Predicting annually varying production of electricity by renewable sources depending on spatio-temporal and climatic conditions

(Avikal Sagar as6804, Neha Gosula nbg2122)
Group 3
(https://github.com/avikalsagar/Predicting_Annual_Variability_Renewable_Energy)

1. Introduction

About 4,231 billion kWh, or 4.23 trillion kWh, of electricity were produced in the US at utility-scale electrical producing facilities in 2022. This power was produced using fossil fuels, which include natural gas, coal, petroleum, and other gasses, to the tune of about 60% while the share of renewable energy was about 21%. About 1.65 billion metric tons—1.82 billion short tons—of carbon dioxide (CO2) were released in 2022 as a result of the utility-scale electric power plants' (plants with at least one megawatt of electric generation capacity) total annual net generation of electricity in the United States of 4.23 trillion kWh from all energy sources. This equated to emissions of CO2 of almost 0.86 pounds per kWh.

In order to promote low-emission, sustainable development, numerous nations are setting high goals for their electrical supply in terms of renewable energy. Planning and operations for the power system will need to adapt in order to accomplish these aims because solar and wind energy have a tendency to be more unpredictable and variable than conventional sources. The process of creating effective methods for supplying variable renewable energy (VRE) to the grid is known as grid integration. The cost-effectiveness of adding VRE to the power system is maximized by good integration techniques, which also preserve or improve system stability and dependability. The integration of renewable energies into the grid appears to be promising with the use of machine learning algorithms.

Predicting annual variability in renewable energy production involves considering various factors that affect generation, such as weather patterns, geographic location, technology efficiency, and system design. Artificial Intelligence & Machine learning in weather forecasting combined with historical data analysis seem to be showing increasingly good results. Reliability of weather forecasts is important when predicting generation of renewable energy. While wind power generation is dependent on wind speeds, solar power generation is dependent on the availability of sunshine. Forecasting methods and sophisticated weather models aid in the anticipation of these elements. Trends in yearly variability can be understood by examining past weather and energy production data. Better forecasts are possible by identifying patterns in seasonal variations, long-term weather cycles, and their effects on the production of renewable energy. By evaluating enormous volumes of data, machine learning algorithms and AI models can improve prediction accuracy. In order to anticipate the output of renewable energy, these models can combine real-time information and learn from patterns in prior data. Predictions are frequently more accurate when various forecasting techniques—such as statistical models, physical models, and machine learning—are combined into hybrid models. These models are able to simultaneously take into account multiple affecting elements.

While these methods improve the accuracy of predicting annual variability in renewable energy, it's important to note that forecasting inherently involves uncertainties due to the complexity of

natural systems and technological dependencies. Continual advancements in technology and data analysis techniques aim to reduce these uncertainties and improve predictive capabilities in the renewable energy sector.

The focus on advancements in the renewable energy sector is more prominent than ever. Governments, investors and developers are expanding the use of existing forms of renewable energy and exploring the commercialisation of newer forms of renewable energy to increase the overall global supply. Predicting the amount of renewable energy that can possibly be generated is one of the most important aspects of renewable energy forecasting for grid operators. Renewable energy sources don't continuously provide electricity, in contrast to other power sources. To avoid generator losses due to constant ramping up and down, power firms must comprehend renewable energy predictability in order to manage resources effectively. Energy providers can match renewable energy generation and demand with the use of this information.

The majority of weather forecasting models use data from wide geographic areas, which may not necessarily be an accurate representation of the local environment. Much more detailed data may be processed by machine learning algorithms in the same amount of time, if not less. Forecasts produced with this accuracy and speed are more accurate and pertinent.
Then, using the same machine learning algorithms, one can forecast how the weather will affect the production of energy. Energy firms can decrease their use of fossil fuels if renewable energy sources yield higher-than-average levels, and vice versa. This adaptability is vital since on certain days wind power alone could generate half of the energy used in say, Texas, while on other days it barely produces anything.

