Outperforming the OMXS30 with Deep Networks

Bryan Yu
Royal Institute of Technology
Stockholm
bryany@kth.se

Carl Hultberg
Royal Institute of Technology
Stockholm
hultber@kth.se

Jimmy Henriksson
Royal Institute of Technology
Stockholm
jimmyhen@kth.se

Abstract

Financal market participants are inclined to ponder whether deep learning models can develop into to a competitive advantage such that they can produce above normal returns and outperform the market. In this report, we investigate various deep learning models and their ability to forecast future price movements in the context of the 30 largest publicly traded Swedish firms that constitute the OMXS30 index. We evaluate architectures such as the Deep feedforward network, CNN, LSTM, CNN-LSTM, and a novel stacked architecture which attempts to capitalize on the GAN's generative ability as well as the LSTM's ability to capture temporal dependencies. Our investigation suggests that there are patterns in the OMXS30 that can could potentially give actors a competitive advantage as shown in the forecasting success of the deep models as compared to the natural inclinations of the market.

Project report for DD2424

1 Introduction

Since the dawn of programming, financial applications have been a hot topic which many researchers has tried to tame in order to gain an advantage to other players on the market. The breadth of the financial industry offers a broad range of areas where digital technology can drive substantial impact on the industry. For example, machine intelligence, accelerated hardware, and complex programs is often used in the context of high-frequency trading to identify mispricings and advantageous opportunities coupled with automatically executed trades completed in fractions of a seconds has resulted in sizeable increases in trading activity and the creation of new hedge funds. Notable events include the flash crash of 2010 and success of firms like DE Shaw and Renaissance Technologies. Machine intelligence has the power to process an immense amount of data in its decision making from several different markets and sources and drive decisions at speeds that humans has no chance of matching. In general, in this environment firms compete on speed and the fastest algorithm translates into increased profitability(Chen 2019).

Historically, the machine learning techniques dominated computational methods on financial time series. However, in recent years there have been an increased amount of research on financial times series from the field of Deep Learning which has outperformed the machine learning equivalents in terms of accuracy. Successful applications of machine intelligence on financial time series has the potential of substantial impact on the industry, and could potentially generate better returns while lowering risk to firms who implement these algorithms - a stronger risk adjusted return. However, for the deep learning implementations on financial times series to be relevant in a real-world setting, the accuracy of the predictions needs to be significantly higher than the naive predictor with 50% accuracy.

In our project we used Long-Short Term Memory (LSTM), Convolutional Neural Network (CNN), Generative Adversive Network (GAN) models, and different variations and combinations of them, in order to predict the trend of the thirty most traded stocks in the Swedish stock market index (STO) called OMXS30. The models were trained on daily price and market capitalization data ranging from January 2000 until April 2020 resulting in 5301 days trading of 48 different stocks. The results were varying depending on the models of choice, but in line with the accuracy of the current literature on the topic. We also investigate a novel model which combines the advantages of a LSTM and GAN where the input of the first LSTM is the price vector and the input of the second LSTM is a estimation of the lookahead prices generated by a GAN which mimics the predictive actions of a market participant. Our results suggests that the Swedish market may be more predictable than other markets which we speculate may be a function many drivers including but not limited to market actors, complexity, size, and regulation.

2 Related Work

The literature review by Sezer et al. (2019) has served as a basis for this project by presenting a systematic overview of the many different fields of deep learning research on financial time series. Sezer et al. (2019) divides the field of research based on the underlying asset and what type of prediction the paper does. Regarding trend, Sezer et al. (2019) points out that there are many similarities with price prediction as the characteristics of the input data are the same and the only difference is the interpretation of the output. Das et al. (2018*a*) used neural networks to predict the movements of individual stock of the SP500 index. Compared to prior work their data-set was significantly bigger with approximately 54 years of daily return data. However, additions and deletions of stocks into the index is more common on the SP500 compared to the OMXS30 and with the time frame of 54 years there have been a lot of changes resulting in a sparsity of 68.54%. Das et al. (2018*a*) handled this by dividing the stock returns into 19 percentiles severely impacting the level of detail in the data, something we did not need to take into account due to low variations of stocks constituting the OMXS30 over the last 20 years. With a look-ahead period of 10 days their model predicted the trend with an overall accuracy of 58% and with a look-ahead period of 30 days the accuracy is about 55% suggesting that the recent history does impact predictability.

