ENERGY COMNSUMPTION PREDICTION MODEL

Using deep learning

Teyar Raid
Malaam Bourhan
Saoudi Lhadi

1 Introduction

Energy consumption is a crucial issue in modern society, as it affects both economic development and environmental sustainability. Accurate predictions of energy consumption can help energy providers and consumers make informed decisions, reduce costs, and reduce environmental impact. In this paper, we investigate the use of statistical techniques and data analysis to predict hourly energy consumption from a dataset of historical energy usage. We explore the usage of neural networks and evaluate their performance in terms of accuracy. Our goal is to develop a robust and efficient prediction model that can be used in real-world applications. We will also discuss the potential use cases and limitations of our model. In summary, the paper aims to provide insight into using statistics and data analysis for the prediction of hourly energy consumption using the data provided.

2 Description of data

The PJM Interconnection is a regional transmission organization (RTO) that coordinates the movement of wholesale electricity across 13 states in the Northeastern and Midwestern United States. The organization maintains and makes available a large dataset of hourly power consumption data, which provides historical information on the energy consumption patterns of various regions within the PJM service area. The data is measured in megawatt-hours (MWh) and sourced from multiple electricity providers, including generators, transmission and distribution companies, providing a comprehensive view of energy consumption in the region. This dataset is divided into multiple .csv files, each one corresponding to a specific region or electricity provider. We chose to work with the pjm_hourly_est.csv file in this study, as it consolidates the usage of all regions, providing a more comprehensive understanding of the whole dataset, thus eliminating the need to merge any files together. The dataset has a single feature, "Datetime" in the format YYYY-MM-DD HH:MM:SS, and 12 columns for the electricity consumption in (MWh) of each provider/region: AEP, COMED, DAYTON, DEOK, DOM, DUQ, EKPC, FE, NI, PJME, PJMW, and PJM_Load. The data is not sorted by date initially, so after sorting it we see that the records start from 1998-04-01 01:00:00 to 2018-08-03 00:00:00, with a total number of instances of 178,262.

Datetime	AEP	COMED	DAYTON	DEOK	DOM	DUQ	EKPC	FE	NI	PJME	PJMW	PJM_Load
1998-04-01 01:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	22259.0
1998-04-01 02:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	21244.0
1998-04-01 03:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	20651.0
1998-04-01 04:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	20421.0
1998-04-01 05:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	20713.0
...	...	...	...	...	...	...	...	...	...	...	...	...
2018-08-02 20:00:00	17673.0	16437.0	2554.0	4052.0	14038.0	1966.0	1815.0	9866.0	NaN	44057.0	6545.0	NaN
2018-08-02 21:00:00	17303.0	15590.0	2481.0	3892.0	13832.0	1944.0	1769.0	9656.0	NaN	43256.0	6496.0	NaN
2018-08-02 22:00:00	17001.0	15086.0	2405.0	3851.0	13312.0	1901.0	1756.0	9532.0	NaN	41552.0	6325.0	NaN
2018-08-02 23:00:00	15964.0	14448.0	2250.0	3575.0	12390.0	1789.0	1619.0	8872.0	NaN	38500.0	5892.0	NaN
2018-08-03 00:00:00	14809.0	13335.0	2042.0	3281.0	11385.0	1656.0	1448.0	8198.0	NaN	35486.0	5489.0	NaN

