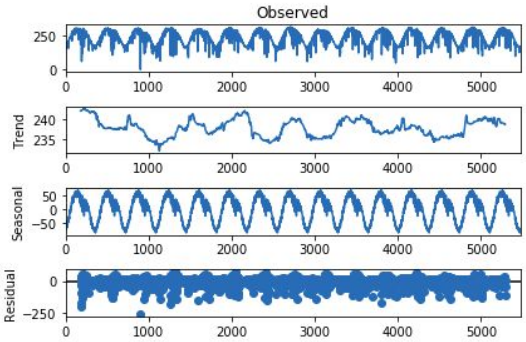

Figure 3.14 Seasonal Decomposition of Daily GHI plot

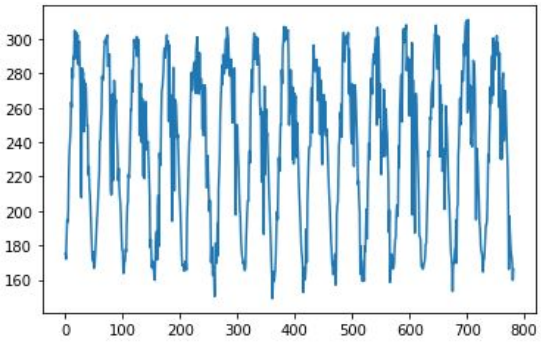

Figure 3.15 Weekly GHI plot

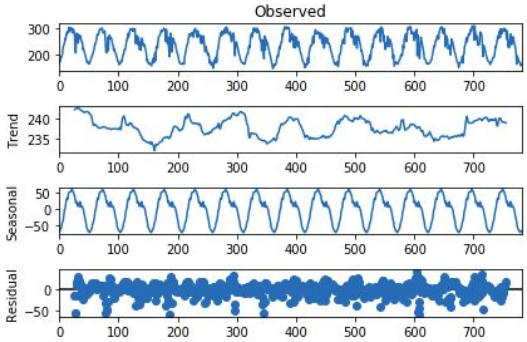

Figure 3.16 Seasonal decomposition of weekly GHI plot.

3.2.1 Stationarity

For applying various time series models we need to ensure that the data points fulfil the condition of stationarity. Stationarity ensures that the process generating the time series does not change overtime. Otherwise, it would be like doing time series analysis on a random data set which would yield no meaning from the underlying pattern. It also ensures that the sample of consecutive data with the same size yields identical covariance irrespective of the starting point. Without stationarity in the data, the time series forecast may behave as a random walk.

For some time series to be classified as stationary (covariance stationarity, weakly stationary), it must satisfy 3 conditions:

- Constant mean
- Constant variance
- Constant covariance between periods of identical distance

We perform the Augmented Dickey Fuller test to find the trend stationarity in our data. It is the most popular test for stationarity. Rejection of the Null hypothesis would enable us to determine the value of 'd' in ARIMA to be 0. Otherwise, we would have to perform some methods like differencing, detrending, transformations etc to make the data stationary.

3.2.1.1 Augmented Dickey Fuller Test.

AR, MA, ARMA, ARIMA and SARIMA models require the time series to be stationary which ensures that the process generating time series does not change overtime. Augmented Dickey Fuller test is a statistical test that enables us to determine whether a unit root is present in a time series sample or not.

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t,$$

where

α is the constant term, β is the coefficient of the time trend and p is the lag order of the AR process.

The unit root test is then carried out under the null hypothesis $\gamma = 0$ against the alternative hypothesis of $\gamma < 0$. Test statistic:

$$DF_t = \frac{\hat{\gamma}}{SE(\hat{\gamma})}$$

H0: A unit root is present.

Ha: No unit root is present and the time series is stationary.

As the value of the ADF test statistic becomes more negative, it is more likely we have trend stationary data.

Results of ADF test:

ADF Statistic: -4.046909
p-value: 0.001184

Critical Values:
1%: -3.471
5%: -2.879
10%: -2.576

Since the test statistic value is less than the critical value at 1% level of significance we reject the null hypothesis and thus, determine that the data is stationary.

3.2.2 ACF and PACF plots

3.2.2.1 Auto Regression

Auto regression gives us the dependent relationship between an observation and some number of lagged observations. We want to account for the pattern of growth or decline in the data. We obtain the order of the Autoregressive terms from the Partial Autocorrelation function plot.

3.2.2.1.1 Partial Autocorrelation Function(PACF)

Partial Autocorrelation Function is a summary of relation between one observation of the time series with the observation prior to it without any contribution from any contribution intervening observations.

For example, the observation today may depend on the observation yesterday and the observation yesterday may in-turn depend on the observation yesterday. The partial autocorrelation between today and yesterday is the correlation after removing the influence of the day before yesterday.

Given a time series z_t, the partial autocorrelation of lag *k*, denoted α (k), is the autocorrelation between z_t and z_{t+k} with the linear dependence of z_t on z_{t+1} through z_{t+k-1} removed;

$$\alpha(1) = \operatorname{corr}(z_{t+1}, z_t), \text{ for } k = 1,$$

$$\alpha(k) = \operatorname{corr}(z_{t+k} - P_{t,k}(z_{t+k}),\ z_t - P_{t,k}(z_t)), \text{ for } k \geq 2,$$

where $P_{t,k}$ (x) is the surjective operator of orthogonal projection of x onto the linear subspace of Hilbert space spanned by $z_{t+1}, \ldots, z_{t+k-1}$.

Using PACF, can help us determine the amount of lag we need to consider, i.e we can infer the prior observation which has non-zero contribution coefficients for the current observation.For each partial correlation we need to test where it is zero or not. We can do so by approximating the lower and upper limits of the partial correlation values from the corresponding critical values. At a 5% significance level the limits are given by $\pm 1.96/\sqrt{n}$ where n is the number of observations being analysed. This approximation can only be used if n>30.