Zhou et al. (2018) proposes a generative framework using LSTM and CNN for adversarial training in order to forecast the stock market. Their experiments shows indication that look-back period and look-ahead period both severely impact prediction accuracy and they find that look-ahead of 5 days and look-back of 20 days gives the best accuracy. Data-wise Zhou et al. (2018) does not predict the movement of a whole trading day and has instead divided a trading day into 242-minute intervals. The effects of this are not discussed in the paper, however, it should not impact the accuracy significantly either way since the characteristics are similar.

3 Data

The OMXS30 consists of 30 companies at any time. Over a given period, companies may enter or exit the index. As such, we have collected the index constituents for each 6 month period from January 2000 to April 2020.

The stock data has been collected from a Bloomberg Terminal exported to a xlsx-file. One challenge with the data collection was to find out which stocks that had historically been included in the stock index. This information could be obtained by running a SQL-query against the Bloomberg Terminal with a loop listing all historical OMXS30 members by their Bloomberg tickers. These Bloomberg tickers was then used and all data was collected for all stocks during the entire sample period.

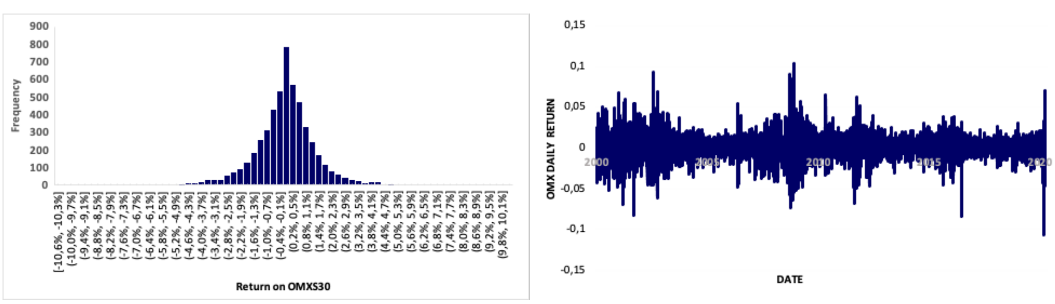
The data includes the daily close prices and current market capitalization of all OMXS30 stocks including the daily reference value of the index itself, with dates on the rows and stocks on the columns. Since the OMXS30 constitution has changed in the investigation period, we ended up with 48 different stock tickers over the sample period.

The data collected consisted of 5301 rows and 96 columns.

ALFA SS Equity		
Dates	PX_LAST	CUR_MKT_CAP
2002-05-15	N/A N/A	N/A N/A
2002-05-16	22,75	N/A N/A
2002-05-17	24,5	10943,8557
2002-05-20	24,5	10943,8557
2002-05-21	24,375	10888,0197
2002-05-22	23,625	10553,0037
2002-05-23	23,75	10608,8397
2002-05-24	23,75	10608,8397

Table 1: Sample of the collected Alfa laval stock data during their IPO period.

One challenge with the data matrix as mentioned in the related works was to handle missing values in the matrix. One way to handle this is by putting the data in buckets. However, this was not required in this work since the data was not sparse enough. The OMXS30 is not updated as frequently as for example S&P500 where the index is updated weekly and consist of 500 companies compared to 30. An example of the issue when the index gets a new member are shown in table 1, where Alfa Laval became public on May 16 2002 and then became a member of the OMXS30 Das et al. (2018*b*). Note the N/A on 2002-05-15.



(a) Histogram of the OMXS30 index return (b) Daily return of the OMXS30 index from 2000-2020

Figure 1

Figure 1 shows a plot of the daily returns on the index and the presence of outliers and periods of high and low volatility.

