Report: Solar Production and Load Prediction Competition

Model Training (report)

Asad Irfan

**1. Introduction**

As our objective was to build energy generation and load forecasting models. The activities that have been followed in this report include data preprocessing technique, feature engineering technique, model selection technique, and model performance evaluation.

**2. Data Preprocessing**

*1. Data Loading and Merging:*

Datasets:

*train\_data. csv*: It preserves information concerning energy generation and the load factor.

*systems\_new. csv*: Contains system metadata.

Combined both the datasets on system\_id and merged the metadata with the main dataset. Followed by the removal of some unnecessary columns.

*2. Timestamp Handling:*

Pulled out year, month, day, hour, minutes from the concatenation of timestamp.

Followed by removing the time stamp column after data extraction step was done.

*3. Feature Engineering:*

Different type features were extracted from the available dataset . The list is below.

* **Day of Week & Weekend Indicator**

In order to capture weekly patterns, new features day\_of\_week and is\_weekend were derived.

* **Time of Day & Season:Time of Day & Season:**

Developed time\_of\_day and season to differentiate between changes during a daily period and the changes in the four seasons.

* **Time Since Start:**

In the last step, computed time\_since\_start as the time which passed from a particular start date.

* **Rolling Means:**

Used 1hr moving average to trend generation\_W and load\_W by calculating the 1hr moving average for each of these variables in order to remove short term volatility.

* **Peak Production:**

Defined the variable peak\_production so as to be able to differentiate between the peak and the non-peak months.

* New features f1, f2, and f3 introduced from load\_capacity, panels\_capacity and the time-specified variables.

**Total features in the final file.**

* generation\_W
* load\_W
* connection\_type
* location
* panels\_capacity
* load\_capacity
* year
* month
* day
* hour
* minute
* time\_standard
* peak\_production
* day\_of\_week
* is\_weekend
* time\_of\_day
* season
* generation\_W\_rolling\_mean\_1h
* generation\_to\_capacity\_ratio
* time\_since\_start
* f1
* f2
* f3

*4. Label Encoding:*

For distinct categorical features connection\_type, location, applied LabelEncoder.

Used the encoder and saved it for future uses using joblib.

*5. Data Scaling:*

So basic type of scaling I used throughout the competition was StandardScaler and MinMaxScaler.

Used Min Max and Standard scaler to normalize features as well as target variables.

Now divide the data into training data set and testing data set using test train split.

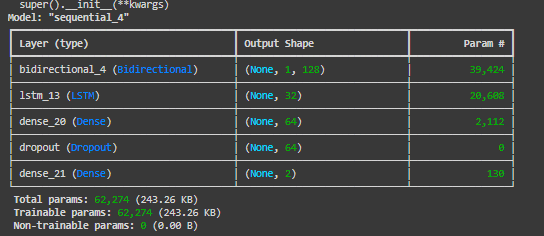
**3. Model Selection and Training**

1. Model Architecture:

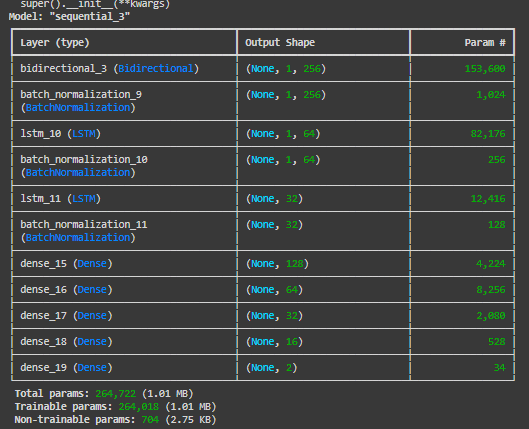
I tried different architectures to train model but among the best were the following three.

