Solar Energy Production Regression Using Weather Data and Machine Learning Methods

Bradley Payne

May 4, 2021

Abstract

There is a need to move the world's energy population away from traditional energy production in favor of carbon-reducing approaches. Forecasting Solar energy generation is crucial to obtain a balanced grid. In this paper, solar energy generation from ASPIRE labs and weather data from the USU Climate Center are combined with machine learning methods to create models that aim to solve this forecasting problem. Future work is briefly discussed.

1 Introduction

Electricity production contributes 25% of the global greenhouse gas emissions [1]. To reduce the amount of gas emission without reducing power demands, clean energy solutions must be properly integrated with the rest of the grid. The demand for clean energy solutions has led to improvements in the manufacturing process of photovoltaic (PV) panels. As a result, PV panels have become much more economical in recent years [2].

The issue with the direct integration of PV panels into the grid is that production is dependant on the weather, and the amount of energy produced fluctuates throughout the day. The need to forecast production is increasingly important to maintain a balanced power grid [3]. This paper uses solar generation data from a 100 KW grid and historical weather data from a nearby climate center to build regression models that intend to predict one hour of solar generation.

The rest of this paper is outlined as follows: Section 2 discusses some previous work researchers have done to solve similar problems. Section 3 is a brief discussion on data processing and additions prior to regression fitting. Section 4 contains the regression methods used to fit the data. Section 5 contains the results of the models. Finally, conclusions and future work are discussed

2 Previous Work

Many researchers have studied the problem of PV production and solar irradiance. Researchers have tried to predict energy generations from 1 - 100 hours into the future using a large variety of statistical and machine learning (ML) methods. Lasso, Auto-regressive, K-nearest Neighbors, Regression Trees, Artificial Neural Networks, deep learning, crisp and fuzzy Support Vector regressive machines, and cloud position modeling are among the many techniques previously used [2; 4; 5; 6; 7; 8]. It is impossible to make a direct comparison between the different methods and previous works because there is a large variance in input and output data sets. Each ML method has strengths and weaknesses, but that discussion is outside of the scope of this work.

In addition to ML fitting methods, data processing and splitting have also been studied to help improve results. Training models on a single location instead of multiple has been shown to improve results. Splitting data into seasons and fitting individually also leads to less error than training on multiple seasons [2]. Other researchers have added solar coordinate information [8] and rolling averages [6] to improve results.

3 Data Processing

The solar data used for this study was obtained from ASPIRE Research Center in North Logan, Utah. They began collecting data for their 100 KW solar grid in July 2019. The data received from them contained the amount of energy produced by the panels throughout daylight hours.

As is the case with any real data, there were a large number of data points that had been corrupted or were garbage collection. 31% of the raw data was corrupted or garbage collected. This affected 44% of the total observed hours. The initial approach was to sum the generation data across a whole hour using an unweighted average of uncorrupted data to replace the corrupted values.

Summing the data in each hour produced poor results. One reason summing the hourly data may not have been a good approach is that the number of data collections within an hour varies from 2 - 500 points. The distribution of data points per hour is shown in Figure 1.

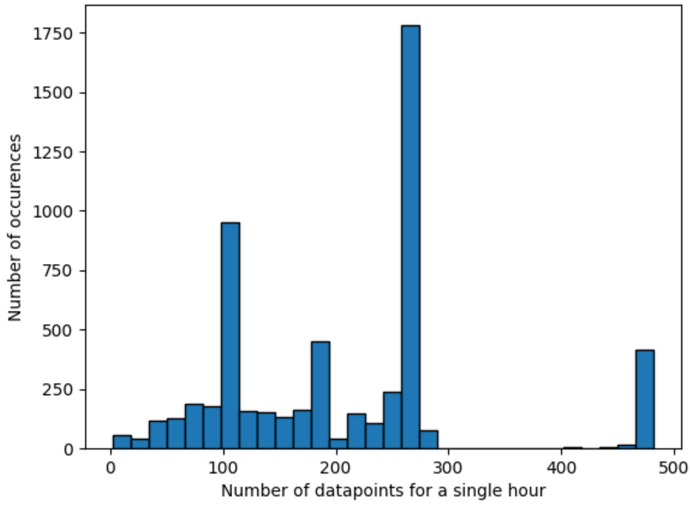

Figure 1: Distribution of the number of data points per hour

Instead of using the sum, corrupted values were completely discarded. The data was then downsampled to a single, average value to represent the amount of energy produced during the hour.

