Python for Energy Systems and Sustainability Analysis 2019/2020

Comparison of different forecasting methods for renewable energy generation in Germany

Kamal Elgamel (Mat.Nr. 4774129)

Youmi Kim (Mat.Nr. 4774232)

MSc. Sustainable System Engineering

University of Freiburg

Contents

1. Introduction

With the help of the response to climate change crisis, environmental issue of fossil fuel and government policies, the share of renewables in the net electricity generation in Germany, that is, the amount coming from the socket, rose from 40.6 percent in 2018 to 46 percent in 2019, and surpassed the share from fossil fuels (40 percent) for the first time. [(1)]

As the portion of renewable resources in energy grid has increased, the importance of balancing the demand and supply is significant. One of the most challenge should be the forecasting renewable energy generation. This is because those are not controllable but rather highly depend on weather. Therefore, we need to forecast the generation as accurate as possible because accurate forecasting will play an important role in power systems operations with respect to technical and economic impacts such as electric power quality, day-ahead planning, energy storage management, ancillary costs associated with general volatility and so on.[(2)] The aim is further stability of the grid and take steps towards more green energy implementation.

In the past decades, a large number of forecasting models and methods have been tried. These methods can be divided into two categories: classical approaches such as auto regressive integrated moving average (ARIMA) models and artificial intelligence (AI) based techniques [(3)]

In this study, three models –ARIMA, Seasonal ARIMA (SARIMA) and Long Short Term Memory (LSTM) are compared to analyzed which model is the most suitable to forecast solar and wind power generation. ARIMA and SARIMA are modified from Auto Regressive Moving Average (ARMA), which is one of the most popular time series forecasting models.[(4)] LSTM is based artificial intelligence techniques. The accuracies of each model are presented in Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) value and models are implemented in Python.

2. Background

2.1. ARIMA model

ARIMA is a type of univariate forecasting model that explains a given time series based on its own past values, which are its own lags and the lagged forecast errors, so that equation can be used to forecast future values. ARIMA model is applied in cases where data show evidence of non-stationarity, where an initial differencing step (the integrated part of the model) can be applied one or more times to eliminate the non-stationarity. [(5)]

The model is often represented as *ARIMA(p,d,q)* where *p, d, q* are non-negative integer hyperparameter. ARIMA model is a conglomeration of three models as follows:

Auto Regression (AR): The model indicates that the predicted output in a time series is a function of its lags, denoted by the hyperparameter *p* which represents the number of lags. *AR(p)* can be written as equation (1) [(6)]

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + e_t \qquad (1)$$

Where c is intercept, φ_p is autoregressive model parameters which are to be estimated, e_t is white noise.

Moving Average (MA): A model that uses the relation between a current data point and the residual error from a moving average model applied to the lagged data point, where *q* is the hyperparameter representing the number of preceding lag values. *MA(q)* can be written as equation (2) [(6)]

$$y_t = c + e_t + \vartheta_1 e_{t-1} + \vartheta_2 e_{t-2} + \cdots + \vartheta_q e_{t-q} \qquad (2)$$

Where c is intercept, ϑ_q is autoregressive model parameters which are to be estimated.

Integrated (I): The differenced series is the change between consecutive observations in the original series. For example, if we only consider a simple first order differenced time series, *I(d)* can be written as equation (3). [(6)]

$$y'_t = y_t - y_{t-1} \qquad (3)$$

Where y'_t is differenced time series.

A full *ARIMA(p,d,q)* model is written as equation (4) where degree *(d)* of first differencing is involved. [(6)]

$$y'_t = c + \varphi_1 y'_{t-1} + \cdots + \varphi_p y'_{t-p} + \vartheta_1 e_{t-1} + \cdots + \vartheta_q e_{t-q} + e_t \qquad (4)$$

Where y'_t is differenced time series, φ_p is autoregressive model parameters, ϑ_q is autoregressive model parameters and e_t is white noise.

In this study, we are forecasting solar/wind power generation increasing over the year which is violate the stationarity, thereby we use ARIMA model which we include a differencing term to transform the model as stationary time series.

2.2. SARIMA model

SARIMA is another variation of ARIMA which is dealing with forecasting using univariate time series data. One major difference between SARIMA and ARIMA is that it takes seasonal components into account for forecasting.

This model is often represented in the form of *SARIMA(p,d,q)(P,D,Q)*$_m$.The first three parameters *(p,d,q)* are as formerly mentioned. On the other hand, the next parameters *(P,D,Q)*$_m$ are described as seasonal part of the model. *P* is the seasonal autoregressive lag, *D* is the seasonal order difference, *Q* is the seasonal moving average order, and finally *m* represents the number of time steps for a single seasonal order.

