Solar Power Generation Forecast: Using Random Forest Regression to Predict Solar Generation Output

Anthony Ganci

College of Engineering, University of Nevada, Reno

CS458: Introduction to Data Mining

Dr. Lei Yang

December 13, 2022

Abstract

Approaching this problem, I made sure to utilize some of the data mining techniques we learned this semester. For example, things like data preprocessing, model selection, and hyperparameter tuning were the techniques I knew I would need to get an acceptable accuracy for this problem. Due to the large nature of the two datasets given to us the first thing I worked on was getting these into forms I could understand and work with. I then proceeded with feature selection and scaling. Once I was happy with my datasets, I decided on a model to use which ended up being Random Forest Regressor. Once I had a baseline for Random Forest Regressor, I moved onto tuning the hyperparameters which I used cross-fold validation to find the best combinations of various parameters. In summary, my approach to this problem was constructing my datasets, data preprocessing, model selection, and hyperparameter tuning.

Introduction

Solar generation is increasingly becoming more reliant on probabilistic forecasting to assist energy companies in predicting demand and knowing when their best times for generation will be. Looking at the data given to us we can see that solar generation occurs roughly twelve hours a day. However, throughout these twelve hours the fluctuation in power generation can be massive. There are many variables which contribute to solar power generation, and it is not just as simple as sunlight hitting the panel as you might expect. For example, some variables to consider are surface pressure, relative humidity, and surface solar radiation down. Accurately predicting solar power generation will become increasingly more important as we transition away from other forms of power generation like fossil fuels and need to accurately know if the solar generation can meet demand.

In this specific problem, we were given a dataset which consisted of twelve independent variables in addition to a zone ID, timestamp, and normalized power generation for that time. The twelve independent variables given to us spanned multiple different units and would be the main source used to predict solar power generation in addition to the time of day. We were given a training dataset and a testing dataset which included the same variables and data. The solar panels were located in Australia in three different zones, and we were expected to predict the solar power generation for each respective zone. A 24-hour forecast of power generation is what we were expected to predict using any model of our choice and any other data mining techniques we wished to use. We would evaluate our models and results using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

The goals I set out to achieve for this project were to accurately predict the solar power generation within an error of 10% for MAE and 15% for RMSE. I came up with these numbers after doing some initial testing with various models to get a rough estimate of what I could expect to get to. The problems I would have to solve to get these numbers were finding out which variables had high correlation to the power generation output, which model would be best suited to this dataset as well as the task given, and tuning my hyperparameters to find the best setup for my model of choice. I would need to find ways to accomplish these problems before I could get within my goal error range. Using the data mining techniques I learned over the course of this semester I was confident that I would be able to meet my goal and get results I found respectable.

Background and related work

To summarize the papers I read to prepare myself for this challenge here is a couple sentences each describing the papers. According to Dang et al. (2022), they describe how Random Forest can only give a predictive value, so it is often paired with probabilistic theory to make a new forecasting model. They also discussed how random forest is better tailored for problems where you are doing high-dimensionality regression. According to Liu et al. (2019), in recent years random forest has become one of the more prominent models used in the prediction of electric energy. They also discuss how they have modified random forest using relational analysis in one case and grey projection in another to achieve better efficiency, performance, and accuracy.

From these papers I was able to gather what has been attempted when using random forest models for the purpose of solar energy prediction. I gathered that random forest was a very strong choice to use because of how popular it had become in recent years to predict energy generation. Considering that random forest is very useful for situations of high-dimensionality and that the scenario I was given fit the description I knew it would work well. According to Liu et al. (2019), “Huang Han et al. used the random forest algorithm to increase the prediction accuracy of the model by adjusting the number of trees.” This was something I was going to attempt to recreate in my model by experimenting with the number of trees via hyperparameter tuning to see if I could achieve similar results. According to Dang et al. (2022), “As for load data, EMD load components, and load price data, these components are processed by min–max normalization in the range of [0,1] for better training results.” I took note of this and made sure I used normalization or a scalar in some form to see if I could also achieve better results with normalized values.

Technical approach

The first step in approaching this problem was getting the train and test sets into a form I could easily work with. I did this by making a Pandas DataFrame from the datasets. The last step of my preparation process was dividing up the train and test DataFrames into subsets based on their respective zones. Once I was done with my preparation, I moved onto the preprocessing stage.

