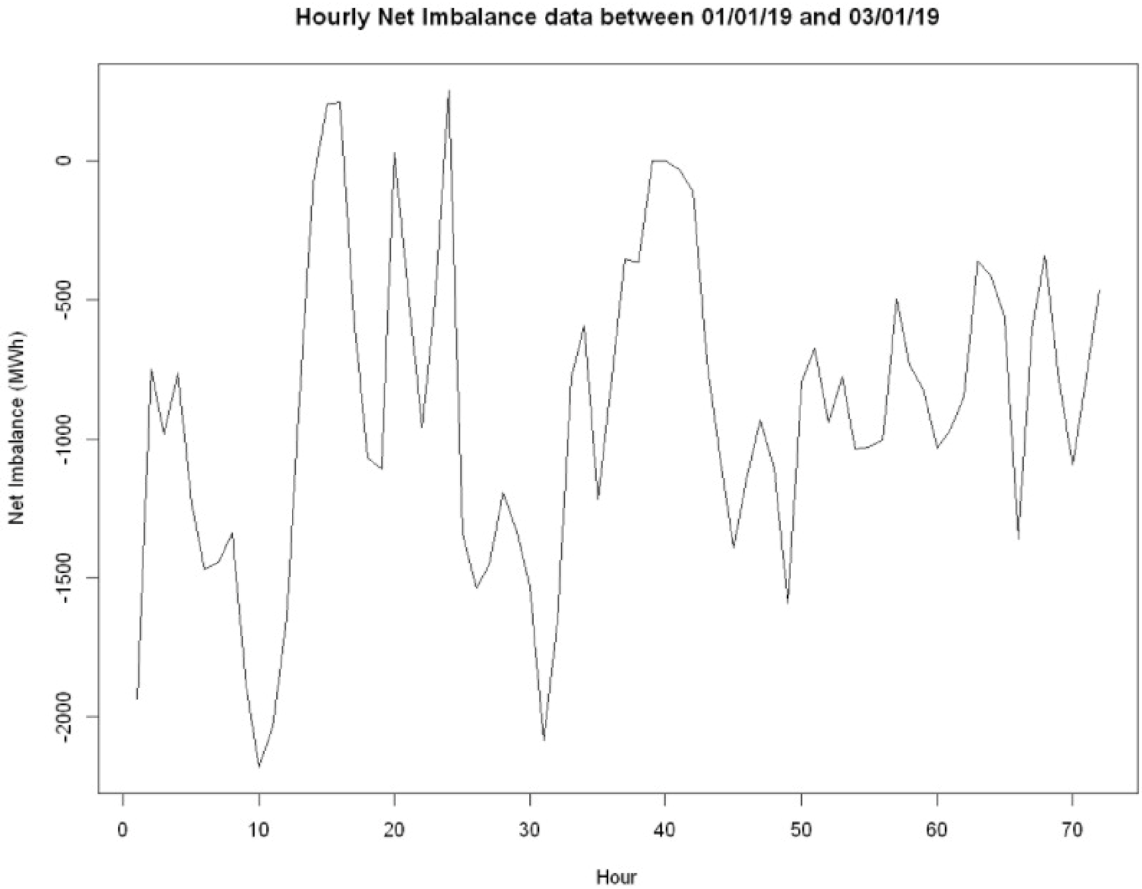
Faculty of Engineering
Department of Industrial Engineering
Fall 2021
IE 48B – Project Report
GROUP 8
2017402177
2017402198
2017402264

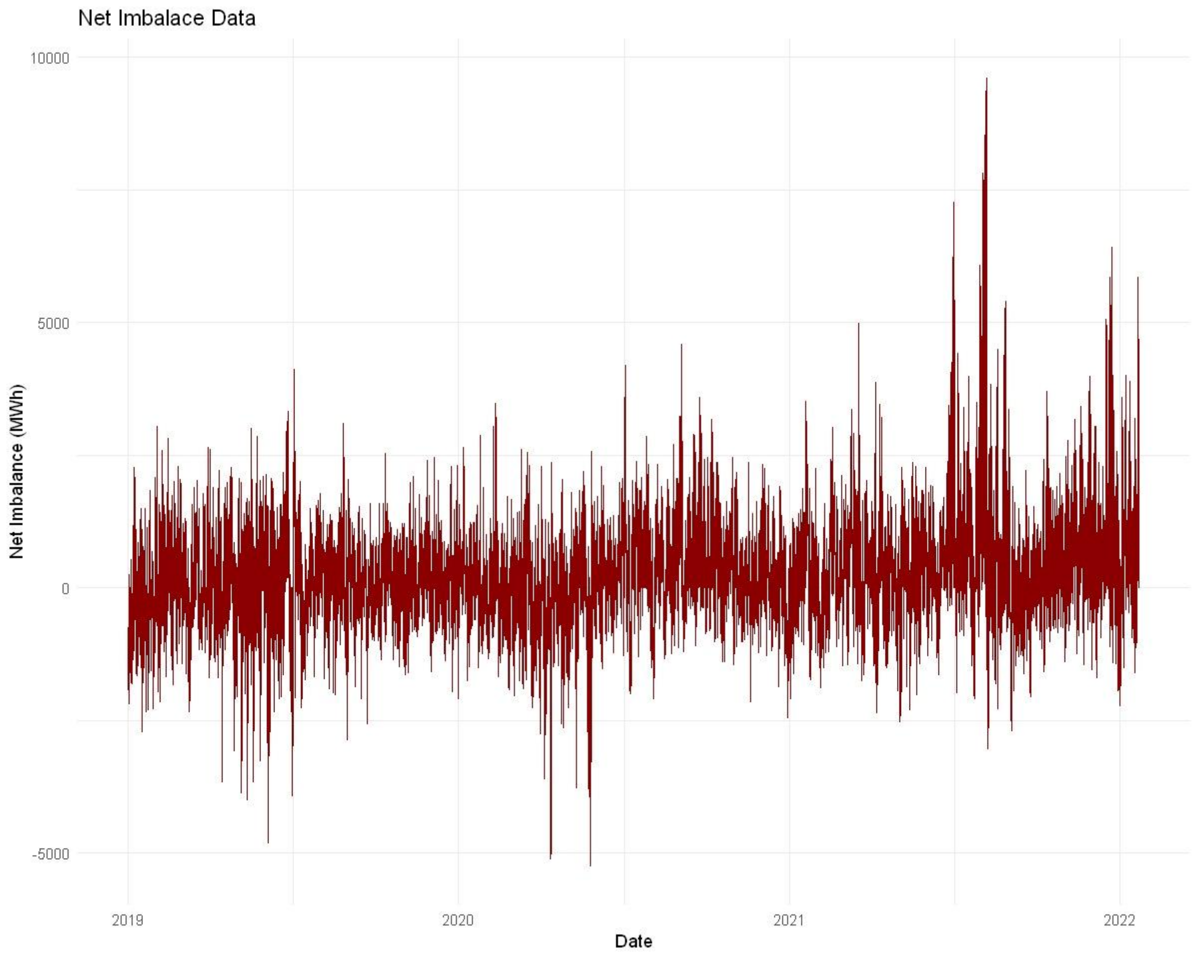
1 Introduction

In this project, the main task is to develop an approach for predicting the sign of imbalance for Turkish electricity market (MWh) from 9 January 2022 to 21 January 2022. Every day, the predictions consist of the intraday sign of imbalance from 12th hour to 23rd hour. Up to previous day's data will be used for predicting the present day's sign of imbalance.

In addition to the past net imbalance data, the hourly weather information data belonging to seven different locations close to the big cities in Turkey are also provided. These cities are Antalya, Adana, Konya, İzmir, Eskişehir, Ankara and İstanbul respectively. Weather information contains downward shortwave radiation at the surface, total cloud cover at the low cloud layer, temperature at 2 meters altitude, wind speed at 10 meters altitude, and wind direction in 10 meters altitude, relative humidity at 2 meters altitude. The best point to start is analysing the data visually. Here is the plot containing three consecutive days:



From the plot, we can see that there is a weak seasonality in the data. Some hours have a similar behavior but mostly data seems to have high variance. Also, there can be seen that there is a trend component.



As mentioned above, seasonality is weak that can be seen from the plot of net imbalance data. Opposite to the 3-day plot above, there is no trend in net imbalance data. There are some outliers in summer 2021. It will be analyzed later in the report. To use weather information effectively, mean of the seven big cities' weather information data is used.

2 Related Literature

For the explainable boosted linear regression method. Lecture material is used.

EBLR generates nonlinear features that are also interpretable. Each generated feature corresponds to a set of decision rules. Iteratively updates the forecasts by generating simple binary features through a two-step process based on regression trees. Extended to probabilistic forecasting by utilizing the empirical distribution of the residuals. Competitive forecasting performance without sacrificing simplicity and interpretability.

3 Approach

In this section, one base model will be established with only day type, month regressors, trend factor and hour regressors.

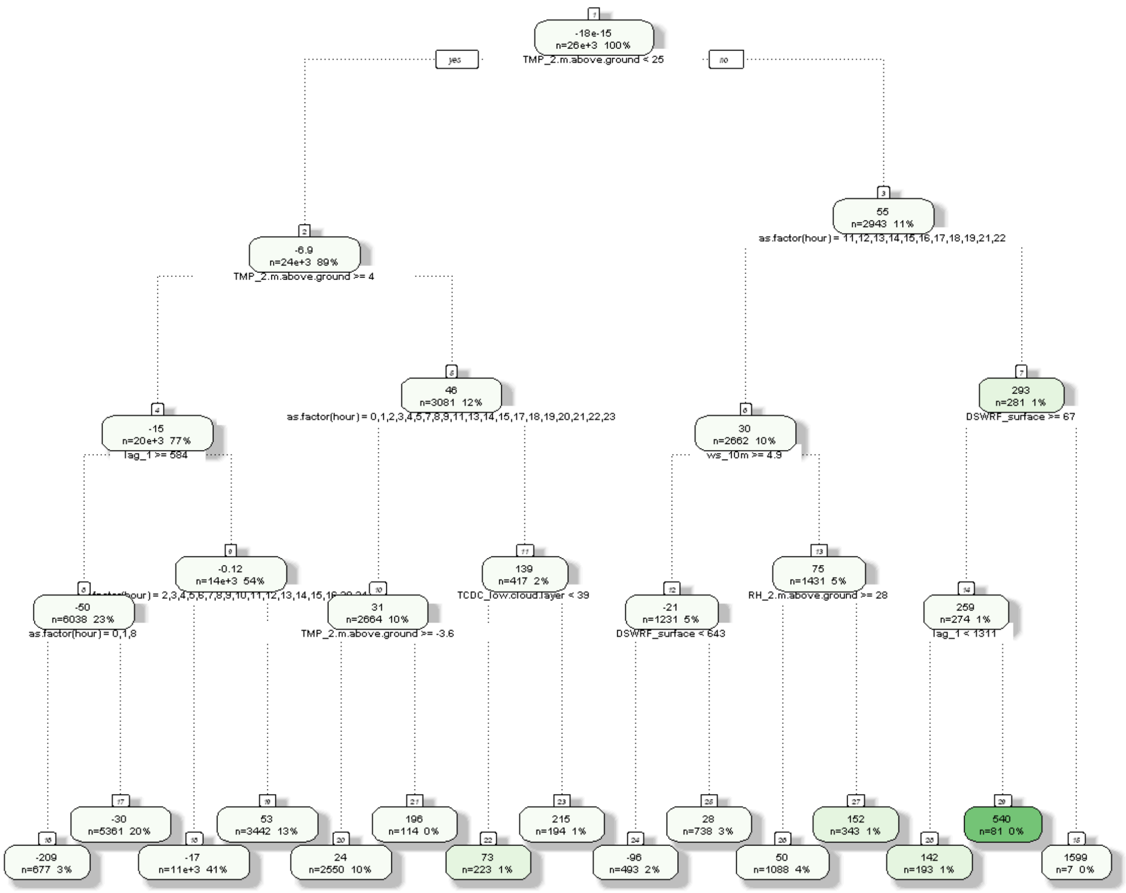
```
Call:
lm(formula = net ~ weekdays + months + years + trend + as.factor(hour),
    data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-5405.8  -564.5   -23.7   516.4  8483.7

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -285.71731   41.14629  -6.944 3.90e-12 ***
weekdaysMonday     -22.40094   22.82068  -0.982 0.326302
weekdaysSaturday  -124.63683   22.78597  -5.470 4.54e-08 ***
weekdaysSunday    -345.70082   22.81924 -15.150  < 2e-16 ***
weekdaysThursday    67.92089   22.78130   2.981 0.002872 **
weekdaysTuesday     41.08477   22.78427   1.803 0.071367 .
weekdaysWednesday   23.49126   22.78271   1.031 0.302503
months2             97.25624   36.72218   2.648 0.008091 **
months3             12.02477   50.83108   0.237 0.812997
months4           -150.76248   69.26475  -2.177 0.029518 *
months5            -94.08395   88.77904  -1.060 0.289266
months6             82.77239  109.06331   0.759 0.447896
months7            297.46725  129.52896   2.297 0.021653 *
months8            289.30371  150.58886   1.921 0.054724 .
months9           -188.10714  171.47423  -1.097 0.272652
months10          -107.19871  192.34946  -0.557 0.577319
months11          -136.32699  213.36378  -0.639 0.522866
months12          -163.16899  234.34707  -0.696 0.486265
years2020         -386.09463  254.82989  -1.515 0.129757
years2021         -505.70156  508.47673  -0.995 0.319968
years2022         -762.84385  757.96996  -1.006 0.314218
trend                0.05395    0.02897   1.862 0.062558 .

 as.factor(hour)1   151.47958   42.21578   3.588 0.000334 ***
 as.factor(hour)2    87.06754   42.21581   2.062 0.039176 *
 as.factor(hour)3    32.88684   42.21586   0.779 0.435977
 as.factor(hour)4   -25.75042   42.21593  -0.610 0.541887
 as.factor(hour)5   -51.31943   42.21602  -1.216 0.224133
 as.factor(hour)6  -157.12496   42.21613  -3.722 0.000198 ***
 as.factor(hour)7  -120.95452   42.21626  -2.865 0.004172 **
 as.factor(hour)8    46.14862   42.21641   1.093 0.274340
 as.factor(hour)9   303.82582   42.21657   7.197 6.33e-13 ***
 as.factor(hour)10  286.56200   42.21676   6.788 1.16e-11 ***
 as.factor(hour)11  422.13202   42.21697   9.999  < 2e-16 ***
 as.factor(hour)12  226.30559   42.21720   5.361 8.37e-08 ***
 as.factor(hour)13  155.70803   42.21745   3.688 0.000226 ***
 as.factor(hour)14  419.50767   42.21772   9.937  < 2e-16 ***
 as.factor(hour)15  409.71742   42.21801   9.705  < 2e-16 ***
 as.factor(hour)16  346.09465   42.21831   8.198 2.56e-16 ***
 as.factor(hour)17  436.00619   42.21864  10.327  < 2e-16 ***
 as.factor(hour)18  417.38499   42.21899   9.886  < 2e-16 ***
 as.factor(hour)19  391.37075   42.21936   9.270  < 2e-16 ***
 as.factor(hour)20  380.94429   42.21974   9.023  < 2e-16 ***
 as.factor(hour)21  406.64401   42.22015   9.632  < 2e-16 ***
 as.factor(hour)22  288.23358   42.22058   6.827 8.87e-12 ***
 as.factor(hour)23  211.79514   42.22103   5.016 5.30e-07 ***
 ---
 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Residual standard error: 991.8 on 26451 degrees of freedom
 Multiple R-squared:  0.1112,    Adjusted R-squared:  0.1097
 F-statistic: 75.19 on 44 and 26451 DF,  p-value: < 2.2e-16
```

Hourly regressors seems to be very effective. Trend and month regressors have a weak performance. Overall, this model is not sufficient for predicting the net imbalance signs. Lag values will be added to this model.

```
Call:
lm(formula = net ~ weekdays + months + years + trend + as.factor(hour) +
    lag_1, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-2263.0  -219.9     2.8   214.5  3274.9

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.209e+02  1.719e+01 -12.847  < 2e-16 ***
weekdaysMonday     6.451e+00  9.531e+00   0.677   0.4985
weekdaysSaturday  -2.609e+00  9.522e+00  -0.274   0.7841
weekdaysSunday    -2.381e+01  9.573e+00  -2.488   0.0129 *
weekdaysThursday   1.584e+01  9.515e+00   1.664   0.0960 .
weekdaysTuesday    1.188e+01  9.516e+00   1.248   0.2121
weekdaysWednesday  1.201e+01  9.515e+00   1.262   0.2070
months2            1.151e+01  1.534e+01   0.750   0.4532
months3            8.368e+00  2.123e+01   0.394   0.6935
months4           -3.143e+00  2.893e+01  -0.109   0.9135
months5            8.161e+00  3.708e+01   0.220   0.8258
months6            2.955e+01  4.555e+01   0.649   0.5165
months7            5.253e+01  5.410e+01   0.971   0.3316
months8            5.481e+01  6.289e+01   0.871   0.3835
months9            1.684e+01  7.161e+01   0.235   0.8141
months10           2.816e+01  8.033e+01   0.351   0.7259
months11           3.178e+01  8.911e+01   0.357   0.7214
months12           3.067e+01  9.787e+01   0.313   0.7540
years2020          1.896e+01  1.064e+02   0.178   0.8586
years2021          6.175e+01  2.124e+02   0.291   0.7712
years2022          1.015e+02  3.166e+02   0.321   0.7486
trend             -1.227e-03  1.210e-02  -0.101   0.9192

as.factor(hour)1   3.426e+02  1.764e+01  19.418  < 2e-16 ***
as.factor(hour)2   1.405e+02  1.763e+01   7.969 1.66e-15 ***
as.factor(hour)3   1.449e+02  1.764e+01   8.215 2.22e-16 ***
as.factor(hour)4   1.355e+02  1.764e+01   7.680 1.64e-14 ***
as.factor(hour)5   1.632e+02  1.764e+01   9.249  < 2e-16 ***
as.factor(hour)6   8.063e+01  1.765e+01   4.569 4.92e-06 ***
as.factor(hour)7   2.129e+02  1.766e+01  12.058  < 2e-16 ***
as.factor(hour)8   3.472e+02  1.766e+01  19.665  < 2e-16 ***
as.factor(hour)9   4.530e+02  1.764e+01  25.683  < 2e-16 ***
as.factor(hour)10  2.016e+02  1.764e+01  11.434  < 2e-16 ***
as.factor(hour)11  3.529e+02  1.764e+01  20.011  < 2e-16 ***
as.factor(hour)12  3.391e+01  1.764e+01   1.922   0.0546 .
as.factor(hour)13  1.413e+02  1.763e+01   8.010 1.20e-15 ***
as.factor(hour)14  4.692e+02  1.764e+01  26.606  < 2e-16 ***
as.factor(hour)15  2.197e+02  1.764e+01  12.454  < 2e-16 ***
as.factor(hour)16  1.650e+02  1.764e+01   9.353  < 2e-16 ***
as.factor(hour)17  3.127e+02  1.764e+01  17.730  < 2e-16 ***
as.factor(hour)18  2.124e+02  1.764e+01  12.039  < 2e-16 ***
as.factor(hour)19  2.033e+02  1.764e+01  11.524  < 2e-16 ***
as.factor(hour)20  2.165e+02  1.764e+01  12.275  < 2e-16 ***
as.factor(hour)21  2.517e+02  1.764e+01  14.269  < 2e-16 ***
as.factor(hour)22  1.100e+02  1.764e+01   6.233 4.64e-10 ***
as.factor(hour)23  1.411e+02  1.764e+01   8.002 1.28e-15 ***
lag_1              9.086e-01  2.568e-03 353.835  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 414.2 on 26449 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.845,     Adjusted R-squared:  0.8447
F-statistic:  3203 on 45 and 26449 DF,  p-value: < 2.2e-16
```

With the addition of the lag value. Performance of our base model is increased significantly in terms of R squared and residual standard error. We will use explainable boosted linear regression with this base model.

Here is a demonstration of one explainable boosted linear regression iteration.



Rattle 2022-Jan-23 16:09:03 a_kok

A decision tree is fitted on the residuals of the base linear regression model with the all the existed features. The path which leads to highest average value is chosen and dummy variables corresponding to that path is created. Those dummy variables are added to the model at every iteration until the improvement decreased to a significantly lower level.

After 5 iterations of explainable boosted linear regression, final model is decided.

```
lag_1                                 ***
tmp_30:trend_22:hour_12_15:lag_1_2229  ***
tmp_30:hour_12_17:trend_15:lag_1_214   ***
tmp_25:hour_11_12:trend_23:hour_10_20  ***
tmp_25:hour_11_12:trend_23:dswrf_118   ***
tmp_25:hour_11_12:lag_1_2766:dswrf_330 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 411.4 on 26756 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.8482,    Adjusted R-squared:  0.8479
F-statistic:  2990 on 50 and 26756 DF,  p-value: < 2.2e-16
```

Outlier Analysis

As mentioned in the previous section, there was some outliers in the in the hourly imbalance data. We decided on a threshold value and created some dummy variables to exclude outlier points.

Here is our final model with the outlier analysis. This model will be used for the prediction purpose.

```
lag_1                                       ***
outlier_small                               ***
outlier_great                               ***
tmp_30:trend_22:hour_12_15:lag_1_2229  ***
tmp_30:hour_12_17:trend_15:lag_1_214   ***
tmp_25:hour_11_12:trend_23:hour_10_20  ***
tmp_25:hour_11_12:trend_23:dswrf_118   ***
tmp_25:hour_11_12:lag_1_2766:dswrf_330 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 282.2 on 26442 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.928,	Adjusted R-squared:  0.9279
F-statistic:  6558 on 52 and 26442 DF,  p-value: < 2.2e-16
```

4 Results

Here are the daily results of our model compared to previous day and previous week naïve forecast.

daily_accuracy	base_daily_accuracy	base_2_daily_accuracy	date
<dbl>	<dbl>	<dbl>	<date>
0.7500000	0.7500000	0.5000000	2022-01-09
0.6666667	0.5000000	0.8333333	2022-01-10
1.0000000	0.6666667	0.5000000	2022-01-11
1.0000000	1.0000000	0.3333333	2022-01-12
0.8333333	0.8333333	0.8333333	2022-01-13
0.2500000	0.4166667	0.3333333	2022-01-14
0.9166667	0.2500000	0.9166667	2022-01-15
0.7500000	0.6666667	0.8333333	2022-01-16
0.4166667	0.1666667	0.5833333	2022-01-17
1.0000000	0.4166667	1.0000000	2022-01-18
1.0000000	1.0000000	1.0000000	2022-01-19
1.0000000	1.0000000	0.8333333	2022-01-20
1.0000000	1.0000000	0.2500000	2022-01-21

As can be seen from the results our model has a much better performance than two others. Overall result is here:

method	base	base_2
<dbl>	<dbl>	<dbl>
0.8141026	0.6666667	0.6730769

5 Conclusion and Future Work

Even though the final model has a decent accuracy, further improvements and extensions can be made in the model to have a better approach.

Firstly, we used lag 1 values in our model but since we don't have the lag 1 values, predictions are used as lag 1 values. Usage of lag 1 may not the best approach.

We had some external variables for 7 big cities. For simplicity the mean of the 7 cities is used but there could be a better approach for the usage of those external variables.

Day, year, trend, and month regressors were not much effective in the final model. It could be possible to get some useful information from those regressors with some representations or extensive analysis.

There were some important outliers in 2021 summer, In final model this outlier period is excluded from model but there could be better analysis for this outlier period.

6 Codes

- The R Markdown of this report is here. (ipynb file)
- The Project description is here. (pdf file)

7 References

Ilic, I., Görgülü, B., Cevik, M., & Baydoğan, M. G. (2021). Explainable boosted linear regression for time series forecasting. Pattern Recognition, 120, 108144.