In [1]:

```
import os
import pandas as pd
import numpy as np

from sklearn.preprocessing import MinMaxScaler
import joblib
import seaborn as sns
sns.set(color_codes=True)
import matplotlib.pyplot as plt
%pylab inline

from numpy.random import seed
seed(1)
from tensorflow import random
random.set_seed(1)
import tensorflow as tf

from keras.layers import Input, Dropout, Dense, LSTM, TimeDistributed, RepeatVector
from keras.models import Model
from keras import regularizers
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:

```
seed(10)
random.set_seed(10)
```

In [3]:

```
data = pd.read_csv('pollutionData158324.csv', header=0, infer_datetime_format=True, parse_d
data.head(5)
```

Out[3]:

timestamp	ozone	particullate_matter	carbon_monoxide	sulfure_dioxide	nitrogen_dioxide	long
2014-08-01 00:05:00	101	94	49	44	87	10.1(
2014-08-01 00:10:00	106	97	48	47	86	10.1(
2014-08-01 00:15:00	107	95	49	42	85	10.1(
2014-08-01 00:20:00	103	90	51	44	87	10.1(
2014-08-01 00:25:00	105	94	49	39	82	10.1(

In [4]:

```
data= data.drop('longitude', axis =1)
data= data.drop('latitude', axis =1)
```

In [5]:

```
data.head(5)
```

Out[5]:

timestamp	ozone	particullate_matter	carbon_monoxide	sulfure_dioxide	nitrogen_dioxide
2014-08-01 00:05:00	101	94	49	44	87
2014-08-01 00:10:00	106	97	48	47	86
2014-08-01 00:15:00	107	95	49	42	85
2014-08-01 00:20:00	103	90	51	44	87
2014-08-01 00:25:00	105	94	49	39	82

In [6]:

```
data.shape
```

Out[6]:

```
(17568, 5)
```

In [7]:

```
train_size = int(len(data) * 0.9)
test_size = len(data) - train_size
data_train, data_test = data.iloc[0:train_size], data.iloc[train_size:len(data)]
print("Training set shape : ", data_train.shape)
print("Testing set shape : ", data_test.shape)
```

```
Training set shape :  (15811, 5)
Testing set shape :  (1757, 5)
```

In [8]:

```
# zamanları frenaks cınsıne donusturduk
data_train_fft= np.fft.fft(data_train)
data_test_fft= np.fft.fft(data_test)
```

In [9]:

```
# normalizasyon

scaler=MinMaxScaler()
X_train = scaler.fit_transform(data_train)
X_test = scaler.transform(data_test)
scaler_filename = "scaler_data"
joblib.dump(scaler, scaler_filename)
```

Out[9]:

```
['scaler_data']
```

In [10]:

```
 # model için shapleri yeniden düzenleme
X_train= X_train.reshape(X_train.shape[0],1,X_train.shape[1])
print("Train data büyüklüğü : ", X_train.shape)
X_test= X_test.reshape(X_test.shape[0],1,X_test.shape[1])
print("Test data büyüklüğü : ", X_test.shape)
```

```
Train data büyüklüğü :  (15811, 1, 5)
Test data büyüklüğü :  (1757, 1, 5)
```

In [11]:

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import SimpleRNN
from keras.layers import Dropout
```

In [12]:

```
# model kurulumu
def RNN_MODEL(X):
    inputs = Input(shape=(X.shape[1], X.shape[2]))
    Regressor1 = SimpleRNN(16, activation='relu', return_sequences=True,
              kernel_regularizer=regularizers.l2(0.00))(inputs)
    Regressor2 = SimpleRNN(5, activation='relu', return_sequences=False)(Regressor1)
    Regressor3 = RepeatVector(X.shape[1])(Regressor2)
    Regressor4 = SimpleRNN(5, activation='relu', return_sequences=True)(Regressor3)
    Regressor5 = SimpleRNN(16, activation='relu', return_sequences=True)(Regressor4)
    output = TimeDistributed(Dense(X.shape[2]))(Regressor5)
    model = Model(inputs=inputs, outputs=output)
    return model
```

In [13]:

```
model = RNN_MODEL(X_train)
model.compile(optimizer='adam', loss='mae')
model.summary()
```

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 1, 5)]            0
_________________________________________________________________
simple_rnn (SimpleRNN)       (None, 1, 16)             352
_________________________________________________________________
simple_rnn_1 (SimpleRNN)     (None, 5)                 110
_________________________________________________________________
repeat_vector (RepeatVector) (None, 1, 5)              0
_________________________________________________________________
simple_rnn_2 (SimpleRNN)     (None, 1, 5)              55
_________________________________________________________________
simple_rnn_3 (SimpleRNN)     (None, 1, 16)             352
_________________________________________________________________
time_distributed (TimeDistri (None, 1, 5)              85
=================================================================
Total params: 954
Trainable params: 954
Non-trainable params: 0
_________________________________________________________________
```

In [14]:

```
nb_epochs = 50
batch_size = 32
history = model.fit(X_train, X_train, epochs=nb_epochs, batch_size=batch_size,
                    validation_split=0.1).history
```

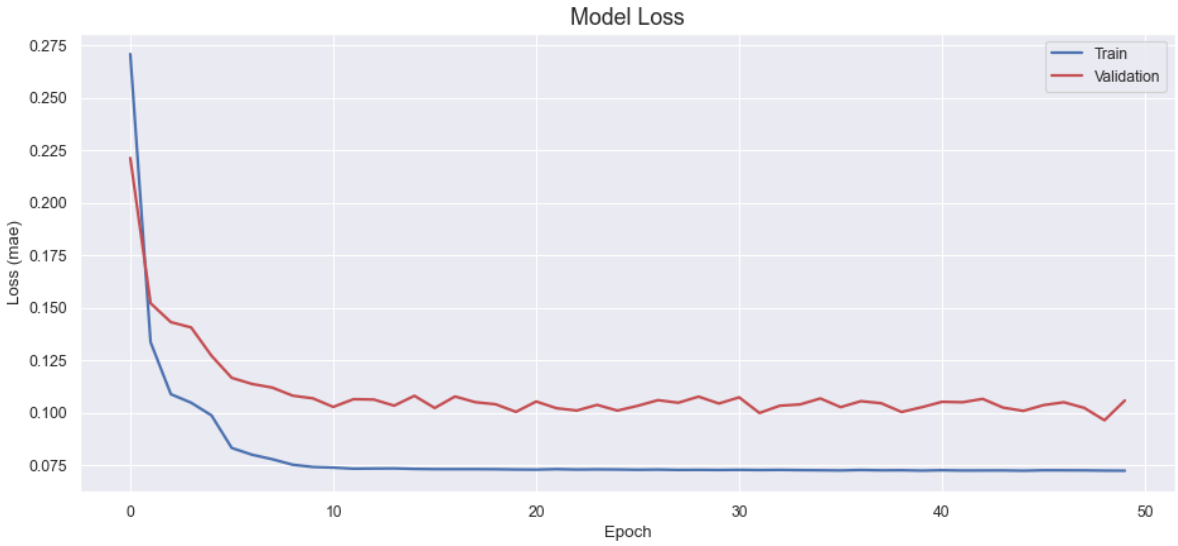
```
Epoch 1/50
445/445 [==============================] - 1s 3ms/step - loss: 0.2708 - val_
loss: 0.2214
Epoch 2/50
445/445 [==============================] - 1s 2ms/step - loss: 0.1337 - val_
loss: 0.1522
Epoch 3/50
445/445 [==============================] - ETA: 0s - loss: 0.109 - 1s 2ms/st
ep - loss: 0.1090 - val_loss: 0.1432
Epoch 4/50
445/445 [==============================] - 1s 2ms/step - loss: 0.1049 - val_
loss: 0.1407
Epoch 5/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0989 - val_
loss: 0.1273
Epoch 6/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0834 - val_
loss: 0.1167
Epoch 7/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0801 - val_
loss: 0.1138
Epoch 8/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0780 - val_
loss: 0.1121
Epoch 9/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0754 - val_
loss: 0.1083
Epoch 10/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0743 - val_
loss: 0.1070
Epoch 11/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0740 - val_
loss: 0.1029
Epoch 12/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0735 - val_
loss: 0.1066
Epoch 13/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0736 - val_
loss: 0.1064
Epoch 14/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0736 - val_
loss: 0.1035
Epoch 15/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0734 - val_
loss: 0.1082
Epoch 16/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0733 - val_
loss: 0.1024
Epoch 17/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0733 - val_
loss: 0.1079
Epoch 18/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0733 - val_
loss: 0.1051
```

```
Epoch 19/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0732 - val_
loss: 0.1042
Epoch 20/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0731 - val_
loss: 0.1005
Epoch 21/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0731 - val_
loss: 0.1055
Epoch 22/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0733 - val_
loss: 0.1023
Epoch 23/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0731 - val_
loss: 0.1012
Epoch 24/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0732 - val_
loss: 0.1039
Epoch 25/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0731 - val_
loss: 0.1011
Epoch 26/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0730 - val_
loss: 0.1034
Epoch 27/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0731 - val_
loss: 0.1061
Epoch 28/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0729 - val_
loss: 0.1049
Epoch 29/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0729 - val_
loss: 0.1078
Epoch 30/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0729 - val_
loss: 0.1045
Epoch 31/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0730 - val_
loss: 0.1074
Epoch 32/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0728 - val_
loss: 0.1000
Epoch 33/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0729 - val_
loss: 0.1035
Epoch 34/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0728 - val_
loss: 0.1041
Epoch 35/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0727 - val_
loss: 0.1069
Epoch 36/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0727 - val_
loss: 0.1028
Epoch 37/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0729 - val_
loss: 0.1056
Epoch 38/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0727 - val_
loss: 0.1046
Epoch 39/50
```

```
445/445 [==============================] - 1s 2ms/step - loss: 0.0728 - val_
loss: 0.1005
Epoch 40/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0726 - val_
loss: 0.1028
Epoch 41/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0728 - val_
loss: 0.1053
Epoch 42/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0726 - val_
loss: 0.1051
Epoch 43/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0726 - val_
loss: 0.1067
Epoch 44/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0727 - val_
loss: 0.1026
Epoch 45/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0725 - val_
loss: 0.1010
Epoch 46/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0728 - val_
loss: 0.1038
Epoch 47/50
445/445 [==============================] - 1s 3ms/step - loss: 0.0727 - val_
loss: 0.1051
Epoch 48/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0727 - val_
loss: 0.1024
Epoch 49/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0726 - val_
loss: 0.0965
Epoch 50/50
445/445 [==============================] - 1s 2ms/step - loss: 0.0725 - val_
loss: 0.1059
```

In [15]:

```
fig, ax = plt.subplots(figsize=(14,6), dpi=80)
ax.plot(history['loss'], 'b', label='Train', linewidth=2)
ax.plot(history['val_loss'], 'r', label='Validation', linewidth=2)
ax.set_title('Model Loss', fontsize=16)
ax.set_ylabel('Loss (mae)')
ax.set_xlabel('Epoch')
ax.legend(loc='upper right')
plt.show()
```



In [16]:

```
X_test
```

Out[16]:

```
array([[[0.545     , 0.38      , 0.5       , 0.955     , 0.73604061]],

       [[0.535     , 0.395     , 0.52      , 0.935     , 0.73096447]],

       [[0.51      , 0.405     , 0.5       , 0.935     , 0.74619289]],

       ...,

       [[0.215     , 0.635     , 0.225     , 0.9       , 0.54314721]],

       [[0.225     , 0.655     , 0.2       , 0.92      , 0.55329949]],

       [[0.23      , 0.65      , 0.19      , 0.945     , 0.54822335]]])
```

In [17]:

```
X_train
```

Out[17]:

```
array([[[0.43      , 0.395     , 0.17      , 0.145     , 0.35025381]],

       [[0.455     , 0.41      , 0.165     , 0.16      , 0.34517766]],

       [[0.46      , 0.4       , 0.17      , 0.135     , 0.34010152]],

       ...,

       [[0.555     , 0.365     , 0.51      , 0.955     , 0.78680203]],

       [[0.555     , 0.375     , 0.52      , 0.97      , 0.76142132]],

       [[0.565     , 0.355     , 0.51      , 0.975     , 0.74619289]]])
```

In [18]:

```
X_train_pred = model.predict(X_train)
```

In [19]:

```
train_mae_loss =pd.DataFrame(np.mean(np.abs(X_train_pred - X_train), axis=1))
```

In [20]:

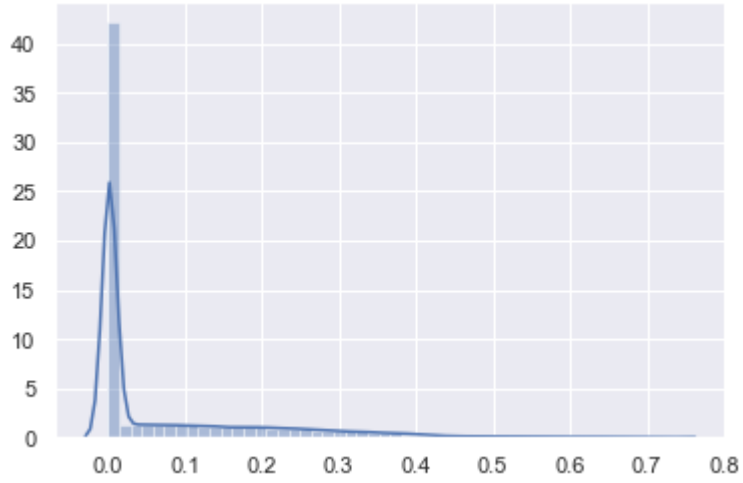
```
train_mae_loss.head(5)
```

Out[20]:

	0	1	2	3	4
0	0.137882	0.001898	0.002032	0.489832	0.000165
1	0.159953	0.001906	0.001988	0.476743	0.000169
2	0.164914	0.001872	0.001976	0.504007	0.000096
3	0.149325	0.001902	0.001980	0.490825	0.000197
4	0.154026	0.001854	0.002000	0.523802	0.000044

In [21]:

```
sns.distplot(train_mae_loss, bins=50, kde=True);
```



In [22]:

```
X_pred = model.predict(X_test)
X_pred = X_pred.reshape(X_pred.shape[0], X_pred.shape[2])
X_pred = pd.DataFrame(X_pred, columns=data_test.columns)
X_pred.index = data_test.index
```

In [23]:

```
X_pred.head(5)
```

Out[23]:

timestamp	ozone	particullate_matter	carbon_monoxide	sulfure_dioxide	nitrogen_dioxide
2014-09-24 21:40:00	0.366981	0.376568	0.498276	0.605667	0.732920
2014-09-24 21:45:00	0.380247	0.391565	0.518180	0.615702	0.727952
2014-09-24 21:50:00	0.372411	0.401578	0.498129	0.604887	0.743163
2014-09-24 21:55:00	0.376101	0.406602	0.498081	0.610820	0.728024
2014-09-24 22:00:00	0.375934	0.421617	0.493081	0.605509	0.743286

In [24]:

```
Xtest = X_test.reshape(X_test.shape[0], X_test.shape[2])
```

In [25]:

```
Xtest
```

Out[25]:

```
array([[0.545     , 0.38      , 0.5       , 0.955     , 0.73604061],
       [0.535     , 0.395     , 0.52      , 0.935     , 0.73096447],
       [0.51      , 0.405     , 0.5       , 0.935     , 0.74619289],
       ...,
       [0.215     , 0.635     , 0.225     , 0.9       , 0.54314721],
       [0.225     , 0.655     , 0.2       , 0.92      , 0.55329949],
       [0.23      , 0.65      , 0.19      , 0.945     , 0.54822335]])
```

In [26]:

```
XtestDataframe=pd.DataFrame(Xtest,columns=['ozone','particullate_matter','carbon_monoxide',
```

In [27]:

```
XtestDataframe
```

Out[27]:

	ozone	particullate_matter	carbon_monoxide	sulfure_dioxide	nitrogen_dioxide
0	0.545	0.380	0.500	0.955	0.736041
1	0.535	0.395	0.520	0.935	0.730964
2	0.510	0.405	0.500	0.935	0.746193
3	0.500	0.410	0.500	0.915	0.730964
4	0.510	0.425	0.495	0.910	0.746193
...	...	...	...	...	...
1752	0.190	0.600	0.205	0.890	0.553299
1753	0.190	0.610	0.200	0.875	0.543147
1754	0.215	0.635	0.225	0.900	0.543147
1755	0.225	0.655	0.200	0.920	0.553299
1756	0.230	0.650	0.190	0.945	0.548223

1757 rows × 5 columns

In [28]:

```
ozonepredict = pd.DataFrame(X_pred.ozone)
ozonepredict
```

Out[28]:

timestamp	ozone
2014-09-24 21:40:00	0.366981
2014-09-24 21:45:00	0.380247
2014-09-24 21:50:00	0.372411
2014-09-24 21:55:00	0.376101
2014-09-24 22:00:00	0.375934
...	...
2014-09-30 23:40:00	0.333384
2014-09-30 23:45:00	0.335643
2014-09-30 23:50:00	0.353074
2014-09-30 23:55:00	0.346776
2014-10-01 00:00:00	0.342001

1757 rows × 1 columns

In [29]:

```
ozonepredict['Loss_mae'] = abs(X_pred.ozone.values - XtestDataframe.ozone.values)
ozonepredict['Threshold'] = 0.2
ozonepredict['Anomaly'] = ozonepredict['Loss_mae'] > ozonepredict ['Threshold']
```

In [30]:

```
ozonepredict.head(10)
```

Out[30]:

timestamp	ozone	Loss_mae	Threshold	Anomaly
2014-09-24 21:40:00	0.366981	0.178019	0.2	False
2014-09-24 21:45:00	0.380247	0.154753	0.2	False
2014-09-24 21:50:00	0.372411	0.137589	0.2	False
2014-09-24 21:55:00	0.376101	0.123899	0.2	False
2014-09-24 22:00:00	0.375934	0.134066	0.2	False
2014-09-24 22:05:00	0.373188	0.156812	0.2	False
2014-09-24 22:10:00	0.376422	0.148578	0.2	False
2014-09-24 22:15:00	0.367285	0.157715	0.2	False
2014-09-24 22:20:00	0.372298	0.157702	0.2	False
2014-09-24 22:25:00	0.365134	0.144866	0.2	False

In [31]:

```
anomalies_ozone = ozonepredict[ozonepredict.Anomaly == True]
print(anomalies_ozone.shape)
anomalies_ozone.head()
```

```
(299, 4)
```

Out[31]:

timestamp	ozone	Loss_mae	Threshold	Anomaly
2014-09-25 17:20:00	0.407184	0.202816	0.2	True
2014-09-26 03:55:00	0.337724	0.212276	0.2	True
2014-09-26 04:00:00	0.332190	0.202810	0.2	True
2014-09-26 04:05:00	0.322625	0.227375	0.2	True
2014-09-26 04:10:00	0.322360	0.242640	0.2	True

In [32]:

```
particullate_matter_predict = pd.DataFrame(X_pred.particullate_matter)
particullate_matter_predict
```

Out[32]:

timestamp	particullate_matter
2014-09-24 21:40:00	0.376568
2014-09-24 21:45:00	0.391565
2014-09-24 21:50:00	0.401578
2014-09-24 21:55:00	0.406602
2014-09-24 22:00:00	0.421617
...	...
2014-09-30 23:40:00	0.596834
2014-09-30 23:45:00	0.606860
2014-09-30 23:50:00	0.631798
2014-09-30 23:55:00	0.651800
2014-10-01 00:00:00	0.646778

1757 rows × 1 columns

In [33]:

```
particullate_matter_predict['Loss_mae'] = abs(X_pred.particullate_matter.values - XtestData
particullate_matter_predict['Threshold'] = 0.2
particullate_matter_predict['Anomaly'] = particullate_matter_predict['Loss_mae'] > particul
```

In [34]:

```
particullate_matter_predict
```

Out[34]:

timestamp	particullate_matter	Loss_mae	Threshold	Anomaly
2014-09-24 21:40:00	0.376568	0.003432	0.2	False
2014-09-24 21:45:00	0.391565	0.003435	0.2	False
2014-09-24 21:50:00	0.401578	0.003422	0.2	False
2014-09-24 21:55:00	0.406602	0.003398	0.2	False
2014-09-24 22:00:00	0.421617	0.003383	0.2	False
...	...	...	...	...
2014-09-30 23:40:00	0.596834	0.003166	0.2	False
2014-09-30 23:45:00	0.606860	0.003140	0.2	False
2014-09-30 23:50:00	0.631798	0.003202	0.2	False
2014-09-30 23:55:00	0.651800	0.003200	0.2	False
2014-10-01 00:00:00	0.646778	0.003222	0.2	False

1757 rows × 4 columns

In [35]:

```
anomalies_particullate_matter = particullate_matter_predict[particullate_matter_predict.Ano
print(anomalies_particullate_matter.shape)
anomalies_particullate_matter.head()
```

```
(0, 4)
```

Out[35]:

timestamp	particullate_matter	Loss_mae	Threshold	Anomaly

In [36]:

```
carbon_monoxide_predict = pd.DataFrame(X_pred.carbon_monoxide)
carbon_monoxide_predict
```

Out[36]:

timestamp	carbon_monoxide
2014-09-24 21:40:00	0.498276
2014-09-24 21:45:00	0.518180
2014-09-24 21:50:00	0.498129
2014-09-24 21:55:00	0.498081
2014-09-24 22:00:00	0.493081
...	...
2014-09-30 23:40:00	0.202321
2014-09-30 23:45:00	0.197298
2014-09-30 23:50:00	0.222284
2014-09-30 23:55:00	0.197315
2014-10-01 00:00:00	0.187365

1757 rows × 1 columns

In [37]:

```
carbon_monoxide_predict['Loss_mae'] = abs(X_pred.carbon_monoxide.values - XtestDataframe.ca
carbon_monoxide_predict['Threshold'] = 0.2
carbon_monoxide_predict['Anomaly'] = carbon_monoxide_predict['Loss_mae'] > carbon_monoxide_
```

In [38]:

```
carbon_monoxide_predict.head(10)
```

Out[38]:

timestamp	carbon_monoxide	Loss_mae	Threshold	Anomaly
2014-09-24 21:40:00	0.498276	0.001724	0.2	False
2014-09-24 21:45:00	0.518180	0.001820	0.2	False
2014-09-24 21:50:00	0.498129	0.001871	0.2	False
2014-09-24 21:55:00	0.498081	0.001919	0.2	False
2014-09-24 22:00:00	0.493081	0.001919	0.2	False
2014-09-24 22:05:00	0.498156	0.001844	0.2	False
2014-09-24 22:10:00	0.523147	0.001853	0.2	False
2014-09-24 22:15:00	0.513181	0.001819	0.2	False
2014-09-24 22:20:00	0.518160	0.001840	0.2	False
2014-09-24 22:25:00	0.503111	0.001889	0.2	False

In [39]:

```
anomalies_carbon_monoxide = carbon_monoxide_predict[carbon_monoxide_predict.Anomaly == True
print(anomalies_carbon_monoxide.shape)
anomalies_carbon_monoxide.head()
```

```
(0, 4)
```

Out[39]:

timestamp	carbon_monoxide	Loss_mae	Threshold	Anomaly

In [40]:

```
sulfure_dioxide_predict = pd.DataFrame(X_pred.sulfure_dioxide)
sulfure_dioxide_predict
```

Out[40]:

timestamp	sulfure_dioxide
2014-09-24 21:40:00	0.605667
2014-09-24 21:45:00	0.615702
2014-09-24 21:50:00	0.604887
2014-09-24 21:55:00	0.610820
2014-09-24 22:00:00	0.605509
...	...
2014-09-30 23:40:00	0.598601
2014-09-30 23:45:00	0.601684
2014-09-30 23:50:00	0.612783
2014-09-30 23:55:00	0.603319
2014-10-01 00:00:00	0.601301

1757 rows × 1 columns

In [41]:

```
sulfure_dioxide_predict['Loss_mae'] = abs(X_pred.sulfure_dioxide.values - XtestDataframe.su
sulfure_dioxide_predict['Threshold'] = 0.2
sulfure_dioxide_predict['Anomaly'] = sulfure_dioxide_predict['Loss_mae'] > sulfure_dioxide_
```

In [42]:

```
sulfure_dioxide_predict.head(10)
```

Out[42]:

timestamp	sulfure_dioxide	Loss_mae	Threshold	Anomaly
2014-09-24 21:40:00	0.605667	0.349333	0.2	True
2014-09-24 21:45:00	0.615702	0.319298	0.2	True
2014-09-24 21:50:00	0.604887	0.330113	0.2	True
2014-09-24 21:55:00	0.610820	0.304180	0.2	True
2014-09-24 22:00:00	0.605509	0.304491	0.2	True
2014-09-24 22:05:00	0.606650	0.288350	0.2	True
2014-09-24 22:10:00	0.608946	0.281054	0.2	True
2014-09-24 22:15:00	0.602126	0.262874	0.2	True
2014-09-24 22:20:00	0.607904	0.232096	0.2	True
2014-09-24 22:25:00	0.598127	0.216873	0.2	True

In [43]:

```
anomalies_sulfure_dioxide = sulfure_dioxide_predict[sulfure_dioxide_predict.Anomaly == True
print(anomalies_sulfure_dioxide.shape)
anomalies_sulfure_dioxide.head()
```

```
(702, 4)
```

Out[43]:

timestamp	sulfure_dioxide	Loss_mae	Threshold	Anomaly
2014-09-24 21:40:00	0.605667	0.349333	0.2	True
2014-09-24 21:45:00	0.615702	0.319298	0.2	True
2014-09-24 21:50:00	0.604887	0.330113	0.2	True
2014-09-24 21:55:00	0.610820	0.304180	0.2	True
2014-09-24 22:00:00	0.605509	0.304491	0.2	True

In [44]:

```
nitrogen_dioxide_predict = pd.DataFrame(X_pred.nitrogen_dioxide)
nitrogen_dioxide_predict
```

Out[44]:

timestamp	nitrogen_dioxide
2014-09-24 21:40:00	0.732920
2014-09-24 21:45:00	0.727952
2014-09-24 21:50:00	0.743163
2014-09-24 21:55:00	0.728024
2014-09-24 22:00:00	0.743286
...	...
2014-09-30 23:40:00	0.550736
2014-09-30 23:45:00	0.540664
2014-09-30 23:50:00	0.540655
2014-09-30 23:55:00	0.550764
2014-10-01 00:00:00	0.545597

1757 rows × 1 columns

In [45]:

```
nitrogen_dioxide_predict['Loss_mae'] = abs(X_pred.nitrogen_dioxide.values - XtestDataframe.
nitrogen_dioxide_predict['Threshold'] = 0.2
nitrogen_dioxide_predict['Anomaly'] = nitrogen_dioxide_predict['Loss_mae'] > nitrogen_dioxi
```

In [46]:

```
nitrogen_dioxide_predict.head(10)
```

Out[46]:

timestamp	nitrogen_dioxide	Loss_mae	Threshold	Anomaly
2014-09-24 21:40:00	0.732920	0.003120	0.2	False
2014-09-24 21:45:00	0.727952	0.003012	0.2	False
2014-09-24 21:50:00	0.743163	0.003030	0.2	False
2014-09-24 21:55:00	0.728024	0.002941	0.2	False
2014-09-24 22:00:00	0.743286	0.002907	0.2	False
2014-09-24 22:05:00	0.738236	0.002880	0.2	False
2014-09-24 22:10:00	0.748363	0.002906	0.2	False
2014-09-24 22:15:00	0.753481	0.002864	0.2	False
2014-09-24 22:20:00	0.743439	0.002754	0.2	False
2014-09-24 22:25:00	0.758728	0.002693	0.2	False

In [47]:

```
anomalies_nitrogen_dioxide = nitrogen_dioxide_predict[nitrogen_dioxide_predict.Anomaly == T
print(anomalies_nitrogen_dioxide.shape)
anomalies_nitrogen_dioxide.head()
```

```
(0, 4)
```

Out[47]:

timestamp	nitrogen_dioxide	Loss_mae	Threshold	Anomaly

In [48]:

```
plt.figure(figsize=(15,10))
plt.plot(anomalies_ozone.ozone, color='black', marker="o", linestyle='-')
plt.plot(X_pred.ozone, color='orange', marker='.', linestyle="dashed")

plt.title('ozone için Aykırı Değer Grafiği ')
plt.xlabel('zaman serisi', fontweight='bold')
plt.ylabel('Mutlak Ortala Hata Değerleri', fontweight='bold')
```

Out[48]:

```
Text(0, 0.5, 'Mutlak Ortala Hata Değerleri')
```

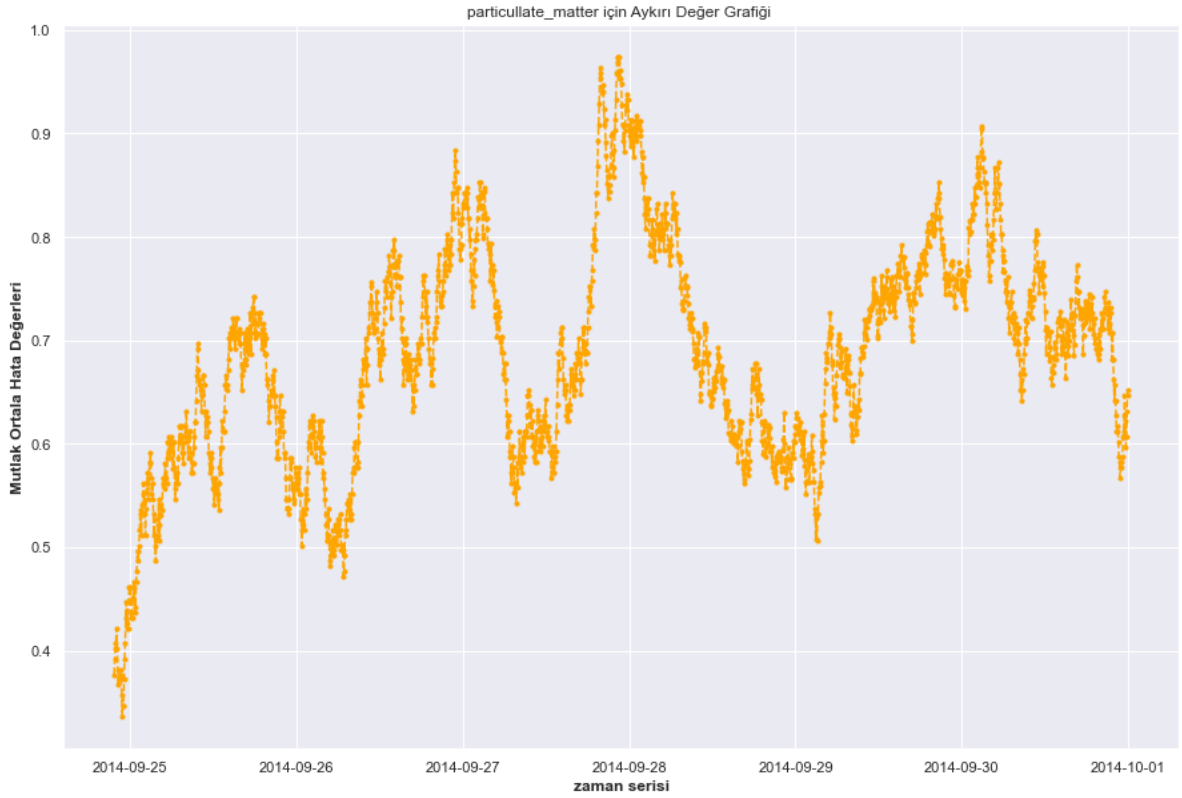
In [49]:

```
plt.figure(figsize=(15,10))
plt.plot(anomalies_particullate_matter.particullate_matter, color='black', marker="o", line
plt.plot(X_pred.particullate_matter, color='orange', marker='.', linestyle="dashed")

plt.title('particullate_matter için Aykırı Değer Grafiği ')
plt.xlabel('zaman serisi', fontweight='bold')
plt.ylabel('Mutlak Ortala Hata Değerleri', fontweight='bold')
```

Out[49]:

```
Text(0, 0.5, 'Mutlak Ortala Hata Değerleri')
```



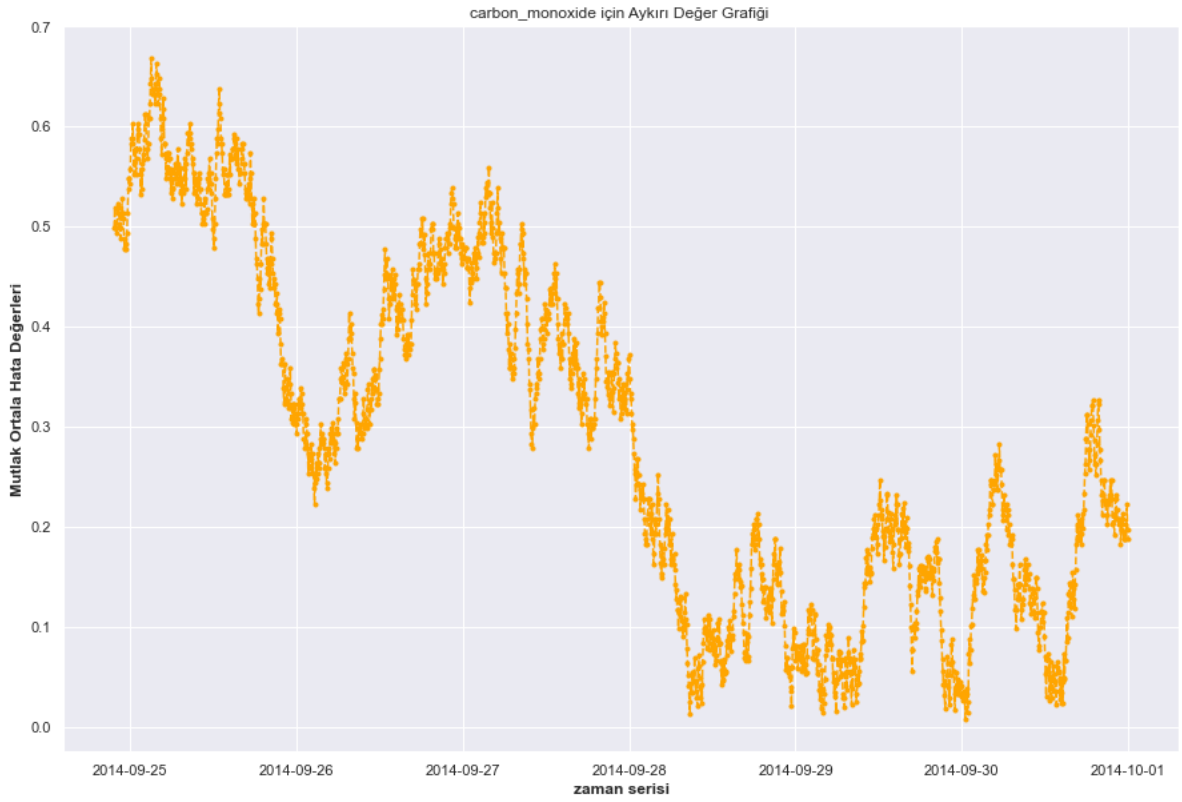
In [50]:

```
plt.figure(figsize=(15,10))
plt.plot(anomalies_carbon_monoxide.carbon_monoxide, color='black', marker="o", linestyle='-
plt.plot(X_pred.carbon_monoxide, color='orange', marker='.', linestyle="dashed")

plt.title('carbon_monoxide için Aykırı Değer Grafiği ')
plt.xlabel('zaman serisi', fontweight='bold')
plt.ylabel('Mutlak Ortala Hata Değerleri', fontweight='bold')
```

Out[50]:

```
Text(0, 0.5, 'Mutlak Ortala Hata Değerleri')
```



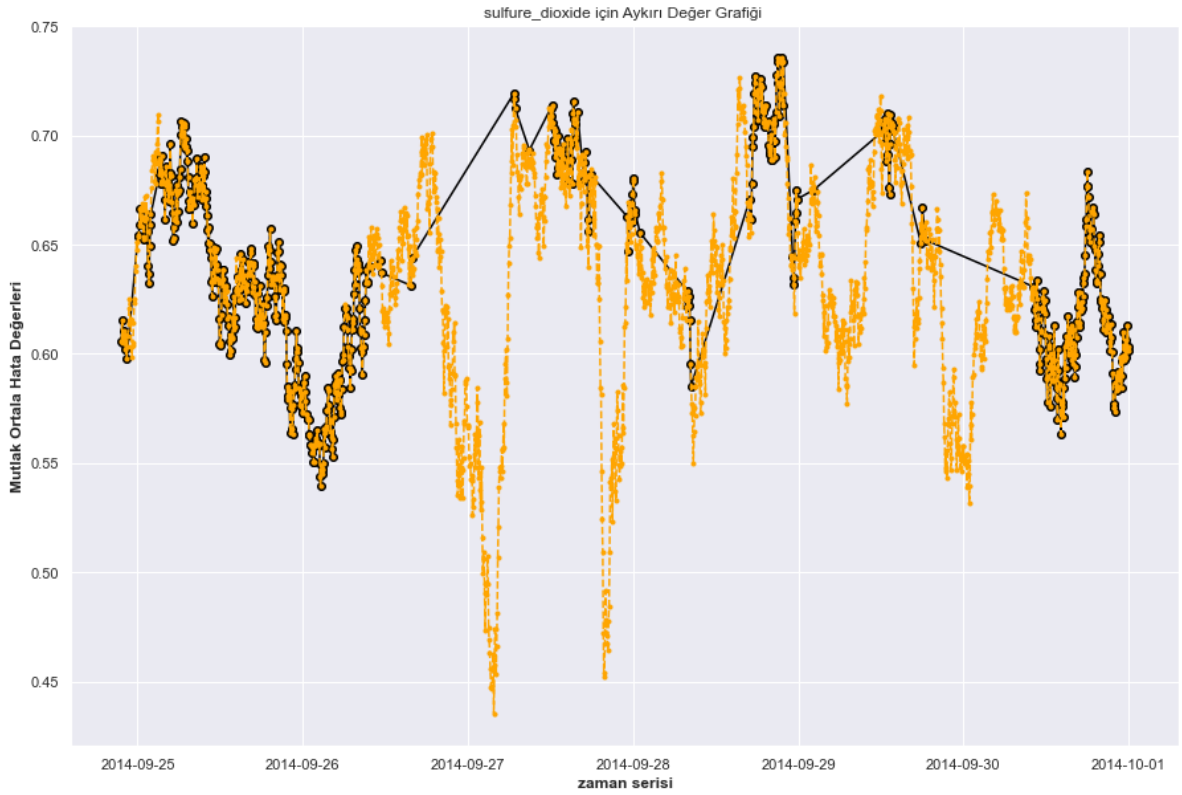
In [51]:

```
plt.figure(figsize=(15,10))
plt.plot(anomalies_sulfure_dioxide.sulfure_dioxide, color='black', marker="o", linestyle='-
plt.plot(X_pred.sulfure_dioxide, color='orange', marker='.', linestyle="dashed")

plt.title('sulfure_dioxide için Aykırı Değer Grafiği ')
plt.xlabel('zaman serisi', fontweight='bold')
plt.ylabel('Mutlak Ortala Hata Değerleri', fontweight='bold')
```

Out[51]:

```
Text(0, 0.5, 'Mutlak Ortala Hata Değerleri')
```



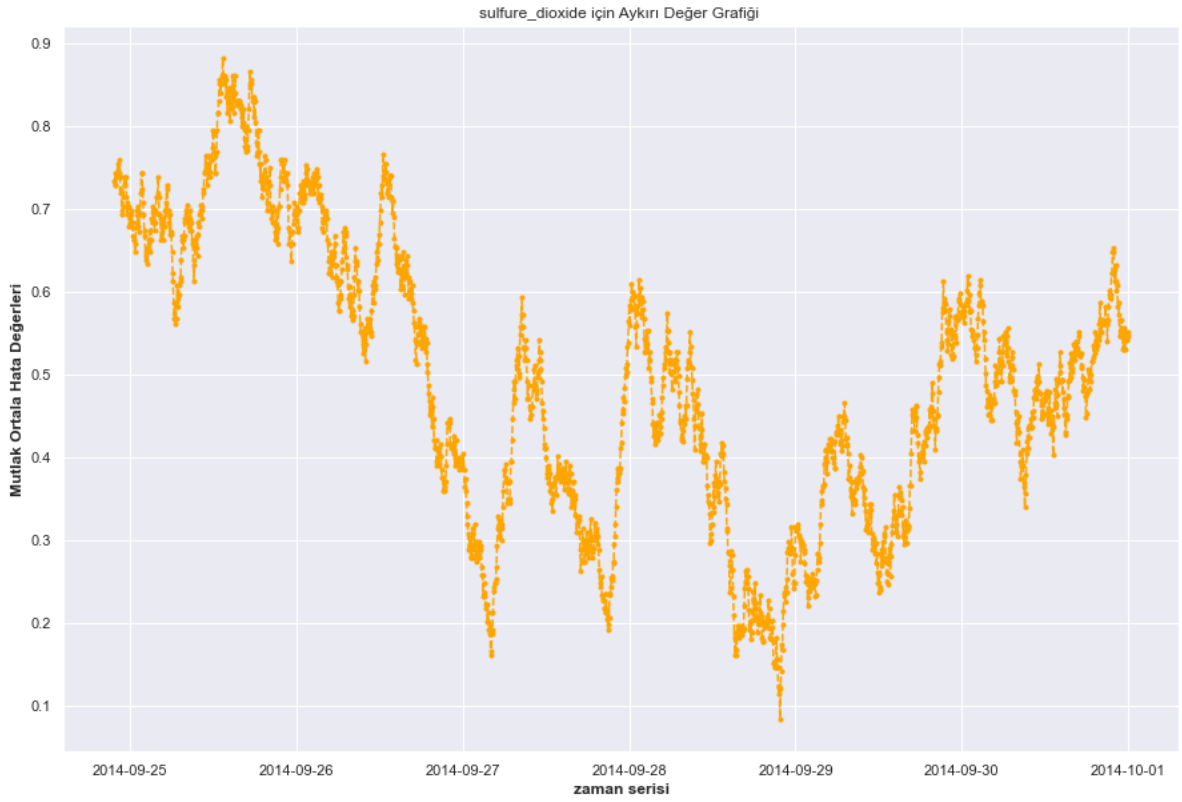
In [52]:

```
plt.figure(figsize=(15,10))
plt.plot(anomalies_nitrogen_dioxide.nitrogen_dioxide, color='black', marker="o", linestyle=
plt.plot(X_pred.nitrogen_dioxide, color='orange', marker='.', linestyle="dashed")

plt.title('sulfure_dioxide için Aykırı Değer Grafiği ')
plt.xlabel('zaman serisi', fontweight='bold')
plt.ylabel('Mutlak Ortala Hata Değerleri', fontweight='bold')
```

Out[52]:

```
Text(0, 0.5, 'Mutlak Ortala Hata Değerleri')
```



In [ ]:

```
```