Jupyter Notebook

May 28, 2020

Abd Al Bassir Abou El Ailah, Essam Saied, Trevor Mazhindu

WorlQuant University - Capstone Project

1 Exploratory Analysis

1.1 Get the data

[1]:

```
# Importing the necessary libraries
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import os
from IPython.display import HTML
from IPython.display import display
from IPython.display import Image
import pyfolio as pf
from statsmodels.graphics.tsaplots import plot_acf
import pickle
plt.rcParams.update({'font.size': 20})
figsize = (16, 10)
figsizesub = (14, 26)
```

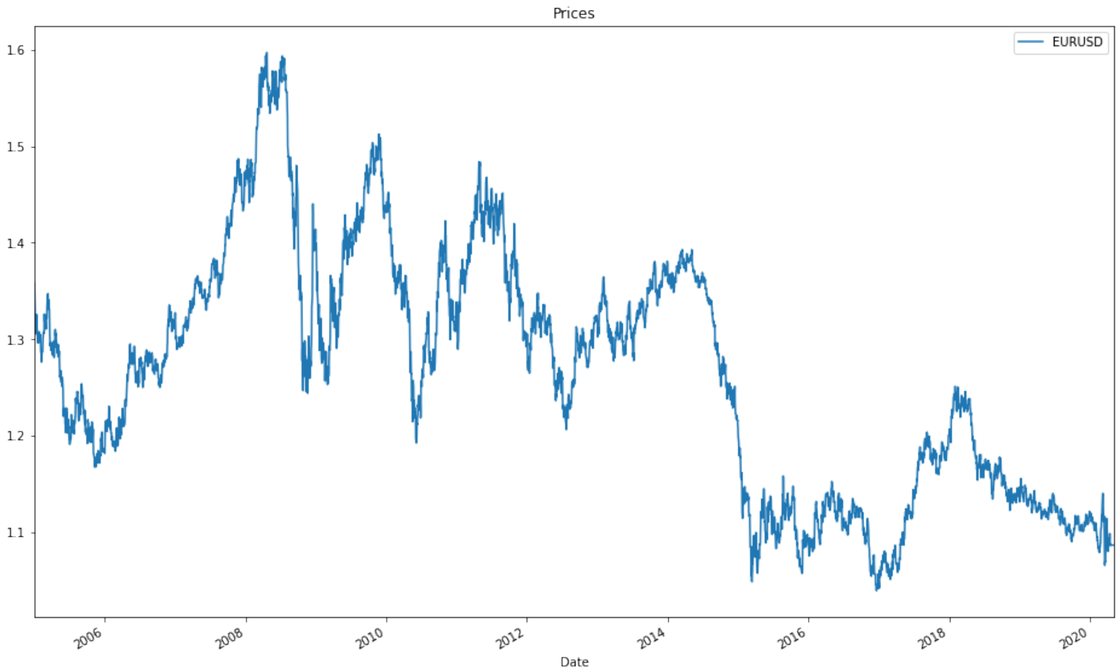
```
/usr/local/lib/python3.6/dist-packages/pyfolio/pos.py:28: UserWarning: Module
"zipline.assets" not found; mutltipliers will not be applied to position
notionals.
  ' to position notionals.'
```

[2]:

```
# Default Parameters
# Settings File
# EMD Denoising Parameters
denoiseLevel = 1
# Features for LSTM Model
inSampleDate = '2018-02-01' # From the begining until this date is for training
features = [
    'ADX', 'ATR', 'BBL', 'C2L', 'C2O', 'CCI', 'C',
    'H2C', 'H2L', 'H2O', 'H', 'L', 'LogPrice',
    'MOM', 'O', 'PercentD', 'PercentK', 'RSI']
# PCA
applyPCA = True
variabilityRatio = 0.95
# LSTM Parameters
lookForward = 1 # Forecast 1 day ahead
nDaysScale = 252 # Rolling z-score normalization is used
timeSteps = 5
nCells = 50
epochs = 35
nLayers = 5
dropout = 0.3
lstmStr = 'PCA_{}_{}_TS_{}_NCells_{}_Epochs_{}_NLayers_{}_Dropout_{}'.format(
    applyPCA, variabilityRatio, timeSteps, nCells, epochs, nLayers, dropout)
#
# Filenames for Cashing
inputFilename = './Input/EURUSD.1Day.csv'
featuresStr = '.'.join(features)
cashDir = './Cashe'
featuresDir = '{}/Features'.format(cashDir)
emdDir = '{}/EMD_{}'.format(featuresDir, denoiseLevel)
modelDir = '{}/Model'.format(cashDir)
resultsDir = '{}/Results'.format(cashDir)
pricesFilename = '{}/prices.pickle'.format(cashDir)
featuresFilename = './Cashe/Features/{}_EMD_{}.pickle'.format(featuresStr,␣
 ↪denoiseLevel)
featuresWithYFilename = './Cashe/Features/{}_Response_EMD_{}_PCA_{}_{}.pickle'.
 ↪format(
    featuresStr, denoiseLevel, applyPCA, variabilityRatio)
modelFilename = './Cashe/Model/Model_{}_EMD_{}_{}.h5'.format(
    featuresStr, denoiseLevel, lstmStr)
equityFilename = './Cashe/Results/EquityCurve_{}_EMD_{}_{}'.format(
    featuresStr, denoiseLevel, lstmStr)
predictionFilename = './Cashe/Results/Prediction_{}_EMD_{}_{}'.format(
    featuresStr, denoiseLevel, lstmStr)
equityFigureFilename = equityFilename + '.png'
equityPickleFilename = equityFilename + '.pickle'
```

