Predicting Solar Power Generation from Weather Forecasts Using Machine Learning Models

Kushal Juneja
kushal19057@iiitd.ac.in

Naval Kumar Shukla
naval19065@iiitd.ac.in

Rishi Singhal
rishi19194@iiitd.ac.in

Udit Narang
udit19120@iiitd.ac.in

Abstract

Predicting future renewable energy generation is important, since the power grid must use fossil fuel based energy as demand varies. We explore prediction models for solar power generation from NSRDB database using machine learning techniques. We compare multiple regression techniques for generating prediction models.

https://github.com/kushal19057/ml-project-solar-intensity-estimation

1. Introduction

Energy from the sun is an inexhaustible resource that creates little to no pollution. However, current methods of collecting solar energy and converting it to electricity can capture and store only a small amount of the available energy from the sun at any given time. The highly variable nature of solar energy production puts stress on fossil fuel based power generation. We aim to predict solar intensity for a given area 48 hours into the future using local time-series weather data. Our goal is to provide high-confidence forecasts of solar generation (via solar intensity) using readily-available weather data. This will enable better regulation of fossil fuel based power generation. We have acquired solar intensity and weather data from NSRDB[1] and Sunrise-Sunset-API[2]. This project aligns with our sustainable development goals and deploying machine learning techniques to solve real world problems.

2. Literature Survey

Navin Sharma et al. compare multiple regression techniques for generating prediction models, including linear least squares regression and support vector machines using multiple kernel functions [4].

Abuella and Choudhary propose a solar energy prediction model using Artificial Neural Network's (ANN's)[3].

3. Dataset

3.1. General Information

The NSRDB dataset includes both observed weather data (temperature, pressure, cloud cover, solar zenith angle ,etc.) and solar intensity data measured in watts per square meter. The dataset includes several solar radiation measures such as Diffused Normal Irradiance (DNI), Diffused Horizontal Irradiance (DHI) and Global Horizontal Irradiance (GHI). We choose to include GHI measurements since it incorporates DHI, DNI and ambient solar radiation reflected from nearby surfaces. This makes it a good indicator for solar panel readings.

The NSRDB data is measured once every 10 minutes. We investigate a single location - Las Vegas, Nevada, USA. We limit our analysis to the year 2019 only.
Our dataset contains more than 50,000 distinct observations, each with 16 features (including Time values such as Month, Day, Hour and Minute) shown in table 1 with corresponding measure of GHI.

We obtain the sunrise and sunset times for each day from Sunrise-Sunset-API[2]. Using this, we add a new boolean column 'isDay'.

Month	Day	Hour
Minute	Temperature	Cloud Type
Fill Flag	Surface Albedo	Ozone
Pressure	Dew Point	Precipitable Water
Wind Direction	Wind Speed	Relative Humidity
Solar Zenith Angle	isDay	

3.2. Exploratory Data Analysis

We plot distribution of each feature using histograms, pie-charts and box-plots. We also study pairwise correlation between features using correlation-matrix & skew index table.

