Metal_Options (3)

April 25, 2021

[177]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.model_selection import train_test_split
```

[178]:
```
# Import required libraries
import pandas as pd
import numpy as np
from pandas.plotting import lag_plot
from pandas.plotting import autocorrelation_plot
from matplotlib import pyplot
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller
import math as math
from scipy.stats import boxcox
from random import randrange
from random import seed
from random import random
from random import gauss
```

[179]:
```
df = pd.read_csv("https://raw.githubusercontent.com/shauryashivam/
 ↪commodity-futures/main/Dataset/Gold15.csv?
 ↪token=AMF2Z3P6KZHZIIIEYQVYHJDARWXZK",header=0, index_col=0, parse_dates=True,
squeeze=True)
```

[180]:
```
df.drop(columns=['Open','High','Low','Adj Close','Volume'],axis=1,inplace=True)
```

[181]:
```
df.head()
```

[181]:
```
                 Close
Date
2006-01-03  530.700012
2006-01-04  533.900024
2006-01-05  526.299988
2006-01-06  539.700012
2006-01-09  549.099976
```

[182]:
```
df.describe()
```

[182]:
```
             Close
count  3742.000000
mean   1225.928113
std     332.620013
min     526.299988
25%    1014.825012
50%    1252.750000
75%    1405.549988
max    2051.500000
```

1 Single Lag

[183]:
```
var = pd.DataFrame(df.values)
dataframe = pd.concat([var.shift(1), var], axis=1)
dataframe.columns = ['t', 't+1']
print(dataframe.head(5))
```

```
            t         t+1
0         NaN  530.700012
1  530.700012  533.900024
2  533.900024  526.299988
3  526.299988  539.700012
4  539.700012  549.099976
```

[184]:
```
var = pd.DataFrame(df.values)
dataframe = pd.concat([var.shift(3), var.shift(2), var.shift(1), var], axis=1)
dataframe.columns = ['t-2', 't-1', 't', 't+1']
print(dataframe.head(5))
```

```
          t-2         t-1           t         t+1
0         NaN         NaN         NaN  530.700012
1         NaN         NaN  530.700012  533.900024
2         NaN  530.700012  533.900024  526.299988
3  530.700012  533.900024  526.299988  539.700012
4  533.900024  526.299988  539.700012  549.099976
```

[185]:
```
var = pd.DataFrame(df.values)
shifted = var.shift(1)
window = shifted.rolling(window=2)
means = window.mean()
dataframe = pd.concat([means, var], axis=1)
dataframe.columns = ['mean(t-1,t)', 't+1']
print(dataframe.head(5))
```

```
   mean(t-1,t)         t+1
```

```
0          NaN  530.700012
1          NaN  533.900024
2   532.300018  526.299988
3   530.100006  539.700012
4   533.000000  549.099976
```

[186]:
```
var = pd.DataFrame(df.values)
window = var.expanding()
dataframe = pd.concat([window.min(), window.mean(), window.max(), var.
 ↪shift(-1)], axis=1)
dataframe.columns = ['min', 'mean', 'max', 't+1']
print(dataframe.head(5))
```

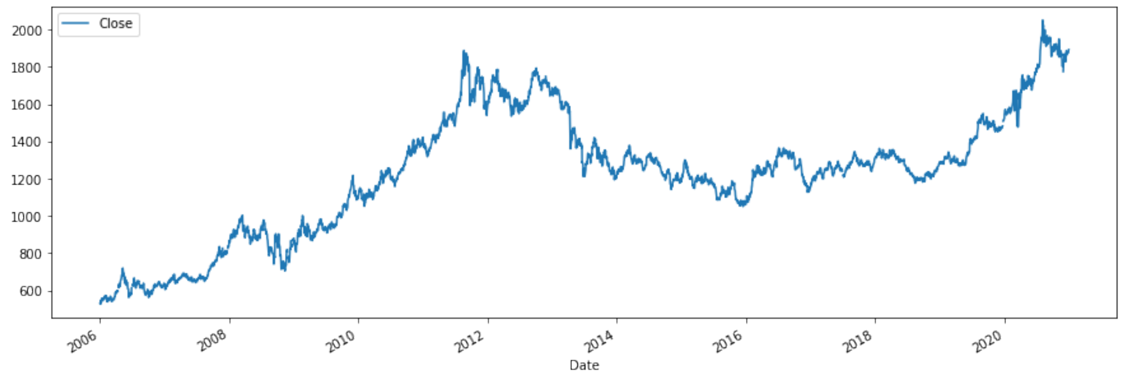
```
          min        mean         max         t+1
0  530.700012  530.700012  530.700012  533.900024
1  530.700012  532.300018  533.900024  526.299988
2  526.299988  530.300008  533.900024  539.700012
3  526.299988  532.650009  539.700012  549.099976
4  526.299988  535.940002  549.099976  544.299988
```

[187]:
```
dataframe = pd.DataFrame()
dataframe['month'] = [df.index[i].month for i in range(len(df))]
dataframe['day'] = [df.index[i].day for i in range(len(df))]
dataframe['Close'] = [df['Close'] for i in range(len(df))]
print(dataframe.head(5))
```