In this study, we consider SARIMA model since we can assume that solar/wind power generation might have certain seasonality based on weather dependency.

2.3. LSTM model

LSTM is one type of Recurrent Neural Network (RNN). To explain what LSTM RNN are, first we should briefly define what a neural network is. Neural network has been motivated from its inception by the recognition that the way the human brain processes and computes information is different than that of a digital computer.

A neural network can then be described as a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use, it resembles the brain in two respects:

- Knowledge is acquired by the network from the environment through a learning process
- Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge. (7)

A normal feedforward artificial neural network assumes that the inputs and the outputs are independent which is inappropriate in forecasting process. Therefore, recurrent neural networks were introduced.

Recurrent neural networks contain cycles that feed the network activations from a previous time step as inputs to the network to influence predictions at the current time step. These activations are stored in the internal states of the network which can in principle hold long-term temporal contextual information. This mechanism allows RNNs to exploit a dynamically changing contextual window over the input sequence history. (8)

Unfortunately, the range of contextual information that standard RNNs can access is in practice quite limited. The problem is that the influence of a given input on the hidden layer, and therefore on the network output, either decays or blows up exponentially as it cycles around the network's recurrent connections. This shortcoming is referred to in the literature as the vanishing gradient problem. (9)

Finally, a particular form of RNN called LSTM was introduced to counter this problem by having the ability to deal with vanishing and exploding gradients.

An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. These blocks can be thought of as a differentiable version of the memory chips in a digital computer (10). LSTM does not use activation function within its recurrent components, the stored (8) values are not modified, and the gradient does not tend to vanish during training. Usually, LSTM units are implemented in "blocks" with several units. These blocks have three or four "gates" (for example, input gate, forget gate, output gate) that control information flow drawing on the logistic function.

3. Method

3.1. Methodology

The procedure flow in this study is shown in the Figure 1. All procedure are conducted in programming language Python. For all model, Solar/Wind generation is used as input data.

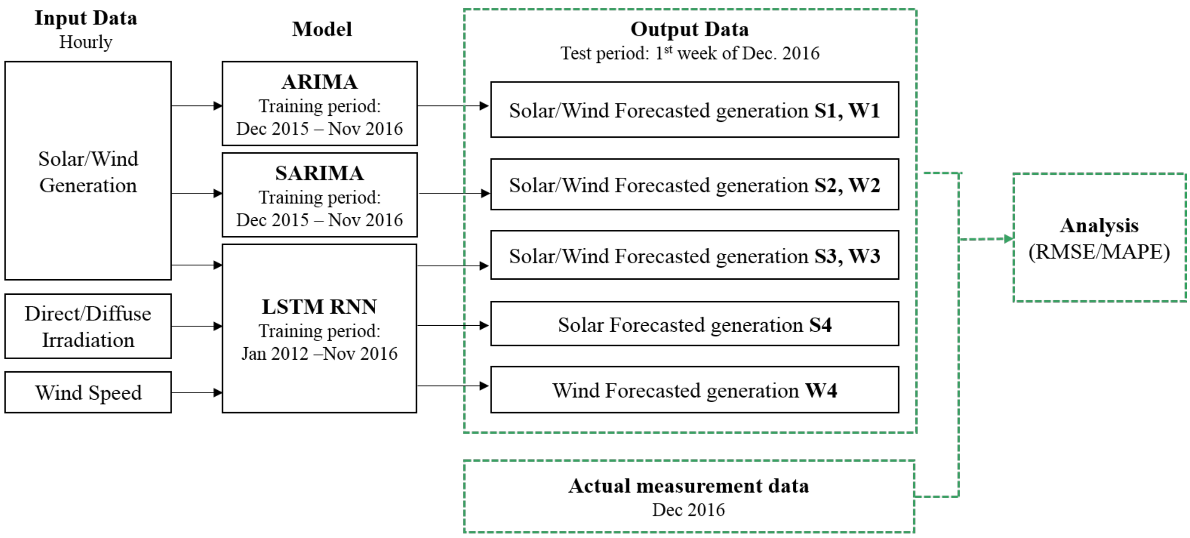

Figure. 1 | Procedure flow of project

As shown in the output data part, each model got the ID for convenience. 'S' represents solar model and 'W' represents wind model. The detailed input data information is represented in Table 1. S1, S2, S3, W1, W2, W3 models are univariate which have one input data while S4 and W4 models are multivariate which have multi input data including weather data. Solar irradiation and wind speed data are used for additional data for respectively for solar generation and wind generation.

