ARMA Time Series Modeling for Solar PV Generation Forecasting

Ninad Gaikwad and Karthikeya Devaprasad

***Abstract*—There has been a recent push to integrate renewable sources of energy into the power grid, with Solar PV being one of the most promising among them. Among other things, effective integration involves the ability to predict generation so that other sources can be curtailed or increased accordingly to meet the power demand. Time series modeling is the preferred technique due to the complexity of the factors. This project will explore the effect of various factors like training data size and resolution, and time series model order on the prediction accuracy. We will also be implementing custom programs to estimate the model parameters, make predictions and evaluate these models.**

***Index Terms*—ARMA, Time Series, Solar PV, Forecasting**

I. Introduction

A. Solar PV

Due to the irreversible effects of Global Warming, there is a worldwide initiative to integrate renewable sources of energy into the power grid. Solar Photo-Voltaic (PV) generation is the most prominent among these that is being explored and is promising due to the lower upfront costs involved, ease of installation and adoption, and the passive nature of generation with minimal human oversight. However, one of the major drawbacks associated with it is the restriction of power generation to hours of daylight. The demands of the grid that are not met by solar will have to be balanced out by other conventional and renewable sources. Therefore, the effective integration of solar will require the ability to predict the amount and duration of solar PV power generation, and effectively using other sources to match the excess demand.

B. Time Series Forecasting

A time-series is a set of data points that is spaced at equal intervals of time. In time-series forecasting, we use time-series data to create system models which use past time instants as input and generate predictions of future time instants as outputs. This model is then used to make predictions about future instants.

Time series forecasting is a very important technique that is used in a wide array of fields like Economics, Statistics, Weather-Forecasting and Signal Processing. It is generally adopted when the system is too complex to be represented as a mathematical model of various other inputs resulting in an output. This may be because of the complex relationships between the various inputs affecting the output, or the lack of input data that is spaced at the same time intervals as the output training data. [1]

In these cases, a time series model is used, wherein the output prediction of a random variable at any time-instant is a linear result of the random variable at past time instants. The output will not depend on other inputs to the system, unlike other conventional system models.

C. Time Series Forecasting for Solar PV Generation Forecasting

The power generated by a Solar PV cell is dependent on various factors like Incident Solar Radiation Intensity, the ambient temperature as well as the meteorological conditions. All these factors are strongly correlated, which makes it very difficult to characterize the power generated as a function of these parameters. [2] Therefore, we choose to use a time-series model where the solar power generation is the only variable considered and the power generation at any time instant is a function of the power generated at past instants. [3]

It is also important to note that forecasting is performed at different temporal scales based on the required application. A few of these are illustrated in Fig. 1.

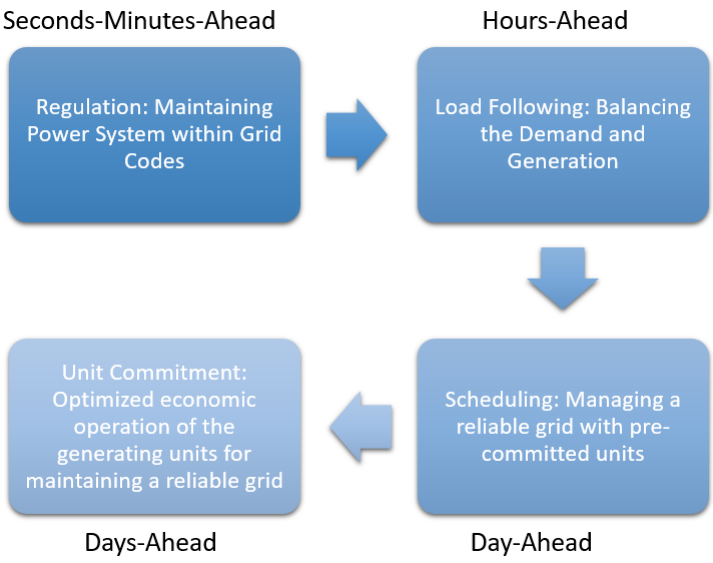

Fig. 1. Utilization of different forecasting temporal scales in Power Systems Operation

D. Auto-Regressive Moving Average (ARMA) Model

"A stationary process is a stochastic process whose unconditional joint probability distribution does not change when shifted in time. It's parameters such as mean and variance do not change over time. Such a process can be described using an ARMA model, consisting of two

polynomials. One of these is for the autoregression (AR) and the second is for the moving average (MA). The AR part involves regressing the variable on its own lagged values. The MA part involves modeling the error term as a linear combination of error terms occurring contemporaneously and at various times in the past. The model is usually referred to as the ARMA (p,q) model where p is the order of the AR part and q is the order of the MA part."
"ARMA models can be estimated using the Box-Jenkins method." [4] [5] [6]

II. Problem Statement

Using ARMA time series modeling to create prediction models for forecasting power generated by Solar Photovoltaic (PV) arrays using the Diamond Solar data. Developing and evaluating ARMA prediction models for various time scales of the Diamond Solar Data. Finally, evaluating the effect of different estimation techniques used to estimate the parameters of the ARMA models on the performance of prediction models.

