Stock Market Analysis

By Nana Ginnerup, Leif Rasmussen og Loui Wentzel

All members contributed equally to this project.

The data

All data is found on Nasdaq (http://www.nasdaqomxnordic.com/aktier)

The data contains all 139 danish stocks with the following information:
Date, Bid, Ask, Opening price, High price, Low price, Closing price, Average price, Total volume, Turnover, Trades.

The time span is (mostly) from 2000-2019, with some newer stocks in between.
Significant amount of NaN values (due to closed days or no transactions)

We then added the following:
Industry, Sub-sector and moving averages

As well as other un-used data series:
Exchange rates, Oil prices, yearly registration of cars, weather, et cetera..

First methods

- Basic BDT
  - Easy and simple to make
  - Good baseline
  - Can contain several dataseries from the real world
- RF
  - Further extension and improvement of BDT
- KNN was briefly examined but dropped

All tree-based algorithms were dropped after examination of proper prediction method and performance evaluation. Rooted problem with Stock data files.

Main method (Long Short-Term Memory)

Long Short-Term Memory (LSTM) is a supervised deep learning backfeeding RNN

Advantages:

- (!) Great results for time-series, thereby taking sequences of data (!)
- Well documented and tested
- Non-linear regression
- Classification possible (Although the time was not available)

Disadvantages:

- Requires large amounts of data
- susceptible to the ‘statistical black box’
- Very slow and tedious algorithm
- Many input- and hyperparameters to optimize and adjust

We found and tested to versions: a manual TensorFlow version and a KERAS version.

Keras

Tensorflow API

Optimizer: Adam

Loss: MSE

run time around 5 hr (full 139 set)

4 Layers (60-60-60-60)

Dropout = 0.5

30 epochs 50 batch size

Possible normalization error

Manual TF

RNN API

Optimizer: Adam

Loss: MSE

run time around 22 hr (full 139 set)

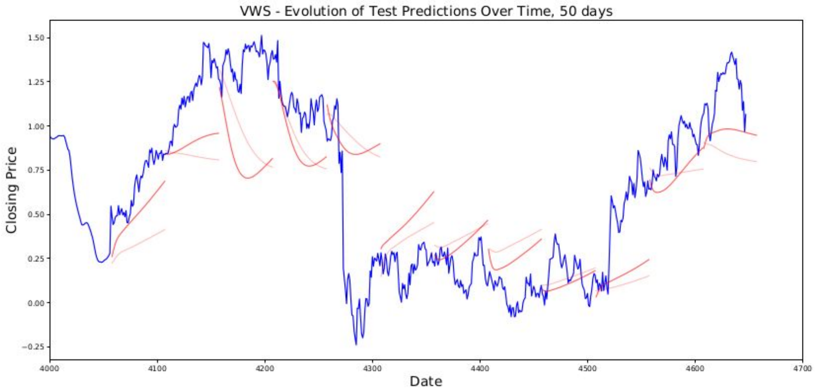
3 Layers (200-200-150)

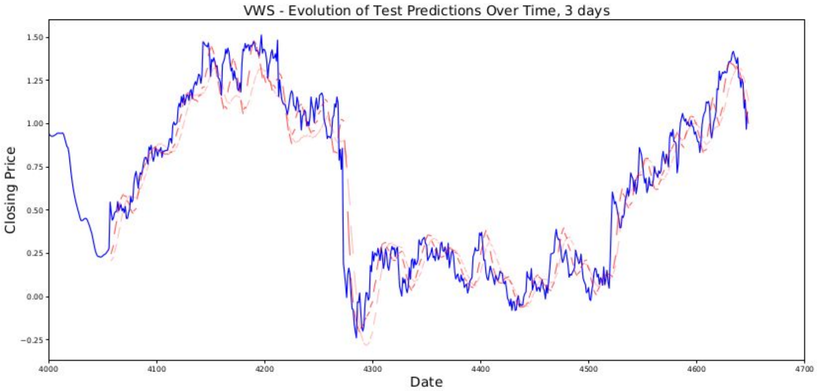
Dropout = 0.2

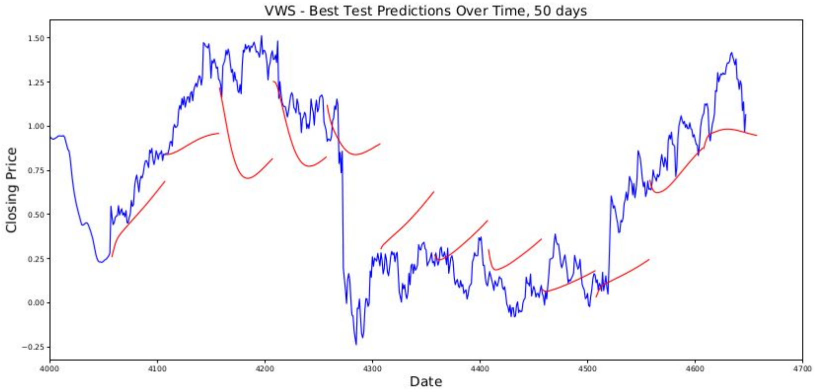
20 epochs 20 batch size

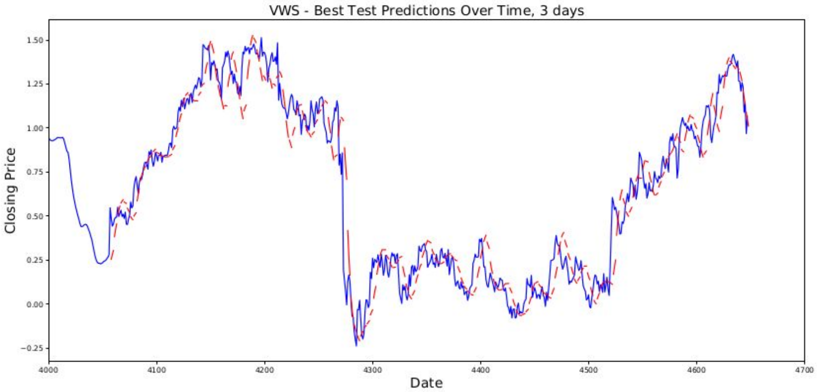
Tensorflow Vestas

MSE(3) = 0.0049 - correct 45.2 % of the time
MSE(50) = 0.062 - correct 16.7 % of the time



