

Snapshot of the algorithm and data collected for the RNN-LSTM with attention

Ongoing Application: High Frequency Quant Trading strategies using machine learning.

Author: Yesser Nasser, PhD

Quantitative Data Scientist – Machine Learning Engineer

Yesser.Nasser@icloud.com

Algorithm Part 1: data collection

Algorithm for data collection

```
1 #- coding: utf-8 -*-
2 """
3 Updated on Wed Aug 28 16:45:26 2019
4 @author: Yesser H. Nasser
5 Collect the data, pre-process the data to handle nan values
6 """
7
8 import re
9 import time
10 import bs4 as bs
11 import pickle
12 import requests
13 import datetime as dt
14 import pandas as pd
15 import random
16 import numpy as np
17 import pandas_datareader.data as web
18 import os
19 import matplotlib.pyplot as plt
20 import matplotlib.dates as mdates
21 import matplotlib.ticker as mticker
22
23 from sklearn import preprocessing
24 from mpl_finance import candlestick_ohlc
25 from matplotlib import style
26 from alpha_vantage.timeseries import TimeSeries
27 from sklearn import preprocessing
28 from collections import defaultdict
29 from datetime import datetime
30
31 style.use('f5')
32
33 start = time.time()
34
35 # =====
36 SEQ_LE = 1
37 TARGET = 1
38 PERIOD = 'D'
39 c = 0.
40
41 # =====
42 weeks = 1, # weeks of data collections
43 atts = ['close', 'volume', 'high', 'low'], # calculate indicators
44 # =====
45 def save_tickers(tickers, f):
46     re = re.compile(r'([a-zA-Z0-9_]+)')
47     so = defaultdict(list)
48     ta = {}
49     ti = {}
50     fo = {}
51
52     for ticker in tickers:
53         if 'GL' in ticker or 'IEX' in ticker:
54             continue
55         with open(f + ticker + '.pkl', 'wb') as f:
56             pickle.dump(tickers, f)
57             print(tickers)
58             return tickers
59 tickers = save_tickers(tickers, f)
```

Name	Type	Size	Value
X_train	float64	(2534, 15, 1298)	[[[6.55484291e-02 2.63960000e-01 8.78285995e+01 ... -1.34856543e-01 ...
X_validation	float64	(310, 15, 1298)	[[[3.00990674e-02 9.96554343e-02 -1.90386226e+00 ... 4.55702883e-02 ...
accuracy	list	20	[0.5714285845361833, 0.694554076874783, 0.7131018274784935, 0.74901342 ...
all_tickers	list	507	['MMM', 'ABT', 'ABBV', 'ABMD', 'ACN', 'ATVI', 'ADBE', 'AMD', 'AAP', 'A ...
atts	list	4	['close', 'volume', 'high', 'low']
batch_size	int	1	5
c	float	1	0.7
count	int	1	2
df	list	7	[Dataframe, Dataframe, Dataframe, Dataframe, Dataframe, Dataframe, Dat ...
elapsed_time_secs	float	1	1021.2800447940826
epochs	int	1	20
i	int	1	320
last_20pct	str	1	2019-08-19 14:22:00
loss	list	20	[0.7363361349680712, 0.5845534333807799, 0.561707640956167, 0.51201543 ...
main_data	DataFrame	(5516, 1300)	Column names: AAPL_Closes, AAPL_VWAP_diff, AAPL_Volumes, ACN_Closes, A ...
main_data_train	DataFrame	(3862, 1300)	Column names: AAPL_Closes, AAPL_VWAP_diff, AAPL_Volumes, ACN_Closes, A ...

IPython console

Console 1/A

```
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 4 Column: 16 Memory: 48 %

- Compiling data
- Building Dataframes
 - Sequences

Name	Type	Size	Value
X_train	float64	(2534, 15, 1298)	[[[6.55484291e-02 2.63960000e-01 8.78285995e+01 ... -1.34856543e-01 ...
X_validation	float64	(310, 15, 1298)	[[[3.00990674e-02 9.96554343e-02 -1.90386226e+00 ... 4.55702883e-02 ...
accuracy	list	20	[0.5714285845361833, 0.694554076874783, 0.7131018274784935, 0.74901342 ...
all_tickers	list	507	['MMM', 'ABT', 'ABBV', 'ABMD', 'ACN', 'ATVI', 'ADBE', 'AMD', 'AAP', 'A ...
atts	list	4	['close', 'volume', 'high', 'low']
batch_size	int	1	5
c	float	1	0.7
count	int	1	2
df	list	7	[Dataframe, Dataframe, Dataframe, Dataframe, Dataframe, Dataframe, Dat ...
elapsed_time_secs	float	1	1021.2800447940826
epochs	int	1	20
i	int	1	320
last_20pct	str	1	2019-08-19 14:22:00
loss	list	20	[0.7363361349680712, 0.5845534333807799, 0.561707640956167, 0.51201543 ...
main_data	DataFrame	(5516, 1300)	Column names: AAPL_Closes, AAPL_VWAP_diff, AAPL_Volumes, ACN_Closes, A ...
main_data_train	DataFrame	(3862, 1300)	Column names: AAPL_Closes, AAPL_VWAP_diff, AAPL_Volumes, ACN_Closes, A ...