III. Aims and Objectives

The aims and objectives are as follows;

- Transforming the given Diamond Solar Data (5min Resolution) to different time scales – 15 min, 60 min, 120 min, 180 min, Daily and Monthly resolution data.
- Using MATLAB's Econometrics Toolbox to develop ARMA models for the above-mentioned time scales and predict solar power generation.
- Developing ARMA estimation models based on Least Squares (LS) and Maximum Likelihood Estimation (MLE) to develop solar power generation models at different time scales using the appropriate model orders obtained previously from the Econometrics Toolbox.
- For intra-day time scales training data of one week and two weeks will be used respectively for training the above-mentioned models and a forecast of one week into the future will be generated.
- Comparing the accuracy of the generated forecasts from the MATLAB's Econometrics Toolbox and the developed ARMA Estimation models, different developed ARMA estimation models, different amount of training data and different time scales will be done.
- Evaluating the effect of different ARMA parameter estimation models on the forecast accuracy.
- Evaluating the effect of amount of training data on the forecast accuracy.
- Evaluating the effectiveness of ARMA prediction on different time scales

IV. Methods

The method comprises of three main parts: Data Pre-Processing, Development of ARMA models using MATLAB's Econometrics Toolbox and Development of ARMA Parameter Estimation techniques based on Least Squares and Maximum Loglikelihood Estimation. The schematic of the method is illustrated in Fig. 2.

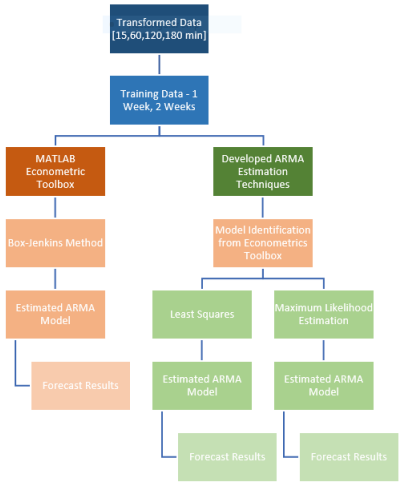

Fig. 2. ARMA Model Development Schematic

A. Data Pre-Processing

The data file Diamond_Solar_data.csv consists of three different time series at 5 minutes resolution corresponding to Diamond300, Diamond304 and Diamond306 contiguously arranged in a single column. This file is processed to give individual files corresponding to Diamond300, Diamond304 and Diamond306 at different time resolutions of 5 min, 15 min, 60 min, 2 hours (120 min), 3 hours (180 min), daily and monthly. The schematic for data pre-processing is illustrated in Fig. 3. The transformed data for 15 minute and one hour intervals are plotted against the original five minute data in Fig. 4. We can observe that averaging the data over an entire hour removes the large fluctuations seen with the five-minute and fifteen-minute data. While this will provide better prediction models, it will also result in a loss of resolution.

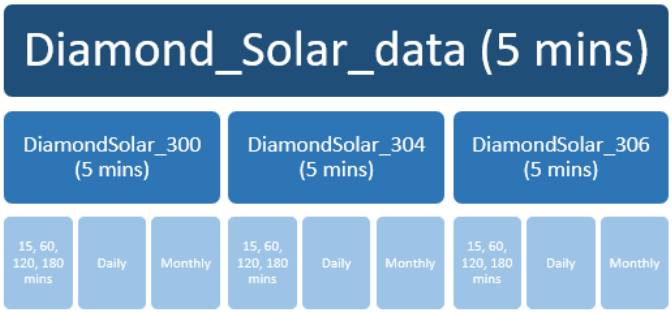

Fig. 3. Data Pre-Processing Schematic

B. ARMA Model Development using MATLAB's Econometrics Toolbox

The ARMA model development is done using Box-Jenkins Method [4], which has the following steps as illustrated in the Fig. 5.

1) **Model Identification:** The univariate time series is checked for stationarity and if not, stationary it has to be converted into a stationary series by applying differencing. The Auto-Correlation Function (ACF) and