178262 rows × 12 columns

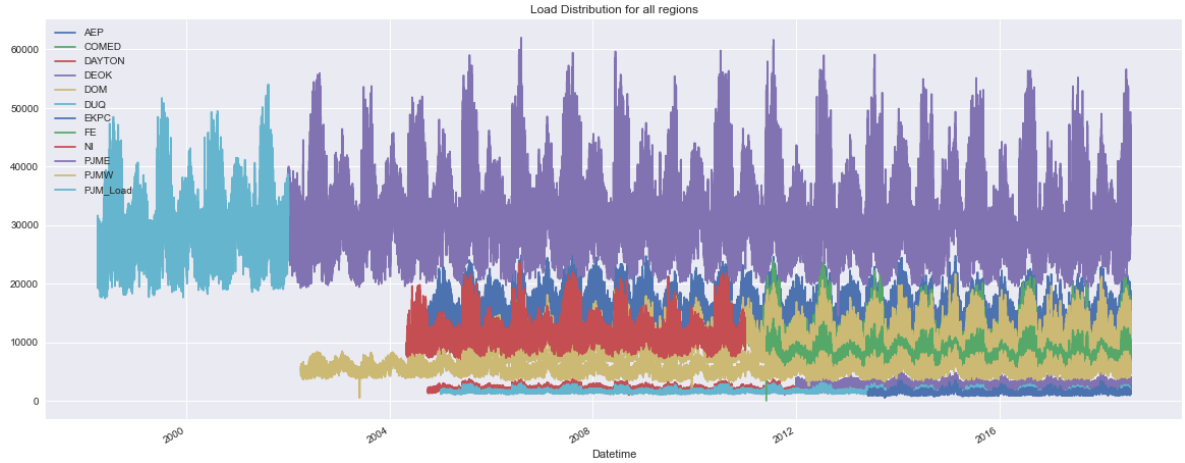
The regions have changed over the years, resulting in missing data for certain dates per region.

3 Statistical analysis

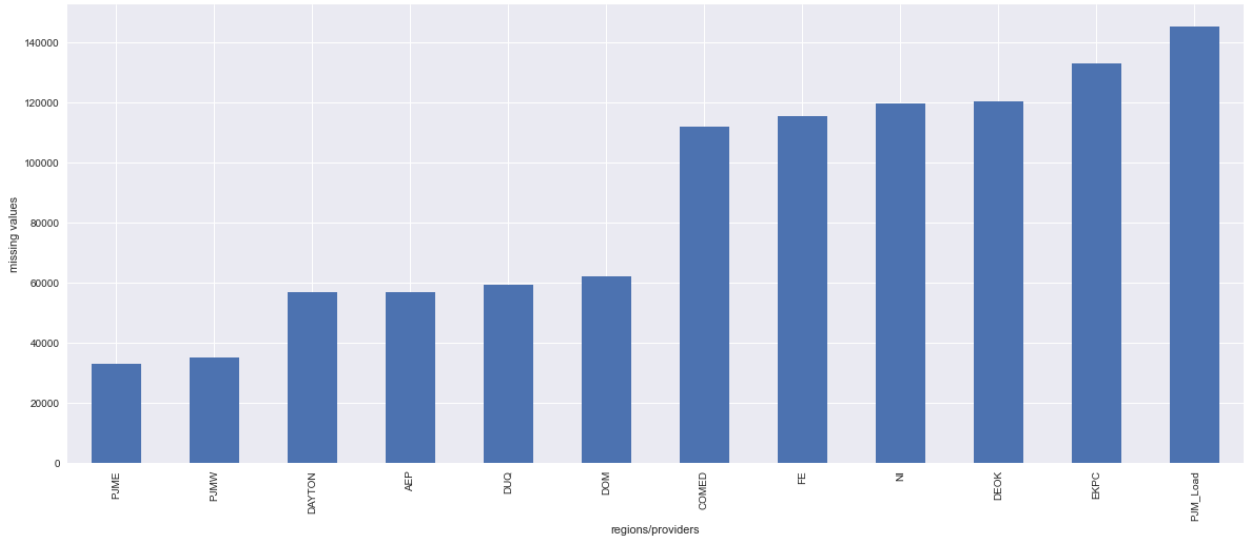
Our data has missing and duplicate values but none of them affects our future model, because missing values here are either before the start of records or after the end of records for a specific region/provider, As for duplicates we can't remove them because in time series type of data, removing them will result in the lose time stamps.

Therefore, after converting our Datetime feature from its string value to datetime64 type we can start visualizing our data right away.