IPython console

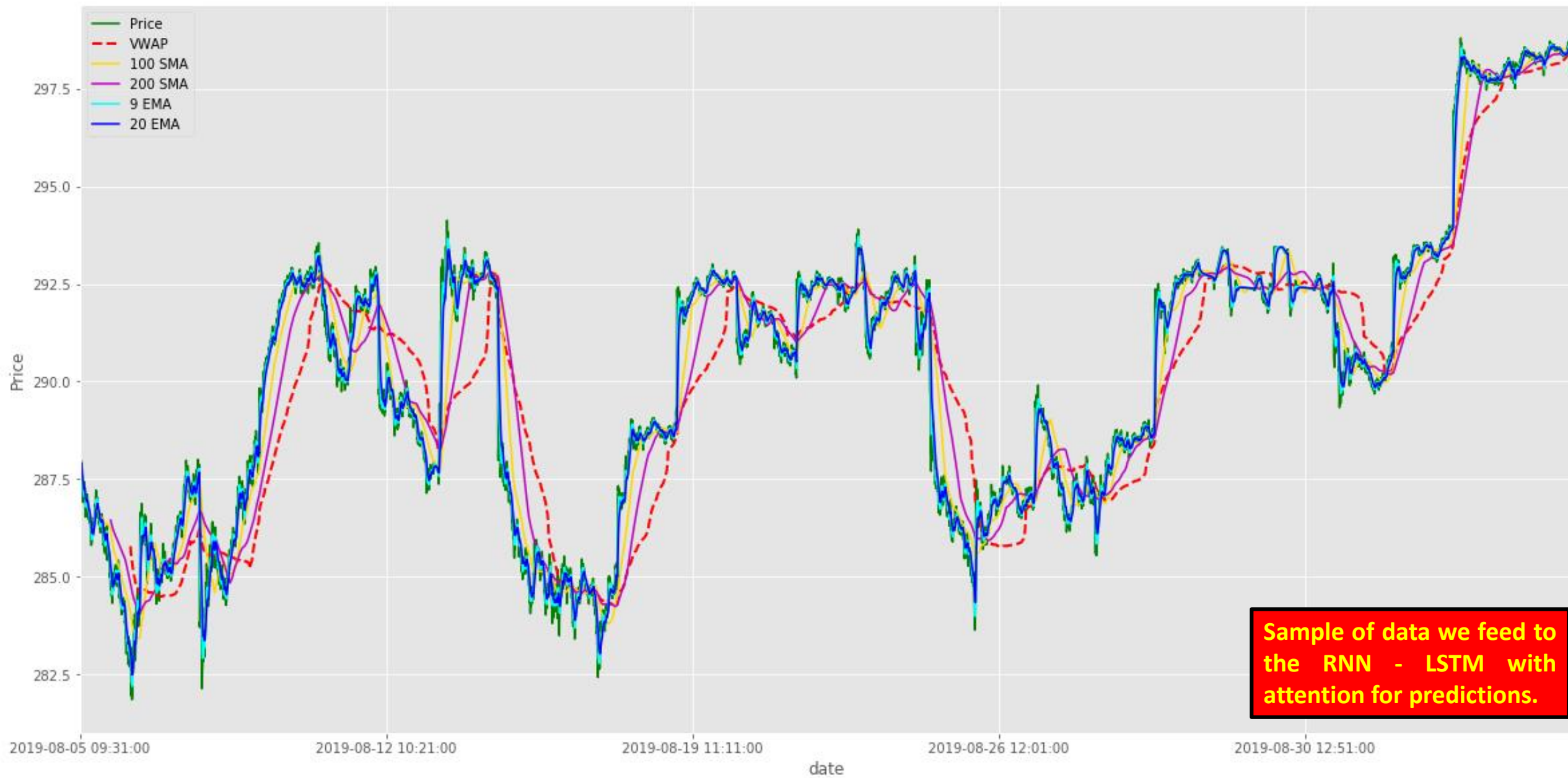
Console 1/A

460								
470								
480								
490								
500								
		MMW_low	ABT_low	ABBV_low	...	GLD_low	SLV_low	USD_low
date					...			
2019-08-12 09:31:00	163.02	86.11	65.20	...	141.7165	15.8500	39.850000	
2019-08-12 09:32:00	163.00	86.11	65.26	...	141.6550	15.8450	40.037300	
2019-08-12 09:33:00	162.68	86.21	65.24	...	141.6600	15.8499	39.965475	
2019-08-12 09:34:00	162.25	86.24	65.23	...	141.7203	15.8501	39.893650	
2019-08-12 09:35:00	162.60	86.24	65.08	...	141.7495	15.8500	39.821825	

```
[5 rows x 507 columns]
```

0
10
20
30
40
50

SPY



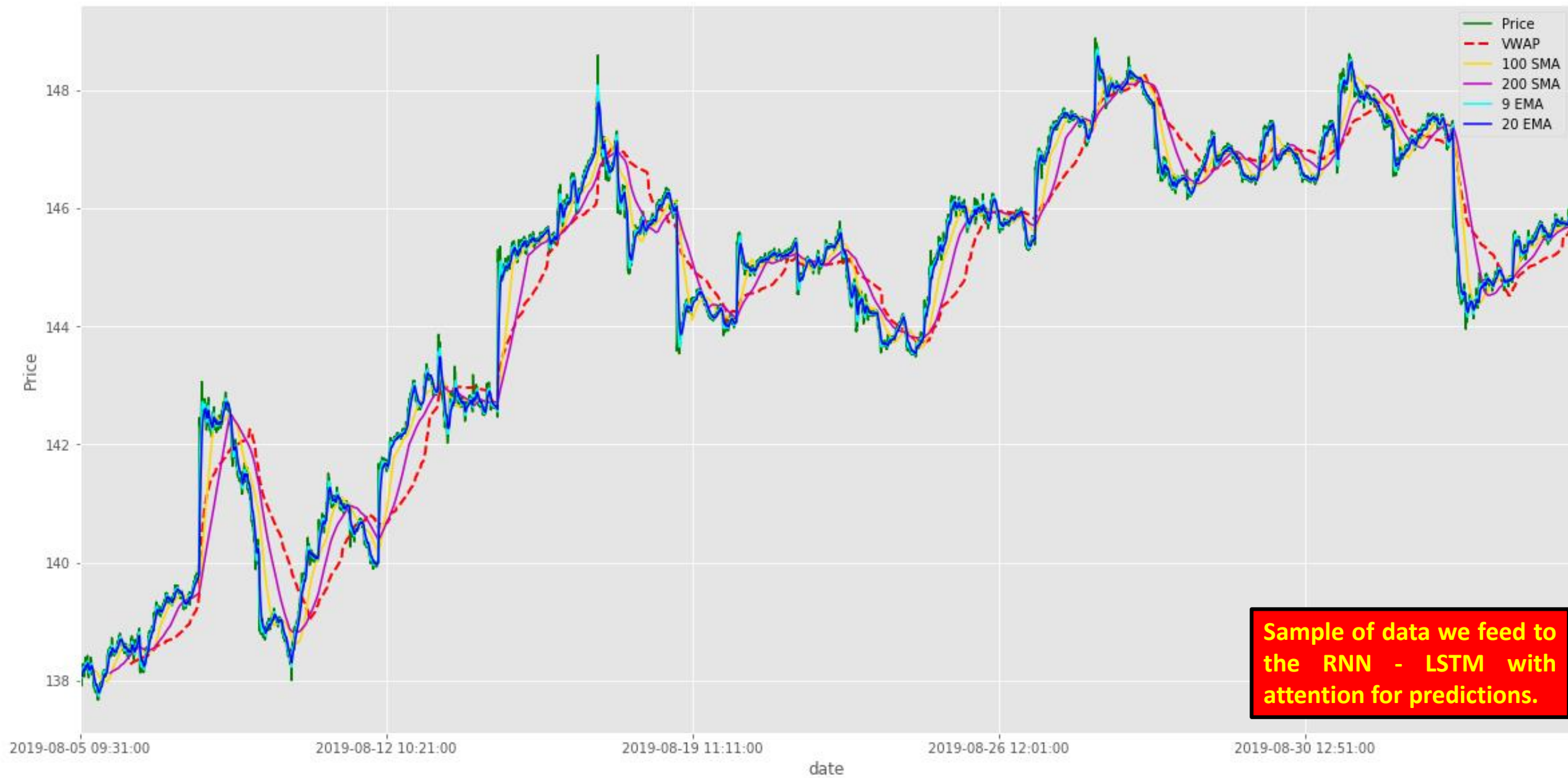
Sample of data we feed to the RNN - LSTM with attention for predictions.

9/7/2019

N.B: Time frame : 1 min

Very resent data!

TLT



Sample of data we feed to the RNN - LSTM with attention for predictions.

9/7/2019

N.B: Time frame : 1 min

GLD

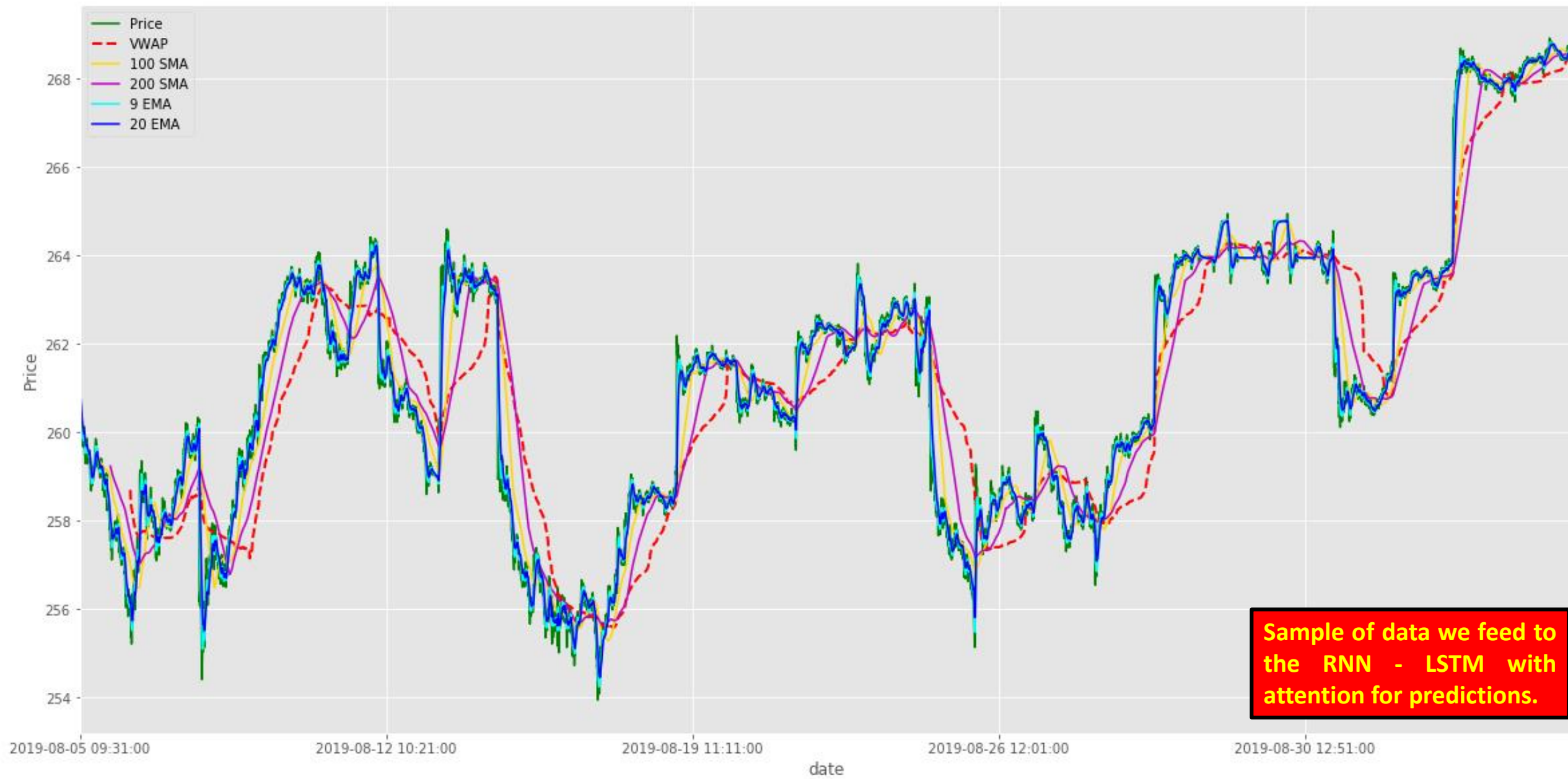


Sample of data we feed to the RNN - LSTM with attention for predictions.

9/7/2019

N.B: Time frame : 1 min

