Time Series Term Project

Prediction of Prices of Electricity

Timur Abdygulov
Reza Jafari, Ph.D
December 11, 2023

Table of Contents

Table of figures and tables.

ABSTRACT

This study focuses on the application of time series analysis to predict the hourly prices of electricity of the data center in Ireland. By analyzing historical price data, including variables such as temperature, wind energy production, national system load, and wind speed, a robust predictive model is developed. The goal of the project is to produce precise and trustworthy forecasts so that people, businesses, and government agencies may plan ahead and make decisions based on expected price patterns. The findings demonstrate how well time series analysis can forecast hourly pricing, which benefits Ireland's resource allocation, risk management, and planning.

INTRODUCTION

This report presents an in-depth analysis of prices of electricity time series data for the ‘Data center’ located in Ireland. With the goal of creating precise prediction models and learning more about the patterns of hourly electricity costs. The analysis includes essential steps such as dataset cleaning, stationarity checks, order determination for the ARIMA process and performance comparison of various models. The objective is to identify the most effective model for predicting prices of electricity.

Cleaning the dataset to guarantee data integrity by correcting missing values, contradictions, and incorrect entries is the first step in the analytical process. In order to minimize their influence on the ensuing modeling procedure.

A stationarity check will be performed in order to accommodate linear time series models. An important supposition for many modeling approaches, stationarity, will be evaluated with the relevant tools. The appropriate transformations will be used if necessary to establish stationarity.

Techniques like the Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), and Generalized Partial Autocorrelation (GPAC) will be used to establish the order of the ARIMA process. By determining the ideal lag structure and parameters, these methods help to enhance the ARIMA model's prediction power.

Different base models will be used in addition to linear models to reflect the intricacies and dynamism of the time series. In order to assess the predicting accuracy of these models, relevant performance measures will be used for training and evaluation.

The purpose of this thorough investigation is to determine the most accurate forecasting model and offer insightful information about the hourly power rates in "Data center." The results will aid in better planning, comprehension, and decision-making.

DESCRIPTION OF DATASET

Initial data was sampled every 30 min and two independent features had a lot of missing values. To address the issue of numerous missing values in the original dataset, a decision was made to resample the data to a 2 hourly frequency. By transitioning to a 2 hourly dataset, we assume to have a more comprehensive and reliable dataset for analysis, free from the gaps caused by missing values.

— Dependent Variable: In this analysis, the dependent variable chosen for the study is the actual prices of electricity at the rime of a record, represented as 'SMPEP2' in the dataset. The unit of prices used in this analysis we assume is ¢/kWh as there is no background on unit measured.

— Independent Variables:

- DateTime: defines date and time of sample
- Holiday: gives name of holiday
- HolidayFlag: 1 if day is a holiday, zero otherwise
- DayOfWeek: (0-6), 0 monday, day of week
- WeekOfYear: running week within year of this date
- Day integer: day of the date
- Month integer: month of the date
- Year integer: year of the date
- PeriodOfDay: denotes half hour period of day (0-47)
- ForecastWindProduction: the forecasted wind production for this period
- SystemLoadEA: the national load forecast for this period, the amount of electricity being consumed or demanded
- SMPEA: the price forecast for this period
- ORKTemperature: the actual temperature measured at Cork airport
- ORKWindspeed: the actual windspeed measured at Cork airport
- CO2Intensity: the actual CO2 intensity in (g/kWh) for the electricity produced
- ActualWindProduction: the actual wind energy production for this period
- SystemLoadEP2: the actual national system load for this period
- SMPEP2: the actual price of this time period, the value to be forecasted

— Preprocessed Dataset:

```
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   DateTime                38014 non-null  object
 1   Holiday                 1536 non-null   object
 2   HolidayFlag             38014 non-null  int64
 3   DayOfWeek               38014 non-null  int64
 4   WeekOfYear              38014 non-null  int64
 5   Day                     38014 non-null  int64
 6   Month                   38014 non-null  int64
 7   Year                    38014 non-null  int64
 8   PeriodOfDay             38014 non-null  int64
 9   ForecastWindProduction  38014 non-null  object
 10  SystemLoadEA            38014 non-null  object
 11  SMPEA                   38014 non-null  object
 12  ORKTemperature          38014 non-null  object
 13  ORKWindspeed            38014 non-null  object
 14  CO2Intensity            38014 non-null  object
 15  ActualWindProduction    38014 non-null  object
 16  SystemLoadEP2           38014 non-null  object
 17  SMPEP2                  38014 non-null  object
dtypes: int64(7), object(11)
```

Table 1: Original Dataset

Table 1 in highlighted area shows us that numerical features are objects that are in need of transformation. Other columns except Holiday do not hold any necessary information, we can drop them and transform ‘Holiday Flag’ to category as its binary feature.

After above mentioned steps we loose some data.

```
Missing values in the entire dataset:
HolidayFlag                 0
ForecastWindProduction      5
SystemLoadEA                2
SMPEA                       2
ORKTemperature            295
ORKWindspeed              299
CO2Intensity                7
ActualWindProduction        5
SystemLoadEP2               2
SMPEP2                      2
dtype: int64
```

Table 2: Table of Missing values

To address this issue we resample data into 2 hours data set. That leaves us with only 21 missing observations in ‘ORKTemperature’ and ‘ORKWindspeed’. And using ‘moving average’ method (performed by hand) we filling those missing values. That leaves us with final dataset with 9504 total observations below in Table 3.

```
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   HolidayFlag            9504 non-null   category
 1   ForecastWindProduction 9504 non-null   float64
 2   SystemLoadEA           9504 non-null   float64
 3   SMPEA                  9504 non-null   float64
 4   ORKTemperature         9504 non-null   float64
 5   ORKWindspeed           9504 non-null   float64
 6   CO2Intensity           9504 non-null   float64
 7   ActualWindProduction   9504 non-null   float64
 8   SystemLoadEP2          9504 non-null   float64
 9   SMPEP2                 9504 non-null   float64
dtypes: category(1), float64(9)
```

Table 3: Clean Dataset

Head of the dataset is bellow as follows:

	HolidayFlag	ForecastWindProduction	...	SystemLoadEP2	SMPEP2
DateTime			...		
2011-11-01 00:00:00	0.0	325.3200	...	2923.1500	54.0625
2011-11-01 02:00:00	0.0	343.1275	...	2586.9150	39.8700
2011-11-01 04:00:00	0.0	343.8800	...	2580.5300	39.8700
2011-11-01 06:00:00	0.0	332.8725	...	3217.5875	52.3425
2011-11-01 08:00:00	0.0	388.7925	...	4185.2900	57.3000

Table 4: Head of dataset

— Plot of Dependent Variable VS Time

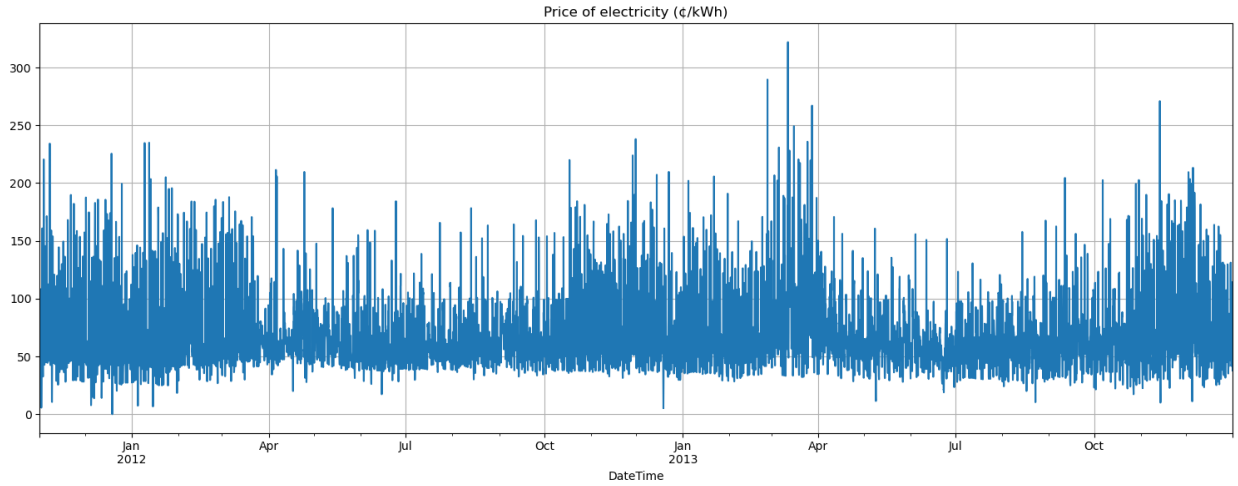


Figure 1: Dependent Variable vs Time

Above plot of dependent feature exhibit some trend pattern. It is not clear where there exists seasonality pattern, further tests are required.