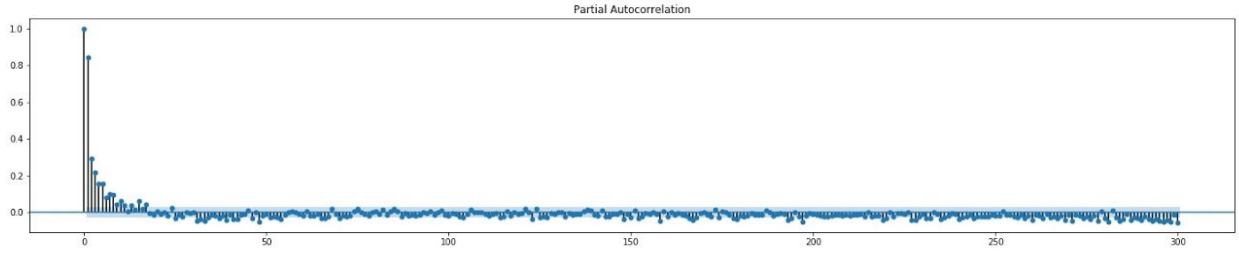

Figure 3.17 Partial Autocorrelation Function Coefficient plot

At 5% significance level, the obtained limits are ± 0.026. We observe from the PACF plot that the order of the AR term is 11. Due to computation constraints we limit the maximum order of the AR term to 5.

3.2.2.1.2 AR Results (Daily Average)

The following predictions results have been obtained from **AR(5)** model on daily averaged data of the last year(2014) for all the 5 regions of Rajasthan:

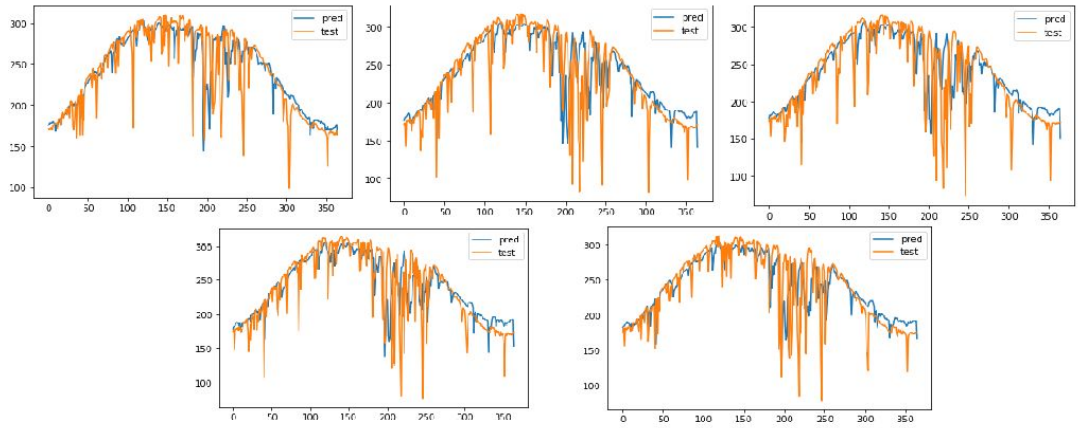

Figure 3.18 AR Daily Forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.679	0.536	0.545	0.583	0.576
MAPE	7.547	11.69	11.589	10.577	10.369

Table 3.10 AR (Daily) Accuracy Metrics.

3.2.2.1.3 AR Results (Weekly Average)

The following predictions results have been obtained from the AR**(5)** model on weekly averaged data of the last year(2014) from all the 5 regions of Rajasthan:

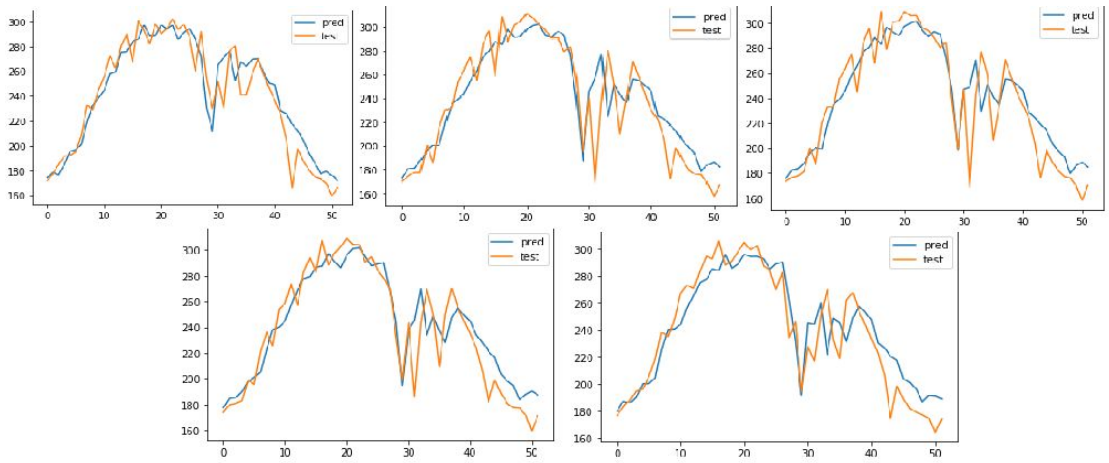

Figure 3.19 AR Weekly Forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.881	0.804	0.823	0.860	0.845
MAPE	5.153	6.924	6.598	5.993	6.116

Table 3.11 AR (Weekly) Accuracy Metrics

3.2.2.1.4 AR Results (Monthly Average)

The following predictions results have been obtained from the AR**(5)** model on monthly averaged data of the last year(2014) from all the 5 regions of Rajasthan:

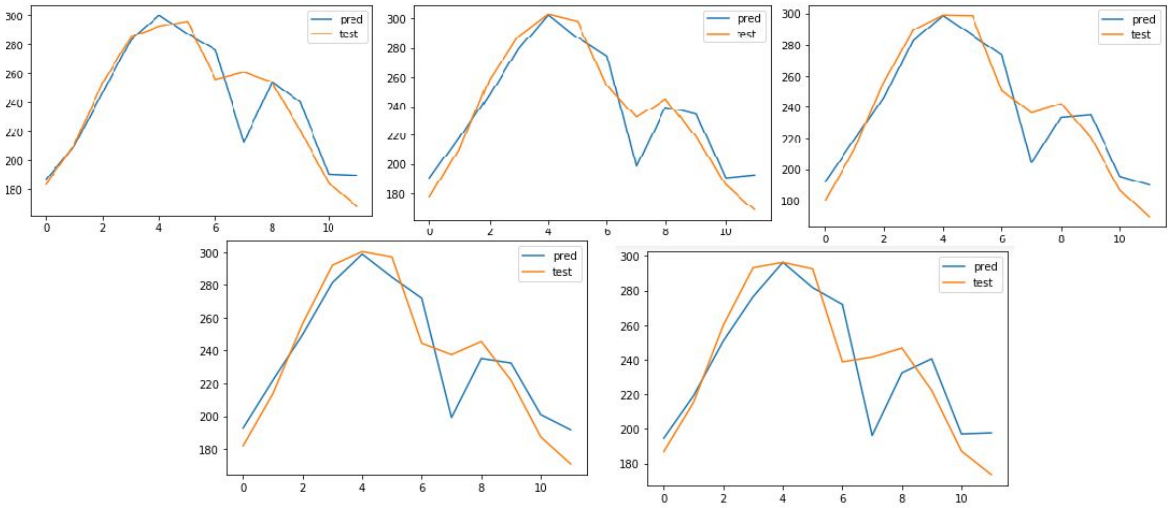

Figure 3.20 AR monthly forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.872	0.820	0.863	0.837	0.755
MAPE	5.2	5.91	5.82	6.365	7.066

Table 3.12 AR (Monthly) accuracy metrics.

3.2.2.2 MA

Moving average gives us the dependency between an observation and a residual error from a moving average model applied to lagged observations. We want to account for the noise (error terms) between consecutive time points. We obtain the order of the moving average terms from the Autocorrelation function plot.

3.2.2.2.1 Autocorrelation

Autocorrelation is the correlation of an observation with a lagged version of itself. In other words, it is the correlation between an observation and the past values. It is different from partial autocorrelation due to the fact that it does not remove the intervening effects of contributions from other observations.

It is defined as $\rho(s,t) = \gamma(s,t) / \sqrt{(\gamma(s,s)*\gamma(t,t))}$, it ranges from $-1 \leq \rho(s,t) \leq 1$ which can be shown using Cauchy-Schwarz inequality $|\gamma(s,t)|^2 \leq \gamma(s,s)*\gamma(t,t)$. Autocorrelation function gives the linear predictability of the time stamp (X_t) using (X_s).

Autocovariance γX (s,t) of stationary time series depends on s and t only through $|s - t|$, thus we can rewrite notation s = t + h, where h represents the time shift.

$$\gamma X\,(t + h, t) = cov(Xt + h,\ Xt) = cov(Xh,\ X0) = \gamma(h,\ 0) = \gamma(h)$$

Where, ACF of a stationary time series $\gamma(h) = cov(Xt + h,\ Xt) = E[(Xt + h - \mu)(Xt - \mu)]$

Autocorrelation Function of Stationary Time Series:

$$\rho(h) = \gamma(t + h,t)/ \sqrt{(\gamma(t + h,t + h)*\gamma(t,t))} = \gamma(h)/ \gamma(0)$$

ACF is used to evaluate Moving Average to determine the order of the MA model.

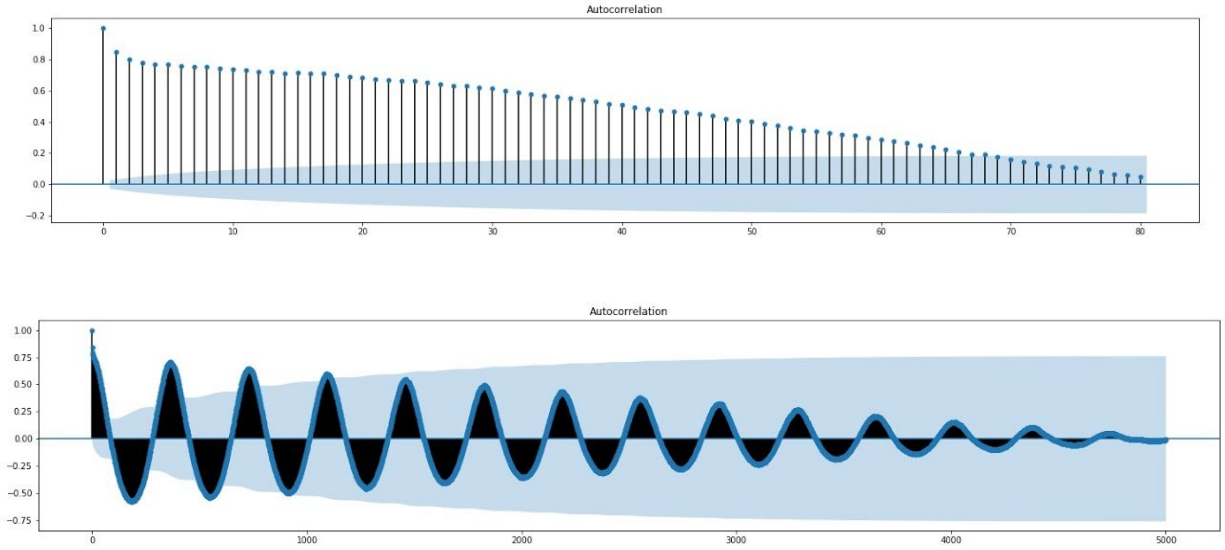

Figure 3.21 Autocorrelation coefficients plot at different granularities.