The first preprocessing method I used was a function from sklearn's feature_selection library called SelectKBest with f_regression as its scoring function because it is recommended for finding predictive features for regression models which is what I want to find. This returned a list of scores assigned to all variables in the dataset and score for their correlation to the power output of the given row. Features TIMESTAMP, VAR134, VAR78, and VAR79 all had a score lower than 300 whereas others where either higher than 300 and most where higher than 1000. I choose to eliminate these features from my function since I believed that they were not representative of any meaningful correlation to power output. The other preprocessing method I used was a StandardScaler from sklearn's preprocessing library. I transformed the train and test values before using them in my model and then inverse transformed them after the prediction to get the true values. These were the only preprocessing methods I used, and I saw small improvements to my Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) when using these.

The model I chose to use was a Random Forest Regressor from sklearn's ensemble library. I chose to use this model after doing basic tests with other models and seeing the initial results for each being much worse than Random Forest Regressor. For example, after doing some more research I found some articles about Time Series Forecasting using Random Forest

Regressors as well as papers discussing using Random Forest for short term forecasting. Which lead me to use Random Forest as my model of choice.

I decided to use sklearn's RandomizedSearchCV to do my hyperparameter tuning. RandomizedSearchCV works by making a parameter grid of random parameters based on information you provided. The parameter grid I used had 3,960 possible combinations and used 3-fold cross-validation. After doing this I got my best possible set of parameters and used this set of parameters in my Random Forest model.

Evaluation results

The test environment I used to conduct my tests was a python notebook. The program was broken into subblocks which I ran consecutively to get all my results. I would display parts of the test set and training set for each zone to make sure that the data I was using was correct based on the csv we were given. After this I would run the prediction model a few times to ensure that there were not incidents where results were not consistent. Once I did this, I was confident that my testing methods were sound and reproducible.

The results I got are roughly consistent from zone-to-zone with there only being a slight difference between all three. I think that the Random Forest Regressor model has a lot of advantages and disadvantages. Better tuning of the hyperparameters for Random Forest I believe could lead you to get at least an MAE score of around 5% and RMSE score of less than 10%. However, I am not sure that getting hyper accurate results is possible with this model, MAE less than 4% and RMSE less than 8%, because the model seems to have diminishing returns after the aforementioned scores. However, another pro of this model is that it was much easier than other

models to get into a fairly accurate prediction range. Which is why I believe this model is very well suited to the data provided to us and is the best choice.

