Statistical Analysis and Forecasting of Solar Energy (Inter-states)

Group 6

November 27, 2020

Introduction

Three-fifth of India's power sector relies entirely on coal [6]. Concerns such as depleting indigenous reserves of natural resources and global warming, have raised an alarm for alternative energy sources. However, a lot of these alternate sources have a tendency of being economically infeasible or geographically or seasonally unavailable.

In this study we focus only on solar energy – which is an ideal renewable energy because of it is abundant, cost-effective, and does not produce pollution. However it comes with its own limitations. solar energy is highly dependent on the amount of sunlight available which in-turn depends on factors such as climate, season, and weather. Our study is restricted to solar parks located in four different cities within the Indian subcontinent, so that highly limits the climate variations. Since our data is available as a time series of the amount of solar energy, the seasonal availability of solar energy throughout the year is reflected as the literal seasonality component of our time series. Additionally, on cloudy and rainy days, we may get insufficient sunlight. In this case, we may want to rely on alternative energy sources. This dependence on weather calls for prediction of the availability of solar energy on different scales – meteorological and demographic. The aim of our study is to do a time series analysis of the availability of solar energy for four states – Andhra Pradesh, Rajasthan, Madhya Pradesh, and Tamil Nadu, and hence prediction. The complete code for the analysis is available here.

Definition: A time series is a sequence of periodically recorded observations of a variable [2].

Throughout our discussion, we measure the quality of predictions with the MAPE metric (Mean Absolute Percentage Error) which is defined as follows:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{F_t - X_t}{X_t}\right|$$

where n is the number of observations, F_t is the forecast at time t, and X_t is the actual observation at time t.

Some key attributes for solar energy forecasting are [7]:

1. **Diffused Horizontal Irradiance (DHI)** which measures the light reaching earth's surface after being scattered by atmosphere.
2. **Direct Normal Irradiance (DNI)** is the sunlight received on surface which is perpendicular to direction of light.
3. **Global Horizontal Irradiance (GHI)** is the light received by the surface which is horizontal to earth's surface.

Also, note the following equation [1]

$$GHI = DHI + DNI \times cos(Z) \tag{1}$$

From the above relation, we can see GHI takes into account both DNI and DHI, and is therefore chosen to be the prime attribute for forecasting.

We've employ several statistical time series prediction techniques such as AR, MA, ARMA, ARIMA, and SARIMA for the predicting the GHI value for the four states. A detailed analysis of the same is available in Discussion section.

Dataset

The given dataset contains hourly data for four solar parks located in Rajasthan, Madhya Pradesh, Andhra Pradesh and Tamil Nadu for the years 2000 through 2014. In every entry, it has the following attributes:

1. Timestamp of measurement
2. DHI and Clearsky DHI
3. DNI and Clearsky DNI
4. GHI and Clearsky GHI
5. Dew Point
6. Temperature
7. Pressure
8. Relative Humidity
9. Solar Zenith Angle
10. Snow Depth
11. Wind Speed

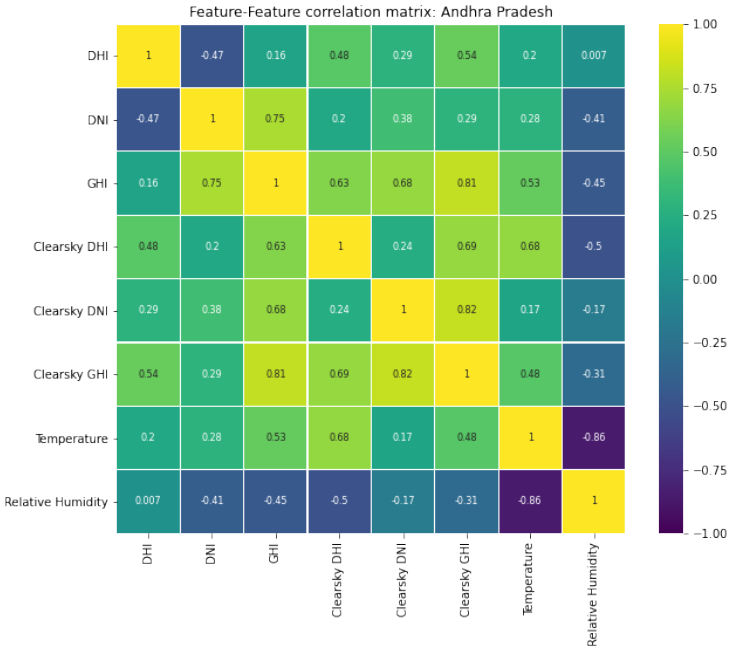
Discussion

Correlation

We obtained the feature-feature correlation maps for different states. For simplicity, we show only one of the states, the other states show similar results which are available in the Appendix.

Consider the heatmap shown in Figure 1. The correlation between DNI and GHI is 0.75, Clearsky GHI and GHI is 0.81, and Clearsky GHI and Clearsky DNI is 0.82. GHI, DNI and DHI are highly correlated. This is expected from the relation between these attributes (See Eq. 1).

Figure 1



Distribution

We first check the GHI data for normality.

H_0: Distribution is normal
H_a: Distribution is not normal

We conduct a 95% significance D'Agostino K^2 Test for all the four states. The K^2 statistic takes care of both skewness and kurtosis for a distribution [4]. For simplicity, we've shown the GHI histogram only for Andhra Pradesh in Figure 2. We got p value ¡ 0.05 and rejected the null hypothesis concluding our data is not normal.

Further, we check other well known distributions to determine a best fit. The candidate distributions considered are: Gamma, Beta, Rayleigh, Logistic, Weibull, Lognormal, Chi-Squared, and

Exponential. The top four best fit obtained are given in the Appendix with AIC values.