We observe from the ACF plot that the order of the MA term comes out to be very high. In fact the auto_arima library fails to calculate the MA value when max(p) and max(d) are set to zero because of computational limitations. Therefore, we limit the maximum order of the MA term to 5.

3.2.2.2.2 MA Results (Daily Average)

The following prediction results have been obtained from **MA(5)** model on daily averaged data of the last year(2014) from all the 5 regions of Rajasthan:

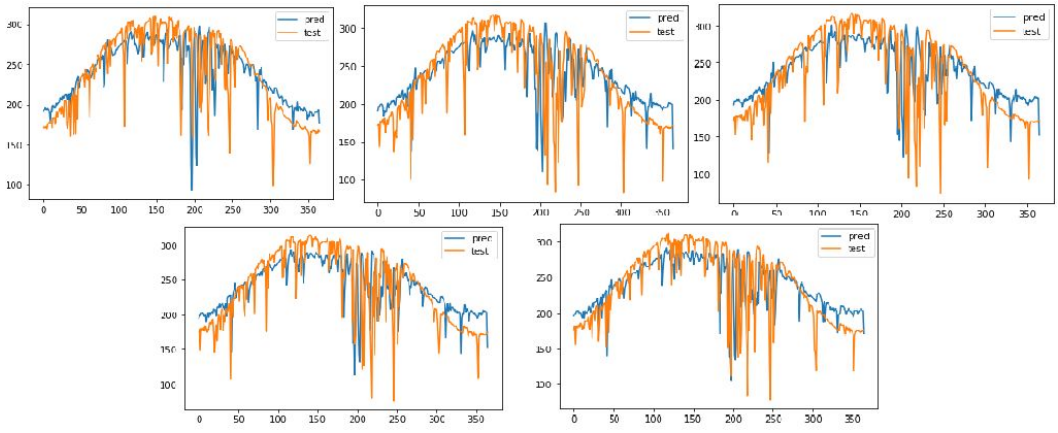

Figure 3.22 MA (Daily) Forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.526	0.422	0.425	0.459	0.443
MAPE	10.697	14.699	14.585	13.552	13.27

Table 3.13 MA (Daily) Accuracy metrics

3.2.2.2.3 MA Results (Weekly Average)

The following prediction results have been obtained from **MA(5)** model on weekly averaged data of the last year(2014) from all the 5 regions of Rajasthan:

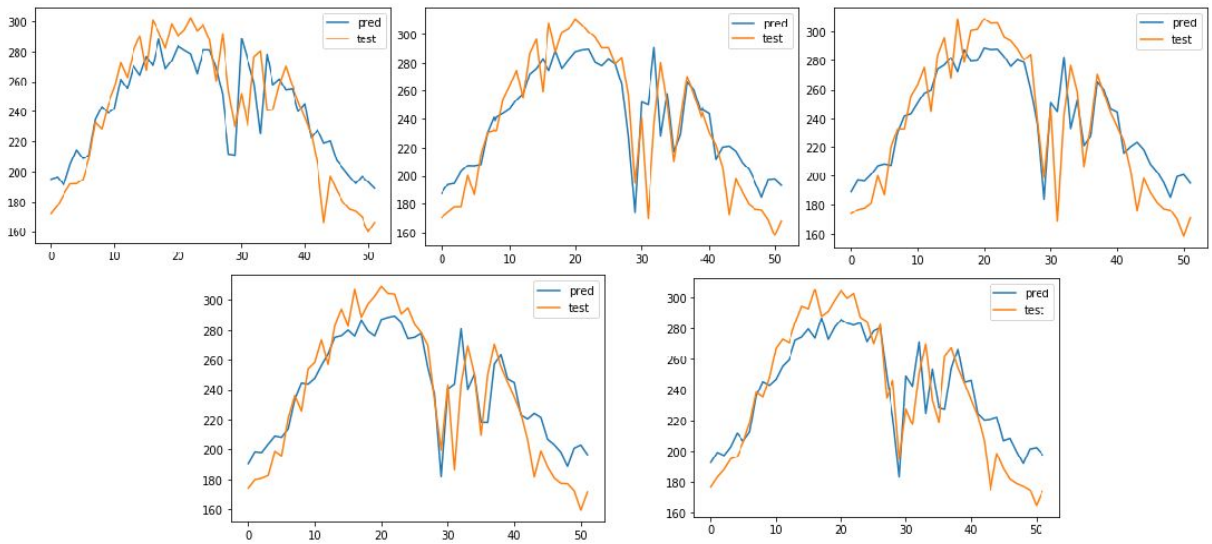

Figure 3.23 MA weekly forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.863	0.736	0.779	0.804	0.801
MAPE	5.58	8.848	8.193	7.527	7.331

Table 3.14 MA (weekly) accuracy metrics

3.2.2.2.4 MA Results (Monthly Average)

The following prediction results have been obtained from MA(5) model on monthly averaged data of the last year(2014) from all the 5 regions of Rajasthan:

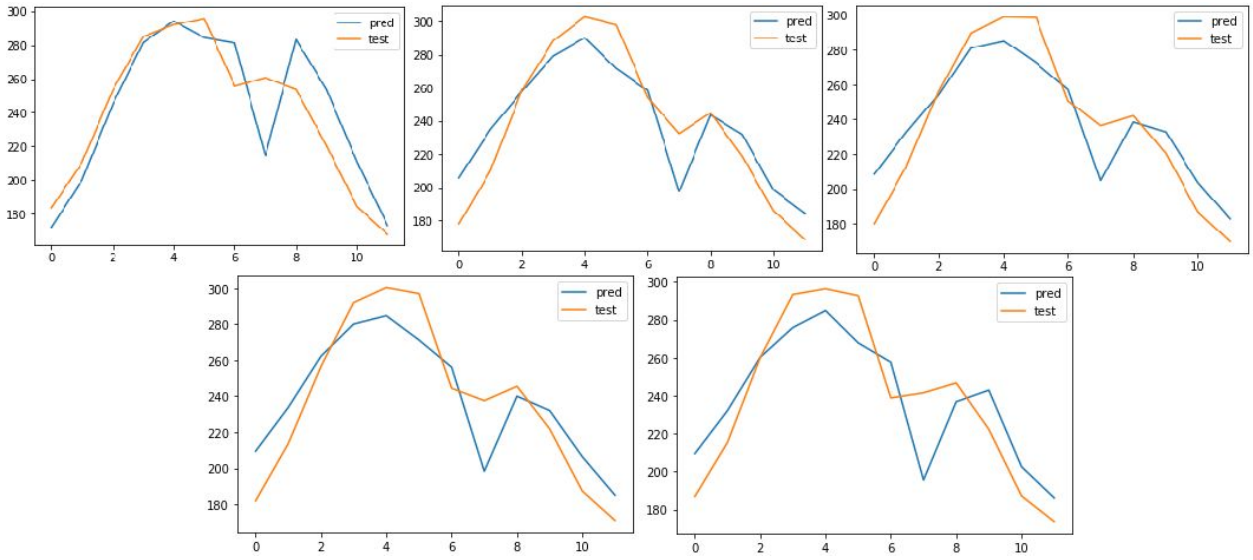

Figure 3.24 MA Monthly forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.872	0.723	0.827	0.788	0.740
MAPE	6.84	7.699	6.936	7.597	7.827

Table 3.15 MA (Monthly) Accuracy Metrics

3.2.3 ARMA

ARMA stands for Autoregressive Moving Average model,and is primarily used to describe weakly stationary stochastic time series in autoregression and moving average.

The parameters of this model are p and q: here p is the order of the AR polynomial and q is the order of the MA polynomial. The AR part in ARMA involves regressing an observation on the past observation. The MA part involves modeling the error term as a linear combination of error terms occurring in various past predictions.

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}.$$

The PACF and ACF plots in the previous sections help us identify the appropriate values of p and q for the ARMA model.

3.2.4 ARIMA

The difference between ARMA and ARIMA comes from the integration term. Integrated refers to the number of times needed to difference a series in order to achieve stationarity. The results of the Augmented Dickey Fuller test show that there is no need for differencing in the series. The following two points give an intuitive relationship between ARMA and ARIMA:

1. ARMA(p,q) is equivalent to ARIMA(p,0,q)

2. Given an ARIMA(p,d,q), if d>0, one can represent this by ARMA(p,q) after differencing the original series d times.

3.2.4.0.1 ARMA and ARIMA Results (Daily Average)

For ARIMA the optimal d came out to be 0 because the data is trend stationary. Thus, the results of ARIMA and ARMA are the same. The following prediction results have been obtained from the ARIMA(2,0,1) model on daily averaged data of the last year(2014) from all the 5 regions of Rajasthan:

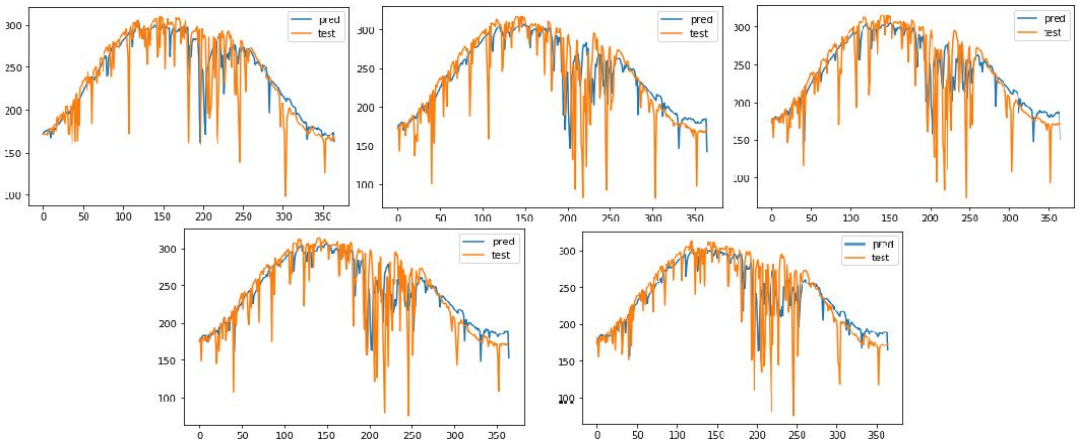

Figure 3.25 ARMA and ARIMA Daily forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.703	0.582	0.5816	0.616	0.602
MAPE	7.345	11.069	11.155	10.153	10.161

Table 3.16 ARMA and ARIMA (Daily) Accuracy Metrics

3.2.4.0.2 ARMA and ARIMA Results (Weekly Average)

The following results have been obtained from ARIMA(2,0,2) model on weekly averaged data of the last year(2014) from all the 5 regions of Rajasthan:

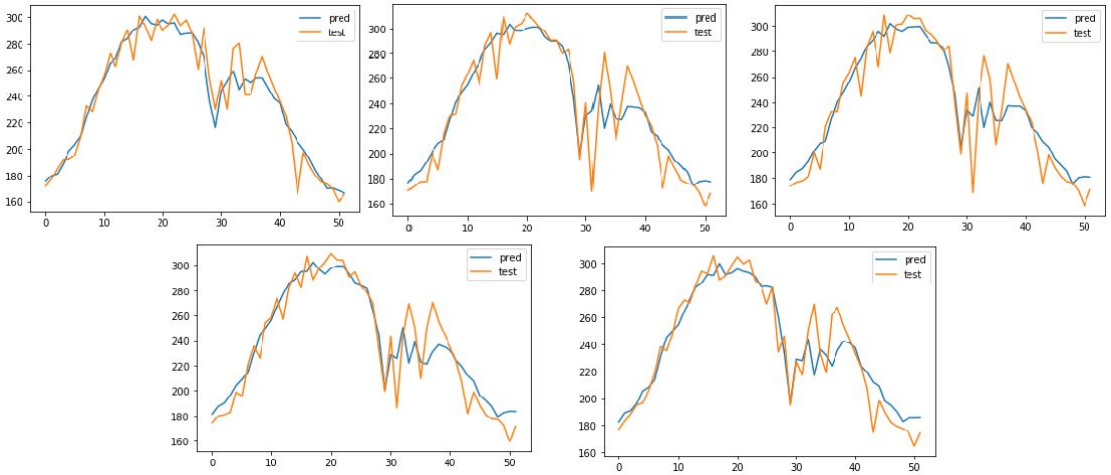

Figure 3.26 ARMA and ARIMA Weekly forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.928	0.863	0.861	0.889	0.890
MAPE	3.699	5.584	5.673	5.081	4.372

Table 3.17 ARMA and ARIMA weekly accuracy metrics

3.2.4.0.3 ARIMA and ARMA Results (Monthly Average)

The following results have been obtained from ARIMA(2,0,1) model on monthly averaged data of the last year(2014) from all the 5 regions of Rajasthan:

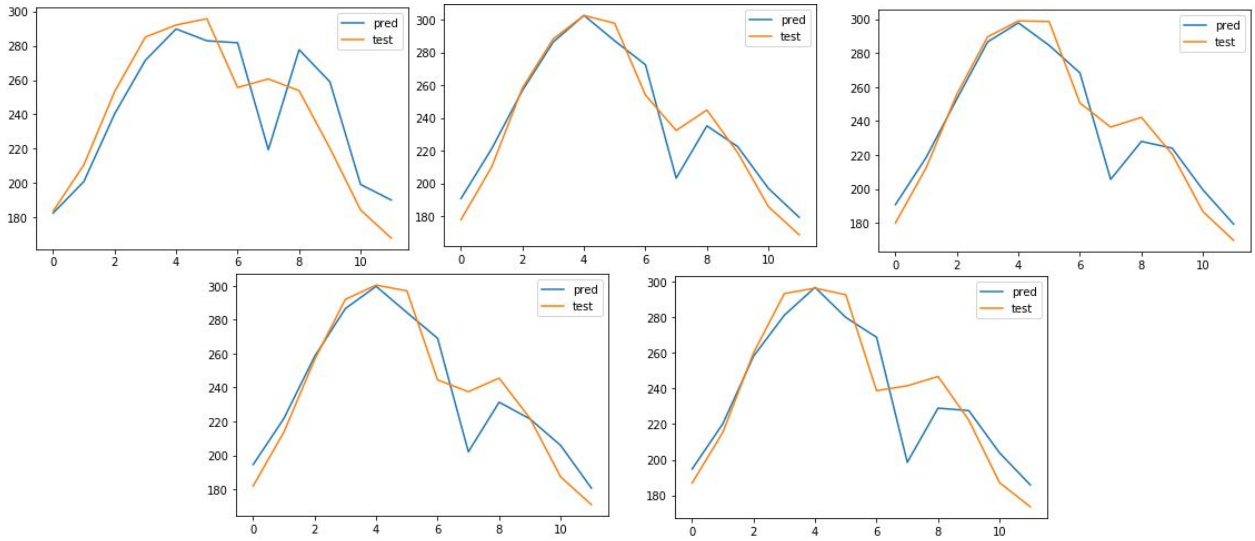

Table 3.27 ARMA and ARIMA monthly forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.900	0.787	0.865	0.801	0.698
MAPE	4.68	6.735	5.358	5.98	8.696

Table 3.18 ARMA and ARIMA (monthly) accuracy metrics

3.2.5 SARIMA

The main drawback of ARIMA is that it does not support seasonal data. To capture the seasonality in the data, SARIMA adds three more hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series. It also adds one more parameter for the time period of the seasonality.

3.2.5.0.1 SARIMA Results (Monthly Average)

The following prediction results have been obtained from SARIMA(3, 0, 1)X(2, 1, 1, 12) model on monthly averaged data of the last year(2014) from all the 5 regions of Rajasthan:

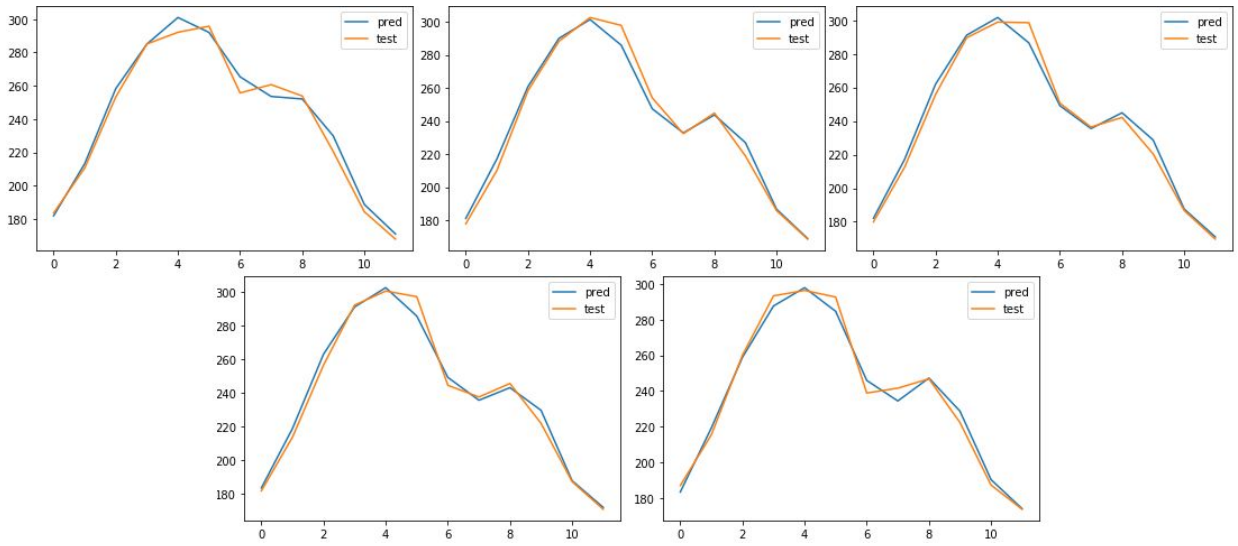

Figure 3.28 SARIMA monthly forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.986	0.981	0.986	0.986	0.985
MAPE	1.56	2.01	1.53	1.57	1.68