```
   month  day                                             Close
0      1    3  Date
2006-01-03    530.700012
2006-01-04    …
1      1    4  Date
2006-01-03    530.700012
2006-01-04    …
2      1    5  Date
2006-01-03    530.700012
2006-01-04    …
3      1    6  Date
2006-01-03    530.700012
2006-01-04    …
4      1    9  Date
2006-01-03    530.700012
2006-01-04    …
```

[188]:
```
df.plot(figsize=(15,5))
```

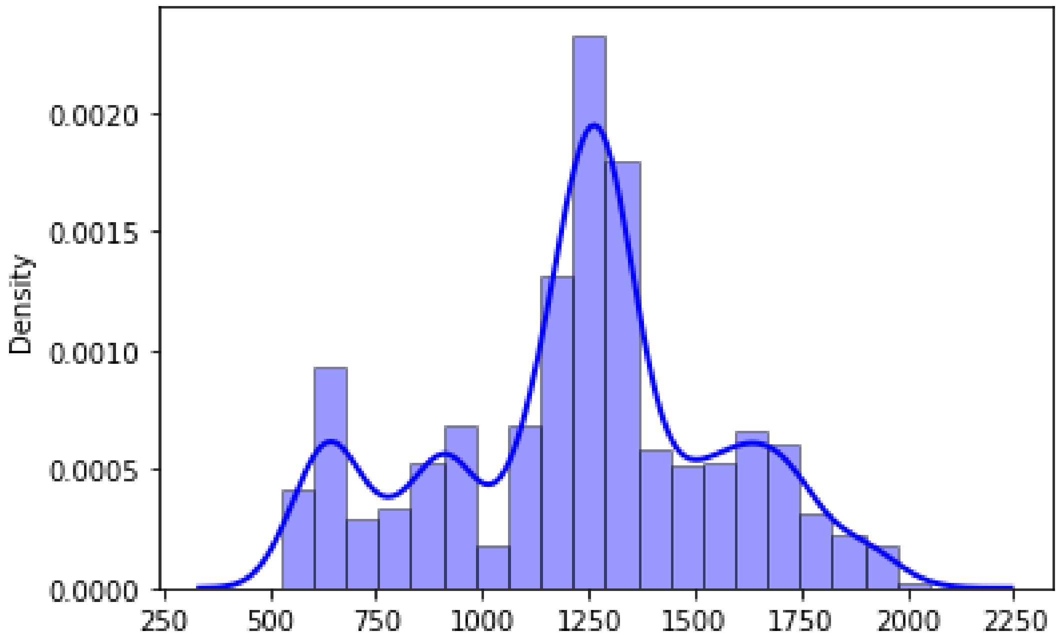
[188]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a478ffd50>
```



[189]:
```python
import seaborn as sns
sns.distplot(df, hist=True, kde=True,
             bins=20,
             color = 'blue',
             hist_kws={'edgecolor':'black'},
             kde_kws={'linewidth': 2})
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

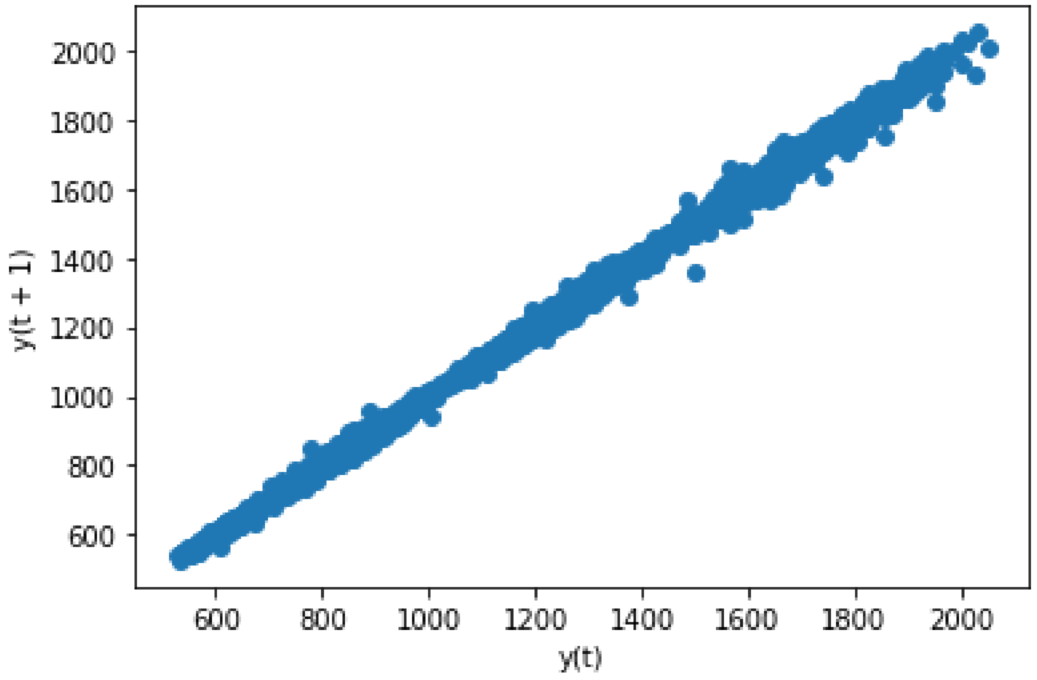
[189]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a46538dd0>
```



```
[190]: df.head(
       )
```

