National Institute of Technology Karnataka, Surathkal

Department of Electrical and Electronics Engineering

Time Series Solar Radiation Forecast using ANN and ARIMA

Presented by:
Anish Bekalkar - 171EE106
Rohan Garg - 171EE136
Skanda Upadhyaya -171EE143
Chaitali Shah - 171EE212
Alagu Parvathy - 171EE251

Contents

1 Introduction **2**

2 Artificial Neural Networks (ANN) **3**

2.1 MinMax Scaling 3

2.2 LSTM-Formula 4

2.3 Mean Squared Error 5

2.4 Results 5

3 Autoregressive Integrated Moving Average Model(ARIMA) **7**

3.1 Block Diagram 8

3.2 Results 9

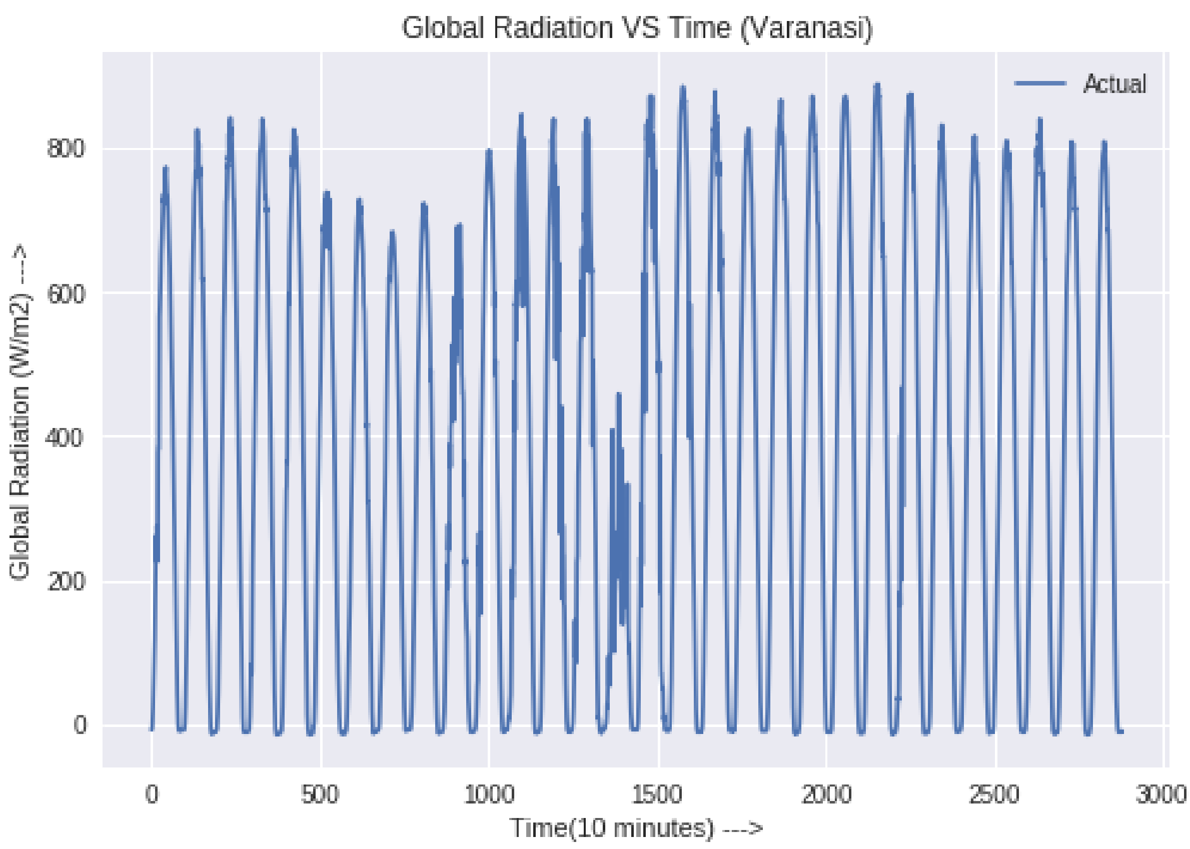
4 Conclusions **11**

5 References **11**

1 Introduction

The global shift in paradigm pertaining to energy generation, accelerated by depletion of conventional sources of energy like fossil fuels, has led to extensive research in the realm of renewable energy. There has been a rapid rise in efforts to increase sustainability of solar energy in the past decade. Solar Radiation forecasting using existing time series data is an essential part of these efforts. A time series is a collection of time ordered observations x_t, each one being recorded at a specific time t (period). Prediction of solar energy helps in better positioning of solar panels and related equipments and also, provides a fair idea of their sustainability. It has many other advantages such as prediction of weather conditions for agricultural and domestic utilisation.