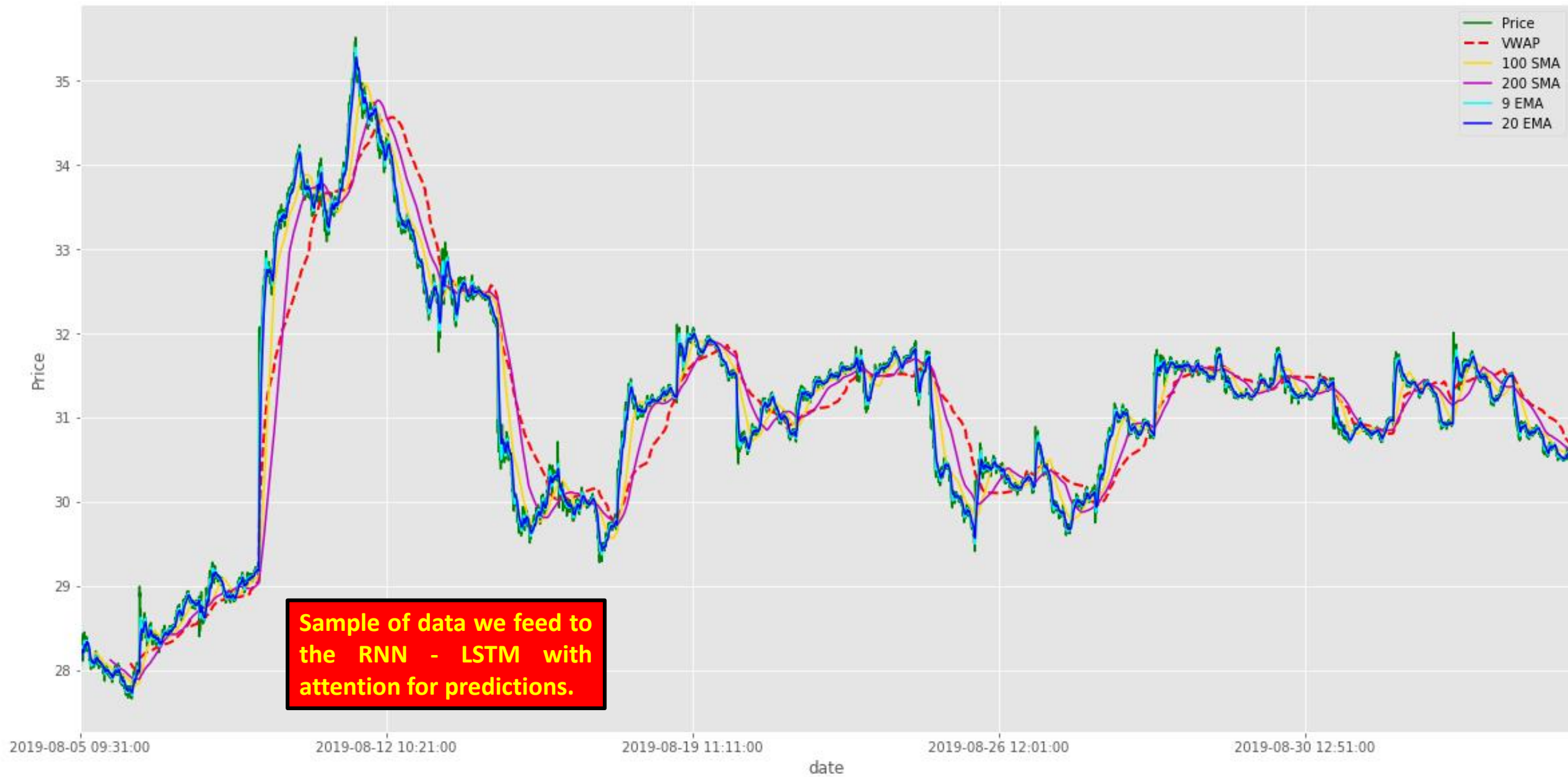
DIA



9/7/2019

N.B: Time frame : 1 min

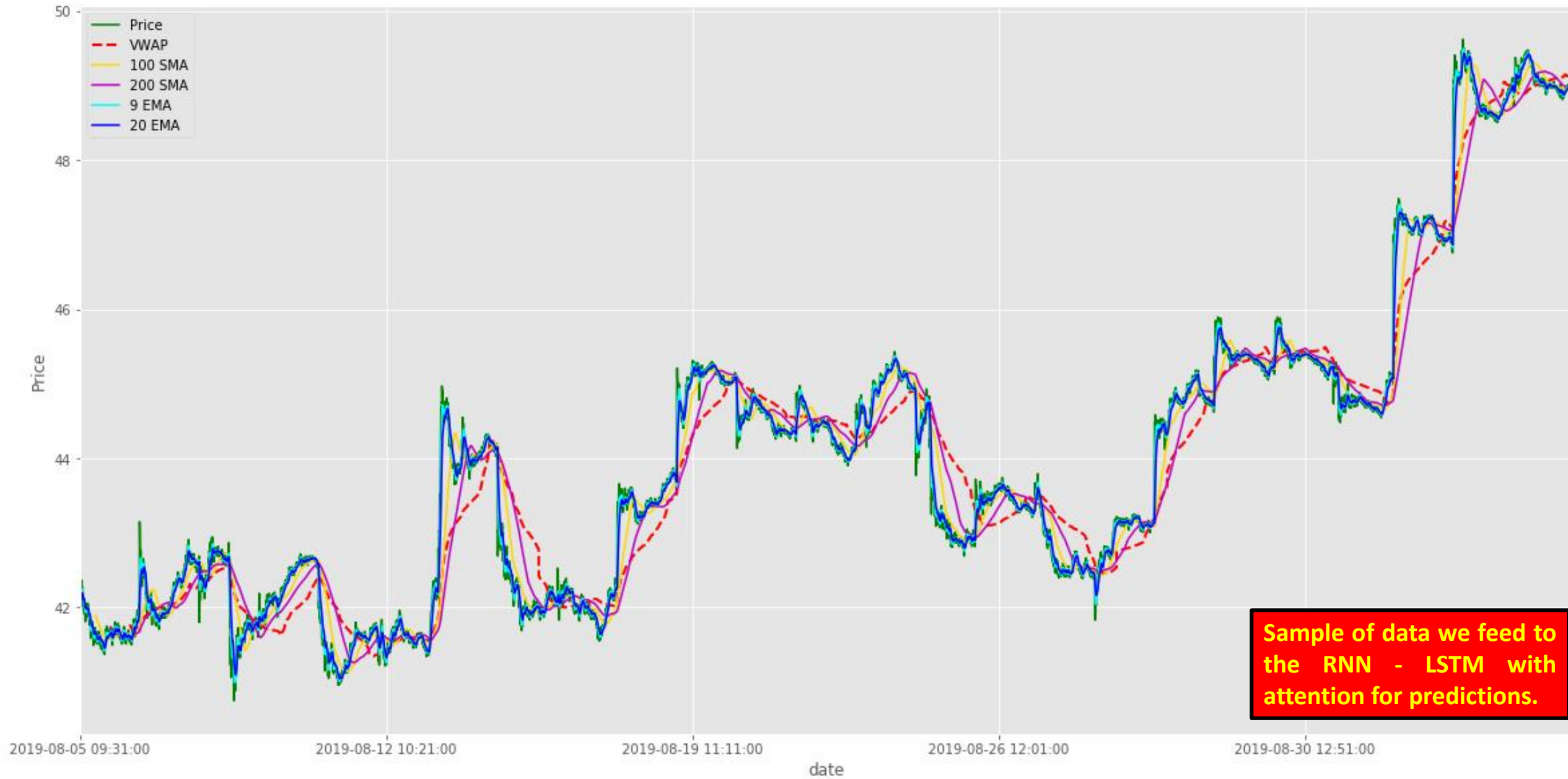
Stock Price - AMD



9/7/2019

N.B: Time frame : 1 min

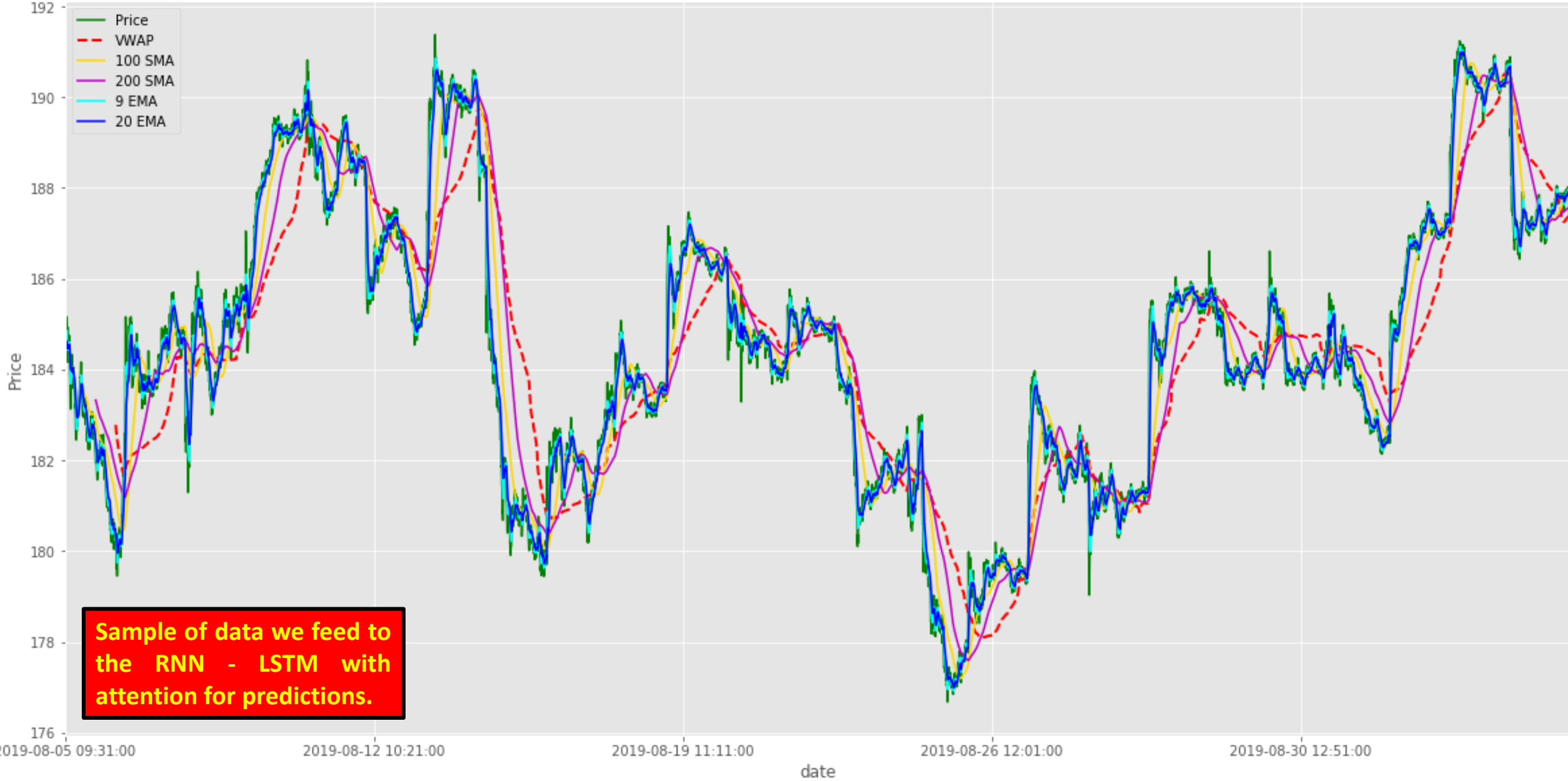