2. Data

2.1 Data description

To predict annual variability, inspired by various advancements in using machine learning for applications in renewable energy, we use ERA5 Climate dataset to get weather related data and also, we use the generation mix database published by Southwest Power Pool (SPP) for the majority of the states served by SPP viz (North Dakota, South Dakota, Nebraska, Kansas, Oklahoma).

SPP data was selected for the years 2018, 2019, 2020, 2021 and 2022. The data showed the generation mix from coal, diesel, wind, nuclear, natural gas, hydropower, waste disposal services and other categories at a 5 minute interval. This dataset was further merged to get the mean hourly data values.

ERA5 is the fifth generation ECMWF reanalysis for the global climate and weather for the past 8 decades. Data is available from 1940 onwards. ERA5 replaces the ERA-Interim reanalysis. Reanalysis combines model data with observations from across the world into a globally complete and consistent dataset using the laws of physics. This principle, called data assimilation, is based on the method used by numerical weather prediction centres, where every so many hours (12 hours at ECMWF) a previous forecast is combined with newly available observations in an optimal way to produce a new best estimate of the state of the atmosphere, called analysis, from which an updated, improved forecast is issued. Reanalysis works in the

same way, but at reduced resolution to allow for the provision of a dataset spanning back several decades. Reanalysis does not have the constraint of issuing timely forecasts, so there is more time to collect observations, and when going further back in time, to allow for the ingestion of improved versions of the original observations, which all benefit the quality of the reanalysis product.

2.2 Data extraction

The SPP data was directly downloaded from their website for each of the years from 2018 to 2022: https://portal.spp.org/pages/integrated-marketplace-generation-mix.

The ERA5 Hourly Land Aggregate dataset was downloaded in our Python notebooks using the Google Earth Engine API. The dataset consists of raster images of multiple bands corresponding to the feature variables of our problem set for each geographic region in consideration. We first extracted these raster images, and then compressed them into numerical values per band using suitable modules of the EE library. They were then stored in arrays and finally concatenated at the end with their corresponding datetime values.

2.3 Data cleaning, merging and aggregation

As mentioned previously, the ERA5 data was on an hourly basis whereas SPP data was recorded at every 15 minute interval. The first step that we performed was to aggregate the data from 15-min intervals onto an hourly basis. Then, this was concatenated with the ERA5 data with date and time as the common column.

Once we had the combined data in place, we checked for null spaces or missing values over the entire dataset. In places necessary, we performed linear interpolation to impute values over these null spaces. In other cases, where the missing data in a particular column was less than 5% of the population, we decided to drop them completely.

Finally, we performed a few more basic data processing tasks such as removing duplicate entries and sorting the dataset in chronological order.

2.4 Data splitting into features and target variables

One of the most crucial tasks in the project – we had to carefully pick the input (features) and output (target) variables. Since we decided to first perform forward modeling followed by inverse modeling (for details, refer to section 3. methodology), we had to first pick the features and target variable for the former followed by feature and target variables for the latter. The format of splitting can be seen below:

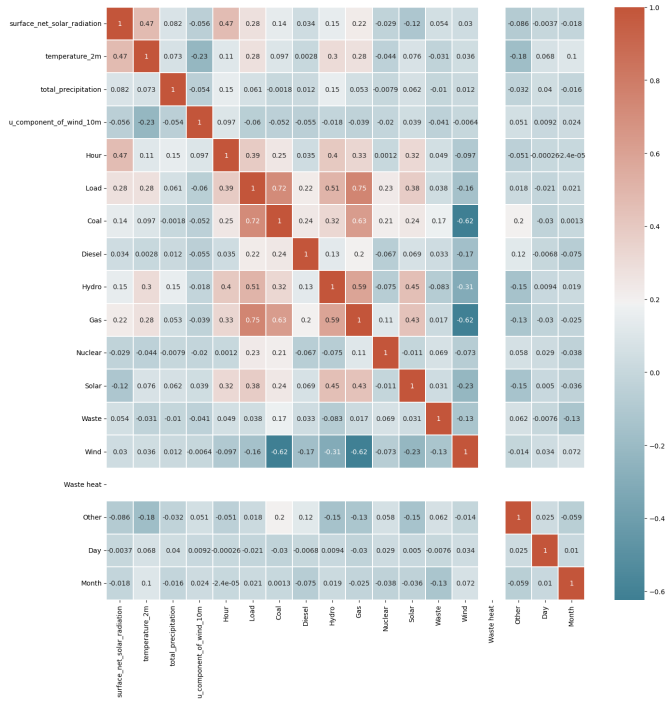
date_time	surface_net_solar_radiation (J/m2)	temperature_2m (K)	total_precipitation (m)	u_component_of_wind_10m (m/s)	Hour	Load (kW)	Coal	Diesel	Hydro	Gas	Nuclear	Solar	Waste	Wind	Waste heat	Other
01-01-2018 01:00	0	254.1925328	4.44E-08	1.074555688	1	435175.406	252979.6	8.8	8437.6	115428.1	24410.6	0	130.8	41819.9	0	326.9
01-01-2018 02:00	0	253.6020729	4.54E-08	1.023605341	2	433136.664	254174.5	0	6373.6	109497.4	24414.7	1.4	130.3	38397.5	0	326.2
01-01-2018 03:00	0	253.1105115	4.54E-08	0.942507374	3	432544.152	254355.6	0	6355.1	111666.6	24408.1	14.2	131.1	38098.6	0	325.1
01-01-2018 04:00	0	252.674437	4.54E-08	0.921039935	4	432570.416	255091.5	0	6351.8	109633.5	24413	54.7	131.9	37790.8	0	327.5
01-01-2018 05:00	0	252.2565711	4.54E-08	0.947823066	5	436211.546	253726	0	6511	119693.8	24413.1	91.1	131	35006.1	0	326.5
01-01-2018 06:00	0	251.8850234	4.55E-08	0.992228593	6	444097.252	252736.8	0	7060.1	124755.1	24409.5	119.3	130	31979.1	0	326.1
01-01-2018 07:00	0	251.0979631	4.87E-08	1.270438239	7	451608.626	254557.8	0	14325.9	125322	24405	120	129.8	26786.2	0	325.5
01-01-2018 08:00	0	250.9491463	5.21E-08	1.154960406	8	457082.791	253146.5	0	15365.4	135910.8	24409.2	120.6	130.9	22166.1	0	324.1
01-01-2018 09:00	0	250.9231513	5.66E-08	0.980637111	9	456300.098	252846.2	0	15609.1	141875.3	24415.6	140.2	131.3	18478.4	0	319.4
01-01-2018 10:00	0	250.8872876	5.99E-08	0.817843779	10	454706.032	253464.5	0	17988.4	140895.7	24408.2	374.6	131.2	14866.7	0	326.4
01-01-2018 11:00	0	250.7931688	7.03E-08	0.715794169	11	450743.393	254796.3	0	18640.3	142523.1	24411.3	689.1	130.6	10997.5	0	331.2
01-01-2018 12:00	0	250.634411	7.75E-08	0.651111093	12	442869.379	256502.5	0	18140.7	131781.1	24407.9	829.7	130.6	8667.5	0	331.3
01-01-2018 13:00	0	250.4477124	9.79E-08	0.615711556	13	432533.499	255207.3	0	17760.9	122489.3	24409.2	921.3	134.5	8296.9	0	334
01-01-2018 14:00	3258.849679	250.2291598	1.10E-07	0.576738107	14	421439.39	254967.1	0	16582	110905	24410.6	987	135.7	8603.7	0	334.3
01-01-2018 15:00	134603.8582	250.4970835	1.29E-07	0.459749966	15	414368.252	257324.2	0	16460	103111.4	24410	1035.3	130.1	9492.2	0	338.2
01-01-2018 16:00	563713.8486	251.6519328	1.46E-07	0.407616866	16	413139.818	258761.8	0	16428.6	97369.3	24413.2	1328.1	92.2	10985.7	0	339.7
01-01-2018 17:00	1279608.915	253.4306547	1.55E-07	0.349769053	17	422849.139	258365.8	0	15906.4	105773.9	24409.5	1526.9	127.5	15099.2	0	340.4
01-01-2018 18:00	2192052.612	255.9247335	5.81E-07	0.284463773	18	451009.111	254550.5	0	18978.4	123079.8	24410.2	465.4	137.3	21037.6	0	335.7
01-01-2018 19:00	3183766.135	256.7529728	5.92E-07	0.249587721	19	467113.266	257909.8	0	19537.9	124706.1	24408	91	138.1	33571.5	0	331.8
01-01-2018 20:00	4129204.804	257.847685	6.13E-07	0.06330865	20	467946.328	259925.1	0	18595.4	121443.8	24413.2	77.9	127.7	44570.5	0	331
01-01-2018 21:00	4905272.982	258.4523307	6.52E-07	-0.158447149	21	466214.525	258928.3	0	18463.7	118559.2	24416.9	79.2	133.6	47181.8	0	329.5
01-01-2018 22:00	5415246.184	258.433371	6.91E-07	-0.347887634	22	460799.838	257310.4	0	15370.8	117338.7	24413.3	85.7	136.1	48599.9	0	333
01-01-2018 23:00	5607371.381	257.601446	7.17E-07	-0.357761532	23	447539.485	255004.8	0	9118.2	109794.8	24413.7	79.5	136.7	49190.6	0	332.3
01-02-2018 00:00	5615334.158	256.4927681	7.29E-07	-0.136622847	0	436122.393	253247.6	0	6538.3	106268.1	24403.9	106.5	133.7	48569.9	0	331.6

The columns highlighted in green correspond to the features for the forward modeling phase, whereas the column ‘Load (kW)’ highlighted in red corresponded to the target variable in the forward modeling phase which was then taken as the feature in the inverse modeling phase. The columns highlighted in orange are finally the target variables in the inverse modeling phase.

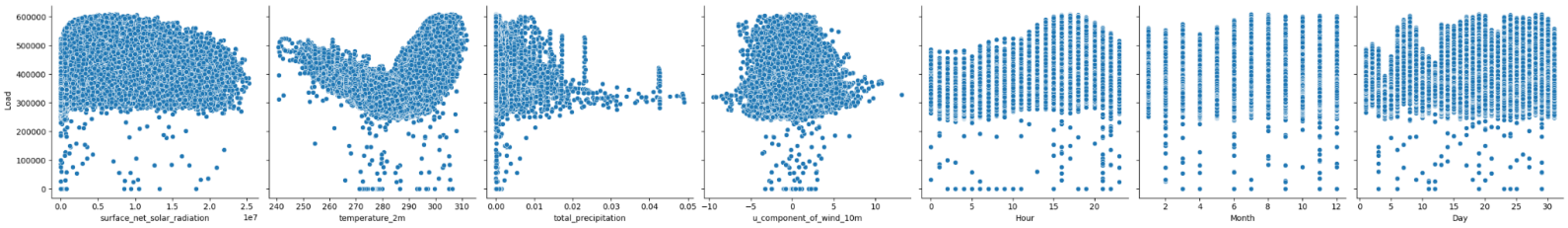
The logic behind this split is that all the climate variables along with seasonality (such as date, time and months) are predictors of the demand load requirement of electricity in each household. For example, during the dark hours of winter months, when the temperature goes below freezing, there is an expected increase in demand loads due to people using heaters and warmers more often than during spring or summer months. On the other hand, to know the precise split of the source of electricity that caters to this required peak demand loads, the demand load along with seasonality itself turns into predictor variables. Hence we chose an forward + inverse modeling procedure with such variables’ split.

2.5 Exploratory Data Analysis

A key step in this project was to visualize the processed and clean data that we have in hand for understanding it a bit more and receiving additional clarity on the correlation as well as the distribution of some of the important variables. For the same, we first plotted a Pearson’s correlation matrix for all variables as shown below:



Next, to visualize the distribution of each variable, we chose to make pair plots as shown below:



3. Methodology

3.1 Forward modeling

Given that the first phase of modeling involves multi-input and single-output, any classical Machine Learning model with suitable hyperparameters would suffice in achieving a great accuracy. SVM, Decision Trees and Random Forests are some of the popular techniques amongst them. For this project, we chose the Random Forest model along with some common boosting techniques such as XG Boost.

We split the features and target variables each into training and test sets with a 3:1 ratio. Random Forests are great in dealing with multiple inputs at various magnitudes. Hence, we did not perform any standardization or normalization. First, we implemented the default random forest regressor from the scikit-learn library using the predefined hyperparameters. Next, we set a range of values for each hyperparameter and using Random Search CV, we performed the regression analysis for each combination of hyperparameters in this space. The Random Search CV returns the hyperparameters that performed the best by achieving higher accuracy or lower errors. Finally, we boosted our forest using the XGBoost algorithm.

The hyperparameters and results of our Random Forest model is as follows:

Model	Number of decision trees (or range provided)	Max depth of tree	Minimum sample splits per node	Minimum leaves per node	R-2 score
Default	100	None	2	1	0.81
Fine-tuned (Randomized Search*)	0-100 (best = 19)	[None, 5, 10, 5] (best = 20)	[10, 15, 20] (best = 10)	[4, 7, 10] (best = 30)	0.85

*5 folds of 10 candidates each

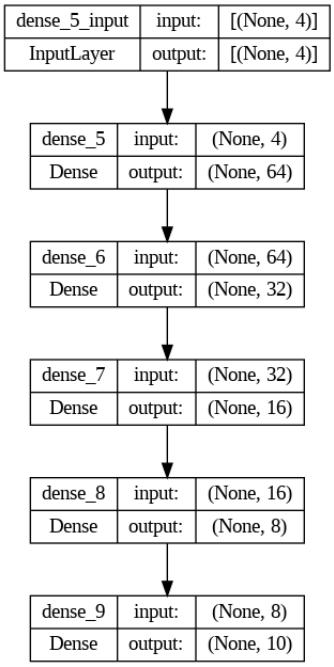
The XGBoosted random forest, just like we thought, outperformed the fine-tuned random forest by achieving an R-2 score of 0.89 with the following hyperparameters:

```
XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=0.9, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.18, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=10, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=98, n_jobs=None,
             num_parallel_tree=None, random_state=42, ...)
```

3.2 Inverse modeling

Once we have the forecasted demand load as a function of climate variables, the next step is to use this demand load as an input and find the distribution from each source of energy to cater to it. In simpler words, we will be performing a single-input multi-output predictive modeling which is sometimes called inverse modeling. Here, unlike in the previous case, classical Machine Learning models do not perform well. Only Deep Learning algorithms can find trends and patterns in complex datasets, especially if there's only a single input.

Hence, we decided to use a deep neural network algorithm. The architecture of our model is shown below:



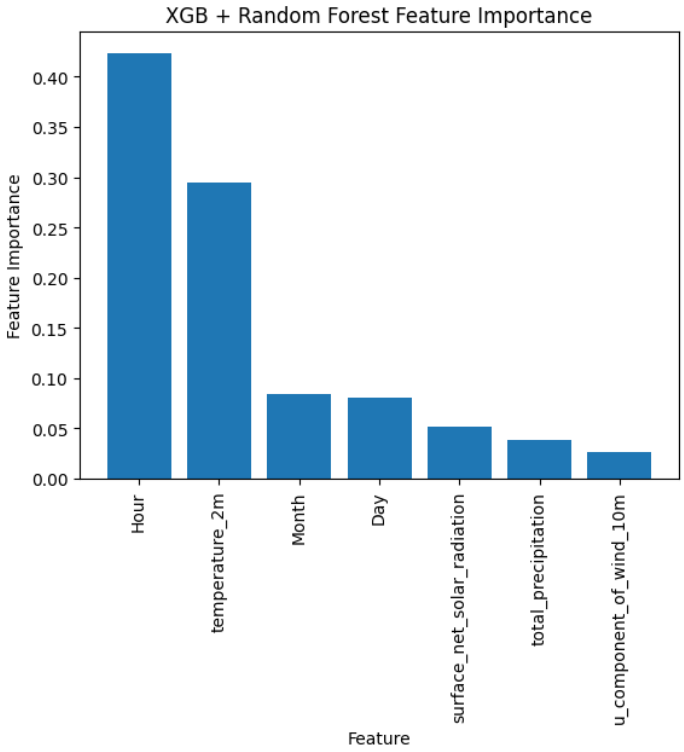
We have an input layer consisting of 4 inputs - the demand load, and we additionally integrated seasonality into it (year, month, hour and day) so that the model learns stronger. The, we built four hidden layers with 64, 32, 16 and 8 neurons respectively while using the rectified linear unit activation function. Finally, we used an output layer with 10 neurons (pertaining to each target variable) with the linear activation function.

To train the model, we used a batch size of 256 and the number of epochs that we chose were 25 (after multiple iterations of experimenting). The chosen optimizer was Adam and the error metric was MSE.

4. Results

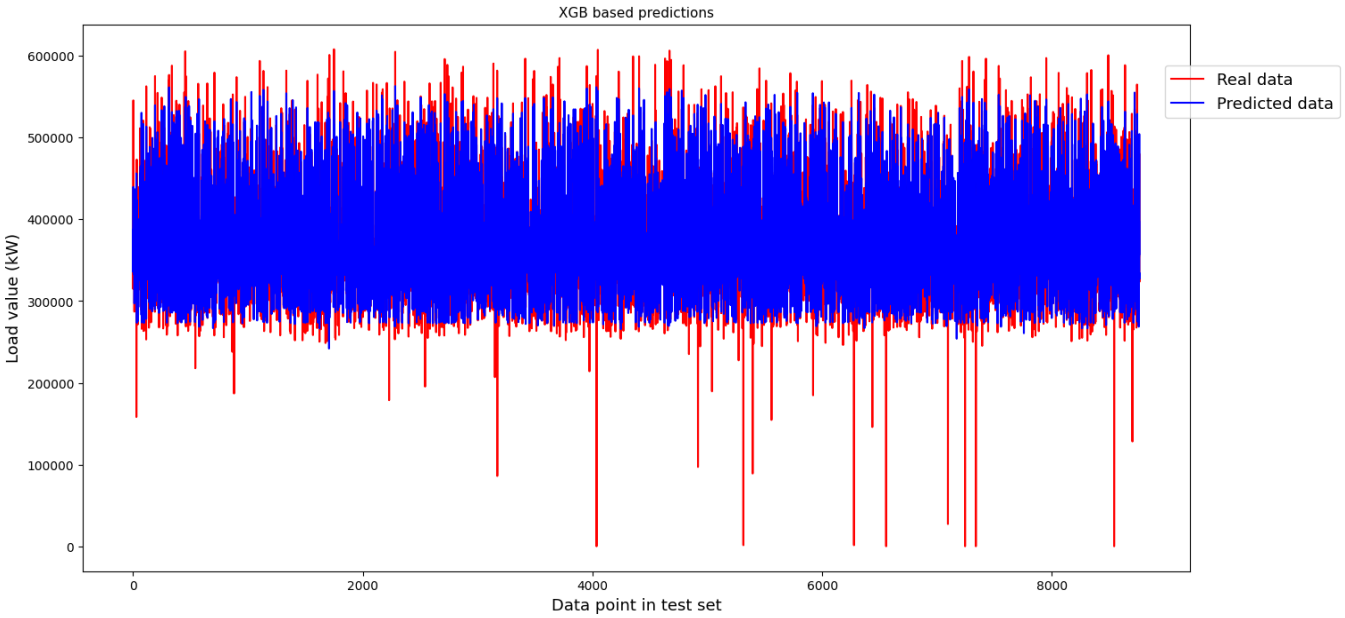
4.1 Forward model

As mentioned in section 3.1 - the XG Boosted Random Forest achieved an impressive R-2 score of 0.89. Since they were built on the hyperparameter-tuned model, we were able to confirm that they were not overtrained. After validating the R-2 score, we printed the most important climate variables that were used to forecast the demand load as shown below:



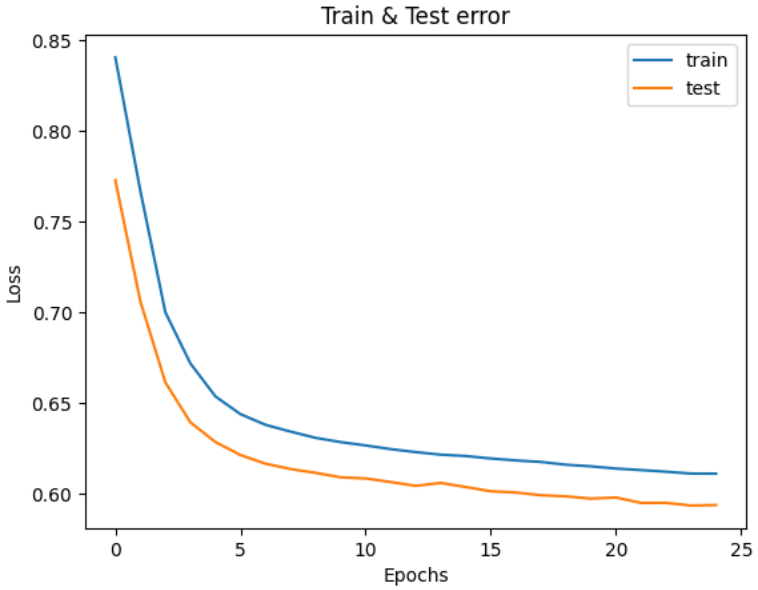
As seen from above, the most important features used in forward modeling predictions were seasonality (time of day) as well as temperature in the region. Logically speaking, it makes sense. The demand for electricity is highest during certain peak hours of the day, and even higher during either freezing conditions (which increases heating requirements) or in the summer (which increases air conditioning requirements). Other climate variables such as precipitation and wind speeds had very little influence.

Finally, we visualize the performance of the model's forecast against its actual values on the test set below:



4.2 Inverse model

As mentioned in section 3.2, we used only 25 epochs because we could see signs of the loss function for both training and test sets converge. Continuing furthermore with more epochs could have resulted in overfitting.



Moreover, a loss (MSE) of as little as 0.60 towards convergence was an indication of the model performing well - even on data where there were more output than input.

4.3 Overall

The main aim of the project was to build a forecasting model that takes in climate variables and spatial coordinates for a region (in Southwest US), and based on these conditions, predicts:

(a) total demand load of electricity
(b) the distribution of electricity generated by each source of fuel

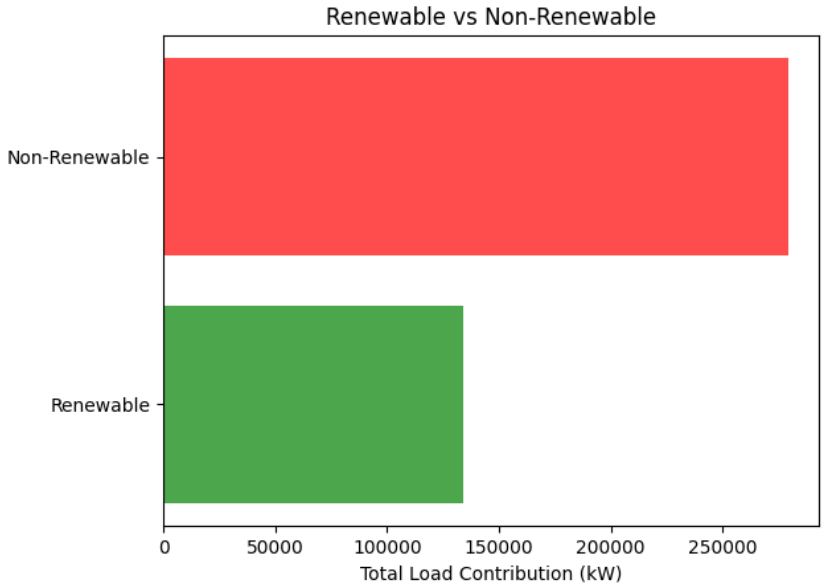
(c) the final distribution by fuel type (renewable v/s non-renewable)

Once we built both the forward and inverse models, the next step was to deploy it. We built a simple model calculator using jupyter python widgets that takes in input through text boxes and returns the results within the output boxes (for numerical ones) and plots (for visualizing the renewable v/s nonrenewable distribution) as shown below:

Temperature (K):	298
Solar Radiation (J/m2):	0
Precipitation (m):	0.001
Wind Velocity (m/s):	10
Hour:	0
Month:	6
Day:	1
Predicted Total Load (kW):	413047.875
Predicted Coal Generate...	158961.828125
Predicted Diesel Generat...	300.7701110839844
Predicted Hydro Generat...	11343.5068359375
Predicted Gas Generate...	83100.0546875
Predicted Nuclear Gener...	25496.271484375
Predicted Solar Generat...	-253.8858642578125
Predicted Waste Genera...	136.4930877685547
Predicted Wind Generat...	122656.671875
Predicted Waste-Heat G...	0.000005089043042971753
Predicted Other Sources...	375.6993103027344

Predict & Save

Reset & Clear History



5. Conclusion

We were able to build a well-defined and streamlined process for data extraction pertaining to climate, electricity demand and load generation followed by model building and model deployment to help understand the forecasted electricity demand and the possible distribution of generation by each source type. Random Forests were used for forward modeling and Neural Networks for inverse modeling processes. Both these models were able to achieve satisfactory results with the Random Forest (and XG Boost) getting an accuracy of 0.89 R-2 score and Neural Networks getting an accuracy of 0.65 MSE. Finally, they were deployed as a single tool where a user can enter the forecasted weather conditions from the next day or week, and the tool generates the forecasted demand load, along with generating the contribution of each fuel source in catering to this load.

6. Future work

- More complex models such as double stacked LSTMs can be used replacing Random Forests as they have historically been able to perform well on time-series models
- More intense hyperparameter tuning could further improve the model accuracy
- The deployed tool (which is currently local) can be wrapped into a Flask application and hosted on GitHub, which can then be run on an ec2 instance of an AWS server for public to use freely
- Finally, the model can be trained stronger with larger databases from all other ISOs and RTOs that publish data for their own regions

7. References

1. **Keith D. Humfeld, Dawei Gu, Geoffrey A. Butler, Karl Nelson, Navid Zobeiry, A machine learning framework for real-time inverse modeling and multi-objective process optimization of composites for active manufacturing control, Composites Part B: Engineering, Volume 223, 2021.** https://www.sciencedirect.com/science/article/abs/pii/S135983682100531X
2. **Breiman, L. Random Forests. Machine Learning 45, Pages 5–32 (2001).** https://doi.org/10.1023/A:1010933404324
3. **Dietterich, T. (1998). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization, Machine Learning, 1–22.** https://doi.org/10.1023/A:1007607513941
4. **Kleinberg, E. (2000). On the algorithmic implementation of stochastic discrimination. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(5), 473–490.** https://ieeexplore.ieee.org/document/857004
5. **A. Khalyasmaa et al., "Prediction of Solar Power Generation Based on Random Forest Regressor Model," 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia, 2019, pp. 0780-0785,** doi: 10.1109/SIBIRCON48586.2019.8958063
6. **Villegas-Mier, C.G.; Rodriguez-Resendiz, J.; Álvarez-Alvarado, J.M.; Jiménez-Hernández, H.; Odry, Á. Optimized Random Forest for Solar Radiation Prediction Using Sunshine Hours. Micromachines 2022, 13, 1406.**

https://doi.org/10.3390/mi13091406
7. A. Lahouar, J. Ben Hadj Slama, Day-ahead load forecast using random forest and expert input selection, Energy Conversion and Management, Volume 103,2015, Pages 1040-1051, https://www.sciencedirect.com/science/article/abs/pii/S0196890415006925
8. H. Shayeghi, A. Ghasemi, M. Moradzadeh, M. Nooshyar, Simultaneous day-ahead forecasting of electricity price and load in smart grids, Energy Conversion and Management, Volume 95, 2015, Pages 371-384. https://www.sciencedirect.com/science/article/abs/pii/S0196890415001363