— ACF/PACF of the Dependent Variable

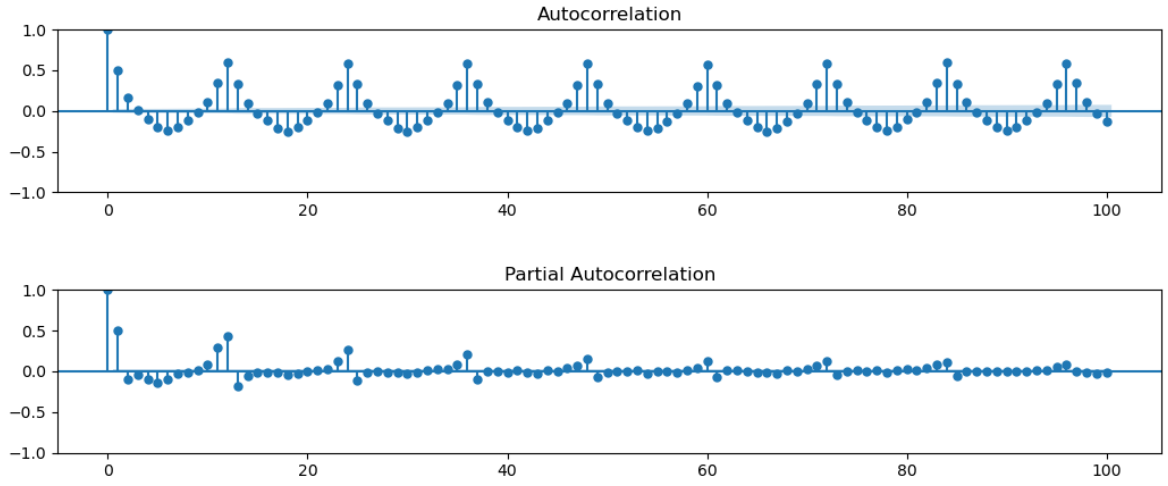


Figure 2: ACF/PACF of dependent feature

From the ACF plot above, we can see that the ACF values have high peaks every 12 lags indicating for high seasonality in data. In the PACF there is cutoff observed at the first lag. The evidence suggests that the process we have here is an AR process with order of 1 and more. We will get more insights on the order of ARMA process in the later section of the report. In the report's subsequent portion, we will learn more about the ARMA process's sequence.

— Correlation Matrix using seaborn heatmap

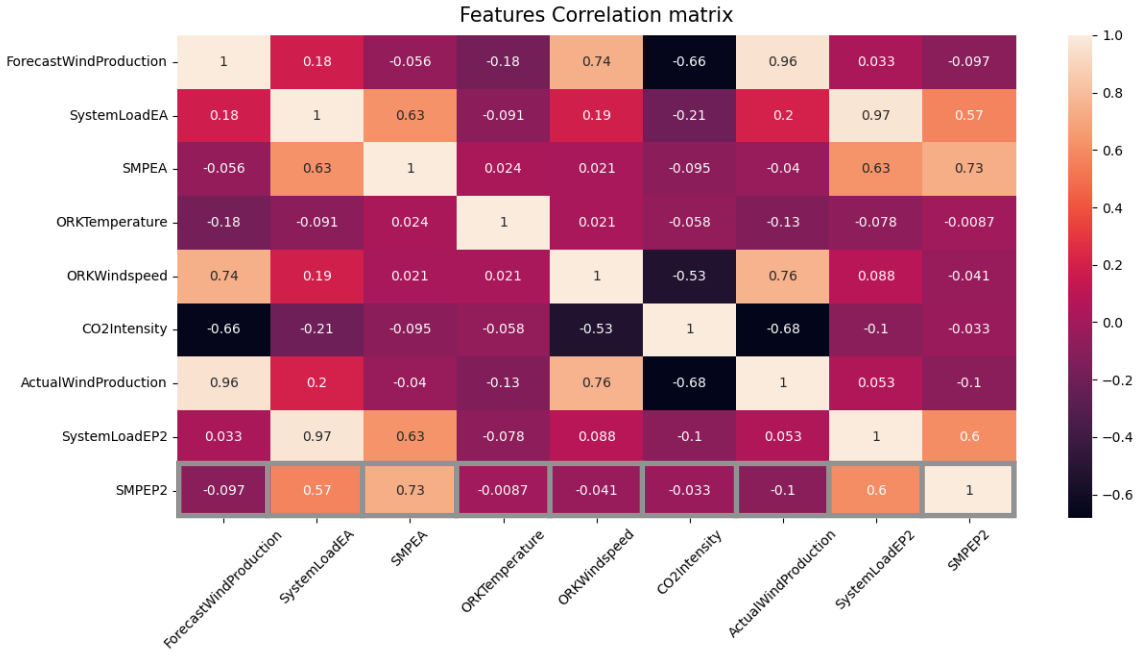


Figure 3: Correlation Matrix

Pearson's correlation coefficient heat map above shows us that there is a liner correlation with all features in highlighted grey area. There is negative correlation with temperature, Windproduction, Windspeed and CO2 with expected positive correlation with SystemLoad feature.

— Splitting the Dataset

```
The shapes of X_train:(7603, 10) and y_train:(7603,)
The shapes of X_test:(1901, 10) and y_test:(1901,)
```

Table 5: Head of dataset

STATIONARITY

— Rolling Mean and Variance Plot

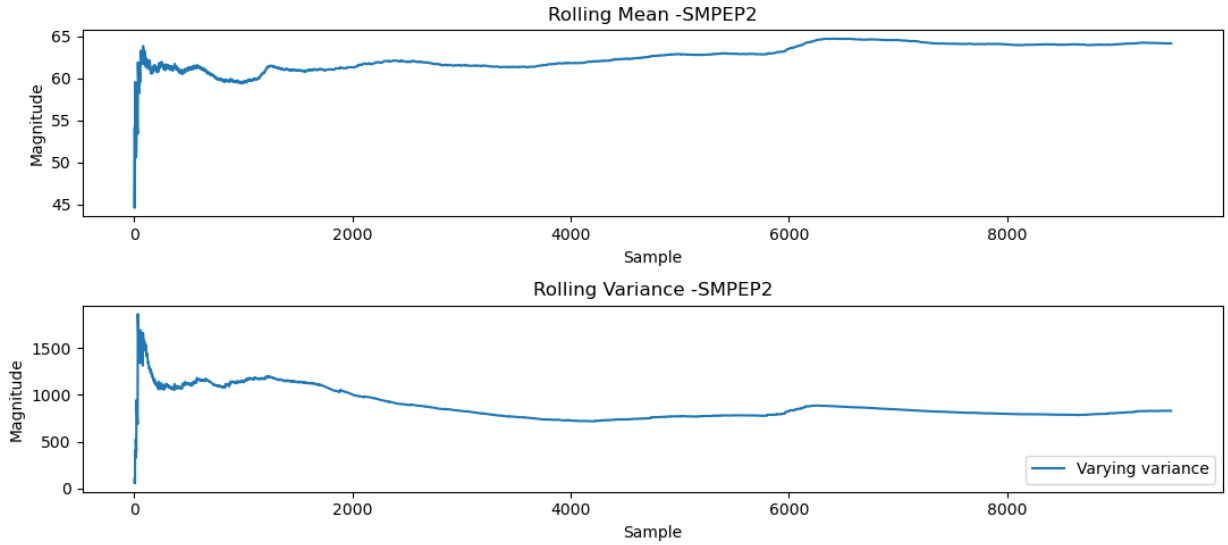


Figure 4:Rolling Mean and Variance of clean data

The Rolling mean and Rolling Variance plots of dependent variable on raw data shows that after a few early samples the mean and variance stabilizes and becomes constant. These two plots suggest that the time series is stationary, but we must preform the ADF and KPSS tests to validate our conclusions.

— Augmented Dickey–Fuller Test (ADF)

Null Hypothesis: Unit root is present: time series is not stationary.

Alternative Hypothesis: unit root is not present: time series is stationary.

In the ‘Table 6’ we can observe that p-value is zero indicating that we can reject the Null hypothesis with more than 95% confidence interval and hence time series is stationary.

— Kwiatkowski–Phillips–Schmidt–Shin Test (KPSS)

Null Hypothesis: Time series is stationary.

Alternative Hypothesis: Time series is not stationary

In the ‘Table 7’, we can see that we have a low p-value of the test Statistics. It means that we can reject the Null hypothesis with confidence concluding that time series is non-stationary.

```
*==========================*
ADF test for SMPEP2:

ADF Statistic: -9.269787
p-value: 0.000000
Critical Values:
    1%: -3.431
    5%: -2.862
    10%: -2.567
None
*==========================*
```

Table 6: ADF test of clean data

```
*==========================*
KPSS test for SMPEP2:

Results of KPSS Test:
Test Statistic            0.732788
p-value                   0.010565
Lags Used                97.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64
None
*==========================*
```

Table 7: KPSS test of clean data

Considering all of the tests above we can conclude that all of the evidence, points out that dataset is stationary. The test and train data were subjected to the same methodology, yielding identical results for the ADF and KPSS tests as well as identical shapes for the Rolling Mean and Rolling Variance plots.

