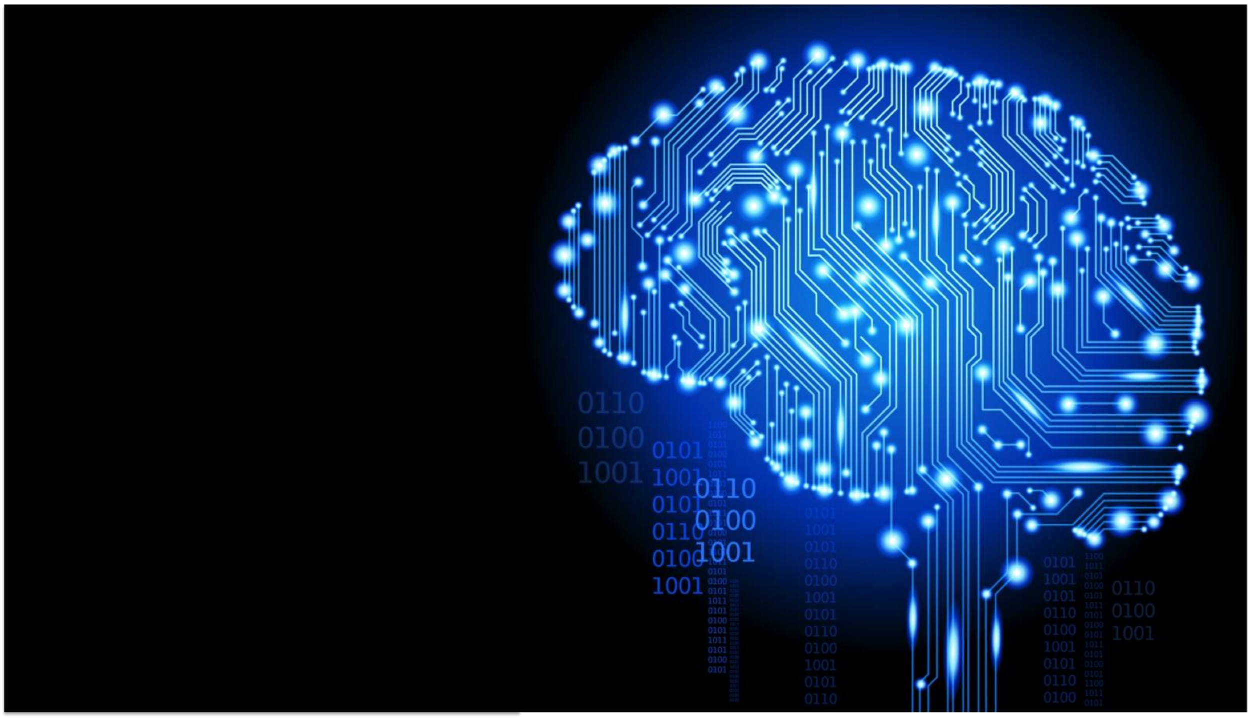
FINA 4103 Final Project



Dynamic Factor Selection with Long Short-Term Memory Recurrent Neural Network

---

Group 1: BERG, Sigurd Anders; CHAN, Hosea; KWOK, Hin Kwan (Billy)

Part 1 - Introduction and Motivation

Background

In recent years, factor investing has become increasingly popular. The amount of capital managed with smart beta, a form of factor investing, has doubled from US$ 510 billion in September 2015 (Draper, 2016) to US$ 1 trillion at the end of 2017 (Thompson, 2017). There is both data and qualitative evidence supporting that exposure to the stock factors can yield returns greater than the returns of a market portfolio with the same amount of risk (Cazalet & Roncalli, 2014). In other words, positive alpha has been achieved.

In this project, we wish to explore the possibility of quantitatively hunting alpha by utilizing the power of time-series data. Specifically, the value, momentum and profitability (also known as earnings quality) styles are explored, as these factor premiums were considered to have intuitive reasons for existing and persisting, and there should be diversifying effect from exposure to all three simultaneously.

