Predicting Solar Energy Output On Meteorological Time-Series Data Using Machine Learning

Caleb Harrison, Phadungsak Tubuntoeng, Xudong Liu
University of North Florida
John E. Mathews Jr. Computer Science
UNF Dr., Jacksonville, FL 32224
{n01304660, n01377880, xudong.liu}@unf.edu

Abstract
Solar energy production using photovoltaic (PV) systems is increasingly popular as a source of renewable energy for numerous applications. However, there is a main challenge with solar energy, namely, the unpredictability of its energy output. Therefore, accurate short-term predicting of the power output for PV systems is essential for effective decision making in power grid management. To this end, this paper focuses on training selected machine learning models, both traditional regression models and deep recurrent neural networks, to accurately predict solar energy output on meteorological time-series data from the Alice Springs solar farm in Australia. These machine learning models include linear regression, gated recurrent unit, recurrent neural network, long short-term memory, and random forest regression. The results of these tests showed that simple ensemble methods can outperform powerful single models and that hyperparameter tuning can greatly improve the performance of a model.

Methodology
To ensure that the comparison to the original source paper is fair, data was acquired from the same source in Alice Springs, Australia via the same provider (DKA Solar Centre 2022)

In order to optimize the models presented in this paper, each model underwent hyperparameter tuning. In order to best capture which parameters are most effective for the models, select parameters were varied at specified intervals. The parameters considered and tested for each model are seen in the following sections accordingly.

Linear Regression
In the project, a linear regression model was trained to use as a baseline for comparison with other models. The hyperparameters selected for use were True for the Fit Intercept, True for Copy X, and False for Positive. All permutations of these boolean parameters were tested to find the best performing set.

Neural Networks
Once the baseline linear regression was finished, a series of Neural Networks (NN) were created. In this project, three hidden layers with a constant learning rate of 0.001 and the Adam optimizer were used for three different types of NN: LSTM, GRU, and SimpleRNN.

SimpleRNN
The next model considered was a simple recurrent neural network (RNN). To determine the appropriate hyperparameters, the activation function and units parameters were tested. The test values were sigmoid, tanh, relu, softmax, and none for the activation function and 2, 4, 12, 16, 32, and 64 for units. The values of sigmoid and 32 were chosen as the best parameters.

LSTM
The hyper parameters considered for use in this model were activation, recurrent activation, and return sequences. For hyperparameter tuning, the parameters tested were the activation function, recurrent activation function, and return sequences. The functions tested were linear, tanh, relu, sigmoid, and softmax while return sequences is a boolean parameter. The chosen values were softmax for activation, sigmoid for recurrent activation, and True for return sequences.

GRU

Copyright © 2024 by the authors.
This open access article is published under the Creative Commons Attribution-NonCommercial 4.0 International License.

The next model considered was a gated recurrent unit (GRU). The hyperparameters considered in testing were activation function with none, softmax, tanh, relu, and sigmoid; units with 2, 12, 16, 32, and 100; and return sequences which is boolean. The optimal parameter selections were found to be none activation function, 16 units, and return sequences set to true.

**Random Forest Regression** A random forest (RF) regression algorithm (Breiman 2001) was the final model tested for this paper. The RF received hyperparameter tuning similarly to the linear regression model via the `gridsearchCV` function. These hyperparameters were set such that `warm_start` was true, `bootstrap` was true, `criterion` was `friedman_mse`, and `max_features` was none. All other parameters used default values.

**Results Summary**

For a complete comparison of the tested models, the MAE and RMSE of the five algorithms are presented in 1. In this figure, 1a are the linear regression results, 1b are the RNN results, 1c are the LSTM results, 1d are the GRU results, and 1e are the RF results in graph form. These results show a comparison of the real power of the system at time index t compared to the predicted power at index t for a randomized selection of 75 indices. In 1f, it can also be seen that the RF outperforms each of the three neural networks. This is likely due to the ensemble model’s diverse number of decision trees that eventually generalize efficiently compared to a singular model that is used in each of the RNNs.

### Conclusion and Future Work

Solar energy produced using PV systems have seen an increase in usage as a power source for many applications but are confined by their unreliable power output. Various works in the literature have tried to use machine learning techniques such as SVM and ANN models to train regressors on PV data in hopes to accurately predict power output. In this paper, we explored the use of traditional regression models such as linear regression and random forest, as well as deep recurrent neural networks such as LSTM and GRU.

In our experiments, it was found that the optimized LSTM model significantly improved upon the one presented in the initial paper which failed to reach even the linear regression baseline for this project. This is likely due to the larger available data which provided a significant advantage to this project over the original work. Despite this significant improvement for the LSTM, the RF regressor proved to be even more reliable for predicting the output power. The output of the ensemble model proved to be far superior to the neural network’s output and shows that ensemble methods should be preferred over even more powerful single models.
References