Figure 2: Distribution Fits for Andhra Pradesh GHI Data

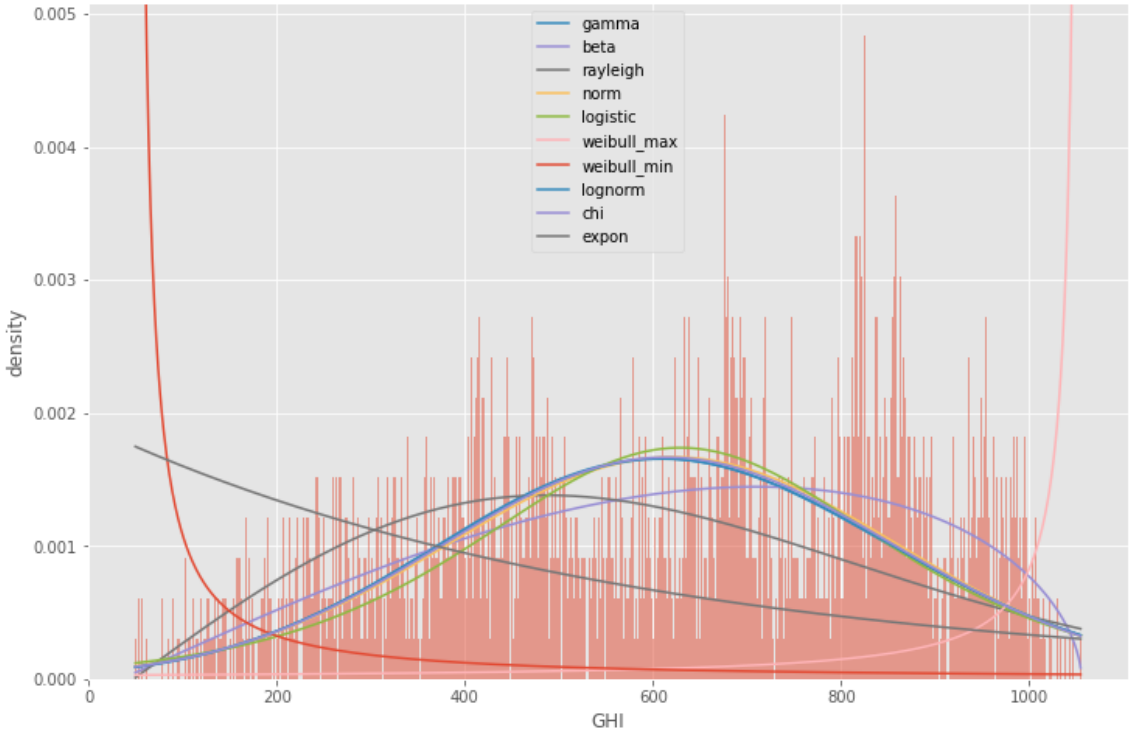
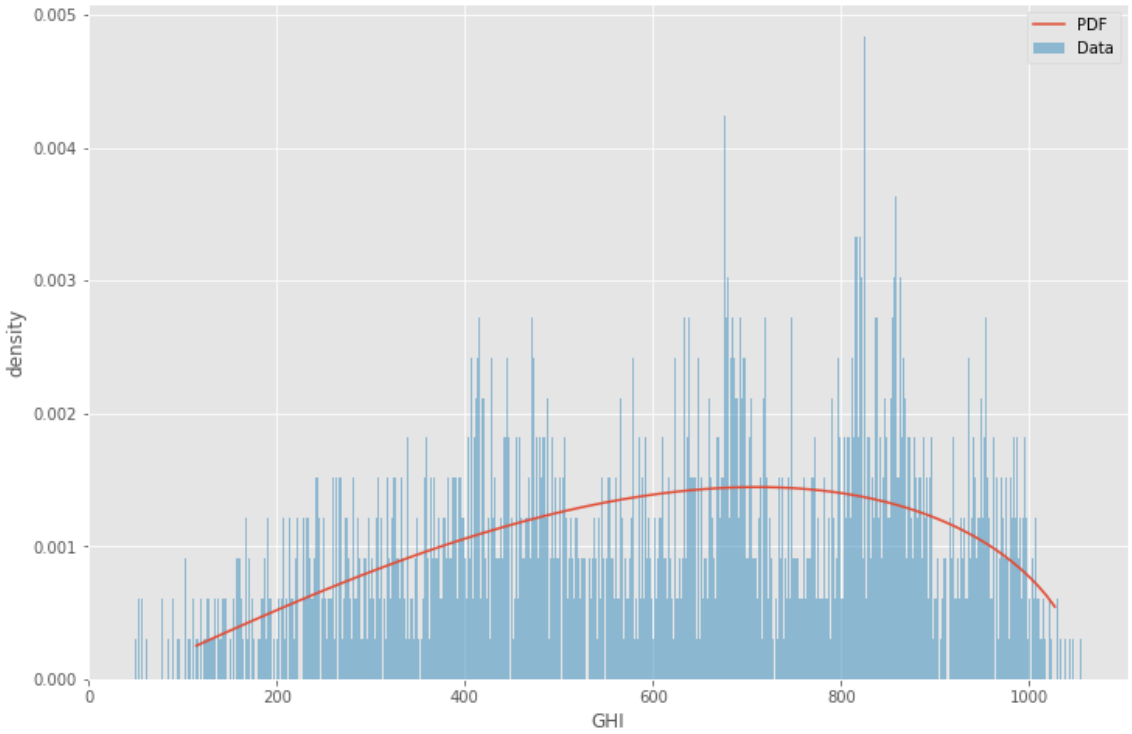
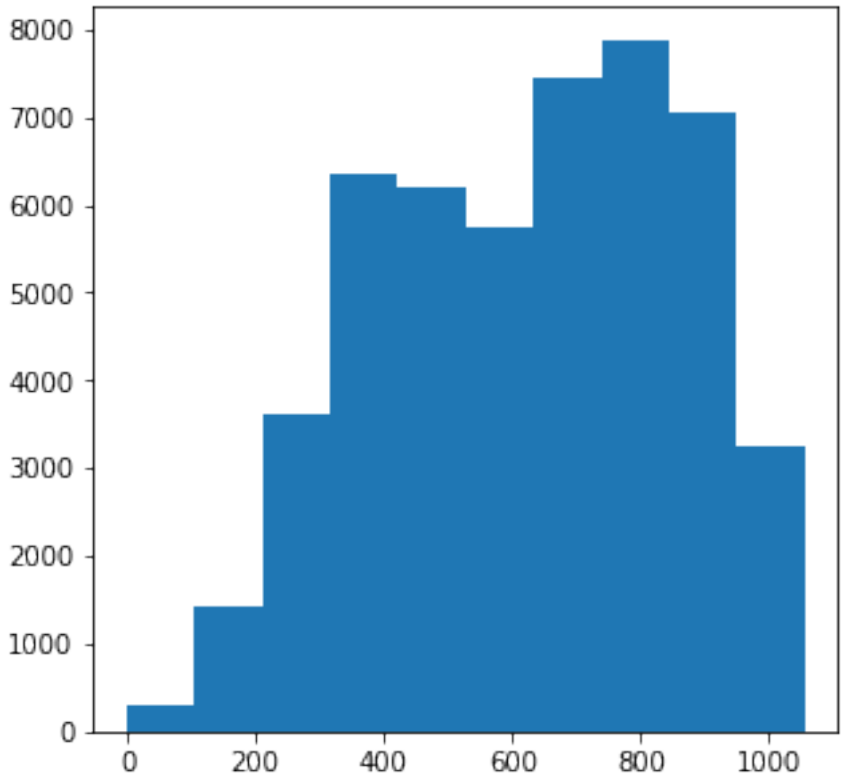


Figure 3: Best Fit (Beta Distribution) for Andhra Pradesh GHI Data



The best fit on the data for year 2000, for each of the states, was obtained on beta distribution with varying parameters. We verified the same with the Akaike Information Criterion (AIC) values. We also observe that data is skewed in the plots, which is validated by the values of α, β parameters (for example $\alpha = 2.0978$, $\beta = 1.5585$ for Andhra Pradesh) obtained for the beta distributions where $\alpha > \beta$.

Figure 4: GHI Histogram for Andhra Pradesh



Stationarity Tests

1. **ADF Test**: To check the stationarity we performed the Augmented Dickey-Fuller test. The hypotheses are as follows:

   H_0: The series has a unit root
   H_a: The series has no unit root

   It was observed that the value of the test statistic was greater (more negative) than the 1% critical values for each of the selected series of GHI data for different states. Thus, we could reject the null hypothesis of a unit root being present in the data set and we can conclude with more than 99% confidence that this is a stationary series.