ZoneID	1	2	3	Overall
MAE	0.059	0.060	0.064	0.061
RMSE	0.105	0.103	0.110	0.106

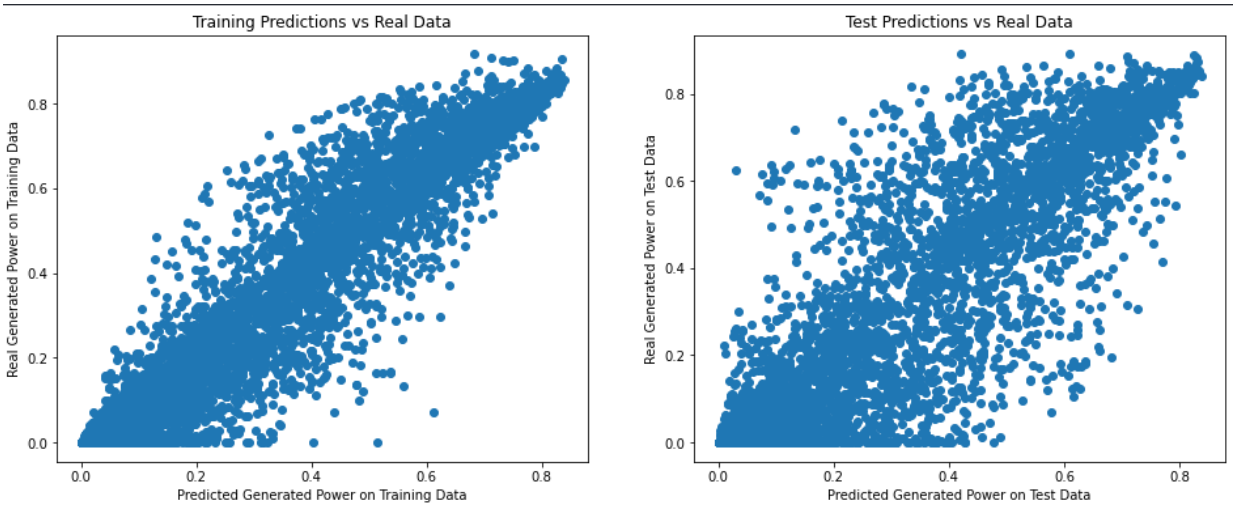

Figure 1: Zone 1 Predicted Power

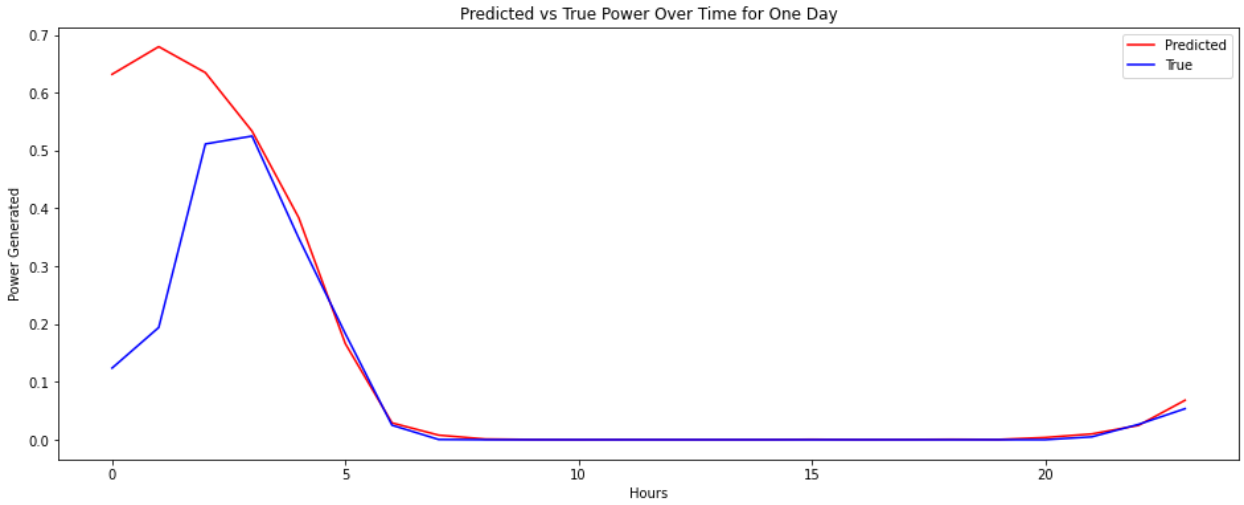

Figure 2: Zone 1 Predicted vs True Over Time for One Day

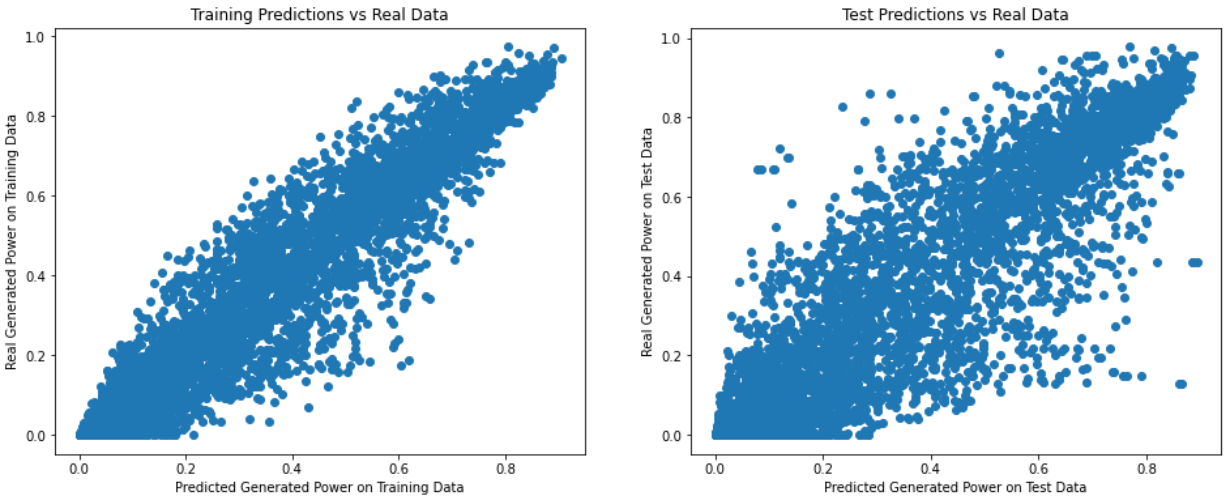

Figure 3: Zone 2 Predicted Power

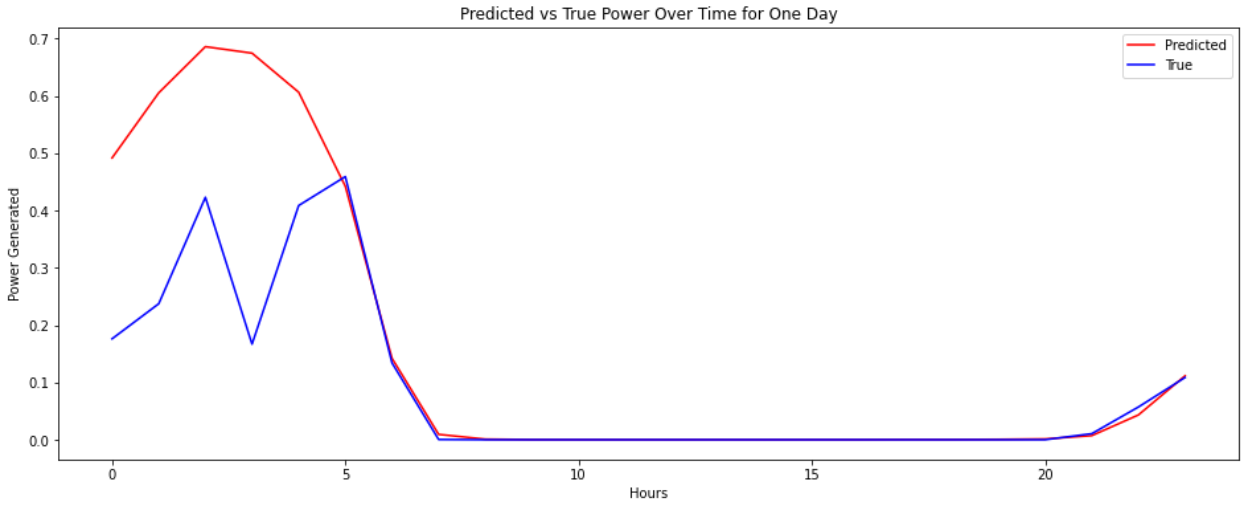

Figure 4: Zone 2 Predicted vs True Over Time for One Day

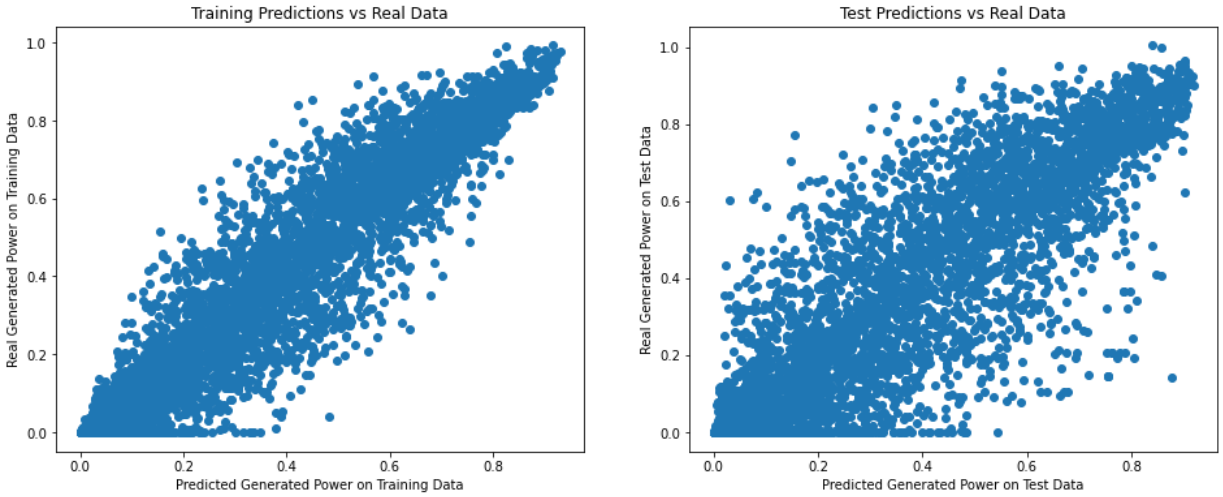

Figure 5: Zone 3 Predicted Power

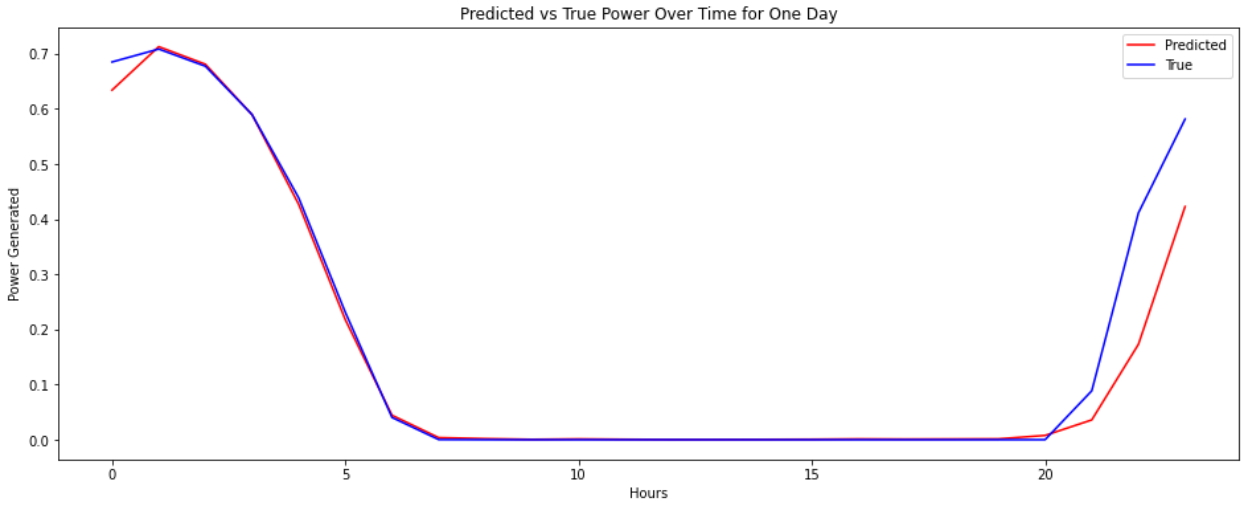

Figure 6: Zone 3 Predicted vs True Over Time for One Day

Conclusion