Bidirectional LSTM Model:

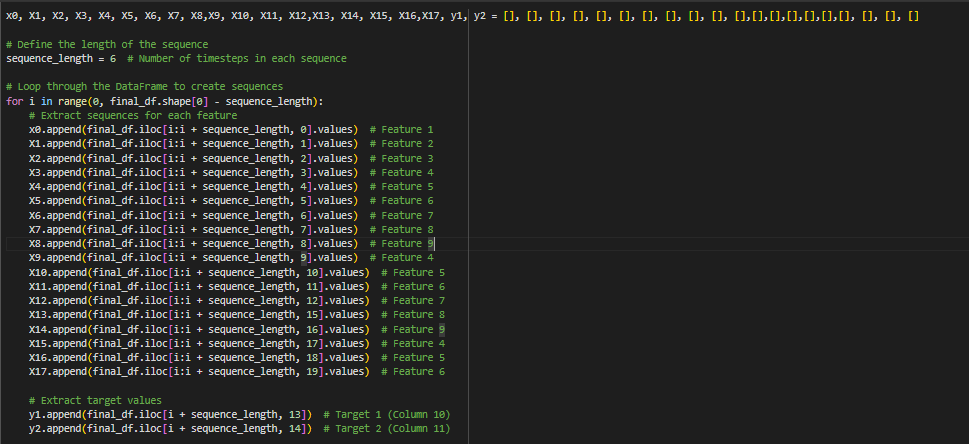
Layers:



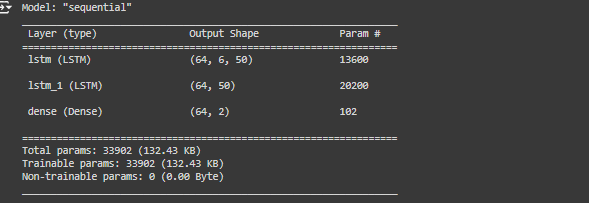
Layers:



So another model which I used was to make a sequences of the data with a time step of 6 which means the model will take the last 6 values and predict the generation and load as a seventh value. Below is a screen shot of the making time steps.



The data was converted in numpy arrays to preprocess using both Standard Scaler and MinMax Scaler in different files to evaluate the results. The architecture used to train this approach was also LSTM and the architecture is as below.



**3. Training**

Process:

Obtained the trained models with using the training dataset with the validation split.

Early-Stopping callback was used to avoid over-fitting and restore the best weights among all the iterations.

**4. Model Evaluation:**

Cross checked the trained models with the test set.

Metrics used: “MAE” and “MSE” is a measure of the accuracy, and accuracy is used to denote how right or wrong the developer is in his or her predictions.

**5. Results and Insights**

Model Performance:

From the description above it is clear that the new developed advanced LSTM model has relatively lower MAE and therefore has better performance as compared to the previous Bidirectional LSTM model.

As expected both models yielded decent scores but the second model was somewhat more accurate and possessed more generalization to unseen data.

**Feature Importance:**

Including features like; hour, season, rolling greatly enhanced the models by adding temporal dimensions as well as reducing noise.

**Challenges:**

*Missing values:*

Challenges included missing values, which need to be dealt with appropriately, and scaling of data.

Data fiduciary responsibility and feature engineering were some of the major constraints faced while working on the project.

*Right Architecture:*

Choosing right architecture for my lstm was also a very difficult task but after trying multiple architectures I finally reached the above three according to the best architectures.

*Over-fitting:*

Overfitting was one of the biggest problem while training the model.

**5. Conclusion**

The contest presented different aspects of data preprocessing and feature selection as well as model design. The proposed complex LSTM-based model provided suitable results in terms of energy generation and load forecasting and it was depended on proper feature selection, and hyperparameters’ optimization.

Inference Code details

The process that has been applied in order to test the test\_masked. csv file was straightforward. First of all the trained model was saved using the model. It was trained to a save() function and then you load it again just to make prediction with the model. predict() function. First, similar to the previous steps of the analysis, the inputs were shaped to meet the requirements of the employed model. After making predictions, the results obtained were transformed to a DataFrame and the output values substituted the previous generation\_W and load\_W. On this, the CDS file was then ready for uploading to the server computer through the virtual channel.

**Note:**

A separate file has been provided for the visualizations I used during the competition.