Forecasting Energy Consumption using Machine Learning

Team:

Usha Sravani Ganta, Sravani Namburu, Subhash Aleti, Sai Thanishvi Daruru, Rahul Chowdary Narramneni

Introduction:

In today's fast-paced industrial world, numerous projects are underway to meet the needs of the people, with energy consumption playing an increasingly important role that must be appropriately utilized in the future. As a result, optimizing energy consumption has become a top priority for the government. The goal of the project is to perform high level exploratory data analysis and build different time series models which will help in forecasting the energy consumption in future.

Summary:

Machine learning is very good at analyzing time series data. Firstly, time series data is all about a set of measurements that happen in a constant or a periodic time frame. Time is the independent variable here. It is a collection of sequential data points and analyzing them will help us in understanding the underlying concept of the data points and the relations and help in optimizing the solutions for a better cause. In this project analyzing the power consumption data set will help in optimizing the power distribution among the states and helps in budling economic status of the country.

The major goals of the project are:

1. To analyze the period where most and least amount of the energy is consumed.
2. To build a prediction model to predict the energy consumed in the state
3. To compare the result metrices of the RNN, LSTM, XG-BOOST algorithms and state the state the best algorithm
4. Also, to provide the future scope and improvisations based on our current study.

We tried to expand the dataset by columns and visualized the power consumption through different types of timeseries components such as year, date, time, month, quarter, isweekday etc.

Dataset Description:

The data set we used is by PJW, it's a regional transmission and part of the eastern international grid in USA, providing the services to different states. The data set contains two columns one is date time column and other is power consumption in Megawatts. The data set contains data from 2014-2018.

Data Preprocessing:

The data set is from Kaggle and downloaded the dataset to our devices and then loaded dataset into the dataframe. There are no null values present in the dataset. The data is continuous, and there are only two columns to perform exploratory data analysis. So, the data is expanded by columns, we created Year, Month, Day, Time, Quarter, Season, Isweekday (which tells us the day is week or weekend, i.e., true for weekday and false for weekend).

```
Datetime    datetime64[ns]
AEP_MW             float64
dtype: object
```

	Datetime	AEP_MW
0	2004-12-31 01:00:00	13478.0
1	2004-12-31 02:00:00	12865.0
2	2004-12-31 03:00:00	12577.0
3	2004-12-31 04:00:00	12517.0
4	2004-12-31 05:00:00	12670.0

Figure1. Data before Preprocessing

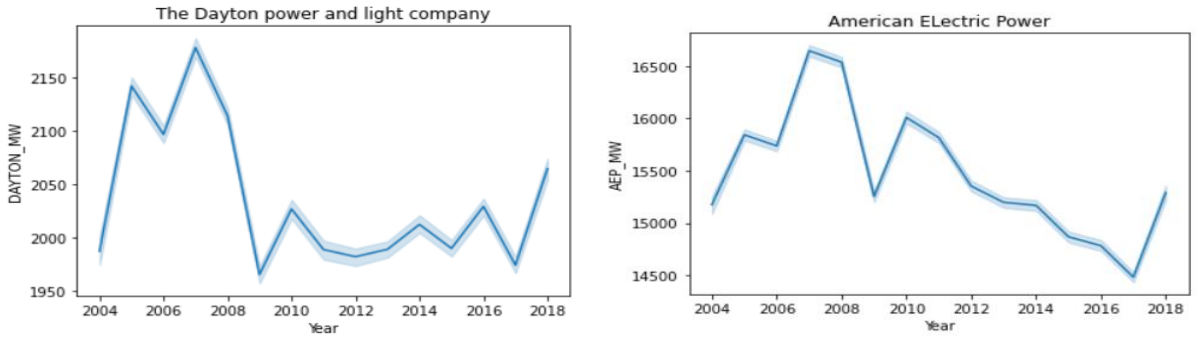
	Datetime	AEP_MW	Year	Month	Day	Time	Quarter	Season	Isweekday
0	2004-12-31 01:00:00	13478.0	2004	12	31	01:00:00	4	Fall	true
1	2004-12-31 02:00:00	12865.0	2004	12	31	02:00:00	4	Fall	true
2	2004-12-31 03:00:00	12577.0	2004	12	31	03:00:00	4	Fall	true
3	2004-12-31 04:00:00	12517.0	2004	12	31	04:00:00	4	Fall	true
4	2004-12-31 05:00:00	12670.0	2004	12	31	05:00:00	4	Fall	true