Each model runs with corresponding input data and brings output data which is the forecasted generation. These outputs are compared with actual measurement data. Then model performance is analyzed using root mean square error (RMSE) and mean absolute percentage error (MAPE) to test the validity of these models.

Because of the computational limitation, the training period for ARIMA and SARIMA models are one year meanwhile the training period for LSTM model is 5 years.

Table 1. Model description

ID	Model	Input data	ID	Model	Input data
S1	ARIMA	Solar generation	W1	ARIMA	Wind generation
S2	SARIMA	Solar generation	W2	SARIMA	Wind generation
S3	LSTM	Solar generation	W3	LSTM	Wind generation
S4	LSTM	Solar generation + Irradiation	W4	LSTM	Wind generation + Wind speed

3.2. Input Data description

For input data, we used already pre-processed data from a project called Open Power System Data (OPSD). (11) The period of data is determined as the dates where all input data exist.

3.2.1. Solar/Wind generation data

As our main input data for all models, we used solar and wind generation data as shown below.

- Variables: Actual solar generation (MW/h), Actual wind generation (MW/h),
- Period: 1. Jan. 2012 – 31. Dec. 2016
- Resolution: Hourly
- Package version: 5. Jun. 2019
- The data has been downloaded from the sources (50Hertz, Amprion, TenneT, and TransnetBW), resampled and merged in a large CSV file with hourly resolution.
- It covers the European countries using a population-weighted mean across all MERRA-2 grid cells within the given country.
- Reference address: https://doi.org/10.25832/time_series/2019-06-05

Both solar and wind generation time series show strong seasonality as shown in Figure. 2. Even though it is slight, increasing trends over the year are shown as well.

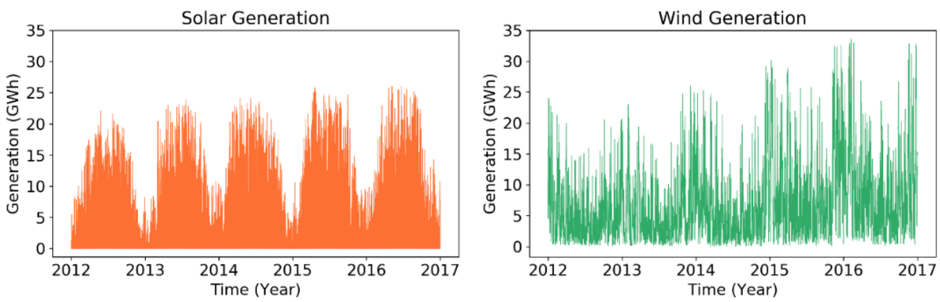

Figure. 2 | Time series of solar generation (left) and wind generation (right) over the simulation period

3.2.1. Weather data

As described in table 1, the multivariate models (S4, S5, W4) have additional weather data as input. We used solar irradiation and wind speed as shown below.

- Variables: Direct Irradiation (W/m^2), Diffuse Irradiation (W/m^2), Wind Speed at 10 m (m/s)
- Period: 1. Jan. 2012 – 31. Dec. 2016
- Resolution: Hourly
- Package version: 9. Apr. 2019
- This data is aggregated by 'Renewables.ninja' from the NASA MERRA-2 reanalysis.
- It covers the European countries using a population-weighted mean across all MERRA-2 grid cells within the given country.
- Reference address: https://doi.org/10.25832/weather_data/2019-04-09

Because the reference data we used is average value of whole Germany, rough assumptions are included. For example, we did not forecast separately sunny day and cloudy/rainy day. With the same reason, we used both direct and diffuse irradiation since we cannot calculate global irradiation with simple summation.

As shown in the Figure. 3, irradiation time series show even stronger seasonality than solar generation. Wind speed time series also shows seasonality. However, none of those has increasing trends over the year.