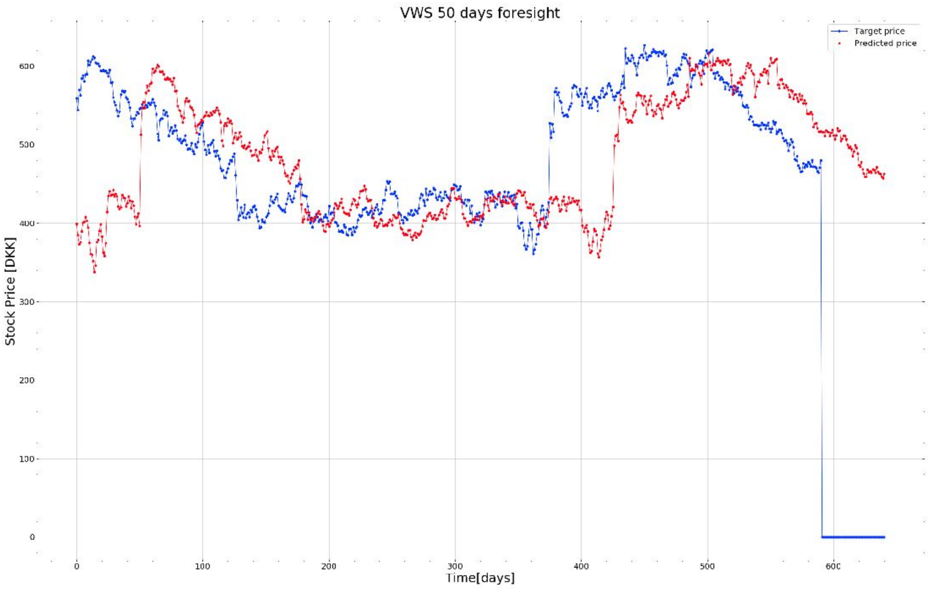






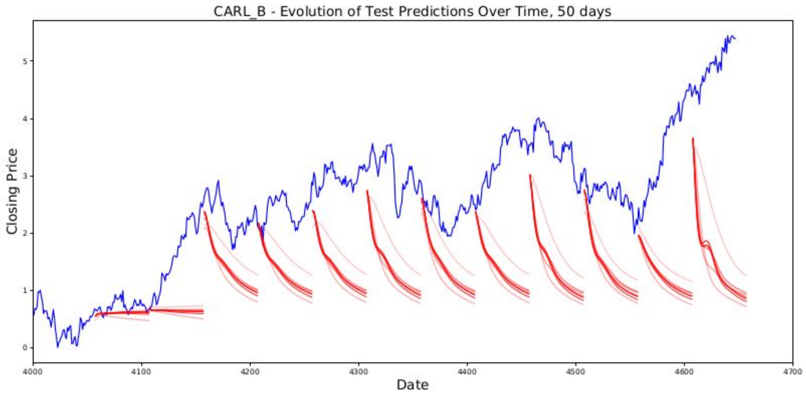
Keras Vestas

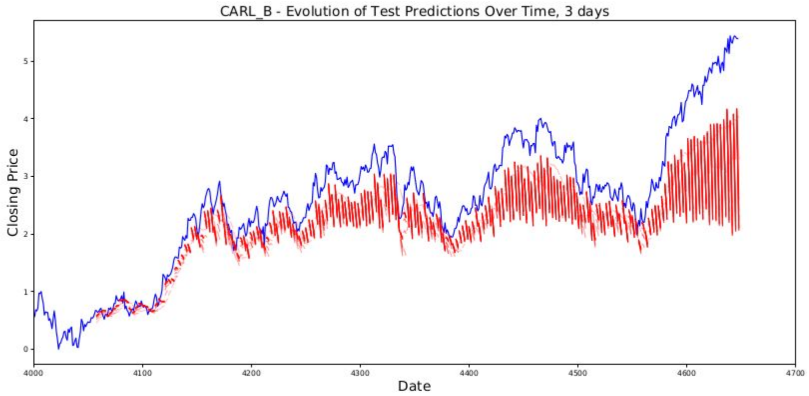
MSE(3) = 0.034
MSE(50) = 1.060

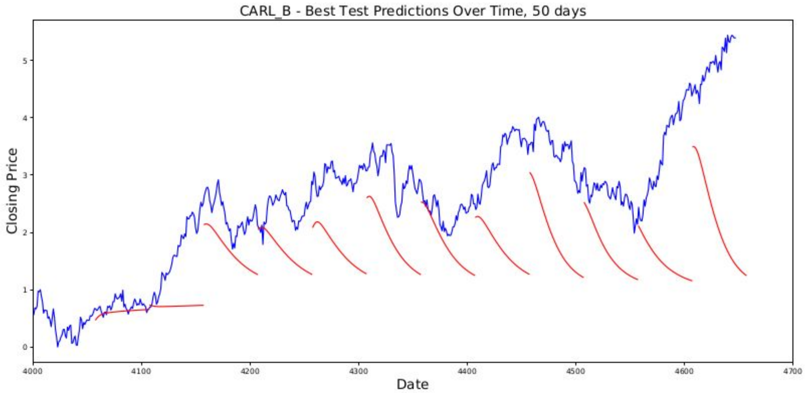


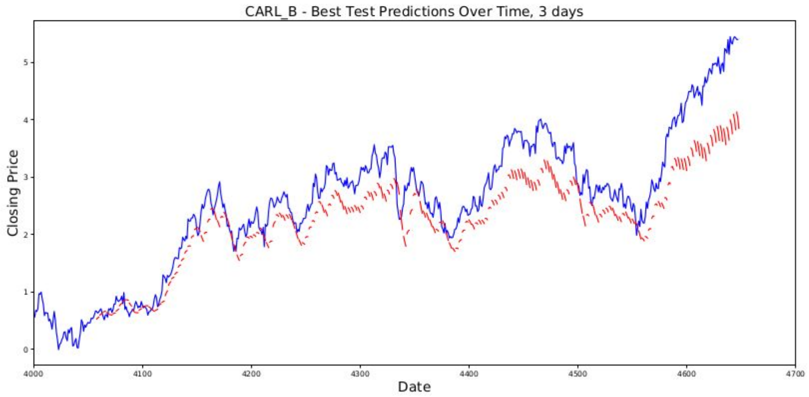
Tensorflow Carlsberg

MSE(3) = 0.16 - correct 45.2 % of the time
MSE(50) = 1.058 - correct 41.7 % of the time