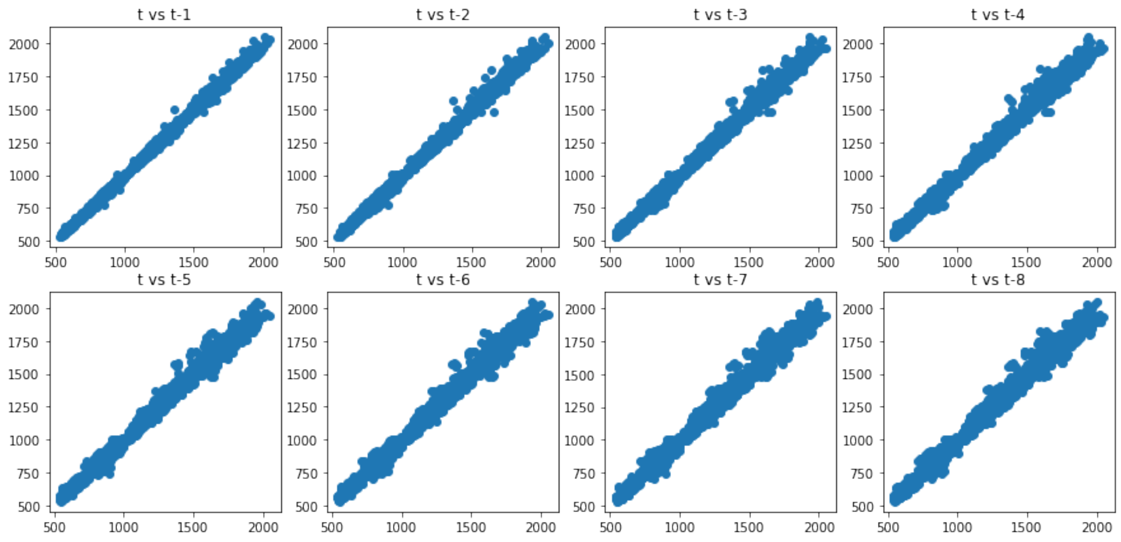
```
[190]:                      Close
       Date
       2006-01-03  530.700012
       2006-01-04  533.900024
       2006-01-05  526.299988
       2006-01-06  539.700012
       2006-01-09  549.099976
```

```
[191]: lag_plot(df)
       pyplot.show()
```