Among the most used models for time series forecast are ARIMA and ANN. The project aims at developing sufficiently accurate models for prediction of global energy radiation. Models were constructed to predict the amount of solar radiation that will make contact with the surface of the earth in a given location ten minutes into the future. Global solar radiation was predicted by using measurements of weather conditions collected at Varanasi, India for the month of May 2015, as inputs. The global radiation was measured at 10 minute intervals from 1 May to 30 May.



Graph 1: Measured Global Radiation

2 Artificial Neural Networks (ANN)

Artificial neural network (ANN) is a computational model based on the structure and function of biological neural networks, with the capacity to learn and model phenomenon to predict results. The ANN structure comprises of multiple layers with various number of neurons in each layer. The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an output layer. There may be multiple intermediate layers in between known as the hidden layers.

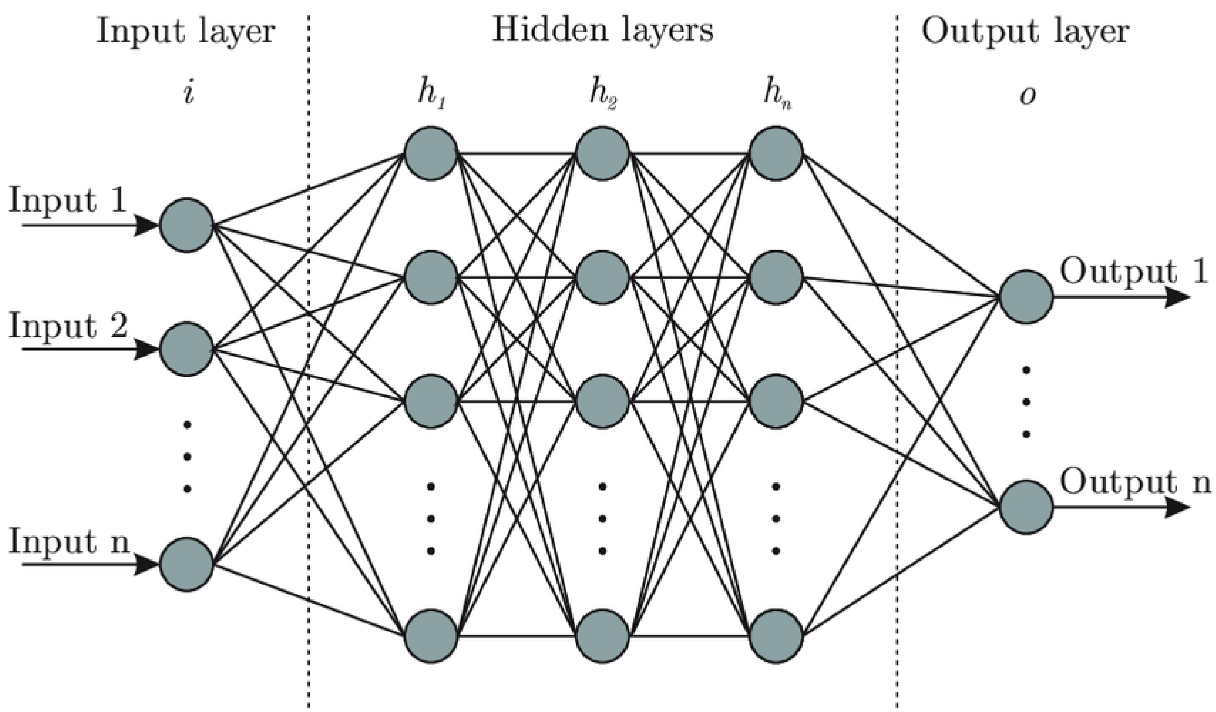


Figure 1: ANN Architecture

Layered feed-forward ANNs use the back-propagation algorithm. The signals are sent to forward in the hierarchy of neurons and the errors are propagated backwards. The activation function of the artificial neurons in ANNs implementing the back-propagation algorithm is a function of weighted sum.

The back-propagation algorithm calculates the dependency of error on output, inputs, and weights and focuses on minimizing error. Weights are adjusted using the gradient descent method.

2.1 MinMax Scaling

Before the input can be fed to the model for training it has to be made suitable for the model. After arranging the data in appropriately times series, it is normalised. An alternative approach to Z-score normalization (or standardization) is the so-called Min-Max scaling.

A Min-Max scaling is typically done via the following equation:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

2.2 LSTM-Formula

Long Short-Term Memory (LSTM) unit RNN is among the most widely used models in Deep Learning for NLP today and has been found to be the most suitable approach to time series forecast under study.

LSTM calculates a hidden state $\mathrm{h_s}$:

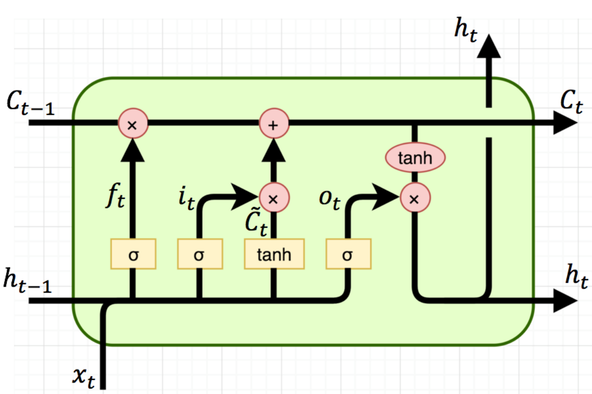
$$\begin{aligned} i_t &= \sigma(x_t U^i + h_{t-1} W^i) \\ f_t &= \sigma(x_t U^f + h_{t-1} W^f) \\ o_t &= \sigma(x_t U^o + h_{t-1} W^o) \\ \overline{C_t} &= tanh(x_t U^g + h_{t-1} W^g) \\ C_t &= \sigma(f_t * C_{t-1} + i_t \overline{C_t}) \\ h_t &= tanh(C_t) * o_t \end{aligned}$$

Denote * as element wise multiplication

Here, **i**, **f**, **o** are called the input, forget and output gates, respectively.
W is the recurrent connection at the previous hidden layer and current hidden layer, **U** is the weight matrix connecting the inputs to the current hidden layer.

$$\bar{c}$$

is a "candidate" hidden state that is computed based on the current input and the previous hidden state.C is the internal memory of the unit.
$\mathbf{h_t}$ is output hidden state, computed by multiplying the memory with the output gate.



(a) Long Short-Term Memory

Figure 2: LSTM model

2.3 Mean Squared Error

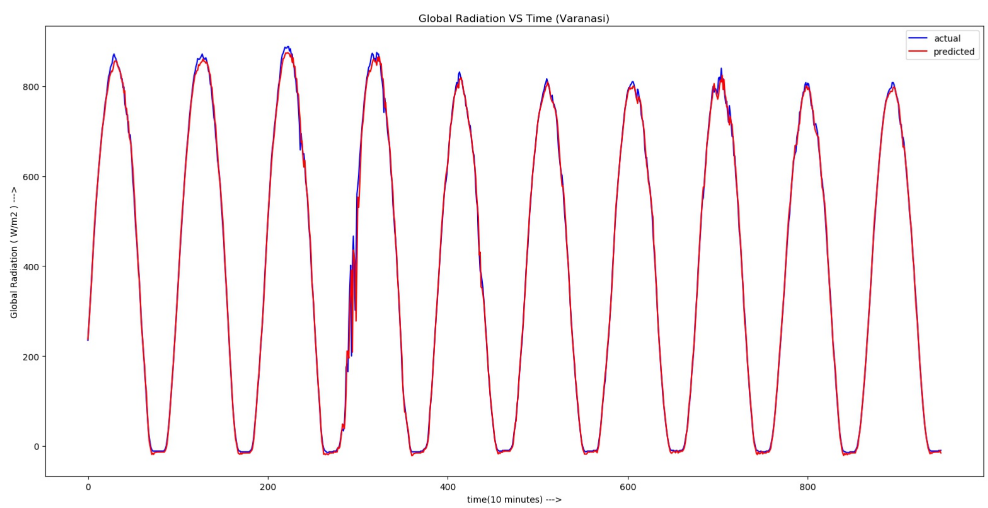
In statistics, the mean squared error (MSE) of an estimator measures the average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss.
The mathematical equation that will give us the mean squared error for all the points.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y_i})^2$$