Figure2. Data after Preprocessing

Exploratory Data Analysis:

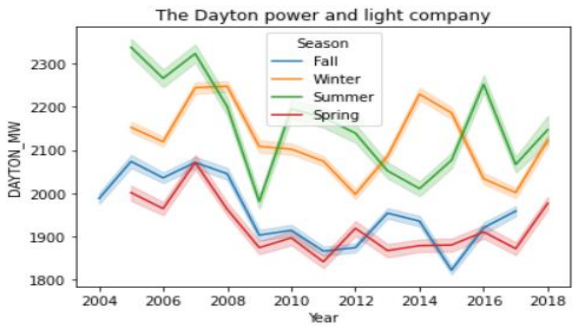
In EDA we just want to observe how the power consumption trends based on the seasons, year, and during the weekdays.

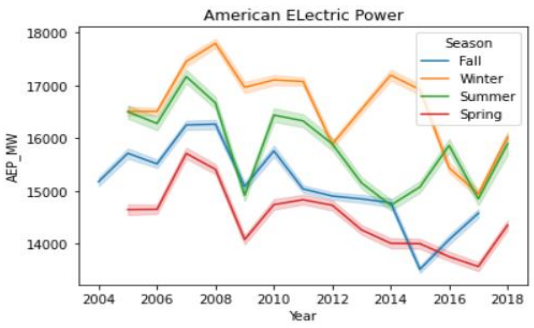
1. EDA of Power Consumption based on Year



The above plots are between the year and energy consumption. In all the graphs we observed that in 2004 energy is very less consumed and from 2006 -2008 there is quite a rise in the consumption, and it got decreased in year 2009 and again in 2017. Based on our research the downfall and raise are due to the change in the policies in the government, because the US elections we in the year 2004,2008 and 2016. The link in the reference will help in finding the information.

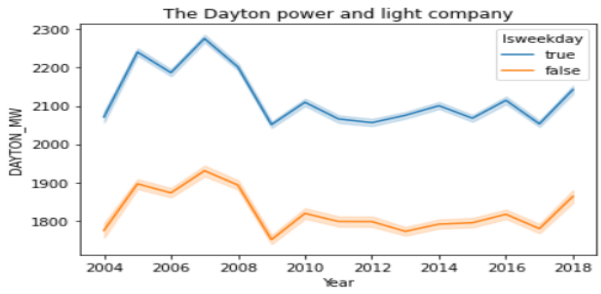
2. EDA of Power Consumption based on Seasons

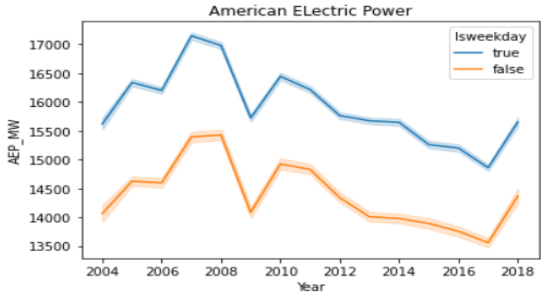




Based on the plots between the seasons and the energy consumptions, the Summers and Winters have the higher energy consumption than spring and fall because, of the high usage of heaters and air conditions. In Fall and Spring, the consumption is not so high because the usage of heaters and AC's decreases.

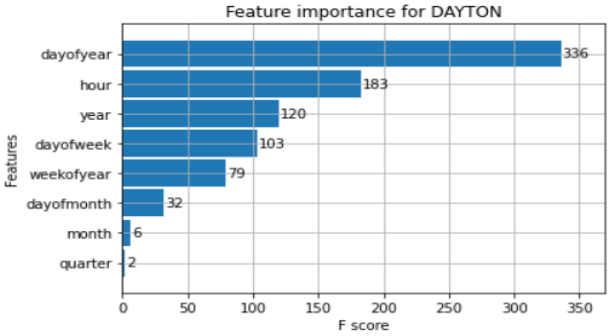
3. EDA of Power Consumption based on Isweekday

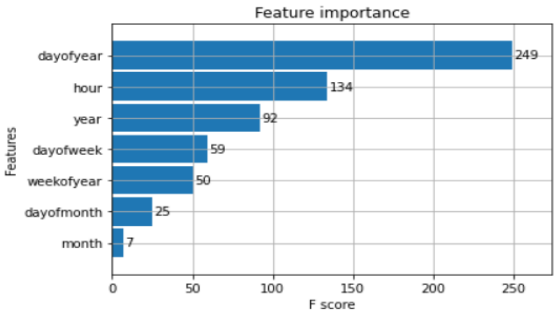




Based on the plots for the isweekday, we can see the power consumption is high during the weekdays, because most of the industries and offices and schools and colleges are running during weekdays and people will not spend most of the time at home.