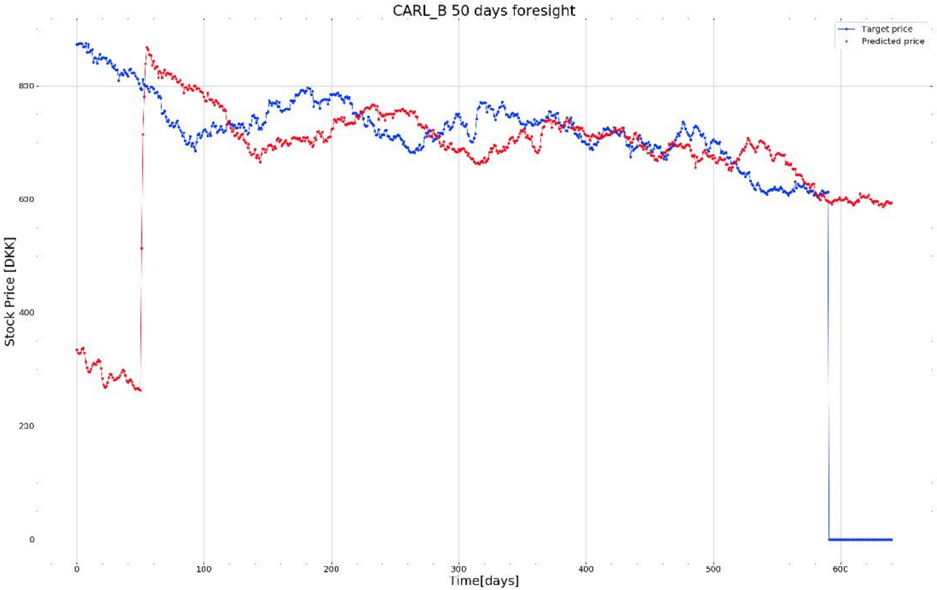


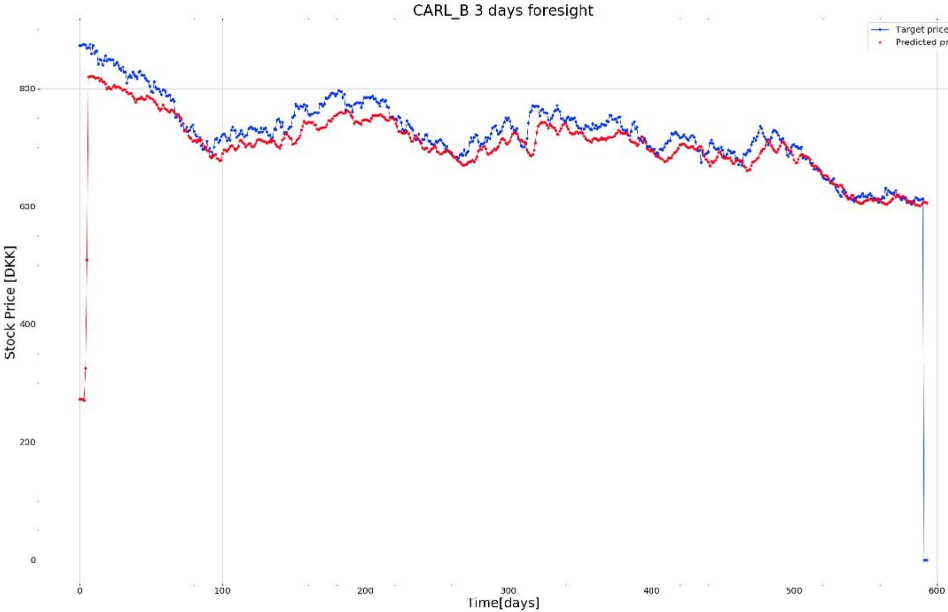




Keras Carlsberg

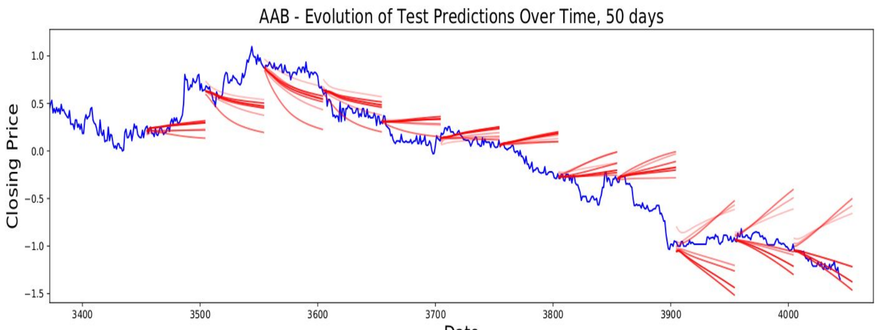
MSE(3) = 5.12
MSE(50) = 6.4

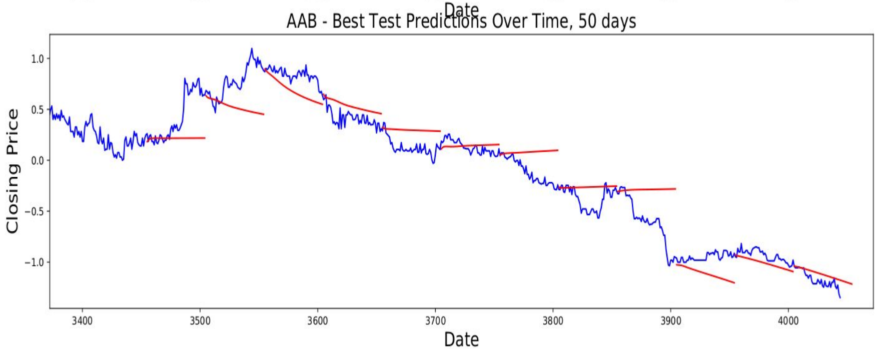


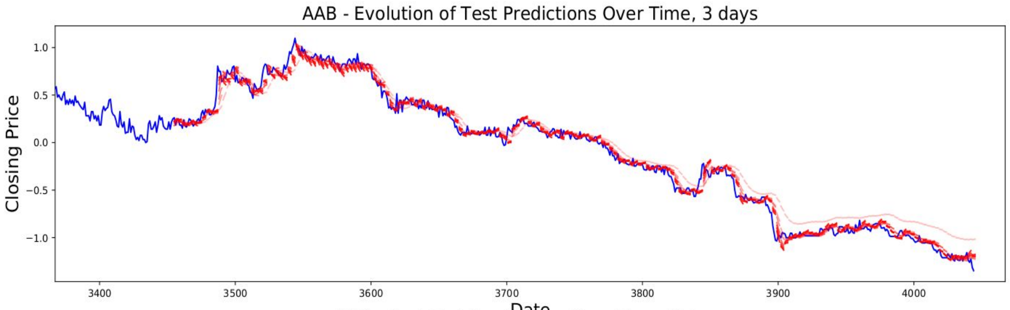


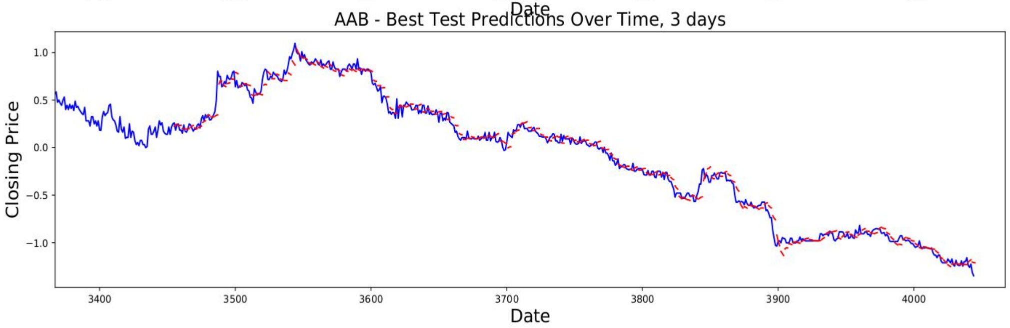
Tensorflow AAB

MSE(3) = 0.049 - correct 33.3 % of the time
MSE(50) = 0.023 - correct 50.0 % of the time



