Forecasting Comparison: Neural Network vs ARIMA

Sampad Kumar Kar(MCS202215)
Shourjya Basu(MCS202207)

20th April 2023

Objective

- We use a power consumption dataset and use it to predict the same using weather conditions
- We perform EDA and necessary data preprocessing to prepare the data for model fitting.
- We fit 'Time Series models(ARIMA)' as well as 'Neural Network models(LSTM and Transformers)' on the dataset
- Our goal is to develop a model that takes in weather conditions at a particular time of the day and returns the power consumption at that time.

The Data

Intro to Dataset

- We are using the Tetuan City power consumption dataset recorder for the year 2017.
- The dataset records weather conditions(temperature, humidity, wind speed, and drift speed) and power consumption in 3 zones every 10 minutes for an entire year.
- Tetuan City is a city on the Mediterranean coast of Morocco. Its weather is affected by both the Mediterranean climate and the Sahara climate and which in turn affects the power consumption patterns in the city.
- We record the monthly temperatures of the city.

Continued...

	Jan	Feb	Mar	Apr	May	Jun
Max temp.(in °c)	19.48	19.32	26.73	26.87	35.46	31.53
Min temp.(in °c)	3.247	5.352	6.804	10.56	13.4	15.15
	Jul	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
Max temp.(in °c)	40.01	35.94	32.97	29.92	23.46	21.15
Min temp.(in °c)	19.46	18.89	13.99	12	5.109	4.54

- As we see, the city has summer in the middle of the year and winter at the starting and end. Due to the effect of the Sahara climate, it faces extremes of temperatures over the course of the data. These affect the power consumption in the city.

Dataset Summary

This data contains the power consumption of metrics (in MW: megawatts) for every 10 minutes starting from 2017-01-01 00:00 till 2017-12-30 23:50. It has a total of 52416 entries.

DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption
2017-01-01 00:00:00	6.559	73.8	0.083	0.051	0.119	34055.69620	16128.87538	20240.96386
2017-01-01 00:10:00	6.414	74.5	0.083	0.070	0.085	29814.68354	19375.07599	20131.08434
2017-01-01 00:20:00	6.313	74.5	0.080	0.062	0.100	29128.10127	19006.68693	19668.43373
2017-01-01 00:30:00	6.121	75.0	0.083	0.091	0.096	28228.86076	18361.09422	18899.27711
2017-01-01 00:40:00	5.921	75.7	0.081	0.048	0.085	27335.69620	17872.34043	18442.40964
...	...	...	...	...	...	...	...	...
2017-12-30 22:20:00	7.650	70.1	0.081	0.062	0.122	34323.95437	28676.28107	15684.99400
2017-12-30 22:30:00	7.480	71.0	0.085	0.062	0.104	33776.42586	28230.74563	15546.69868
2017-12-30 22:40:00	7.390	71.2	0.079	0.066	0.100	33387.07224	27814.66708	15396.87875
2017-12-30 22:50:00	7.340	71.0	0.084	0.037	0.119	32815.20913	27564.28352	15172.14886
2017-12-30 23:00:00	7.070	72.5	0.080	0.059	0.093	32158.17490	27273.39675	14987.75510

Data Smoothing

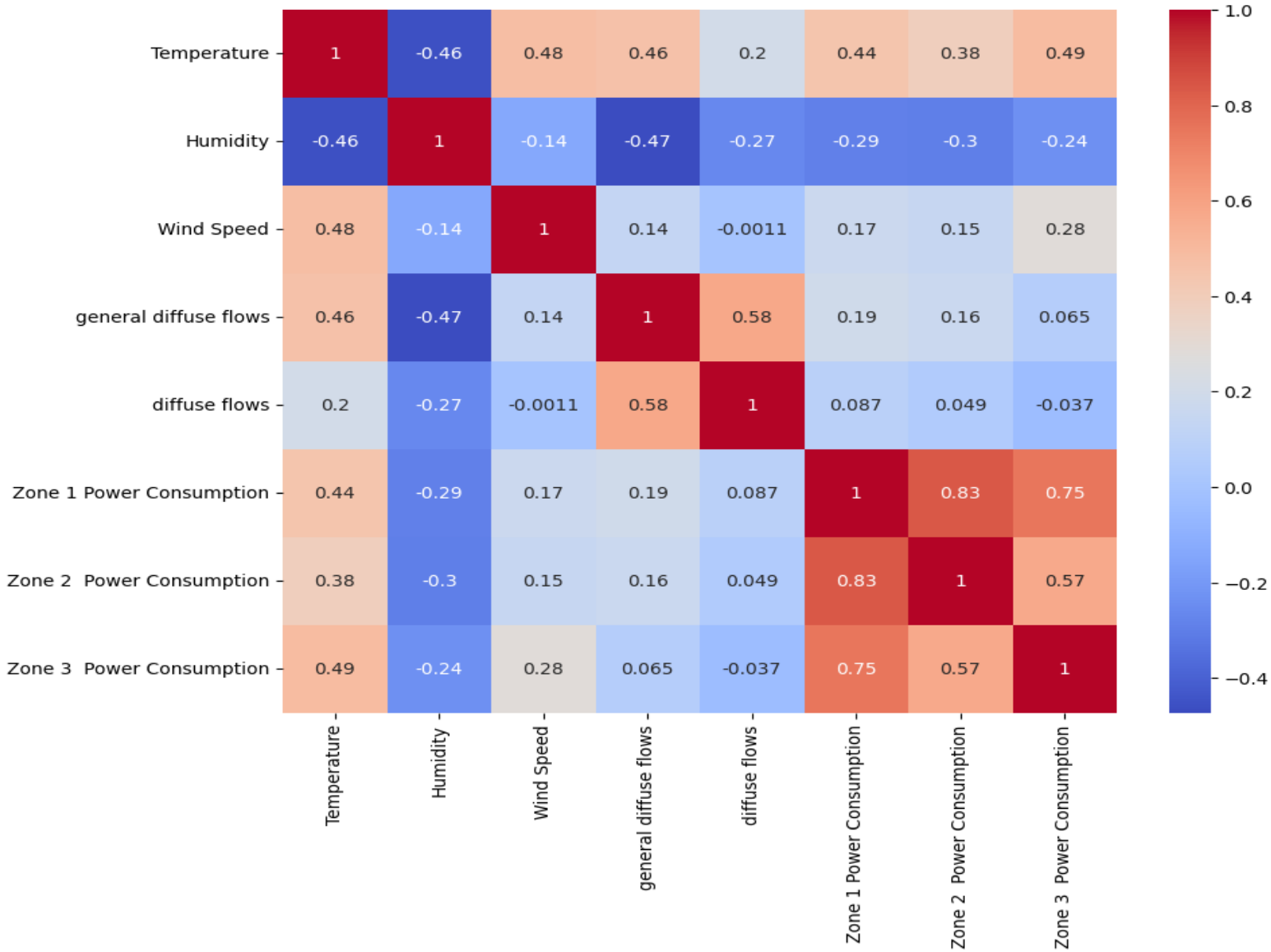
Since processing the data every 10 minutes is too much for our model, we will aggregate the data to 1 hour, by taking the mean of the power consumption of every 6 metrics. We do the same for other features as well.

DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption
2017-01-01 00:00:00	6.196833	75.066667	0.081833	0.063500	0.098833	29197.974683	18026.747720	19252.048193
2017-01-01 01:00:00	5.548833	77.583333	0.082000	0.056833	0.112500	24657.215190	16078.419453	17042.891567
2017-01-01 02:00:00	5.054333	78.933333	0.082333	0.063000	0.129167	22083.037973	14330.699088	15676.144578
2017-01-01 03:00:00	5.004333	77.083333	0.082833	0.059833	0.141000	20811.139240	13219.452887	14883.855422
2017-01-01 04:00:00	5.097667	74.050000	0.082333	0.058000	0.122833	20475.949367	12921.580547	14317.108433
...	...	...	...	...	...	...	...	...
2017-12-30 14:00:00	14.565000	38.140000	0.077833	456.550000	38.210000	29815.969578	26180.423440	10597.839137
2017-12-30 15:00:00	14.461667	41.245000	0.077500	362.750000	46.116667	28771.609633	25654.495243	10424.009605
2017-12-30 16:00:00	14.255000	41.560000	0.077167	226.316667	201.666667	28546.514577	25164.160785	10592.076832
2017-12-30 17:00:00	13.775000	44.448333	0.077833	81.493333	103.070000	33979.214195	29879.717705	15039.615845
2017-12-30 18:00:00	10.771667	58.323333	0.076500	3.185833	3.251500	37929.531053	32764.651735	16786.554622

Exploratory Data Analysis

- The shape of the data is: (8736, 8)
- The number of missing values is: 0
- The number of duplicated values is: 0
- We plot the correlation matrix(P.T.O.) and infer the following:
  - Temperature, Wind Speed, and general diffuse flows are positively correlated to all the power consumption zones. Higher use of air conditioners could be a possible explanation for this.
  - Humidity, on the other hand, shows a slight negative correlation with the power consumption of all three areas.
  - Power consumption in each of the three zones, show a high correlation between each other. This suggests that power consumption in these zones, tend to increase or decrease together.

Correlation Matrix

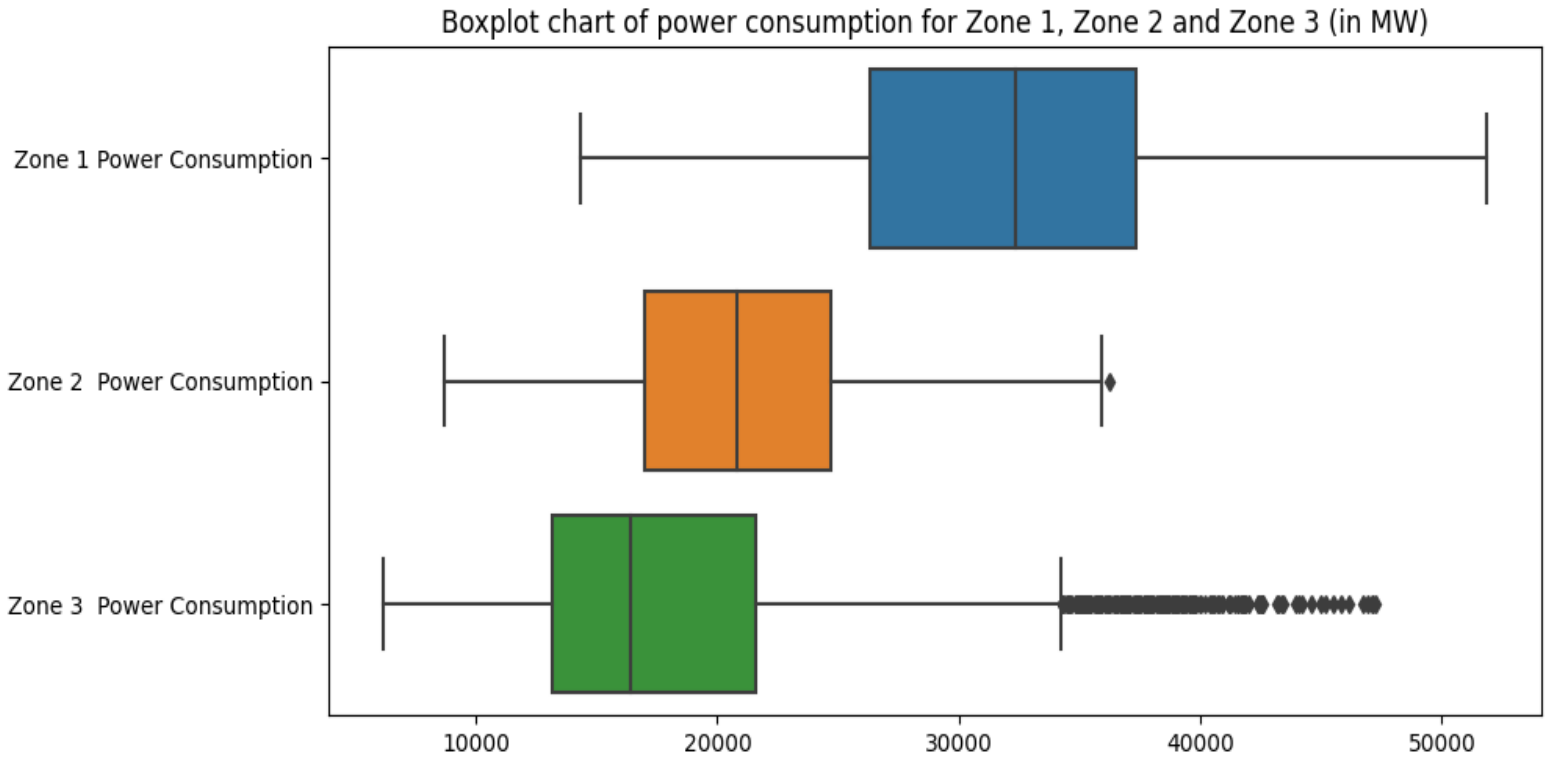


Feature Creation

We add some features to our data for better performance of the models.