4. Feature Importance





Based on the plots the feature importance is calculated where we got the day hour and year as the top 3, which we can tell power consumption depends on day activates. The plots are behaving parallelly to our assumptions and our known facts which tells the data is correct and we can start modelling our data.

Modelling:

We have taken the data time column and power consumption for modeling. Before modelling we did few steps. That is Resampling the data based on data time index and normalizing the data using Minmax scalar and divided the data into test and train.

The need of resampling the data:

On our study on time series data, resampling is a methodology of economically using the data y quantifying the uncertainty and to improve accuracy.

The need of Normalization:

As it's continuous data, we need to bring them on a common scale, without making differences in the range of values.

1. RNN:

It's a deep neural network, used to solve and perform analysis on sequential data. So, Time series data can be processed and analyzed using RNN. It will learn from the previous iteration during the training, because it uses their internal state memory. At each layer the error minimizes. The activation function we used is tanh, because tanh supports multilayer network and supports backpropagation, which is exact the main concept in RNN. Also used Adam optimizer, which will help in fastening computational time of the model fitting.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn (SimpleRNN)      (None, 20, 40)            1680

 dropout (Dropout)           (None, 20, 40)            0

 simple_rnn_1 (SimpleRNN)    (None, 20, 40)            3240

 dropout_1 (Dropout)         (None, 20, 40)            0

 simple_rnn_2 (SimpleRNN)    (None, 40)                3240

 dropout_2 (Dropout)         (None, 40)                0

 dense (Dense)               (None, 1)                 41

=================================================================
Total params: 8,201
Trainable params: 8,201
Non-trainable params: 0
```

Figure 3. Fitting the RNN model

```
Epoch 1/5
77/77 [==============================] - 11s 74ms/step - loss: 0.1034 - val_loss: 0.0024
Epoch 2/5
77/77 [==============================] - 5s 68ms/step - loss: 0.0163 - val_loss: 0.0016
Epoch 3/5
77/77 [==============================] - 5s 69ms/step - loss: 0.0091 - val_loss: 0.0014
Epoch 4/5
77/77 [==============================] - 5s 69ms/step - loss: 0.0063 - val_loss: 0.0011
Epoch 5/5
77/77 [==============================] - 5s 68ms/step - loss: 0.0049 - val_loss: 9.4142e-04
```

Figure 4. Training the RNN model Epochs =5, Batch size= 1000

2. LSTM:

It is special kind of recurrent neural network that is capable of learning long term dependencies in data. This is achieved because the recurring module of the model has a combination of four layers interacting with each other. We used the same activation function and optimizer and performed model fitting.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 20, 40)            6720

 dropout_3 (Dropout)         (None, 20, 40)            0

 lstm_1 (LSTM)               (None, 20, 40)            12960

 dropout_4 (Dropout)         (None, 20, 40)            0

 lstm_2 (LSTM)               (None, 40)                12960

 dropout_5 (Dropout)         (None, 40)                0

 dense_1 (Dense)             (None, 1)                 41

=================================================================
Total params: 32,681
Trainable params: 32,681
Non-trainable params: 0
```

Figure 5. Fitting the LSTM model

```
Epoch 1/5
77/77 [==============================] - 11s 57ms/step - loss: 0.0298 - val_loss: 0.0176
Epoch 2/5
77/77 [==============================] - 3s 35ms/step - loss: 0.0190 - val_loss: 0.0143
Epoch 3/5
77/77 [==============================] - 3s 35ms/step - loss: 0.0114 - val_loss: 0.0064
Epoch 4/5
77/77 [==============================] - 3s 35ms/step - loss: 0.0073 - val_loss: 0.0040
Epoch 5/5
77/77 [==============================] - 3s 35ms/step - loss: 0.0051 - val_loss: 0.0024
```

Figure 6. Training the LSTM model Epochs =5, Batch size= 1000

3. XG_BOOST:

Extreme Gradient boosting is based on Gradient Boosted decision trees. This algorithm tries to accurately predict a target variable by combining the estimates of a set of simpler, weaker models. It contains homogeneous models and trains the weaker models in a sequential approach. We used hyperparameters, objective as “reg:squarederror” and n_estimators as 1000.