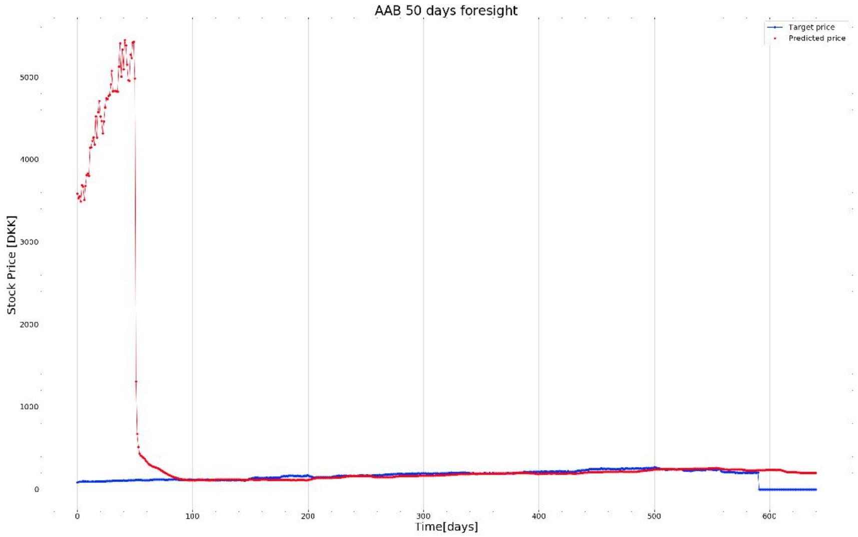


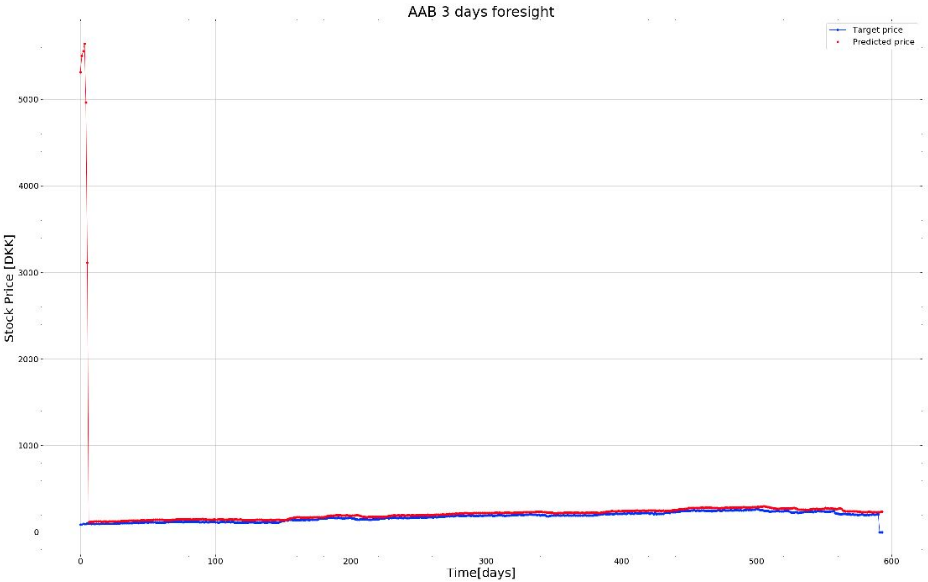




Keras AAB

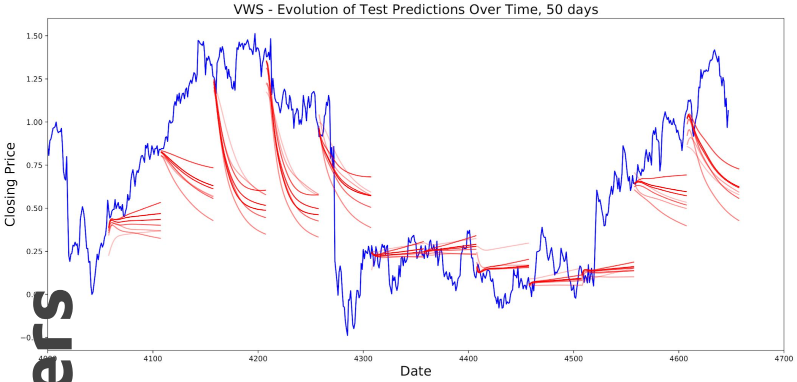
MSE(3) = 0.37
MSE(50) = 1241

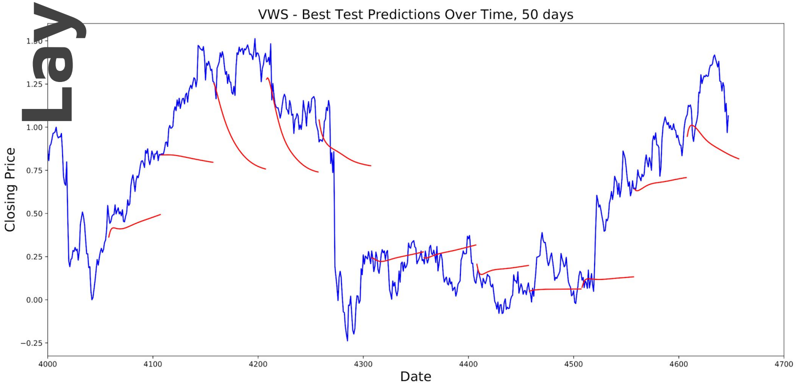




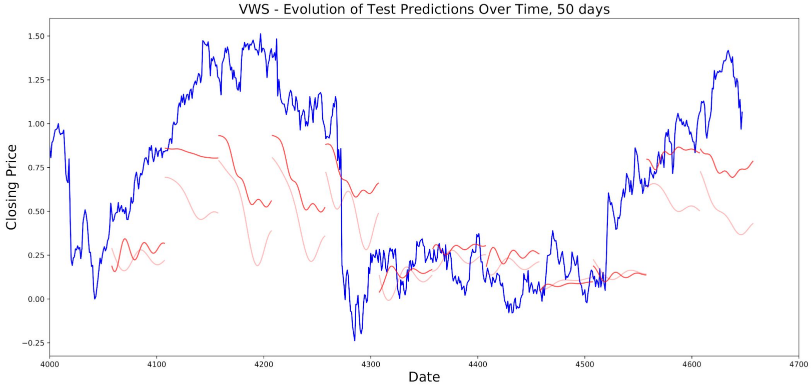
Tensorflow - Layers

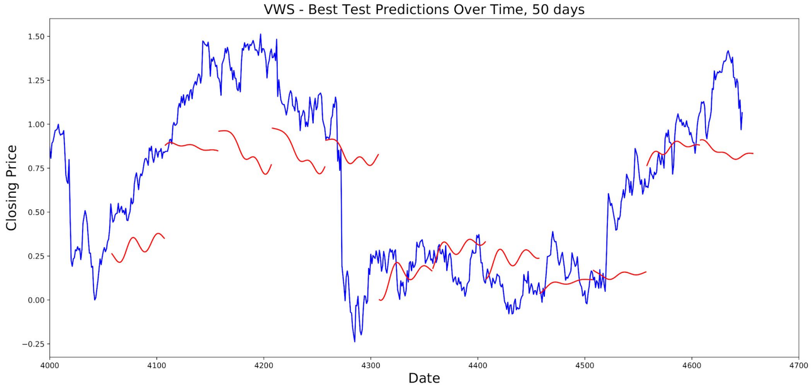
3 Layers





10 Layers

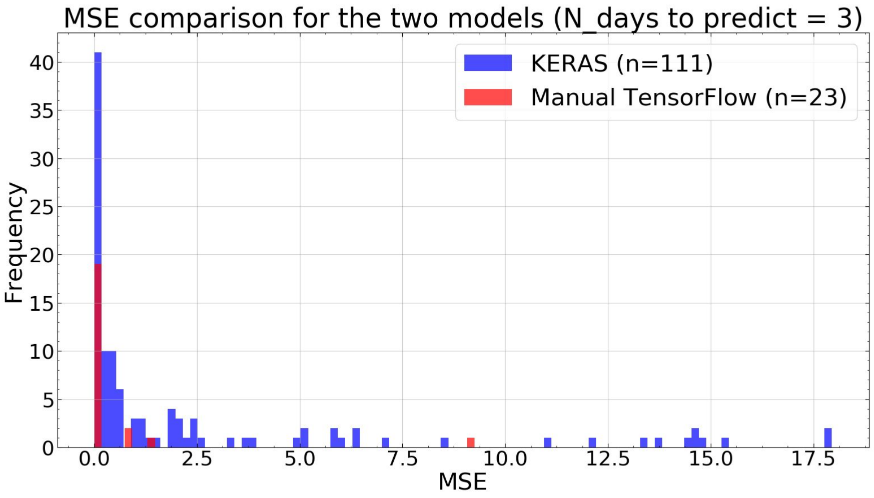


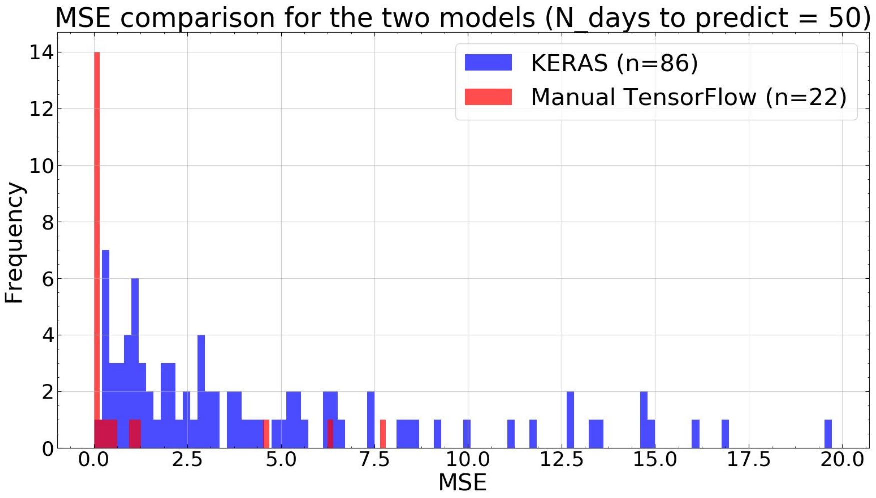


Evaluation of ML algorithms

Low count in the Manual TensorFlow version, due to very long runtimes and faulty runsetups the night before hand-in..

All 137 stocks are present in KERAS, but are enormous outliers (ie horrible models)





What’s next?

- Optimize input parameters further