4 Methods

4.1 Data Preprocessing

Our dataset considers a 20 year period in which the constituents of the OMXS30 has changed over time. As such, we conducted a processing step to the data such that companies not member to the OMXS30 at a given date will have their prices masked. Therefore, at any point in time, the feature vector consists of approximately 30 active features (note that there are situations when OMXS30 consisted of only 29 companies). Finally, we split the dataset using a 60-20-20 training-validation-test ratio on the dataset. The reported accuracy is determined per performance on the test set.

4.2 Look ahead-Look back

Look back and look ahead serves as a proxy for market actors' tendency to review historical data as a guide to future decision making where the look back is defined as the number of past trading days the actor takes into consideration and the look ahead is defined as the number of trading days into the future. We considered look backs of 10, 20, 60 days and look ahead of 5, 10, and 20 days.

With regards to the look back, the feature vector is augmented such that the price features of all days in the look back period is combined. Therefore, the feature vector for a 60 day look back is larger than that of a 5 day look back. This reflects the additional information available to an actor who considers a longer slice of history. With regards to look ahead, the price of the OMXS30 into the future as compared to the current trading day, determined whether the price has increased or decreased and were labeled as 0 and 1 appropriately.

4.3 Multi Layer Perceptrons and Deep Feedforward Networks

Multi layer perceptrons(MLPS) and their extensions Deep feedforward networks(DFF) are traditional classes of neural networks with one or multiple layers of fully connected nodes trained via back propagation Goodfellow et al. (2016). In our experiments, the MLP contains one hidden layer of 100 node while the DFF contains three layers of 100 nodes each, with dropout and batch normalization inserted between each layer and topped with a sigmoidal activation.

4.4 Convolutional Neural Networks - CNN

CNNs are a class of deep, feed forward architectures with reported success in areas of computer vision and image analysis. Their marked characteristic is the application of repeated layers of convolution, normalization, and pooling, with a fully connected layer at the end. LeCun et al. (1999). In our experiments, the CNN contains two sets of convolution and max pooling with kernels of size 2x2 and stride 2x2 followed by a two layer MLP of 100 nodes and topped with a sigmoidal activation.

4.5 Long Short Term Memory - LSTM

Long Short Term Memory networks as per Hochreiter & Schmidhuber (1997) are known for their ability to deal with the vanishing/exploding gradient problem faced with recurrent networks. Thus, they are particularly suited for learning long-term dependencies which makes them a prime candidate in capturing patterns in price trends. For our experiments, we apply a network with one LSTM layer of 100 hidden units topped with a fully connected layer with a sigmoidal activation.

4.6 CNN LSTM

The combination of CNNs, LSTM, and MLPs have been studied by Sainath et al. (2015) and attempts to take advantage of the ability to capture long-term dependency of the LSTM layer while maintaining the advantages of the CNN's convolution and pooling layers which simultaneously reduces the number of trainable parameters while seeking hierarchical structure in the data.

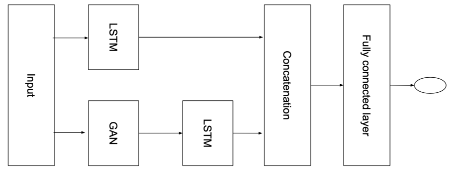


Figure 2: Stacked Model

We intuit that the CNN and LSTM combination would be appropriate for the task of price trend prediction with two major drivers (1) the LSTM addition allows for capturing long term dependencies between prices and (2) the series of convolutional and pooling facilitates identifying structure in the look back prices. The intuition is that consistent price movements results in reduced information gain due to momentum effects. Thus, convolving over these time period would allow the model to maintain accuracy while increasing training times through reducing the number of trainable weights. For our experimentation, we stack the above CNN and LSTM.

4.7 Generative Adversarial Networks - GAN

GANs were introduced by Goodfellow et al. (2014) as a deep architecture which learns through an adversarial process where two models are trained simultaneously. The generative model attempts to capture the joint distribution of the dataset while the discriminating model attempts to predict whether a sample comes from the original data or whether it was fabricated. Thus, the generative model attempts to maximize its chances of fooling the discriminative model while the discriminative model attempts to minimize its chances of being fooled. This adversarial process has been used successfully with results in image generation[Goodfellow (2016), Radford et al. (2015), Karras et al. (2017), Brock et al. (2018)].

We hypothesize that the GAN's adversarial training can simulate the habits of short term traders and other market participants. The generative ability of the GAN simulates the trader's forecast into the look ahead period. The GAN is trained on historical prices and reflects how traders study historical price movement to develop an intuition of towards market dynamics.

4.8 Stacked Model

We introduce an experimental model's which attempts to capitalize on the benefits of GAN's generative ability and LSTM's ability capture temporal dependencies. Firstly, the GAN is trained on the price vectors in an attempt to learns the join distribution of price movements. Afterwards, the learned GAN is inserted as a component of the stacked model's architecture, which receives the current price vector and generates a projection into the look ahead period. Simultaneously, a parallel LSTM network receives the current date's price vector. The outputs of the two parallel networks are concatenated and sent to a fully connected layer with one output node with a sigmoidal activation. The network is trained end-to-end with the adam optimizer and a learning rate of 0.0001.

5 Experiments

In order to evaluate the actual accuracy we took the same approach as Das et al. (2018*a*) and calculated the frequency of each label, e.g. the number of times the market trend was up or down in the validation and training data. For example, if the market trend was up 52% of the days in the validation data then for our model to be better than a naive implementation is has to have higher accuracy than 52%. In the table under the result section, the frequency is shown for every look-ahead period as the "baseline".

5.1 Parameter evaluation

In previous papers within the field of deep learning applications on financial time series from Das et al. (2018*a*) and Zhou et al. (2018) there have been indications that the look-ahead and look-back period have a significant impact on accuracy. Furthermore, in Zhou et al. (2018) paper they saw that the LSTM required a lower learning rate than the CNN to maximize accuracy. Therefore, it felt natural for us to have the aforementioned statements as a basis for our experiments and in our parameter evaluation. Furthermore, for learning rate we used the package Keras' optimizer ADAM which is an adaptive learning rate optimization algorithm who uses stochastic gradient descent and finds individual learning rates for each parameter. ADAM has default input for learning rate, decay, beta to mention a few, we decided to use a lower learning rate of 0.0001 since the training time still was manageable.

5.2 Results

	N = 5			N = 10			N = 20		
	M=10	M=20	M=60	M=10	M=20	M=60	M=10	M=20	M=60
Baseline	0.552	0.552	0.552	0.554	0.554	0.554	0.566	0.566	0.566
MLP	0.497	0.548	0.525	0.594	0.454	0.493	0.667	0.611	0.661
DFN	0.562	0.568	0.561	0.619	0.630	0.624	0.699	0.723	0.671
LSTM	0.547	0.564	0.558	0.613	0.637	0.652	0.652	0.665	0.652
CNN	0.543	0.551	0.543	0.586	0.579	0.611	0.612	0.617	0.574
CNN LSTM	0.546	0.548	0.547	0.595	0.579	0.624	0.624	0.658	0.628
SM	0.542	0.551	0.542	0.569	0.571	0.575	0.514	0.561	0.561

Table 2: Summary of direction prediction accuracy where N = lookahead and M = lookback.