2.4 Results

On rigorous evaluation done by continuous variation of parameters of the most accurate results were found with Delay = 5; First layer = tanh - 20 neurons; Second layer linear - 20 neurons configuration. The principal parameters which influence the complexity of the network and its learning are the number of inputs, the number of hidden layer and their number of neurons, the activation (or transfer) function, the learning algorithm and the comparison function used during the learning phase.

The most accurate forecast is shown in the graph below.



Graph 2: Forecast results using ANN

It was found that with an increase in the number of epochs the accuracy of the model increased. Unfortunately, it also causes a rise in the computation time required. A healthy compromise between accuracy and time taken was reached at 500 epochs and all the following iterations have been run for 500 epochs. Similarly the optimum batch size for the model was chosen to be 16, and other parameters were varied to obtain the results shown in the table below. Accuracy also improves with delay. The relationship between number of layers, neurons and their activation functions is not linear and hence optimisation is done by manually varying until best results are found.

Delay	Activation Function		MSE(loss function)	Number of hidden layers	Number of Neurons		RMSE Predictions	
	First Layer	Second Layer			Layer 1	Layer 2	Percentage	W/m2
1	tanh	tanh	0.0063	2	20	20	5.743078592	25.9395721
1			0.0064	2	40	40	5.801491663	26.20340381
1			0.0064	2	60	60	5.778254553	26.09844881
1		linear	0.0063	2	20	20	5.749202655	25.9672335
1			0.0064	2	40	40	5.800485735	26.19886112
1			0.0064	2	60	60	5.766636811	26.04597454
2	tanh	tanh	0.0054	2	20	20	4.856178625	21.93373915
2			0.005	2	40	40	4.360166755	19.69342102
2			0.005	2	60	60	4.345872507	19.62885492
2		linear	0.0055	2	20	20	5.015354389	22.65268274
2			0.0055	2	40	40	4.886491557	22.07065466
2			0.0055	2	60	60	5.185825308	23.42264218
3	tanh	tanh	0.0044	2	20	20	4.239807771	19.14979571
3			0.0042	2	40	40	3.925030672	17.72805465
3			0.004	2	60	60	4.159513843	18.78713594
3		linear	0.0048	2	20	20	4.610147494	20.82249684
3			0.0046	2	40	40	4.126730349	18.63906514
3			0.0042	2	60	60	4.229707611	19.10417817
4	tanh	tanh	0.004	2	20	20	3.879917975	17.524296
4			0.0032	2	40	40	3.98204455	17.98556735
4			0.0026	2	60	60	4.159414912	18.78669161
4		linear	0.0042	2	20	20	3.938292646	17.78795567
4			0.0034	2	40	40	4.089078965	18.46900558
4			0.003	2	60	60	4.377020374	19.76954419
5	tanh	tanh	0.0033	2	20	20	3.73262929	16.85904124
5			0.0019	2	40	40	5.096927367	23.02112224
5			0.0014	2	60	60	4.371208391	19.74329137
5		linear	0.0037	2	20	20	3.63866838	16.43465101
5			0.0028	2	40	40	4.83121667	21.82099619
5			0.0021	2	60	60	4.504410188	20.34491943

Results obtained from ANN model

3 Autoregressive Integrated Moving Average Model(ARIMA)

An ARIMA model is a class of statistical models for analyzing and forecasting time series data. ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. Briefly, they are:

AR: Autoregression
A model that uses the dependent relationship between an observation and some number of lagged observations.

I: Integrated
The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

MA: Moving Average
A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA(p,d,q) where the parameters are substituted with integer values to indicate the specific ARIMA model being used.

The parameters of the ARIMA model are defined as follows:
p: The number of lag observations included in the model, also called the lag order.
d: The number of times that the raw observations are differenced, also called the degree of differencing.
q: The size of the moving average window, also called the order of moving average.