Results:

We have run RNN, LSTM and XG_BOOST based on the data we have, the following are the data analysis and result comparisons.

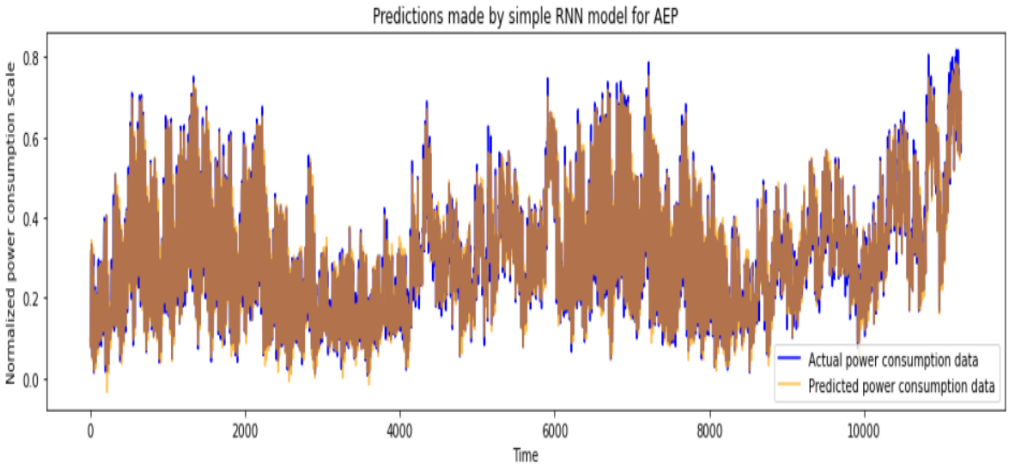


Figure 6. Plot showing the analysis of predicted and actual values on the AEP test data for RNN.

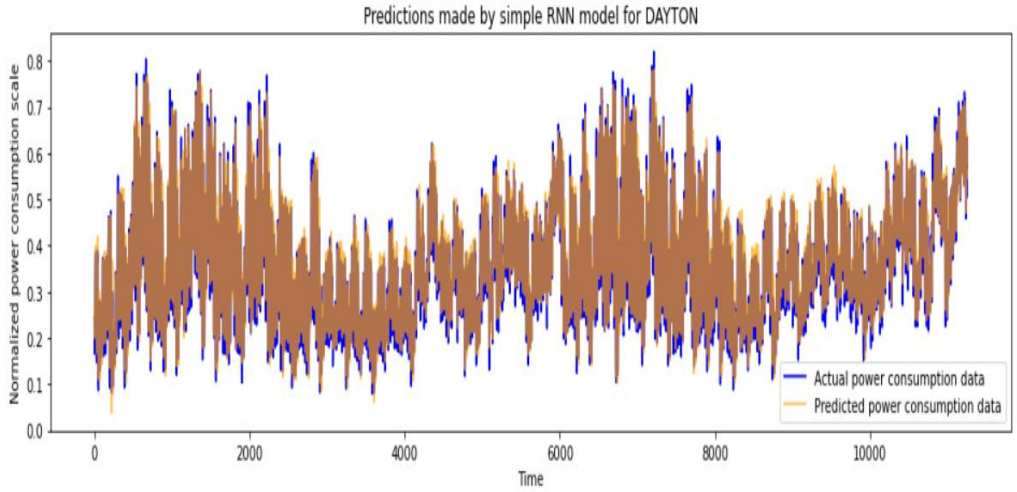


Figure 7. Plot showing the analysis of predicted and actual values on the DAYTON test data for RNN.

Based on the observation we can state that the predicted and actual values are matched, and the error was minimum. For now, we can sate RNN is running good with our data set.