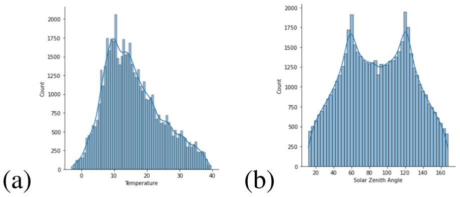

Figure 1: Plots: (a) Temperature (b) Solar Zenith

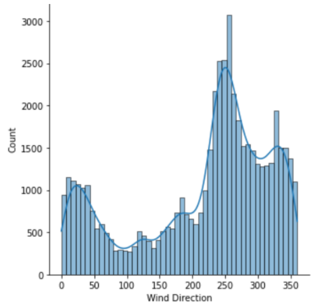

Figure 2: Wind Direction

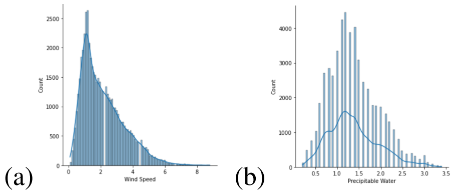

Figure 3: Plots: (a) Wind Speed (b) Precipitable Water

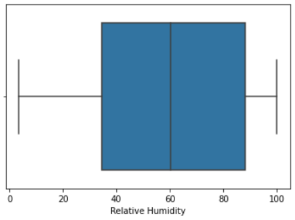

Figure 4: Relative Humidity

Feature	Skew Index
Month	-0.01
Day	0.07
Hour	0
Minute Temperature	0 0.55
Cloud Type	1.30
Dew Point	-1.34
Fill Flag	3.96
Ozone	0.78
Relative Humidity	-0.091
Solar Zenith Angle	-0.00013
Surface Albedo	-0.71
Pressure	0.046
Precip. Water	0.62
Wind Direction	-0.73
Wind Speed	0.99
isDay	-0.038

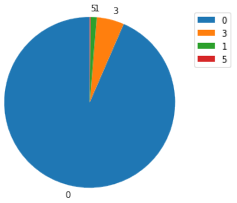

Figure 5: Fill Flag

3.3. Data Preprocessing

We spotted outliers in the distribution curves. However, we assumed that they are the natural part of the weather observations we are studying, hence we didn't remove them. Also we didn't find any missing/NaN values in our dataset, which would result in imbalance observations.

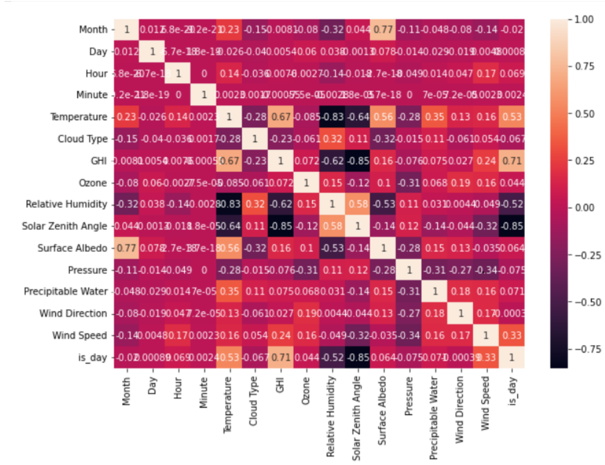

Figure 6: Correlation Matrix

The Correlation Matrix shows that features 'Dew Point' and 'Precipitable Water' are highly correlated. Since highly correlated features have almost the same effect on the dependent variable, we drop one of them. We test the performance of our baseline model by dropping both of the features one by one. Based on the results, we choose to drop 'Dew Point'.

The Pie-Chart for 'Fill Flag' feature shows that more than 90% observations correspond to '0'. It indicates that the value is not available. Therefore 'Fill Flag' adds negligible information to the dataset. Hence, we drop 'Fill Flag'.

3.4. Dataset Preparation

Our task is to predict solar intensity values 48 hours into the future. Thus, we establish a one-to-one mapping between current weather observations and GHI values 48 hours into the future.

4. Methadology

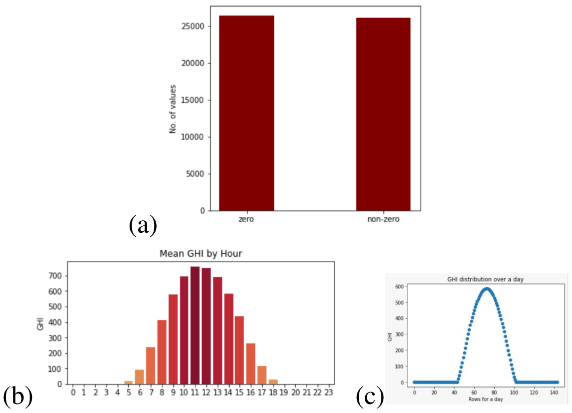

Figure 7: (a) GHI Sparsity (b) Mean GHI by Hour (c) GHI distribution over day

For the purpose of our project, we applied two different methodologies as follows:

1. In the first methodology, we made a single model and trained it on 80% of the data and then used the same for testing as well.
2. Another alternative approach that we adopted was to make two different models: one for the day data & one for the night data. In this method, we divided our training set into two different sets, day & night according to the *isDay* column and trained the day & night models using the day & night training data respectively. Similarly while testing, we again divided the test set into two different sets, day & night using the *isDay* column & then predicted the GHI values of day & night data using the day & night models respectively. Finally we combined the two prediction sets & computed the final RMSE of the test set.

