Surface Albedo Prediction Using Random Forests

Srinidhi Budhiraju
Arizona State University
Tempe, United States
sbudhir3@asu.edu

Sameeksha Katoch
Arizona State University
Tempe, United States
skatoch1@asu.edu

Andreas Spanias
Arizona State University
Tempe, United States
spanias@asu.edu

Yiannis Tofis
University of Cyprus
Nicosia, Cyprus
ytofisol@ucy.cy

***Abstract*—With rising concerns over climate change, there is an increasing need for renewable energy sources. Photovoltaic(PV) systems are one of the most environmentally friendly ways of producing energy. However, the fluctuations in power outputs from utility scale PV arrays makes it difficult to incorporate them into electric grids. The power output is directly related to the irradiance and the irradiance is related to the surface albedo, which is the fraction of sunlight reflected by a surface. If we can predict the surface albedo, we can predict the power output. Using random forest regression, we can make predictions of the power output based on various features. In response to this prediction, the topology of the system may be reconfigured.**

***Index Terms*—machine learning, random forest regression, surface albedo**

I. Introduction

As concerns over the climate continue to grow, there is an increasing need for renewable sources of energy. A key way to reduce the depletion fossil fuels is by utilizing Photovoltaic(PV) systems as a source of energy [1]. While PV systems decrease greenhouse gas emissions, there are still improvements to be made.

Although PV systems have many benefits, their power fluctuations make them difficult to implement in a power grid [4]. Therefore, it would be beneficial to be able to predict the power output of the system. The power output is directly related to the solar irradiance [4]. If we can obtain an accurate prediction of the solar irradiance, we can get a close prediction for the power output. The ratio of the global horizontal irradiance (GHI) to the ground-reflected irradiance (GRI) is the surface albedo [2]. So, if we can predict the surface albedo, we can predict the power output.

Surface albedo is the fraction of sunlight that is reflected by a surface. This fraction is not constant for all surfaces. The various factors influencing surface albedo include the season, cloud coverage,and solar elevation [2]. Currently, a ground albedo of 0.2 is assumed when modeling PV systems. However, this assumption is unreliable. As a result of inaccurately predicting the PV surface albedo, PV power output will be inaccurate leading to a less than optimal design [3].

In order to reduce the fluctuations in the power output as mentioned above, the topology of the system can be reconfigured in response to the prediction found by the random forest algorithm as shown in Figure 1 [5].

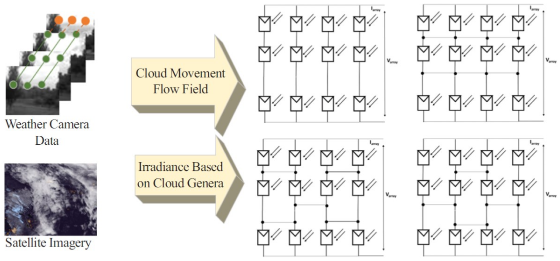


Fig. 1. Topology Reconfiguration Due to Cloud Movement [5]

In this paper we will predict surface albedo using random forest regression. The main benefit of using the random forest algorithm is that it has high accuracy due to using multiple loosely related trees rather than relying on one single model. The features that will be used with this algorithm are dew point, solar zenith angle, cloud type, wind speed, precipitable water, relative humidity, temperature, and pressure. This data will be coming from a weather station close to the location of the solar panels.

As shown in Figure 2, we will use the weather station data to predict the surface albedo using random forest regression. Next, we will compare the predicted surface albedo values to the ground truth surface albedo. We will calculate the mean squared loss function to determine the accuracy of our predictions.