Motivation

There are a few problems with the existing approaches to factor investing.

1) Factor investors often select a single investment style. There are no easy ways to combine or dynamically apply each style to changing market conditions under the same model.
2) Although most data in Finance are panel data, factor investors usually treat them as cross-sectional data. The time dimension information is thrown away and wasted in the process of data concatenation.
3) Static models are used for finding the relation between factors and returns. The dynamic nature of market is often overlooked. In fact, by capturing the market conditions across time and the sequential pattern of signals, more relations between future performance and factors can be exploited.
4) Traditional factor models, like linear regression and principal component analysis, can only capture linear relations between returns and factors. In the reality, it is very unlikely that returns linearly depend on a number of factors.
5) The problems of overspecification and multicollinearity in regression models hugely restrict the addition of more factors.

By solving the above problems, we can in theory boost the accuracy of the prediction of return.

Intuition

We have the following intuitions that support our model.

A. Temporal Dimension of Signals and Returns

Relationship between factors and returns can change over time. More specifically, the stock performance depends on the temporal and spatial (feature space) patterns over the the past k months, where k can be from 3 months to 24 months. Therefore, the model should have some kind of memory to remember the previous data. It should be able to extract new and useful patterns while forgetting old and inaccurate patterns.

B. Economic Intuition and Diversification Effects among Investment Styles

The three styles of factors considered are:

Value

The value style is based on the observation that relatively cheap assets tend to outperform more expensive companies. Book value versus market price is an example of a signal that can describe value. There are two potential reasons why, intuitively, exposure to the value style should yield persisting excess returns.

- Higher risk associated with holding value stocks during economic downturn, so the stocks are cheap to compensate.
- Value stocks are neglected by investors focusing on growth stocks for which they overpay.

Momentum

Momentum refers to how a stock's recent performance tends to continue in the near future and can for example be measured by the past month's return. As for the value style, two potential justifications of excess returns are given.

- When new information that should impact the valuation of a firm becomes available, the market reacts slowly and the stock price gradually changes to the new price. This implies that the market is not completely efficient, because according to the efficient market hypothesis the prices at a given time should be the fair value.
- The second explanation is highly related to the concept of asymmetric information. In the market, some investors will have more information about a company than others. Uninformed investors need to take this into account, so when they see a stock performing very well (or poorly), they may attribute it to better information possessed by other parties. Based on this, uninformed investors may follow, and thus further drive prices up or down depending to the direction of the momentum.

Profitability

Also known as earnings quality, profitability refers to how stocks from profitable companies tend to yield higher returns. The measure of profitability can for example be the ratio of gross profit over assets, and it can be justified by:

- Many investors expect that a firm's profits should have mean-reversion over time. However, profits are usually due to good management and/or business model or other favourable characteristics of the firm, and thus similar results should be expected to happen in the future also, as the core of the company would stay somewhat constant. This does of course not apply to all cases, but on average it is a reasonable conclusion.

We believe that a portfolio constructed with exposure to all three styles simultaneously should enjoy benefits of diversification, because one would expect negative correlation between some of the styles. Israel & Villalon (2013) showed that the excess returns associated with value and momentum as well as the value and profitability styles were negatively correlated for US large cap stocks from 1980 to 2012. Intuitively, while momentum-based investing involves taking a long position in a stock that has performed strongly recently, value-based investing could require a short position in the same stock if it has become overvalued after the recent jump in price. Moreover, even if a stock is considered undervalued, our net position may be zero if the profitability of the firm (earnings quality) is simultaneously bad.

C. Diminishing Profitability of Exploiting Simple Relations among Signals and Returns

Once a profitable relation among signals and returns has been found, its profitability starts to decrease. However, the profit gained by more complicated relations should sustain longer than simpler relations. Thus, our model aims at capturing those non-linear dynamic patterns instead of the obvious one.