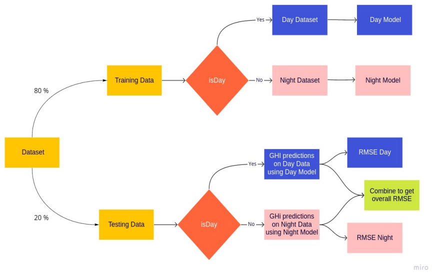

Figure 8: Schematic Diagram

NOTE:
As from Figure 7, we could clearly see that the GHI column is highly sparsed with around 50% values equal to zero. Also since most of the zero values correspond to the the time before sunrise and after sunset. Therefore, while designing the 2nd methodology, we have taken this issue into account.

To test the performance of our models we have the used the RMSE score using the following formula:-

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

We considered linear regression, along with lasso and ridge as our baseline models. We also have performed linear regression with feature expansions (with and without PCA) to improve the performance.

4.1. Feature Expansion

In order to expand the feature set for predicting the GHI values, we added weather observations from the adjacent past time points. We started with 15 features, and after *nth* expansion, the number of features increased to 15*(n+1).

4.2. PCA

As one can observe that for large values of *n*, the feature set becomes very large, adding to the complexity of the models. In order to reduce the model complexity without compromising the performance, we applied PCA to get the features with sufficiently high explained variance.

Apart from linear regression, we also have applied polynomial regression, regression trees, SVM regression & Artificial Neural Networks for both the methodologies that we proposed and compared the results.

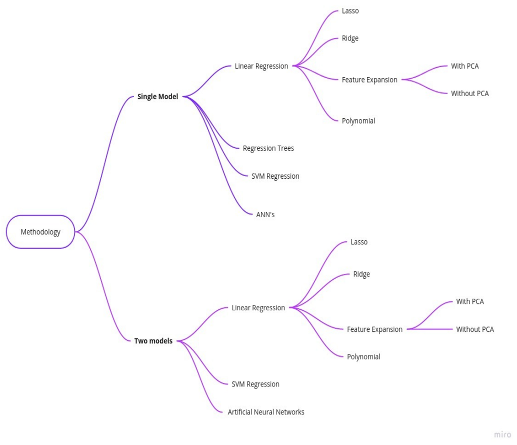

Figure 9: Schematic Diagram

5. Results and Analysis

5.1. Linear Regression: Baseline Model

Table 1: Single Model (Baseline)

Model	Train RMSE	Test RMSE
Linear	151.554	152.019
Lasso	151.554	152.018
Ridge	151.554	152.017

Table 2: Day-Night Model (Baseline)

Model	Train RMSE	Test RMSE
Linear	81.135	82.182
Lasso	81.135	82.182
Ridge	81.137	82.178

We observed low variance in the baseline LR models (both in the single as well as in the day-night model). However, as shown above, the day-night model performed better than the single model because of the sparsity in the data set. Also since the models are not over-fitting, applying lasso and ridge did not helped to improve the performance.

5.2. Linear Regression: Feature Expansion

As an extension to the baseline linear regression model, we included more weather observations into our feature set for predicting solar intensity. To add more weather observations, we use the weather observations from the adjacent past time points.

Table 3: Expansions v/s # Features

Expansions	1	2	5	30	60
# Features	30	45	90	465	915

On increasing the number of expansions, we observed that both train and test rmse decreased till 25 expansions. But on further expansions, we started to observe an increase in the variance due to the high dimensionality of the data.

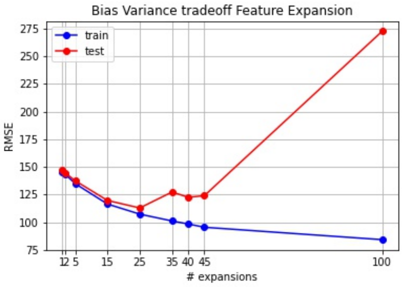

Figure 10: Feature Expansion without PCA

In order to reduce the dimensions and the complexity of our model, we applied PCA to get the features with high explained variance.

Table 4: Single Model with PCA

Degree	PCA components	Train rmse	Test rmse
1	7	162.11	162.19
5	15	161.53	161.55
25	50	114.45	115.91
40	80	107.04	107.84
100	250	97.94	99.39

Table 5: Day Night Model with PCA

Degree	PCA components	Train RMSE	Test RMSE
1	7	82.90	83.83
5	15	83.44	84.39
25	50	82.41	83.74
40	80	80.32	81.86
100	250	78.03	79.84

When we applied feature expansion with PCA, the complexity of the model decreased. Therefore with the increase in the number of the expansions, both the train and test rmse decreased.

5.3. Polynomial Regression

We now apply a mathematical feature expansion. We expand our feature set to include the square of each feature, as well as the pairwise interaction between each pair of features. Since the expansion in the number of features was exponential, we applied polynomials for degrees 2, 3, and 4 only.

Table 6: Degress v/s # Features

Degree	2	3	4
# Features	136	816	3876

Table 7: Single Model

Polynomial Degree	Train RMSE	Test RMSE
2	78.95	79.96
3	69.49	70.53
4	51.98	55.99

Table 8: Day Night Model

Polynomial Degree	Train RMSE	Test RMSE
2	75.66	76.81
3	63.36	65.58
4	45.11	55.63

As we were increasing the degree of the polynomial regression, the train and test rmse reduced. Although at degree 4, we can see the variance has increased. We were not able to increase the degree beyond 4 because of the limited computational resources.

5.4. Regression Trees

We apply regression trees. Since regression trees are able to capture non-linear and arbitrarily shaped decision boundaries, we expect these to perform better than linear regression techniques discussed above.

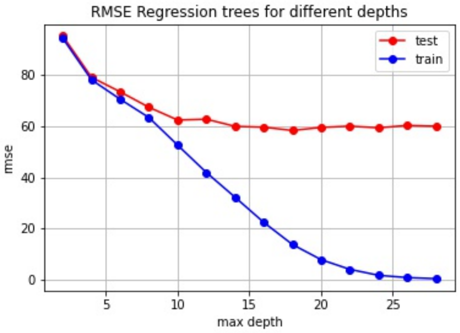

Figure 11: Regression Tree RMSE vs depth

Regression Tree	Train RMSE	Test RMSE
Without Pruning	0	59.9
Pre Pruning	52.6	62.4

On applying regression trees without pruning, we observed high variance and low bias. We used pruning techniques by restricting max depth of tree. As shown in the graph, the bias-variance tradeoff increases after max depth = 10. Thus, we choose max depth = 10. This improves the bias-variance tradeoff.

5.5. SVM Regression

We then applied Support Vector Machines because of its ability to handle non-linearity in the data. The performance of the SVM depends on the selection of an appropriate kernel function and parameters. In our work, we used four distinct SVM kernel functions: Linear Kernel, Polynomial Kernel, Sigmoid Kernel, and the Radial Basis Function (RBF). The SVM uses the kernel function to transform the data from the input space to the high-dimensional feature space.

Table 9: Single Model

Kernel	Train RMSE	Test RMSE
rbf	118.73	119.5
sigmoid	221.71	221.05
linear	153.62	153.97
polynomial	171.53	169.9

Kernel	Train RMSE	Test RMSE
rbf	90.52	90.64
sigmoid	113.21	113.81
linear	85.63	87.71
polynomial	111.65	110.97

Table 10: Day Night Model

5.6. Artificial Neutal Networks

As we know that the above models are not too complex, so we made use of ANN as well. We used the Keras library to implement the ANN model. We tried various combinations of activation functions, number of layers & layer sizes and compared the train & test RMSE that we got for all these combinations while keeping in mind the time complexity during the training of the model. The following table has the best three RMSE values that we got from the different combinations that we made use of.