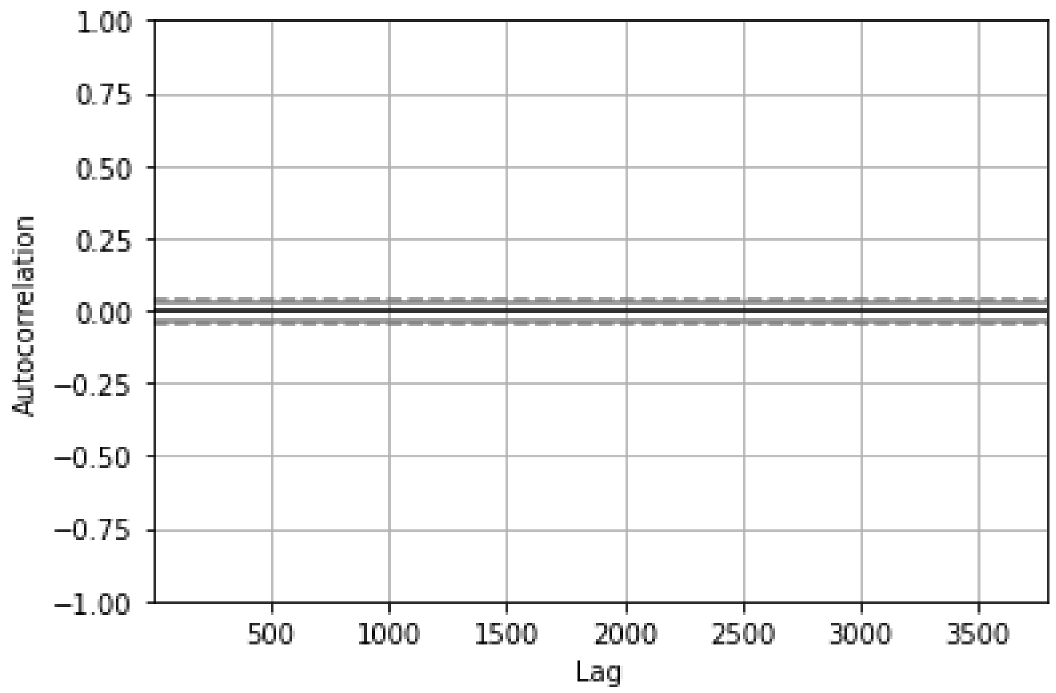
```
[192]: values = pd.DataFrame(df.values)
       lags = 8
       columns = [values]
       for i in range(1,(lags + 1)):
           columns.append(values.shift(i))
       dataframe = pd.concat(columns, axis=1)
       columns = ['t']
       for i in range(1,(lags + 1)):
```

```
    columns.append('t-' + str(i))
dataframe.columns = columns
pyplot.figure(1,figsize=(15,7))
for i in range(1,(lags + 1)):
    ax = pyplot.subplot(240 + i)
    ax.set_title('t vs t-' + str(i))
    pyplot.scatter(x=dataframe['t'].values, y=dataframe['t-'+str(i)].values)
pyplot.show()
```



[193]:
```
autocorrelation_plot(df)
```

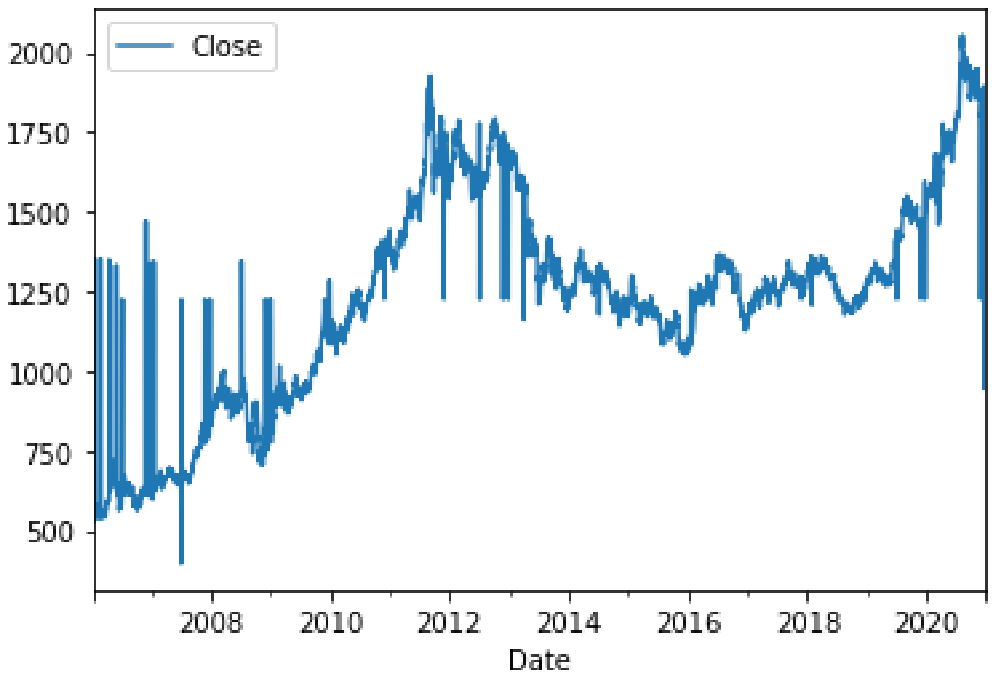
[193]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a46b45390>
```



```
[194]: df=df.fillna(df.mean())
```

```
[195]: upsampled = df.resample('D').mean()
       interpolated = upsampled.interpolate(method='quadratic')
       interpolated.plot()
```

```
[195]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0a79198c10>
```



[196]:
```
pip install mxnet
```