A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing in order to make it stationary, i.e. to remove trend and seasonal structures that negatively affect the regression model
.
A value of 0 can be used for a parameter, which indicates to not use that element of the model. This way, the ARIMA model can be configured to perform the function of an ARMA model, and even a simple AR, I, or MA model.

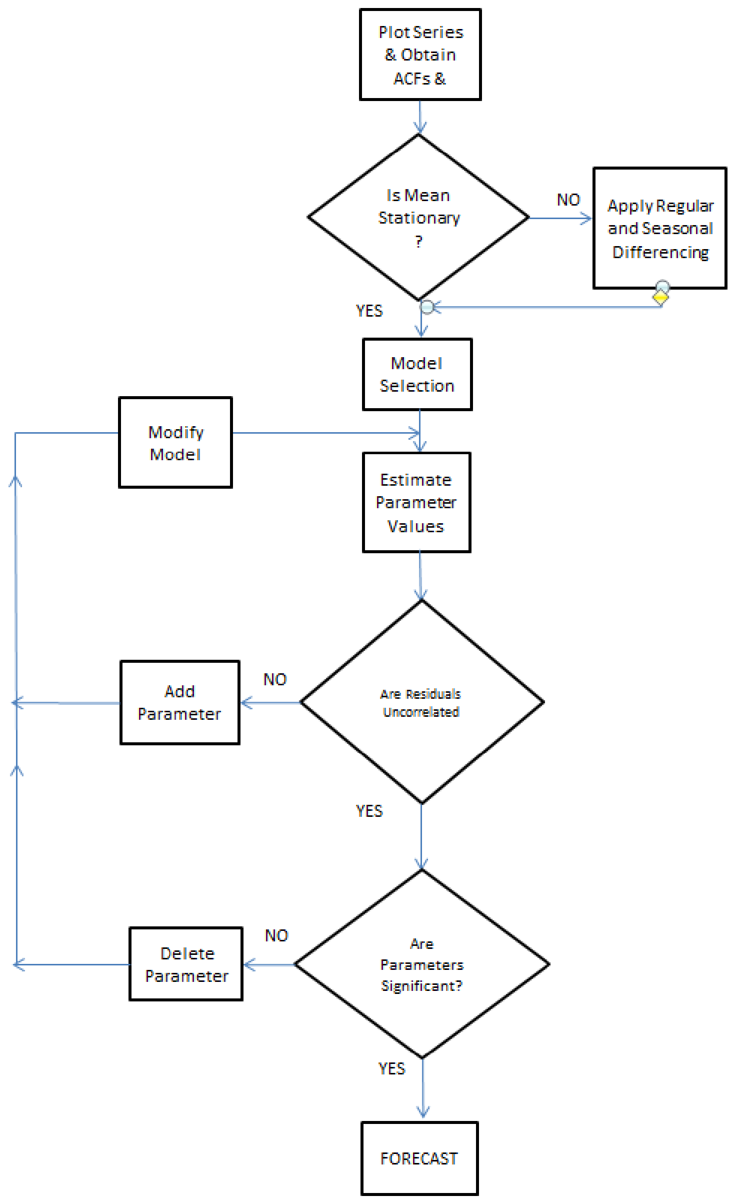
The forecasting equation is constructed as follows. First, let y denote the d_{th} difference of Y, which means:

If d=0, $y_t = Y_t$
If d=1, $y_t = Y_t - Y_{t-1}$

In terms of y, the general forecasting equation is:

$$\hat{y}_t = \mu + \phi_t y_{t-1} + ... + \phi_p y_{t-p} - \theta_1 e_{t-1} - .... - \theta_p e_{t-p}$$

3.1 Block Diagram



Working of ARIMA model

3.2 Results

This is a process that uses time series analysis and diagnostics to discover good parameters for the ARIMA model.

In summary, the steps of this process are as follows:

1.**Model Identification**: Use plots and summary statistics to identify trends, seasonality, and autoregression elements to get an idea of the amount of differencing and the size of the lag that will be required.