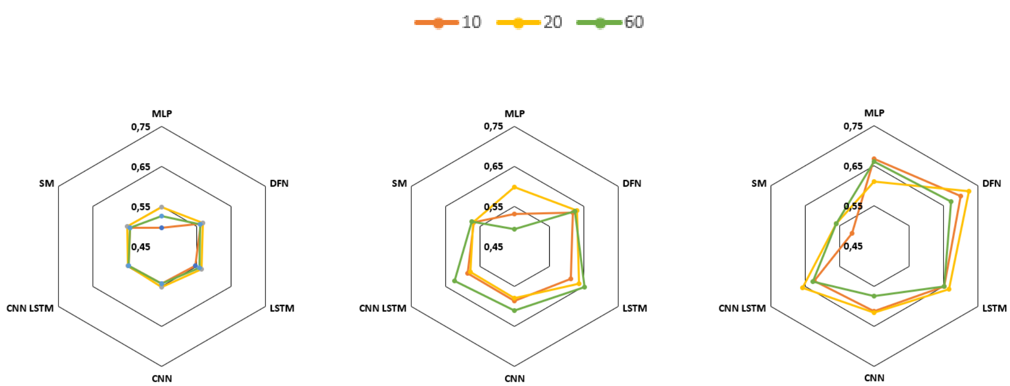


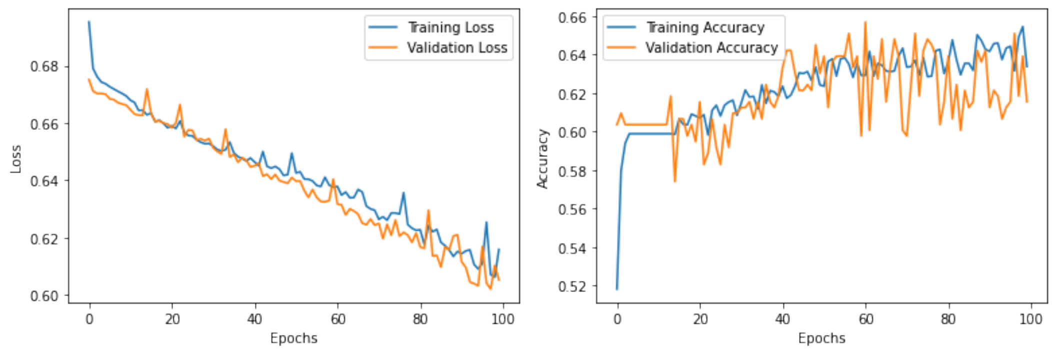
Figure 3: Model accuracy with look-ahead in order of 5, 10 and 20 days

It is interesting to note that the accuracy is better when N is 20 than when N is 10 or 5 compared Zhou et al. (2018) which had the best overall accuracy for N = 5 and worst for N = 20. The relationship is illustrated in figure 4, showing that accuracy is highest when N is 20. This implies that very short-term trends are inefficient for predicting the price movement on the OMXS30 index. Due to various differences between the actors that constitute a market (i.e. culture, preferences, demographics, etc), it is expected that every market behaves differently and therefore it is not reliable to compare outcomes of models which are trained and tested on different markets and seek to capture the patterns in its constituents. For the same N different values on M will affect the result. Overall, M = 20 gives best results where it performed better than when M is 10 or 60 in 11 of 15 different cases. The model with the best overall performance was the DFN with an average accuracy of 62.9% which is well above random guessing.

The model that performed best was DFN with an average accuracy at 62,9%. The percentage of days that the market went up in the sample period is 50,1%.

We trained all of our models on the same data-set with 100 epochs at a learning rate starting on 0.0001 (adaptive) and below in figure 4 are the results of the DFN model with 20 days look-back and look-ahead. Both the loss and the accuracy shows some erratic behaviour, and the loss does not converge indicating that an increase in epochs could potentially result in better results. However, the authors notes that the market is a complex system and that extending the training into smaller decimal places does not necessarily result in improved decision making at the investment level.

Furthermore, further training would not improve the explanability of the question, "Why?" and as such it is sufficient to identify the existance of patterns.



(a) Training and validation loss of the DFN model with 20 days look-back and look-ahead.

(b) Training and validation accuracy of the DFN model with 20 days look-back and look-ahead.

Figure 4

However, the DFN model with 60 days look-back and 10 day look-ahead shows a completely different pattern where the loss is converging as per fig 5. On the other hand, the accuracy is worse indicating that it likely found a local minimum. We speculate that due to the increased dimensionality of the input vector, the ability to arrive at a local minimum is increased along with the tendency to identify spurious patterns which may preempt the search for the global optima.

(a) Training and validation loss of the DFN model with 60 days look-back and 10 days look-ahead.

(b) Training and validation accuracy of the DFN model with 60 days look-back and 10 days look-ahead.