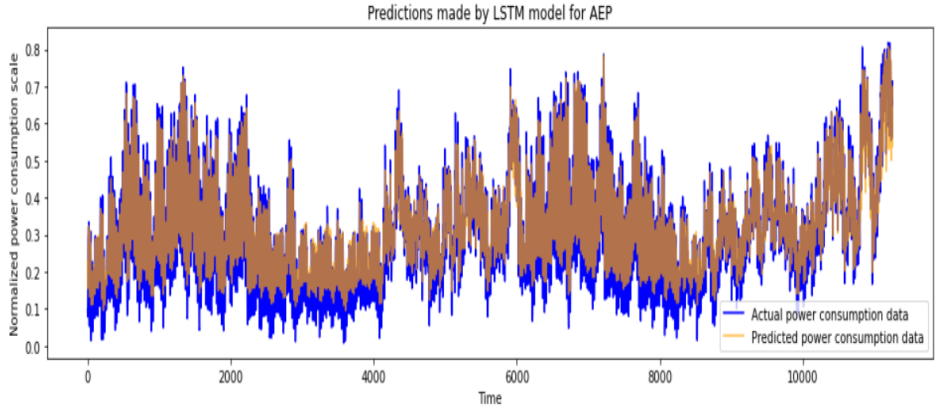


Figure 8. Plot showing the analysis of predicted and actual values on the AEP test data for LSTM

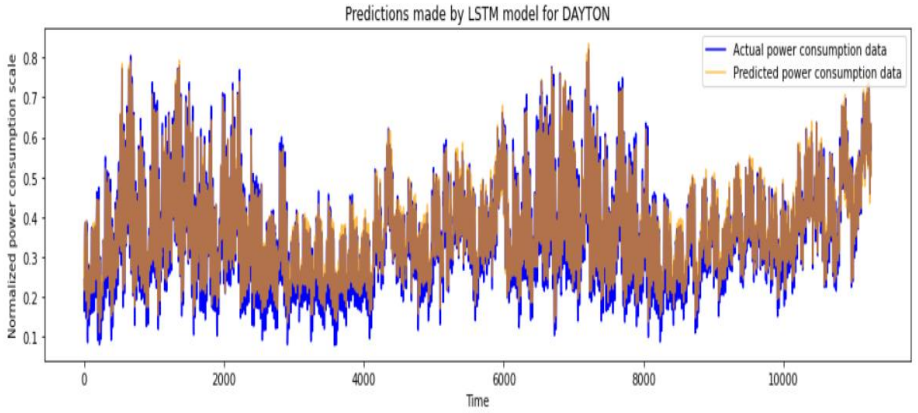


Figure 9. Plot showing the analysis of predicted and actual values on the DAYTON test data for LSTM

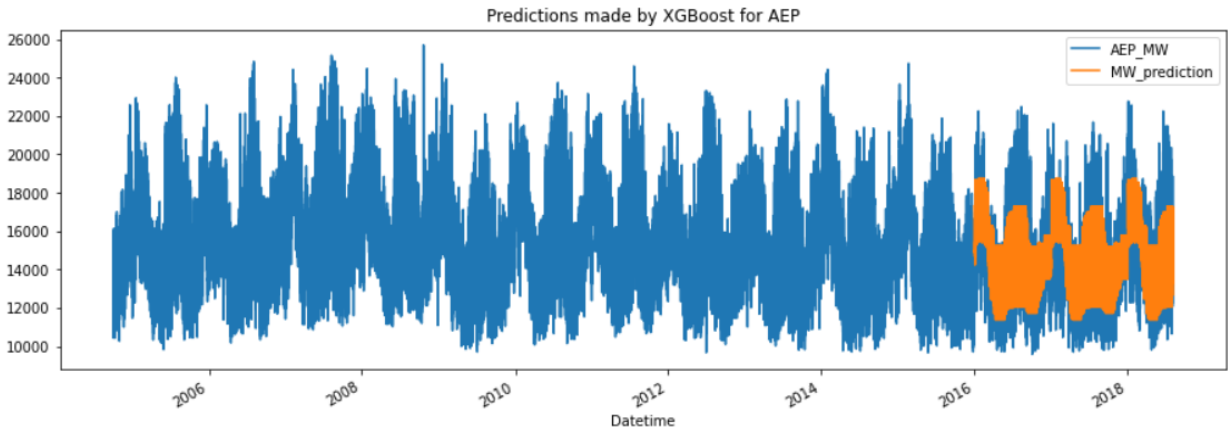


Figure 10. Plot showing the analysis of predicted and actual values on the AEP test data for XG_BOOST