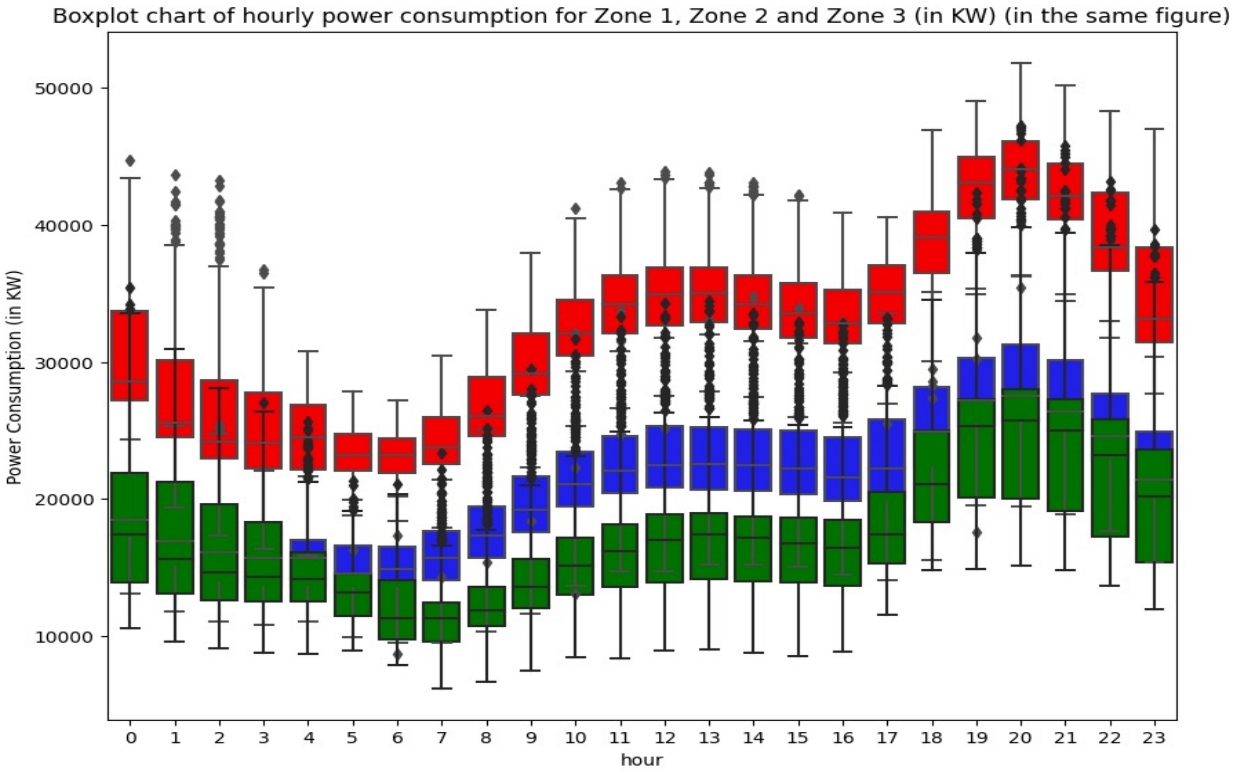
DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption	hour	dayofweek	quarter	month	year	dayofyear	dayofmonth	weekofyear
2017-01-01 00:00:00	6.196833	75.066667	0.081833	0.063500	0.098833	29197.974683	18026.747720	19252.048193	0	6	1	1	2017	1	1	52
2017-01-01 01:00:00	5.548833	77.583333	0.082000	0.056833	0.112500	24657.215190	16078.419453	17042.891567	1	6	1	1	2017	1	1	52
2017-01-01 02:00:00	5.054333	78.933333	0.082333	0.063000	0.129167	22083.037973	14330.699088	15676.144578	2	6	1	1	2017	1	1	52
2017-01-01 03:00:00	5.004333	77.083333	0.082833	0.059833	0.141000	20811.139240	13219.452887	14883.855422	3	6	1	1	2017	1	1	52
2017-01-01 04:00:00	5.097667	74.050000	0.082333	0.058000	0.122833	20475.949367	12921.580547	14317.108433	4	6	1	1	2017	1	1	52
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2017-12-30 14:00:00	14.565000	38.140000	0.077833	456.550000	38.210000	29815.969578	26180.423440	10597.839137	14	5	4	12	2017	364	30	52
2017-12-30 15:00:00	14.461667	41.245000	0.077500	362.750000	46.116667	28771.609633	25654.495243	10424.009605	15	5	4	12	2017	364	30	52
2017-12-30 16:00:00	14.255000	41.560000	0.077167	226.316667	201.666667	28546.514577	25164.160785	10592.076832	16	5	4	12	2017	364	30	52
2017-12-30 17:00:00	13.775000	44.448333	0.077833	81.493333	103.070000	33979.214195	29879.717705	15039.615845	17	5	4	12	2017	364	30	52
2017-12-30 18:00:00	10.771667	58.323333	0.076500	3.185833	3.251500	37929.531053	32764.651735	16786.554622	18	5	4	12	2017	364	30	52

Power Consumption in the zones



Based on the above boxplot, we can conclude that Zone 1 has the highest power consumption, while Zone 3 has the least.

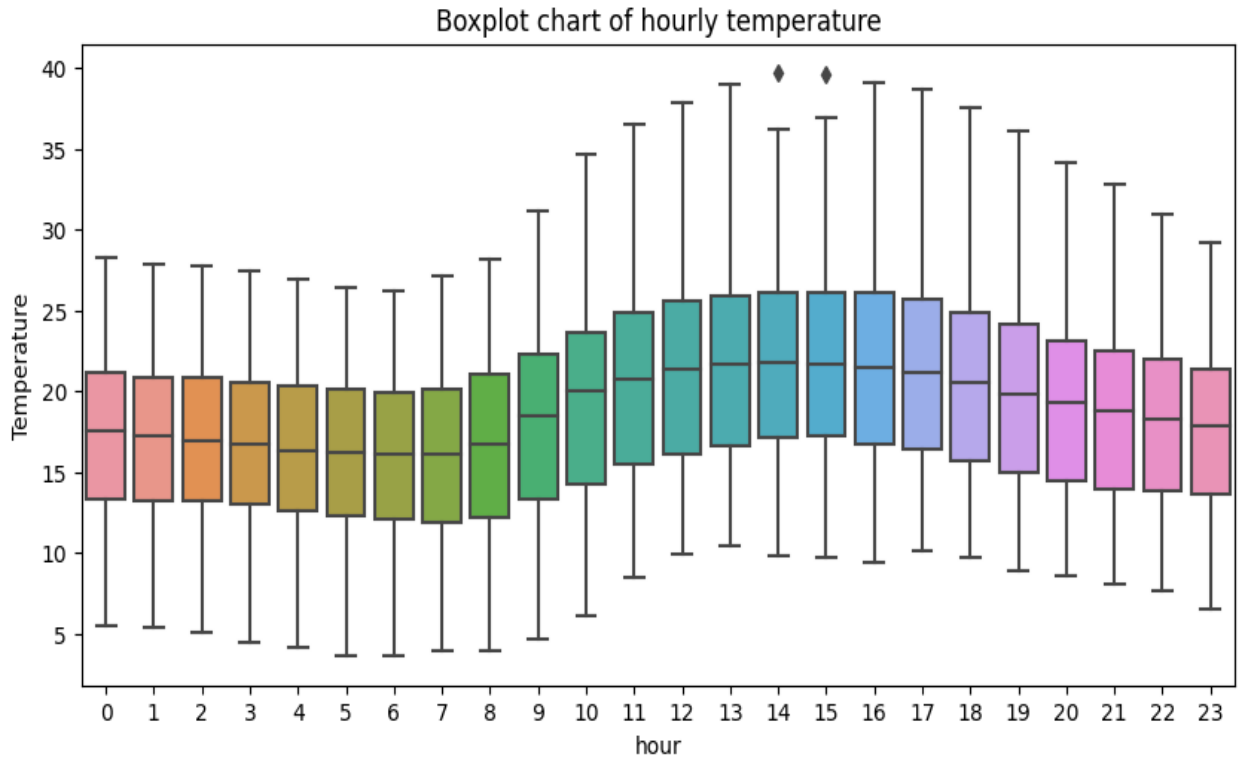
Power Consumption in the zones



Again, based on the above plot we can see how Zone 1 has higher average consumption compared to Zone 2 and Zone 3. On top of that, the average consumption increases during the evening hours for all three areas.

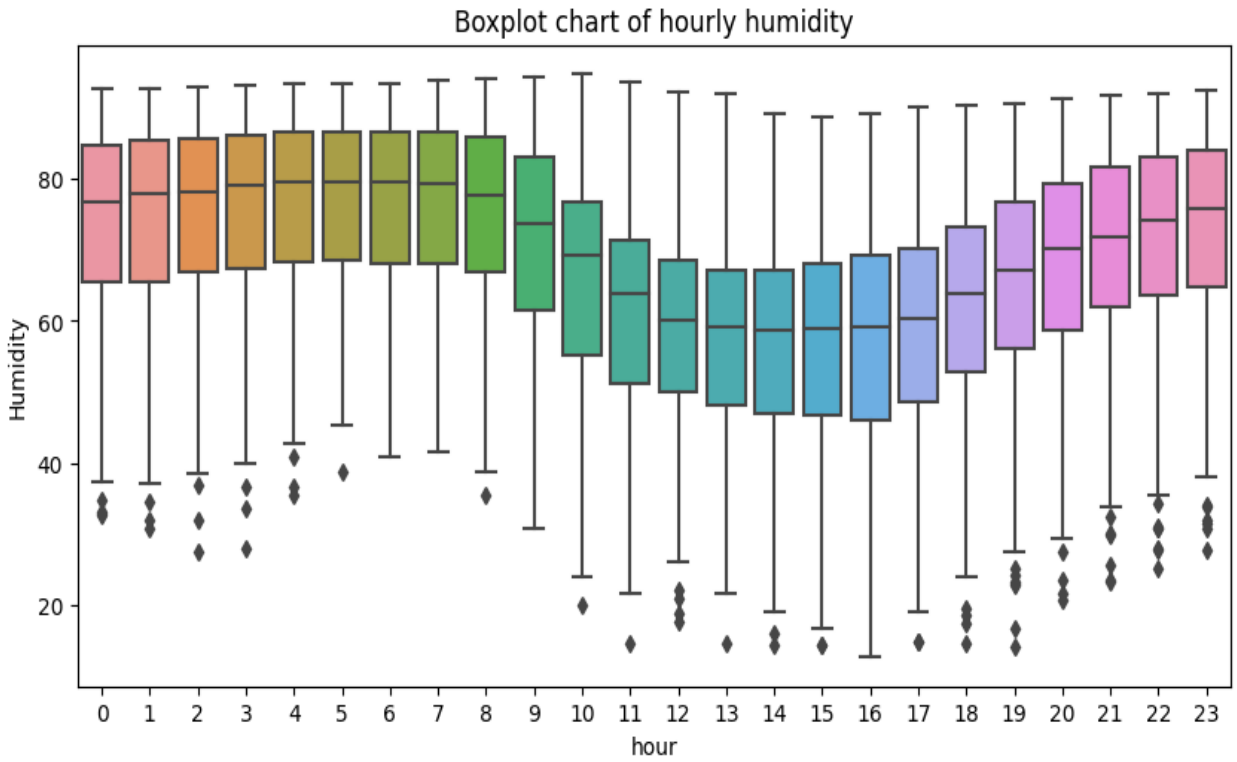
Hourly temperature chart

The above feature has a direct correlation with the hourly temperature. We can observe this below:



Hourly humidity chart

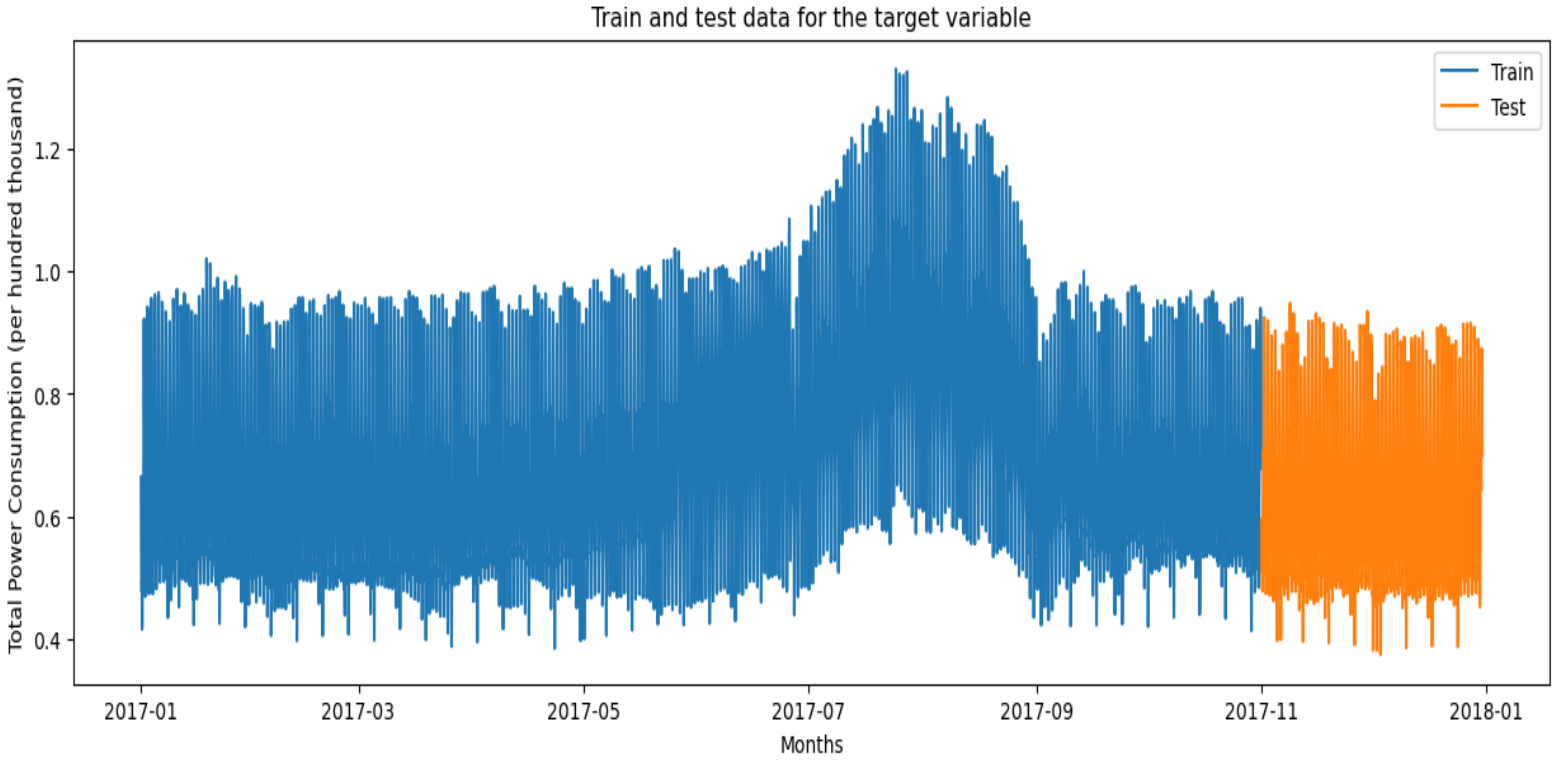
On the other hand, power consumption is inversely correlated with humidity. Hence, humidity follows the exact opposite trend.



Data Preprocessing

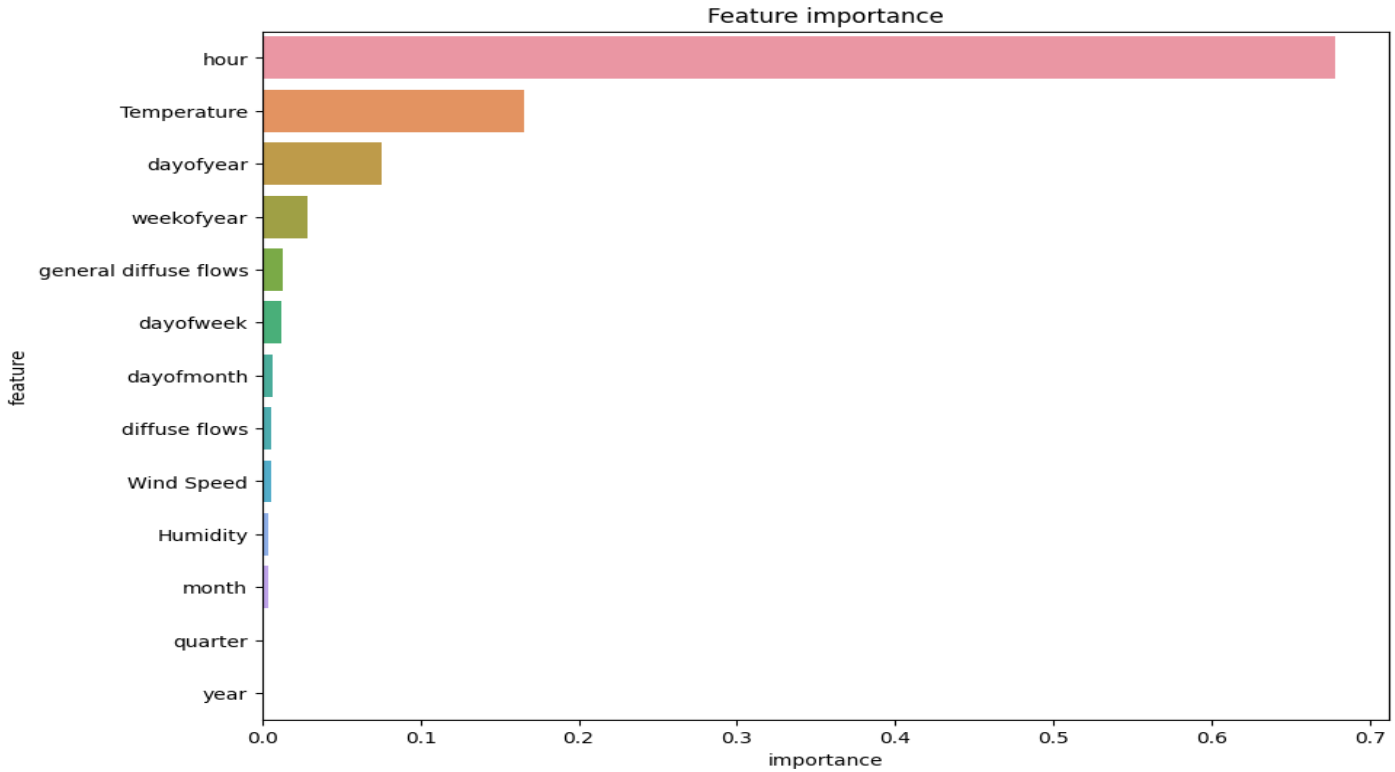
Train-Test split

Here, 70% of the data available is from **January 1st until October 10th**, so that's going to be our **training set**. The remaining 30% of the data is from **October 11th until December 30th**, which is our **test set**.



Feature Importance

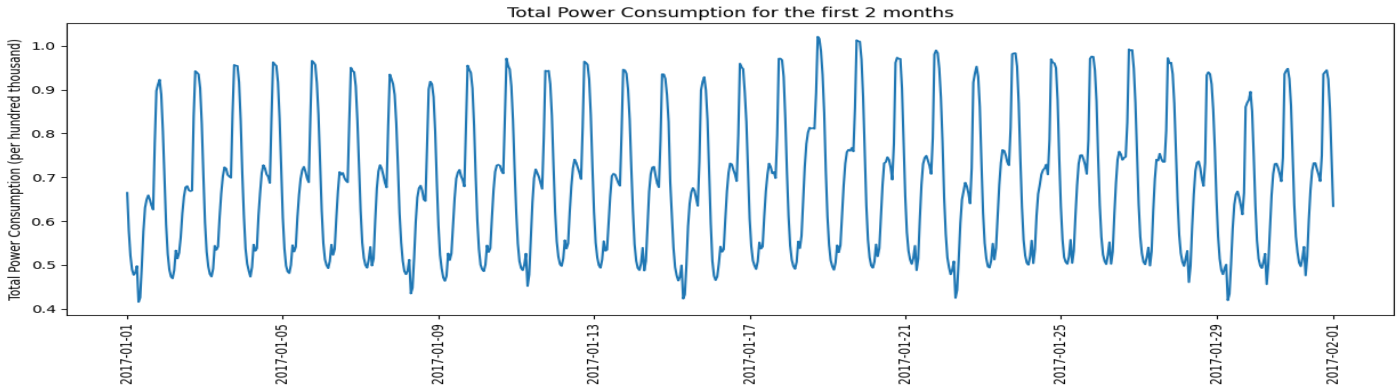
We calculate the feature importance of each of the features in the dataset, using the **Random Forest Regressor** algorithm.



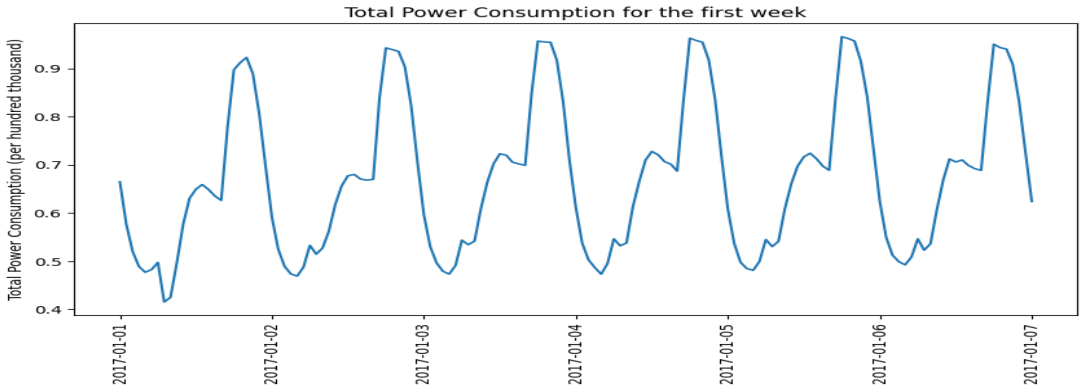
Time Series Analysis

Power Consumption - 1

Firstly, we take a look at the data spanning over the first 2 months, to check the variation of the power consumption over a few months of data.