2. **KPSS Test**: It is always better to apply both the tests (ADF and KPSS), so that it can be ensured that the series is stationary or if it needs any kind of differencing to make it stationary. To validate the results of stationarity we performed KPSS test.

   H_0: The series is stationary
   H_a: The series has a unit root (it is not stationary)

   We got a p-value (0.1) larger than the alpha and can not reject the null hypothesis in this test. Thus we conclude our null hypothesis is true and the series is stationary.

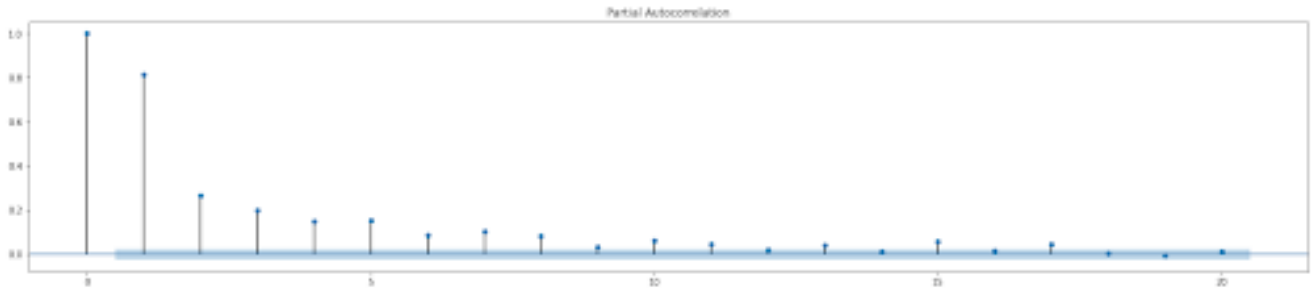
   Our results correspond to Case 2 from statsmodels [5] documentation* i.e. both the tests (ADF and KPSS) agree that the series is stationary and we can conclude the same.

Time Series Analysis

1. **Remove Hourly Variation**: We converted the hourly time series values to a daily time series by summing up all the values over a day. This removes the variation of GHI values over the span of a day, which are not relevant to our analysis and forecasting. It also reduces the size of our time series, making computation relatively easier.

2. **Partial Autocorrelation Function (PACF)**: Partial Autocorrelation is a summary of the relationship between an observation in a time series with observations at prior time steps with the relationship of intervening observations removed. PACF removes the indirect correlation between current step and the lag. This gives us a good metric for the value of p for our Autoregression (AR) model.

Figure 5: PACF Plot

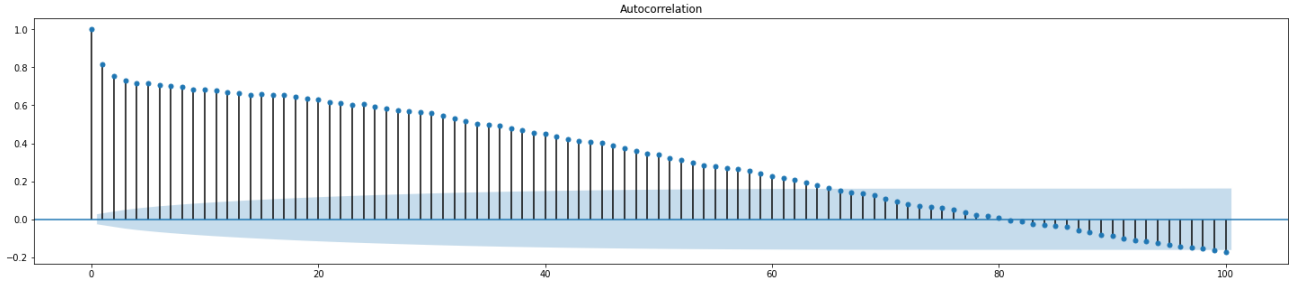


   Simply, the lags which have PACF above the threshold would be useful for AR models. In our case, from a plot of PACF for the first 20 lags, a possible value of p (for AR) is 17.

*Documentation Reference: https://www.statsmodels.org/dev/examples/notebooks/generated/stationarity_detrending_adf_kpss.html

3. **Autocorrelation Function(ACF)**: Similar to PACF, the values of ACF plot that are above the shaded region (Confidence Interval) indicate that the corresponding lag and all the lags before it are correlated to the current time series observation. Thus, it acts as a decent indicator for estimating the number of lags in Moving Average (MA).

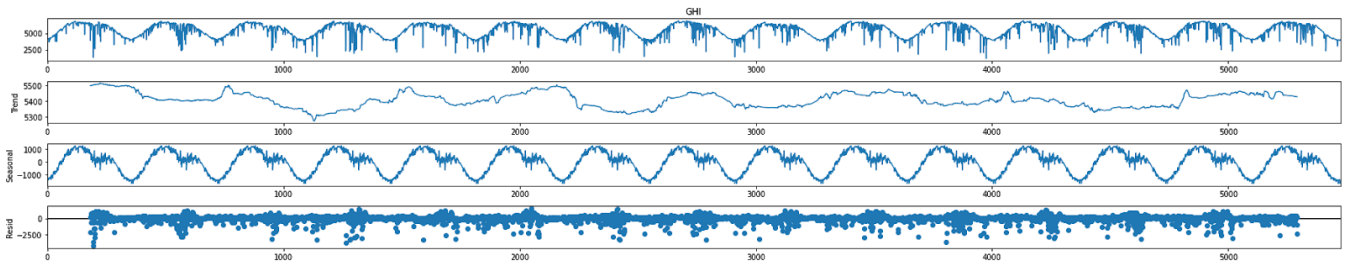
Figure 6: ACF Plot



4. **Decomposition**: Rajasthan dataset was used for the time series analysis to begin with. A plot of the time series clearly shows a seasonality component, literally too, as they vary according to real seasons apparently. The time series was decomposed into Trend, Seasonality and Residual components.

   One particular outlier (June 23, 2002) had all observed values of the day as zero. It was a null observation, that we proceeded to replace with the average between the observations on preceding and succeeding days.

Figure 7: Time Series Decomposition – Rajasthan (Daily)



5. **Prediction**:

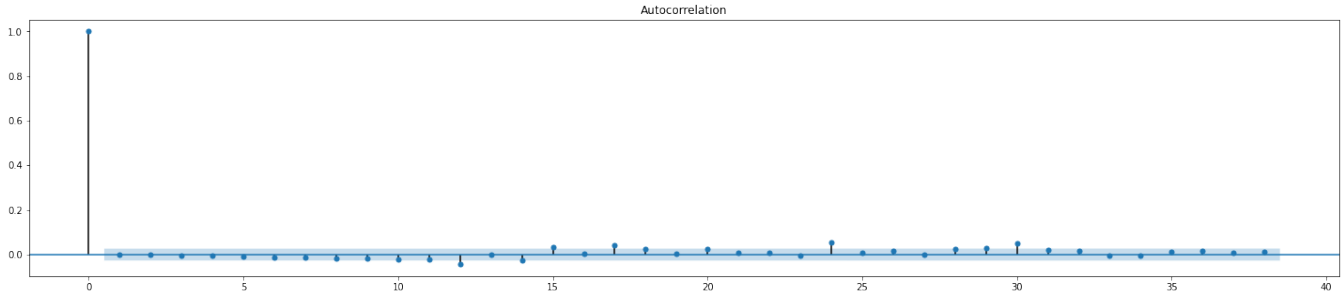
   - **AR** AutoRegressive (AR) model is a multiple regression model where we forecast the variable of interest using a linear combination of past values of the same variable in the time series. An AR model of order p, referred to as AR (p), can be written as [3]:

$$\theta_p(B)x_t = (1 - \alpha_1 B - \alpha_2 B^2 - .. - \alpha_p B^p)x_t = w_t$$

   where B is the backshift operator and w_t is white noise.