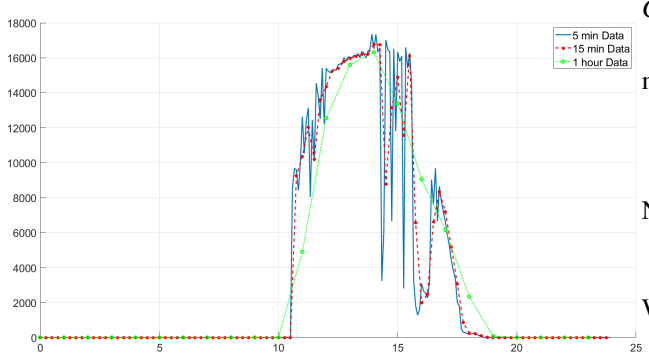

Fig. 4. Plot of Transformed Data for 15 min and 1 hour

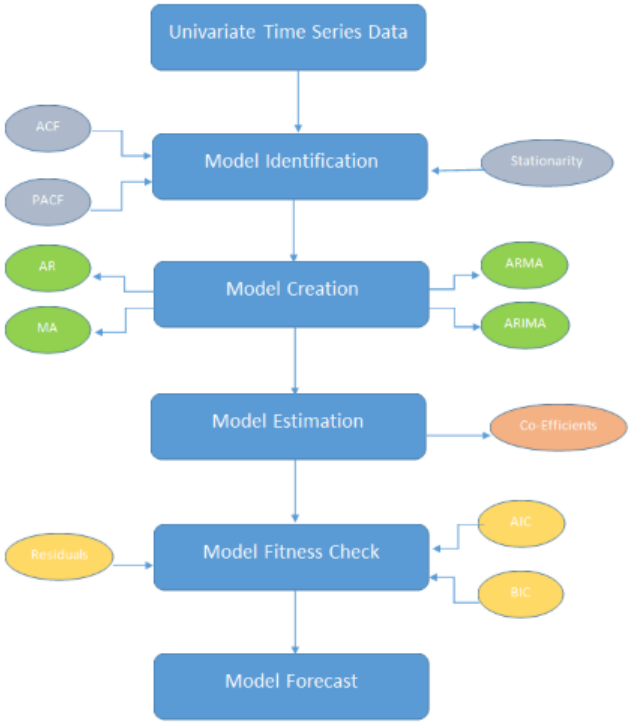

Fig. 5. ARMA Model Development Schematic

the Partial Auto-Correlation Function (PACF) of the stationary univariate time series are checked for determining the order of the AR and MA process generating the series.

2) **Model Creation:** On the basis of the Model Identification a number of models around the determined AR and MA orders are created.
3) **Model Estimation:** These models are then estimated to give the appropriate coefficients.
4) **Model Fitness Check:** The estimated models are then evaluated using Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). Lower the AIC and BIC better the model.
5) **Model Forecast:** The best model determined from the previous step is used to forecast the series.

C. ARMA Parameter Estimation Techniques

1) Least Squares Estimation Method: The parameter estimation of an ARMA process using Least Squares is as follows;

$$\widehat{y_t} = a_1 y_{t-1} + a_2 y_{t-2} + ... + b_1 \epsilon_{t-1} + b_2 \epsilon_{t-2} + ... + \epsilon_t$$

Now;

$$\widehat{y_t} = ay + b\epsilon$$

Where;

$$a = [a_1 \;\; a_2 \;\; a_3....]$$
$$b = [b_1 \;\; b_2 \;\; b_3....]$$
$$y = [y_{t-1} \;\; y_{t-2} \;\; y_{t-3}...]^T$$
$$\epsilon = [\epsilon_{t-1} \;\; \epsilon_{t-2} \;\; \epsilon_{t-3}...]^T$$

The Least Squares Estimate for the Parameters is given by;

$$\widehat{\theta}(a,b) = \arg\min_{\theta} \frac{1}{2}\left[(y_t - \widehat{y_t})^2\right]$$

2) Maximum Likelihood Estimation Method: The maximum likelihood function for an ARMA process when the error is assumed to be distributed normally is give as follows;

$$\widehat{\theta}(a,b) = \arg\max_{\theta} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_t - \widehat{y_t})^2}{2\sigma^2}}$$

V. Initial Experiments and Results

We have performed some initial experiments on the fifteen minute data and made a few forecasts assuming an AR(4) model with seasonality of 24 hours. The results are as shown in Fig. 6 and Fig. 7.

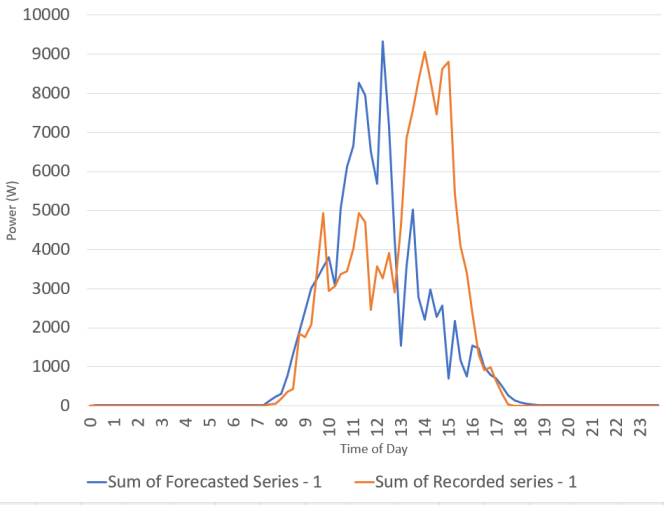

Fig. 6. Forecast results for 1st day with a week of training data

We can see that the forecast with just one week of training data appears to be closer to the actual observations than the forecast made with two weeks of training data. Therefore, we will need to use an appropriate amount of training data based on the forecast duration we are looking to make, with