```
Requirement already satisfied: mxnet in /usr/local/lib/python3.7/dist-packages
(1.8.0.post0)
Requirement already satisfied: numpy<2.0.0,>1.16.0 in
/usr/local/lib/python3.7/dist-packages (from mxnet) (1.19.5)
Requirement already satisfied: requests<3,>=2.20.0 in
/usr/local/lib/python3.7/dist-packages (from mxnet) (2.23.0)
Requirement already satisfied: graphviz<0.9.0,>=0.8.1 in
/usr/local/lib/python3.7/dist-packages (from mxnet) (0.8.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.20.0->mxnet)
(2020.12.5)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.20.0->mxnet) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.20.0->mxnet)
(1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests<3,>=2.20.0->mxnet) (2.10)
```

[197]:
```
import time
import numpy as np

from mxnet import nd, autograd, gluon
from mxnet.gluon import nn, rnn
import mxnet as mx
import datetime
import seaborn as sns

import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.decomposition import PCA

import math

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler

import xgboost as xgb
from sklearn.metrics import accuracy_score
```

[198]:
```
df.describe()
```

[198]:
```
             Close
count  3788.000000
mean   1225.928113
std     330.593703
min     526.299988
25%    1050.474976
50%    1250.000000
75%    1397.725006
max    2051.500000
```

2 BUILDING MODEL

2.1 LSTM-GRU

[199]:
```
model = tf.keras.Sequential()
model.add(tf.keras.layers.GRU(5,activation = 'relu', input_shape=(1,1)))
model.add(Dense(100,activation='relu'))
model.add(Dense(1))
model.compile(loss='mse',optimizer='adam',metrics=['mae'])
```

2.2 LSTM CNN Attention

[200]:
```
!pip install keras-attention
!pip install keras-self-attention
```

```
Requirement already satisfied: keras-attention in /usr/local/lib/python3.7/dist-
packages (1.0.0)
```

```
Requirement already satisfied: keras in /usr/local/lib/python3.7/dist-packages
(from keras-attention) (2.4.3)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages
(from keras->keras-attention) (3.13)
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.7/dist-
packages (from keras->keras-attention) (1.4.1)
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages
(from keras->keras-attention) (2.10.0)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-
packages (from keras->keras-attention) (1.19.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from h5py->keras->keras-attention) (1.15.0)
Requirement already satisfied: keras-self-attention in
/usr/local/lib/python3.7/dist-packages (0.49.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages
(from keras-self-attention) (1.19.5)
Requirement already satisfied: Keras in /usr/local/lib/python3.7/dist-packages
(from keras-self-attention) (2.4.3)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages
(from Keras->keras-self-attention) (3.13)
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages
(from Keras->keras-self-attention) (2.10.0)
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.7/dist-
packages (from Keras->keras-self-attention) (1.4.1)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from h5py->Keras->keras-self-attention) (1.15.0)
```

[201]:
```
import os
import time
import warnings
import numpy as np
import pandas as pd
import operator
from functools import reduce
import h5py
from numpy import newaxis
from keras.layers.core import Dense, Activation, Dropout
from keras.layers import Convolution1D, MaxPooling1D, Flatten, ␣
 ↪Embedding,Bidirectional, GRU
from keras.layers import Conv1D, GlobalMaxPooling1D, merge
from keras.layers.recurrent import LSTM
from keras.models import Sequential
from keras_self_attention import SeqSelfAttention
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler, Normalizer
from sklearn.metrics import mean_squared_error, mean_absolute_error
from math import sqrt
```

```
[202]: def create_dataset(dataset, look_back=1, columns = ['Close']):
           dataX, dataY = [], []
           for i in range(len(dataset.index)):
               if i < look_back:
                   continue
               a = None
               for c in columns:
                   b = dataset.loc[dataset.index[i-look_back:i], c].to_numpy()
                   if a is None:
                       a = b
                   else:
                       a = np.append(a,b)
               dataX.append(a)
               dataY.append(dataset.loc[dataset.index[i-look_back], columns].
        ↪to_numpy())
           return np.array(dataX), np.array(dataY)
```

```
[175]: look_back = 7 # 10, 13
       sc = StandardScaler()
       df.loc[:, 'Close'] = sc.fit_transform(df.Close.values.reshape(-1,1)) # fit.
        ↪transform()
       print(df.loc[:, 'Close'])

       # Create training data
       #train_df = df.loc[df.index < pd.to_datetime('2010-01-01')]
       train_df = df.loc[df.index < df.index[int(len(df.index)*0.8)]]
       train_x, train_y = create_dataset(train_df, look_back=look_back)


       # Construct the whole LSTM + CNN
       model = Sequential()
       # LSTM
       model.add(GRU(6,input_shape = (look_back, 1), input_dim=1 ,  ␣
        ↪return_sequences=True))

       #model.add(LSTM(input_shape = (look_back,1), input_dim=1, output_dim=6,␣
        ↪return_sequences=True))
       #model.add(Dense(1))
       #model.add(Activation('relu')) # ReLU : y = max(0,x)

       # Attention Mechanism
       model.add(SeqSelfAttention(attention_activation='sigmoid', name='Attention'))

       # CNN
       model.add(Convolution1D(input_shape = (look_back,1),
                               filters=64,# 32,128
                              kernel_size=2,
                              activation='relu',
                               ))
       #model.add(MaxPooling1D(pool_length=2))