TIME SERIES DECOMPOSITION

— Seasonally Adjusted Plot

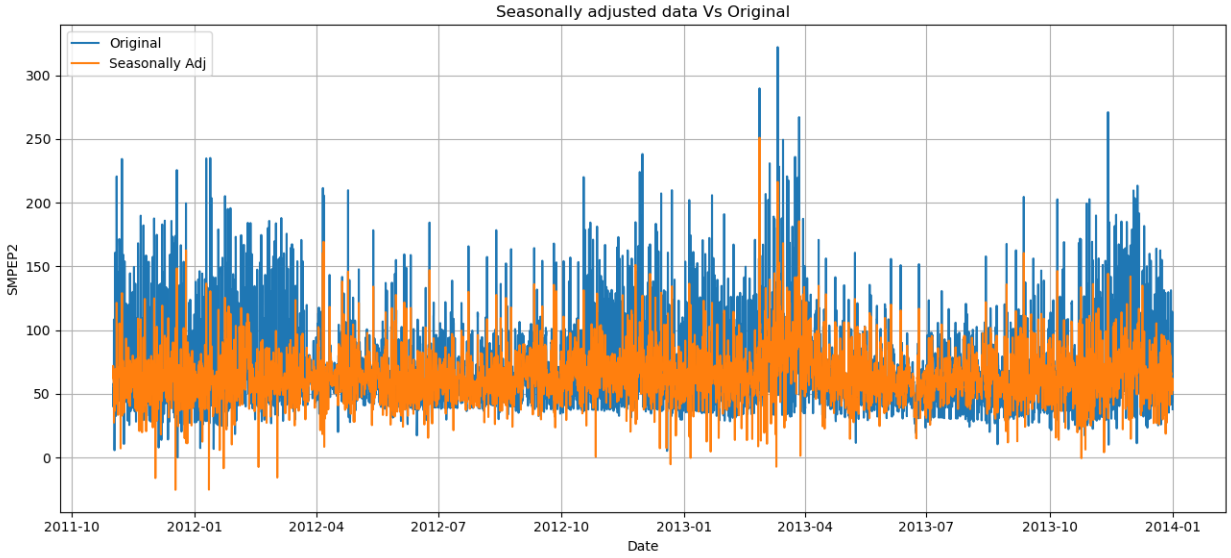
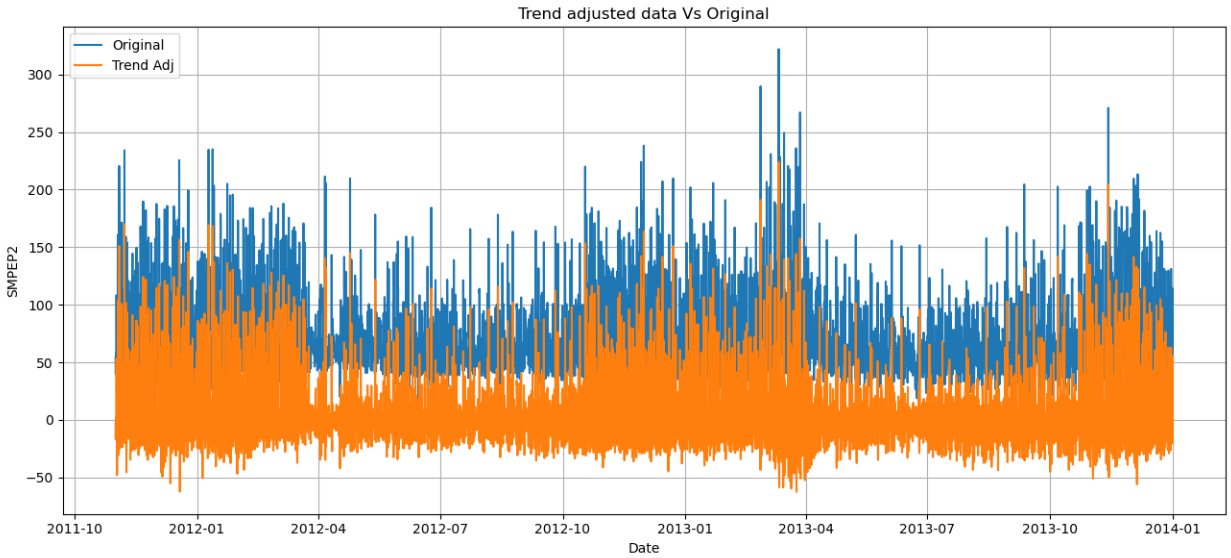


Figure 5: Seasonally Adjusted Plot Vs Original

— Trend Adjusted Plot

Figure 6: Trend Adjusted Plot Vs Original



‘Figure 5' of 'Seasonally Adjusted Plot Vs Original’ indicates that the time series have strong seasonality since there is a significant difference when the seasonal component of the time series is removed.

In the ‘Figure 6’, the only difference between the detrended and original plots is the amplitude of the time series, which remains unchanged. The fact that everything else appears to be comparable suggests that there is little trend component in the time series.

— Strength of Trend and Seasonality

```
The strength of trend for this data set is 40.13%
The strength of seasonality for this data set is 76.71%
```

Table 8: Strength of Trend and Seasonality

Table 8 of ‘Strength of Trend and Seasonality’ proved our initial assumption that was based on ‘Figure 5’ and 'Figure 6’. The strength of Seasonality is around 77% where as the strength of Trend is 40%.

The following is a methodology to gauge the strength of Seasonality:

$$F_S = \max\{0, 1 - (Var(Rt) / Var(St + Rt))\}$$

Formula 1: Strength of Seasonality

The following is a methodology to gauge the strength of Trend:

$$F_T = \max\{0,1 - (Var(Rt) / Var(Tt + Rt))\}$$

Formula 2: Strength of Trend

HOLT-WINTERS METHOD

After applying the Holt-Winters technique package on the train set, the model was utilized to make predictions on the test set. The following outcomes were attained:

— Holt-Winters (Additive Model):

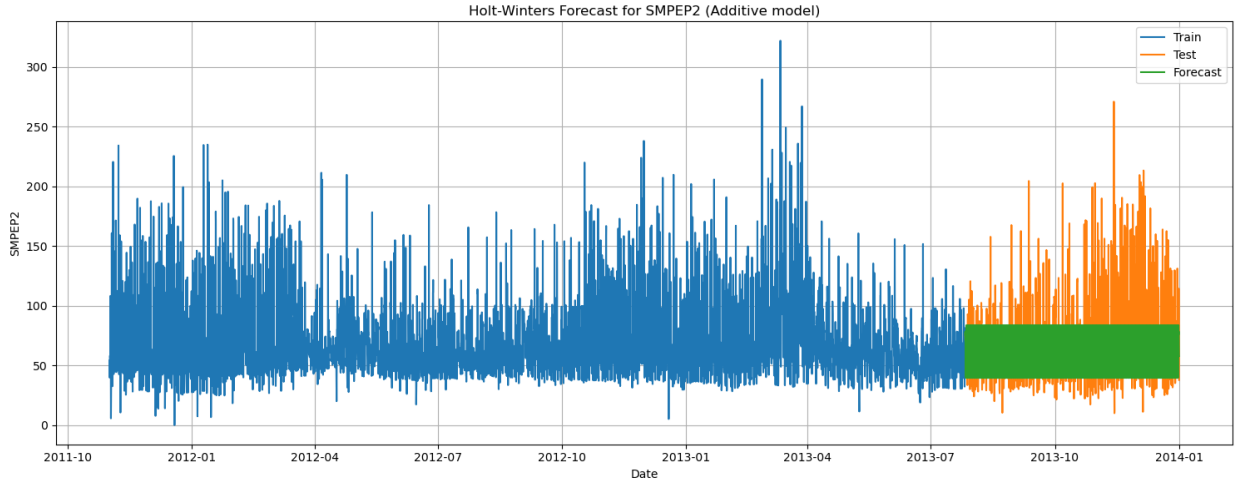


Figure 7: Holt-Winters (Additive Model)

— Holt-Winters (Multiplicative Model):

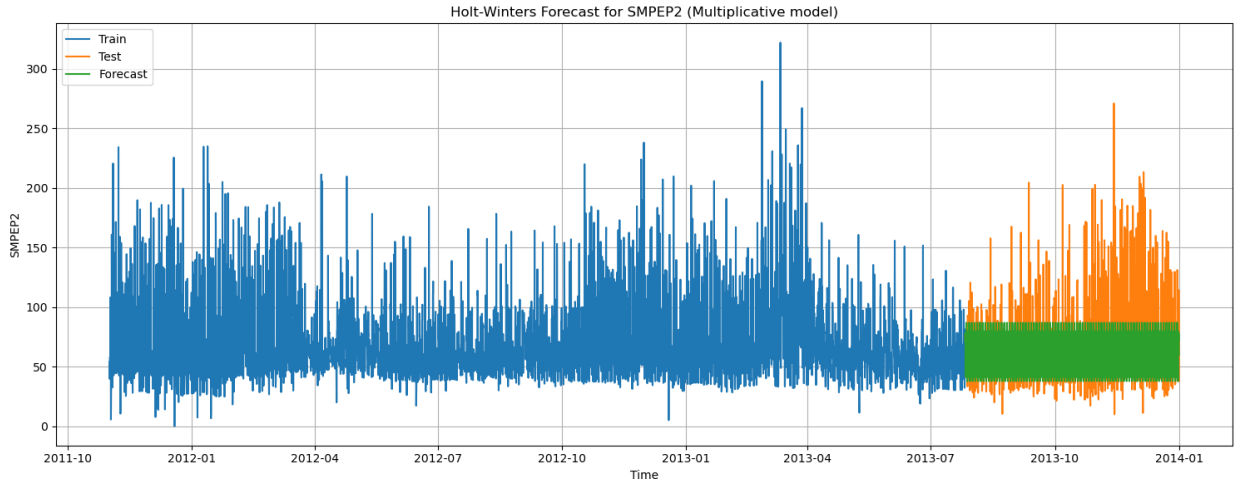


Figure 8: Holt-Winters (Multiplicative Model)