Since historical weather forecasts are not kept, historical weather data was used as input to try to predict the solar output. The limitation of using historical observations is that we are assuming the weather forecast to forecast solar energy is a perfect model. Perfect weather forecasting is not the case, but it serves as a starting point for future work.

The USU Climate Center keeps detailed weather observations for each hour. This data has 44 columns including temperature, humidity, pressure, precipitation, etc. A full list of the columns can be found on the USU climate center page [9]. Several columns contained timestamp strings of certain occurrences such as the highest wind speed during the hour. These columns were removed and 33 columns remained as input.

To supplement the weather data, month, day of year, hour of day, and sun coordinates were added as inputs. The sun coordinates used are the Azimuth and Zenith angles computed with the Astral python package. Each of these supplemented data columns were encoded as two-dimensional numbers by taking the sin and cosine. The reason this is done is that using the raw numbers does not properly represent their similarity. For example, the weather and solar generation of January and December are fairly similar at this location. However, using raw month values of 1 and 12 indicates a large difference that the models would have to interpret. Sine and Cosine are periodic functions, so the similarity is made obvious to the models.

4 Methods

An 85%-15% training/validation split was used for this study. Many ML models are sensitive to the various scales of input and output data. To eliminate this sensitivity, all of the data was scaled to a range of [0,1] before fitting. All input columns were scaled using a standard min-max scaler. Generation data was scaled using a quantile transformer with uniform distribution. Figures 2 and 3 show the distribution of solar generation before and after scaling.

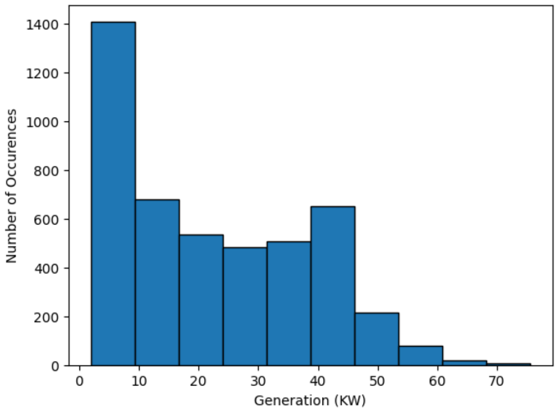

Figure 2: Generation distribution before scaling

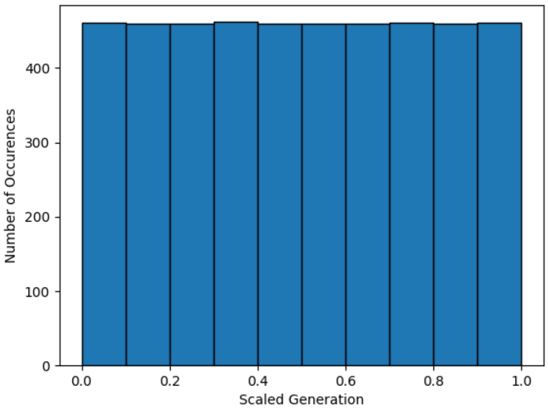

Figure 3: Generation distribution after scaling

The output data was scaled to have a uniform distribution to reduce model bias towards a certain generation output. If there are a large number of lower values, the model will tend to underestimate times when generation is higher. This phenomenon is less evident in regression problems but is especially prevalent in classification problems when one category has a much larger sample size. Other scaling methods were tested on the generation data, and the uniform scaling produced the models with the lowest error after inverse transforming the data.