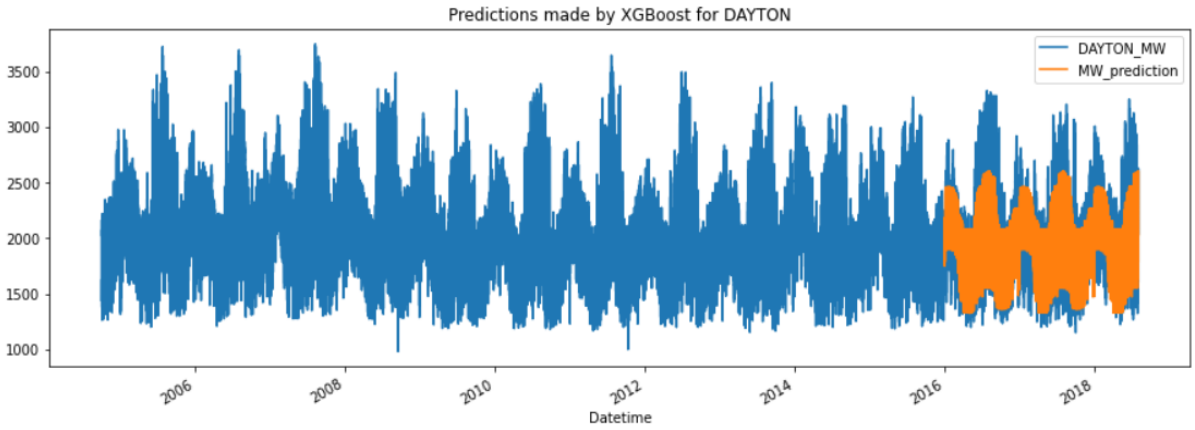


Figure 11. Plot showing the analysis of predicted and actual values on the DAYTON test data for XG_BOOST

Based on the result analysis and plots, the LSTM is better than XG_BOOST. So let's look into the analysis of LSTM AND RNN.

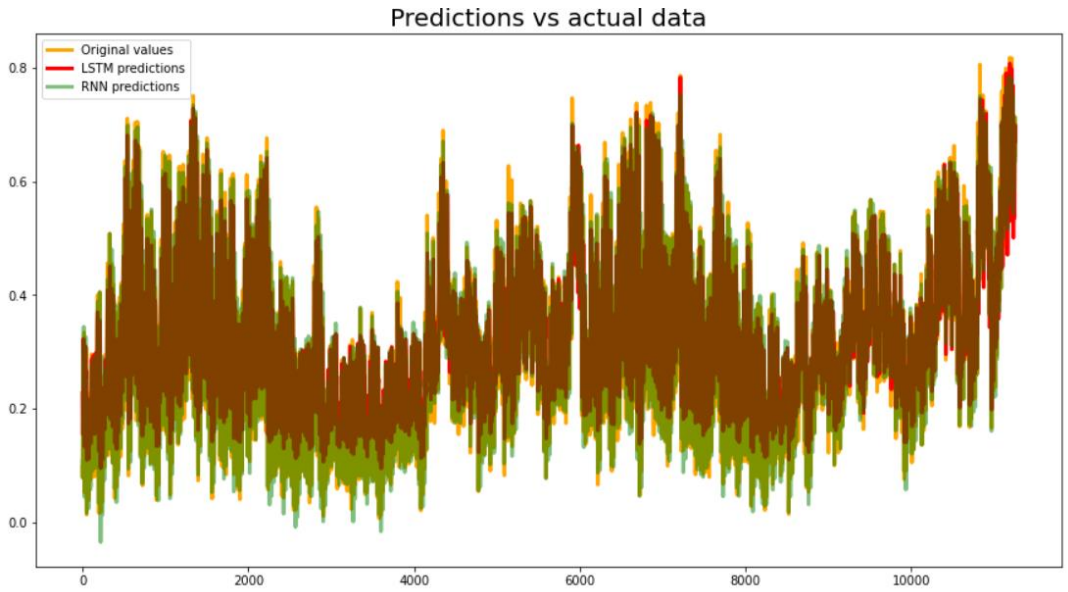


Figure 12. Plot showing the analysis of predicted and actual values on the AEP test data for RNN & LSTM