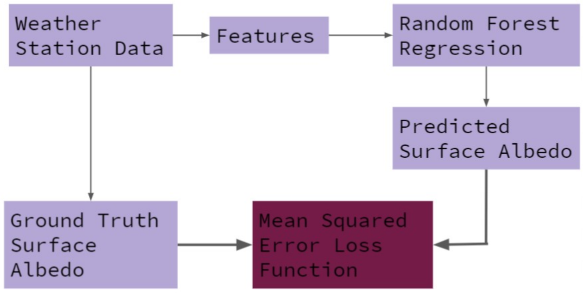


Fig. 2. Flow Chart

II. National Solar Research Database Dataset

A. Data Features

The features in the National Solar Research Database (NSRDB) dataset include DHI (Diffuse Horizontal Irradiance), DNI (Direct Normal Irradiance), GHI (Global Horizontal Irradiance), Clearsky DHI, Clearsky DNI, Clearsky GHI, Cloud Type, Dew Point, Solar Zenith Angle, Fill Flag, Surface Albedo (what we want to predict), Wind Speed, Precipitable

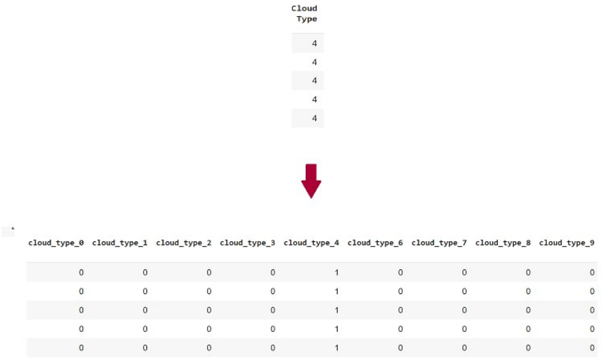
Water, Wind Direction, Relative Humidity, Temperature, and Pressure. The data for these features were collected over an entire year at 30 minute time intervals.

The DHI is the amount of radiance that is not received directly by the sun per unit area of a surface. The DNI is the amount of radiation received perpendicularly in a direct path from the sun per unit area of surface. The GHI is the total amount of radiation received per unit area of surface. It is the sum of DHI and DNI. The dew point is the temperature that air must be cooled to in order to become saturated with water vapor. The solar zenith angle is the sun ray's angle from the vertical. The surface albedo is the fraction of sunlight reflected by a surface. Precipitable water is the amount of water vapor in a column of the atmosphere. Relative Humidity is the percentage of water of vapor in the air.

B. Preprocessing Data

The first change made to the dataset was removing the unnecessary rows and columns. These include things such as timestamps, which are unnecessary in predicting the surface albedo from weather conditions. This was done to simply make the dataset easier to work with. In addition to that, the Clearsky DHI, Clearsky DNI, Clearsky GHI, and Fill Flag were removed from the dataset as well.

After the initial change to the dataset, there were three significant things that were done in order to preprocess the data. First, one-hot encoding was performed on the 'Cloud Type' feature of the dataset. One-hot encoding is the process of converting categorical data into a format that works best with machine learning algorithms. Essentially, a column is created for each possible category. Next, every column is assigned either a 0 or 1 depending on the category of the data point [7]. Figure 3 shows how one-hot encoding transformed the 'Cloud Type' feature of the dataset.



Cloud Type
4
4
4
4
4

cloud_type_0	cloud_type_1	cloud_type_2	cloud_type_3	cloud_type_4	cloud_type_6	cloud_type_7	cloud_type_8	cloud_type_9
0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0

Fig. 3. One-Hot Encoding Example

Second, feature scaling was applied to the data so that the distribution was close to a normal distribution (with a mean of 0 and a standard deviation of 1). Each of the features have a different amount of variance, and scaling accounts for these variances [8]. The scaling technique used for this data set was standardization, which was performed using StandardScalar from sklearn.preprocessing. The formula used in StandardScalar is:

$$x' = \frac{x - \mu}{\sigma}$$

where x' is the scaled value, x is the unscaled value, μ is the mean of the data, and σ is the standard deviation.

The last preprocessing step applied to the dataset was splitting it into train and test data. This was done using train_test_split from sklearn.modelselection [8]. For this project, 20% of the data was used for testing and 80% was used for training. In addition to that, the dataset was further split into the X dataset and the y dataset. The y dataset includes only the surface albedo data. The X dataset includes everything, except for the surface albedo features, from the original dataset.