Although the PACF plot recommends that the value of p should be around 17, on running a grid search for the best fit to AR, we got a value of p equal to 12. We ran an AR model with p as 12 over the entire time series and got an MAPE of 6.83%. A small p-value for a coefficient on running the model means that the particular lag corresponding to that coefficient might be significant in predicting the current value. We got p-values for each coefficient out of the 12 to be less than 0.05, except for the 9^{th} coefficient; this makes sense considering that the 9^{th} lag in the PACF plot turned out to be insignificant. On running the above stated model, we get the following ACF plot for the residuals.

Figure 8: ACF Plot – AR (8)

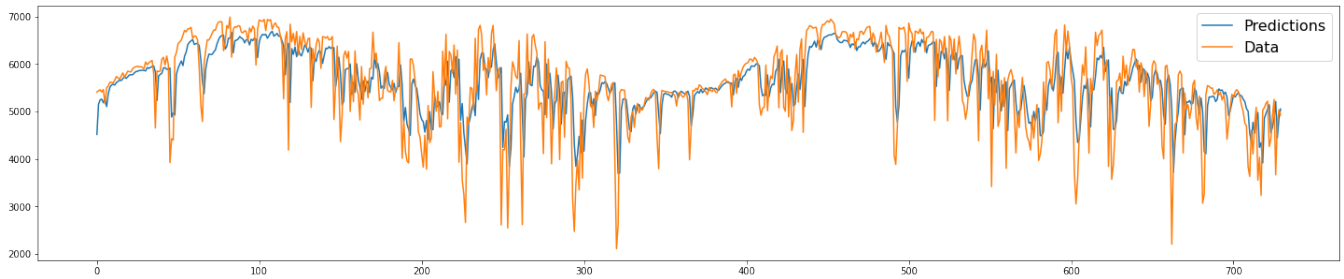


Note that an ideal ACF plot for residuals should have all the points within the blue region, indicating that the correlations are statistically zero (the residuals can be called white noise). Clearly there are a few significant points in our plot, showing that AR might not be the best model for this time series. But for the sake of completeness, we have gone ahead with an AR analysis anyway.

As in AR and every other method from here, our analysis consists of a rolling forecast. In a rolling forecast, we have a train and test split of the data. In our case, it was 13 years for the train set and the last two years for the test set. We first fit the model on the training set and then predict the value for the next day. Then the actual value of GHI for the next day is added to the training set from the test set and the process is repeated again. This is done until we have exhausted all the points in the test set. This is a computationally intensive task, which takes around 2 hours for even small values of p.

Because of this problem, we decided to go with a p value of 8 based on the PACF plot for our rolling forecast. It gave us an MAPE of 9.45% on the entire test set. Similarly to the process described above, weekly and monthly rolling forecasts were done, and we got MAPEs of 11.58% and 13.23% respectively (see the plots in the Appendix for reference). As expected, the MAPE values increased as we increased the span of each forecast.

Figure 9: Daily Predictions – AR (8)



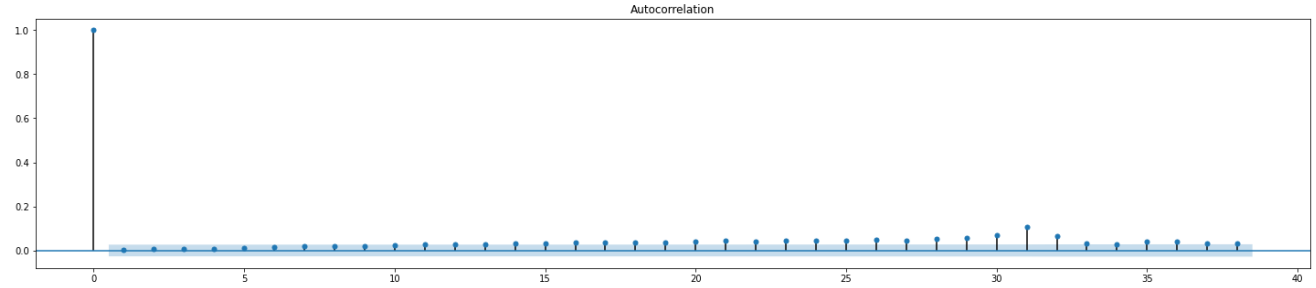
- **MA** As opposed to AR which uses past values of forecast variable in time series for regression analysis, the Moving Average(MA) model uses past forecast errors in a regression-like model. An MA(q) process can be expressed as follows [3]:

$$x_t = (1 + \beta_1 B + \beta_2 B^2 + ..{}_{\beta q} B^q) w_t = \phi_q(B) w_t$$

where ϕ_q is a polynomial of order q and B is the backshift operator. Here, each value of x_t can be thought of as a weighted moving average of the past q forecast errors.

The ACF plotted earlier suggests that the value of q should be around 60 for MA. However, this is computationally expensive and leads to very large training times. So instead, we use the value of q as 30. Running the MA model over the entire time series with this q yields a Mean Absolute Percentage Error (MAPE) value of 7.28%. Also, the following ACF plot for residuals is obtained.

Figure 10: Residuals ACF Plot – MA (30)



As expected, we can clearly see in the above plot that there are significant autocorrelations (outside blue region) for lags after 30. Now, although the value of q as 30 provides a good fitting model, it is almost impossible for us to train the model with such a high value of q. So instead, we use MA (6) for rolling forecast.

For daily forecasting, MA (6) fits decently and gives a MAPE value of 8.79%, marginally better than AR. However, for weekly and monthly predictions, MAPE values are significantly larger. MAPE for weekly forecast is 14.25% and for monthly forecast is 16.97%. Clearly, MA model doesn't perform very well at forecasting for this time series.

- **ARMA** AutoRegressive Moving Average or ARMA, as the name suggests, is a combination of the previous two models, AR and MA. It is used to describe a stationary time series in terms of two component polynomials, one of which corresponds to autoregression and the other to moving averages. The model has two parameters, p and q, for Auto-regression and Moving Averages respectively. ARMA (p,q) model is as follows [3]:

  $$\theta_p(B)x_t = \phi_q(B)w_t$$

  where θ_p is a polynomial of order p and ϕ_q is a polynomial of order q.

  Due to computational constraints, we did a grid search for the hyper-parameters p and q each in the range of 1 to 20 and got the best values as 12 and 10 respectively. The model turns out to be a decent fit for the time series, as the p-values for most of the 22 coefficients are less than 0.05; this argument is further strengthened by the ACF plot of the residuals (please see the plot in the appendix for reference). The MAPE for the same was 6.78%.