- Include data from the real world in aiding the predictive power of the analysis
- Extend to European or American stocks
- Compare the analysis with moving averages
- Do proper imputation of data for NaN values

Conclusion

The stock market is a large and difficult terrain when walking in ML boots.

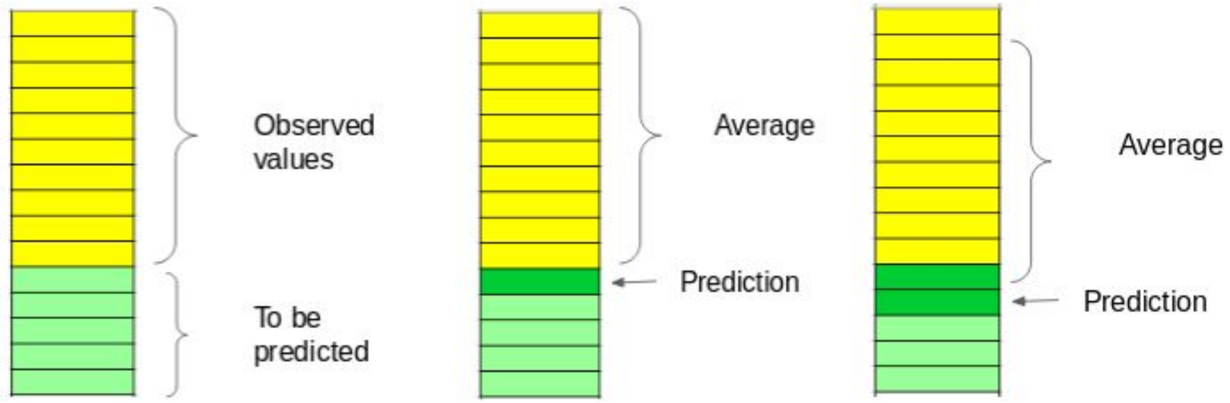
We predicted quite well on many stale stocks, but failed to reach any deeper understanding of larger changes in the market (which are the ones of interest)

The huge amount of large outliers, show, that the suddenly rising stocks are not detected by our algorithm.

All-in-all, it is no surprise no one has cracked the stock market with its unpredictable and volatile data series, making it very difficult to find any meaningful patterns.