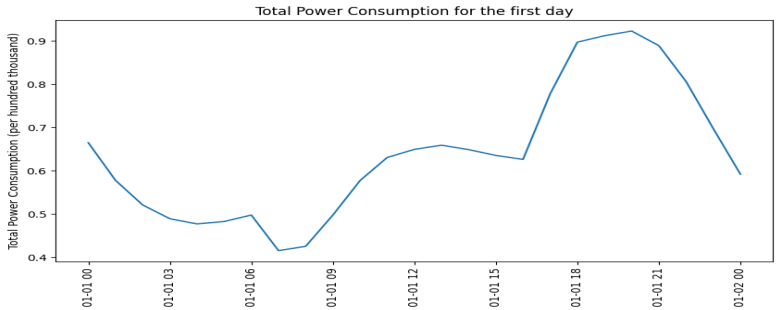


This signifies the presence of a seasonal component in the data. We zoom in further to see the variation in the power consumption over a week.

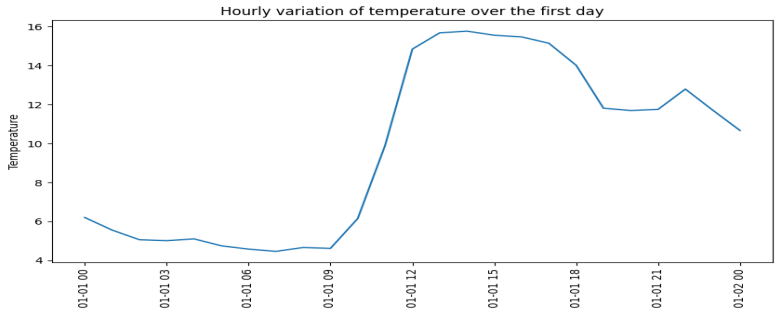


Power Consumption - 2

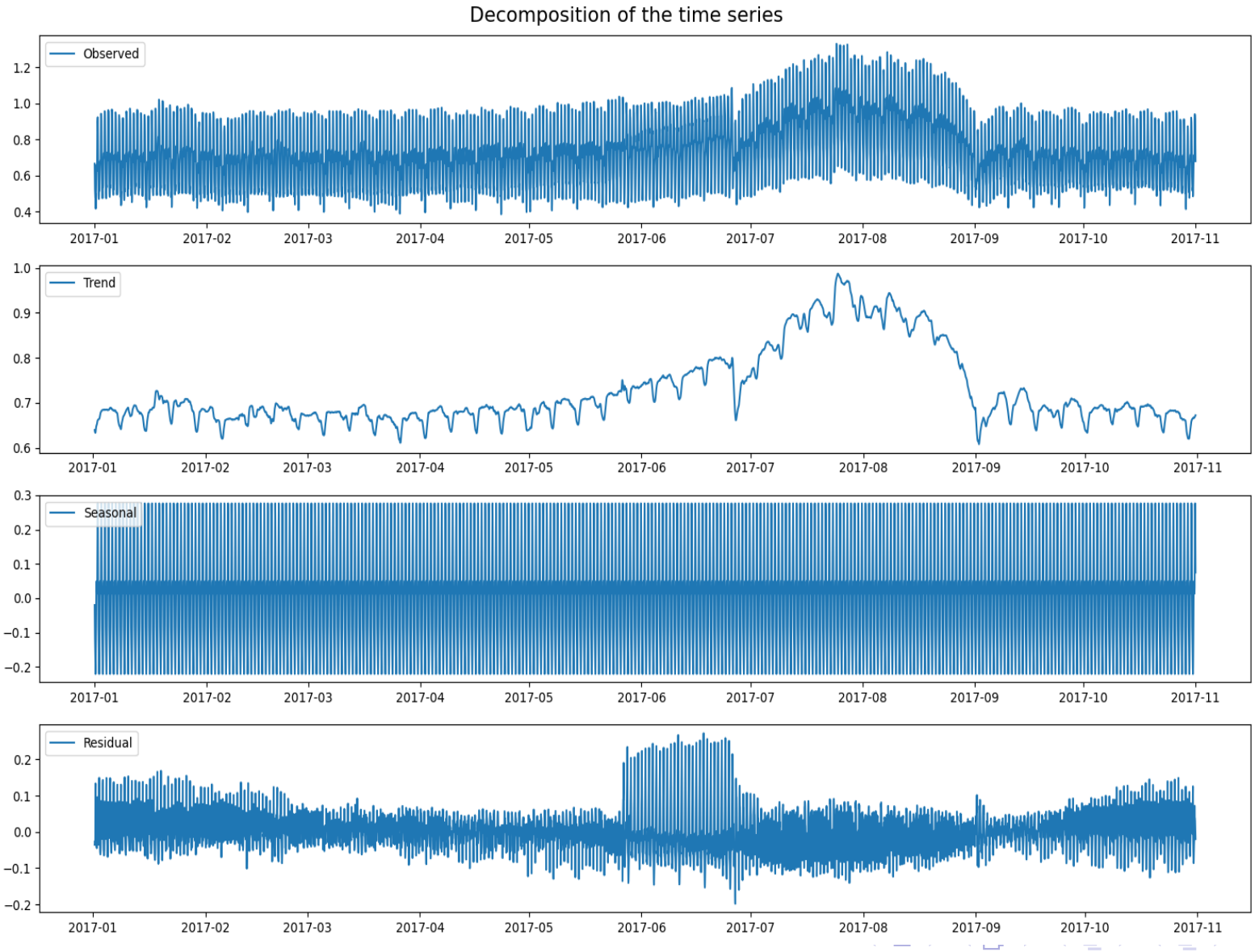
This clearly shows the presence of daily seasonality in the data. Now, we zoom in even further to see the variation in the power consumption over a day, to pin down possible hourly patterns.



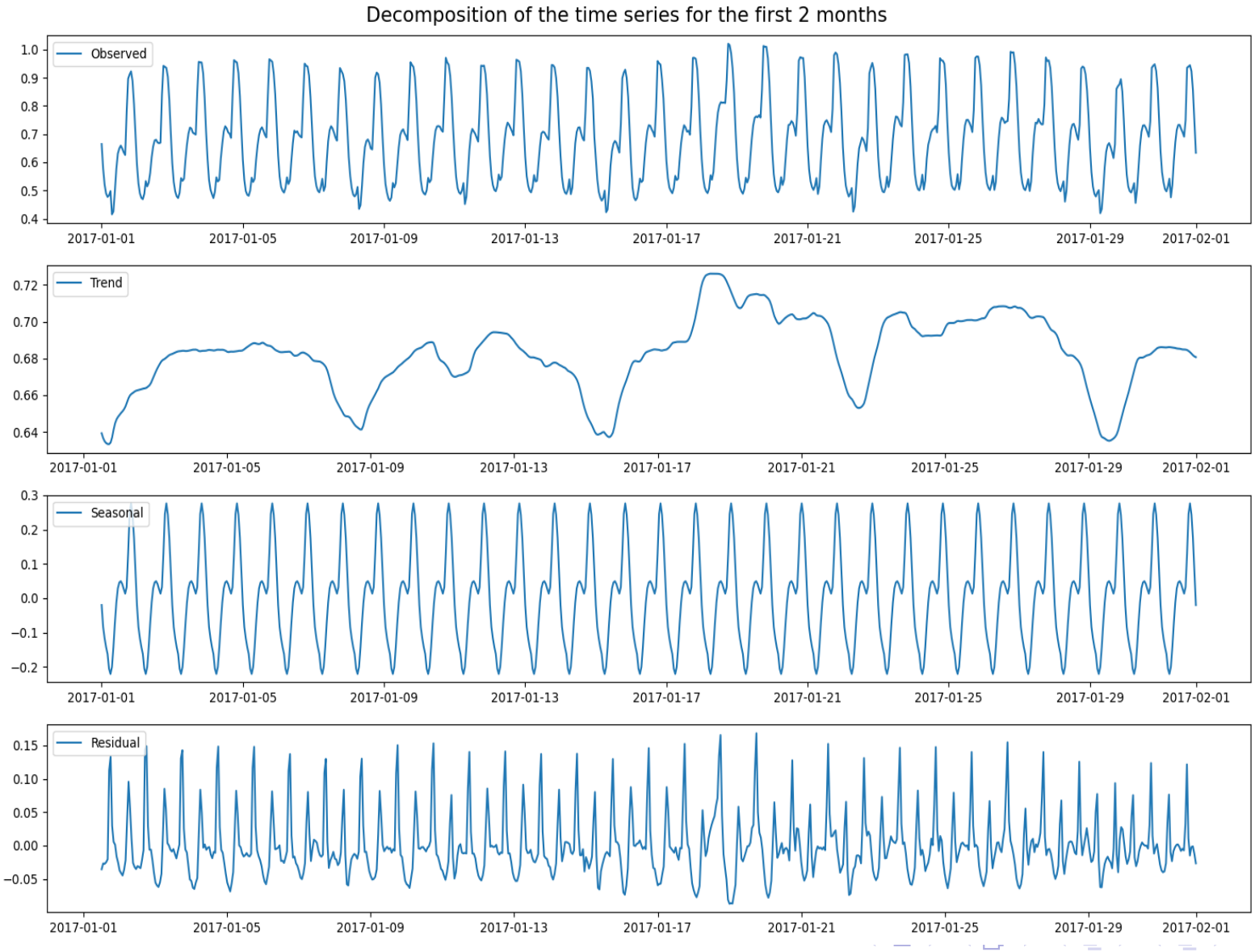
This clearly shows the presence of daily seasonality in the data. Now, we zoom in even further to see the variation in the power consumption over a day, to pin down possible hourly patterns.



