Intra-Hour Time Structural Solar Prediction with RNN

Member: 106033233 周聖諺, 106000122 林奕馨

Project Description

Solar Power is a fluctuating energy. This variability can do harm to the grid with existing electricity. If we can predict the immediate power, not only can solar power plant utilize battery beforehand to balance the fluctuation of Solar power, but also the utility management system (like Taipower Company) can correctly schedule spinning reserves and demand response.

There are currently different time span of prediction, intra hour, intra day, and day ahead, in which the forecast skill decrease when time span decrease.Thus, our group aims to maximize the accuracy of prediction of intra hour and we select forecasting horizons of 1 min, 5 min, 15 min, 60min.

In intra hour forecast, there are two dominant input mainly use in current thesis:

1. Sky Images
2. Meteorological records:
    a. ghi/dni solar irradiance
    b. solar elevation
    c. weather data

The most popular model used is ANN and there are few literature use RNN to compare over ANN. Thus, our team design experiments to achieve two goals, one is to analyze the performance of RNN and pick Linear model and ANN as our comparison models. sky image and Meteorological input to predict solar irradiance. Then,we will do some comparison and discussion on which model performs better in terms of accuracy.

Dataset Source

In this research, we use the “comprehensive dataset for the accelerated development and benchmarking of solar forecasting methods” dataset provided by Hugo T. C. Pedroa, David P. Larsona, and Carlos F. M. Coimbra. Below is the link of the possible dataset. https://zenodo.org/record/2826939#.XfM9uWQzZPZ

The data consist of three years (2014–2016) of quality-controlled, 1-min resolution global horizontal irradiance(ghi) and direct normal irradiance(dni) ground measurements in California.They also provide overlapping data from commonly used exogenous variables, including sky images, satellite imagery, Numerical Weather Prediction forecasts, and weather data.

We take the irradiance, weather data, and sky image as our input.

Experiment 1:

In this experiment, we take current weather data and irradiance as input to predict future irradiance.

For weather input, we take in ambient temperature,relative humidity, pressure, wind speed, wind direction, maximum wind speed, and precipitation. For irradiance, we take current GHI, DNI, DHI as input to predict future irradiance in GHI.

An Example of Input Data:

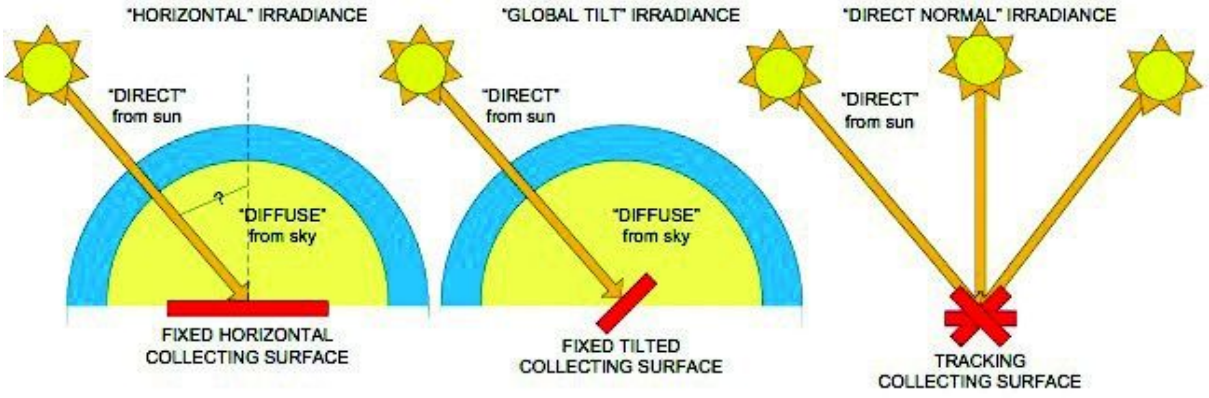
timeStamp	ghi	dni	dhi	air_temp	relhum	press	windsp	winddir	max_windsp	precipitation
1/2/2014 8:00	0	0	0	7.32	56.56	1010	1.8	43.61	3.7	0

a. ghi: Global Horizontal Irradiance (GHI) is the total amount of shortwave radiation received from above by a surface horizontal to the ground. This value is of particular interest to photovoltaic installations and includes both Direct Normal Irradiance (DNI) and Diffuse Horizontal Irradiance (DHI).
GHI = Direct Normal (DNI) X cos(θ) + Diffuse Horizontal (DHI)

b. dni: Direct Normal Irradiance (DNI) is the amount of solar radiation received per unit area by a surface that is always held perpendicular (or normal) to the rays that come in a straight line from the direction of the sun at its current position in the sky.

c. dhi: Diffuse Horizontal Irradiance (DHI) is the amount of radiation received per unit area by a surface (not subject to any shade or shadow) that does not arrive on a direct path from the sun, but has been scattered by molecules and particles in the atmosphere and comes equally from all directions

d. air_temp: Temperature in air
e. relhum: Relative Humidity in percetage %
f. press: Atomosphere Pressure in hPa
g. windsp: Wind Speed
h. windir: direction of the source of the wind
i. max_windsp: Maxium wind speed during a period.
j. precipitation: Precipitation during a period



Experiment 2:

For sky image experiment, we take in ghi, dni, clear sky irradiance, clear sky index, and sky image features.

timestamp	ghi_5min	dni_5min	ghi_clear_5min	dni_clear_5min	ghi_kt_5min	dni_kt_5min	elevation_5min	AVG(R)	STD(R)	ENT(R)
2014-01-02 16:00:00	74.342	507.16	45.481545	236.113747	1.2	1.2	5.821237	124.20426	40.03870	5.62222
2014-01-02 16:05:00	89.860	559.84	56.989441	277.436445	1.2	1.2	6.611024	127.85084	38.97730	5.61076
2014-01-02 16:10:00	105.300	599.62	69.080774	316.517024	1.2	1.2	7.392244	138.77230	36.65666	5.59870
2014-01-02 16:15:00	118.220	614.02	81.599285	353.096042	1.2	1.2	8.164625	121.77990	39.78908	5.60416
2014-01-02 16:20:00	129.880	619.94	94.419980	387.116288	1.2	1.2	8.927891	128.31020	39.77270	5.57416

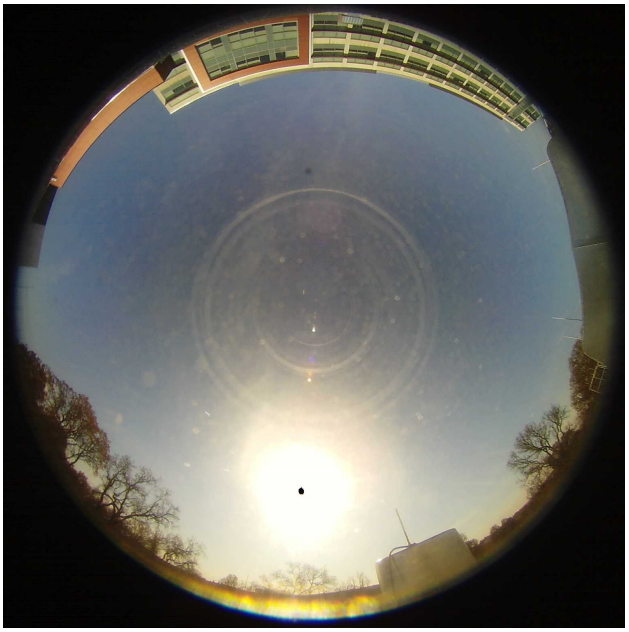


fig: Original Sky Image

timestamp	ghi_5 min	dni_5 min	ghi_clear_ 5min	dni_clear_ 5min	ghi_kt_5 min	dni_kt_5 min	elevation_5 min
1/2/2014 16:00	74.342	507.1 6	45.481545	236.1137 47	1.2	1.2	5.821237

AVG(R)	STD(R)	ENT(R)	AVG(G)	STD(G)	ENT(G)	AVG(B)	STD(B)
124.204 3	40.0387	5.62222	144.587 8	33.0362	5.5488	157.200 7	26.0946 8

ENT(B)	AVG(RB)	STD(RB)	ENT(RB)	AVG(NRB)	STD(NRB)	ENT(NRB)
5.27436	0.77746	0.14316	4.7065	-0.13226	0.08764	4.07194

Figure1. Input data format (left 7 columns are meteological records, right 3 columns are the sky image features)

1. Irradiance and elevation:
   For irradiance, we take in ghi and dni as described in the previous sections. Also, elevation is the measured height of sun.
2. Clear-sky index and irradiance:
   Clear sky index is defined as kt = I/Ics, where I denotes GHI or DNI, and Ics is the respective clear-sky irradiance, which means the predicted irradiance if there's no cloud in the sky at that time.
3. RGB sky images features:
   The color data extracted from the sky-dome pixels are flattened into r, g, b values. Among which, red to blue ratio and the normalized red to blue ratio are derived, since it indicates the cloudiness of the sky.
   Then, three features are calculated with these five vectors, which are average, standard deviation, and entropy in every image taken, yielding a total 15 features. The formula is shown in figure
   a. Average:
   b. Standard deviation: it determines how much varied this vectors in the picture.
   c. Entropy: *Entropy* is a probability-based measure used to calculate the amount of uncertainty.

- Average

$$\mu = \frac{1}{N}\sum_{i=1}^{N} v_i.$$

- Standard deviation

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i - \mu)^2},$$

- and entropy

$$e = -\sum_{\substack{i=1 \\ p_i \neq 0}}^{N_B} p_i \log_2(p_i),$$

Figure 2. Sky image features formula

where vi represents one of the five vectors, N is the number of elements in the vector, and pi is the relative frequency for the ith bin out of NB = 100 bins evenly spaced.

Data Preprossesing

We first split the 3 years data into training and testing dataset. We make the data that from 2014~2015 as training set, and 2016 for testing set. Then, we clear the data by dropping off those rows with zero irradiance, since it means that the time of the measurement is at night, which is not our prediction target.

Our prediction targets (train_label & test_label) are the global horizontal irradiance(ghi). Since we need to predict for different time span, we need to shift the y label by the time we want to predict. For instance, if we want to predict for irradiance 5 min later, we need to shift the y label 1 space (1 unit = 5 min)

Algorithm

For algorithm, we choose three different methods for comparison, Lasso Regression, ANN, and RNN.