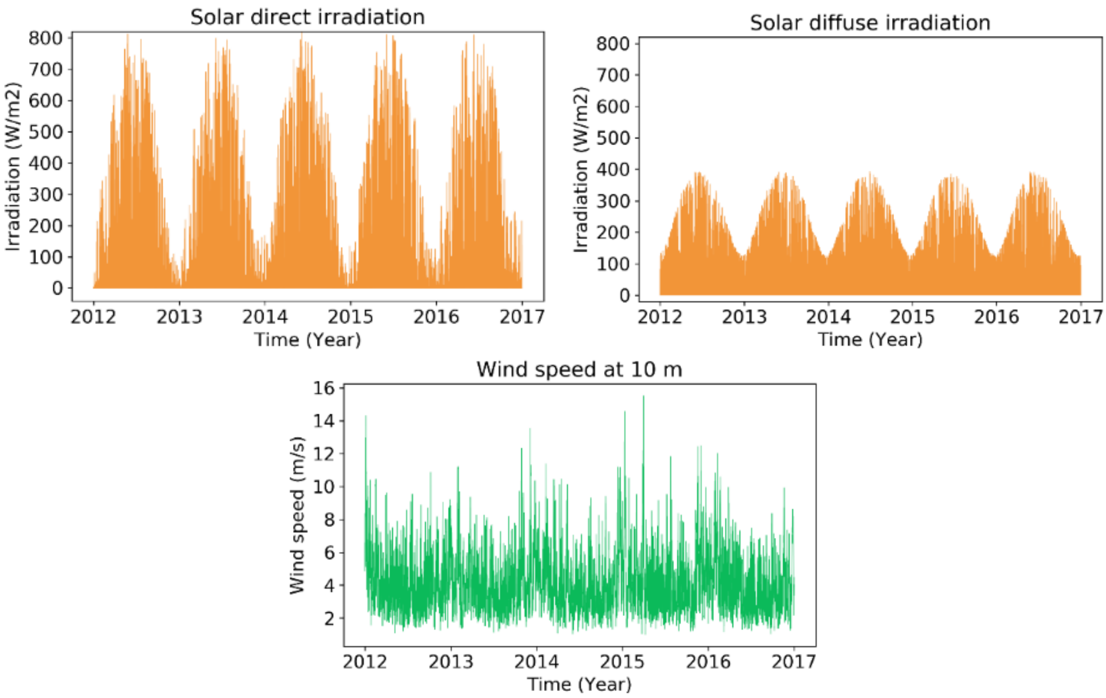

Figure. 3 | Time series of solar irradiation (Lest and Middle) and wind speed at 10 m (Right) over the simulation period

3.3. Model description

3.3.1. ARIMA / SARIMA

ARIMA arnd SARIMA models are univariate which has one input data. We used Solar and Wind generation data with training period of 1 year from December 2015 to November 2016. Hourly prediction is simulated and 1st week of December is tested for validation.

Applied models are conducted in three steps as shown in the figure 4.

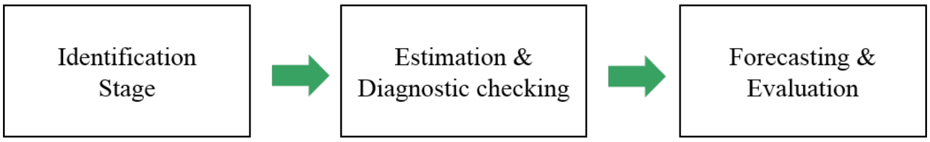

Figure. 4 | Three Steps for ARIMA & SARIMA modeling

- **Identification stage**

Candidate ARIMA models are identified for this state. Many tests for computing the best parameters are performed. Popularly, Partial autocorrelation and autocorrelation functions are used to identify the parameters p and q respectively. Stationarity tests can be also performed to determine whether differencing is needed or not.

- **Estimation and diagnostic checking**

This stage is done to estimate the parameters of the chosen model, by producing diagnostic statistics to help judge the adequacy of the model such as Akaike Information Criterion (AIC).

- **Forecasting and evaluation**

Future values of the time series are finally forecasted and the accuracy of the forecast is tested against different set of values of the time series that was not used for training.

In this study, a built in function in Python called auto.arima() was used for the identification and estimation stage. Model estimation is a tedious stage which requires large time consuming and high processing power with much trial and error. This auto arima function performs these stages and recommend the best parameter (order) for the models with the least AIC and BIC. According to the result of auto.arima, the order of each model is decided as shown in Table 2

Table. 2 | The parameter of each model with AIC and BIC values

ID	Model	Parameter	AIC	BIC
S1	ARIMA	(5, 1, 3)	131580.273	131651.078
W1	ARIMA	(2, 1, 3)	132854.582	132904.146
S2	SARIMA	(1, 1, 1) (0, 1, 1, 12)	132138.816	132174.212
W2	SARIMA	(1, 1, 1) (1, 1, 1, 24)	131409.798	131452.265

3.3.2. LSTM

LSTM models can be both univariate and multivariate. In this study, we simulated for both. Solar and Wind generation data were used as base input data for both models. For multivariate model, additional solar irradiation and wind speed data were used respectively for Solar and Wind generation data. Training period is 5 years from January 2012 to November 2016. This is because in case of LSTM model the computing time required was much less than ARIMA and SARIMA models to allow much available data included. Meanwhile, same as ARIMA and SARIMA model, hourly prediction is simulated and 1st week of December is tested for validation.

