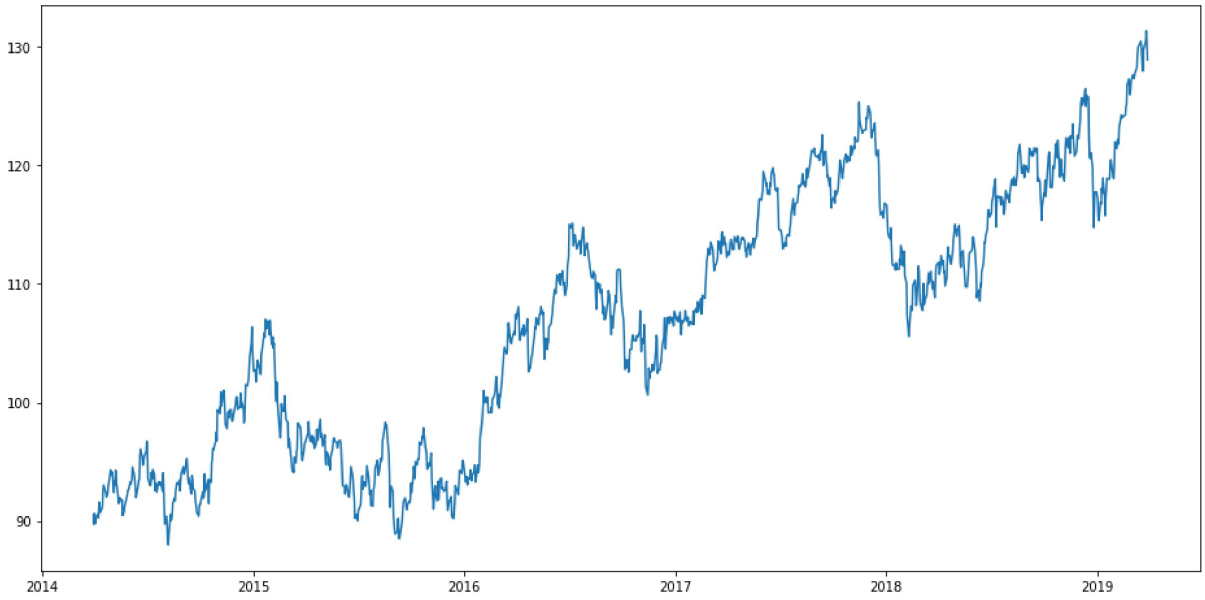
In [66]:
```
eft = "VPU"
```

In [67]:
```
#import packages
import pandas as pd
import numpy as np

#to plot within notebook
import matplotlib.pyplot as plt
%matplotlib inline

#setting figure size
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 20,10

#for normalizing data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))

#read the file
df = pd.read_csv('./ml_data/' + eft + '.csv')

#print the head
df.head()

#importing required libraries
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM
```

In [68]:
```
#setting index as date
df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')
df.index = df['Date']

#plot
plt.figure(figsize=(16,8))
plt.plot(df['Price'], label='Price history')
```

Out[68]: [<matplotlib.lines.Line2D at 0x23c60e61ba8>]



In [69]:
```
#creating dataframe
data = df.sort_index(ascending=True, axis=0)
new_data = pd.DataFrame(index=range(0,len(df)),columns=['Date', 'Price'])
for i in range(0,len(data)):
    new_data['Date'][i] = data['Date'][i]
    new_data['Price'][i] = data['Price'][i]

#setting index
new_data.index = new_data.Date
new_data.drop('Date', axis=1, inplace=True)

#creating train and test sets
dataset = new_data.values

train = dataset[0:1030,:]
valid = dataset[1030:,:]

#converting dataset into x_train and y_train
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(dataset)

x_train, y_train = [], []
for i in range(60,len(train)):
    x_train.append(scaled_data[i-60:i,0])
    y_train.append(scaled_data[i,0])
x_train, y_train = np.array(x_train), np.array(y_train)

x_train = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))

# create and fit the LSTM network
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1],1)))
model.add(LSTM(units=50))
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x_train, y_train, epochs=1, batch_size=1, verbose=2)

#predicting 246 values, using past 60 from the train data
inputs = new_data[len(new_data) - len(valid) - 60:].values
inputs = inputs.reshape(-1,1)
inputs  = scaler.transform(inputs)

X_test = []
for i in range(60,inputs.shape[0]):
    X_test.append(inputs[i-60:i,0])
X_test = np.array(X_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
closing_price = model.predict(X_test)
closing_price = scaler.inverse_transform(closing_price)
```