Model Overview

Based on the above intuitions, we wish to construct a dynamic time-series model using machine learning. We have designed a model to be the foundation a trading strategy with exposure to each of the three styles while considering the problems previously identified.

A model that could dynamically adjust the emphasis (weights) given to signals as their predictive powers change was desired to tackle the issues of signal selection and changing market conditions. To manage the downward risk, the model should be able to also take the prevailing macroeconomic conditions into account through the application of top-down signals and produce signals to buy or sell shares of a given company. Additionally, while investment strategies with exposure to styles like the ones considered here are mostly long-only strategies, a equal-weighted long/short strategy was considered with the flexibility to take up long and/or short positions depending on the model output.

Extensive use of prediction techniques in the equity market has thinned out the return of investments. We believe that sophisticated methods such as machine learning algorithms are required to effectively capture the underexplored signals in the market, so as to yield a higher accuracy in return prediction. This subsequently leads to a portfolio with higher returns.

Based on the above specifications, we opted for a Recurrent Neural Network (RNN) model with Long Short-Term Memory (LSTM) cells. RNN-models are considered the best deep learning models for predicting time series, which matched the nature of the problem at hand. Different stocks respond differently to signals in the same time period so instead of training a large and complex model for all stocks, we build one model for each stock to reduce the interference among stocks. The details concerning the model are provided in Part 3: Methodology. The model selects and weighs 9 signals of the three styles described above and 3 top-down signals, and it produces predictions of the next month’s returns from which a long/short decision is made for a stock at a given time. Rebalancing is done monthly.

Part 2 - Data

The model was created and tested using a subset of the data provided by the course instructor. The investment universe was restricted to Russell 1000, and the Russell 1000 Index returns were obtained for the purpose of benchmarking. The following describes how a subset of the data provided by the course instructor was selected, filtered and transformed.

Signal Selection

The number of bottom-up signals considered were narrowed down to 9 signals, all belonging to the price momentum, earnings quality and valuation categories. Industry-specific signals were removed from consideration, as they applied to only small subsets of the stocks. Furthermore, signals for which significant amounts of data were discarded, with the specific criteria being >500 missing data points. Combination signals were also removed, with the reason being that if a particular combination of signals could be used to predict returns, our model would likely discover superior combinations, rendering the arbitrary combination signals provided obsolete. The full list of retained bottom-up signals used can be found in the table below. Three top-down signals provided were included in the model.

Bottom Up Signals		Top Down Signals
LogMktCap	ShortIntTrdVol	FedFundsRate
us_R3_CREFb_PutCallParity	us_R3_CREFb_VolumeRatio	CPI
14DayRSI	DivP	TermSpread
5DMoneyFlowVol	GPMargin	
previous_return		

Data Filtering

Since the signals in the period of 1996 to 2003 are very ill-formed (i.e. there are too many missing values) and are far from the current period, the data in this period were excluded, leaving us with the data from 2004 onwards.

Data Transformation

Normalization has been done to make the signals more comparable and in the same order of magnitude, so that the model can adjust its weights more consistently across input signals. The normalization is shown below.

$$Norm(S) = \frac{S - min(S)}{max(S) - min(S)}$$

Such normalization does not affect the accuracy of data because this is just a change in unit.

Furthermore, the target return for each training example is discretized into positive (1) and negative (0) return instead of continuous values. This allows the model to easily distinguish profitable and non-profitable stocks.

Training, Validation and Testing Split

The data was further divided into training, validation and testing data. The data from January 2004 to June 2010 constituted the training set and the validation set was data from July 2010 to December 2014. The remaining data ranging from January 2015 to February 2017 were used for backtesting. Validation set is used for hyperparameter selection while testing set is used to improve the robustness and prevents overfitting. Also, the validation set does not use any information from the training set to prevent reuse of data and maximize the generality.