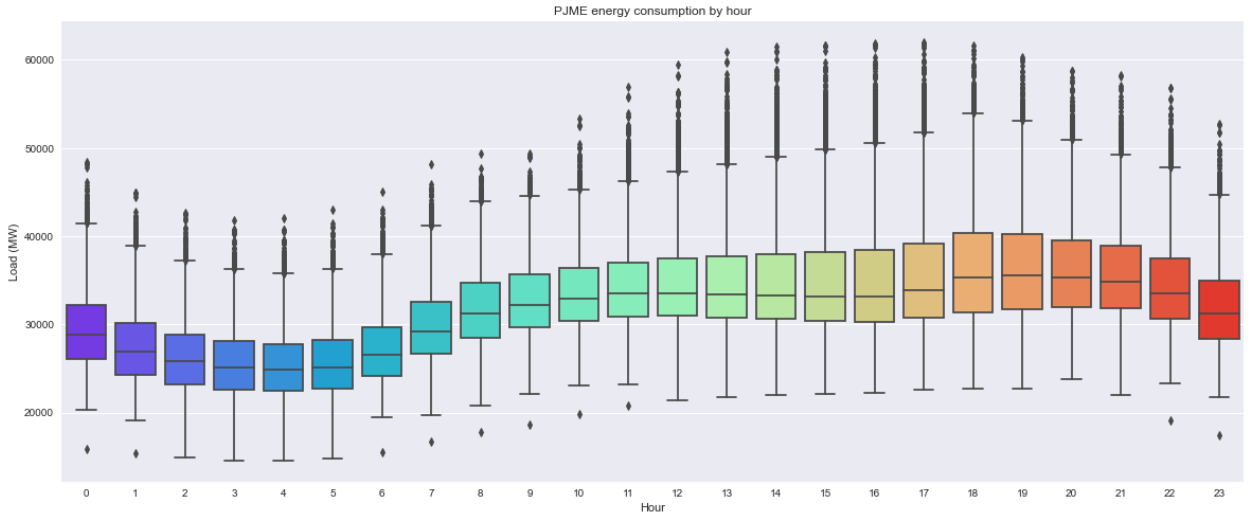
After plotting the whole distribution of load for all regions/providers, we see that each one of them has a different energy consumption pattern. Therefore, it is logical to make predictions for each region separately, as otherwise, our model will have low accuracy.

