In [1]:
```
%config InlineBackend.figure_format = 'retina'
```

Individual predictor training - Horizon: 5 datapoints

This Jupyter notebook is the first out of three notebooks that are used to train the individual predictors to benchmark the system performance. In total, 15 individual predictors will be trained on stock and index data. For the benchmarking process, predictors from the predictorsI.py and predictorsII.py are trained. Each predictor will be served with an input batch that is used to dertmine the forecast estimation. The input size will be set at 20 data points (20 trading days). Furthermore, 3 forecasting horizions are considered: 5, 30 and 60 datapoints into the future.

In [2]:
```
%run ../tools/dataloader.py
%run ../tools/predictorsI.py
%run ../tools/predictorsIII.py
```

Dataset

The first dataset used is the stock price of Ford Motor Company (F). Prices are in USD and listed on NYSE - Nasdaq. The data is extracted via the Yahoo Finance API accessed via the pandas data reader function. The adjusting closing price was used to train the following predictors.

Link to website: https://uk.finance.yahoo.com/quote/F/history?p=F

In [3]:
```
data = DataLoader('F', '2010-01-01', '2018-01-01')
prices = data.get_adjclose()
```

In [4]:
```
print(data)
```

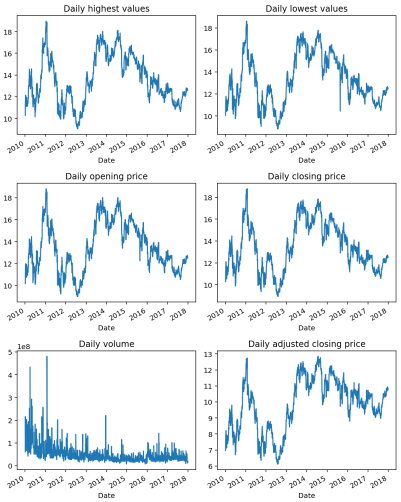
```
Total count of rows: 2013
Ticker: F
Start: 2010-01-01
End: 2018-01-01
```

In [5]:
```
data.statistics()
```

Out[5]:

	High	Low	Open	Close	Volume	Adj Close
count	2013.000000	2013.000000	2013.000000	2013.000000	2.013000e+03	2013.000000
mean	13.614367	13.325782	13.482623	13.471868	4.781743e+07	9.834059
std	2.189850	2.183796	2.189163	2.185088	3.550829e+07	1.612406
min	9.030000	8.820000	8.990000	8.920000	7.128800e+06	6.123986
25%	11.910000	11.600000	11.790000	11.760000	2.745320e+07	8.703906
50%	13.280000	13.000000	13.140000	13.150000	3.782080e+07	10.073213
75%	15.450000	15.170000	15.300000	15.280000	5.442480e+07	11.056963
max	18.969999	18.610001	18.809999	18.790001	4.808795e+08	12.847414

In [6]:
```
data.plotting_grid()
```



In [7]:
```
predictor1 = BasicUnivariatePredictor(20, 5, prices)
predictor1.create_lstm()
predictor1.model_blueprint()
```

```
Model: "sequential"
```

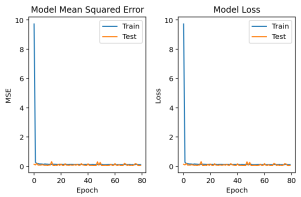
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 20, 40)	6720
lstm_1 (LSTM)	(None, 20, 50)	18200
lstm_2 (LSTM)	(None, 50)	20200
dense (Dense)	(None, 5)	255

```
Total params: 45,375
Trainable params: 45,375
Non-trainable params: 0
```