Part 3 - Methodology

Recurrent Neural Networks (RNN) are considered the best deep learning models for modelling time series data. Within the RNN space, Long Short-Term Memory (LSTM) is considered the best approach and is used in the vast majority of RNN models. Since we believe that each stock behaves differently, we built one LSTM model for each of the stocks. The results of the LSTM models are used to decide the long and short list of for each of the months.

Mathematical Formulation

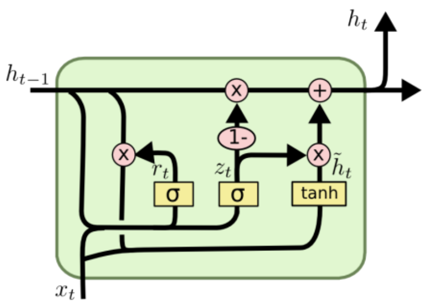
At a given time t, the bottom-up and top-down signals will be used to predict the returns observed at time t+1, in other words the model predicts the coming period returns based on the signal values observed at the beginning of the period.

$$ExpectedReturn_{t+1} = f(B(k)_{0,t}, \ldots, B(k)_{s,t}, T(j)_t)$$

Set of Bottom-up Signals of Stock S is $B(k)_{s,t} = \{B(k)_{0,t}, \ldots, B(k)_{i,t}\}$

Set of Top-down Signals of Stock S is $T(j)_t = \{T(j)_{0,t}, \ldots, T(j)_{0,t}\}$

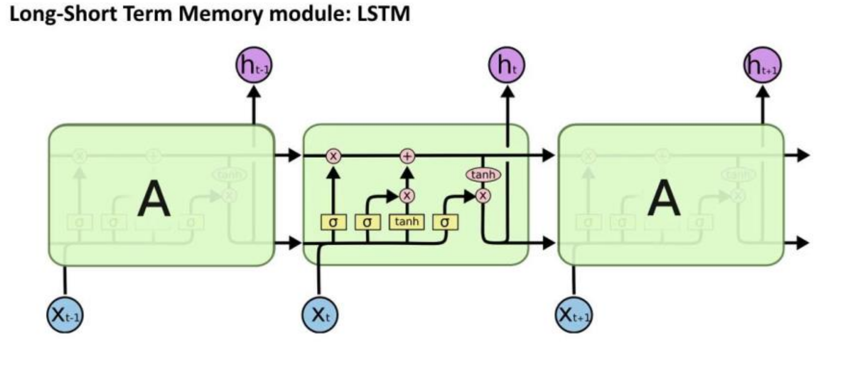
Function f represents the mapping from LSTM network input to output. The network consists of two parts. The first part is three layers of LSTM cells. The internal structure and mathematical expression of each cell is shown below.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

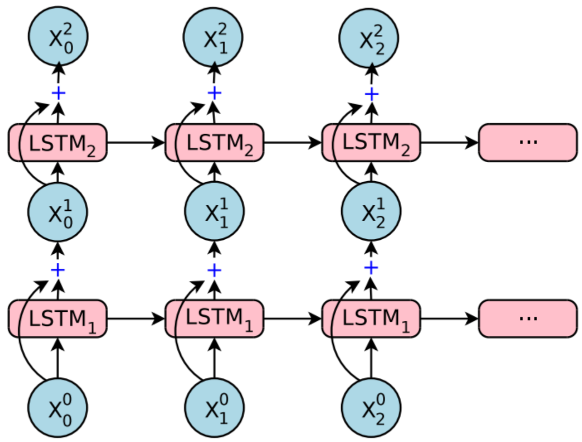
Detail of LSTM is available here. (https://colah.github.io/posts/2015-08-Understanding-LSTMs)

These cells are cascaded together to form a layer.