  As discussed before, owing to heavy computational costs, the grid search for the best parameters for the rolling forecast was done in a much smaller space. This resulted in the values of p and q as 3 and 1. The daily rolling forecast gave an MAPE of 7.06% while the weekly and monthly forecasts gave MAPEs of 8.63% and 10.23% respectively.

- **ARIMA** AutoRegressive Integrated Moving Average or ARIMA is a generalization of ARMA. Unlike ARMA, which is only applicable in stationary models, ARIMA can be applied on non-stationary models as well. It does this by using differencing to convert the non stationary model into a stationary one. Mathematically, d order differencing is of the form [3]:

  $$(1 - B)^d x_t$$

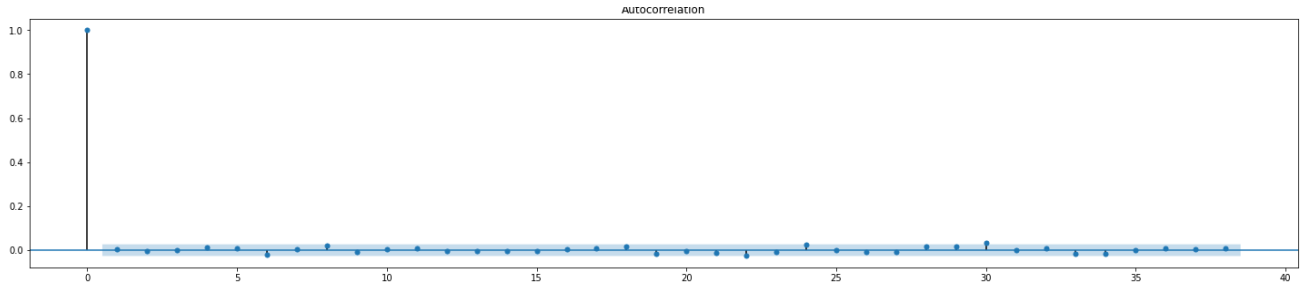
  where B is the backshift operator. A series is integrated of order d if differencing by order d results in white noise. ARIMA(p, d, q) performs ARMA(p, q) on data that has been integrated by order d, which is the meaning of the added I - for integration. This model can be succinctly expressed by the equation [3]:

  $$\theta_p(B)(1 - B)^d x_t = \phi_q(B)w_t$$

  where θ_p and ϕ_q are polynomials of order p and q respectively.

  Though the data is already confirmed to be stationary by our previous tests, we still performed ARIMA for testing purposes. We performed a grid search for parameters with $d \geq 1$ which best fit our model. We found that an order of $(12, 1, 10)$ fit our model best(MAPE 6.76%), and resulted in insignificant correlation in the residuals.

Figure 11: Residuals ACF Plot – ARIMA (12, 1, 10)



However, this model is too computationally intensive to use on a rolling forecast. Instead, we used an order of $(3, 1, 1)$ for our rolling forecast, giving us an MAPE of 6.97%, the best of all the forecasting methods employed so far. This model also performed exceedingly well on weekly and monthly predictions, giving an MAPE of 8.49% and 10% respectively.

Figure 12: Daily Predictions – ARIMA (12, 1, 10)

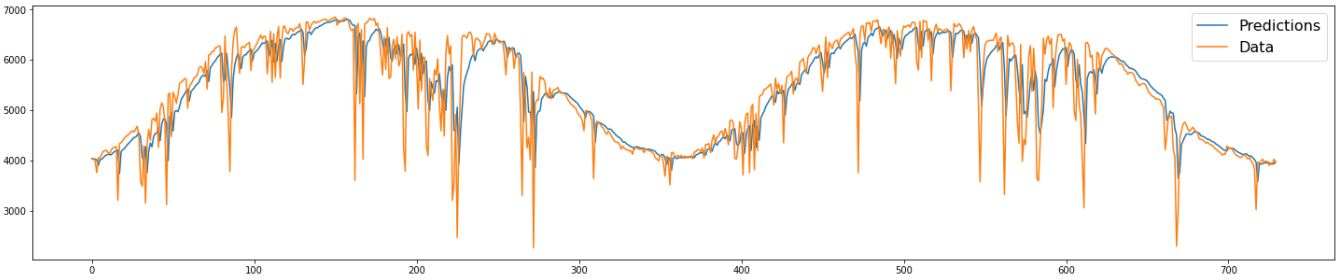


Figure 13: Weekly Predictions – ARIMA (12, 1, 10)

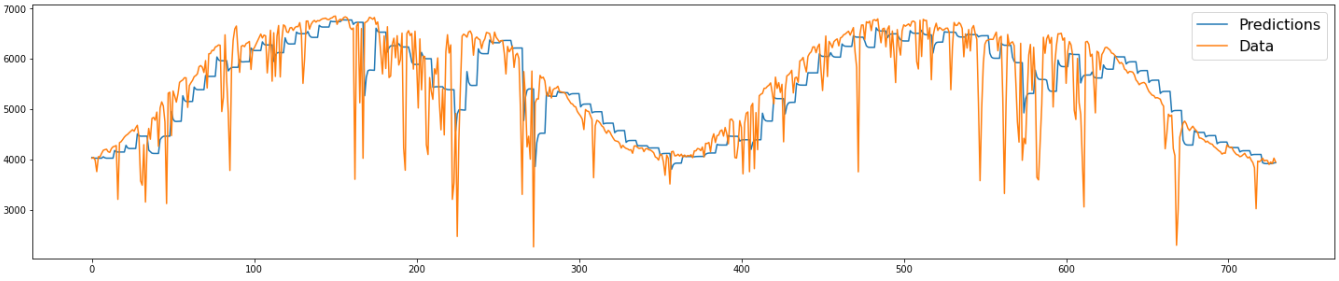
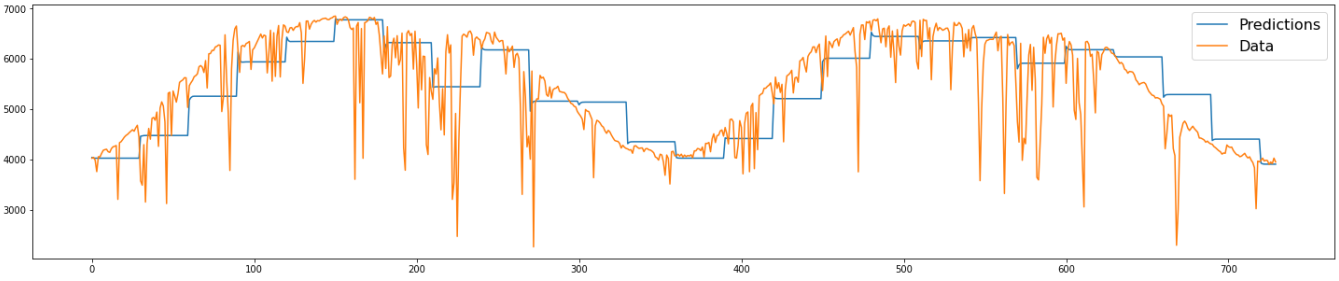


Figure 14: Monthly Predictions – ARIMA (12, 1, 10)