Stock Price - MU



9/7/2019

N.B: Time frame : 1 min

Stock Price - FB

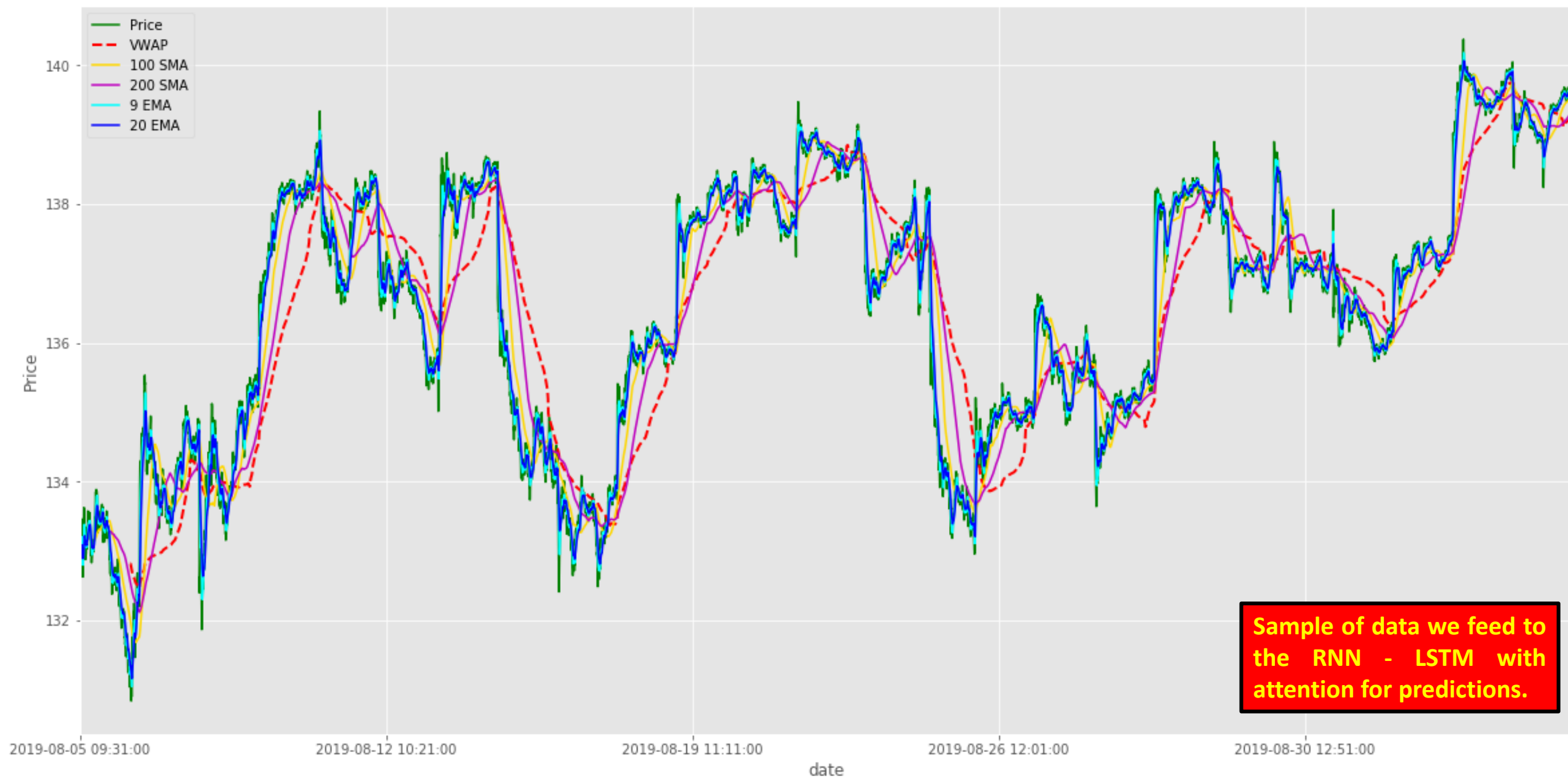


Sample of data we feed to the RNN - LSTM with attention for predictions.