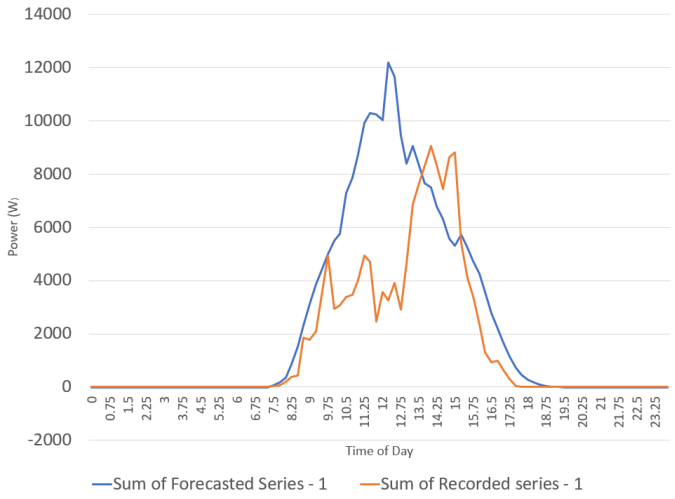

Fig. 7. Forecast results for 1st day with two weeks of training data

a week's data sufficient to make a forecast for a single day. Apart from that, we can see that the day for which the forecast has been made has anomalous recordings due to possible cloud conditions. Therefore, we can see that even a well-tuned ARMA model cannot account for it and will not be able to predict the actual output under such conditions.

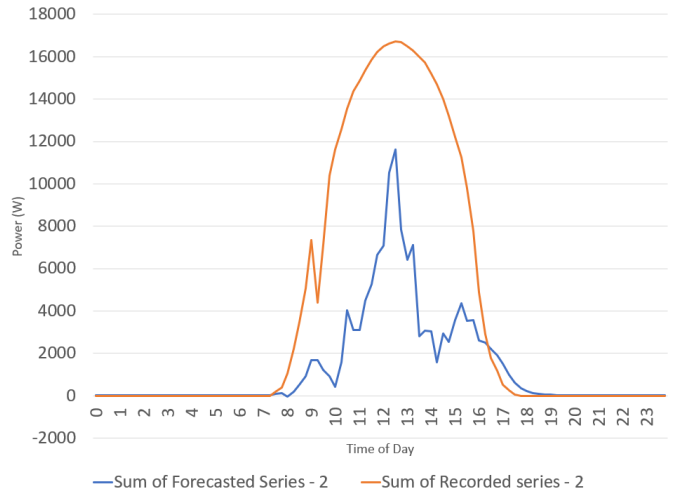

Fig. 8. Forecast results for 2nd day with a week of training data

Fig. 8 and Fig. 9 show the forecasts for the second day after the training period. We can observe that the accuracy reduces as we make forecasts further away from the training period.

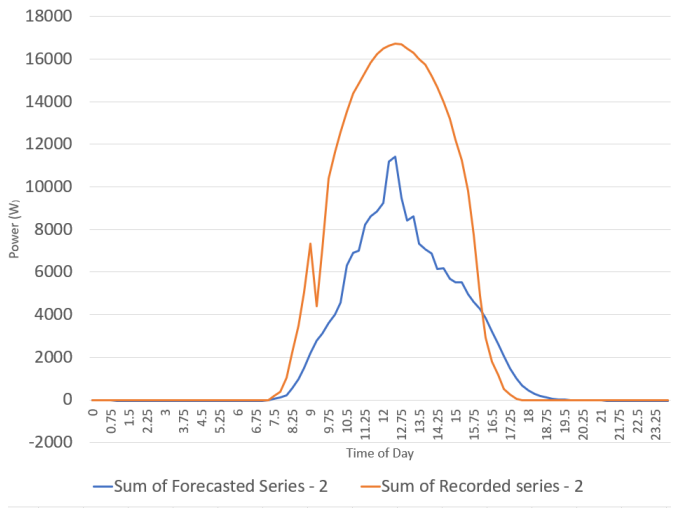

Fig. 9. Forecast results for 2nd day with two weeks of training data

VI. Results

A. Dealing with Non-stationary Data

Time series forecasting works well with data that is at least weakly wide-sense stationary. There are various mathematical tests to examine a time series for this. For our analysis, we use the KPSS (Kwiatkowski, Phillips, Schmidt, and Shin) test for trend stationarity and the ADF (Augmented Dickey-Fuller) test for the presence of a unit root.

The outputs for these tests and their related implications are listed in Table I. If the conclusion is that the series is trend stationary, using seasonality should help make the series wide-sense stationary (WSS). In the case of the tests indicating the presence of a unit root, single lags can be used to turn it into a WSS series.