LSTM model is conducted in four steps as shown in the figure 5.

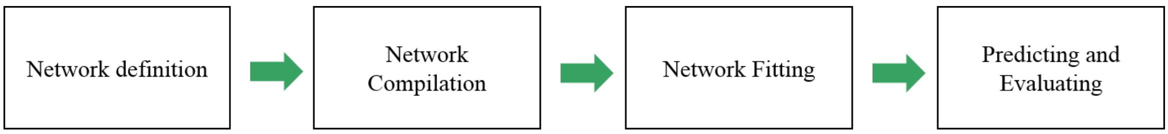

Figure. 5 | Four Steps for LSTM model construction

- **Network defining**

Neural Networks are defined in Keras as a sequence of layers. The container for these layers is the sequential class. The network consists of four layers: One input layer for data input of three dimensions shape including rows, time steps, and columns, followed by two hidden layers, and finally a dense layer. The input layer as well as the hidden layers of our model has 50 neurons, while the output dense layer has only one neuron.

- **Network compilation**

Compilation is considered as an efficiency step, by transforming the input into a highly efficient sequence of series of matrices to be executed later. An optimization function must be defined for the network together with a loss function that aims to be minimized by the optimization function to evaluate the model. In this study, Adaptive Moment Estimation (Adam) is chosen as optimization function with mean squared error. The loss minimization over time for solar and wind power generation can be seen in Figure 6 respectively.

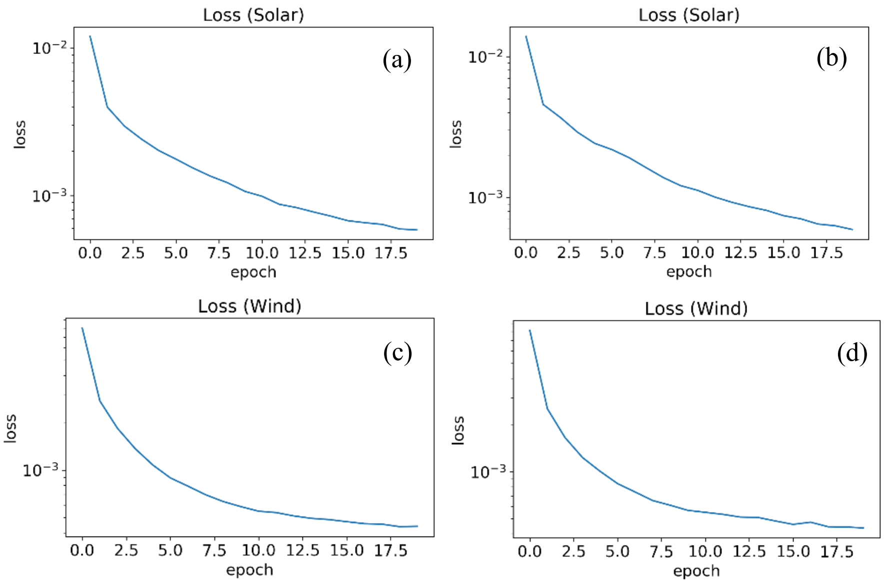

Figure 6 | Loss according to epochs for Solar univariate (a), Solar multivariate (b), wind univariate (c) and wind multivariate (d)

- **Network Fitting**

Once the network is compiled, it can be fit. This means we can adapt the weights on a training data set. Fitting the network requires the training data to be specified, both a matrix of input patterns X, and an array of matching output patterns Y. The network is trained using the backpropagation algorithm and optimized according to the optimization algorithm and loss function specified when compiling the model. The backpropagation algorithm requires that the network be trained for a specified number of epochs or exposures to the training dataset. In this study, we used 20 epochs for simplicity. Each epoch can be partitioned into groups of input-output pattern pairs called batches. This defines the number of patterns that the network is exposed to before the weights are updated within an epoch. It is also an efficiency optimization, ensuring that not too many input patterns are loaded into memory at a time. In this study a batch size of 250 is adopted.