9/7/2019

N.B: Time frame : 1 min

Stock Price - MSFT

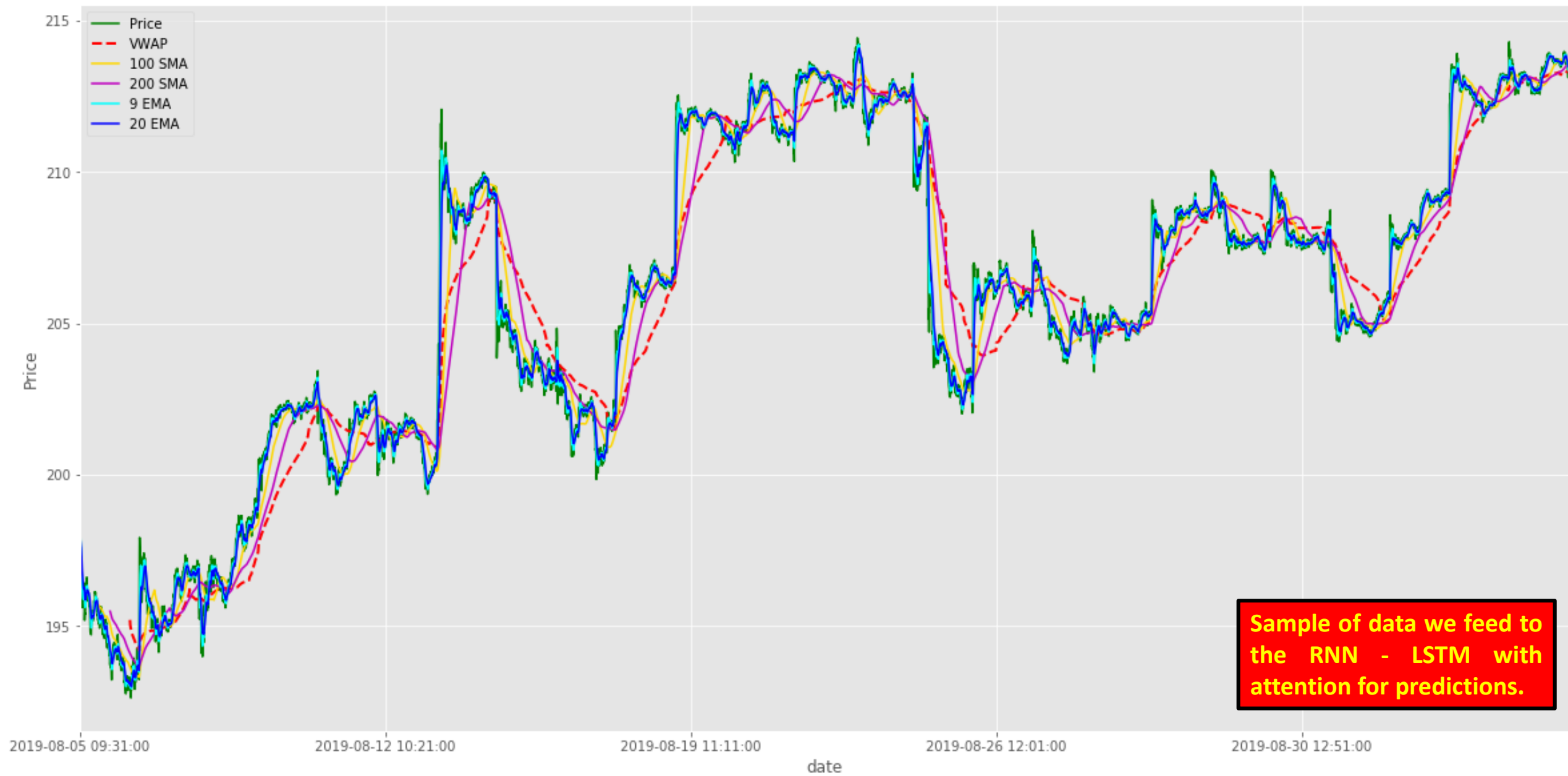


Sample of data we feed to
the RNN - LSTM with
attention for predictions.

9/7/2019

N.B: Time frame : 1 min

Stock Price - AAPL



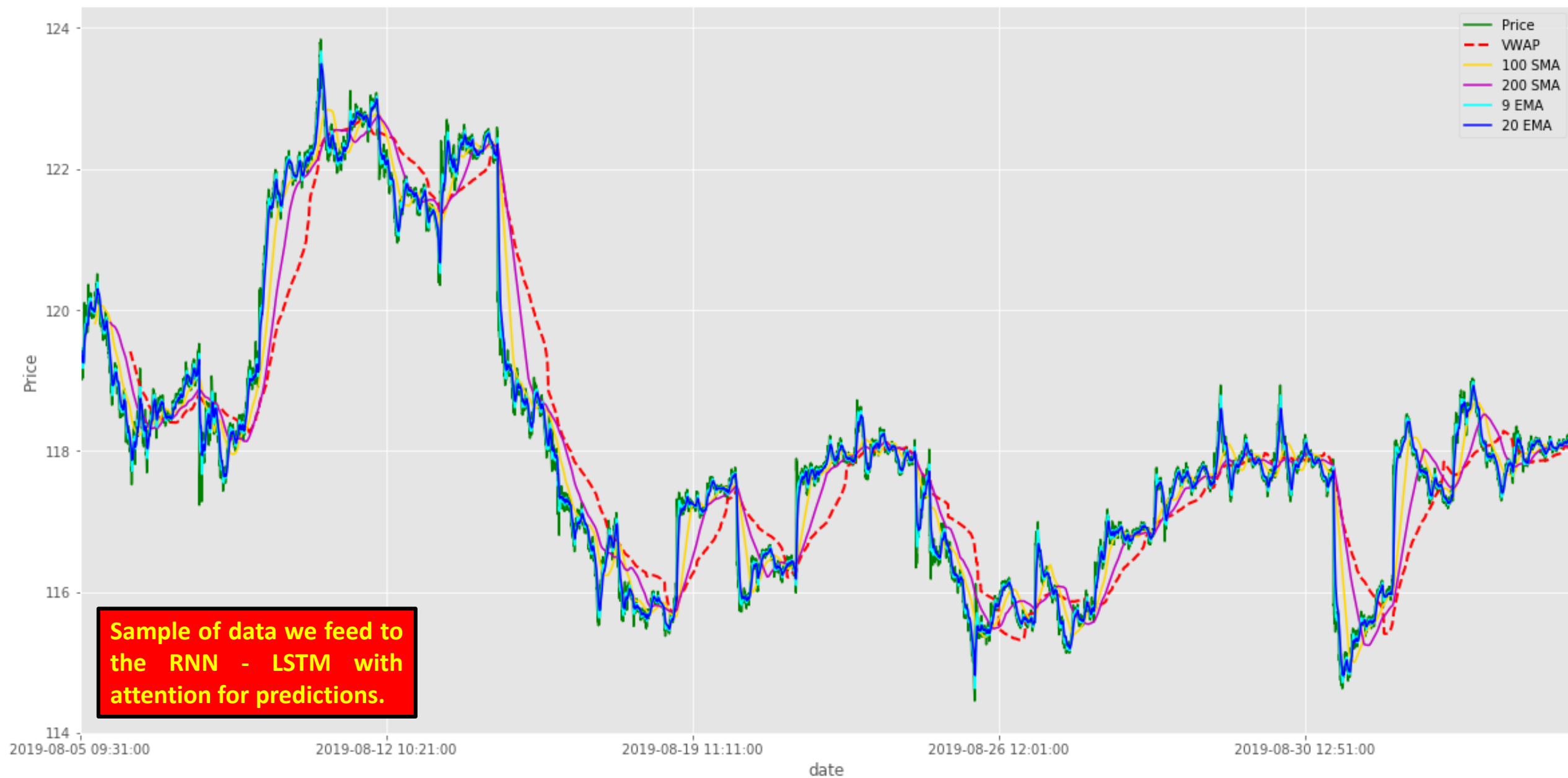
Sample of data we feed to
the RNN - LSTM with
attention for predictions.