— Holt-Winters (Additive Model) zoomed:

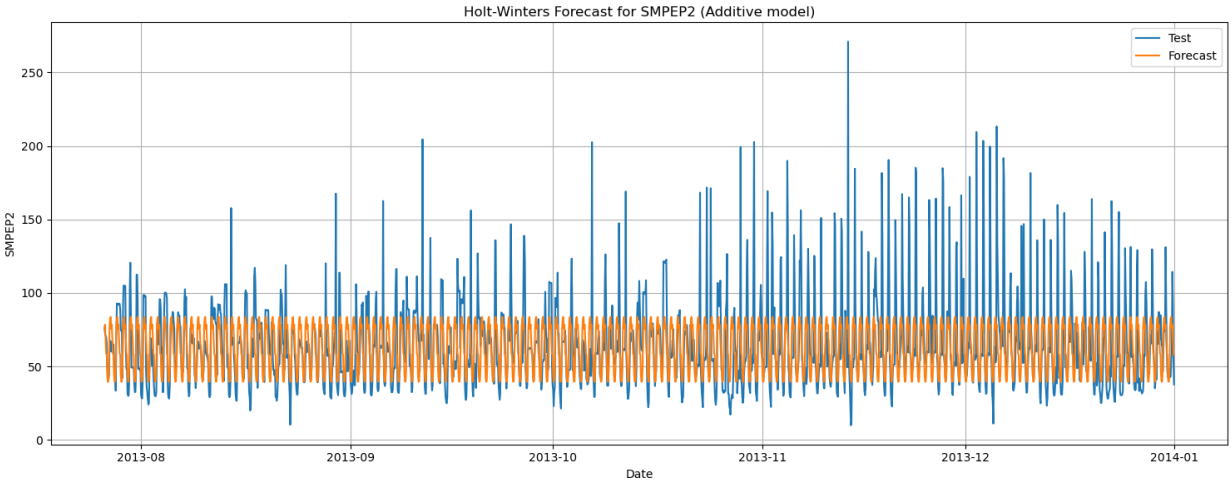


Figure 7.1: Holt-Winters (Additive Model)

— Holt-Winters (Multiplicative Model) zoomed:

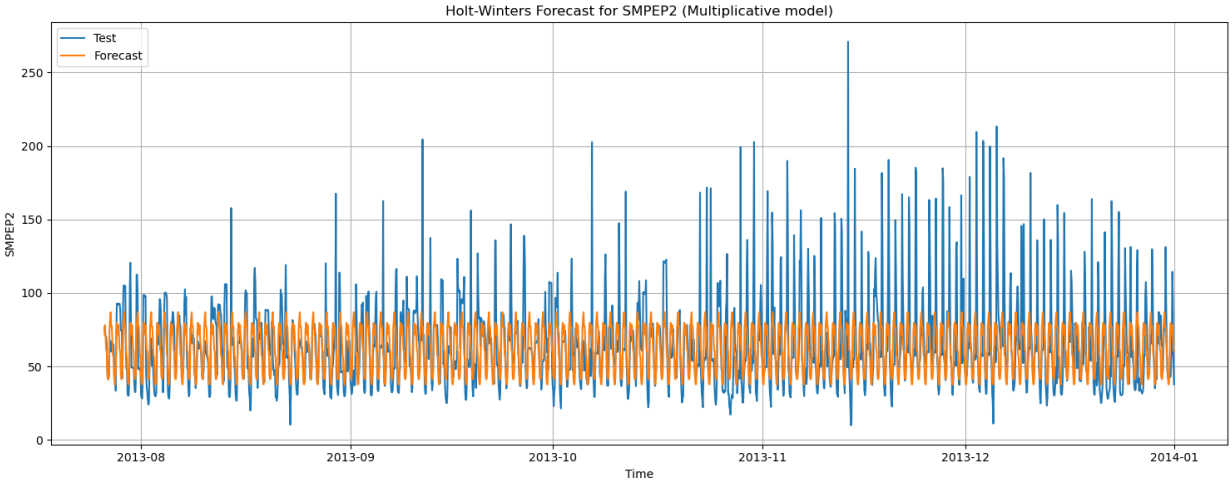


Figure 8.1: Holt-Winters (Additive Model)

By zooming in, we see in ‘Figure 8’ of Holt-Winters (Multiplicative Model) performance is much better compared to ‘Figure 7’. But Bothe models fail to capture the seasonal patterns very well. The following performance metrics were derived from Holt-Winters ‘Additive’ and ‘Multiplicative’ models

- Additive model:
  - MSE add: 704.709
  - RMSE add: 26.546
  - Variance add: 690.82
- Multiplicative model:
  - MSE mul: 695.395
  - RMSE mul: 26.37
  - Variance mul: 690.82

Holt-Winters (Multiplicative Model) does a decent job compared to Additive model. Although overall Holt-Winters model's prediction performance is not great.

FEATURE SELECTION/ELIMINATION AND MULTIPLE LINEAR REGRESSION

Checking Collinearity:

```
Singular values:
 [178.07237252 150.59094937 100.06650868  97.57071351  68.51100486
  63.64551115  48.49683316  19.93113019  18.86377984  11.90386386]
```

Table 9: Singular values

Condition number is equal to 14.959. The output value of condition number is acceptable and might not severely affect the model's stability or coefficient estimates

From the 'Table 9' singular values output shows us huge some numbers of selected features. This disparity in magnitude suggests that the first few variables or components might be highly influential or capture a substantial amount of variance in the data compared to the rest. The decreasing pattern of the singular values suggests decreasing importance or strength. It seems there are potentially 8 significant dimensions or directions that capture most of the variance in dataset as the last value is close to 0.

— Backward Stepwise Regression

	Adj. R^2	AIC	BIC	MSE	F-statistic
1st iteration	0.592	1.46E+04	1.467E+04	0.475	1225
2nd iteration – ORKWindspeed	0.592	1.46E+04	1.466E+04	0.476	1378
3rd iteration – ORKTemperature	0.592	1.46E+04	1.466E+04	0.475	1574
4rth iteration – CO2Intensity	0.591	1.46E+04	1.465E+04	0.476	1835

Table 10: OLS summary

Table 10 represents values discovered during Backward Stepwise Regression model performance. After eliminating features in the order mentioned we can take a look at final version of a model in ‘Table 10’ below.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 SMPEP2   R-squared:                       0.592
Model:                            OLS   Adj. R-squared:                  0.591
Method:                 Least Squares   F-statistic:                     1835.
Date:                Mon, 11 Dec 2023   Prob (F-statistic):               0.00
Time:                        11:05:42   Log-Likelihood:                -7295.4
No. Observations:                7603   AIC:                         1.460e+04
Df Residuals:                    7596   BIC:                         1.465e+04
Df Model:                           6
Covariance Type:            nonrobust
============================================================================================
                               coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------
const                       -0.0121      0.007     -1.631      0.103      -0.027       0.002
ForecastWindProduction       0.1765      0.024      7.241      0.000       0.129       0.224
SystemLoadEA                -0.2513      0.041     -6.134      0.000      -0.332      -0.171
SMPEA                        0.6003      0.009     63.282      0.000       0.582       0.619
ActualWindProduction        -0.2166      0.025     -8.676      0.000      -0.266      -0.168
SystemLoadEP2                0.4699      0.040     11.806      0.000       0.392       0.548
HolidayFlag                  0.1373      0.037      3.706      0.000       0.065       0.210
==============================================================================
Omnibus:                     3467.856   Durbin-Watson:                   1.391
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            47824.162
Skew:                           1.821   Prob(JB):                         0.00
Kurtosis:                      14.735   Cond. No.                         12.6
==============================================================================
```

Table 11: OLS summary

Compared to other iterations final model it is not that different compared to previous steps. But its over all performance is much better as it has the highest ‘F-statistics’ with all of the features being significant (p-value < 0.05).

```
Singular values:
 [138.90478535 120.04260125  87.2713946   59.3196905   18.6031538
  17.12734032  11.05456218]
```

Table 12: Singular values

Although if we refer to the ‘Table 12’ we can notice that compared to its original values there are still some highly correlated features. This is expected outcome, as features described in ‘Table 11’ are similar in the context that i.e. ‘SystemLoad’ is predicted value at that time where as ‘SystemLoad2’ is actual value at that time, same goes for ‘ForecastWindProduction’ & ‘ActualWindProduction’ and ‘SMPEA’ which is the forecasted price at that time.

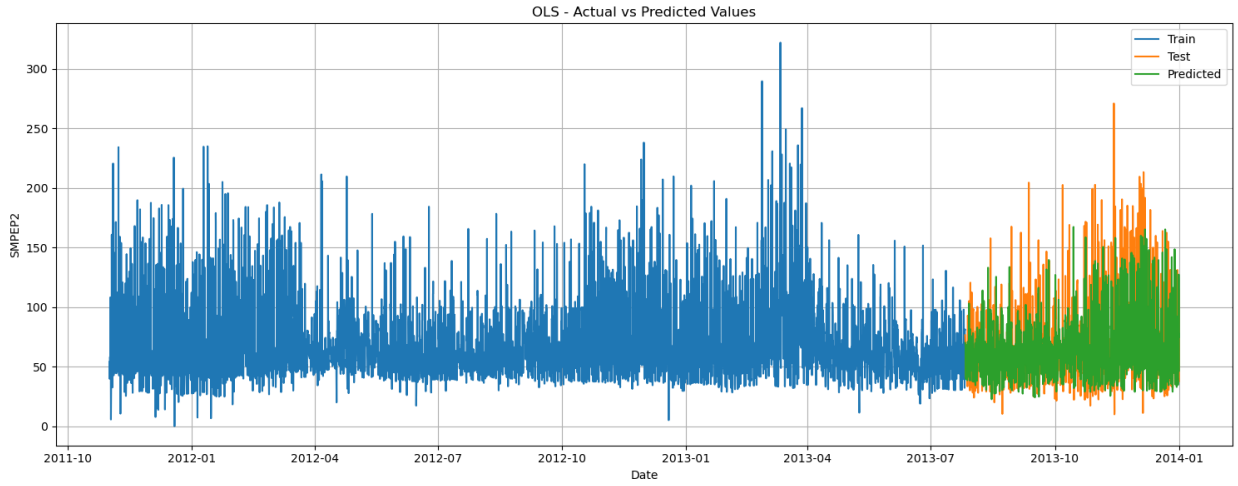


Figure 9: OLS - Predicted Vs Actual

The preceding plot shows that the predicted values almost coincide with the test set even though it have high variance in the data, this indicated a decent performance of the OLS model on the test set.

```
Residual MSE OLS: 395.079
Residual RMSE OLS: 19.877
Residual Variance: 394.185
```

Table 13: OLS - test data performance metrics

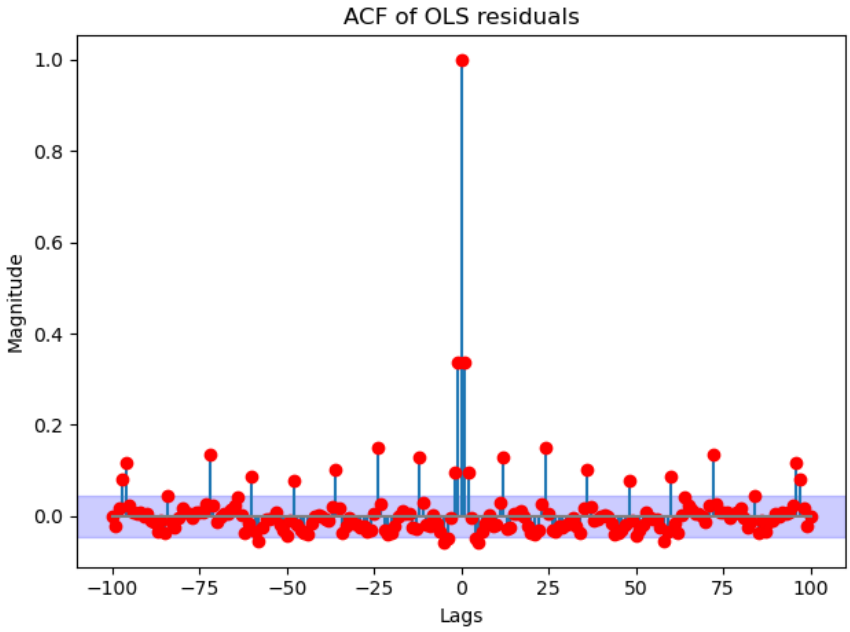


Figure 10: ACF of OLS residuals

‘Table 13’ of performance metrics shows us a good indicators of a model, OLS model preforms very well for the dataset. However, Ljung-Box Q Statistic: 524.403 at lag 100 is very high indicating that OLS residuals are not white. Although, there are some spikes in the ACF of the OLS residuals according to its pattern we may describe them as almost white noise indicating that OLS almost captured model very well. The reason why OLS is performing well on the test set might be the overfitting.

BASE MODELS

We must construct the basic models as benchmarks in order to compare the ARIMA(SARIMA) model.

The following basic models were constructed: 1) Average, 2) Naïve, 3) Drift and 4) Exponential and Simple Smoothing (SES).

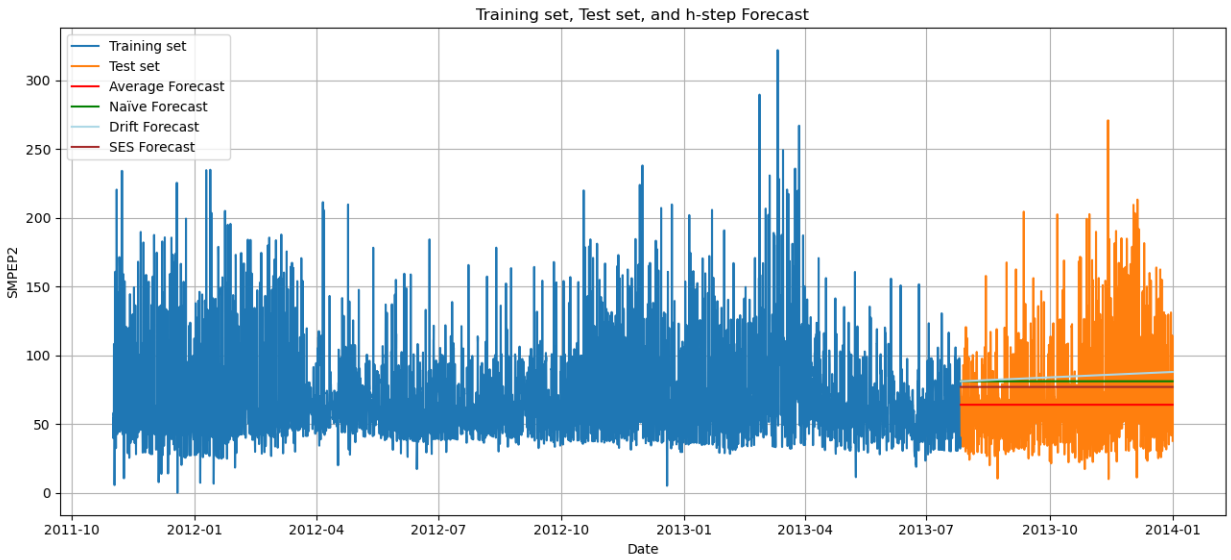


Figure 11: Base models

```
MSE of train data Average 811.9
MSE of test data Average 904.61
RMSE of train data Average: 28.49
RMSE of test data Average: 30.08
Variance of train data Average 808.49
Variance of test data Average 904.5
```

Table 14: Average performance metrics

```
MSE of train data Naïve 822.29
MSE of test data Naïve 1184.64
RMSE of train data Naïve: 28.68
RMSE of test data Naïve: 34.42
Variance of train data Naïve 822.29
Variance of test data Naïve 904.51
```

Table 15: Naïve performance metrics

```
MSE of train data Drift 823.61
MSE of test data Drift 1309.72
RMSE of train data Drift: 36.19
RMSE of test data Drift: 36.19
Variance of train data Drift 823.61
Variance of test data Drift 904.65
```

Table 16: Drift performance metrics

```
MSE of train data SES 782.7
MSE of test data SES 1063.58
RMSE of train data SES: 36.19
RMSE of test data SES: 36.19
Variance of train data SES 782.7
Variance of test data SES 904.5
```

Table 17: SES performance metrics

From the ‘Table 14’ till ‘Table 17’ above, we can see that all the base models are not able to perform very well on the test set and their performance metrics are almost like each other even in train set.

ARIMA(SARIMA) MODEL ORDER DETERMINATION

After feeding the raw data in to the GPAC table, there was no observable order patterns. That’s why seasonal differencing was preformed on data to get rid of seasonality and find new patterns.

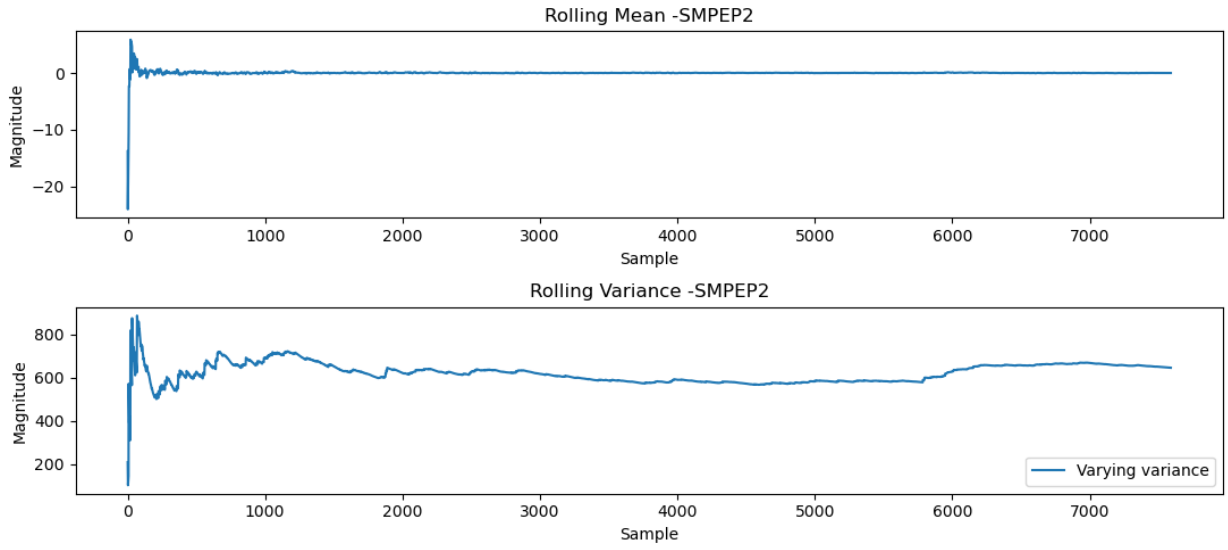


Figure 12: Rolling mean & var of seasonally differenced data

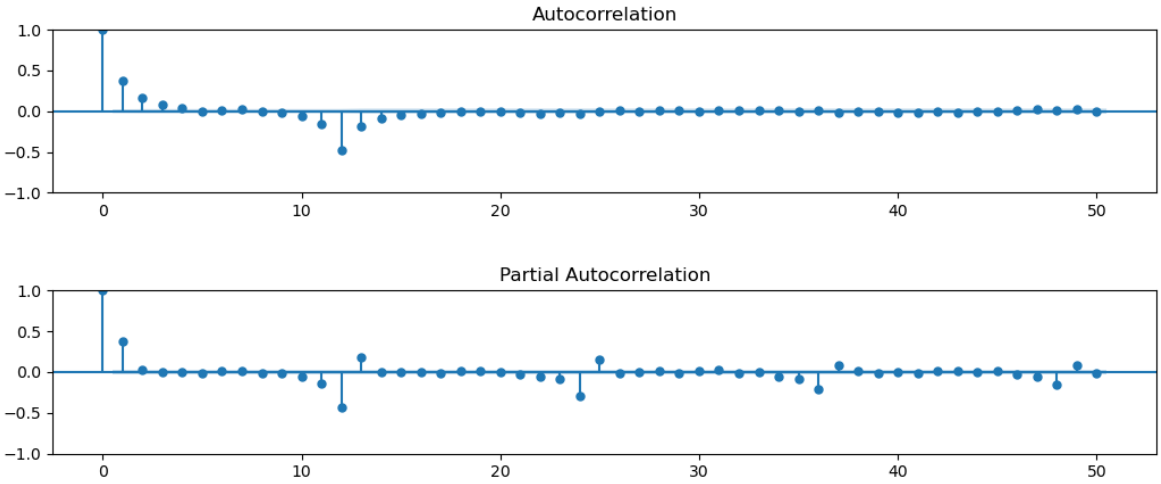


Figure 13: ACF/PACF of seasonally differenced data

We perform tests to see if seasonally differenced data is stationary. ‘Figure 12’ shows that differenced data is stationary. ADF and KPSS tests have similar results as of seasonally differenced data as ‘Table 6’ and ‘Table 7’.

ACF/PACF now shows us that we have order of non-seasonal AR(1).

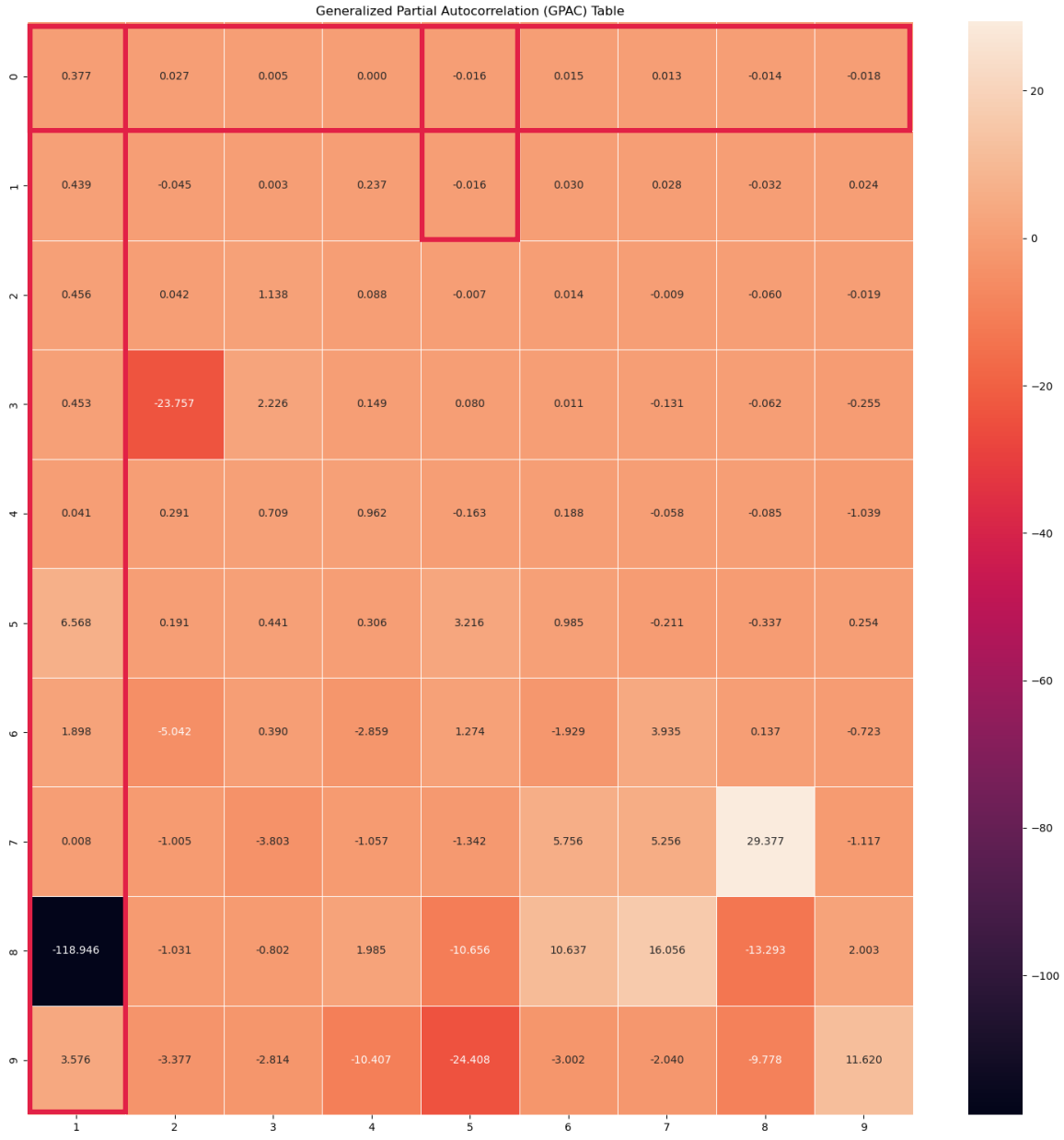


Figure 14: GPAC of seasonally differenced data

‘Figure 14’ represents a GPAC table of seasonal differenced data. From that table we can indicate possible orders of seasonal AR(1), AR(5) and MA(0), MA(1).

1. ARIMA(1,0,0)xARIMA(1,1,0)$_{12}$

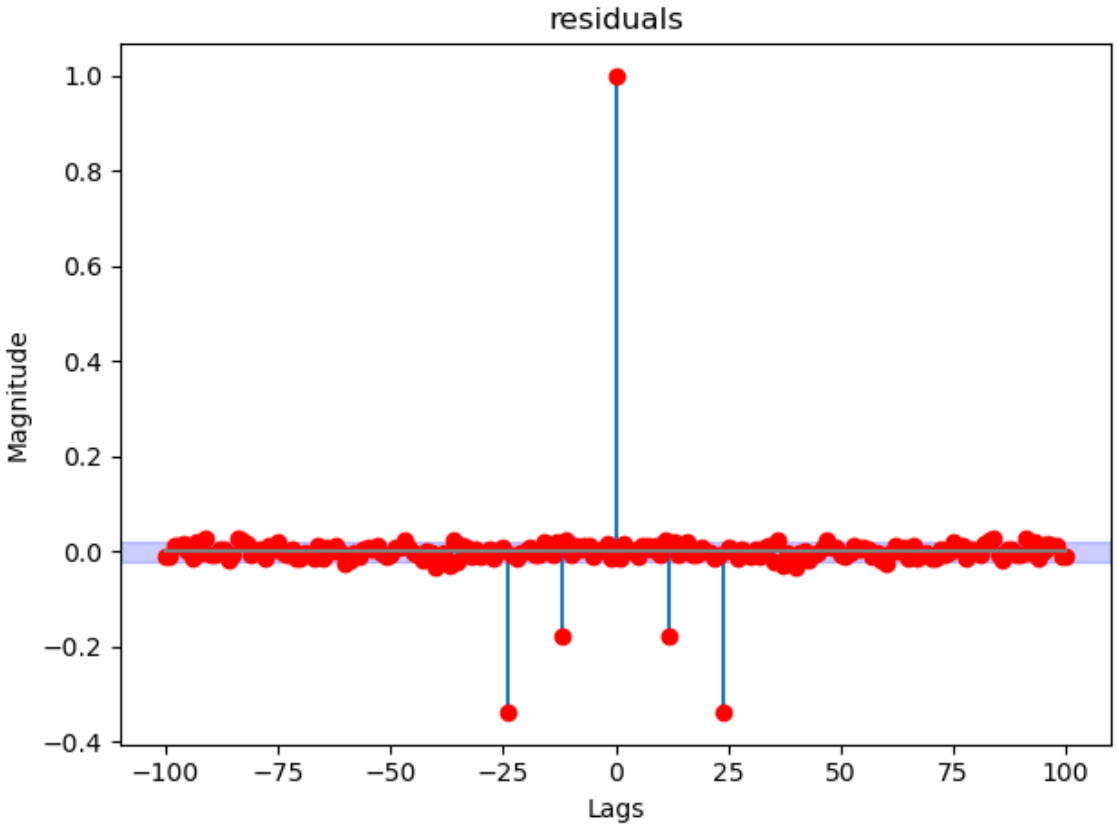


Figure 15: ACF - ARIMA(1,0,0)xARIMA(1,1,0)$_{12}$

```
Train dataset
Residual Variance: 364.482
Residual Mean: 0.034
Residual MSE: 364.483

Test dataset
Residual Variance: 480.874
Residual Mean: 2.01
Residual MSE: 484.914
        lb_stat      lb_pvalue
80  1198.024202  7.797178e-199
Ljung-Box Q Statistic: 1198.024
Chi critical: 132.309
The residual is NOT white
```

Table 18: Performance — ARIMA(1,0,0)xARIMA(1,1,0)$_{12}$

2. ARIMA(1,0,0)xARIMA(5,1,1)$_{12}$

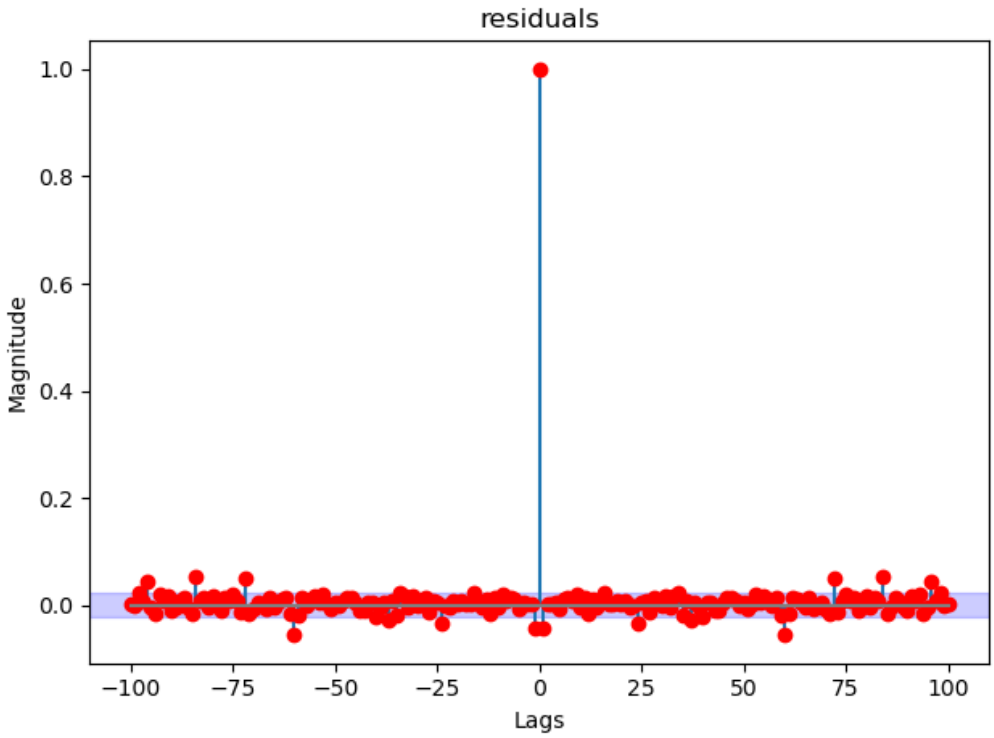


Figure 16: ACF — ARIMA(1,0,0)xARIMA(5,1,1)$_{12}$

```
Train dataset
Residual Variance: 266.27
Residual Mean: 0.039
Residual MSE: 266.272

Test dataset
Residual Variance: 410.296
Residual Mean: 0.149
Residual MSE: 410.318
       lb_stat  lb_pvalue
80  144.655671   0.000013
Ljung-Box Q Statistic: 144.656
Chi critical: 132.309
The residual is NOT white
```

Table 19: Performance — ARIMA(1,0,0)xARIMA(5,1,1)$_{12}$

3. ARIMA(1,0,0)xARIMA(6,1,1)$_{12}$

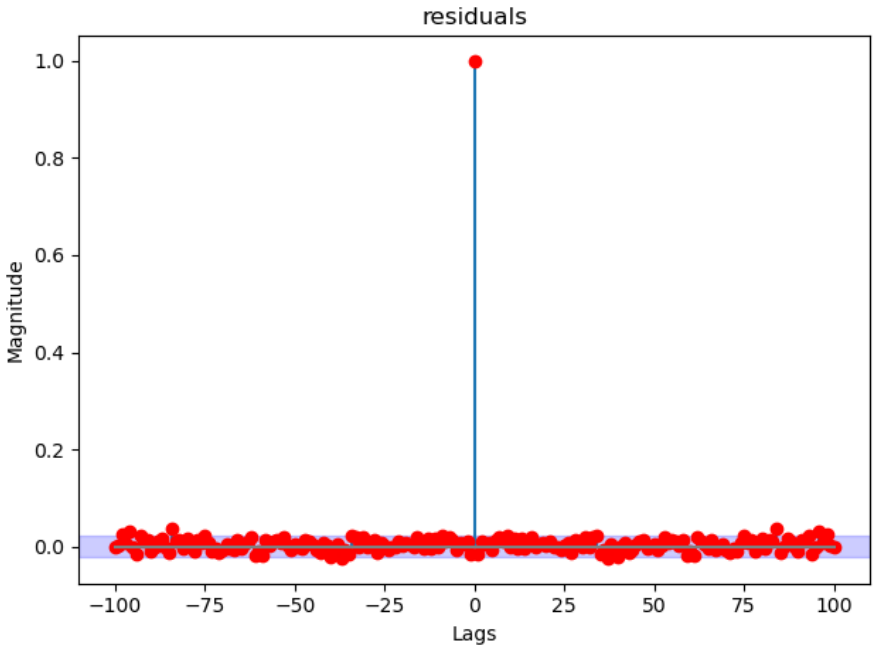


Figure 17: ACF — ARIMA(1,0,0)xARIMA(6,1,1)$_{12}$

```
Train dataset
Residual Variance: 261.567
Residual Mean: 0.029
Residual MSE: 261.568