- **Predicting the values and evaluating**

As soon as we confirm the validity and accuracy of the model, forecasting can now commence. The output predictions are then provided by the dense layer of the LSTM structure.

3.4. Analysis description

The final step of any forecasting process is the measurement of the forecasting accuracy. In this study, we used Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE).

3.4.1. MAPE

The mean absolute percentage error (MAPE) is a statistical measure of how accurate a forecast system is. It measures this accuracy as a percentage, and can be calculated as the average absolute percent error for each time period minus actual values divided by actual values, this is given by equation (5)

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (5)$$

Where n is the number of samples, A_t is the actual value and F_t is the forecasted value.

However, MAPE suffers from some drawbacks as well, one of the most relevant ones we noticed for this study, was its likeability to give infinite values. This is evident if the actual values were zero, which was relevant in the solar power generation forecasting for the periods where the sun does not shine. It turns out that some forecasting software nevertheless reports a MAPE for such series, simply by dropping periods with zero actuals [12]. However, this is not a good method, as it implies that we do not care at all about what we forecasted if the actual was zero. That is why we introduced RMSE as well.

3.4.2. Root Mean Square Error

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors) which is widely used in forecasting to assess the accuracy of the model. Residuals mean how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. IT is calculated as the equation (6)

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(A_t - F_t)^2}{A_t}} \quad (6)$$

Where n is the number of samples, A_t is the actual value and F_t is the forecasted value.

4. Result

4.1. Solar forecasting

As shown in the figure 7, The ARIMA model predicts subzero values and thereby overshoots. Moreover, the zero plateau is not predicted as perfect horizontal line but as a line with a slope. This is because of the nature of solar generation which has zero values during the night. This might cause the reason for confusion to the model. Therefore, ARIMA model seems not suitable for model having regular intermittency. In addition, the peaks tend to be also overpredicted .

In contrast the SARIMA model predicts a zero plateau better and only slightly overshoots into subzero in two cases for 1 week period even though it does not predict exact zero value. As shown in the table 4, this is clearer when it comes to RMSE which is only 65% of RMSA of ARIMA value. This improvement must be driven from the consideration of seasonality. The maximum peaks are predicted accurately less overshoots as well.

The LSTM Model does not overshoot into subzero and predicts the plateau. Even though LSTM recognize the plateau, the RMSE value is higher than SARIMA model, this seem to simply improve by increasing Epoch number of LSTM as we discussed in chapter 3.3.2 LSTM with Figure 6. As with the ARIMA model there is the tendency to overshoot the maximum peak value. The tendency to overshoot is reduced when the LSTM is combined with an additional irradiation model but not big effect.

MAPE value is not presented because zero values during the night will cause infinite value on MAPE analysis.