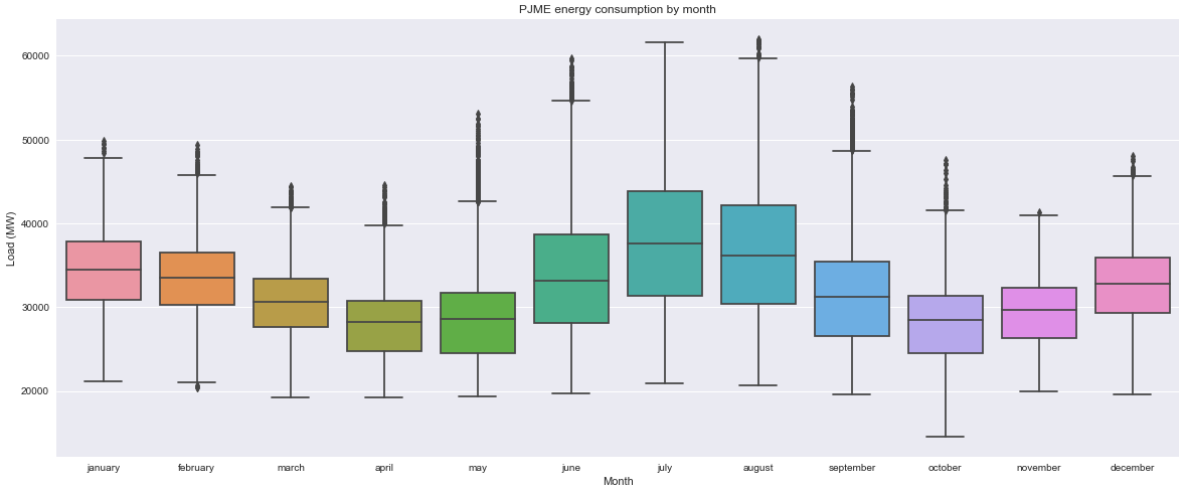


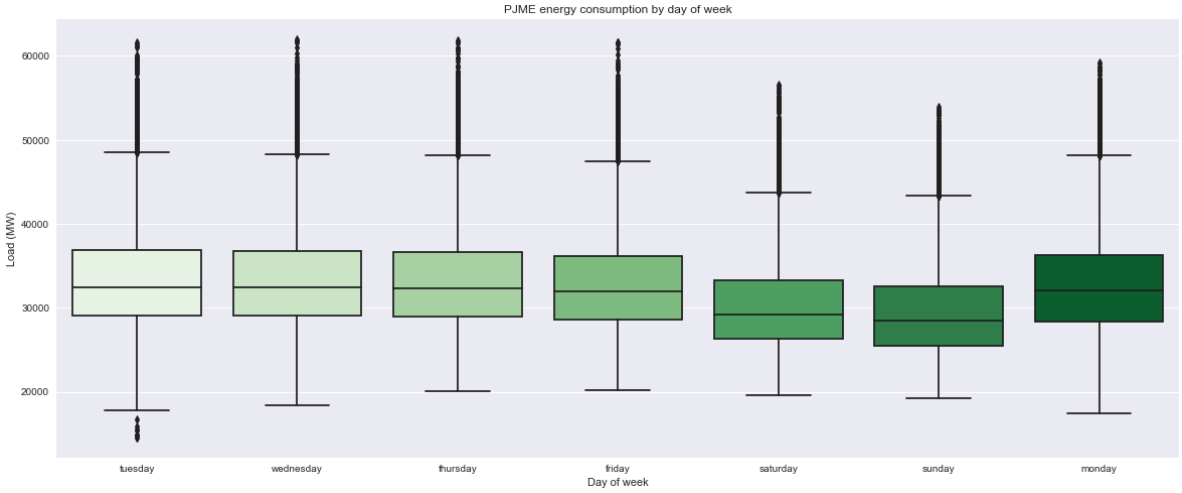
We choose the region/provider with the most data "least missing values" to achieve the best accuracy possible. Plotting the count of missing values of each region/provider shows that PJME has the least missing values at 32,896.

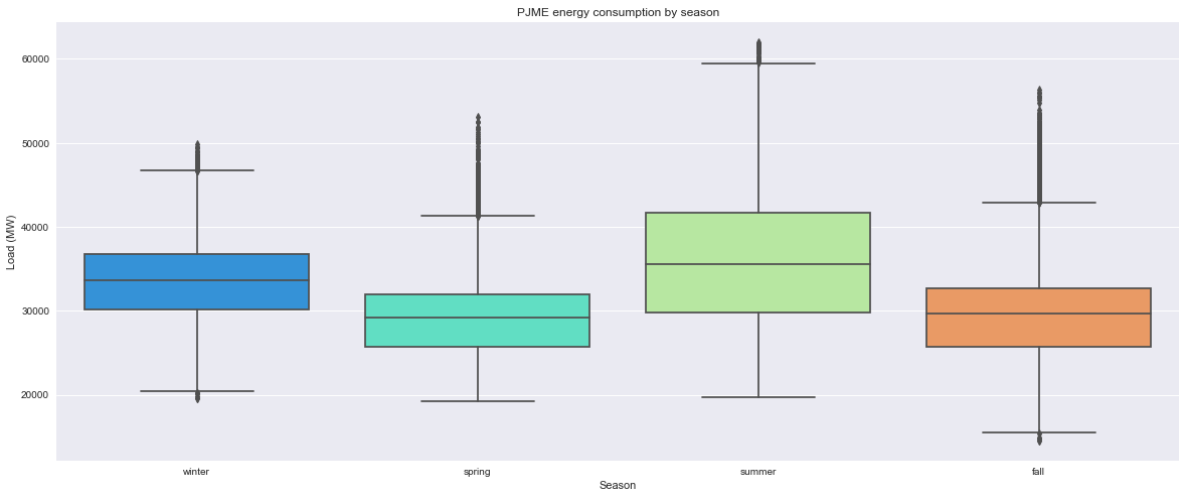


As previously mentioned, we can drop the missing values because there is no need to replace them. To better visualize the data in hand, we split our Datetime feature into hour, day_of_week, month, year, season to be used for hourly, weekly, monthly, yearly and seasonal visualization/analysis. This allows us to use boxplots to visualize the new features and gain insights on the data.