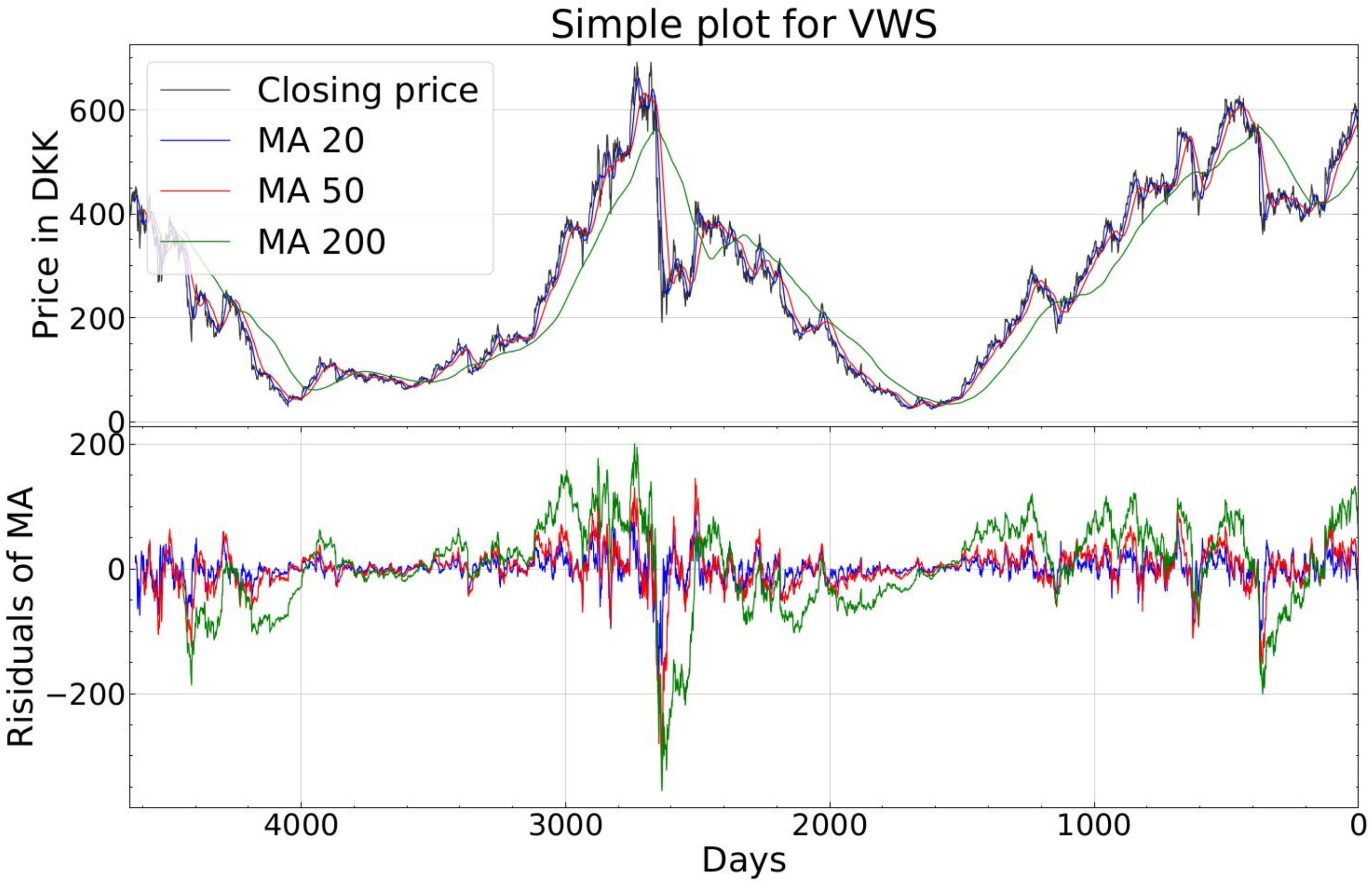
Thank you!