In [8]:
```
predictor1.fit_model(80, 0)
```


Out[8]:
```
<tensorflow.python.keras.callbacks.History at 0x2c1daa4ae50>
```

In [9]:
```
predictor1.show_performance()
```



In [10]:
```
predictor1.save_model()
```

```
WARNING:tensorflow:From C:\Users\Max\Anaconda3\envs\sys\lib\site-packages\tensorflow\python\training\tracking\tracking.py:111: Model.state_updates (from tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
WARNING:tensorflow:From C:\Users\Max\Anaconda3\envs\sys\lib\site-packages\tensorflow\python\training\tracking\tracking.py:111: Layer.updates (from tensorflow.python.keras.engine.base_layer) is deprecated and will be removed in a future version.
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
INFO:tensorflow:Assets written to: C:\Users\Max\Documents\GitHubPrivate\arguing-predictors\notebooks\assets
```

In [12]:
```
predictor2 = BasicUnivariatePredictor(20, 5, prices)
predictor2.create_mlp()
predictor2.model_blueprint()
```

```
Model: "sequential_2"
```

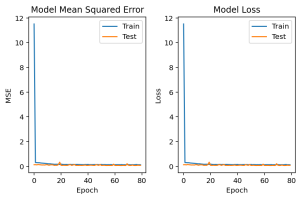
Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 50)	1050
dense_6 (Dense)	(None, 25)	1275
dense_7 (Dense)	(None, 25)	650
dense_8 (Dense)	(None, 5)	130

```
Total params: 3,105
Trainable params: 3,105
Non-trainable params: 0
```

In [13]:
```
predictor2.fit_model(80, 0)
```

Out[13]:
```
<tensorflow.python.keras.callbacks.History at 0x2c1e787bb80>
```

In [14]:
```
predictor2.show_performance()
```



In [15]:
```
predictor2.save_model()
```

```
INFO:tensorflow:Assets written to: C:\Users\Max\Documents\GitHubPrivate\arguing-predictors\notebooks\assets
```

In [16]:
```
predictor3 = BasicUnivariatePredictor(20, 5, prices)
predictor3.create_cnn()
predictor3.model_blueprint()
```

```
Model: "sequential_3"
```

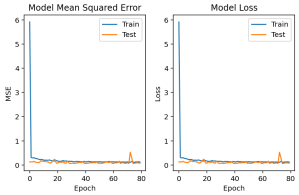
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 19, 64)	192
conv1d_1 (Conv1D)	(None, 18, 32)	4128
max_pooling1d (MaxPooling1D)	(None, 9, 32)	0
flatten (Flatten)	(None, 288)	0
dense_9 (Dense)	(None, 50)	14450
dense_10 (Dense)	(None, 5)	255

```
Total params: 19,025
Trainable params: 19,025
Non-trainable params: 0
```

In [17]:
```
predictor3.fit_model(80, 0)
```

Out[17]:
```
<tensorflow.python.keras.callbacks.History at 0x2c1ea46fd90>
```

In [18]:
```
predictor3.show_performance()
```



In [19]:
```
predictor3.save_model()
```

```
INFO:tensorflow:Assets written to: C:\Users\Max\Documents\GitHubPrivate\arguing-predictors\notebooks\assets
```

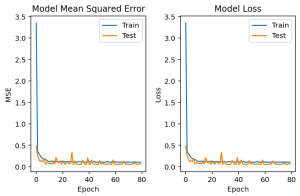
In [20]:
```
predictor4 = BasicUnivariatePredictor(20, 5, prices)
predictor4.create_bilstm()
predictor4.model_blueprint()
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (None, 20, 100)           20800
_________________________________________________________________
lstm_4 (LSTM)                (None, 50)                30200
_________________________________________________________________
dense_11 (Dense)             (None, 5)                 255
=================================================================
Total params: 51,255
Trainable params: 51,255
Non-trainable params: 0
_________________________________________________________________
```

In [21]:
```
predictor4.fit_model(80, 0)
```

Out[21]:
```
<tensorflow.python.keras.callbacks.History at 0x2c1e20680d0>
```

In [22]:
```
predictor4.show_performance()
```



In [23]:
```
predictor4.save_model()
```

```
INFO:tensorflow:Assets written to: C:\Users\Max\Documents\GitHubPrivate\arguing-predictors\notebooks\assets
```