[3]:

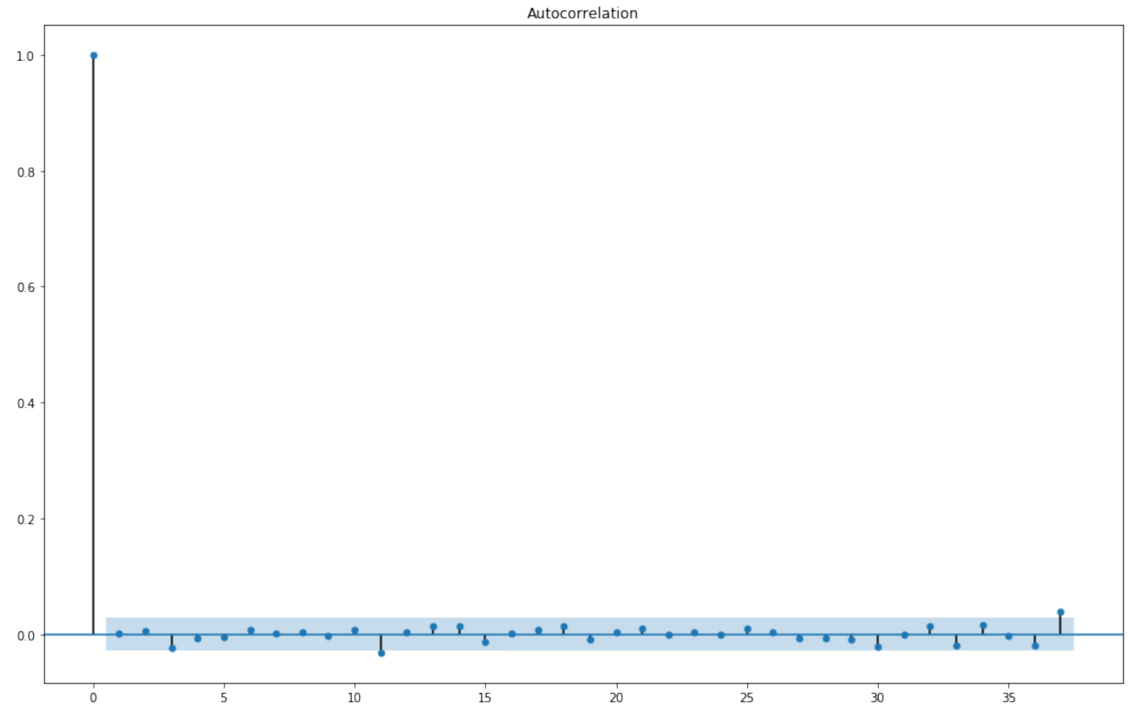
```
dfPrices = pd.read_pickle('../Cashe/prices.pickle')
dfPrices.plot(figsize=figsize, title='Prices');
plt.xlabel('Date');
plt.savefig('./Images/1_prices.png')
```



2 Autocorrelation

[4]:

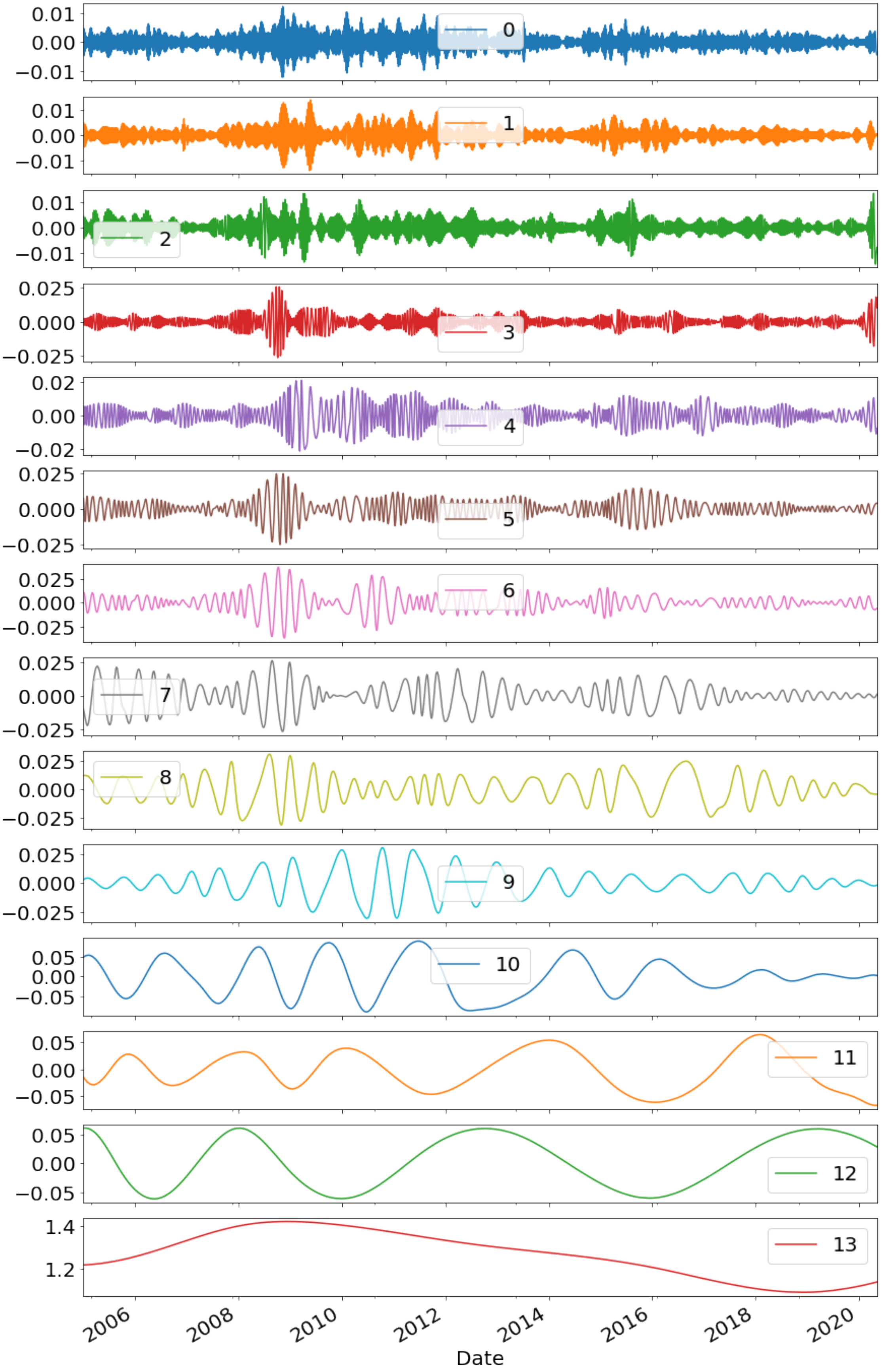
```
fig = plt.figure(figsize=figsize)
ax = fig.add_subplot(111)
plot_acf(dfPrices.pct_change().fillna(0), ax=ax)
plt.savefig('./Images/2_autocorrelation.png')
```



3 Empirical Mode Decomposition (EMD)

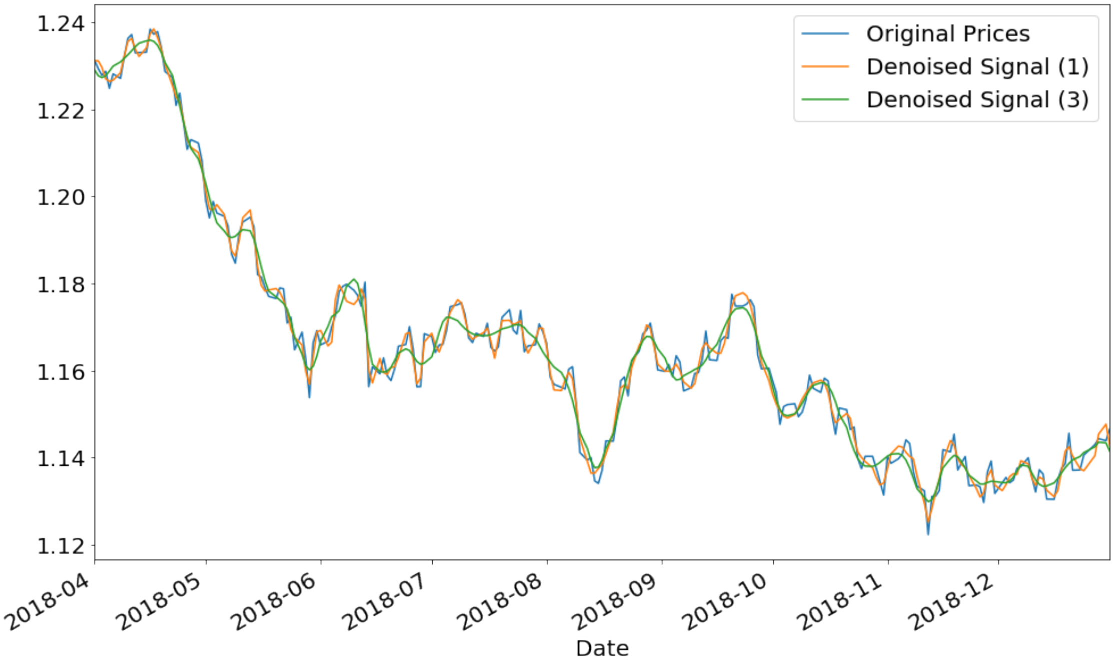
[45]:

```
filename = '../Cashe/globalData_Imfs.pickle'
if os.path.exists(filename):
    with open(filename, 'rb') as f:
        dictImfs = pickle.load(f)
dfImfs = dictImfs['close']
dfImfs.plot(subplots=True, figsize=figsizesub);
plt.xlabel('Date');
plt.savefig('./Images/3_emd.png')
```


0.01
0.00
−0.01
0
1
2
0.025
0.000
−0.025
3
0.02
0.00
−0.02
4
5
6
7
8
9
0.05
0.00
−0.05
10
11
12
1.4
1.2
13
2006
2008
2010
2012
2014
2016
2018
2020
Date

[46]:

```
nL = dfImfs.shape[1]
original = dfImfs.sum(axis=1)
denoiseSig1 = dfImfs[range(1, nL)].sum(axis=1)
denoiseSig3 = dfImfs[range(3, nL)].sum(axis=1)
dfEMD = pd.DataFrame({
    'Original Prices' : original,
    'Denoised Signal (1)' :denoiseSig1,
    'Denoised Signal (3)' :denoiseSig3
})
dfEMD.loc['2018-04-01':'2019-01-01'].plot(figsize=figsize);
plt.xlabel('Date');
plt.savefig('./Images/4_denoise.png')
```



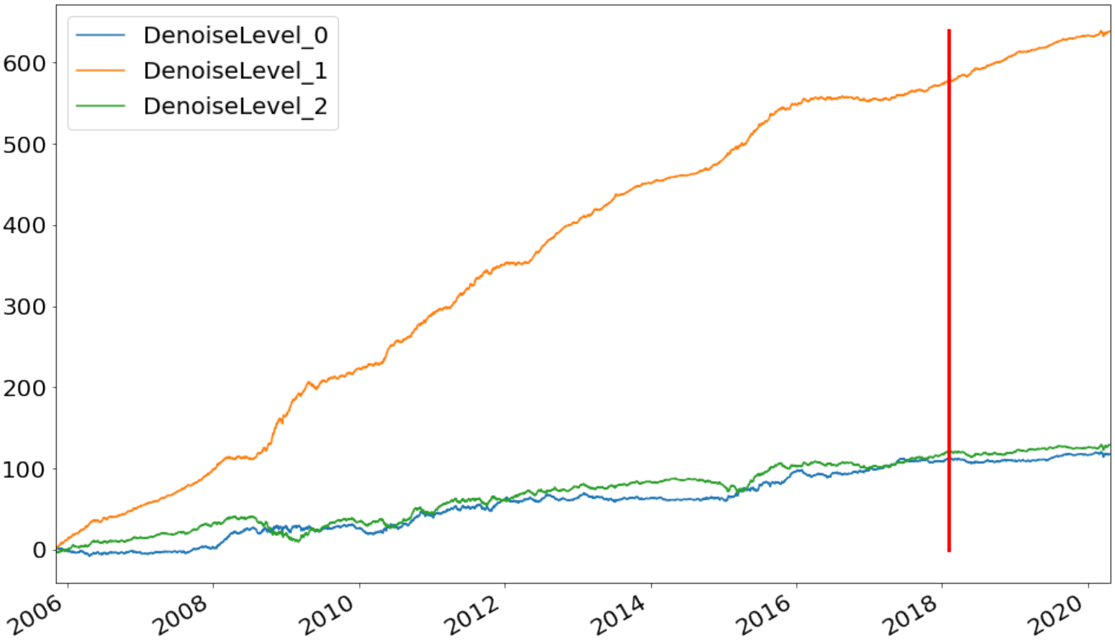
4 LSTM Analysis

4.1 Denoise Levels

[47]:

```
denoiseLevelList = [0, 1, 2]
dfDenoise = pd.DataFrame()
for denoiseLevel in denoiseLevelList:
    lstmStr = 'PCA_{}_{}_TS_{}_NCells_{}_Epochs_{}_NLayers_{}_Dropout_{}'.format(
        applyPCA, variabilityRatio, timeSteps, nCells, epochs, nLayers, dropout)
    equityFilename = '../Cashe/Results/EquityCurve_{}_EMD_{}_{}'.format(
        featuresStr, denoiseLevel, lstmStr)
    equityPickleFilename = equityFilename + '.pickle'
    ec = pd.read_pickle(equityPickleFilename)
    dfDenoise['DenoiseLevel_{}'.format(denoiseLevel)] = ec
dfDenoise.cumsum().plot(figsize=figsize);
XInSample = [pd.to_datetime(inSampleDate), pd.to_datetime(inSampleDate)]
YInSample = [0, dfDenoise.cumsum().max().max()]
plt.plot(XInSample, YInSample, '-r', linewidth=3)
plt.savefig('./Images/5_lstm_denoise.png')
srDenoise = np.sqrt(252) * dfDenoise.mean() / dfDenoise.std()
srDenoise.name = 'Sharpe Ratio'
print(pd.DataFrame(srDenoise))
```

```
                Sharpe Ratio
DenoiseLevel_0      0.760276
DenoiseLevel_1      4.262323
DenoiseLevel_2      0.834108
```

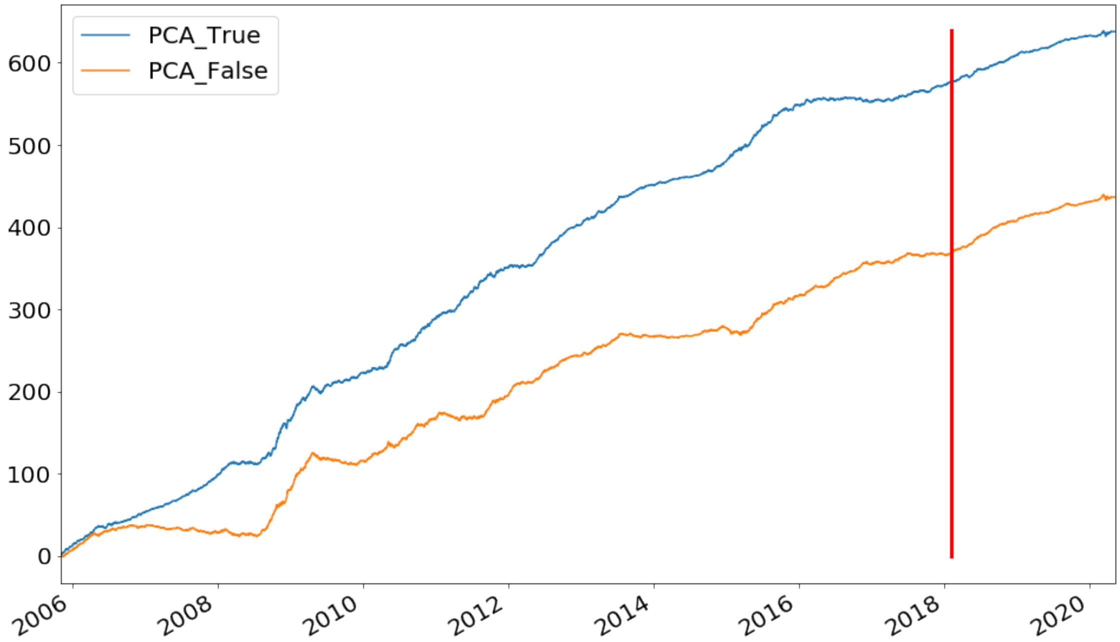


4.2 PCA

[48]:

```
denoiseLevel = 1
applyPCAList = [True, False]
dfPCA = pd.DataFrame()
for applyPCA in applyPCAList:
    lstmStr = 'PCA_{}_{}_TS_{}_NCells_{}_Epochs_{}_NLayers_{}_Dropout_{}'.format(
        applyPCA, variabilityRatio, timeSteps, nCells, epochs, nLayers, dropout)
    equityFilename = '../Cashe/Results/EquityCurve_{}_EMD_{}_{}'.format(
        featuresStr, denoiseLevel, lstmStr)
    equityPickleFilename = equityFilename + '.pickle'
    ec = pd.read_pickle(equityPickleFilename)
    dfPCA['PCA_{}'.format(applyPCA)] = ec
dfPCA.cumsum().plot(figsize=figsize);
XInSample = [pd.to_datetime(inSampleDate), pd.to_datetime(inSampleDate)]
YInSample = [0, dfPCA.cumsum().max().max()]
plt.plot(XInSample, YInSample, '-r', linewidth=3)
plt.savefig('./Images/6_lstm_pca.png')
srPCA = np.sqrt(252) * dfPCA.mean() / dfPCA.std()
srPCA.name = 'Sharpe Ratio'
print(pd.DataFrame(srPCA))
```

```
           Sharpe Ratio
PCA_True       4.256271
PCA_False      2.858343
```

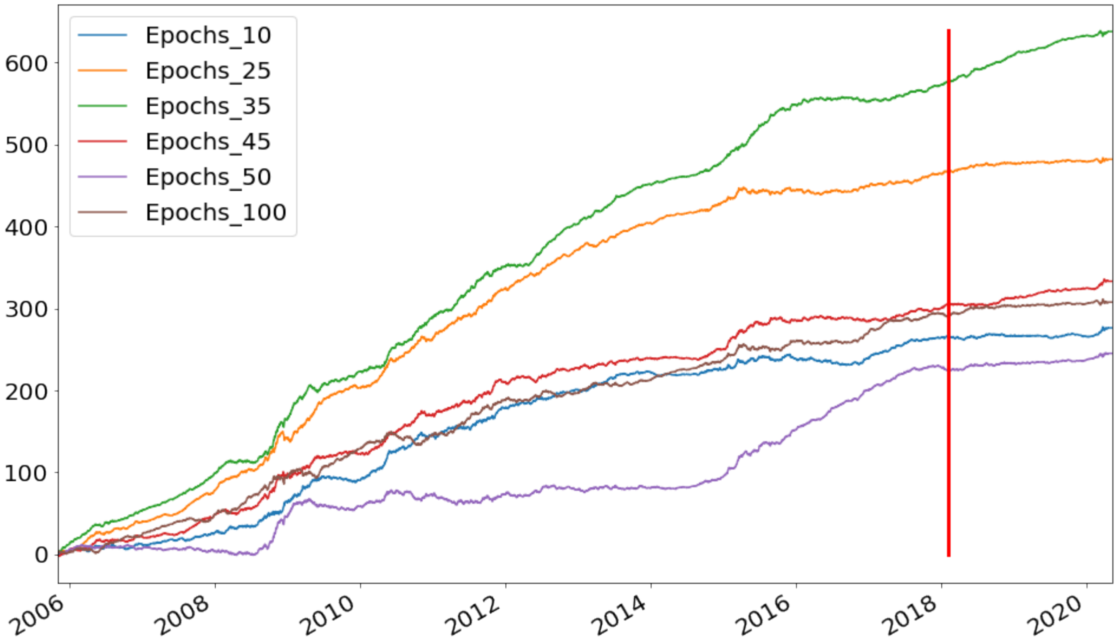


4.3 Epochs

[49]:

```
applyPCA = True
epochsList = [10, 25, 35, 45, 50, 100]
dfEpochs = pd.DataFrame()
for epochs in epochsList:
    lstmStr = 'PCA_{}_{}_TS_{}_NCells_{}_Epochs_{}_NLayers_{}_Dropout_{}'.format(
        applyPCA, variabilityRatio, timeSteps, nCells, epochs, nLayers, dropout)
    equityFilename = '../Cashe/Results/EquityCurve_{}_EMD_{}_{}'.format(
        featuresStr, denoiseLevel, lstmStr)
    equityPickleFilename = equityFilename + '.pickle'
    ec = pd.read_pickle(equityPickleFilename)
    dfEpochs['Epochs_{}'.format(epochs)] = ec
dfEpochs.cumsum().plot(figsize=figsize);
XInSample = [pd.to_datetime(inSampleDate), pd.to_datetime(inSampleDate)]
YInSample = [0, dfEpochs.cumsum().max().max()]
plt.plot(XInSample, YInSample, '-r', linewidth=3)
plt.savefig('./Images/7_lstm_epochs.png')
srEpochs = np.sqrt(252) * dfEpochs.mean() / dfEpochs.std()
srEpochs.name = 'Sharpe Ratio'
print(pd.DataFrame(srEpochs))
```

```
            Sharpe Ratio
Epochs_10       1.791631
Epochs_25       3.167233
Epochs_35       4.256271
Epochs_45       2.166894
Epochs_50       1.587519
Epochs_100      1.998332
```



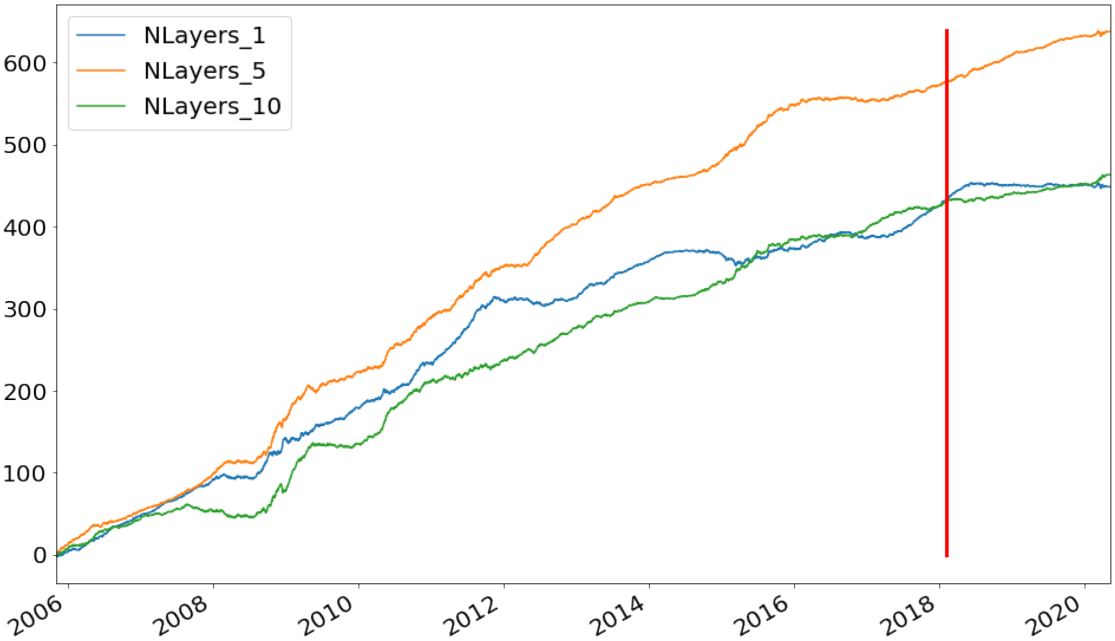
5 N Layers

[50]:

```
epochs = 35
nLayersList = [1, 5, 10]
dfNLayers = pd.DataFrame()
for nLayers in nLayersList:
    lstmStr = 'PCA_{}_{}_TS_{}_NCells_{}_Epochs_{}_NLayers_{}_Dropout_{}'.format(
        applyPCA, variabilityRatio, timeSteps, nCells, epochs, nLayers, dropout)
    equityFilename = '../Cashe/Results/EquityCurve_{}_EMD_{}_{}'.format(
        featuresStr, denoiseLevel, lstmStr)
    equityPickleFilename = equityFilename + '.pickle'
    ec = pd.read_pickle(equityPickleFilename)
    dfNLayers['NLayers_{}'.format(nLayers)] = ec
dfNLayers.cumsum().plot(figsize=figsize);
XInSample = [pd.to_datetime(inSampleDate), pd.to_datetime(inSampleDate)]
YInSample = [0, dfNLayers.cumsum().max().max()]
plt.plot(XInSample, YInSample, '-r', linewidth=3)
plt.savefig('./Images/8_lstm_nLayers.png')
srNLayers = np.sqrt(252) * dfNLayers.mean() / dfNLayers.std()
srNLayers.name = 'Sharpe Ratio'
print(pd.DataFrame(srNLayers))
```