Table 11: Single Model(LeakyR means LeakyRelu)

# Layers	Layer sizes	Activation Functions	Train RMSE	Test RMSE
4	14-64-32-1	LeakyR, LeakyR LeakyR, LeakyR	41.42	46.93
4	14-64-32-1	Relu, LeakyR LeakyR, LeakyR	54.21	56.69
4	14-64-32-1	Relu, Relu LeakyR, LeakyR	73.79	74.43

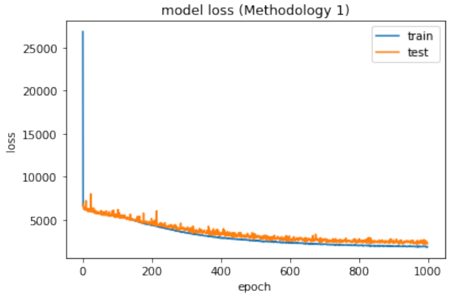

Figure 12: Loss Plot: Single Model

Table 12: Day Night Model(LeakyR means LeakyRelu)

# Layers	Layer sizes	Activation Functions	Train RMSE	Test RMSE
4	14-64-32-1	LeakyR, LeakyR LeakyR, LeakyR	50.36	45.58
4	14-64-32-1	Relu, LeakyR LeakyR, LeakyR	59.95	53.33
4	14-64-32-1	Relu, Relu LeakyR, LeakyR	64.88	60.91

After observing the 2 tables it is quite clear that we got best results in the case when we used 4 layers of size 14,64,32,1 with activation functions [LeakyRelu,LeakyRelu,LeakyRelu,LeakyRelu] in the respective layers for both types of methodology.

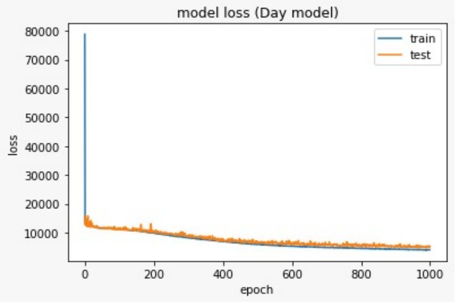

Figure 13: Loss Plot: Day Model

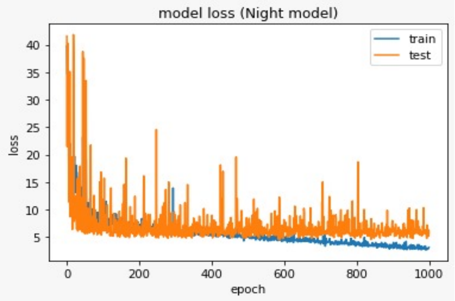

Figure 14: Loss Plot: Night Model

6. Conclusion

6.1. Outcomes

We are pleased to inform that our new methodology of introducing an ensemble of 2 models (Day Model & Night model) proved to be highly potent as we were able to get very good train & test RMSE values than the ones we got with 1 model only which required too much training time for complex models like ANN to get lesser RMSE values. One of the limitations of our project was that we did not have the time series data for years before 2019. Also we have currently looked into only one geographical location for doing our predictions.

6.2. Future Work

Future improvements on our project could be to use more complex Deep Learning models along with different regularization techniques to achieve better RMSE values. We could also extend our dataset by acquiring the weather data of various other geographical locations. This entire idea could be extended to be implemented in various solar power plants operation sites for their effective functioning over the year.

6.3. Member Contribution

Each member has contributed to the following areas:

Rishi: EDA, Coming up with methodology II, Neural Networks, SVM Regression, Feature Expansion without pca, Polynomial Regression, Result Analysis, Report Making
Udit: Data pre-processing, Coming up with methodology II, Baseline LR, Polynomial Regression, SVM Regression, Feature Expansion with PCA, Result , Results Analysis, Report Making
Naval: Data pre-processing, Coming up with methodology II, Baseline LR (with Lasso and Ridge), Feature Expansion with PCA, Neural Networks, Results Analysis, Report Making
Kushal: Data Preparation, EDA, Coming up with methodology II, Feature Expansion without PCA, Neural Networks, Regression Trees, Results Analysis, Report Making

References

[1] https://nsrdb.nrel.gov.

[2] https://sunrise-sunset.org/api.

[3] M. abuella and b. chowdhury, solar power forecasting using artificial neural networks, 2015 north american power symposium (naps), 2015, pp. 1-5, doi: 10.1109/naps.2015.7335176.

[4] N. sharma, p. sharma, d. irwin and p. shenoy, predicting solar generation from weather forecasts using machine learning, 2011 ieee international conference on smart grid communications (smartgridcomm), 2011, pp. 528-533, doi: 10.1109/smartgridcomm.2011.6102379.