9/7/2019

N.B: Time frame : 1 min

12

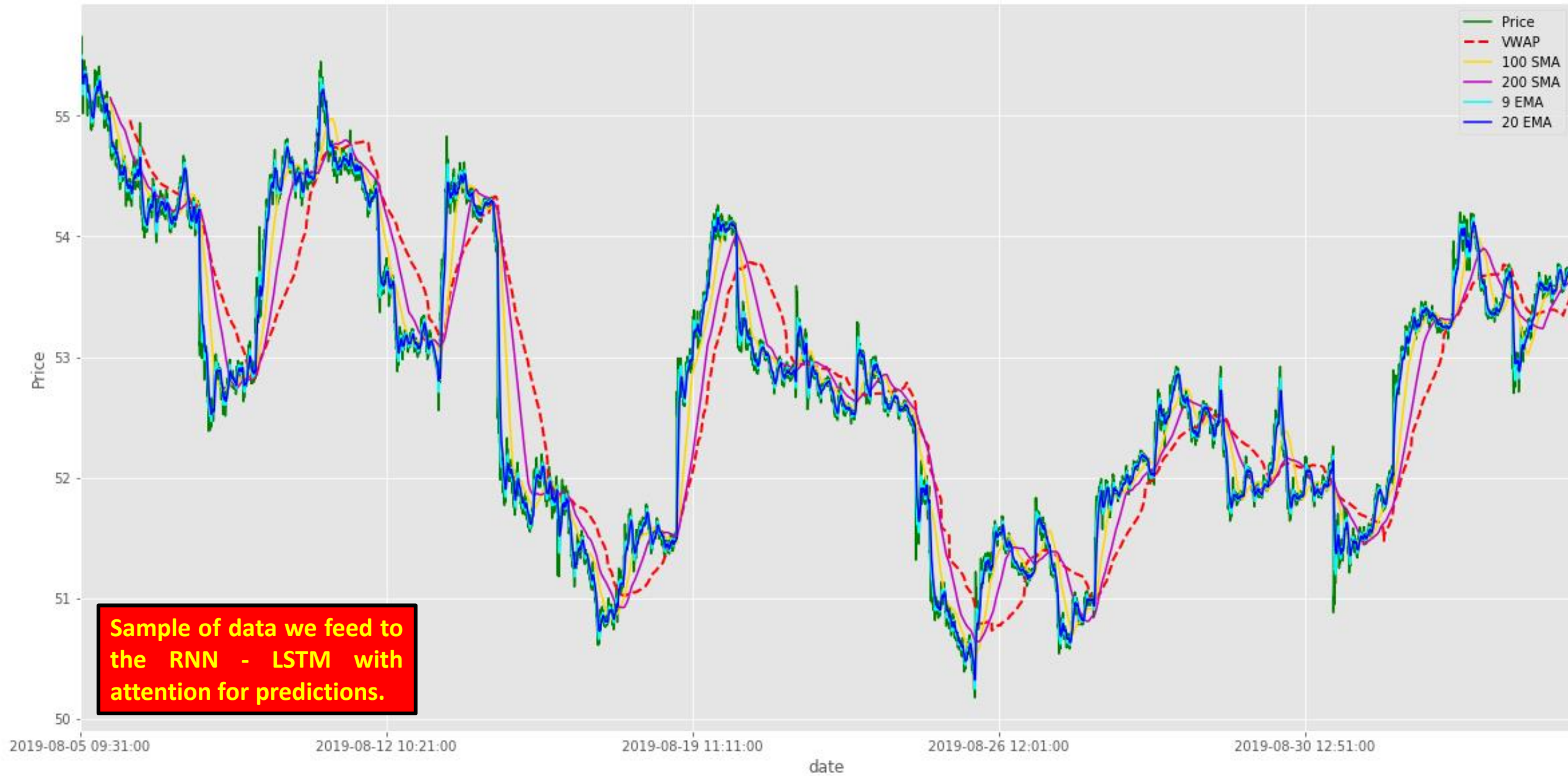
Stock Price - CVX



9/7/2019

N.B: Time frame : 1 min

Stock Price - COP

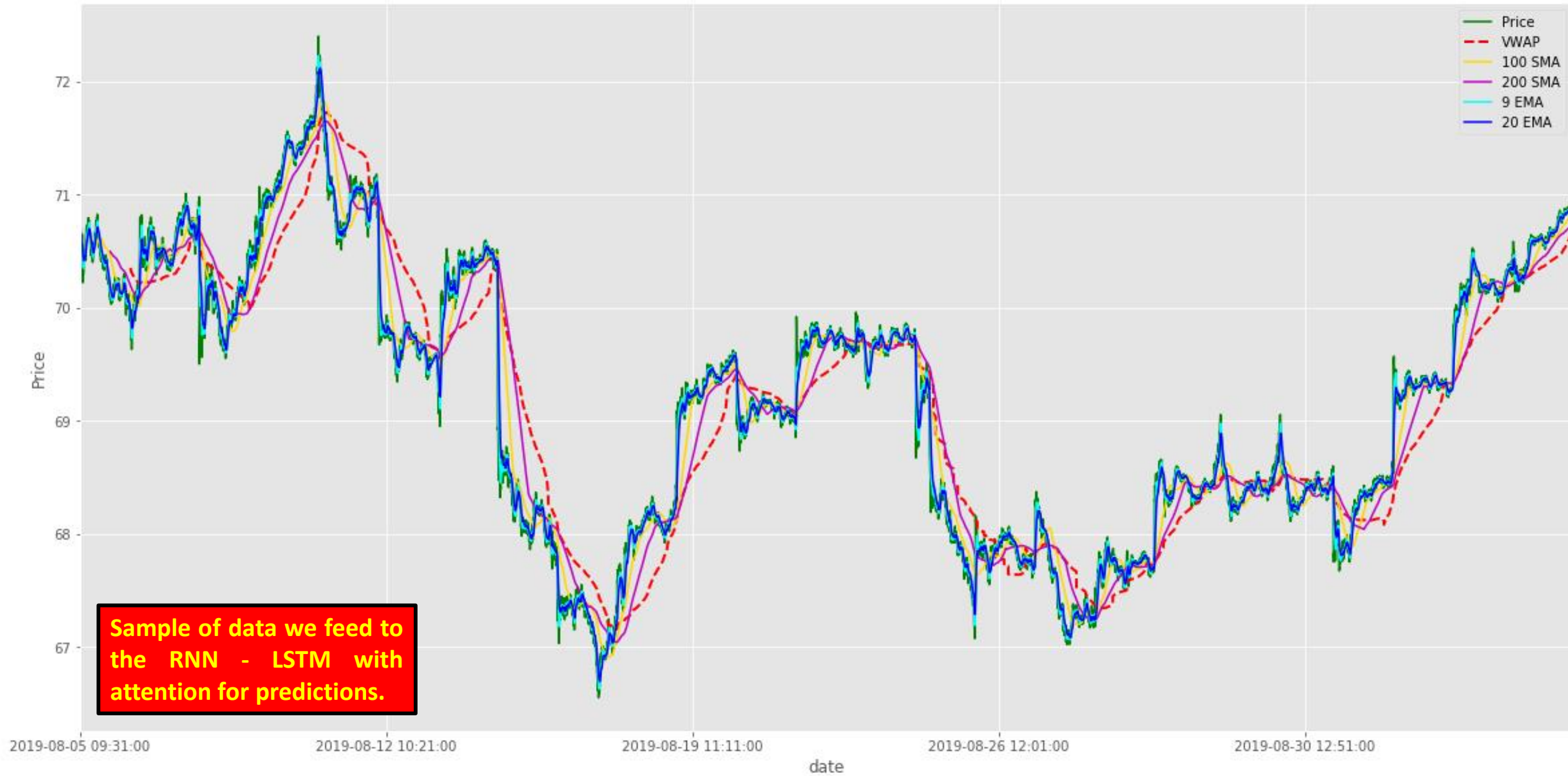


Sample of data we feed to the RNN - LSTM with attention for predictions.