       '''model.add(Convolution1D(input_shape = (look_back,1),
                               nb_filter=64,
                               filter_length=2,
                               border_mode='valid',
                               activation='relu',
                               subsample_length=1))'''
       model.add(MaxPooling1D(pool_size=(2)))

       model.add(Dropout(0.25))

       #model.add(Dense(250))
       #model.add(Dropout(0.25,input_shape=(2,)))
       model.add(Activation('relu')) # ReLU : y = max(0,x)
       model.add(Dense(1))
       model.add(Activation('linear')) # Linear : y = x

       # Print whole structure of the model
       print(model.summary())

       # training the train data with n epoch
       model.compile(loss="mse", optimizer="adam") # adam, rmsprop
       result = model.fit(np.atleast_3d(np.array(train_x)),
                 np.atleast_3d(train_y),
                 epochs=100,
                 batch_size=80, verbose=1, shuffle=False)


       with open('data_lstm_attention_cnn_palladium.txt','w') as f:
           f.write(str(result.history))

       model.save('lstm_attention_cnn_palladium.h5')


       # Make prediction and specify on the line chart
       predictors = ['Close']
       df['Pred'] = df.loc[df.index[0], 'Close']
       for i in range(len(df.index)):
           if i < look_back:
               continue
           a = None
           for c in predictors:
               b = df.loc[df.index[i-look_back:i], c].to_numpy()
               if a is None:
                   a = b
               else:
                   a = np.append(a,b)
               a = a
           y = model.predict(a.reshape(1,look_back*len(predictors),1))
           df.loc[df.index[i], 'Pred']=y[0][0]

       df.loc[:, 'Close'] = sc.inverse_transform(df.loc[:, 'Close'])
       df.loc[:, 'Pred'] = sc.inverse_transform(df.loc[:, 'Pred'])

       def mape(y_true, y_pred):
           n = len(y_true)
           mape = sum(np.abs((y_true - y_pred) / y_true)) / n * 100
           return mape

       # present the line chart and some parameters like MSE, which reflects the␣
        ↪accuracy of the model in sample or out sample

       plt.grid(ls='--')
       plt.plot(df.loc[df.index < df.index[int(len(df.index)*0.8)], 'Pred'], 'orange',␣
        ↪label = 'Insample Prediction')
       plt.plot(df.loc[df.index >= df.index[int(len(df.index)*0.8)], 'Pred'], 'g',␣
        ↪label = 'Outsample Prediction')
       plt.plot(df.Close ,'b', label = 'Price')
       plt.xlabel('Date')
       plt.ylabel('Closing Price')
       #print('%e'%mean_squared_error(df.loc[df.index < pd.
        ↪to_datetime('2010-01-01'),'Close'],df.loc[df.index < pd.
        ↪to_datetime('2010-01-01'),'Pred']))
       #print('%e'%mean_squared_error(df.loc[df.index >= pd.
        ↪to_datetime('2010-01-01'),'Close'],df.loc[df.index >= pd.
        ↪to_datetime('2010-01-01'),'Pred']))
       print('The RMSE is ','%e'%sqrt(mean_squared_error(df.loc[df.index >= df.
        ↪index[int(len(df.index)*0.8)], 'Close'], df.loc[df.index >= df.
        ↪index[int(len(df.index)*0.8)], 'Pred'])))
       print('The RMAE is ','%e'%sqrt(mean_absolute_error(df.loc[df.index >= df.
        ↪index[int(len(df.index)*0.8)], 'Close'], df.loc[df.index >= df.
        ↪index[int(len(df.index)*0.8)], 'Pred'])))
       print('The MAPE is ','%e'%mape(df.loc[df.index >= df.index[int(len(df.index)*0.
        ↪8)], 'Close'], df.loc[df.index >= df.index[int(len(df.index)*0.8)], 'Pred']))


       plt.legend()
       plt.savefig("lstm_attention_cnn_palladium.eps", format='eps', dpi=1000)
       plt.show()
```

```
Date
2011-01-03    0.097506
2011-01-04   -0.107242
2011-01-05   -0.130921
2011-01-06   -0.140206
2011-01-07   -0.153671
                ...
2020-12-24    0.000000
2020-12-28    2.208134
2020-12-29    2.219741
2020-12-30    2.272205
2020-12-31    2.281955
Name: Close, Length: 2518, dtype: float64
Model: "sequential_19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_5 (GRU)                  (None, 7, 6)              162
_________________________________________________________________
Attention (SeqSelfAttention) (None, None, 6)           449
_________________________________________________________________
conv1d_9 (Conv1D)            (None, None, 64)          832
_________________________________________________________________
max_pooling1d_8 (MaxPooling1 (None, None, 64)          0
_________________________________________________________________
dropout_6 (Dropout)          (None, None, 64)          0
_________________________________________________________________
activation_12 (Activation)   (None, None, 64)          0
_________________________________________________________________
dense_12 (Dense)             (None, None, 1)           65
_________________________________________________________________
activation_13 (Activation)   (None, None, 1)           0
=================================================================
Total params: 1,508
Trainable params: 1,508
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/100
26/26 [==============================] - 3s 9ms/step - loss: 1.1212
Epoch 2/100
26/26 [==============================] - 0s 9ms/step - loss: 0.6248
Epoch 3/100
26/26 [==============================] - 0s 8ms/step - loss: 0.3597
Epoch 4/100
```