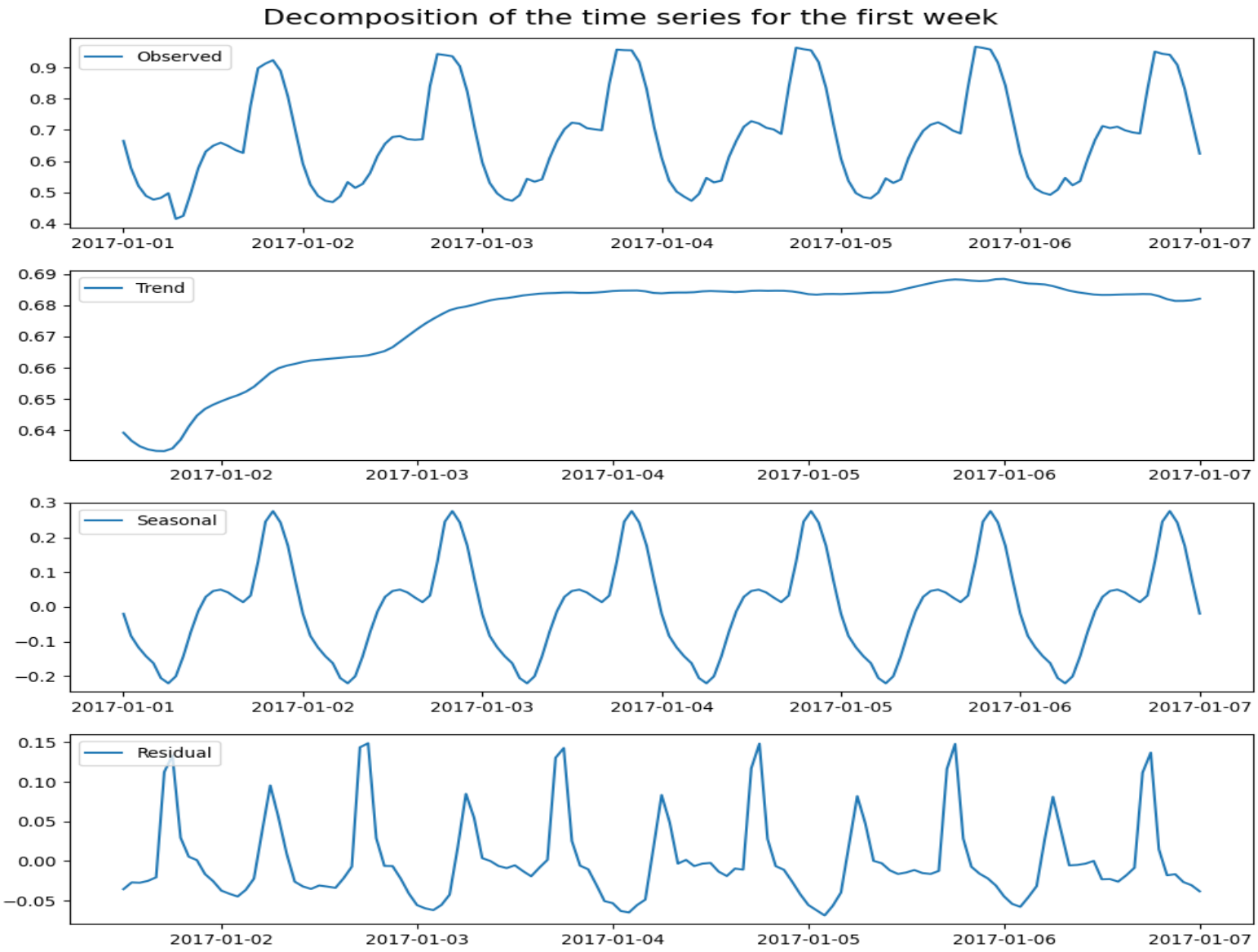
Decomposition - 1



Decomposition - 2



Decomposition - 3



Stationarity Check - ADF test

We use **Augmented Dickey-Fuller** (ADF) test to check for stationarity in the data. The null hypothesis of the ADF test is that the time series is **non-stationary**. The test results comprise of a **test statistic** and some **critical values** for different confidence levels. For **p-values** less than 0.05, we can reject the null hypothesis, i.e. the time series is **stationary**.

ADF-test on observed data

ADF-test Observations (on Original Data):	
Test Statistic	-2.416591
p-value	0.137108
#Lags Used	36.000000
Number of Observations Used	7259.000000
Critical Value (1%)	-3.431251
Critical Value (5%)	-2.861938
Critical Value (10%)	-2.566982

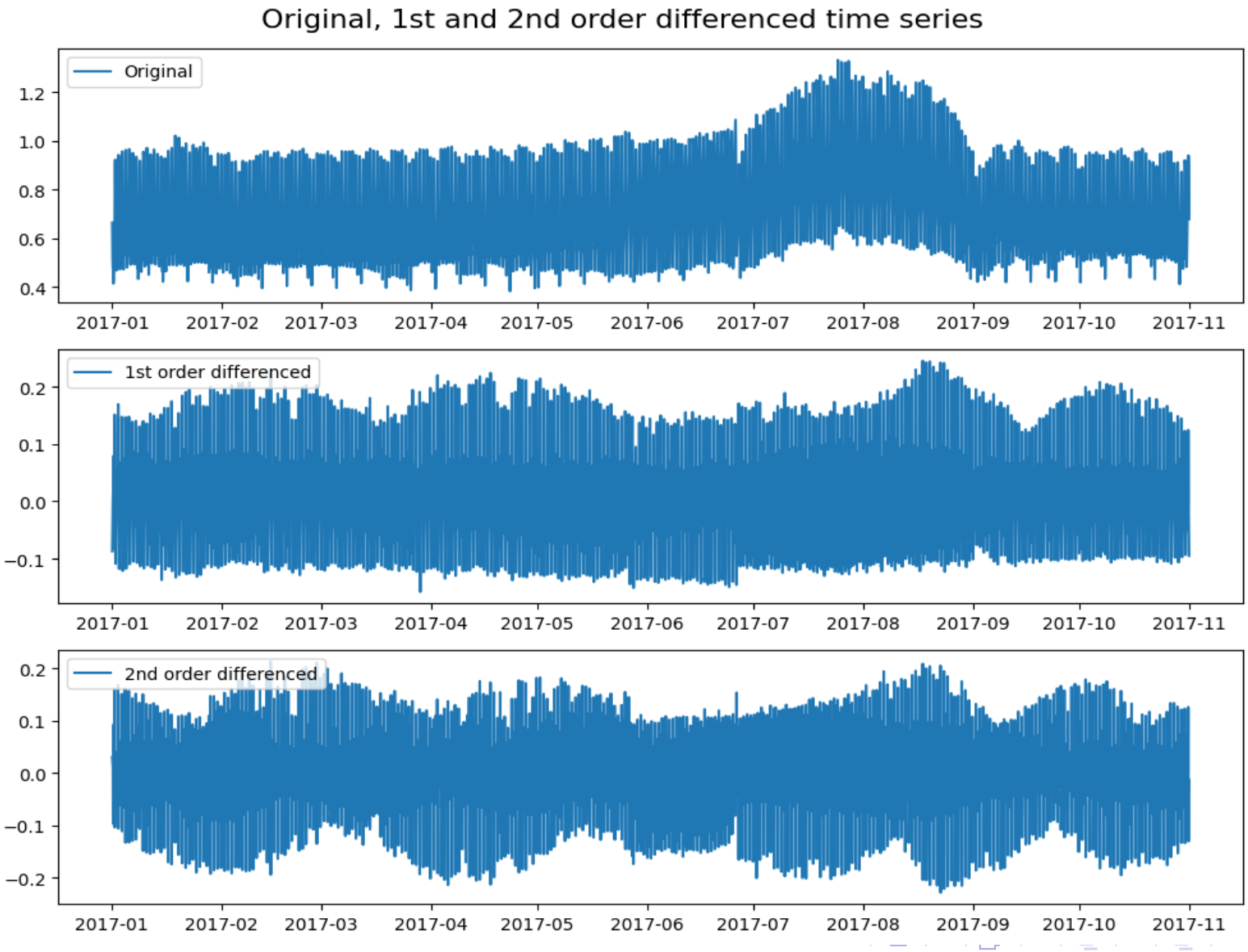
Since the **p-value** (0.13) is greater than 0.05 (significance level of 5%), we fail to reject the null hypothesis and conclude that the time series is **non-stationary**.

ADF-test on first order differenced data

ADF-test Observations: (on First Order Differenced Data)	
Test Statistic	-19.776232
p-value	0.000000
#Lags Used	36.000000
Number of Observations Used	7258.000000
Critical Value (1%)	-3.431251
Critical Value (5%)	-2.861938
Critical Value (10%)	-2.566982

Since, the **p-value** (0.0) is smaller than 0.05 (significance level of 5%), we reject the null hypothesis and conclude that the first order differenced time series is **stationary**.

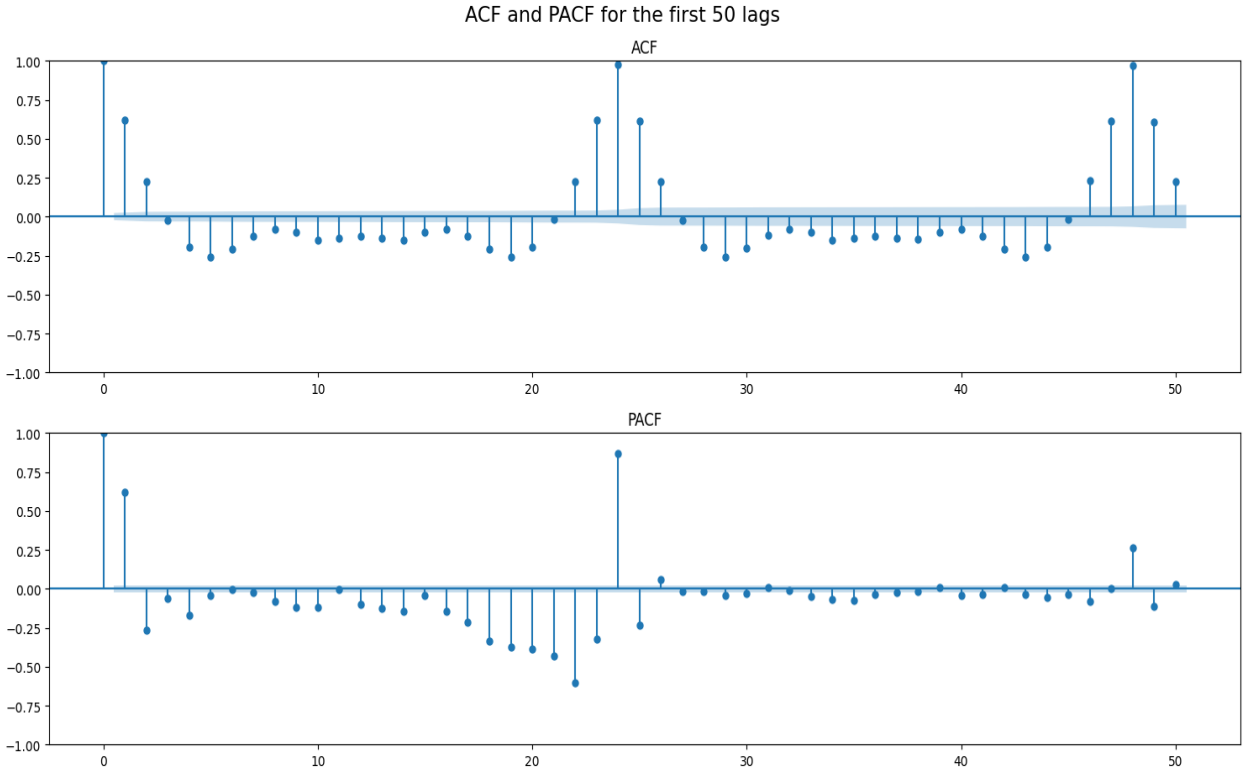
Differencing



ACF and PACF

Before trying to fit classical time series models like ARIMA, we need to find the optimal parameters for the model.
We do this by plotting the **Autocorrelation Function (ACF)** and **Partial Autocorrelation Function (PACF)** plots to find the optimal values of p and q for the first and second order difference data (as they are stationary based on ADF-Test).

ACF PACF - 1st order differenced time series



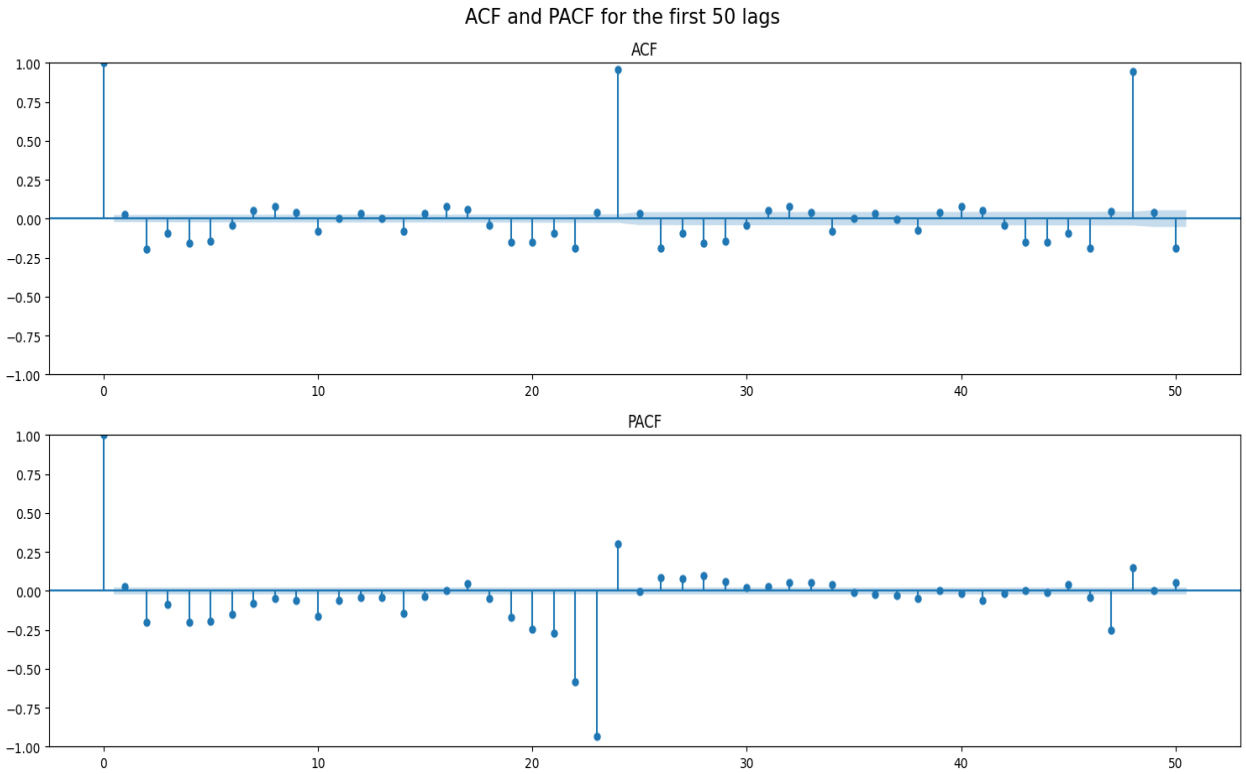
We observe sudden peaks near `lag = 24` and `lag = 24*2 = 48`. These peaks signify the presence of **daily** seasonality in the data.

Based on the `ACF` and `PACF` plots above we also decide upon the following parameters for the `ARIMA` model:

- `d = 1` : This represents the no. of differencing required to make the time series stationary.
- `q = 3` : This represents the no. of moving average terms in the series. After inspecting the `ACF` plot, we can see that the first 3 lags are significant, so we choose `q = 3`.
- `p = 3` : This represents the no. of autoregressive terms in the series. After inspecting the `PACF` plot, we can see that the first 3 lags are significant, so we choose `p = 3`.

So, we try fitting a `ARIMA(3, 1, 3)` model to the data.

ACF PACF - 2nd order differenced time series



As before, we again observe sudden peaks near `lag = 24` and `lag = 24*2 = 48`. These peaks signify the presence of **daily** seasonality in the data.

Again, based on the `ACF` and `PACF` plots above we decide upon the following parameters for the `ARIMA` model:

- `d = 2` : This represents the no. of differencing required to make the time series stationary.
- `q = 1` : This represents the no. of moving average terms in the series. After inspecting the `ACF` plot, we can see that the first lag is significant, so we choose `q = 1`.
- `p = 1` : This represents the no. of autoregressive terms in the series. After inspecting the `PACF` plot, we can see that the first lag is significant, so we choose `p = 1`.

So, we try fitting a `ARIMA(1, 2, 1)` model to the data.

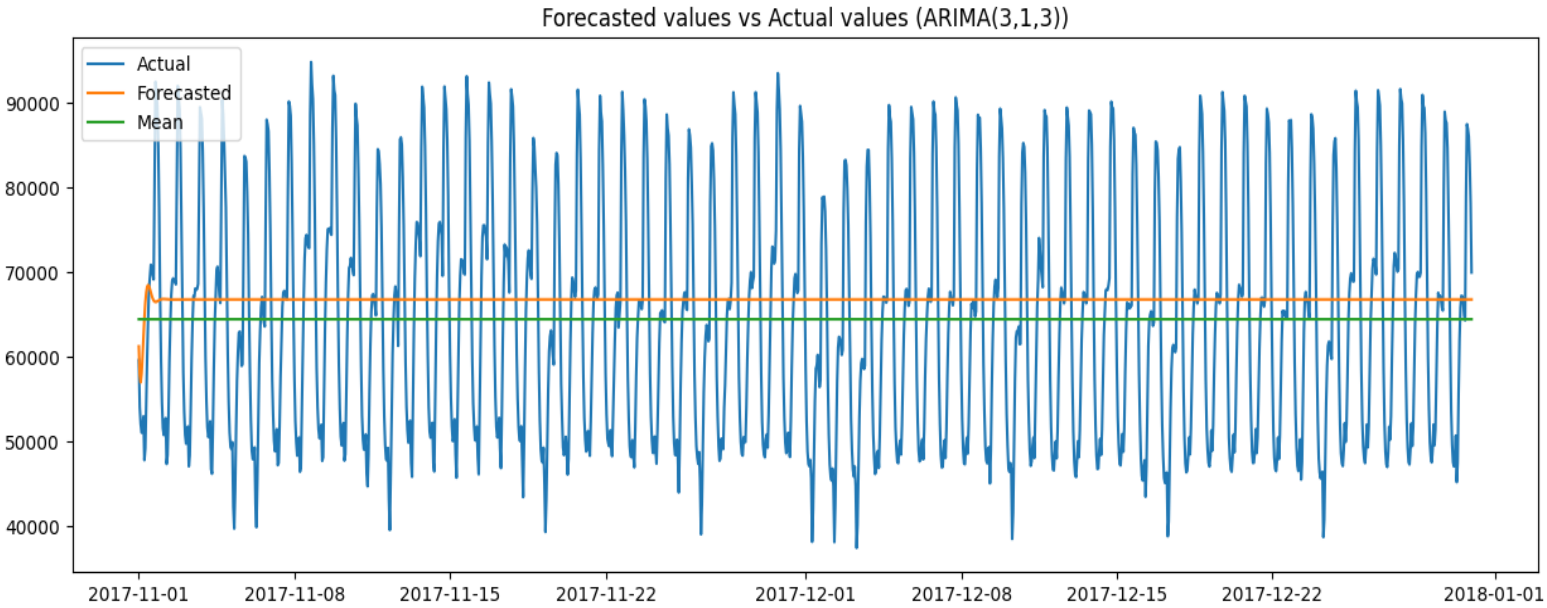
ARIMA (3,1,3)

SARIMAX Results

Dep. Variable:	Total Power Consumption	No. Observations:	7296
Model:	ARIMA(3, 1, 3)	Log Likelihood	12253.014
Date:	Thu, 20 Apr 2023	AIC	-24492.027
Time:	09:15:42	BIC	-24443.763
Sample:	01-01-2017	HQIC	-24475.431
	- 10-31-2017		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.7186	0.096	7.467	0.000	0.530	0.907
ar.L2	0.5080	0.121	4.196	0.000	0.271	0.745
ar.L3	-0.5042	0.047	-10.718	0.000	-0.596	-0.412
ma.L1	-0.0889	0.095	-0.932	0.351	-0.276	0.098
ma.L2	-0.8388	0.060	-14.052	0.000	-0.956	-0.722
ma.L3	-0.0570	0.041	-1.379	0.168	-0.138	0.024
sigma2	0.0020	2.04e-05	99.638	0.000	0.002	0.002

Ljung-Box (L1) (Q):	0.46	Jarque-Bera (JB):	8580.73
Prob(Q):	0.50	Prob(JB):	0.00
Heteroskedasticity (H):	1.02	Skew:	1.17
Prob(H) (two-sided):	0.70	Kurtosis:	7.77



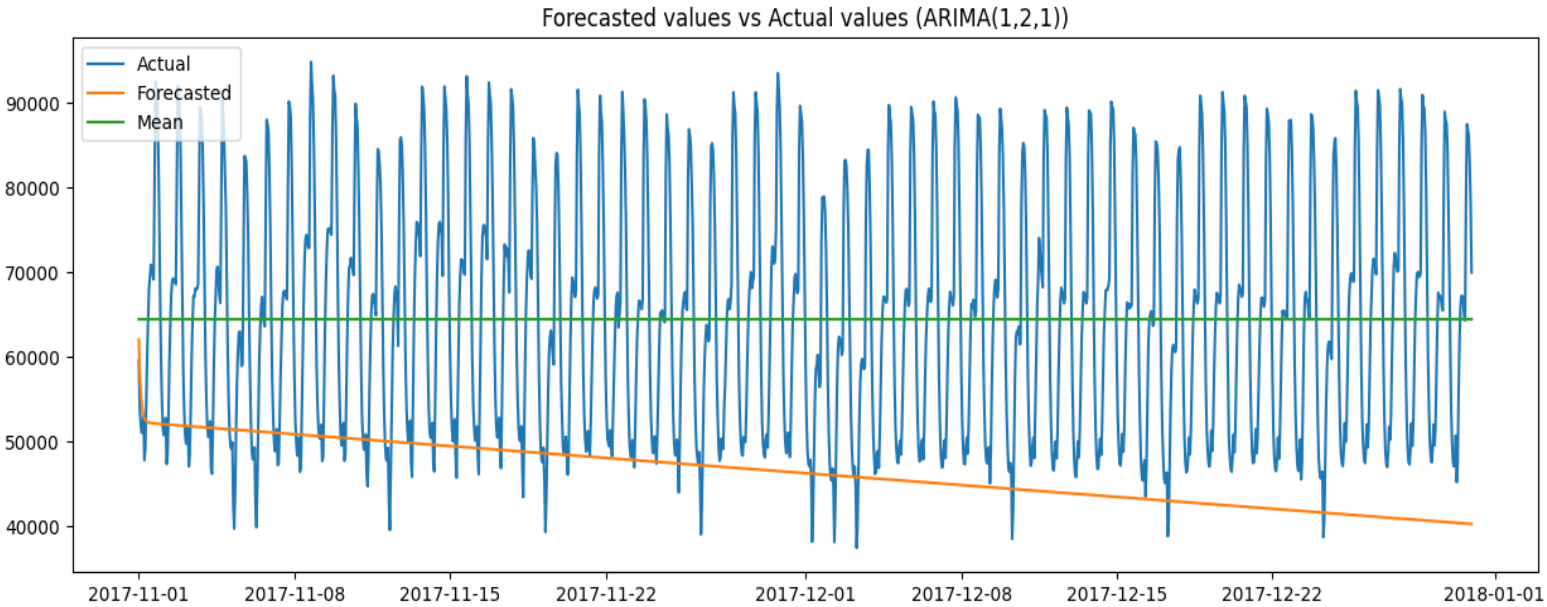
Arima (1,2,1)

SARIMAX Results

Dep. Variable:	Total Power Consumption	No. Observations:	7296
Model:	ARIMA(1, 2, 1)	Log Likelihood	11402.593
Date:	Thu, 20 Apr 2023	AIC	-22799.186
Time:	09:34:05	BIC	-22778.501
Sample:	01-01-2017 - 10-31-2017	HQIC	-22792.073
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.6230	0.011	58.207	0.000	0.602	0.644
ma.L1	-1.0000	0.221	-4.534	0.000	-1.432	-0.568
sigma2	0.0026	0.001	4.676	0.000	0.001	0.004

Ljung-Box (L1) (Q):	198.00	Jarque-Bera (JB):	2331.66
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	1.04	Skew:	0.70
Prob(H) (two-sided):	0.32	Kurtosis:	5.39