Table 3.19 SARIMA (monthly) accuracy metrics

3.2.5.0.2 Implementation Details for AR, MA, ARMA, ARIMA, SARIMAX:

1. We use the SARIMAX algorithm provided by the statsmodels library in the data. As the data is univariate, we use no exogenous variables in the SARIMAX algorithm, thus it effectively becomes a SARIMA algorithm. We use a grid search to find the best model by obtaining the model with the least AIC value.
2. The model is defined by seven parameters: p: Trend autoregression order, d: Trend difference order, q: Trend moving average order. P: Seasonal autoregressive order. D: Seasonal difference order. Q: Seasonal moving average order. m: The number of time steps for a single seasonal period. We set the values of the parameters to get results for AR, MA, ARMA, ARIMA, SARIMA.
3. The method we use is the conditional sum of squares' and 'maximum likelihood estimation'. It means that the estimated mean of the distribution is based on a normal distribution with its peak

at the highest probability point of the observed values. MLE's role in the algorithm is to determine the values for the parameters of the model with a high degree of probability that the model's results will be close to the observed (given) data.

4. Outputs: We get the following outputs from the SARIMAX:
    a. Log-Likelihood: The log-likelihood value is a simpler representation of the maximum likelihood estimation. This value on its own is quite meaningless, but it can be helpful if you compare multiple models to each other. A better model has higher log-likelihood values.
    b. AIC, BIC, HQIC: AIC stands for Akaike's Information Criterion. It takes in the results of the MLE as well as the total number of your parameters. Since adding more parameters to your model will always increase the value of the maximum likelihood, the AIC balances this by penalizing for the number of parameters. BIC (Bayesian Information Criterion) induces more penalization for more number of parameters than AIC. The formulas for the parameters are given below:
    Let n = number of observations (e.g. data values, frequencies)
    k = number of parameters to be estimated (e.g. the Normal distribution has 2: mu and sigma)
    Lmax = the maximized value of the log-Likelihood for the estimated model (i.e. fit the parameters by MLE and record the natural log of the Likelihood.)

SIC (Schwarz information criterion, aka Bayesian information criterion BIC)

$$SIC = \ln[n]k - 2\ln[L_{max}]$$

AIC (Akaike information criterion)

$$AIC_c = \left(\frac{2n}{n-k-1}\right)k - 2\ln[L_{max}]$$

HQIC (Hannan-Quinn information criterion)

$$HQIC = 2\ln[\ln[n]]k - 2\ln[L_{max}]$$

    c. Other results we obtain are the estimated coefficients of our model:
        i. ar.L1 gives the AR term with a lag of one.
        ii. ma.L1 and ma.L2 give the MA terms with lag of one and two.
        iii. The 'std err' columns is an estimate of the error of the predicted value. It tells you how strong is the effect of the residual error on your estimated parameters (the first column).
        iv. The 'z' is equal to the values of 'coef' divided by 'std err'. It is thus the standardised coefficient.
        v. The P>|z| column is the p-value of the coefficient. It is really important to check these p-values before you continue using the model. If any of these values are higher than your given threshold (usually 0.05), you might be using an unreliable coefficient that might cause misleading results.

3.3 Machine Learning Model (LSTM)

We use an LSTM based machine learning model with three LSTM layers. We use LSTM as opposed to other deep networks as LSTM solves the vanishing gradient problem which exists in other Recurrent Neural Networks. We also tried a Bidirectional LSTM model, but the results were very poor and training the model took a long time. The summary of the LSTM model can be seen below:

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_13 (LSTM)               (None, 12, 256)           264192
_________________________________________________________________
lstm_14 (LSTM)               (None, 12, 128)           197120
_________________________________________________________________
lstm_15 (LSTM)               (None, 64)                49408
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 65
=================================================================
Total params: 510,785
Trainable params: 510,785
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.29 LSTM Model Summary

3.3.0.0.1 LSTM results (Monthly Average)

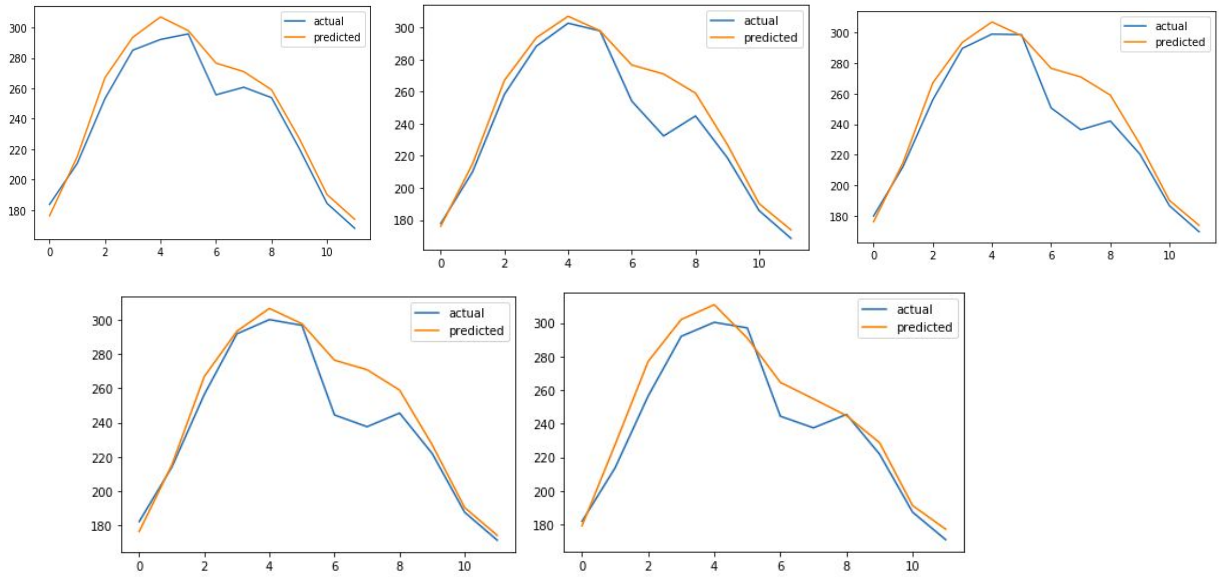

Figure 3.30 LSTM monthly forecasts

Dataset	Raj_1	Raj_2	Raj_3	Raj_4	Raj_5
R-squared	0.942	0.894	0.889	0.882	0.872
MAPE	3.678	4.20	4.259	4.08	4.13

Table 3.20 LSTM Results

4 Conclusions and future research directions

4.1 Observations from the forecast results:

1. The MAPE and the R-squared values show that the model tests best on the solar park corresponding to Rajasthan 1 but shows comparatively less accuracy in other solar parks of Rajasthan.
2. The best performing model is SARIMA, followed by LSTM and then ARMA (ARIMA). This validates the fact that SARIMA performs better for seasonal data.
3. We observe that the AR and MA models built on the weekly data give better results on Rajasthan 1 data while the same models built on monthly data show better generalization on the monthly data.
4. The ARMA/ARIMA model performs better for weekly data than monthly data.

4.2 Conclusions:

1. Using KS test, we realize that none of the distributions are a good fit on the data.
2. Augmented Dickey Fuller test shows that the data is trend stationary.
3. For monthly data, SARIMA is the best performing model.
4. Due to system/hardware limitations we could not apply SARIMA on daily and weekly data and so (out of ARIMA, AR, and MA) ARIMA is the best performing model for daily and weekly data.
5. We get an insight about the granularity (daily, weekly, monthly) of the data to be used for forecasting purposes. Different models show better results on different granularities.

4.3 Future research direction:

1. We have not considered the effect of other variables in our data on the GHI time series forecasting. We can use SARIMAX with different exogenous variables and compare prediction quality.
2. In recent literature, grid based spatial ARIMA models have been proposed. Similar models have been used for COVID-19 epidemic prediction. These models can be used to account for the spatial aspect in the data.
3. LSTM models can be improved by hyperparameter tuning and by using LSTM-CNN networks. We can also smoothen out the data (by using moving average) for better forecasts using deep learning techniques.

5 Bibliography

1. Atique, S., et al. “forecasting of total daily solar energy generation using ARIMA: A case study.” *9th Annual Computing and Communication Workshop and Conference*, vol. 1, no. 1, 2019, pp. 114-119. *IEEE*.
2. Machine Learning Mastery. “SARIMA Models.” *Time series forecasting*, 17 August 2018, https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/. Accessed 15 November 2020.
3. Medium. “Distribution fitting with Python.” *Medium*, 2020, https://medium.com/@amirarsalan.rajabi/distribution-fitting-with-python-scipy-bb70a42c0aed. Accessed 15 November 2020.
4. Ministry of Renewable Energy. “Renewable Energy Sources.” *Ministry of New and Renewable Energy (MNRE)*, 1992, http://www.mnre.gov.in/. Accessed 25 November 2020.
5. Pasari, Sumanta, and Venkata S. Nandigama. “Statistical Modeling of Solar Energy.” *Enhancing Future Skills and Entrepreneurship*, Springer, Scham, 2020, pp. 157-165. *Springer*. Accessed November 2020.
6. Pasari, Sumanta, and Aditya Shah. “Time Series Auto-Regressive Integrated Moving Average Model for Renewable Energy Forecasting.” *Enhancing Future Skills and Entrepreneurship*, Springer, Cham, 2020, pp. 71-77. *Springer*.
7. Pongo, R., et al. “The Grid-Based Spatial ARIMA Model: An Innovation for Short-Term Predictions of Ocean Current Patterns with Big HF Radar Data.” *Advances in Intelligent Systems and Computing*, 936 ed., IEEE Xplore, 2019, pp. 26-36. *IEEE Xplore*.
8. Roy, S., et al. “Spatial prediction of COVID-19 epidemic using ARIMA techniques in India.” *Model. Earth Syst. Environ.*, IEEE Xplore, 2020, pp. 0-10. *IEEE Xplore*, https://doi.org/10.1007/s40808-020-00890-y.
9. Solar Irradiance- Wikipedia. “Solar Irradiance.” *Wikipedia*, 22 August 2019, https://en.wikipedia.org/wiki/Solar_irradiance#:~:text=The%20solar%20constant%20is%20a,element%20perpendicular%20to%20the%20Sun. Accessed 25 November 2020.
10. StatsModels. “SARIMAX Algorithm.” *statsmodels.tsa.statespace.sarimax.SARIMAX*, 2020, https://statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html. Accessed 19 November 2020.
11. University of Pittsburgh. “Time Series: Autoregressive models AR, MA, ARMA, ARIMA.” *ARIMA Models*, 23 October 2018, http://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class16.pdf. Accessed 21 November 2020.
12. Wikipedia. “Dickey Fuller Test.” *Dickey Fuller Test*, 2013, https://en.wikipedia.org/wiki/Augmented_Dickey%E2%80%93Fuller_test. Accessed 13 November 2020.