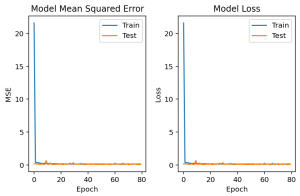
In [24]:
```
predictor5 = HybridUnivariatePredictor(2, 20, 5, prices)
predictor5.create_cnnlstm()
predictor5.model_blueprint()
```

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
time_distributed (TimeDistri (None, None, 9, 64)       192
_________________________________________________________________
time_distributed_1 (TimeDist (None, None, 8, 32)       4128
_________________________________________________________________
time_distributed_2 (TimeDist (None, None, 4, 32)       0
_________________________________________________________________
time_distributed_3 (TimeDist (None, None, 128)         0
_________________________________________________________________
lstm_5 (LSTM)                (None, None, 50)          35800
_________________________________________________________________
lstm_6 (LSTM)                (None, 25)                7600
_________________________________________________________________
dense_12 (Dense)             (None, 5)                 130
=================================================================
Total params: 47,850
Trainable params: 47,850
Non-trainable params: 0
_________________________________________________________________
```

In [25]:
```
predictor5.fit_model(80, 0)
```

Out[25]:
```
<tensorflow.python.keras.callbacks.History at 0x2c1eb7a7b50>
```

In [26]:
```
predictor5.show_performance()
```



In [27]:
```
predictor5.save_model()
```

```
INFO:tensorflow:Assets written to: C:\Users\Max\Documents\GitHubPrivate\arguing-predictors\notebooks\assets
```


6/3/2021 ModelTraining

Model Mean Squared Error
Train
Test
MSE
Epoch
Model Loss
Train
Test
Loss
Epoch

In [19]:
predictor3.save_model()

INFO:tensorflow:Assets written to: C:\Users\Max\Documents\GitHubPrivate\arguing-predictors\notebooks\assets

In [20]:
predictor4 = BasicUnivariatePredictor(20, 5, prices)
predictor4.create_bilstm()
predictor4.model_blueprint()

Model: "sequential_4"
____
Layer (type) Output Shape Param #
===
bidirectional (Bidirectional (None, 20, 100) 20800
____
lstm_4 (LSTM) (None, 50) 30200
____
dense_11 (Dense) (None, 5) 255
===
Total params: 51,255
Trainable params: 51,255
Non-trainable params: 0
____

In [21]:
predictor4.fit_model(80, 0)

Out[21]:
<tensorflow.python.keras.callbacks.History at 0x2c1e20680d0>

In [22]:
predictor4.show_performance()

Model Mean Squared Error
Train
Test
MSE
Epoch
Model Loss
Train
Test
Loss
Epoch

In [23]:
predictor4.save_model()

INFO:tensorflow:Assets written to: C:\Users\Max\Documents\GitHubPrivate\arguing-predictors\notebooks\assets

In [24]:
predictor5 = HybridUnivariatePredictor(2, 20, 5, prices)
predictor5.create_cnnlstm()
predictor5.model_blueprint()

Model: "sequential_5"
____
Layer (type) Output Shape Param #
===
time_distributed (TimeDistri (None, None, 9, 64) 192
____
time_distributed_1 (TimeDist (None, None, 8, 32) 4128
____
time_distributed_2 (TimeDist (None, None, 4, 32) 0
____
time_distributed_3 (TimeDist (None, None, 128) 0
____
lstm_5 (LSTM) (None, None, 50) 35800
____
lstm_6 (LSTM) (None, 25) 7600
____
dense_12 (Dense) (None, 5) 130
===
Total params: 47,850
Trainable params: 47,850
Non-trainable params: 0
____

In [25]:
predictor5.fit_model(80, 0)

Out[25]:
<tensorflow.python.keras.callbacks.History at 0x2c1eb7a7b50>

In [26]:
predictor5.show_performance()

Model Mean Squared Error
Train
Test
MSE
Epoch
Model Loss
Train
Test
Loss
Epoch

In [27]:
predictor5.save_model()

INFO:tensorflow:Assets written to: C:\Users\Max\Documents\GitHubPrivate\arguing-predictors\notebooks\assets