Lasso regression is a linear regression that can shrink the data values to a central mean by using L1 regularization, which is by adding penalty equals to absolute value of the coefficients to the cost function. Its goal is to minimize

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

When lambda = 0, no parameters will be eliminate, when lambda increase, the importance of cost function to weight will decrease.
The input parameter we give for lasso is 10 times cross validation(cv=10) and 10000 maximum iterations time(max_iter=10000) to prevent non convergence.

For **Artificial Neural Network (ANN)**, it is the most popular method mamy literature uses. We added three dense layer and on output layer for training. The first two are with 64 neurons and the last two are to converge back to the original data size.

For **Recurrent Neural Network (RNN)**, the reason why we want to choose this model is because it is specialized in analyzing data in time series. No matter input data of weather records, sky image or irradiance features, they are all in a time sequence. However, we didn't see much literature discussing on comparison of RNN predictions with the current popular method, ANN. Thus, we decide to give it a try.

RNN take previous hidden layer in addition to original data input as total input for current prediction, which is some sort of memorizing previous data. An important method to reduce gradient vanish or explode problem of RNN is Long short-term memory(LSTM), which can be used to forget some weights during the process of training. It requires three dimension input, which are

1. X-Axis: Samples. Each sample is a 2-dimension matrix. X-axis of the matrix(column) is features. Y-axis of the matrix(row) is time steps. We input a matrix every time to predict a sequence of irradiance target.
2. Y-Axis: Time Steps: It's the most important part of RNN. We input past external data and irraiance to
3. Z-Axis: Features. One feature is one observation at a time step.

Following is the data structure of input :
For time span = 5 min, time step = 4, hte input matrix is as following:

SAMPLE1	Feature1	Feature2	Feature3	
Time 16:00	data 1	data 2	data 3	target 16:05
Time 16:01	data 4	data 5	data 6	target 16:06
Time 16:02	data 7	data 8	data 9	target 16:07
Time 16:03	data 10	data 11	data 12	target 16:08

SAMPLE2	Feature1	Feature2	Feature3	
Time 16:01	data 4	data 5	data 6	target 16:06
Time 16:02	data 7	data 8	data 9	target 16:07
Time 16:03	data 10	data 11	data 12	target 16:08
Time 16:04	data 13	data 14	data 15	target 16:09

Thus, we need to fold our data into three dimensions by fold_time_step functions. Next, we build RNN with the same layer structure as ANN. We add LSTM in the first two layers, and use Dropout to probabilistically exclude input and recurrent connections to LSTM units from activation and weight updates. Lastly, we choose relu function as our final activation since solar irradiance is always larger than zero.

Result

Figure A. input: Weather.csv/ Irradiance.csv / Output: ghi
(sample and prediction frequency resolution=1 min)
Result: NRMSE
Train Data: 1028982 samples
Test Data: 522720 samples
time_span = 1, 5, 15, 60
time_step = 5
epoch = 5
mod = 100

Model\Time Span	1 min	5 min	15 min	60 min
RNN: time_steps=5	0.16	0.21	0.26	0.40
ANN t=5	0.37	0.52	0.29	0.42
Linear	0.128	0.24	0.30	0.51

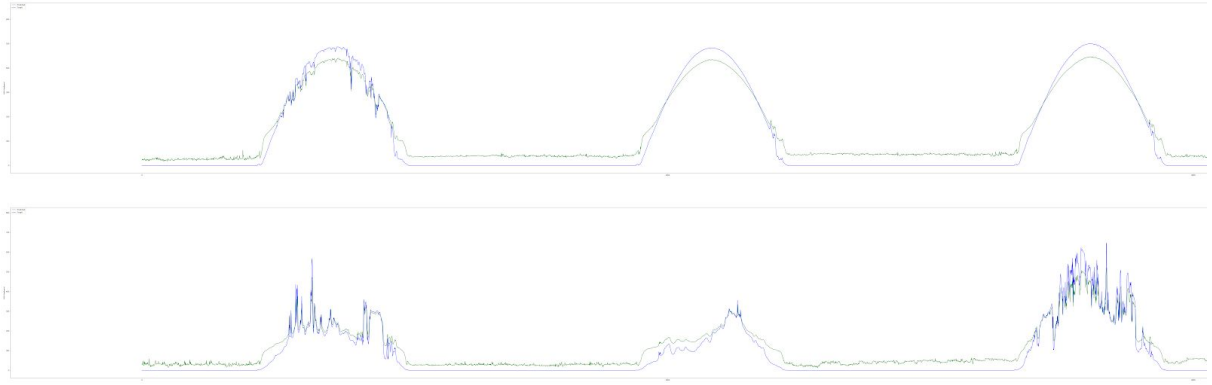
For weather input, we pick time span of 1, 5, 15, 60 min and the prediction frequency is 1/1min.

For the same model, in general, the accuracy increase as time span increase. However, there's an exception for ANN in predicting irradiance 5 min later.