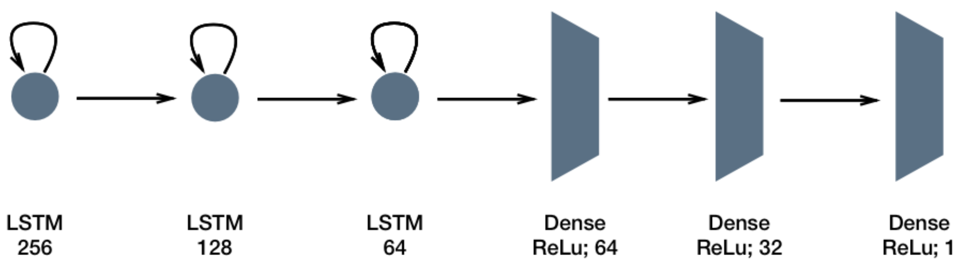


long-short term memory modules used in an RNN

Layers are cascaded together to form a grid network.



The grid network is then bridged to 3 layers of multi-layer perceptrons.



Based on the output (a value between 0 or 1), we set two dynamic thresholds to extract the buy and sell lists. Specifically, we first compute a midpoint of prediction for each stock and taking the average of the second largest and second smallest return. The upper threshold is determined by multiplying the midpoint by 1.05 while the lower threshold is determined by multiplying the midpoint by 0.95. Any stock at time t with prediction higher than the upper threshold is longed and included in the buy list. Any stock at time t with prediction lower than the lower threshold is shorted and included in the sell list.

The model is updated using the cross entropy loss function using gradient descent.

$$CrossEntropy(X,Y) = \sum p(x) \times log(q(y))$$

where $p(x)$is the probability of occurrence of outcome x of variable X, while $q(y)$is the probability of occurrence of outcome y of variable Y.

Model Tuning

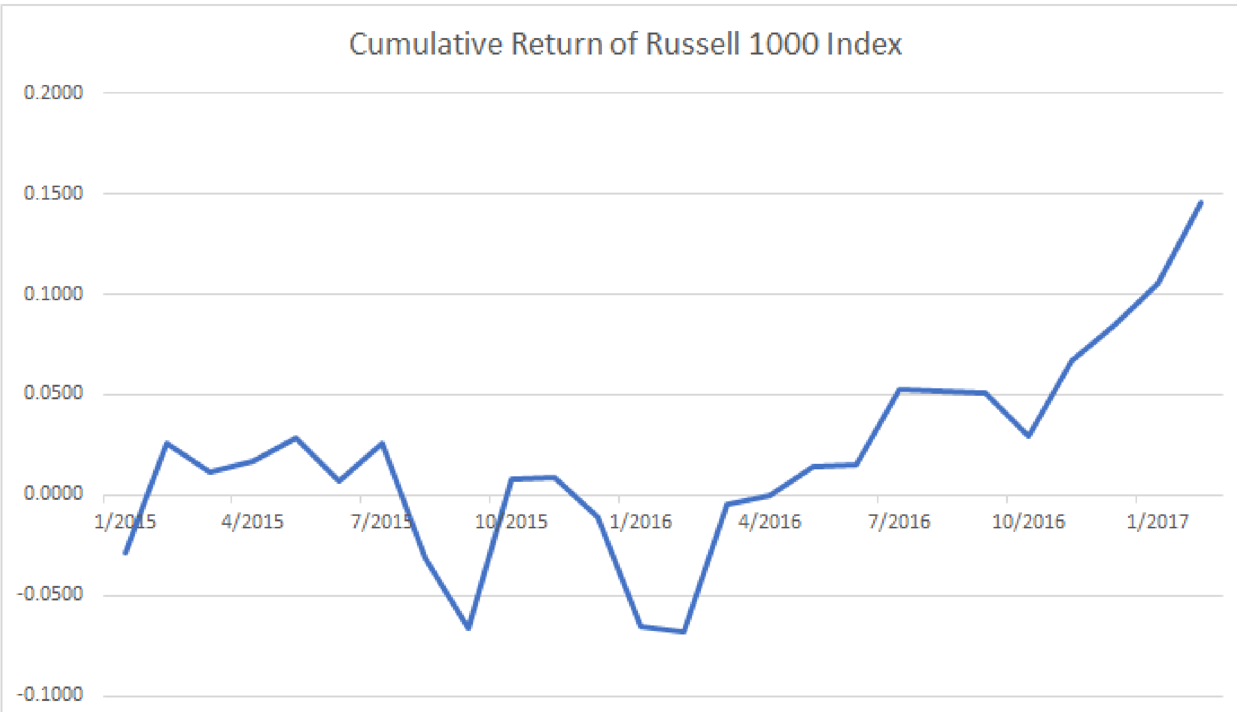
There are two ways to train the model - cluster-based and single-stock training. We decided to compare the performance of these two approaches and choose the better. For cluster-based training, the input of each epoch corresponds to a group of stocks. For single-stock training, the input of each epoch corresponds to a single stock. In both cases, the list of models are tuned separately to maximize the accuracy on the validation set. The structure of LSTM models and the values of hyperparameters vary slightly across stocks. After tuning the model, we obtained test accuracies ranging from 53% to 71%, measured by the number of correct prediction over the total number of prediction.