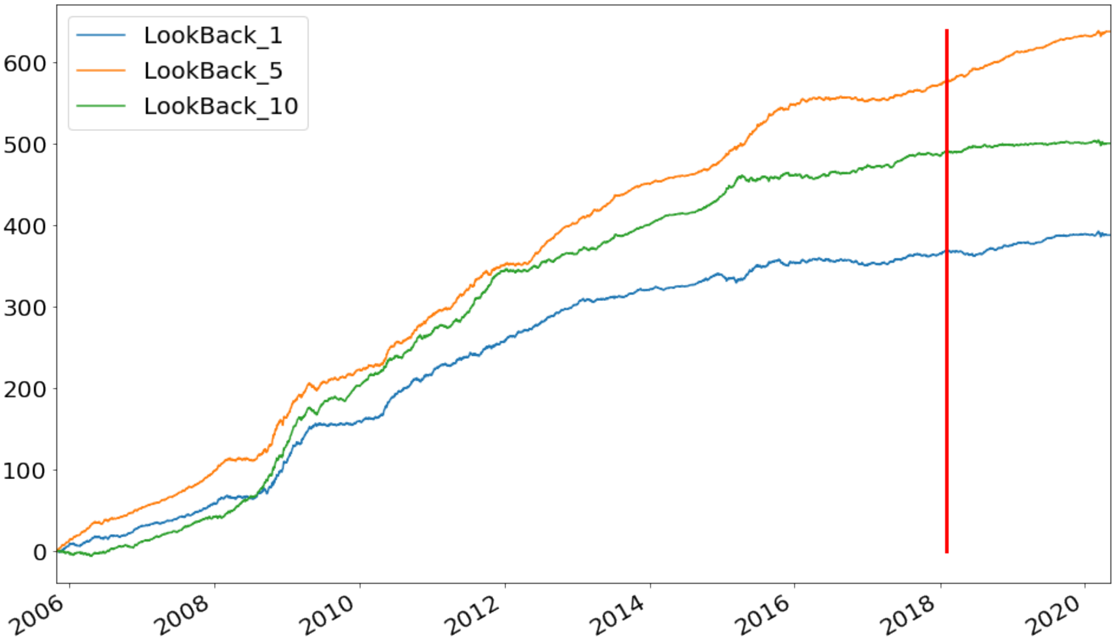
```
            Sharpe Ratio
NLayers_1       2.941109
```

```
NLayers_5       4.256271
NLayers_10      3.040046
```



[52]:
```
nLayers = 5
timeStepsList = [1, 5, 10]
dfTimeSteps = pd.DataFrame()
for timeSteps in timeStepsList:
    lstmStr = 'PCA_{}_{}_TS_{}_NCells_{}_Epochs_{}_NLayers_{}_Dropout_{}'.format(
        applyPCA, variabilityRatio, timeSteps, nCells, epochs, nLayers, dropout)
    equityFilename = '../Cashe/Results/EquityCurve_{}_EMD_{}_{}'.format(
        featuresStr, denoiseLevel, lstmStr)
    equityPickleFilename = equityFilename + '.pickle'
    ec = pd.read_pickle(equityPickleFilename)
    dfTimeSteps['LookBack_{}'.format(timeSteps)] = ec
dfTimeSteps.cumsum().plot(figsize=figsize);
XInSample = [pd.to_datetime(inSampleDate), pd.to_datetime(inSampleDate)]
YInSample = [0, dfTimeSteps.cumsum().max().max()]
plt.plot(XInSample, YInSample, '-r', linewidth=3)
plt.savefig('./Images/9_lstm_timeSteps.png')
srTimeSteps = np.sqrt(252) * dfTimeSteps.mean() / dfTimeSteps.std()
srTimeSteps.name = 'Sharpe Ratio'
print(pd.DataFrame(srTimeSteps))
```

```
             Sharpe Ratio
LookBack_1       2.531904
LookBack_5       4.256271
LookBack_10      3.300597
```


LookBack_1
LookBack_5
LookBack_10
600
500
400
300
200
100
0
2006
2008
2010
2012
2014
2016
2018
2020

```
[ ]:
```