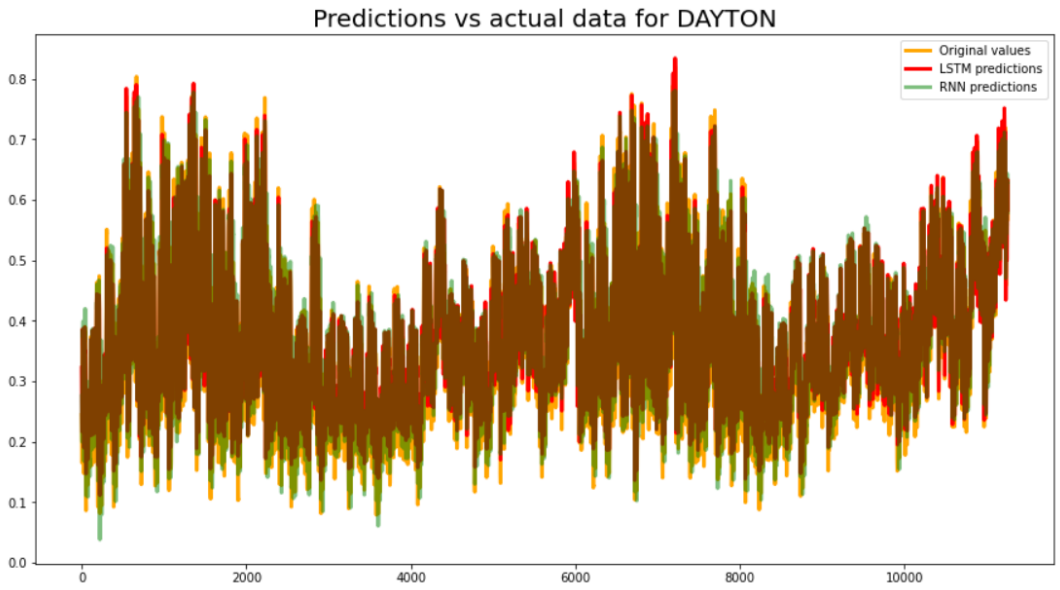


Figure 13. Plot showing the analysis of predicted and actual values on the DAYTON test data for RNN & LSTM

Based on the observations, we can state that RNN predicted values match more to the actual values than the LSTM. It might be because the there are fewer parameters to train. Therefore base on the plot observations, we can state that RNN works better than LSTM.

Result comparisons based on the Metrics:

Metrics	RNN	LSTM	XG_BOOST
R^2	0.963	0.904	0.52
MSE	0.0008	0.002	294
RMSE	0.029	0.047	171
MAE	0.020	0.035	137

Figure 14. Result comparison of the metrics for RNN, LSTM, XG_BOOST for AEP grid

Metrics	RNN	LSTM	XG_BOOST
R^2	0.946	0.893	0.62
MSE	0.0009	0.001	544
RMSE	0.031	0.043	233
MAE	0.023	0.034	179

Figure 15. Result comparison of the metrics for RNN, LSTM, XG_BOOST for DAYTON grid

Based on the result metrices we can say that RNN algorithm performs best when compared with both LSTM and XG_BOOST algorithms by giving minimum mean squared error, root mean squared error, mean absolute error and the maximum R-square values.

Discussion:

Project Impact

Based on the observation on the models that are fitted we state that RNN model works better, As we have data till 2018, we can try to synthesize the data to get feature consumption values and try to optimize the consumption of the energy distribution among the grids. This will highly impact the economic status of the country as proposed and help in protecting the non-renewable resources.

Future Work:

1. Modeling the energy consumption and predicting the future demand of energy to be distributed in the power distribution sectors is a challenging task.
2. Moreover, the noise disturbance and uncertain nature of weather conditions also play a backbreaking step to predict the rate of energy consumption. In this context, we implement a powerful RNN based energy consumption prediction model which is validated by simulating with the hourly based collected data set.
3. Additionally, the performance of the proposed model was evaluated through the performance metrics. We aim to incorporate various dimensional dataset such as the occupant's behavior and the climate condition to verify and test the predicted variable of our proposed network as the further research work.

References:

https://www.britannica.com/topic/United-States-Presidential-Election-Results-1788863

https://doi.org/10.1016/j.petrol.2019.106682

https://medium.com/analytics-vidhya/combining-time-series-analysis-with-artificial-intelligence-the-future-of-forecasting -5196f57db913

https://www.advancinganalytics.co.uk/blog/2021/06/22/10-incredibly-useful-time-series-forecasting-algorithms

https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption