Algorithmic trading using LSTM-models for intraday stock predictions

David Benjamin Lim & Justin Lundgren

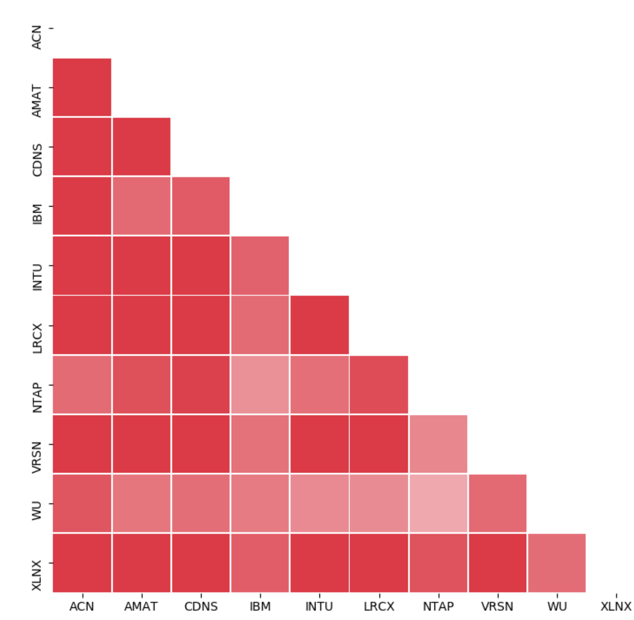
Abstract

- We investigate deep learning methods for return predictions on a portfolio of stocks in the information technology sector.
- We deploy standard time series models alongside with an LSTM network and and recently developed method called R2N2, which is an LSTM network applied to the residuals from the standard time series models.
- We obtain very modest results in terms of standard metrics such as accuracy, mean squared error and AUC.
- In terms of financial metrics however, we find that our model allows us to performs better than what the market portfolio does. Applying a simple short/long strategy we obtain a daily return of 0.49% using the LSTM.

Data set

Data

- We used a dataset between 9/11/2017 to 2/16/2018 of S&P stocks.
- In particular, we used a portfolio of 10 stocks from the information technology sector.
- We wanted the stocks to be correlated in order to improve the accuracy of the model.
- Features are: Open, Close, High, Low, Volume.



Method & Model

Data preprocessing

- We ended up using 4 ground features for training the model. These were previous returns, difference between high and low and min-max-scaled open price and volume.

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}.$$

VAR/VARMAX

- The VAR model is a simple statistical model that describes the returns of a time series at time t as a function of previous time steps, plus an error term that is normally distributed with mean zero and fixed variance:

$$y_t = \sum_{i=1}^{p} A_i y_{t-i} + A_0 + \varepsilon_t.$$

- The VARMAX model is a VAR model that incorporates exogenous features and a moving average of the noise:

$$y_t = \sum_{i=1}^{p} A_i y_{t-i} + A_0 + Bx_t + \sum_{j=1}^{q} B_j \varepsilon_{t-j} + B_0.$$

LSTM

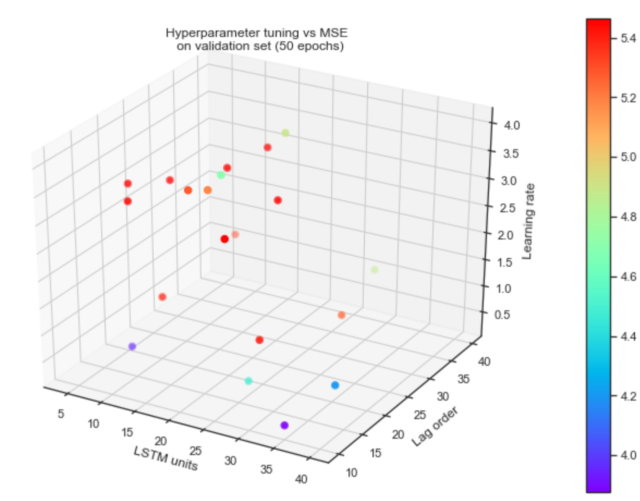
- The key idea behind LSTM is that each layer has a memory, ability to forget not useful information, and remember beneficial information.
- We refer the reader to [6] for a thorough explanation of how an LSTM network works.

LSTM implementation

- The table below shows the architecture of our network.

Layer	Units/Rate
LSTMx10	4
Dropout	0.1
Concatenate	-
LSTM	10
Dropout	0.1
Dense	10

Table 1: LSTM network



- The idea is that the first individual LSTM layer should remember individual dependencies for that stock an the second joint LSTM unit should remember joint dependencies.
- We train on a customized mean squared error that penalizes for getting the sign wrong.

$$\ell(\mathtt{data}) = \frac{\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}{\sum_{i=1}^{n} \mathbb{I}\{\text{sign}(r_i) = \text{sign}(\hat{r}_i)\} + \epsilon}$$

- Hyper parameter tuning gave that learning rate = 1e-4, lag order = 24, LSTM units in first layer = 4.

R2N2

- The idea behind R2N2 is that a simple time series model should capture linear dependencies. Then we train on the residuals from this model to capture the non-linear dependencies.
- Predict return at time $t+1$ using the value ate time t and the residual at time t.
- For R2N2 we used the exact same network as for the LSTM, which partly explains it's bad performance.

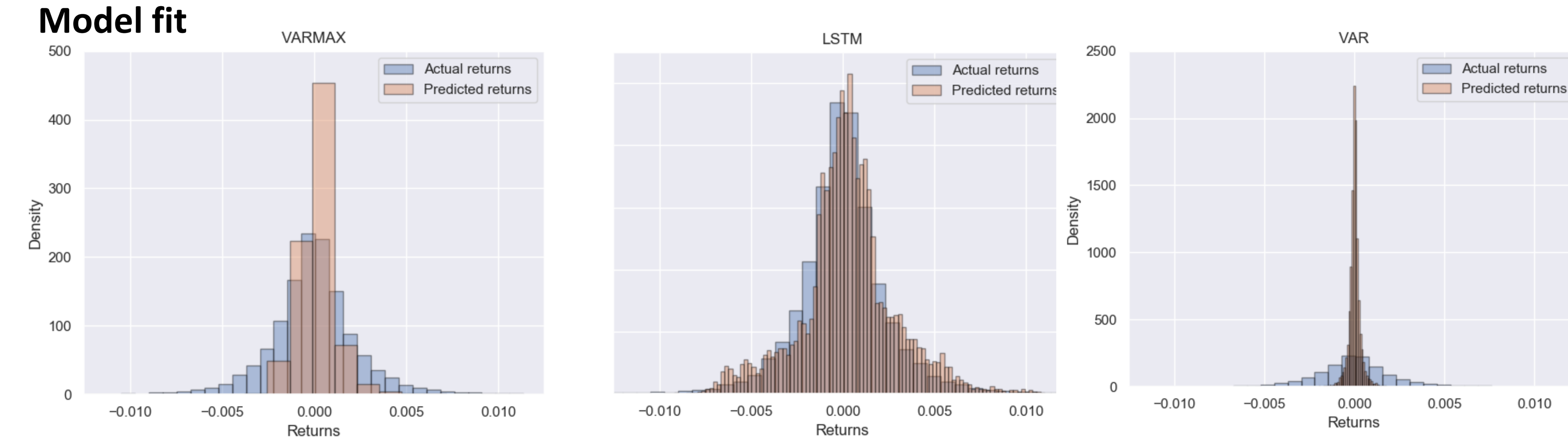
Results

Trading strategy

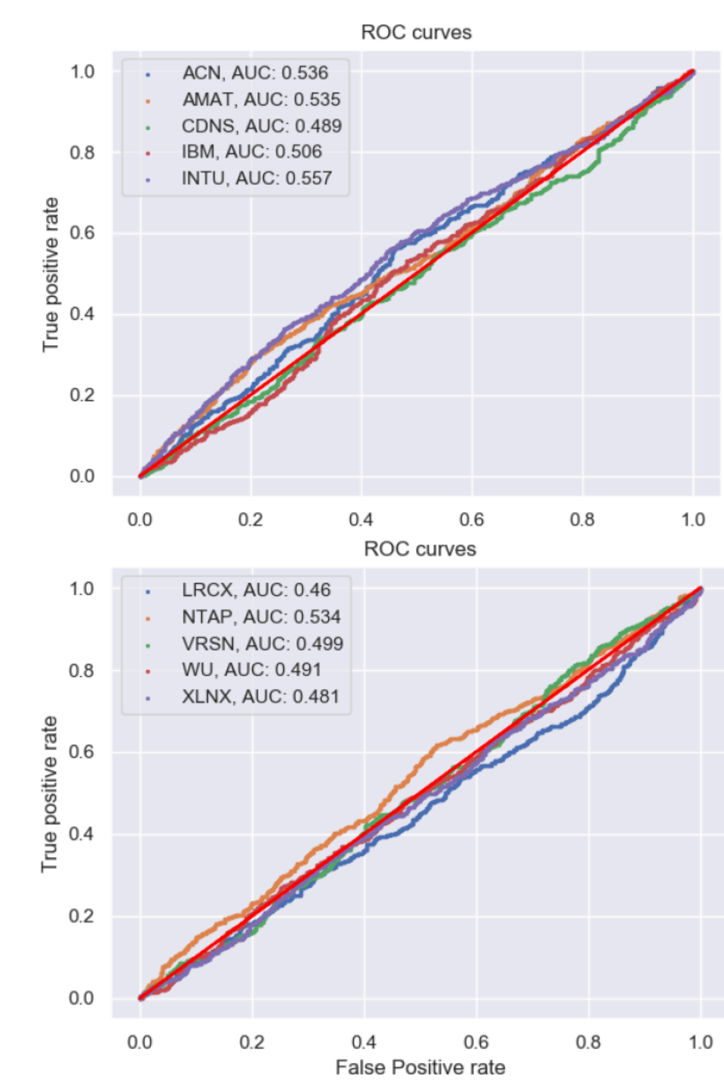
- If model predict a positive return, go long. Else, go short.
- LSTM gave the best return, with a daily return of 0.49% .
- Five minute Sharpe ratio was 6.84e-2