TABLE I
THE KPSS AND ADF TEST FOR STATIONARITY

Result \ Test	**KPSS Test**	**ADF Test**
Null Hypothesis	Trend Stationary	Unit Root is present
Alternative Hypothesis	Unit Root is present	No Unit Root, can be stationary or trend stationary

Once the series has been deemed to be sufficiently WSS, we move on to determining the order of the ARMA model. This is done using the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). The outputs of these two functions are used to decide the model order using the various outcomes described in Table II, with the ACF deciding the order of the MA component and the PACF deciding the order of the AR component.

TABLE II
FINDING ARMA MODEL ORDER FROM ACF AND PACF PLOTS

Model	**ACF**	**PACF**
AR	Decaying	Significant till p Lags
MA	Significant till q Lags	Decaying
ARMA	Decaying	Decaying

We have used the ACF and PACF functions on a few different versions of the time series in the provided data.

In the first instance, we have applied the functions on the unaltered original data series. The results are as shown in Fig. 10. We observe that the ACF has an exponentially decaying plot while the PACF has a plot that decays much faster, with about 4 significant lag orders.

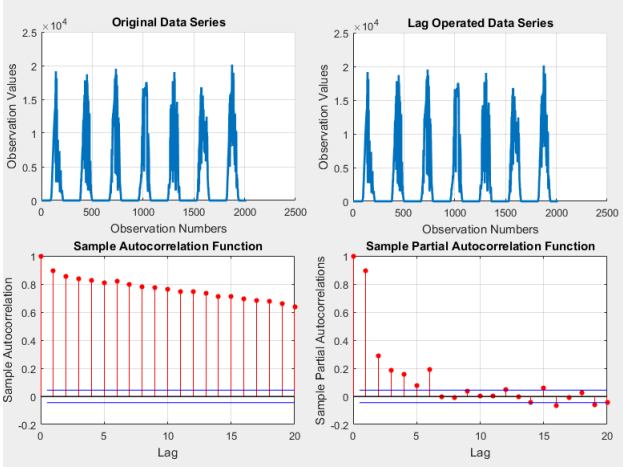

Fig. 10. Diamond Solar 300 undifferenced data with ACF and PACF Plots

In the second instance, we use the functions on a single differenced version of the time-series. The results are as shown in Fig. 11. We observe that both the ACF and PACF is decaying fast, with 2 significant lag orders in the MA component and 3 significant lag orders in the AR component.

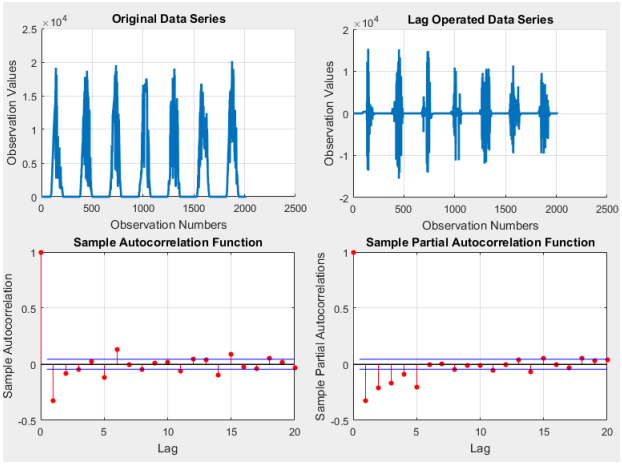

Fig. 11. Diamond Solar 300 original and single differenced data with ACF and PACF Plots

In the third instance, the functions are used on a version of the original data that has one single lag and a seasonal lag of 24 hours.The results are as shown in Fig. 12. We can see that there is one significant lag order in the MA component and 3 significant lag orders in the AR component.

B. ARMA Model Selection and Evaluation

Once we have decided the number of significant lag orders using the ACF and PACF, we move on to creating the actual

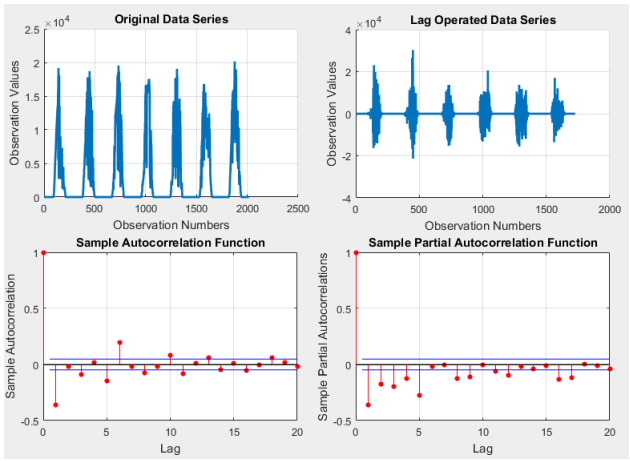

Fig. 12. Diamond Solar 300 original and single-seasonal differenced data with ACF and PACF Plots

time series models and estimating their parameters. From the previous section, we have seen that the various versions of the time series each have associated AR and MA components. Therefore, we will be creating ARMA models for each of these series and then pick the best model for a given forecast window based on the average absolute error percentage.