III. Random Forest Regressor

To make surface albedo predictions, we used a random forest regressor. The random forest algorithm uses a number of loosely related trees to make a prediction for surface albedo based on the input data. Essentially, each tree makes it's own prediction independent of the other trees. Then, the predictions of all the trees are averaged. This is shown in the diagram in Figure 4.

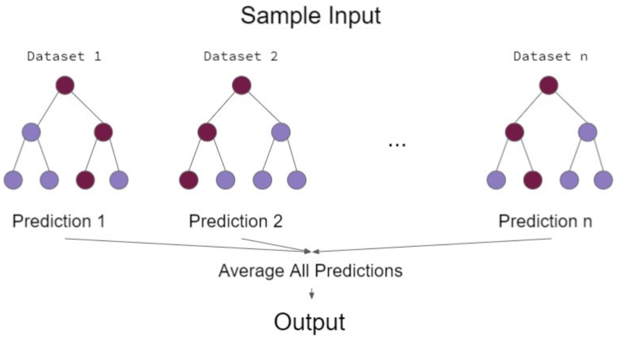


Fig. 4. Random Forest Regressor Diagram

To use the algorithm, we used RandomForestRegressor in sklearn.ensemble. The method takes in several parameters. In order to apply the algorithm effectively, we used hyperparameter tuning to find out the best value for each of the parameters. The parameters of RandomFoestRegressor include n_estimators, criterion, and max_depth. For the other parameters we simply used the default values [8].

A. Number of Trees

n_estimators is an integer representing the number of trees used in the random forest algorithms. The default value for n_estimators is 100 [8]. The graph below in Figure 5 shows how the RMSE value changes with varying inputs for the n_estimators parameter.

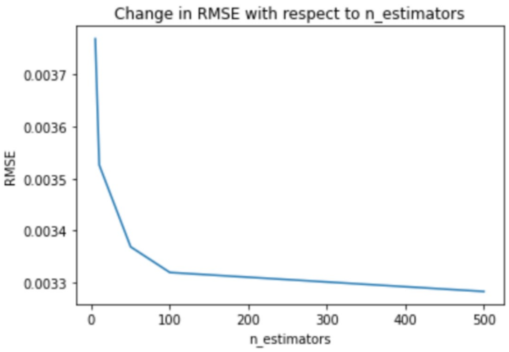


Fig. 5. Change in RMSE with Respect to the Number of Trees Graph

B. Criterion

criterion is a string describing how to measure the split quality. The two choices for this parameter are ”mse”, mean sqared error, and ”mae”, mean average error. Since ’mse’ squares the errors, it is more sensitive to outliers. It is also the default parameter input [8]. The graph below in Figure 6 shows how the RMSE value changes with varying inputs for the criterion parameter.

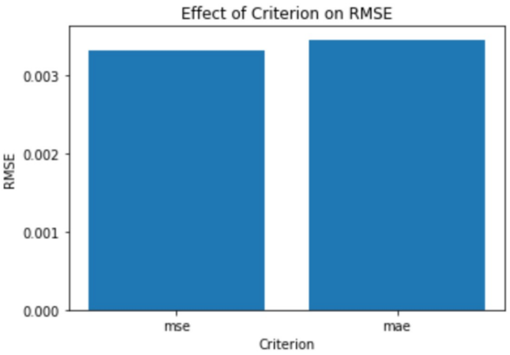


Fig. 6. Change in RMSE With Respect to Function to Measure Split Quality

C. Maximum Depth of Tree

max_depth is an integer that represent the maximum depth of the tree. The default value for this parameter is ”None”. When the default value for the parameter is used, the decision tree is expanded until all its leaves are pure or until all the leaves contain less than min_samples_split [8]. The graph below in Figure 7 shows how the RMSE value changes with varying inputs for the max_depth parameter.