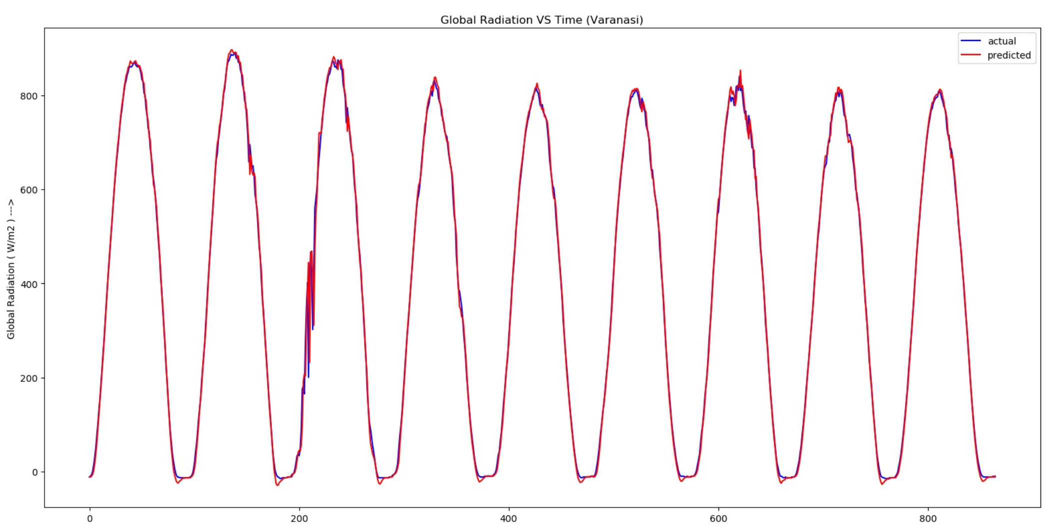
2.**Parameter Estimation**: Use a fitting procedure to find the coefficients of the regression model.

3.**Model Checking**: Use plots and statistical tests of the residual errors to determine the amount and type of temporal structure not captured by the model.

The process is repeated until either a desirable level of fit is achieved on the in-sample or out-of-sample observations (e.g. training or test datasets).

The best results were obtained for the condition where p=4, d=2 and q=0.

The most accurate forecast is shown in the graph below.



Graph 3: Forecast results using ARIMA

The various results obtained by varying the values of p, q, d have been recorded in the table given below along with the corresponding RMSE(W/m^2).

P	D	Q	RMSE (W/m^2)
1	0	0	26.00250998
2	0	0	23.11534954
2	1	1	18.56921139
3	0	0	21.78007093
3	1	1	17.90292072
4	0	0	19.75514437
4	1	0	17.73216355
4	1	1	17.61780457
4	2	0	16.8108336
4	2	1	17.95959486
5	0	0	17.21542207
5	0	1	DNC
5	0	2	DNC
5	1	0	17.36305125
5	1	1	17.87746984
5	1	2	DNC
5	2	0	17.28802832
6	0	0	16.57010753
7	0	0	16.62997106

Results obtained from ARIMA model

4 Conclusions

Aritificial Neural Networks are better equipped to account for the non linearity of models as compared to ARIMA. As solar radiation is influenced by a number of unpredictable factors like environmental and atmospheric conditions, models like ANN that account for non linear modelling parameters are more sought for. Also, artificial neural networks (ANNs) have the advantage of accepting multiple data fields as input, rather than being limited to univariate input. The minimum RMSE in case of ARIMA is 16.8108336 W/m^2 and for ANN is 16.43465101 W/m^2. ANN has a better overall prediction but both the model can ameliorated by suitably changing modelling parameters.

5 References

1. 'Forecasting of preprocessed daily solar radiation time series using neural networks' - Christophe Paoli, Cyril Voyant, Marc Muselli, Marie-Laure Nivet
2. 'Artificial neural network forecast of monthly mean daily global solar radiation of selected locations based on time series and month number' - Chigbogu Godwin Ozoegwu
3. 'Solar Radiation Time Series Prediction' - Cameron Hamilton, Walter Potter, Gerrit Hoogenboom, Ronald McClendon, Will Hobbs
4. 'How to Create an ARIMA Model for Time Series Forecasting in Python'- by Jason Brownlee in Time Series
5. 'Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks'
6. 'Build or Make your own ARIMA forecasting model?'- Posted by Tom Reilly
7. https://isaacchanghau.github.io/post/lstm-gru-formula/