In summary, using a Random Forest Regression model I achieved in creating a model which predicts the solar power generation over a year period. I used data preprocessing techniques such as feature selection and a standardized scaling to assist my model in predicting power generation output. Using SelectKBest which used correlation score to show which features were not particularly useful I eliminated features TIMESTAMP, VAR134, VAR78, and VAR79 which increased my accuracy by roughly 1-2%. Then I used hyperparameter tuning techniques such as RandomizedGridCV to test hyperparameters using 3-fold cross-validation to score them. Doing this gave me by best possible set of hyperparameters and helped me get a more accurate score. In conclusion, using various data mining techniques I learned over the course of this semester I achieved a mean absolute error of 0.061 and root mean squared error of 0.106 using a Random Forest Regression model to predict solar power generation.

References

Dang, Peng, L., Zhao, J., Li, J., & Kong, Z. (2022). A Quantile Regression Random Forest-Based Short-Term Load Probabilistic Forecasting Method. Energies (Basel), 15(2), 663–. https://doi.org/10.3390/en15020663

Liu, Hu, Y., & Ai, X. (2019). Research on Power Load Forecasting Based on Random Forest Regression. IOP Conference Series. Earth and Environmental Science, 252(3), 32171–. https://doi.org/10.1088/1755-1315/252/3/032171