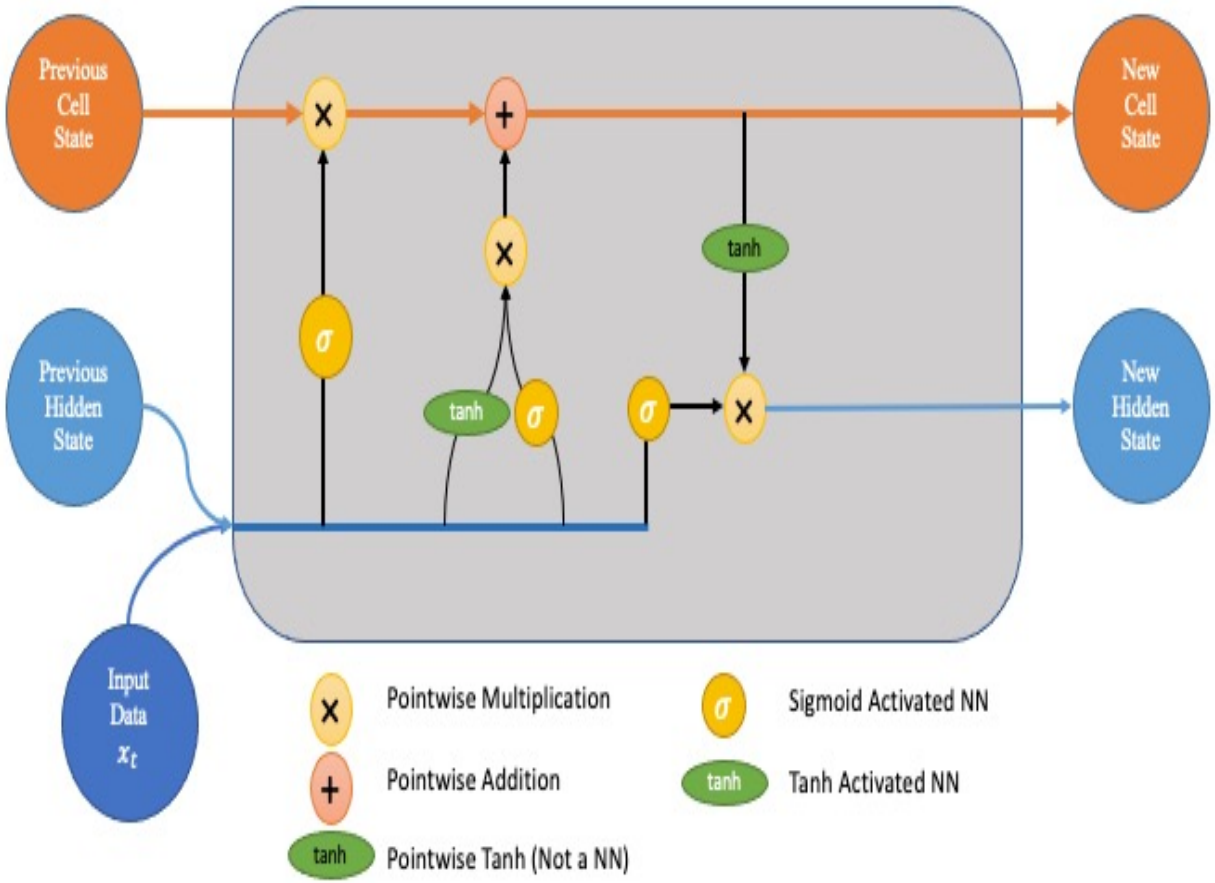


Neural Networks

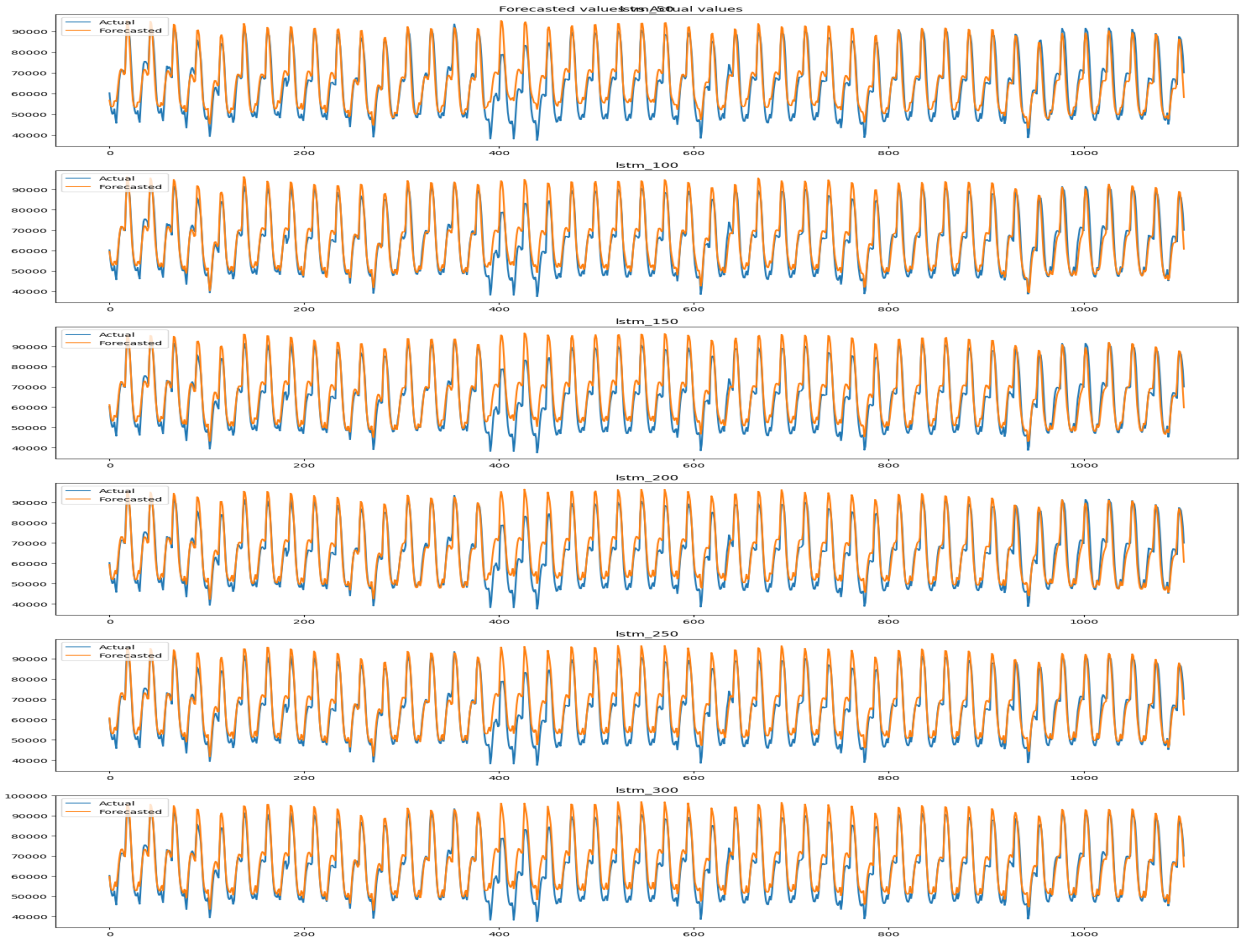
Generating Sequences

- For neural networks we consider hourly data spanning over exactly two weeks into the past. So we use a sequence of length 336.
- We convert this time series data into a supervised learning problem by creating sequences of data. We use the *create_sequences* function to generate sequences of data.
- Each data point in the sequence contains 8 features which makes each training example a tensor of size (336,8).

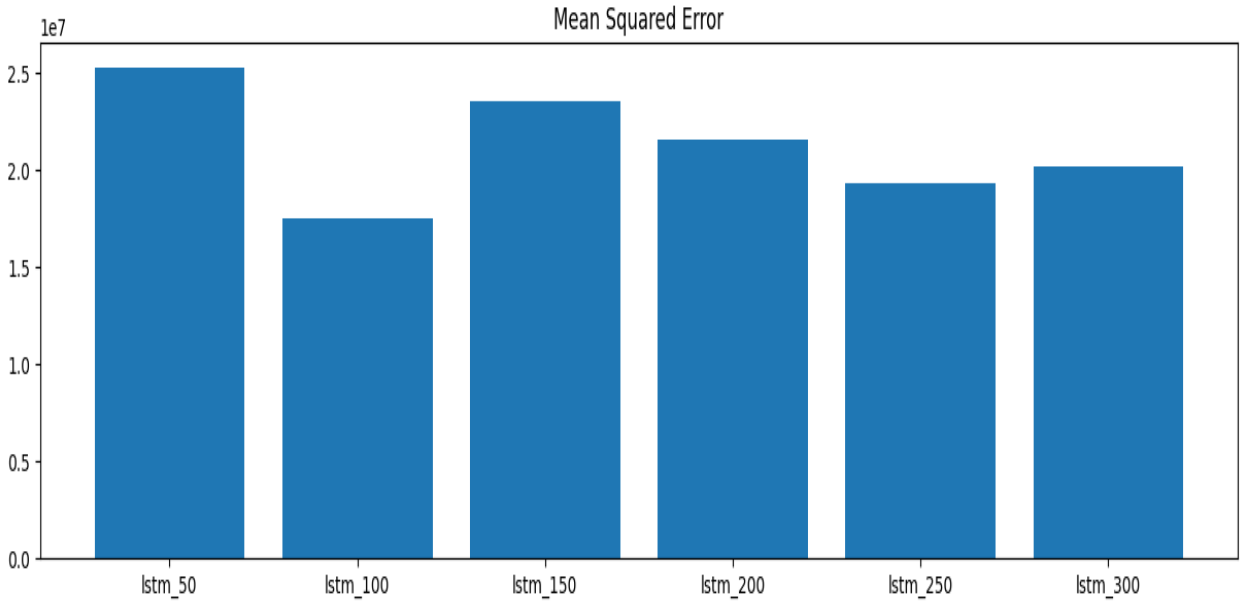
LSTM


Previous Cell State
New Cell State
Previous Hidden State
New Hidden State
Input Data x_t
tanh
σ
Pointwise Multiplication
Pointwise Addition
Pointwise Tanh (Not a NN)
Sigmoid Activated NN
Tanh Activated NN