D. Maximum Number of Features

max_features represents the maximum number of features to consider when determining the best split. The default input for the parameter is ”auto” [8]. The graph below in Figure 8 shows how the RMSE value changes with varying proportions for the max_features parameter.

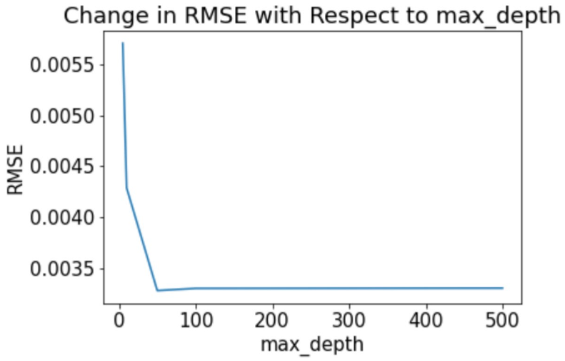


Fig. 7. Change in RMSE With Respect to the Maximum Depth of Random Forest Trees Graph

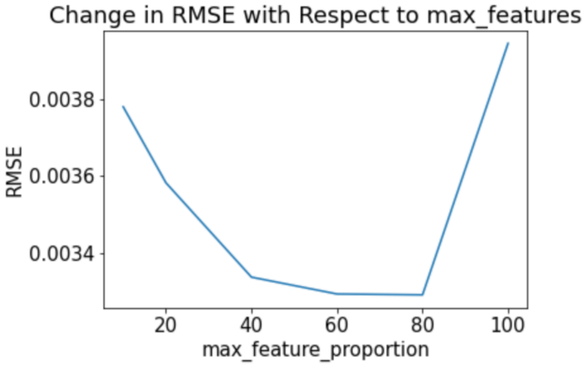


Fig. 8. Change in RMSE With Respect to the Maximum Depth of Random Forest Trees Graph

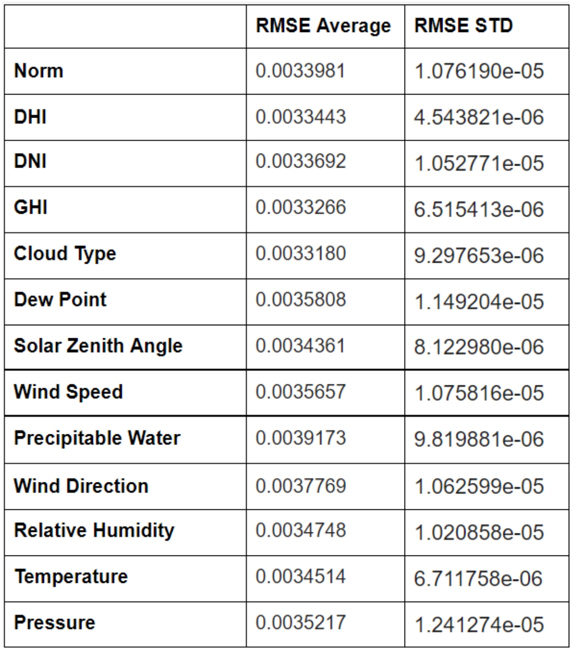
E. Hyperparameter Tuning

Hyperparameter tuning allows us to determine the optimal values for the parameters of RandomRegressor. Rather than manually testing each set of parameter values, we take advantage of RandomizedSearchCV from sklearn.model_selection. Given a grid of parameter values, RandomuzedSearchCV determines the best set of parameter values to use [8].

F. Feature Importance

To determine the feature that has the most significant impact on the surface albedo prediction, we rely on feature importance. First, we manually determined the importance of each feature using the parameter values determined by using hyperparameter tuning. Essentially, we removed each feature from the NSRDB dataset and looked at how the RMSE value changed. A significant increase in the RMSE value indicates that the feature is important for making surface albedo predictions. The table below in Figure 9 demonstrates how the RMSE value changes when each feature from the NSRDB dataset is removed.