Figure 5

For the stacked model, we found that performance was not statistically different from the natural base rate of the market. During training, we noted that the generative component tended to make learning increasingly erratic over time with occasional perfect accuracy on the training sets but not meaningful improvements on the validation. We speculate that the GAN's estimate can contribute positively with a series of fortunate generations and vice versa - much as the emotional of a human actor. Therefore, the author notes the stacked model leaves much room for improvement such as the application of the Wasserstein GAN as per Arjovsky et al. (2017) which is known to improve convergence properties as well as increasing the training duration as to improve the quality of generated estimates.

6 Conclusion

From our experiments, we note that there may exists patterns in historical OMXS price data that can indicate future price trend as captured in the out performance of the deep learning models against the market's natural tendency. Furthermore, we find that a look back period greater than a look ahead period seems to improve performance which aligns with the intuition of the considering historical features in determining future decisions. Another conjecture is that momentum effects may be in effect, that is if the market is moving upwards, it will tend to continue to do so and vice versa. Within the scope of our investigation, we found the Deep feedforward network performed the best with up to 67% trend prediction accuracy while our experimental model brought incremental value to the task. The authors do not discount the potential of the other models.

Recently, transformer networks using attention mechanisms has demonstrated remarkable performance when used in language processing tasks which faces similar challenges in capturing temporal importance as per Vaswani et al. (2017). Thus, investigating price trend forecasting can be extended by applying attention networks and more novel architectures such as graph neural networks and variational auto-encoders. Outside of trend forecasting, it would be interesting to consider the attention network's ability to discern features from published annual and quarterly reports and whether features discerned can be useful in making longer term predictions on the strength of a publicly traded company - reflecting the tendency of fundamentalists to review company publications.

Past performance is not indicative of future results. With respect to commission costs and transaction fees, actual performance in the market may not be repeatable nor representative by the simulated results. The authors of this report reports no conflicts of interest.

References

Arjovsky, M., Chintala, S. & Bottou, L. (2017), 'Wasserstein gan', *arXiv preprint arXiv:1701.07875* .

Brock, A., Donahue, J. & Simonyan, K. (2018), 'Large scale gan training for high fidelity natural image synthesis', *arXiv preprint arXiv:1809.11096* .

Chen, J. (2019), 'High-frequency trading (hft)'.
URL: *https://www.investopedia.com/terms/h/high-frequency-trading.asp*

Das, S., Mokashi, K. & Culkin, R. (2018*a*), 'Are markets truly efficient? experiments using deep learning for market movement prediction'.

Das, S. R., Mokashi, K. & Culkin, R. (2018*b*), 'Are markets truly efficient? experiments using deep learning algorithms for market movement prediction', *Algorithms* **11**, 138.

Goodfellow, I. (2016), 'Nips 2016 tutorial: Generative adversarial networks', *arXiv preprint arXiv:1701.00160* .

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), Generative adversarial nets, *in* 'Advances in neural information processing systems', pp. 2672–2680.

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Karras, T., Aila, T., Laine, S. & Lehtinen, J. (2017), 'Progressive growing of gans for improved quality, stability, and variation', *arXiv preprint arXiv:1710.10196* .

LeCun, Y., Haffner, P., Bottou, L. & Bengio, Y. (1999), Object recognition with gradient-based learning, *in* 'Shape, contour and grouping in computer vision', Springer, pp. 319–345.

Radford, A., Metz, L. & Chintala, S. (2015), 'Unsupervised representation learning with deep convolutional generative adversarial networks', *arXiv preprint arXiv:1511.06434* .

Sainath, T. N., Vinyals, O., Senior, A. & Sak, H. (2015), Convolutional, long short-term memory, fully connected deep neural networks, *in* '2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 4580–4584.

Sezer, O. B., Gudelek, M. U. & Ozbayoglu, A. M. (2019), 'Financial time series forecasting with deep learning : A systematic literature review: 2005-2019'.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), Attention is all you need, *in* 'Advances in neural information processing systems', pp. 5998–6008.

Zhou, X., Pan, Z., Hu, G., Tang, S. & Zhao, C. (2018), 'Stock market prediction on high-frequency data using generative adversarial nets', *Mathematical Problems in Engineering* **2018**, 1–11.