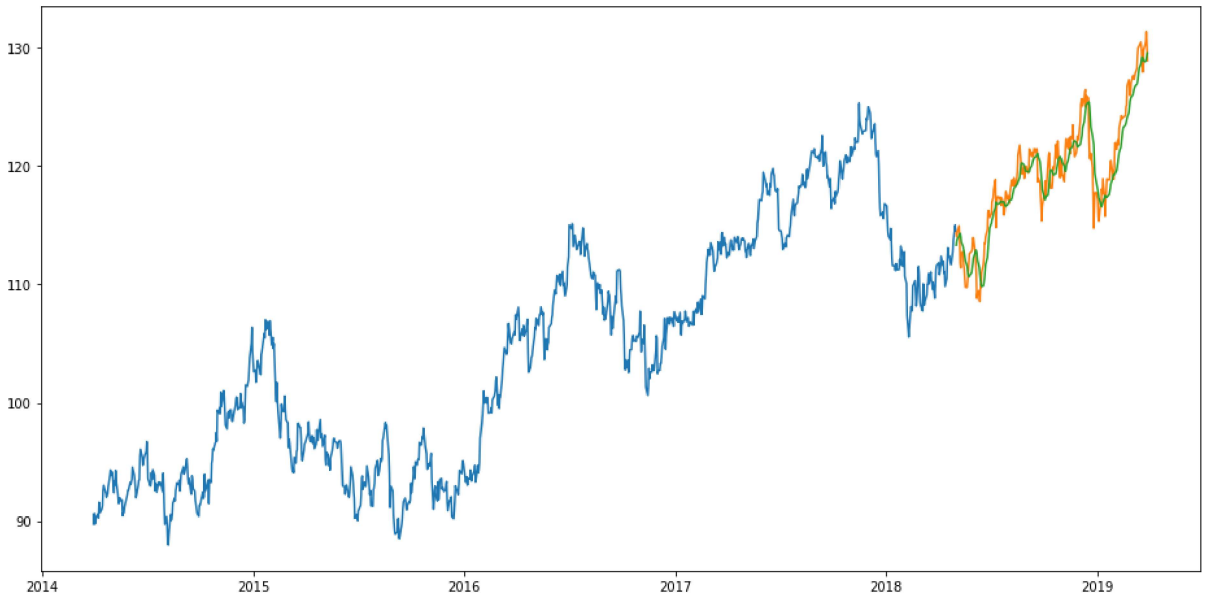
```
C:\Users\matth\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: Da
taConversionWarning: Data with input dtype object was converted to float64 by
MinMaxScaler.
  warnings.warn(msg, DataConversionWarning)
```

```
Epoch 1/1
 - 50s - loss: 0.0034
```

In [70]:
```
#for plotting
train = new_data[:1030]
valid = new_data[1030:]
valid['Predictions'] = closing_price
plt.figure(figsize=(16,8))
plt.plot(train['Price'])
plt.plot(valid[['Price','Predictions']])
```

```
C:\Users\matth\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/sta
ble/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy)
  after removing the cwd from sys.path.
```

Out[70]:
```
[<matplotlib.lines.Line2D at 0x23c68104160>,
 <matplotlib.lines.Line2D at 0x23c68104ef0>]
```



In [71]:
```
date = data['Date'][1030:]
pred = valid['Predictions']

d = pd.DataFrame(index=range(0,len(pred)),columns=['Date', 'Predictions'])
for i in range(0,len(pred)):
    d['Date'][i] = str(date[i]).split(' ')[0]
    d['Predictions'][i] = pred[i]

#print(d)
np.savetxt(eft + ".csv", d, fmt='%s', delimiter=",")
```

In [ ]:
```

```