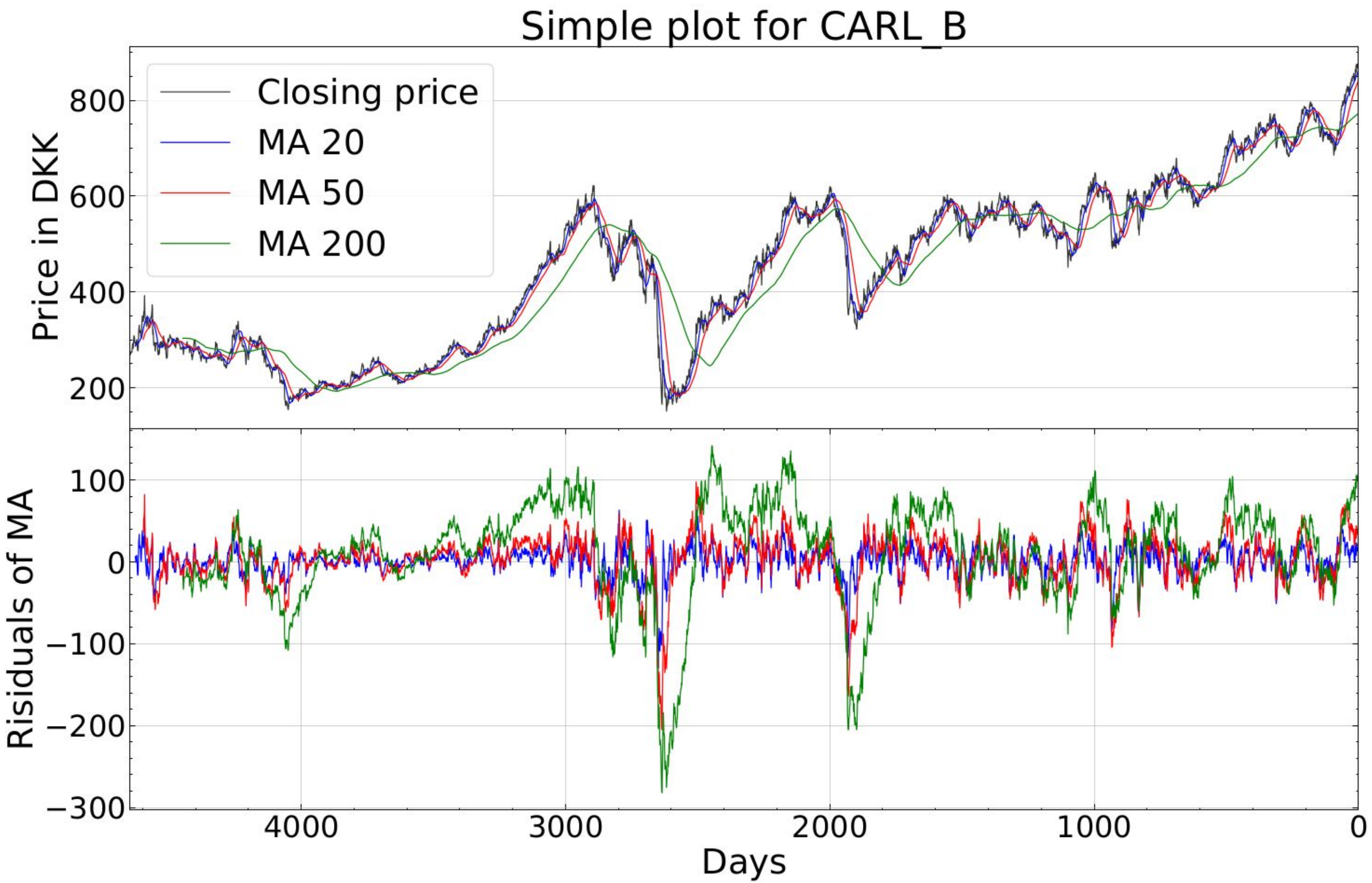
Trend indicator: Moving average

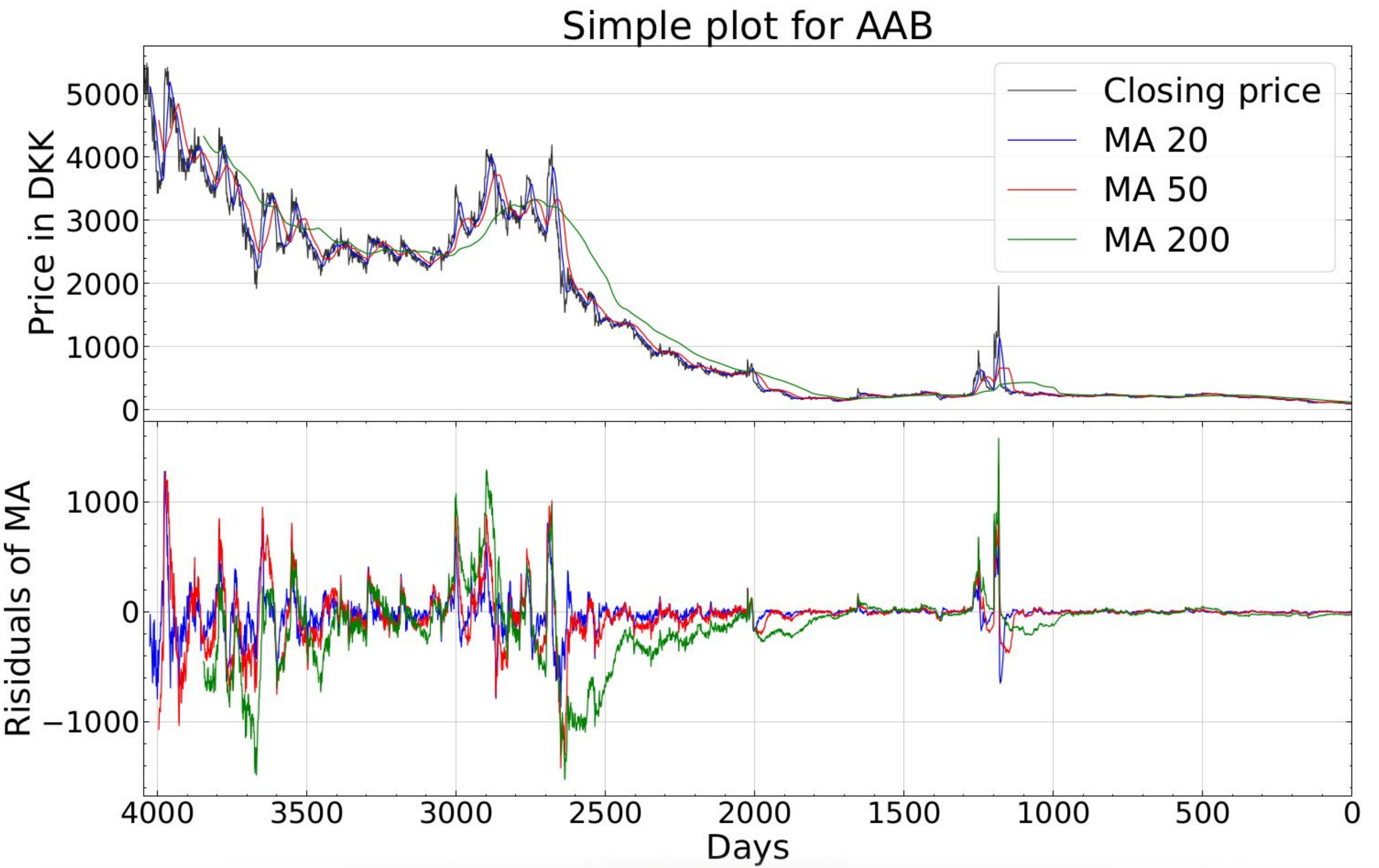


In the world of stocks, it's customary to use moving averages as a tendency measure.

We've computed (M)oving (A)verage: MA20, MA50 and MA200


Simple plot for VWS
Closing price
MA 20
MA 50
MA 200
Price in DKK
600
400
200
0
200
0
−200
Risiduals of MA
4000
3000
2000
1000
0
Days


Simple plot for CARL_B
Closing price
MA 20
MA 50
MA 200
Price in DKK
800
600
400
200
Risiduals of MA
100
0
−100
−200
−300
4000
3000
2000
1000
0
Days


Simple plot for AAB
Closing price
MA 20
MA 50
MA 200
Price in DKK
5000
4000
3000
2000
1000
0
Risiduals of MA
1000
0
−1000
4000
3500
3000
2500
2000
1500
1000
500
0
Days