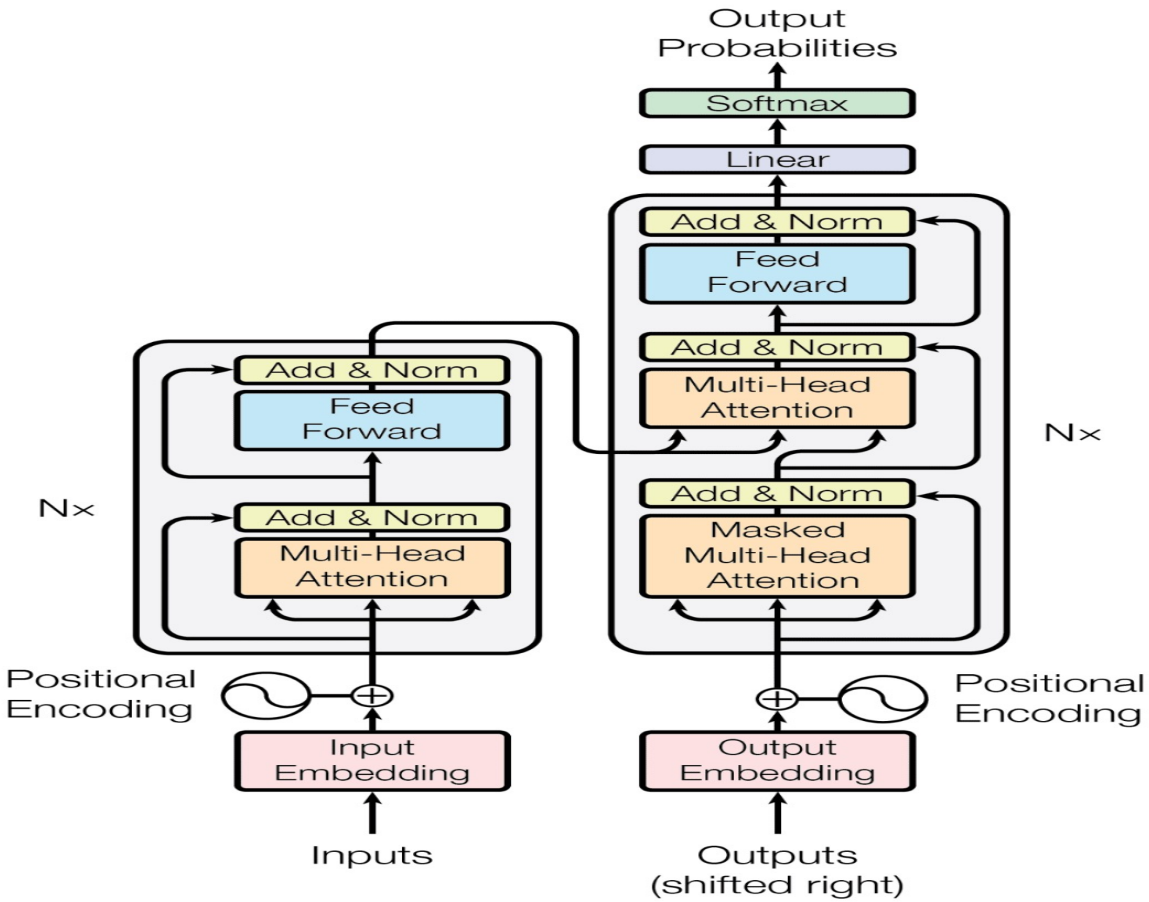
LSTM-Results



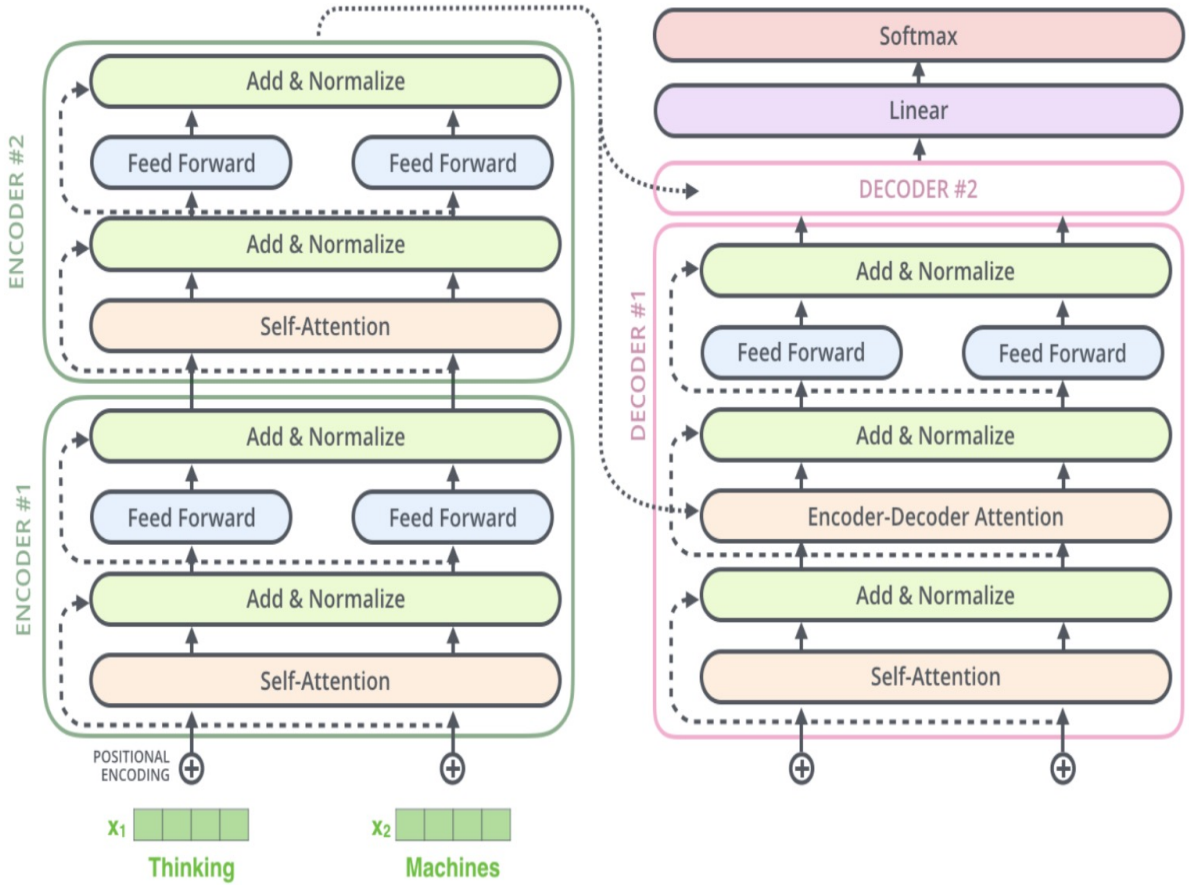
LSTM-MSE


Mean Squared Error
1e7
2.5
2.0
1.5
1.0
0.5
0.0
lstm_50
lstm_100
lstm_150
lstm_200
lstm_250
lstm_300

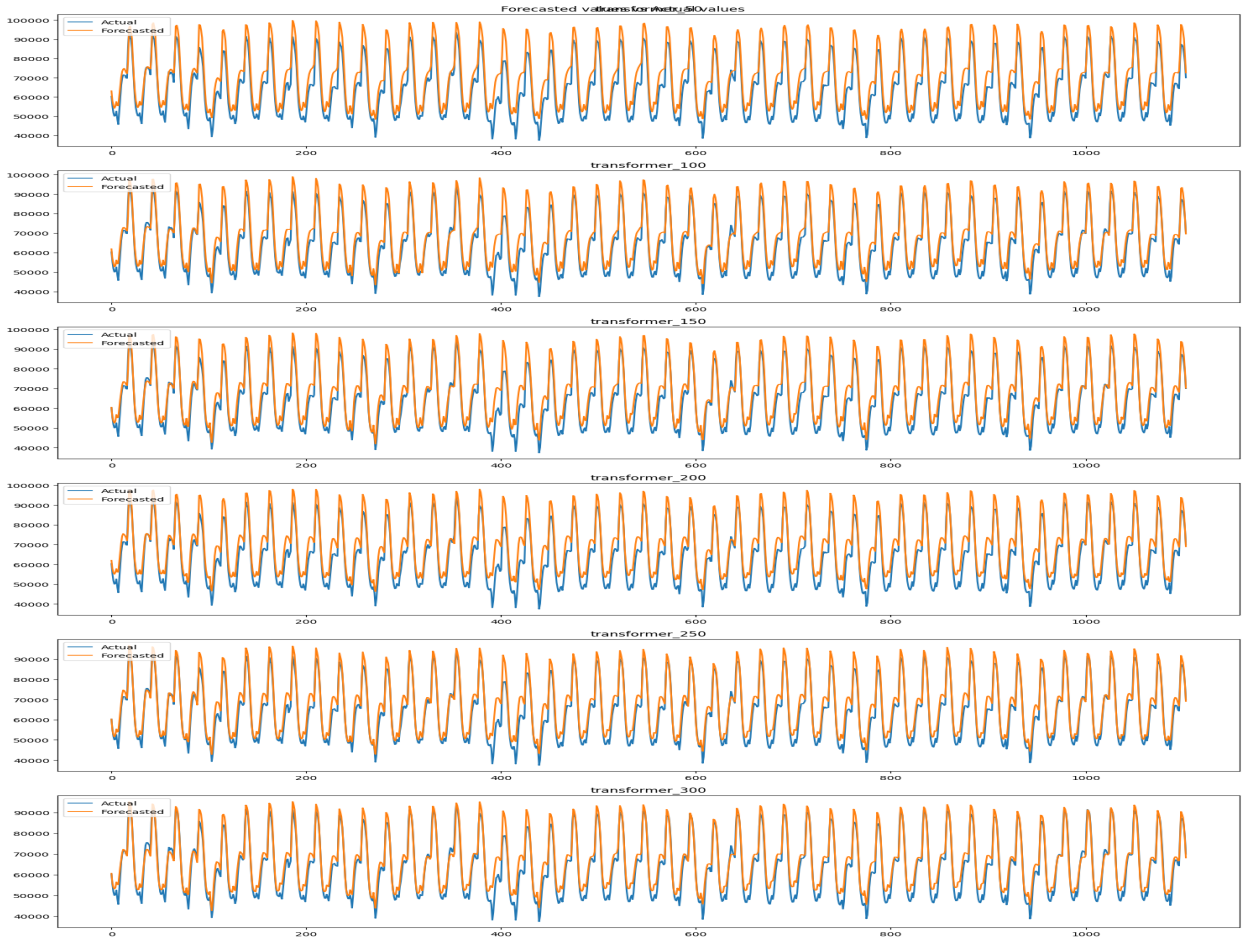
Transformer



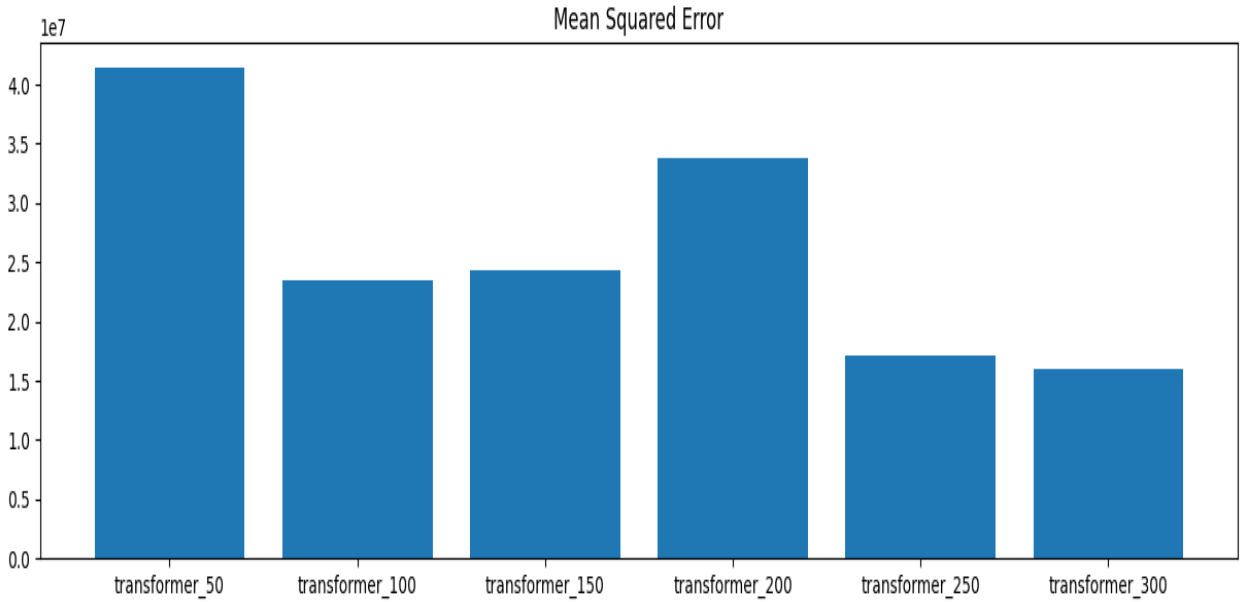
Transformer



Transformer-results



Transformer-MSE



Results

Models	MSE on Test Data (MW^2)
ARIMA(3,1,3)	203,237,826.00
ARIMA(1,2,1)	539,569,276.00
LSTM (Best)	17,565,033.00
Transformer (Best)	**15,999,286.00**

MSE Ratio	MSE Ratio
ARIMA(3,1,3)/Transformer	12.70293099
ARIMA(1,2,1)/Transformer	33.72458471
LSTM/Transformer	1.097863555

Things to Ponder

- Why did we smoothen the original data from 10 minutes intervals to hourly intervals?
- Could the Teacher Forcing method have improved the forecasting performance?(For both neural networks and ARIMA)

THANK YOU