Various ML and deep learning methods have been used in the literature to solve this problem. The differrent methods used were K-nearest neighbors (KNN), random forests (RF), support vector regression (SVR), artificial neural networks, (ANN), and recurrent neural networks (RNN). An explanation of these methods is outside the scope of this work. Sci-kit Learn and Keras python packages were used to implement the different methods because they are well-tested libraries and easy to use.

After model fitting, the predictions were inverse transformed to get an accurate model score. Models are scored using four metrics: coefficient of determination R^2, mean absolute percent error (MAPE), mean absolute error (MAE), and root mean squared error (RMSE). These four metrics were chosen because they each give different insights into how well the model is performing. For example, MAE describes the average residuals between actual and estimated value, and RMSE indicates the standard deviation of the residual error.

5 Results

The five regression methods mentioned in the previous section were fit on the training data. After being fit with the training data, both the training and input data were fed through the models to obtain the predictions. The predictions were inverse transformed using the quantile transformer used to scale the ground truth initially. The raw truth and inverse predictions were evaluated using the four metrics previously mentioned. The results on training data for all models are shown in Table 1. The validation results are given in Table 2.

Table 1: All models scored on training data

	$R^2 \uparrow$	MAPE (%) $\downarrow$	MAE (KW) $\downarrow$	RMSE (KW) $\downarrow$
KNN	0.905	19	3.18	4.96
RF	0.987	5.29	0.955	1.84
SVR	0.922	16.2	2.88	4.5
ANN	0.900	23.9	3.44	5.11
RNN	0.781	30.2	5.4	7.54

Table 2: All models scored on validation data

	$R^2 \uparrow$	MAPE (%) $\downarrow$	MAE (KW) $\downarrow$	RMSE (KW) $\downarrow$
KNN	0.551	52.6	6.69	9.46
RF	0.751	34.0	4.54	7.05
SVR	0.667	45.7	5.62	8.15
ANN	0.752	53.3	4.54	7.03
RNN	0.481	54.1	7.25	10.17

For all experiments performed on this data set, random forests had the best results on training and validation data sets. Figures 4 and 5 show the regression of random forests on the data. If all the points were to lie on the diagonal line, the model would be perfect.