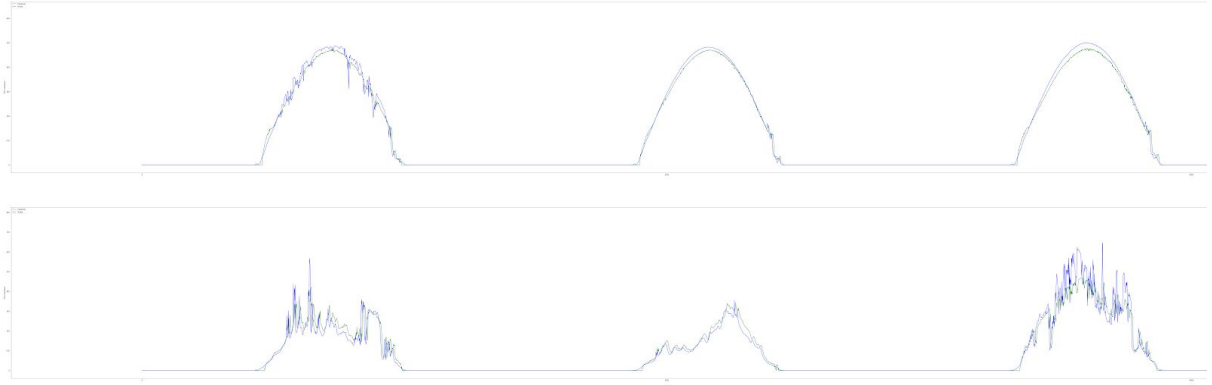
For the same time span, the performance among different models is RNN>ANN>Linear. This is because as we know from measurements, solar irradiance does not distributed linearly. While for ANN, we can train the model through multiple layers and neurons, which increase its accuracy. The reason why error is larger when forecasting irradiance 1 min ahead may because in a shorter run, the variability of solar irradiance is not quite big, so using linear prediction may be more accurate than considering past record data.

As for ANN and RNN

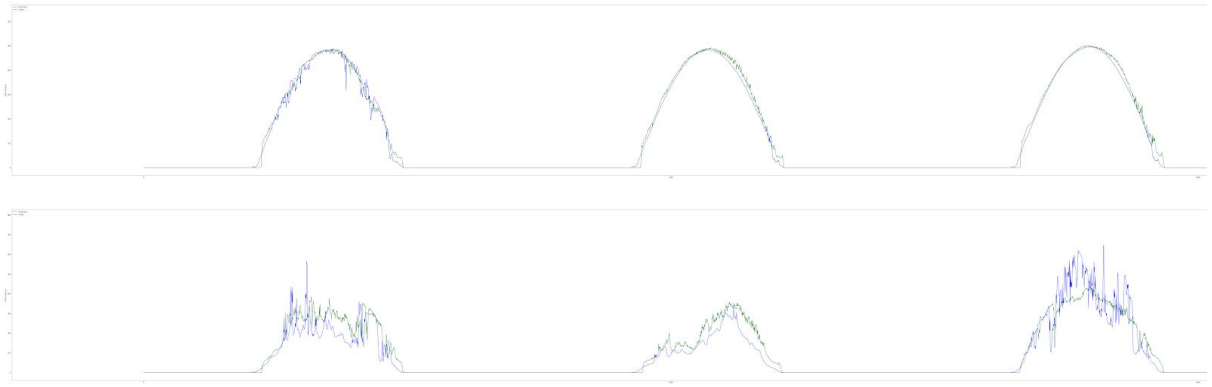
RNN 1min:



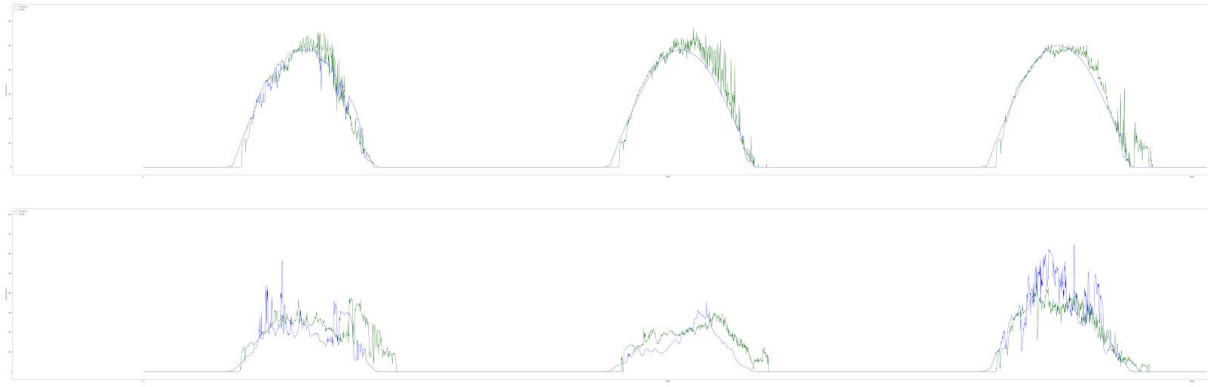
RNN 5min:



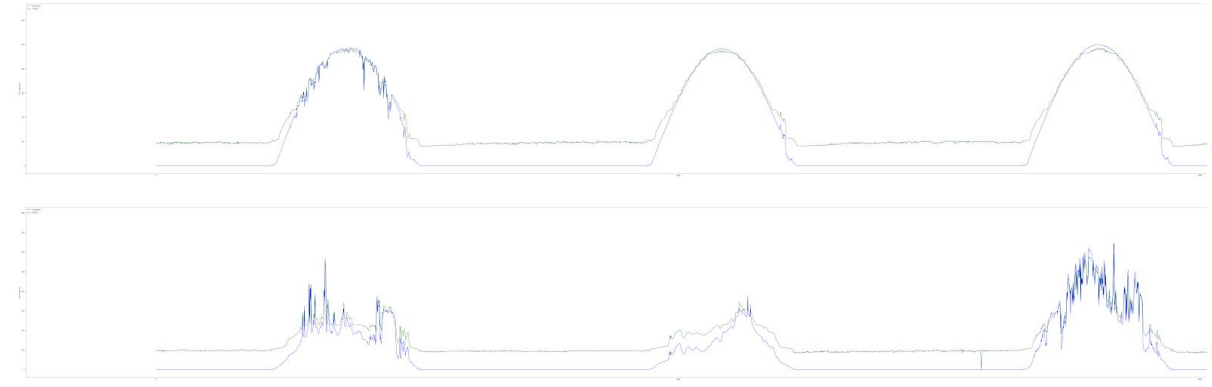
RNN 15min:



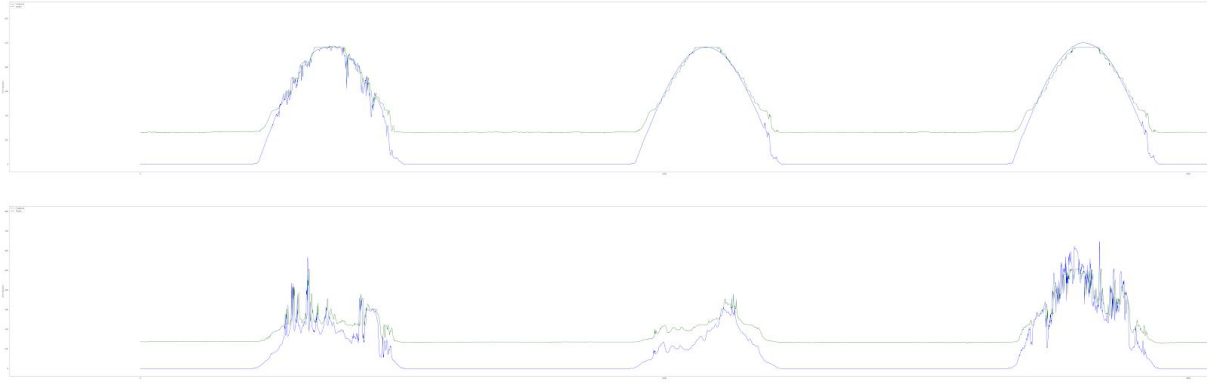
RNN 60min:



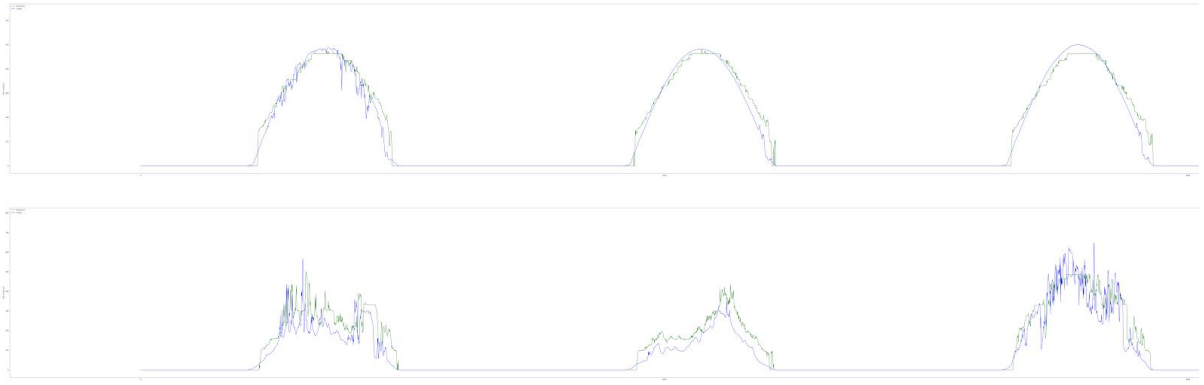
ANN 1min:



ANN 5min:



ANN 15min:



ANN 60min:

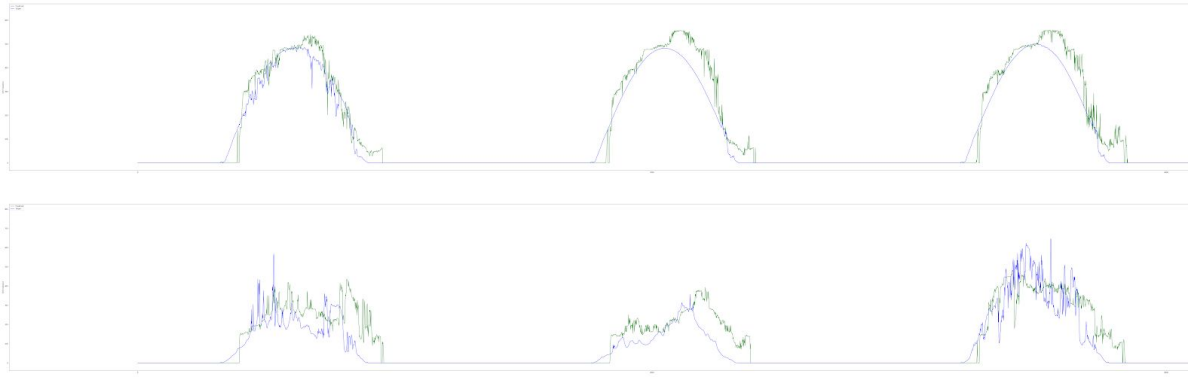


Figure B.input: Irradiance.csv/sky_image_features.csv Output: ghi

Model\Time Span	5 min	15 min	60 min	
RNN: time_steps=5	0.113	0.147	0.208	
ANN	0.113	0.150	0.241	
Linear	0.116	0.165	0.279	

For sky image input of figure B., we only select time span of 5 min, 15 min, and 60 min due to the sample frequency of the original image data is 1/5min.

For the same model, we can see that as time span increase, the nRMSE increase. This is quite intuitive and straightforward. Since if we want to predict something far ahead in the future, the accuracy will decrease if we take in the same input. We can see that the increasing rate of error is Linear>ANN>RNN, which means as time span is larger, the error of linear model increase more rapidly than the other two.

For the same time span in general, RNN outperforms other models. This kind of outperformance is more obvious in a longer time span, especially in 60 mins. For 5 min time span, accuracy of RNN equals to that of ANN, whereas for 60 min time span, RNN is 0.208 and ANN is 0.241, Linear is 0.279. This is because we set the time step as a constant 5, meaning that we always take in the past 5(time steps)*5(min / time steps)=25 min data as

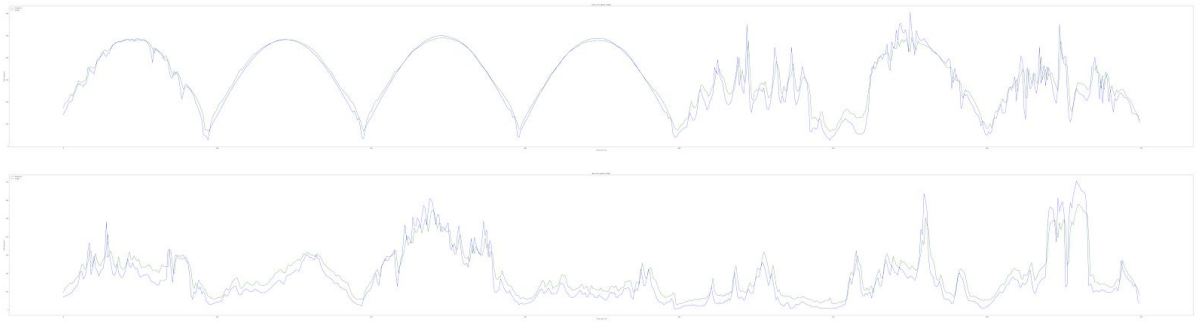
input to predict the next irradiance. In this way, it indicates when we want to predict the irradiance longer time ahead, the importance of considering the input data in a sequence is more significant. Again, linear model performs the worst, since solar irradiance is not linear.

Comparison among Input

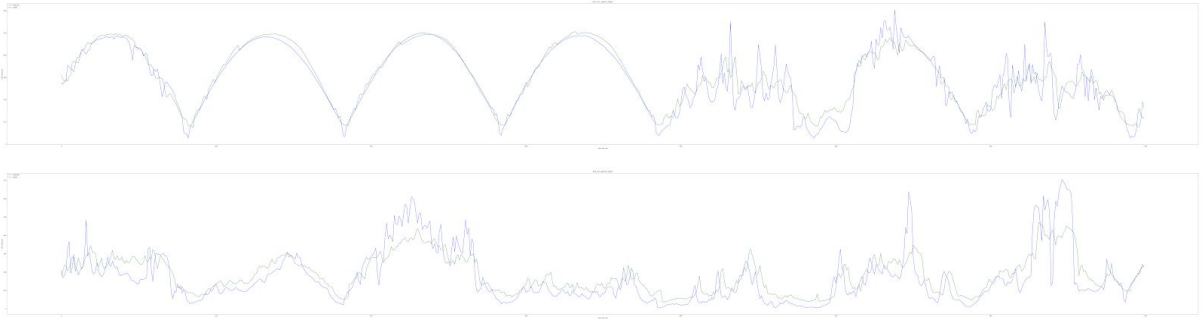
If we compare the input data from figure A. and figure B. together, we can see that the performance is better if we take in sky image than take in weather data. This may because sky image shows a higher more direct correlations with solar irradiance by indicating red to blue ratio, whereas weather data is influenced by other environmental factors simultaneously. However, this judgement can not be 100% confirmed through this experiment, since the sample and prediction frequency of weather data(1 min) and sky image data(5 min) is different.

Also, for weather input, the outperformance of RNN model is more significant than the sky image input. This may indicates that weather data is more time related than sky image, or it may indicates when sample frequency is higher, RNN can indicate more informations. Again this can not be judged unless we do extra experiment.

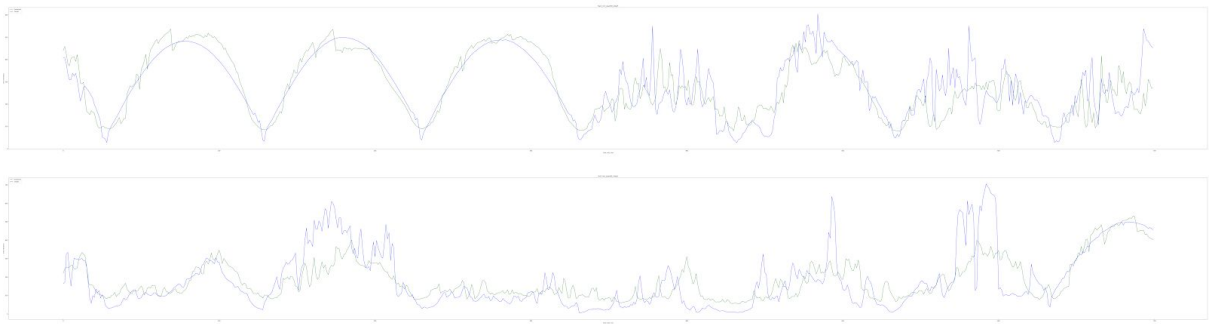
Skyimg RNN 5min:



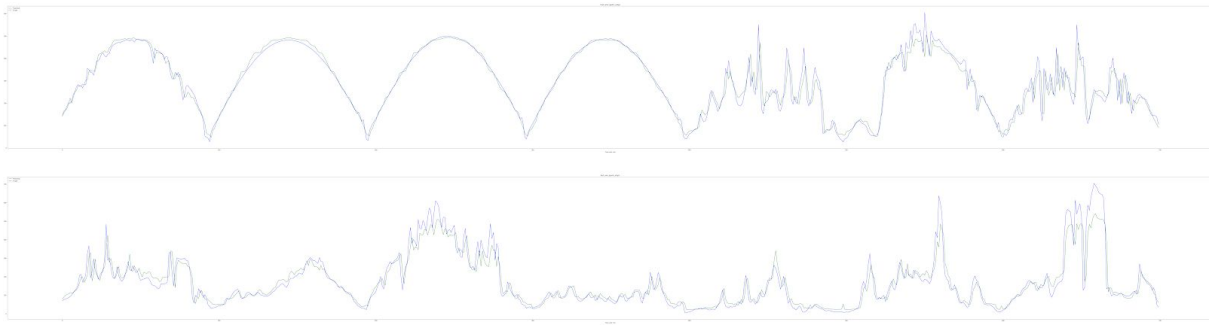
Skyimg RNN 15min:



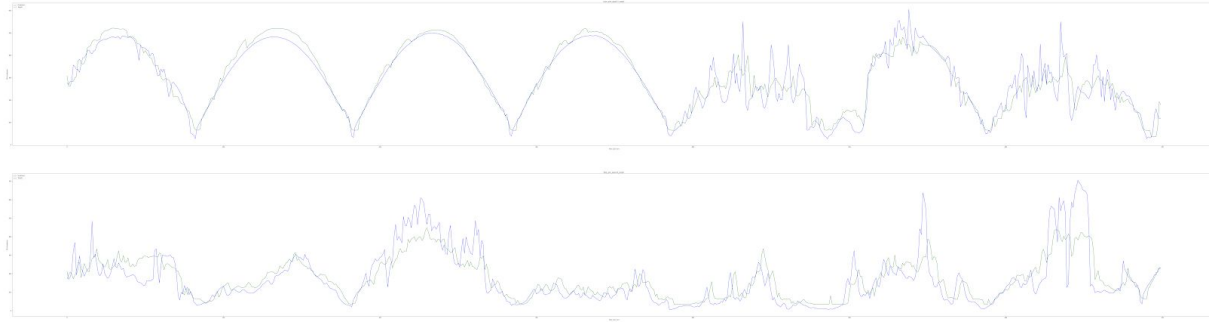
Skyimg RNN 60min:



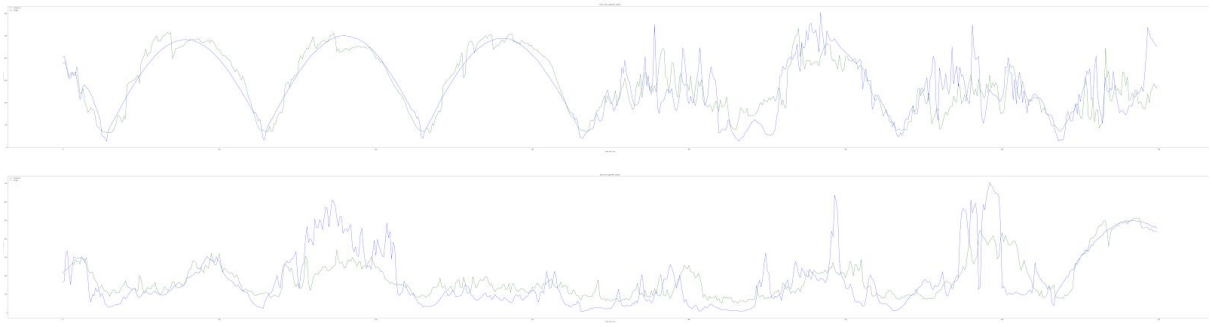
Skyimg ANN 5min:



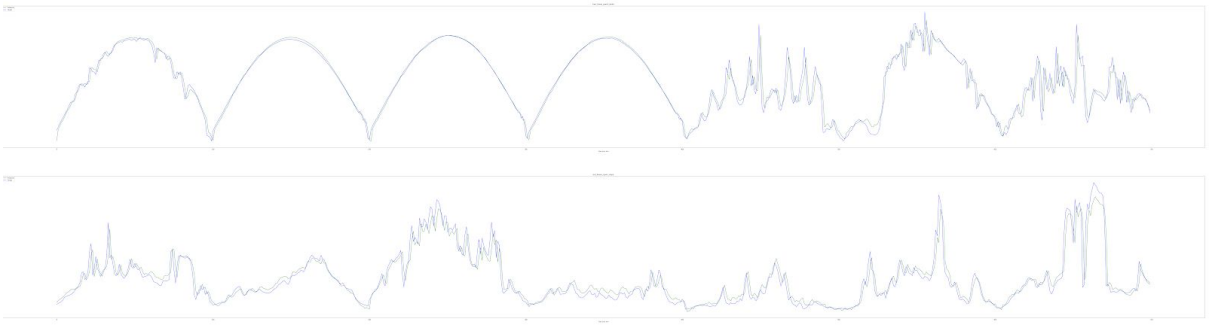
Skyimg ANN 15min:



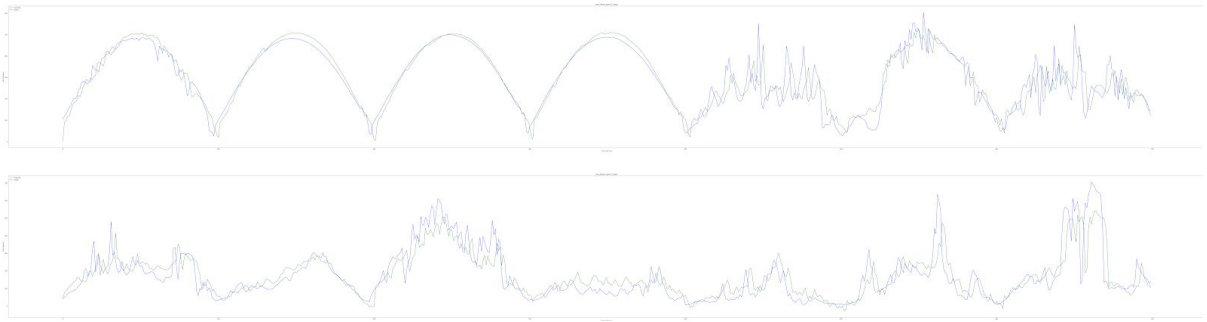
Skyimg ANN 60min:



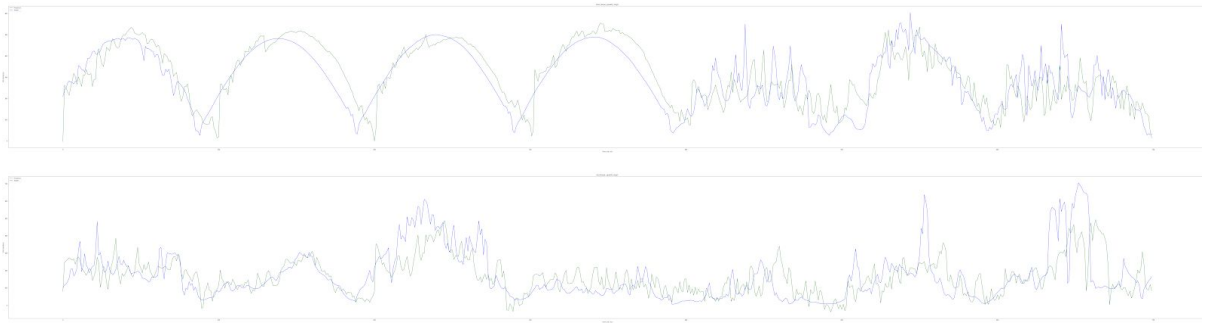
Skyimg Linear 5min:



Skyimg Linear 15min:



Skyimg Linear 60min:



Conclusion & Future Work

In this experiment, we shows that in intra-hour forecasts, ANN or RNN model will be better for forecasting than linear model. While for longer time span forecasting RNN outperforms other model the best. For instance, for 60 min forecasting horizon, RNN decrease the nRMSE by 7% than the linear model.

The future work may be to average the weather data from 5 min backward and to see if the feature importance of weather data is really less than sky image features in the same sample frequency.

We can also take in 1 min resolution sky image data to see if it can predict accurately. Such a short time span prediction will be beneficial when solar power company needs to connect their panel to the grid in the future. They can decide whether they needs to use battery to complement the variation of solar power. Another problem that sky image now encounter is that camera may easily get dusted, which will affect the resolution of the image.

Thus, inventing a way to automatically detect when the camera shall be cleaned can be another direction in the future.