For the undifferenced series, we will be creating 3 AR models (AR-1, AR-2 and AR-3). For the differenced series with one single lag, we will be creating 3 MA models (MA-1, MA-2 and MA-3) and 3 ARMA models(ARMA-1, ARMA-2 and ARMA-3). The details for these models are as shown in Table III.

TABLE III
ARMA MODELS SPECIFICATIONS FOR EVALUATION

Model	Single Lag	AR	MA	SAR	SMA
AR-1	0	1,2,3	0	0	0
AR-2	0	1,288		0	0
AR-3	0	1,288,576, 864,1152		0	0
MA-1	1	0	1,2,3	0	0
MA-2	1	0	1	0	288
MA-3	1	0	1,576, 864,1152	0	288
ARMA-1	1	1,2,3	1,2,3	0	0
ARMA-2	1	1,288	1,288	288	288
ARMA-3	1	1,288,576	1,288,576	288	288

Once these models have been created and their parameters have been estimated, they are evaluated for various forecast window lengths using average absolute error percentage. The models have all been trained on 2 weeks of data. The results are as shown in Table IV. We can see that the AR-3 model for undifferenced series provides the best results for shorter forecast windows. MA-2 and ARMA-3 for single differenced series provide the best results for longer forecast windows. The forecasts for one day-ahead for all these models are represented graphically in Fig. 13.

C. Effect of Amount of Training Data

We next try to determine the amount of training data that would provide more accurate forecasts. To do this, we have

TABLE IV
FORECAST ERROR PERCENTAGE PER FORECAST PERIOD FOR DIFFERENT ARMA MODELS

FP\Model	AR-3	MA-2	ARMA-3
5 minute	**20.1**	30.4	52.8
15 minute	**7.2**	18.3	31.5
30 minute	**6.6**	28.9	19.9
1 hour	17.6	20.2	**15.0**
3 hour	35.4	**17.7**	26.9
6 hour	27.9	**18.8**	26.6
12 hour	70.6	**22.2**	30.1
24 hour	83.4	**26.2**	35.6
2 day	87.6	**38.3**	43.2
3 day	145.3	98.7	**94.1**
4 day	127.7	79.5	**76.9**
5 day	123.2	69.1	**68.3**
6 day	132.7	68.3	**66.2**
7 day	167.9	114.2	**106.2**

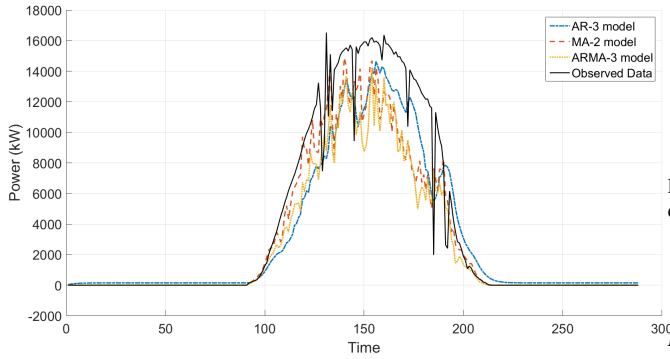

Fig. 13. Day-Ahead Power Prediction with AR-3, MA-2 and ARMA-3 Models

picked the MA-2 model which provides the best results for day-ahead forecasts (the most commonly used temporal scale in the power industry). We will be training four different versions of this model on 1 week, 2 weeks, 3 weeks and 4 weeks of training data. These models are then used to make forecasts for various periods and the results are tabulated in Table V.

TABLE V
FORECAST ERROR PERCENTAGE PER FORECAST PERIOD FOR DIFFERENT AMOUNTS OF TRAINING DATA WITH MA-2

FP\Training Data	1 Week	2 Weeks	3 Weeks	4 Weeks
5 minute	**22.7**	30.4	35.8	36.0
15 minute	**14.7**	18.3	20.2	20.3
30 minute	**26.7**	28.9	28.8	28.6
1 hour	20.0	20.2	19.7	**19.5**
3 hour	**15.7**	17.7	18.1	18.0
6 hour	**18.4**	18.8	19.4	19.3
12 hour	23.1	**22.2**	23.3	23.3
24 hour	27.3	**26.2**	27.6	27.5
2 day	**38.3**	38.3	39.1	39.1
3 day	100.1	98.7	96.1	**96.0**
4 day	80.5	79.5	79.0	**78.9**
5 day	70.0	**69.1**	70.4	70.3
6 day	70.4	68.3	67.1	**67.1**
7 day	117.7	114.2	111.2	**111.2**

We can see that the highest accuracy for shorter forecast periods is achieved when we provide shorter amounts of training data like 1 week. For longer forecast periods, we get better results when we train it on longer periods training data.

Using training periods that are longer than 4 weeks will average out the predictions and will not provide good results at forecast periods of 1 week and less.

Day-ahead forecasts are made using the MA-2 models trained on different lengths of training data. The results are illustrated in Fig. 14.