Model	MSE	Acc	Return	SR
VARMAX	1.21e-5	0.506	3.73	2.16e-2
LSTM	1.56e-5	0.516	8.18	6.84e-2
R2N2	8.84e-6	0.501	3.11	3.04e-2
Market	8.62e-6	0.484	−5.53	−2.02e-2

Model fit

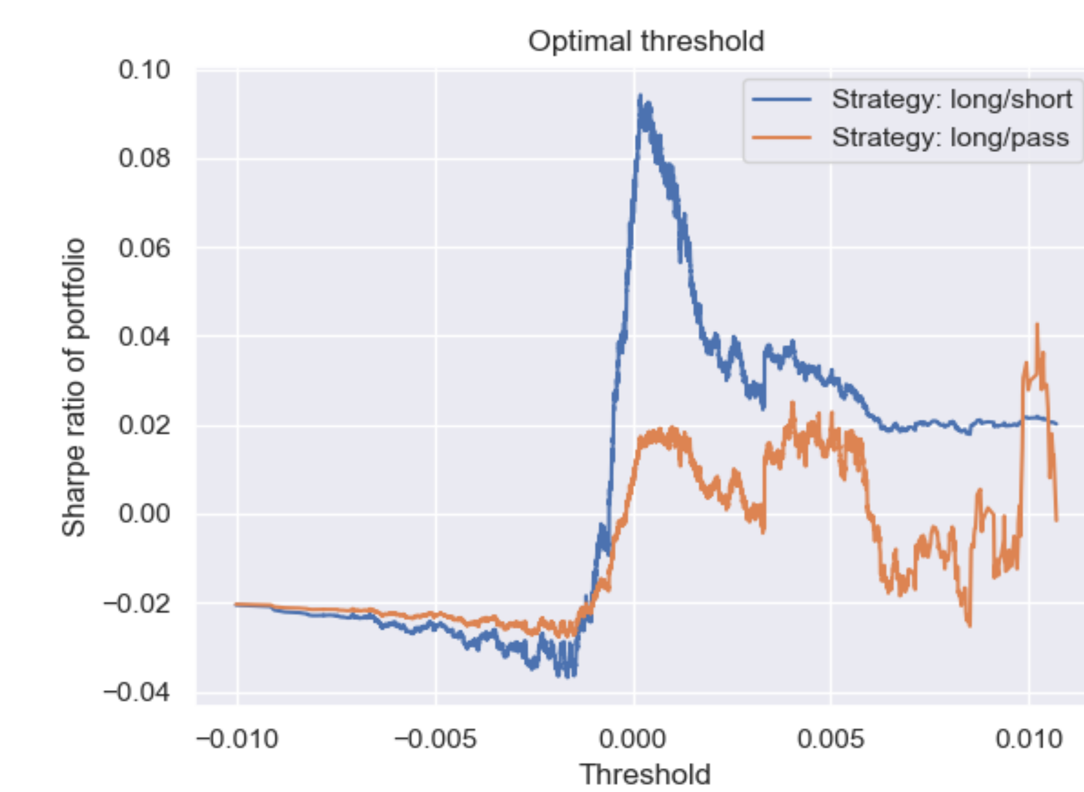


Stock metrics



Trading results

- For a future deployment of this trading strategy we show how the test set varies over different long/short-thresholds.



Conclusion

- All models fail on standard metrics. This is partly due to how random returns actually are and partly due to the fact that all returns are very close to zero since the time interval is so small.
- In financial metrics however, our LSTM models beets simply shorting the market (since the market happened to be negative for our test set). Perhaps the model somehow manages to distinguish between the magnitude of returns rather than having great accuracy.
- The R2N2-network does not seem to work at all. This is likely due to the fact that the VARMAX residuals already are so bad and that the network is tuned to fit the LSTM model.
- Due to time constraints we did not fit the R2N2 networks specifically, it would have been interesting to see the results from this model as others such as [2] have obtained good results from this method.