```
26/26 [==============================] - 0s 8ms/step - loss: 0.1765
Epoch 5/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0808
Epoch 6/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0522
Epoch 7/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0526
Epoch 8/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0512
Epoch 9/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0476
Epoch 10/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0509
Epoch 11/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0463
Epoch 12/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0477
Epoch 13/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0432
Epoch 14/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0453
Epoch 15/100
26/26 [==============================] - 0s 10ms/step - loss: 0.0458
Epoch 16/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0440
Epoch 17/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0445
Epoch 18/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0424
Epoch 19/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0424
Epoch 20/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0417
Epoch 21/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0409
Epoch 22/100
26/26 [==============================] - 0s 10ms/step - loss: 0.0370
Epoch 23/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0395
Epoch 24/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0393
Epoch 25/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0372
Epoch 26/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0350
Epoch 27/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0374
Epoch 28/100
```

```
26/26 [==============================] - 0s 9ms/step - loss: 0.0364
Epoch 29/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0345
Epoch 30/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0357
Epoch 31/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0361
Epoch 32/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0374
Epoch 33/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0338
Epoch 34/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0326
Epoch 35/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0339
Epoch 36/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0346
Epoch 37/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0331
Epoch 38/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0341
Epoch 39/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0350
Epoch 40/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0343
Epoch 41/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0348
Epoch 42/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0357
Epoch 43/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0358
Epoch 44/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0433
Epoch 45/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0482
Epoch 46/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0556
Epoch 47/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0677
Epoch 48/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0611
Epoch 49/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0662
Epoch 50/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0522
Epoch 51/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0494
Epoch 52/100
```

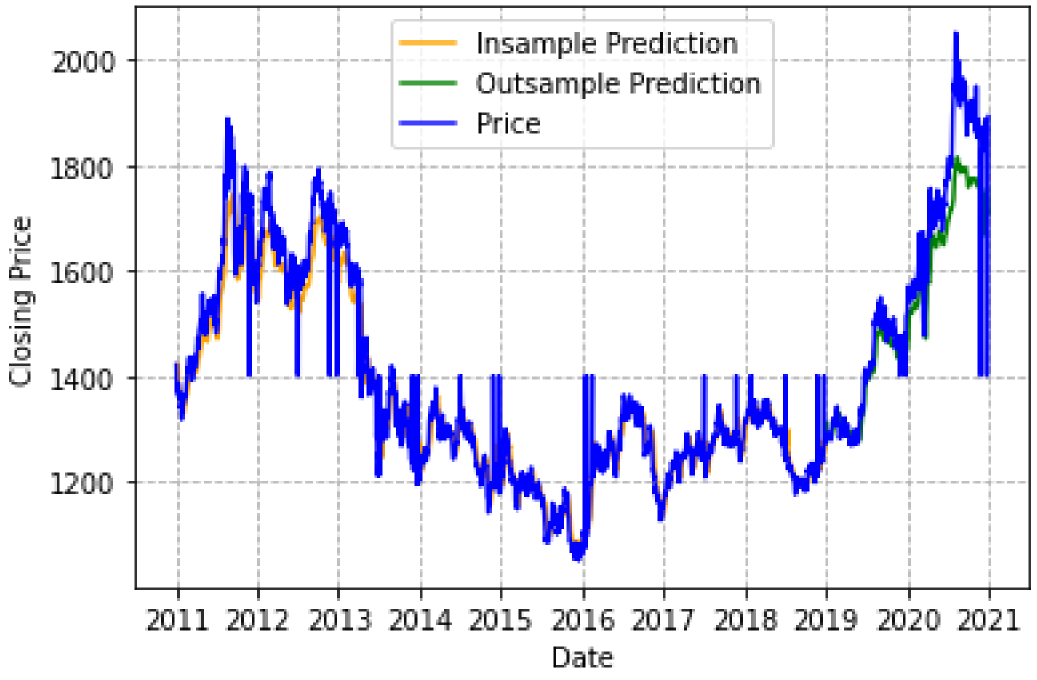
```
26/26 [==============================] - 0s 9ms/step - loss: 0.0406
Epoch 53/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0388
Epoch 54/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0367
Epoch 55/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0346
Epoch 56/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0351
Epoch 57/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0345
Epoch 58/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0334
Epoch 59/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0333
Epoch 60/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0351
Epoch 61/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0354
Epoch 62/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0352
Epoch 63/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0367
Epoch 64/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0361
Epoch 65/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0353
Epoch 66/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0359
Epoch 67/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0356
Epoch 68/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0360
Epoch 69/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0380
Epoch 70/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0401
Epoch 71/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0423
Epoch 72/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0428
Epoch 73/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0444
Epoch 74/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0448
Epoch 75/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0491
Epoch 76/100
```

