Forecasting Solar Panel Power Outputs

Forecasting the Power Outputs of Photovoltaic Cells: a comparison of statistical, machine learning and deep learning models

This project demonstrates my skills using Python to build statistical models, machine learning models and deep learning models for multivariate time series forecasting and compare them using performance metrics and data visualisations.

Intro and Background

With the effects of climate change increasing (The Causes of Climate Change, 2023); diminishing fossil fuel resources (Shafiee and Topal, 2009); and rising energy costs (Guan et al., 2023), global energy production will become more reliant on renewable energy sources such as PV installations (Hu et al., 2016). Worldwide, countries are turning to renewable energy sources to combat these issues.

Forecasting the power outputs of PV systems is useful for several reasons:

* Aids in meeting government renewable energy goals.
* Helps solar panel companies predict lifespans and payback periods leading to economic benefits for the companies and prosumers.
* Allows grid companies to prepare for potential high periods of generations from these systems.

The technical process followed during this project is outlined in Figure 1.

Process

**Analyse how the residential PV systems power output changes with time and analyse weather features effects on power output.**

The distributions for each variable were examined by the density plots and box plots shown in Figure 2. The Shapiro-Wilk test for normality was used alongside these visualisations to test if the target variable of *Monthly Power Generation (kWh)* had a normal distribution. The result from this determined the target was not normally distributed.

Therefore, Spearman’s correlation was used to test the correlations between the variables and the target, Figure 3. These relationships are plotted in Figure 4 and the Spearman’s correlations and the corresponding p-values are displayed in Table 1.

**Incorporate target lags and weather data to build three long-term, multivariate forecasting models: statistical, machine learning and deep learning.**

**Use predictions for twelve months to compare results using plots and performance metrics.**

The models were used to make predictions for the twelve-month period at the end of the dataset. The predictions from the models were compared using three performance metrics: RMSE, MAE and R2, the results of these are show in Table x. The LSTM-RNN produced the best values for the RMSE and MAE and the second-best value for the R2, with the SARIMAX II model producing the best R2 result. The model predictions were also compared using visualisations. Figure x shows each of the predictions made by the models plotted next to the true power output values for that period. Both SARIMAX models and XGB model without normalisation have a similar pattern to their predictions that mimics the shape of the true values but overestimates the power outputs for the 04/23 – 06/23 and the 08/23 months. Normalisation appears to stop the XGB model overestimating the power outputs. This model gives close estimates of power output up to the 06/23, then starts to underestimate. The LSTM-RNN model produces a curve to estimate the power output, following the pattern of the true values less closely than the other models, but gives more accurate predictions overall. Figure y shows linear regression for the predicted monthly power outputs compared against the true outputs. The R2 values, Table x, correspond to these. The more accurate the model predictions, the closer the points lie to the line. Models appear to make better predictions when the true power output is low, these low power output values correspond to the winter months. However, the winter months for the test set lie the closest in time to the end of the training set, which may be causing there more accurate predictions. Future work could use different length horizon times for a model to evaluate whether winter months or values closer to the end of the training set are predicted more accurately.