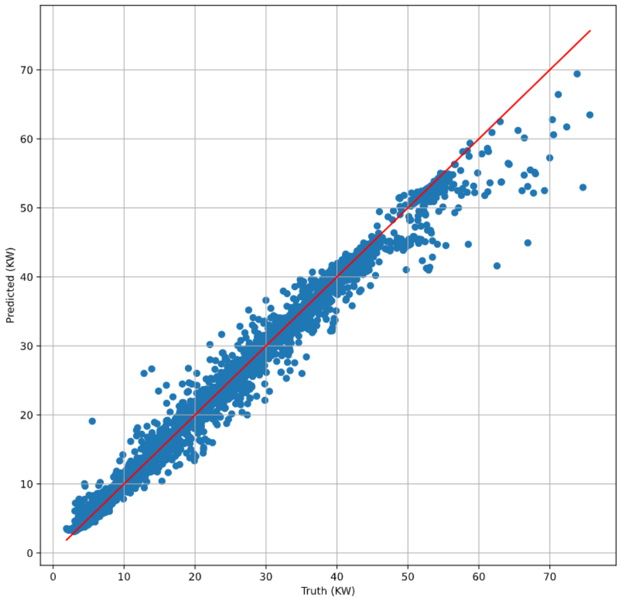

Figure 4: RF training regression

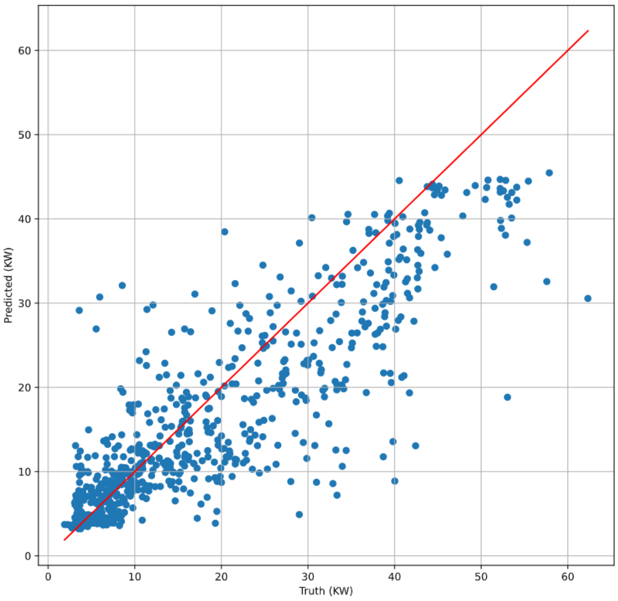

Figure 5: RF validation regression

Using the different error metrics, random forests were found to have the best results on both training and validation data. However, The error of the random forest was still fairly large on the validation set. This led to the decision to split the data into seasons. Only random forests were considered for training separate models on 'warm' and 'cold' seasons. The warm season in Logan, UT was determined to be approximately April through September. The cold season is the rest of the year. The data was split into seasons. Then, seasonal data was split with an 85% - 15% training - validation split. Figures 6 and 7 show the regression of the warm season model.

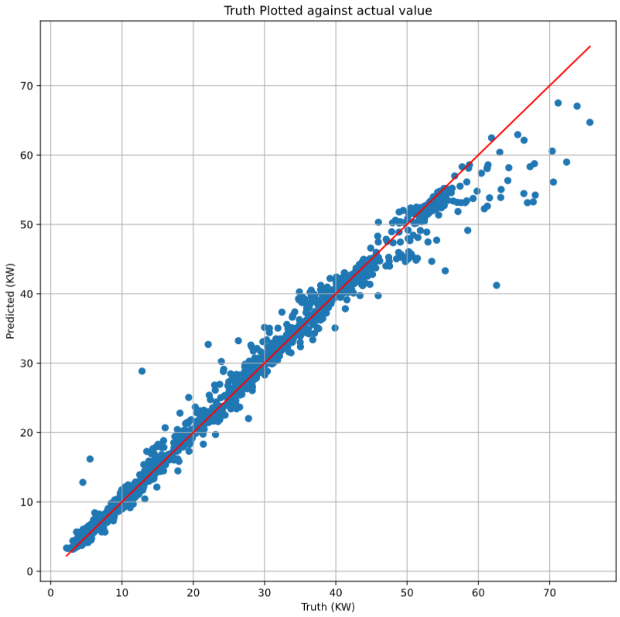

Figure 6: RF on warm training data

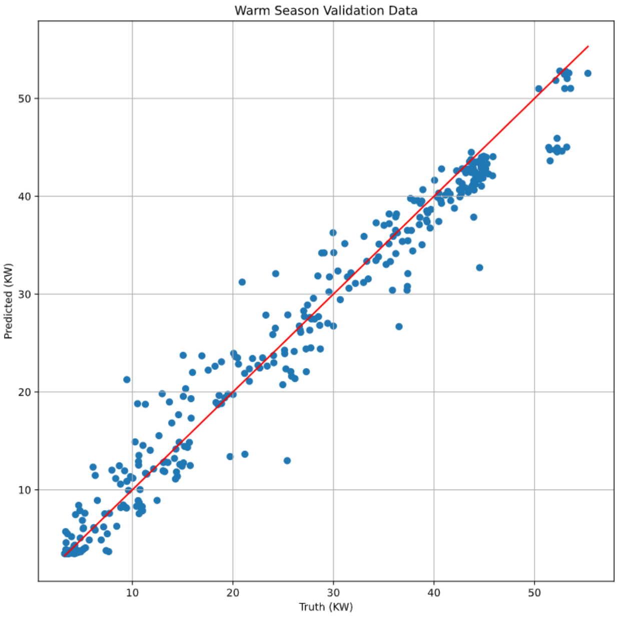

Figure 7: RF on warm validation data

The fit of training and validation of the warm model is significantly tighter than the model using the whole data set. The cold season, however, does not generalize in the same way. Figures 7 and 8 show the regression of the cold season model.