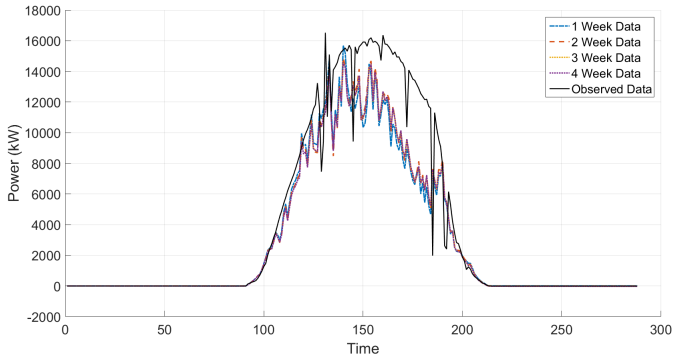

Fig. 14. Day-Ahead Power Prediction with MA-2, at 5 min resolution with different amounts of training data

D. Effect of Time Resolution of Data Series

As we had discussed earlier, different temporal scales of forecast are used for various applications. Therefore, it makes sense to examine the effect of the time resolution of the training data on the accuracy of the forecast. From an observation of the training data, we can see that the time series plot is more jagged at high resolutions like 5 minutes while it is more rounded out at lower resolutions like 1 hour. Therefore, this should have a bearing on the forecast accuracy.

To perform this study, we consider 4 different versions of the MA-2 model that have been trained on two weeks worth of data at 4 different time resolutions of 5 minutes, 15 minutes, 1 hour and 2 hours. This model is then used to make a single day-ahead prediction. The forecast error percentage for the 4 different resolutions is tabulated in Table VI.

TABLE VI
FORECAST ERROR PERCENTAGE PER DAY FOR DIFFERENT TIME RESOLUTION TRAINING DATA WITH MA-2

Time Resolution	Forecast Error Percentage [1st Day]
5 min	26.2
15 min	21.6
1 hour	21.8
2 hour	21.1

We can see that the error percentage is lowest at low resolutions like 1 hour and 2 hour. While these time series will provide better forecasts, we will lose out on the resolution required for certain applications. We can also observe the plots for day-ahead forecast with different resolutions for training data in Fig. 15 and Fig. 16.

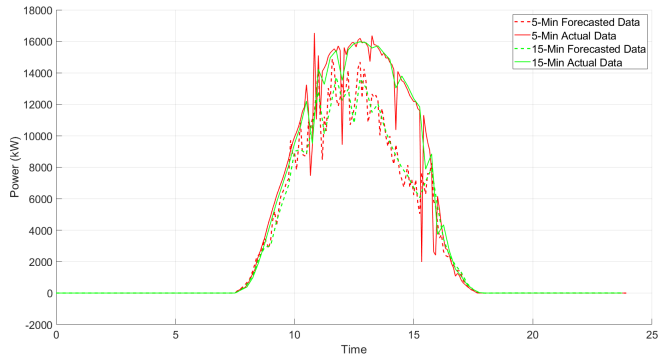

Fig. 15. Day-Ahead Power Prediction with MA-2, 2-Week Data at 5 and 15 min resolution

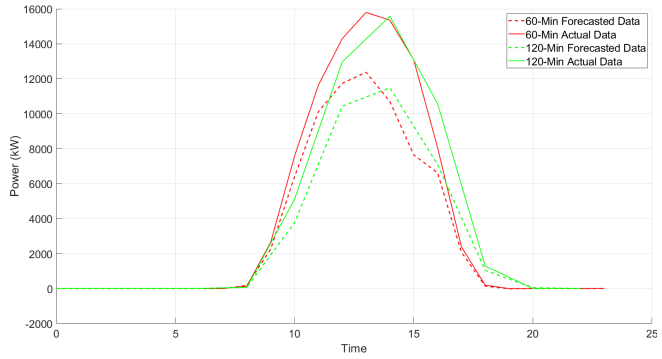

Fig. 16. Day-Ahead Power Prediction with MA-2, 2-Week Data at 60 and 120 min resolution

E. Comparison of MATLAB and Self-Developed AR and MA Models

We have so far established the best parameters to be making a single day-ahead forecast. In this next section, we are going to compare the MATLAB model estimation function with the custom program we have written that estimates the model parameters using Least Squares estimation and Max Likelihood estimation.

To do this experiment, we are picking the AR-3 model that gave us good forecasts at shorter time periods and the MA-2 model that gave us good results at longer forecast periods. We will be training both of these models on 2 weeks worth of data at a time resolution of 5 minutes and estimate the parameters for these models using MATLAB's estimation function, as well as our own program for Least Squares estimation and Max Likelihood estimation. We then use these models to make forecasts for different periods. The forecast error percentages are tabulated in Table VII and Table VIII.

We have also plotted the day-ahead forecasts for all of these models in Fig. 17 and Fig. 18.From these studies, we can see that the MATLAB estimated model provides the best forecast accuracy, but the Least Squares and Max Likelihood estimated models are not that far-off from the MATLAB predictions.