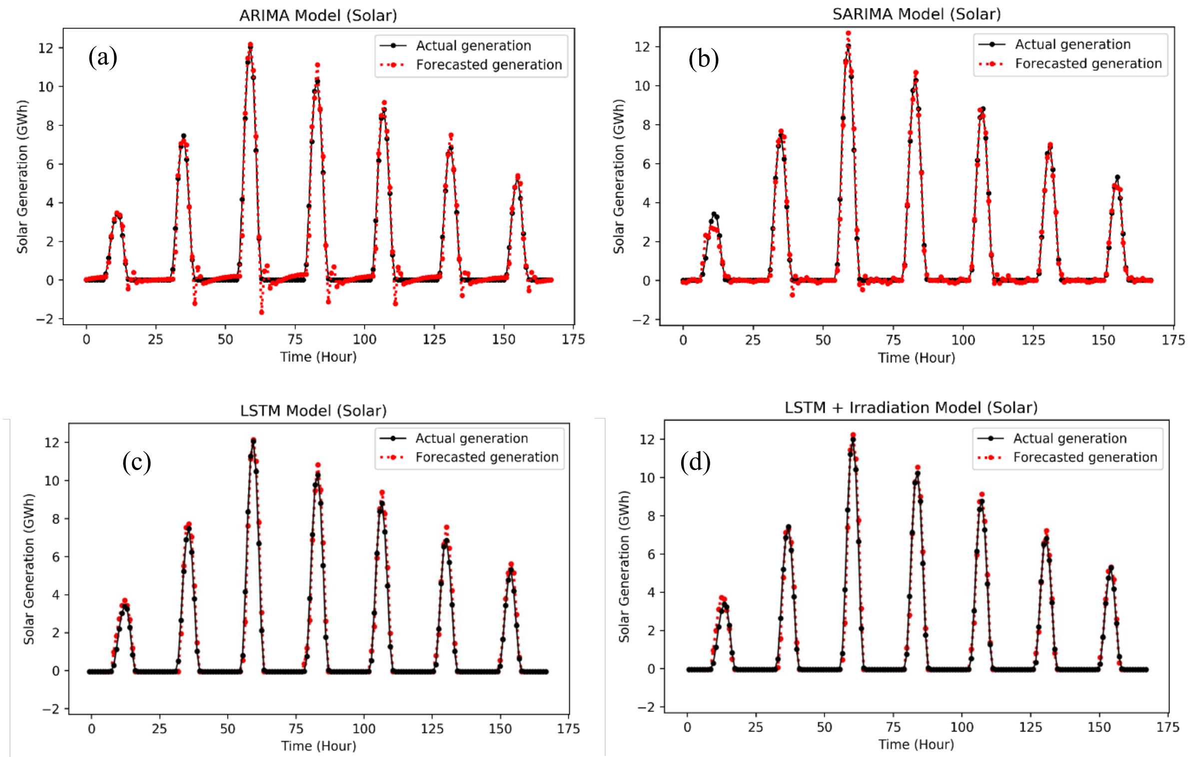

Figure 7 | Solar forecasting result for (a) ARIMA, (b) SARIMA, (c) LSTM and (d) LSTM with irradiation data

Table. 4 | Comparison of result for solar forecasting

	ARIMA	SARIMA	LSTM	LSTM with irradiation data
RMSE	434.30	282.19	343.95	336.05

4.2. Wind forecasting

Comparison between models are represented in Figure 8 and Table 4. In general, ARIMA and SARIMA models are superior to LSTM models showing very good prediction over the period. Among ARIMA and SARIMA, ARIMA shows slightly lower errors both in RMSE and MAPE. Although the difference of RMSE between ARIMA and SARIMA by around 1%, MAP values between those differ by around 10%.

For LSTM, despite of longer training period comparing to ARIMA and SARIMA models, the error came out around 60% higher in MAPE value. However, this could be derived from comparably low number of Epoch we defined because of limitation for computation time. Again, more Epochs will result higher prediction result. The model using additional wind speed data improved the prediction in comparison to the LSTM.

Between LSTM models, univariate LSTM model underestimates in the beginning while multivariate LSTM with wind speed slightly overestimate in the same period. When there is a local minimum, it slightly overshoot in multivariate LSTM. Nonetheless, those are very slight difference.