The graph below in Figure 10 shows that same information as a bar graph. It indicates that the precipitable water and wind direction are the most important features for the surface albedo



	RMSE Average	RMSE STD
Norm	0.0033981	1.076190e-05
DHI	0.0033443	4.543821e-06
DNI	0.0033692	1.052771e-05
GHI	0.0033266	6.515413e-06
Cloud Type	0.0033180	9.297653e-06
Dew Point	0.0035808	1.149204e-05
Solar Zenith Angle	0.0034361	8.122980e-06
Wind Speed	0.0035657	1.075816e-05
Precipitable Water	0.0039173	9.819881e-06
Wind Direction	0.0037769	1.062599e-05
Relative Humidity	0.0034748	1.020858e-05
Temperature	0.0034514	6.711758e-06
Pressure	0.0035217	1.241274e-05

Fig. 9. Table of Average and Standard Deviation of RMSE After Feature Removal

prediction. On the other hand, the cloud type and irradiances actually make the predictions worse.

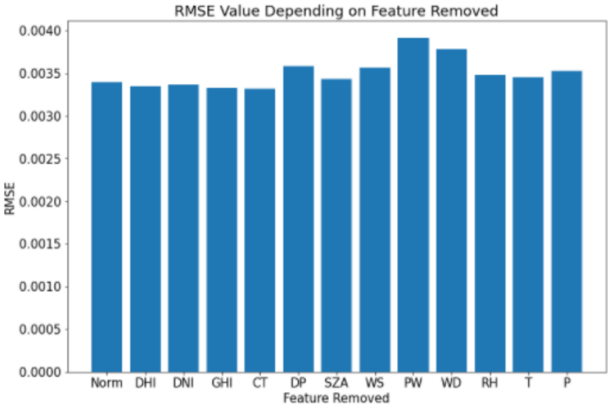


Fig. 10. Manually Determining Feature Importance

IV. Principal Component Analysis

Principal component analysis (PCA) is a dimension-reduction statistical technique that transforms a large set of variables in a dataset to a smaller set. The advantage of using PCA is that it simplifies the dataset and therefore, reduces the training time of the model. The disadvantage is that there is a trade off of the accuracy for the reduced training time. The goal is to find a good balance between them [9].

The graph below in Figure 11 demonstrates the importance of each component of the NSRDB dataset. The x-axis shows the components (although the name of the component is not known here) and the y-axis shows the proportion of variance caused by each of the components. The graph shows that after around the fourth component, the graph begins to level off. This indicates that using the first four features of the dataset may be sufficient to accurately predict the surface albedo.

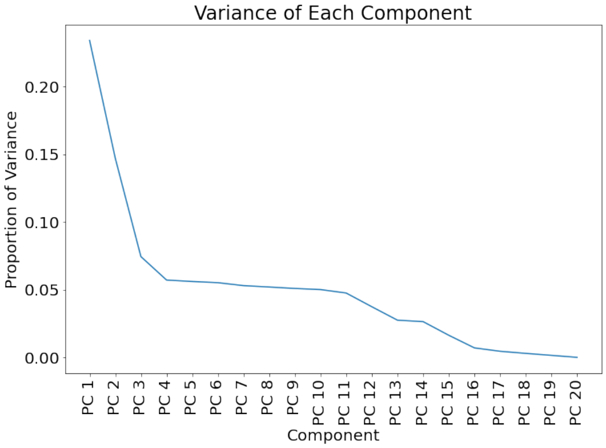


Fig. 11. PCA Variance Plot

The RMSE plot in Figure 12 demonstrates how the RMSE value changes as the number of components used changes. Similar to what was shown in the variance plot, the graph below shows that 4-5 components may be sufficient to predict the surface albedo due to the fact that after that the RMSE decreases very little after that.