- **SARIMA** Seasonal ARIMA model uses differencing at a lag equal to the seasonality to remove additive seasonal effects. It also introduces autoregressive and moving average terms at the same lag, resulting in the order being expressed as $(p, d, q)(P, D, Q)_s$, where s is the seasonality. This can be expressed mathematically as [3]:

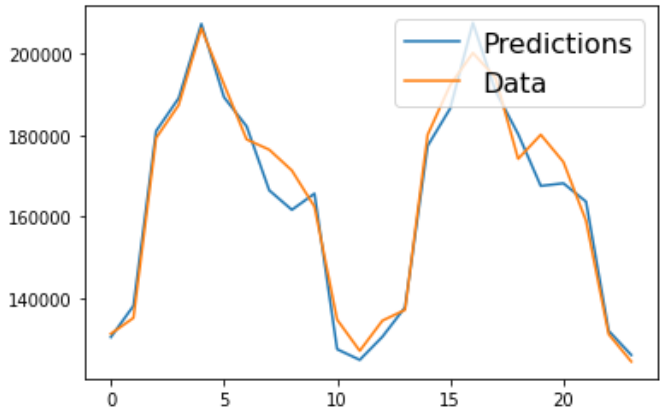
$$\Theta_P(B^s)\theta_p(B)(1-B^s)^D(1-B)^d x_t = \Phi_Q(B^s)\phi_q(B)w_t$$

where $\Theta_P, \theta_p, \Phi_Q, \phi_q$ are polynomials of orders P, p, Q and q respectively.

The seasonality in SARIMA appears as an exponent in this model and therefore a daily seasonality of 365 is highly computationally expensive. So instead, we aggregate the data into monthly data (by summing over the entire month), thus reducing the seasonality to 12.

Through a grid search we obtain the optimal parameters for our SARIMA model. It was found that a non-seasonal order of $(3, 0, 3)$ and seasonal order of $(2, 0, 1)$ worked best for our data. The results are shown in the graph below, which gives us a MAPE of only 2%.

Figure 15: SARIMA Predictions



- **Selection of Model and Analysis for Other States** From the MAPE values obtained for all the above models, the best model for Time Series Forecasting of our data could possibly be SARIMA. But because of the extremely high computation cost, we cannot afford daily time series prediction with SARIMA. Hence, we choose ARIMA as the most optimal model and our model of choice for Time Series forecasting for other states. Their results (MAPE values) of their fits are tabulated below:

State	MAPE
Rajasthan	6.97%
Andhra Pradesh	11.45%
Tamil Nadu	15.47%
Madhya Pradesh	10.14%

Table 1: MAPE Values of ARIMA Models of Different States

Conclusion

In our analysis above, we performed exploratory data analysis on the dataset, and tested multiple autoregression models for forecasting. From our testing, we found ARIMA to give the best results while remaining sufficiently easy to compute. We also tried our trained model on the other states in our dataset. This yielded less than satisfactory results, which is expected considering the seasonal and climatic variations between the states.

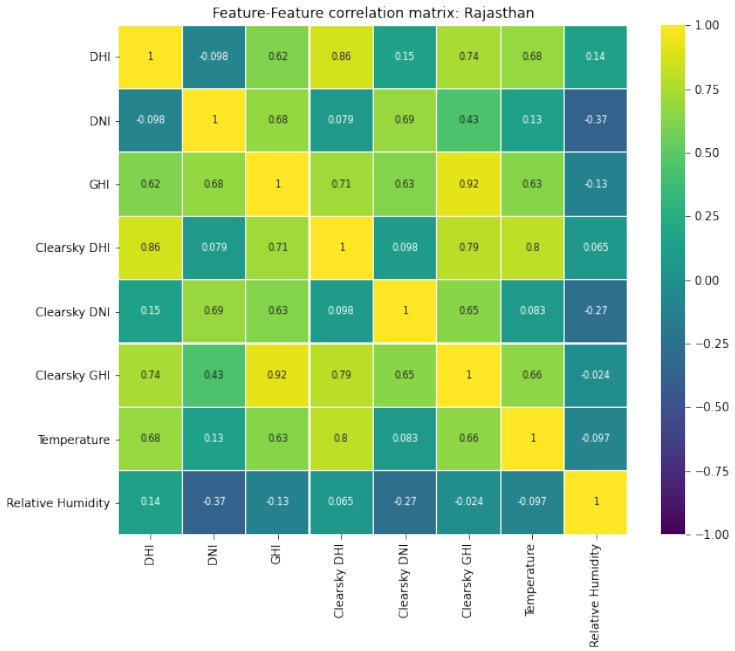
Considering the results, it is possible the SARIMA model would work better for our time series. However, we lack sufficient compute power to train a SARIMA model for daily forecasting, due to the large period. In the future, it is worth exploring this model and seeing if it can give us better results.

Appendix

Correlations

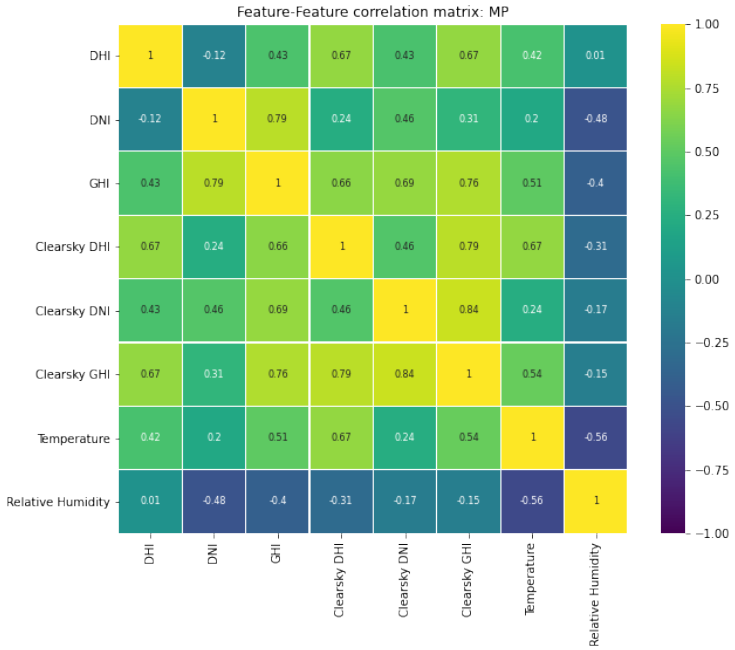
For Rajasthan, the heatmap shown in Figure 16 shows the correlation between *Clearsky DHI* and *DHI* to be 0.86, and *Clearsky GHI* and *GHI* to be 0.92, and *Clearsky GHI* and *Clearsky DHI* to be 0.79.