Test dataset
Residual Variance: 413.232
Residual Mean: 0.582
Residual MSE: 413.571
    lb_stat  lb_pvalue
80  82.4275   0.404155
Ljung-Box Q Statistic: 82.427
Chi critical: 132.309
The residual is white
```

Table 20: Performance — ARIMA(1,0,0)xARIMA(6,1,1)$_{12}$

Although we found that the most optimal oder for seasonal AR(5) from GPAC table its residuals at lag 80 are not white. But if we will increate seasonal order of AR to AR(6) model's performs does increases slightly but its residuals at lag 80 are white, making ARIMA(1,0,0)xARIMA(6,1,1)$_{12}$ the best model we could get, which also could be seen in 'Figure 18'.

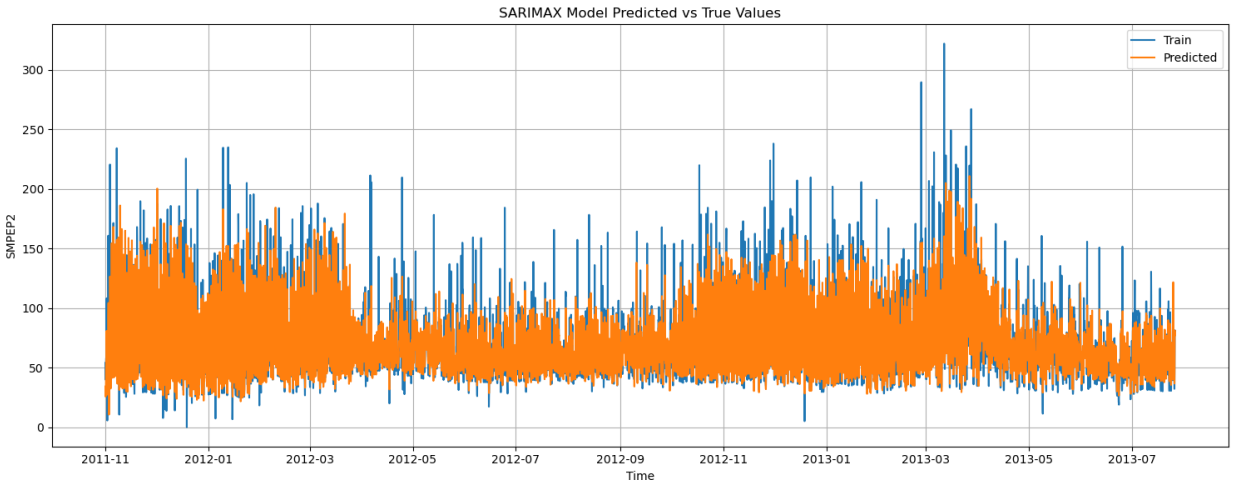


Figure 18: 1-step prediction — ARIMA(1,0,0)xARIMA(6,1,1)$_{12}$

DIAGNOSTIC ANALYSIS

To estimate 'SARIMAX' model parameters we utilized from statsmodels liabrary sm.tsa.SARIMAX() function. The following table shows us estimated coefficients for 'SARIMA' model that utilizes exogenous variables for better predictions.

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.621e-06	1.11e+04	1.46e-10	1.000	-2.17e+04	2.17e+04
HolidayFlag	1.1679	1.364	0.856	0.392	-1.505	3.841
ForecastWindProduction	0.0109	0.002	4.938	0.000	0.007	0.015
SystemLoadEA	-0.0087	0.002	-4.629	0.000	-0.012	-0.005
SMPEA	0.3946	0.008	50.907	0.000	0.379	0.410
ORKTemperature	-0.1656	0.115	-1.446	0.148	-0.390	0.059
ORKWindspeed	-0.0012	0.045	-0.027	0.978	-0.089	0.087
CO2Intensity	-0.0116	0.007	-1.707	0.088	-0.025	0.002
ActualWindProduction	-0.0183	0.003	-6.922	0.000	-0.024	-0.013
SystemLoadEP2	0.0180	0.002	10.035	0.000	0.014	0.022
ar.L1	0.3368	0.006	56.131	0.000	0.325	0.349
ar.S.L12	0.0009	0.009	0.094	0.925	-0.017	0.019
ar.S.L24	0.0114	0.009	1.214	0.225	-0.007	0.030
ar.S.L36	0.0410	0.010	4.260	0.000	0.022	0.060
ar.S.L48	0.0367	0.009	3.921	0.000	0.018	0.055
ar.S.L60	0.0222	0.009	2.366	0.018	0.004	0.041
ar.S.L72	0.0392	0.010	4.050	0.000	0.020	0.058
ma.S.L12	-0.9345	0.005	-192.426	0.000	-0.944	-0.925
sigma2	268.5718	1.571	170.922	0.000	265.492	271.652

Table 21: Coefficients — ARIMA(1,0,0)xARIMA(6,1,1)$_{12}$

From the ‘Table 21’ we can see that a2 and a3 are insignificant as their confidence interval contains 0, which is respectively proven but high p-value as well. More over, features ‘ORKTemperature’, ‘ORKWindspeed’ and ‘CO2Intensity’ are not significant, hence there might be a weak or negligible linear relationship between the exogenous variables and the target variable in the same table.

Since the ACF of residuals from ‘Figure 18’ suggest that residuals are white at lag 80, it means that the estimator is unbiased as it has extracted almost all the information and is generalizing well for the dataset. Chi-Square Test, Variance of Residual and

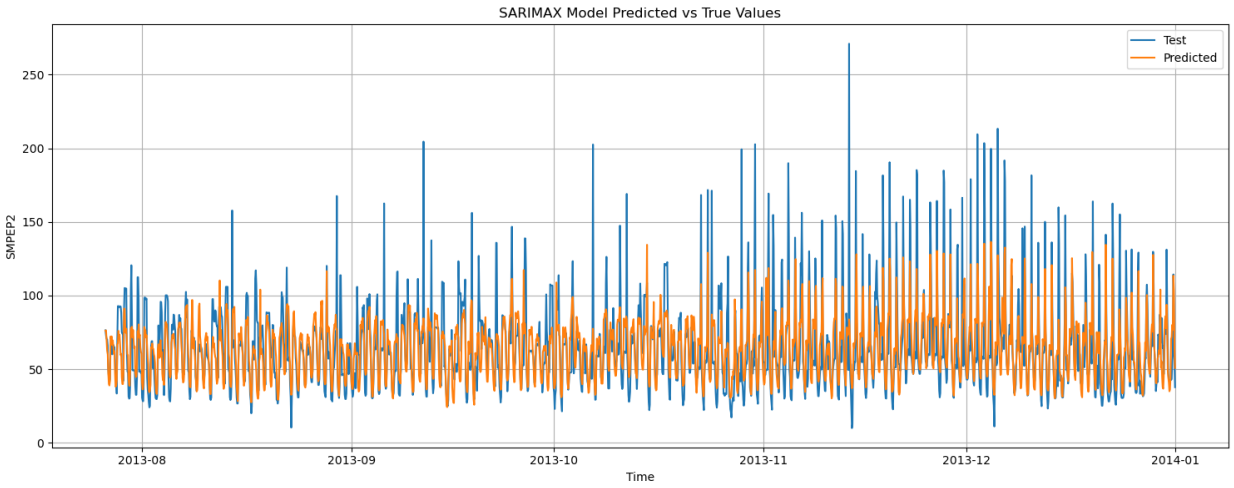


Figure 19: Performance of SARIMA on Test Set (Zoomed)

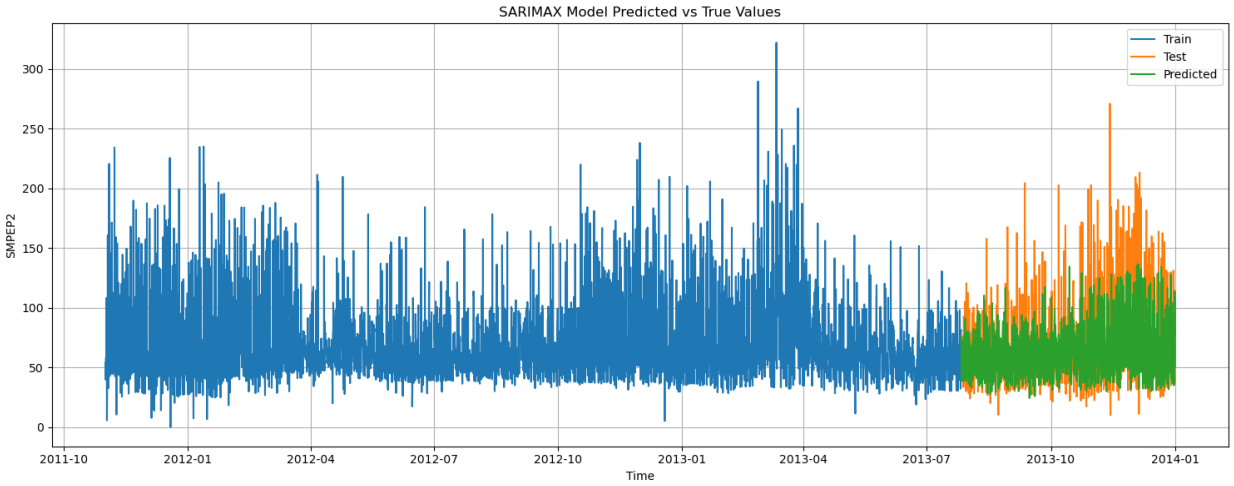


Figure 20: Performance of SARIMA on Test Set

Forecast Error can be seen in ‘Table 20’, in the same table we can also observe models performance. In the following figures below we can observe in performance visually

FINAL MODEL SELECTION

The SARIMAX model has the lowest total MSE, RMSE, MAE, and Q values on the train data, according to all the metrics for all the models listed above in the report, refer to the ‘Tables: 14, 15, 16, 17 and 20’. Although its performance is lacking compared to OLS on the test data. This difference in performance might be due to the overfitting in and high variance in the entire dataset.

FORECAST FUNCTION

Because we have seasonal and non-seasonal parameters we got the following Multiplicative model:

(1-0.3368q)(1-0.0410q^36)(1-0.0367q^48)(1-0.0222q^60)(1-0.0392q^72)(1-0.0392q^72)∇^0∇(12)^1 y(t) = (1+0.9345q^12)e(t)

Custom forecast function of Multiplicative model will also generalize very well on the test set. Although dues to the computational limitations it might take a while.

SUMMARY AND CONCLUSION

The data set is limited on preciseness of what regions its refers to and what type of electricity prices are. As there different price rates i.e ‘Commercial’ rates, ‘Transportation’ rates, ‘Residential rates’ and others. We only assume that its locations Ireland based on the description of ‘ORKWindspeed’ and ‘ORKTemperature’. Even though models performances OLS and SARIMA are not great they are doing a decent job with given information. By increasing the scope of independent variables we can possibly increase models performance as well as predicting power. We also have to consider high variance in prices of electricity that is affected positively by such event like holiday that could be seen in ‘Tables 11 and 21’, political situation, weather conditions, Fuel prices, Regulations. Because of high seasonality SARIMA model was utilized but its computational demanding on computers CPU.

REFERENCES

- statsmodels.regression.linear_model.OLS - statsmodels 0.15.0 (+109). (n.d.). https://www.statsmodels.org/devel/generated/statsmodels.regression.linear_model.OLS.html

- statsmodels.tsa.holtwinters.ExponentialSmoothing - statsmodels 0.15.0 (+109). (n.d.). https://www.statsmodels.org/devel/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.html

- Forecasting Principles and Practice, 3rd Edition Author(s): Rob J Hyndman George Athanasopoulos ISBN-13: 978-0987507136

- statsmodels.tsa.SARIMAX- statsmodels 0.15.0 (+109). (n.d.). https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html

APPENDIX

All the utilized function such as ACF/PACF, Rolling mean and Rolling Variance, GPAC table and others were generated in this courses Labs and Home works.