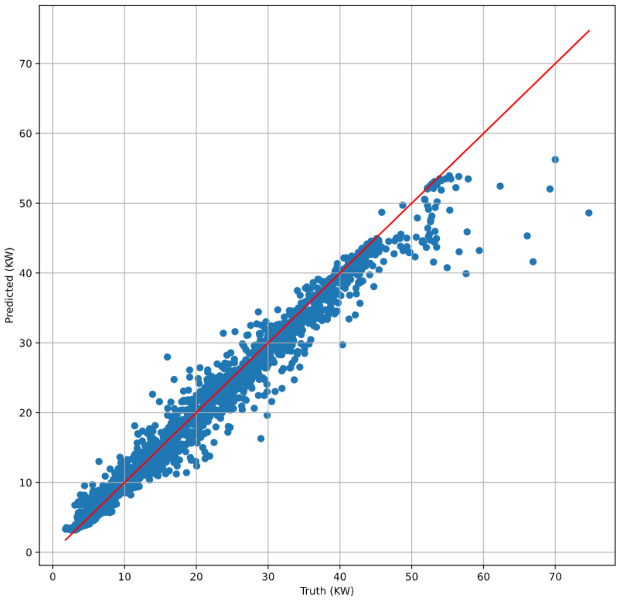

Figure 8: RF on cold training data

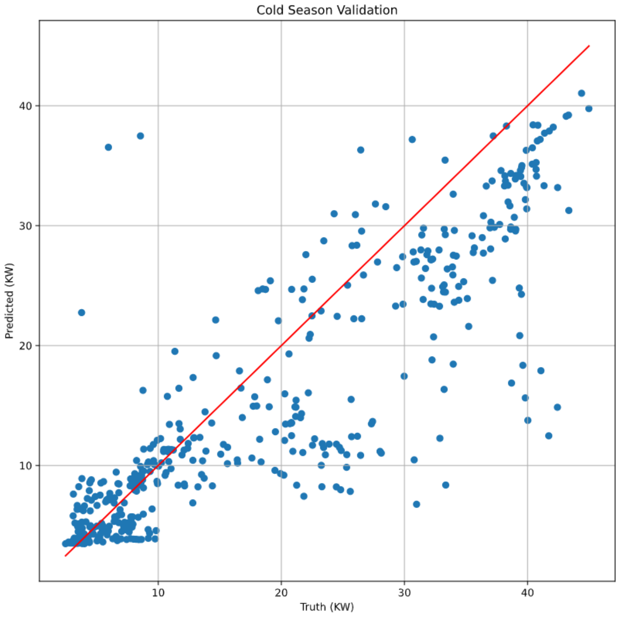

Figure 9: RF on cold validation data

The MAPE and overall fit (R^2) of the cold season validation were slightly better than the performance on the whole data set. Table 3 shows the error metrics of the warm, cold, and unsplit models. The combined train and test from table 3 are the results of combining the outputs of the warm and cold models and evaluating the whole data set. Doing this allows for a more direct comparison with the model trained on the whole data set.

Table 3: Metrics of fitting Random Forests on training and validation data

	$R^2 \uparrow$	MAPE (%) $\downarrow$	MAE $(KW) \downarrow$	RMSE $(KW) \downarrow$
Warm Train	0.984	4.1	0.795	1.83
Warm Test	0.952	11.8	1.82	2.73
Cold Train	0.980	7.81	1.49	2.44
Cold Test	0.772	30.7	5.43	7.81
Combined Train	0.984	6.04	1.07	2.07
Combined Test	0.851	26.1	3.56	5.72
Unsplit Train	0.987	5.30	0.962	1.85
Unsplit Test	0.752	33.2	4.54	7.03

6 Conclusions

This project used historical weather from USU Climate Center and solar production data from nearby ASPIRE labs to fit several models. The methods used in this paper were based on previous related research and experiments with this particular data set. The resultant model determined energy production with a reasonable amount of error.