Figure 16



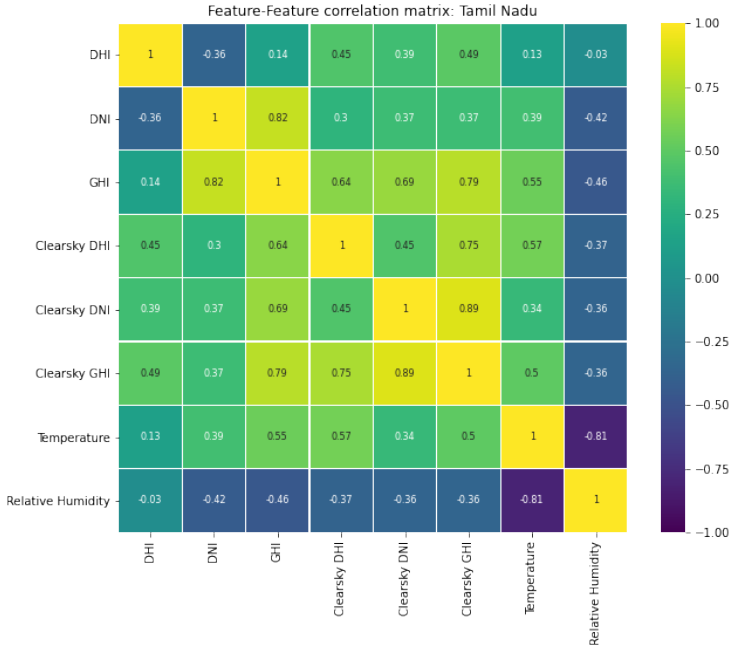
Similarly, For Madhya Pradesh, the heatmap in Figure 17 shows the correlation between *DNI* and *GHI* to be 0.79, and *Clearsky DHI* and *GHI* to be 0.79, and *Clearsky GHI* and *GHI* to be 0.84.

Figure 17



For Tamil Nadu, the highly correlated attributes are *DNI* and *GHI* at 0.82, and *Clearsky GHI* and *GHI* at 0.79, and *Clearsky GHI* and *DNI* at 0.89, as shown in Figure 18.

Figure 18



Best Fit Distributions

	sumsquare_error	aic	bic
beta	0.000013	1421.401782	-63613.186020
gamma	0.000022	1441.941676	-61812.543828
rayleigh	0.000024	1428.413071	-61545.327263
logistic	0.000024	1445.734975	-61494.044630

(a) Andhra Pradesh

	sumsquare_error	aic	bic
beta	0.000009	1410.276005	-64596.613640
gamma	0.000018	1435.407270	-62391.430261
logistic	0.000020	1438.694917	-62160.285448
rayleigh	0.000027	1415.108646	-61150.958232

(b) Rajasthan

	sumsquare_error	aic	bic
beta	0.000010	1400.246544	-64530.387497
gamma	0.000017	1426.013610	-62676.291714
logistic	0.000018	1432.577083	-62414.654118
rayleigh	0.000020	1419.626815	-62111.907019

(c) Madhya Pradesh

	sumsquare_error	aic	bic
beta	0.000016	1397.924221	-62743.828020
gamma	0.000028	1425.948371	-60980.545642
rayleigh	0.000030	1418.583996	-60812.825722
logistic	0.000031	1433.271306	-60725.208475

(d) Tamil Nadu

Figure 19: Best Four Distribution Fits

Forecasts

Figure 20: Weekly Predictions – AR (8)

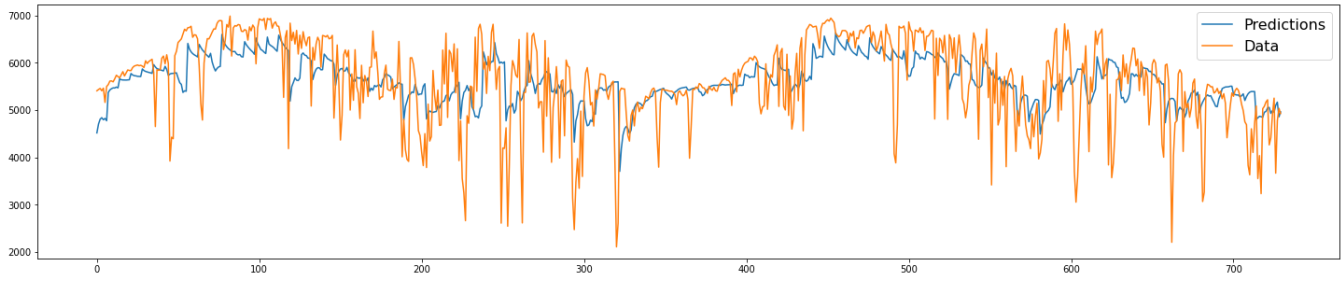


Figure 21: Monthly Predictions – AR (8)

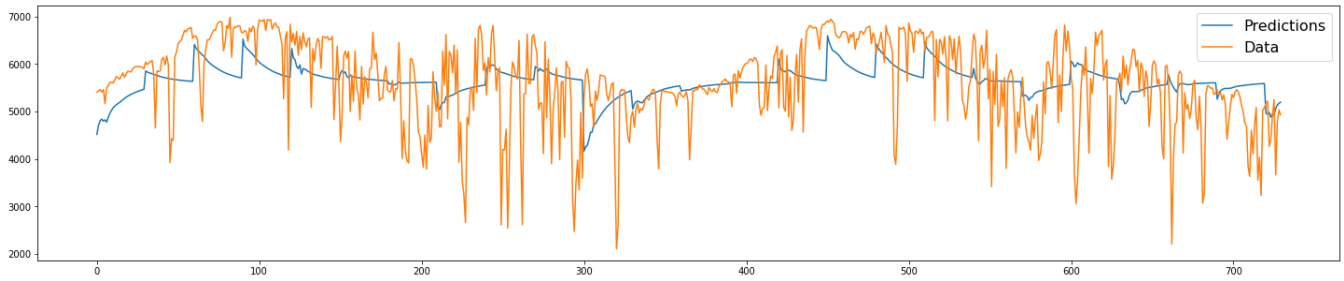


Figure 22: Daily Predictions – MA (6)

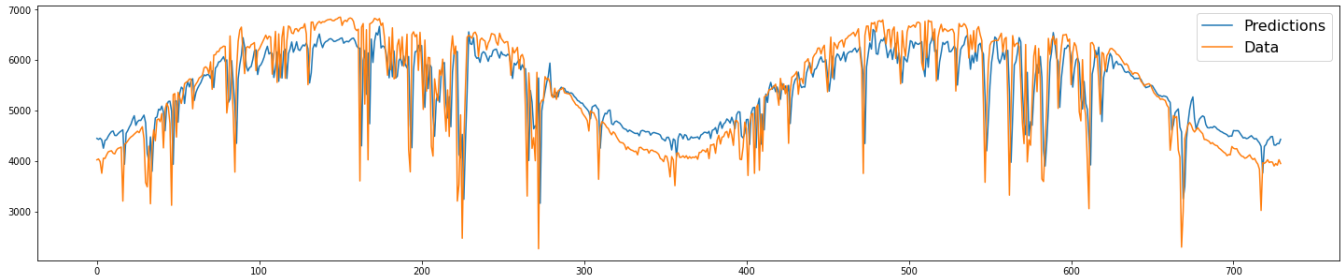


Figure 23: Weekly Predictions – MA (6)