TABLE VII
FORECAST ERROR PERCENTAGE PER FORECAST PERIOD FOR AR-3 ESTIMATED WITH MATLAB, SELF-DEVELOPED LS AND MLE

FP\Models	AR-3 MATLAB	AR-3 LSE	AR-3 MLE
5 minute	20.1	46.9	46.9
15 minute	**7.2**	51.2	51.2
30 minute	**6.6**	47.5	47.5
1 hour	17.6	51.3	51.3
3 hour	35.4	50.8	50.8
6 hour	27.9	38.2	38.2
12 hour	70.6	143.4	143.4
24 hour	83.4	169.2	169.2
2 day	87.6	164.0	164.0
3 day	145.3	205.5	205.5
4 day	127.7	198.6	198.6
5 day	123.2	203.0	203.0
6 day	132.7	217.2	217.2
7 day	167.9	237.4	237.4

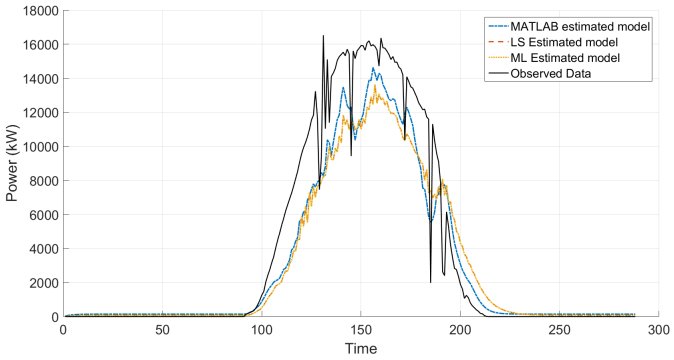

Fig. 17. Day-Ahead Power Prediction with AR-3 estimated model, 2-Week Data at 5 min resolution with MATLAB, self-developed LS and MLE

TABLE VIII
FORECAST ERROR PERCENTAGE PER FORECAST PERIOD FOR MA-2 ESTIMATED WITH MATLAB, SELF-DEVELOPED LS AND MLE

FP\Models	MA-2 MATLAB	MA-2 LSE	MA-2 MLE
5 minute	30.4	**3.0**	**3.0**
15 minute	18.3	7.3	7.9
30 minute	28.9	11.3	17.6
1 hour	20.2	**16.4**	23.6
3 hour	**17.7**	19.9	23.4
6 hour	**18.8**	20.6	25.6
12 hour	**22.2**	27.9	30.5
24 hour	**26.2**	32.9	36.0
2 day	**38.3**	48.3	52.6
3 day	**98.7**	123.5	128.0
4 day	**79.5**	99.9	105.0
5 day	**69.1**	85.7	91.5
6 day	**68.3**	85.1	90.6
7 day	**114.2**	141.9	146.9

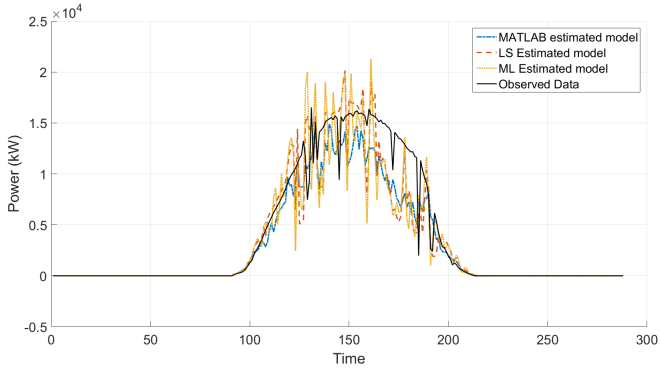

Fig. 18. Day-Ahead Power Prediction with MA-2 estimated model, 2-Week Data at 5 min resolution with MATLAB, self-developed LS and MLE

VII. Conclusion

We have successfully created all the code to preprocess the data and convert it to a series spaced at an appropriate time interval. We have also used this data to create an ARMA model using the in-built MATLAB functions and find estimates for the parameters in the model. The model is able to predict the power generation to a reasonable accuracy level, though further improvements can be made. The final submission will include programs to create a model and estimate its parameters using Least Squares and Max Likelihood estimates.

References

[1] P. A. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017.

[2] W. Zhou, H. Yang, and Z. Fang, "A novel model for photovoltaic array performance prediction," *Applied energy*, vol. 84, no. 12, pp. 1187–1198, 2007.

[3] R. Huang, T. Huang, R. Gadh, and N. Li, "Solar generation prediction using the arma model in a laboratory-level micro-grid," in *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*. IEEE, 2012, pp. 528–533.

[4] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[5] R. Baillie, "Maximum likelihood estimation of time series models," 2017.

[6] N. Sandgren, P. Stoica, and P. Babu, "On moving average parameter estimation," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 2348–2351.