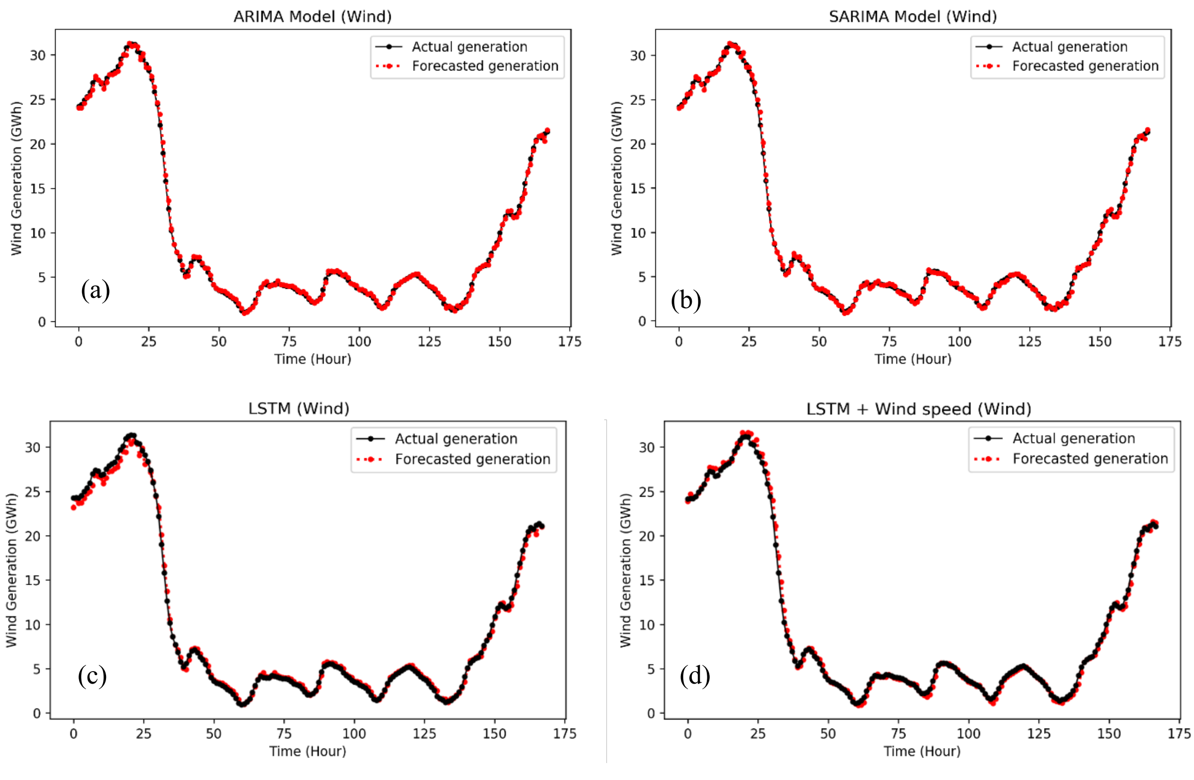

Figure 8 | Wind forecasting result for (a) ARIMA, (b) SARIMA, (c) LSTM and (d) LSTM with wind speed data

Table. 4 | Comparison of result for wind forecasting

	ARIMA	SARIMA	LSTM	LSTM with wind speed data
RMSE	339.59	343.29	483.07	448.20
MAPE	4.33	4.74	7.22	5.06

5. Conclusion

Due to the increasing penetration of renewable energy resources to the grid, the balance of supply and demand must be maintained. Therefore, time series forecasting models must be developed to improve the forecasting accuracy to help decision makers optimize their decisions.

ARIMA forecasting was introduced and showed a good accuracy for short term prediction. Another variation of ARIMA was introduced known as SARIMA capable of capturing the seasonal trends of the timeseries increasing the accuracy of the model. However, the former methodologies are univariate in nature, as well as computationally taxing and time consuming. In addition, since weather data is known for its intermittency, long term forecasting with these models are not recommended.

Long Short Term Memory Recurrent Neural Networks were introduced to make up for the aforementioned weaknesses, being multivariate time series models, with high accuracy and ability to compute large quantity of data with ease. Moreover, being not as time consuming as the former models. Future enhancements can be made applying a Bidirectional LSTM model which increases the accuracy of forecasting even more.

Reference

1. **Burger, Bruno.** *Public Net Electricity Generation in Germany.* Freiburg, Germany : Fraunhofer Institute for Solar Energy Systems ISE, 15 January 2020.

2. *A Weather-Based Hybrid Method for 1-Day Ahead Hourly Forecasting of PV Power Output.* **Hong-Tzer Yang, Chao-Ming Huang, Yann-Chang Huang, and Yi-Shiang Pai.** s.l. : IEEE Transactions on Sustainable Energy, 5(3), 917–926, 2014.

3. *Comparison of ARIMA and ANN models used in electricity price forecasting for power market.* **Gao Gao, Kwoklun Lo, Fulin Fan.** 2017.

4. **George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung.** *Time Series Analysis: Forecasting and Control, 4th edition.* s.l. : Wiley, Oxford, 2008.

5. **Hyndman, R.J., & Athanasopoulos G.** *Forecasting: principles and practice.* s.l. : OTexts, 2nd edition, 2018.

6. **Weidlich, Anke.** *Operations Research for Energy Systems.* lecture note 8 and 9 : INATECH, Albert-Ludwigs-Universität Freiburg, 2020.

7. **Haykin, S.S.** *Neural Networks: A Comprehensive Foundation.* s.l. : Prentice Hall, 1999.

8. **Beaufays, Haşim Sak and Andrew Senior and Françoise.** Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. 2014.

9. *A Novel Connectionist System for Unconstrained Handwriting Recognition.* **Schmidhuber, A. Graves and M. Liwicki and S. Fernández and R. Bertolami and H. Bunke and J.** 2009, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 855-868.

10. **Schmidhuber, A. Graves and J.** Framewise phoneme classification with bidirectional LSTM networks. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* 2005, pp. 2047-2052 vol. 4.

11. **Open Power System Data. [Cited: 14 Feb 2020.] https://data.open-power-system-data.org/.**

12. Jim Hoover, *Measuring Forecast Accuracy: Omissions in Today's Forecasting Engines and Demand-Planning Software.* International Institute of Forecasters, 20016, issue 4, pp. 32-35

Division of tasks

Both students work on together (Teamwork!)