```
26/26 [==============================] - 0s 9ms/step - loss: 0.0453
Epoch 77/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0462
Epoch 78/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0426
Epoch 79/100
26/26 [==============================] - 0s 10ms/step - loss: 0.0418
Epoch 80/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0396
Epoch 81/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0404
Epoch 82/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0398
Epoch 83/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0406
Epoch 84/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0378
Epoch 85/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0358
Epoch 86/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0353
Epoch 87/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0348
Epoch 88/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0333
Epoch 89/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0318
Epoch 90/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0326
Epoch 91/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0325
Epoch 92/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0341
Epoch 93/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0343
Epoch 94/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0350
Epoch 95/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0361
Epoch 96/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0362
Epoch 97/100
26/26 [==============================] - 0s 8ms/step - loss: 0.0407
Epoch 98/100
26/26 [==============================] - 0s 9ms/step - loss: 0.0438
Epoch 99/100
26/26 [==============================] - 0s 10ms/step - loss: 0.0435
Epoch 100/100
```

```
26/26 [==============================] - 0s 9ms/step - loss: 0.0432
```

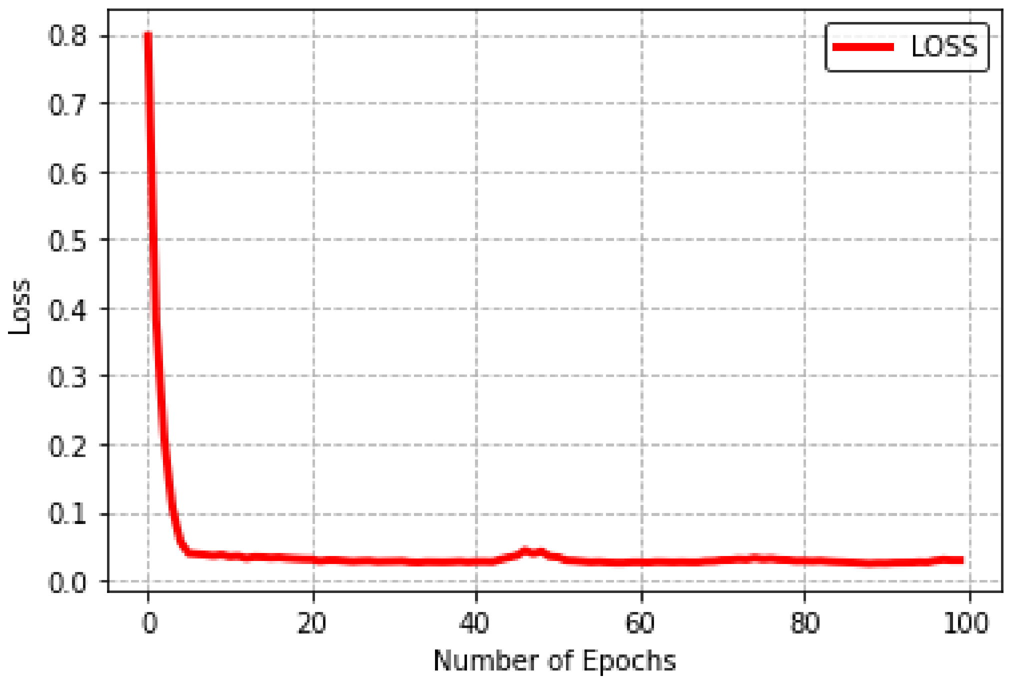
```
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
```

```
The RMSE is  8.342831e+01
The RMAE is  7.827046e+00
The MAPE is  3.551202e+00
```



[176]:
```python
# sketch loss
#plt.cla() # clear the axis
plt.grid(ls='--')
plt.plot(result.epoch,result.history['loss'],label='LOSS',c='r',lw=3)
#plt.scatter(result.epoch,result.history['loss'],s=15,c='r')
plt.xlabel('Number of Epochs')
plt.ylabel('Loss')
plt.legend(loc='upper right', frameon=True, edgecolor='black')
plt.savefig("LC_loss.eps", format='eps', dpi=1000)
plt. close(0)
```

```
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
```


LOSS
Loss
Number of Epochs
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
20
40
60
80
100

```
[176]:
```