On the data set used in this study, random forests produced models with much higher accuracy on both training and validation sets. One reason may be that random forests are generally more robust to noise in the output data [10].

Another finding of this work is scaling output data to a normal distribution helped the model to not underestimate generation during peak hours. Before making this adjustment, the models tended to greatly underestimate generations above 50 KW. This is likely due to the model being rewarded for fitting a large amount of low generation data points.

The final finding was that fitting models based on seasons result in lower error than using a model fit on several. This approach was mentioned in previous studies [2]. One reason that this approach may have decreased total error is that the range of generation values for the seasons is different. It is also possible the cold weather causes more noise in the sensors or panels themselves. The warm season benefited greatly from separate training, while the cold season performed about the same.

7 Future Work

There are many areas of this work that could be improved. Other authors use various other techniques more extensively that help produce accurate results. Isaksson and Conde [2] effectively use AutoRegressive (AR) models to use previous data to benefit their results. Kamarouthu [8] shows that modeling cloud movements with a numerical model is a great benefit, especially in short-term forecasting.

Some of the variables used as input to the regression models may not have a clear statistical relation with the amount of solar energy produced. Fuzzy regression techniques use fuzzy systems to produce accurate models in cases where input and output variables have a vague relationship. Baser and Demirhan [7] showed how using modern Fuzzy regression techniques can produce solar radiation forecasting models that are more robust to outliers. There may be an advantage to extending this work to use fuzzy models.

In addition to other fitting and feature engineering methods. This work also needs to be extended to include the ability to forecast energy. Some of the inputs used from the historical weather data may need to be modeled to use as input for a forecasting model.

References

[1] "Global Greenhouse Gas Emissions Data," Online, Available: https://www.epa.gov/ghgemissions/global-greenhouse-gas-emissions-data, Accessed: 03-May-2021.

[2] E. Isaksson and M. Karpe Conde, "Solar Power Forecasting with Machine Learning Techniques", *Dissertation,* 2018.

[3] R.H. Inman, H.T.C. Predro, and C.F.M. Coimbra, "Solar Forecasting Methods for Renewable Energy Integration," *Progress in Energy and Combustion Science,* vol. 39, pp.535-576, 2013.

[4] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting Solar Generation from Weather Forecasts Using Machine Learning," *IEEE International Conference on Smart Grid Communications,* pp. 528-533, 2011.

[5] J.F. Torres, A. Troncoso, I. Koprinska, Z. Wang, and F. Martinínez-Àlvarez, "Deep Learning for Big Data Time Series Forecasting Applied to Solar Power," *Advances in Intelligent Systems and Computing,* 2019.

[6] S. Mohanty, P.K. Patra, S.S. Sahoo, and A. Mohanty, "Forecasting of Solar Energy with Application for a Growing Economy Like India: Survey and Implication," *Renewable and Sustainable Energy Reviews,* vol. 78, pp. 539-553, 2017.

[7] F. Baser and H. Demirhan, "A Fuzzy Regression with Support Vector Machine Approach to the Estimation of Horizontal Global Solar Radiation," *Energy,* vol. 123, pp. 229-240, 2017.

[8] P.S. Kamarouthu, "Solar irradiance Prediction Using XG-Boost with the Numerical Weather Forecast," *All Graduate Theses and Dissertations,* 7896, 2020.

[9] U.S.U.- 2020, climate.usu.edu, Online, Available: https://climate.usu.edu/mchd/dashboard /dashboard.php?network=USUwxamp;station=1279257amp;units=Eamp;showgraph=0amp; Accessed: 04-May-2021.

[10] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, pp. 5-32, 2001.