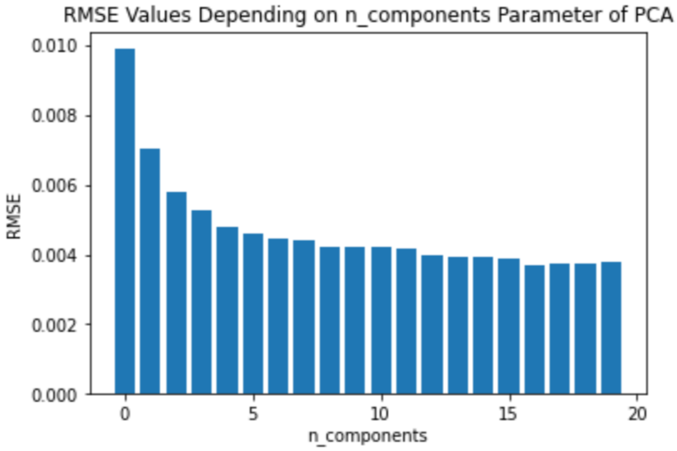


Fig. 12. Change in RMSE with Respect to the Number of Components Graph

V. Conclusion and Future Work

From PCA, we were able to determine that 4 components were sufficient to make accurate predictions for the surface albedo. From the feature ranking, we were able to determine that those 4 components are precipitable water, wind direction, dew point, and wind speed. The lowest RMSE value that we were able to achieve was 0.00370587 on the test data.

In the future, there should be more work done on how all the features in the dataset relate to each other. In addition to that, there should also be some work done on how PCA and autoencoders compare with each other for this dataset.

ACKNOWLEDGMENT

This work is partly sponsored by NSF REU Grant 1854273 and ASU SenSIP Center.

REFERENCES

[1] Y. Kotak, M.S. Gul, T. Muneer, and S.M. Ivanova, “Investigating the Impact of Ground Albedo on the Performance of PV Systems,” CIBSE, Technical Symposium. London, UK 16-17, April 2015.

[2] B. Marion, “Albedo Data Sets for Bifacial PV Systems,” 2020 47th IEEE Photovoltaic Specialists Conference (PVSC), 2020.

[3] M. Gul, Y. Kotak, T. Muneer, and S. Ivanova, “Enhancement of Albedo for Solar Energy Gain with Particular Emphasis on Overcast Skies,” Energies, vol. 11, no. 11, p. 2881, Oct. 2018.

[4] C. T. Clack, “Modeling Solar Irradiance and Solar PV Power Output to Create a Resource Assessment Using Linear Multiple Multivariate Regression,” Journal of Applied Meteorology and Climatology, vol. 56, no. 1, pp. 109–125, 2017.

[5] S. Rao, S. Katoch, V. Narayanaswamy, G. Muniraju, C. Tepedelenlioglu, A. Spanias, P. Turaga, R. Ayyanar, and D. Srinivasan, “Machine Learning for Solar Array Monitoring, Optimization, and Control,” Synthesis Lectures on Power Electronics, vol. 7, no. 1, pp. 1–91, 2020.

[6] A. Sharma, “Decision Tree vs. Random Forest - Which Algorithm Should you Use?,” Analytics Vidhya, 12-May-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/. [Accessed: 07-Jun-2021].

[7] M. Lukic, “One-Hot Encoding in Python with Pandas and Scikit-Learn,” Stack Abuse, 06-Apr-2020. [Online]. Available: https://stackabuse.com/one-hot-encoding-in-python-with-pandas-and-scikit-learn. [Accessed: 12-Jul-2021].

[8] M. Blondel, M. Brucher, L. Buitinck, D. Cournapeau, N. Dawe, S. Du, V. Dubourg, E. Duchesnay, A. Fabisch, V. Fritsch, S. Ghosh, A. S. Gollonet, C. Gorgolewski, J. Grobler, B. Holt, A. Joly, T. R. Jones, K. Kastner, M. Kumar, R. Layton, W. Li, P. Losi, G. Louppe, V. Michel, J. Millman, A. Passos, F. Pedregosa, P. Prettenhofer, V. Rajagopalan, J. Schreiber, J. Vanderplas, D. Warde-Farley, and R. Weiss, “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[9] I. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 13-Apr-2016. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202. [Accessed: 12-Jul-2021].