PJME energy consumption by month
Load (MW)
60000
50000
40000
30000
20000
january
february
march
april
may
june
july
august
september
october
november
december
Month


PJME energy consumption by day of week
Load (MW)
60000
50000
40000
30000
20000
tuesday
wednesday
thursday
friday
saturday
sunday
monday
Day of week


PJME energy consumption by season
Load (MW)
60000
50000
40000
30000
20000
winter
spring
summer
fall
Season

we can see that the electricity consumption from 0 to 8 am is lower than the consumption through the rest of the day. ,We also see that July and August have the highest electricity consumption while October has the least consumption,and that weekends "Saturdays and Sundays" have a lower consumption compared to weekdays. As for the seasonal view, we see that summer has remarkably higher electricity consumption than other seasons Now that we saw the data in hand clearly, we can start preparing it for the model, we first split the data into 70% for training and 30% for testing. Next, we normalize the data as it's a crucial step in machine learning to make our values in a common scale (between 0 and 1) and to stabilize the gradient descent steps. with that out of the way we can then start talking about the model

4 The Model

For time series forecasting, there are several options to choose from, such as Extreme Gradient Boosting (XGBoost), ARIMA, and Long Short-term Memory Networks (LSTM). We initially tried XGBoost, but it did not provide the accuracy we desired (72%), so we moved on to LSTM. LSTM is a subclass of recurrent neural networks (RNN) that are particularly well-suited for sequential data, which means data samples that change over time. In order to use our data with an LSTM model, we first need to convert our Dataframe to a dataset matrix,This requires structuring our inputs in the format [samples, time steps, features], time steps are something related to RNN, they are basically the number of previous time steps to use as input variables to predict the next time period. To achieve this, we grouped each 5 time steps together for the input when creating the dataset matrix. We then reshaped the input to the desired structure before building the neural network.

1. first we initialize our sequential model because our model will be a sequence of neural network layers

2. we add our first layer of type LSTM, and since its the first layer we need to specify the right input shape, we also need to specify the number of units "nodes/cells"

3. we add a dropout layer to help us prevent overfitting

4. lastly we add a dense layer, the layer that will actually give us the predicted values, we then compile our network and specify the loss function, and optimizer. we can then fit our model and give it the validation_data, epochs number "number of complete passes through the training dataset" and the batch_size "number of samples processed before the model is updated".

5 Results

After training the model and evaluating its performance, we obtained an accuracy of 84.6%. We then sought to further improve the accuracy of our model by exploring different hyperparameter settings. One approach to achieve this is to manually try different configurations, while another option is to use a model tuner, which is what we chose to do. We used the Keras Tuner library to perform a search for the best hyperparameters for our LSTM model. We defined a range of values for the number of units in the LSTM layer and a range of learning rates to test. Once the search was completed, the Keras Tuner provided us with the model that had the optimal hyperparameters. To evaluate the performance of this new model, we ran an R2 score test on the test data and obtained an accuracy of 98%. We then saved the model in .h5 format. It is worth noting that an accuracy of 98% is quite high, and it may indicate that the model is overfitting the data. To confirm this, we should also evaluate the model's performance on unseen data and check for other signs of overfitting such as a large difference between training and validation accuracy.

6 Conclusion

In this research paper, we presented an investigation of using statistical techniques and data analysis to predict hourly energy consumption from a dataset of historical energy usage provided by the PJM Interconnection. We focused on the usage of Long Short-term Memory Networks (LSTM), a type of recurrent neural network, and evaluated their performance in terms of accuracy. One of the main limitations of this study is that the model is only trained and tested on data from one specific region and may not generalize well to other regions or countries. Additionally, the model may be sensitive to changes in the energy consumption patterns or the availability of data. There are several directions for further research that could be pursued. For example, further investigations could be conducted on how to improve the model's generalization capabilities, or if this model we created is overfitting or not. Overall, this research highlighted the importance of understanding the underlying patterns and trends in energy consumption data, and the value of using statistical techniques and data analysis to predict energy consumption. It also showed the potential of neural networks in this application, and the importance of careful model selection and hyperparameter tuning to achieve optimal performance.