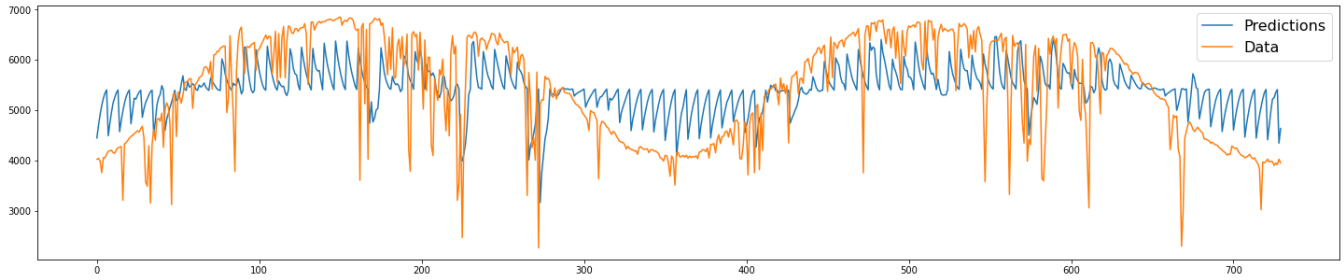


Figure 24: Monthly Predictions – MA (6)

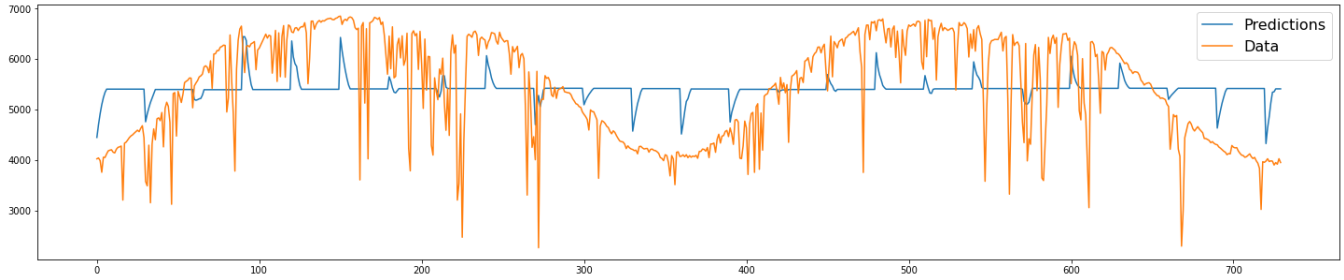


Figure 25: Daily Predictions – ARMA (12, 10)

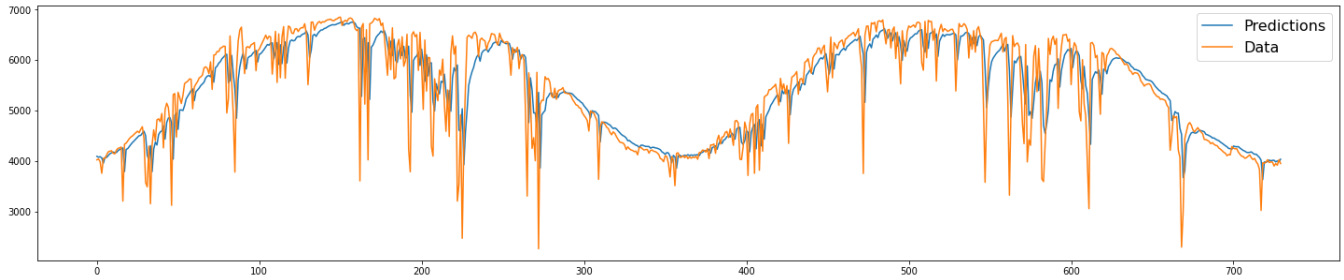


Figure 26: Weekly Predictions – ARMA (12, 10)

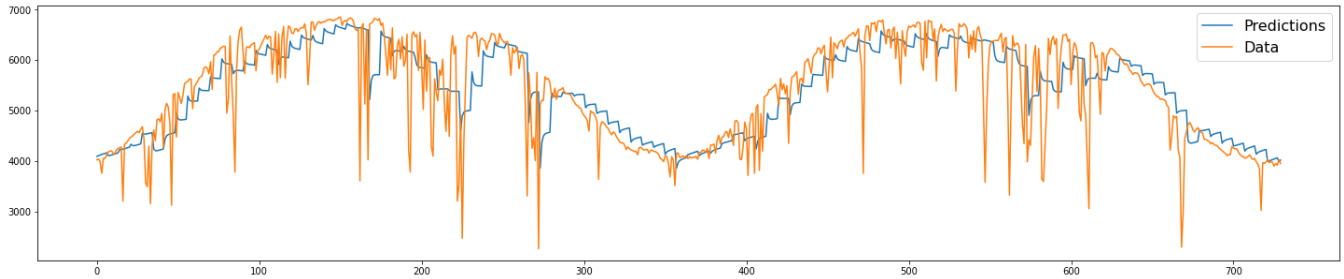
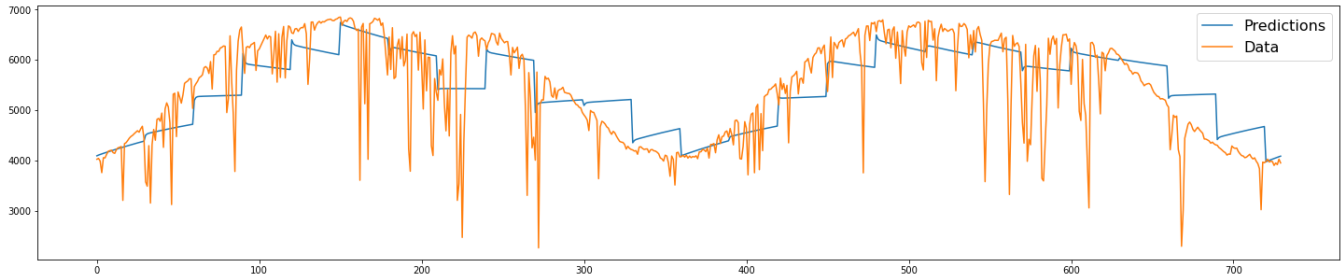


Figure 27: Monthly Predictions – ARMA (12, 10)



References

[1] National Renewable Energy Laboratory (NREL). *Solar Resource Glossary*. 2020. URL: https://www.nrel.gov/grid/solar-resource/solar-glossary.html (visited on 11/27/2020).

[2] D.R. Anderson et al. *Statistics for Business & Economics*. Cengage Learning, 2013.

[3] Paul Cowpertwait and Andrew Metcalfe. *Introductory Time Series with R*. Springer, 2009.

[4] Ralph B. D'Agostino and Albert Belanger. "A Suggestion for Using Powerful and Informative Tests of Normality". In: *The American Statistician* 44.4 (1990), pp. 316–321.

[5] Skipper Seabold and Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[6] A. Upadhyay and A. Chowdhury. "Solar Energy Fundamentals and Challenges in Indian Restructured Power Sector". In: *International Journay of Scientific and Research Publications (IJSRP)* 4 (10 2014).

[7] S. Vashishtha. *Differentiate between the DNI ,DHI and GHI*. 2012. URL: https://firstgreenconsulting.wordpress.com/2012/04/26/differentiate-between-the-dni-dhi-and-ghi/ (visited on 11/27/2012).