9/7/2019

N.B: Time frame : 1 min

Stock Price - XOM



9/7/2019

N.B: Time frame : 1 min

Algorithm Part 2: Data preprocessing

Algorithm for:

- Data processing;
- Scaling;
- Balancing;
- Shifting;
- Shuffling;
- ...
- ...

Name	Type	Size	Value
X_train	float64	(2534, 15, 1298)	[[[6.55484291e-02 2.63960000e-01 8.78285995e+01 ... -1.34856543e-01 ...
X_validation	float64	(310, 15, 1298)	[[[3.00990674e-02 9.96554343e-02 -1.90386226e+00 ... 4.55702883e-02 ...
accuracy	list	20	[0.5714285845361833, 0.694554076874783, 0.7131018274784935, 0.74901342 ...
all_tickers	list	507	['MMM', 'ABT', 'ABBV', 'ABMD', 'ACN', 'ATVI', 'ADBE', 'AMD', 'AAP', 'A ...
atts	list	4	['close', 'volume', 'high', 'low']
batch_size	int	1	5
c	float	1	0.7
count	int	1	2
df	list	7	[Dataframe, Dataframe, Dataframe, Dataframe, Dataframe, Dataframe, Dat ...
elapsed_time_secs	float	1	1021.2800447940826
epochs	int	1	20
i	int	1	320
last_20pct	str	1	2019-08-19 14:22:00
loss	list	20	[0.7363361349680712, 0.5845534333807799, 0.561707640956167, 0.51201543 ...
main_data	DataFrame	(5516, 1300)	Column names: AAPL_Closes, AAPL_VWAP_diff, AAPL_Volumes, ACN_Closes, A ...
main_data_train	DataFrame	(3862, 1300)	Column names: AAPL_Closes, AAPL_VWAP_diff, AAPL_Volumes, ACN_Closes, A ...

Variable explorer History log

IPython console

Console 1/A

```
460
470
480
490
500
date      MMM_low  ABT_low  ABBV_low  ...  GLD_low  SLV_low  USD_low
2019-08-12 09:31:00  163.02   86.11   65.20  ...  141.7165  15.8500  39.850000
2019-08-12 09:32:00  163.00   86.11   65.26  ...  141.6550  15.8450  40.037300
2019-08-12 09:33:00  162.68   86.21   65.24  ...  141.6600  15.8499  39.965475
2019-08-12 09:34:00  162.25   86.24   65.23  ...  141.7203  15.8501  39.893650
2019-08-12 09:35:00  162.60   86.24   65.08  ...  141.7495  15.8500  39.821825
```

[5 rows x 507 columns]

```
0
10
20
30
40
50
```

IPython console File explorer Help

Algorithm Part 2: Model building

Model building:

- LSTM
- Attention Model

```
324 Spl_Mov_Avr_200_diff,
325 Exp_Mov_Avr_9_diff,
326 Exp_Mov_Avr_20_diff]
327
328 main_data = pd.concat(df, axis=1)
329 main_data = main_data[sorted(main_data.columns)]
330
331 # ===== define future (shift data) =====
332 main_data['future'] = main_data[f'{TARGET_TO_PREDICT}_Closes'].shift(-PERIOD_TO_PREDICT)
333 main_data['target'] = list(map(classify, main_data[f'{TARGET_TO_PREDICT}_Closes'], main_data['future']))
334
335 # this jsut
336 main_data
337
338
339
340 main_data
341
342 # =====
343 times =
344 last_20p
345
346 main_data
347 main_data
348
349 X_train,
350 X_valida
351
352 print(f'
353 print(f'
354 print(f'
355
356 # =====
357 import k
358 import t
359 from ker
360 from ker
361 from att
362
363 model =
364 model.ad
365 model.ad
366 model.ad
367 #
368 model.ad
369 model.ad
370 model.ad
371 #
372 model.ad
373 model.ad
374 model.ad
375 model.ad
376
377 model.add(D
378 model.add(Dropout(0.1))
379 model.add(Dense(2, activation = 'softmax'))
380
381 model.summary()
```

Name	Type	Size	Value
main_data_train	DataFrame	(3862, 1300)	Column names: AAPL_Closes, AAPL_VWAP_diff, AAPL_Volumes, ACN_Closes, A ...
main_data_validation	DataFrame	(1654, 1300)	Column names: AAPL_Closes, AAPL_VWAP_diff, AAPL_Volumes, ACN_Closes, A ...
msg	str	1	Execution took: 0:17:01 secs (Wall clock time)
save_dir	str	1	saved_models
save_fname	str	1	saved_models\28082019-162523-e20.h5
start_time	float	1	1567033702.9244797
symbol	str	1	USD
symbols_1	list	8	['SPY', 'IWM', 'DIA', 'IEF', 'TLT', 'GLD', 'SLV', 'USD']
target_corr	Series	(507,)	Series object of pandas.core.series module
tckr	str	1	AAPL
tickers	list	507	['MMM', 'ABT', 'ABBV', 'ABMD', 'ACN', 'ATVI', 'ADBE', 'AMD', 'AAP', 'A ...
tickers_corr	list	186	['AAPL', 'RMD', 'CMCSA', 'FAST', 'CE', 'LRCX', 'T', 'WDC', 'VMC', 'STZ ...
tickers_to_drop	list	321	['MAS', 'JNPR', 'DD', 'RJF', 'DXC', 'NEM', 'SLV', 'AKAM', 'MPC', 'PPL' ...
times	list	5516	['2019-08-05 15:00:00', '2019-08-05 15:01:00', '2019-08-05 15:02:00', ...
tokeep	str	1	open
training_epochs	int	1	20

Variable explorer | History log

IPython console

Layer (type)	Output shape	Param #
lstm_1 (LSTM)	(None, 15, 128)	730624
dropout_1 (Dropout)	(None, 15, 128)	0
batch_normalization_1 (Batch Normalization)	(None, 15, 128)	512
lstm_2 (LSTM)	(None, 15, 128)	131584
dropout_2 (Dropout)	(None, 15, 128)	0
batch_normalization_2 (Batch Normalization)	(None, 15, 128)	512
lstm_3 (LSTM)	(None, 15, 128)	131584
dropout_3 (Dropout)	(None, 15, 128)	0
batch_normalization_3 (Batch Normalization)	(None, 15, 128)	512
attention_1 (Attention)	(None, 128)	143

IPython console | File explorer | Help

9/7/2019

17

Permissions: RW | End-of-lines: CRLF | Encoding: UTF-8 | Line: 1 | Column: 1 | Memory: 46 %

Algorithm Part 3: Running Model – preliminary results

To learn more about the progress on this project and looking to apply this workflow please get in touch at: Yesser.Nasser@icloud.com