Part 4: Empirical Results

Benchmark

The Russell 1000 Index is used for benchmarking our strategy, which aligns with the industry practice of comparing factor portfolio returns to the market returns. Some key summary statistics of the Russell 1000 Index are presented below for the data starting from January 2015 to February 2017:

Annualized Return	Annualized Volatility	Annualized Sharpe Ratio	Skewness	Maximum Drawdown
8.59%	11.61%	0.740	0.189	13.28%



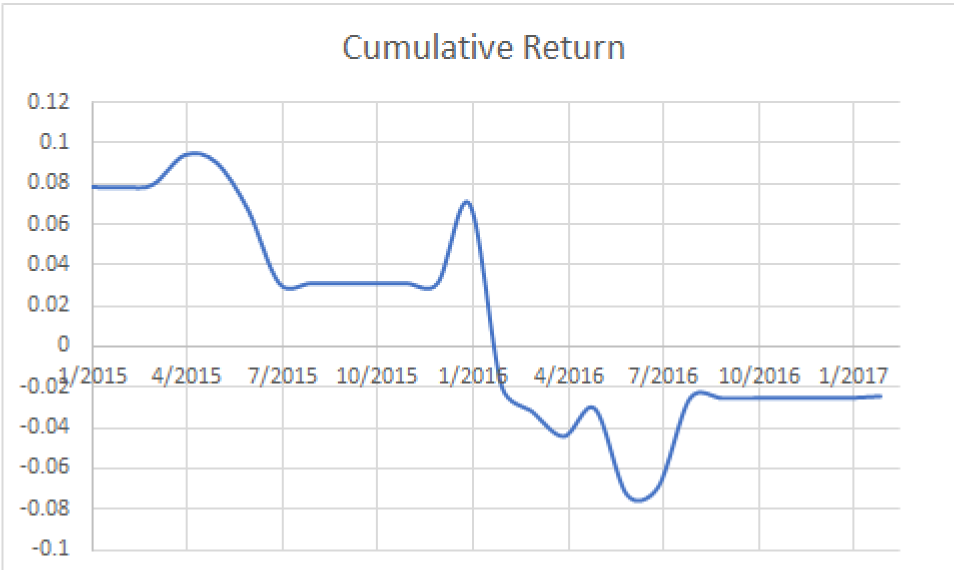
Cumulative return of Russell 1000 Index ETF

From the tables and figures above, some insights regarding the model's performance relative to the market can be extracted.

Cluster-based Training

Some performance indicator of the clustering approach is listed below:

Annualized Return	Annualized Volatility	Annualized Sharpe Ratio	Skewness	Maximum Drawdown
-0.66%	10.67%	-0.066	-0.088	15.31%



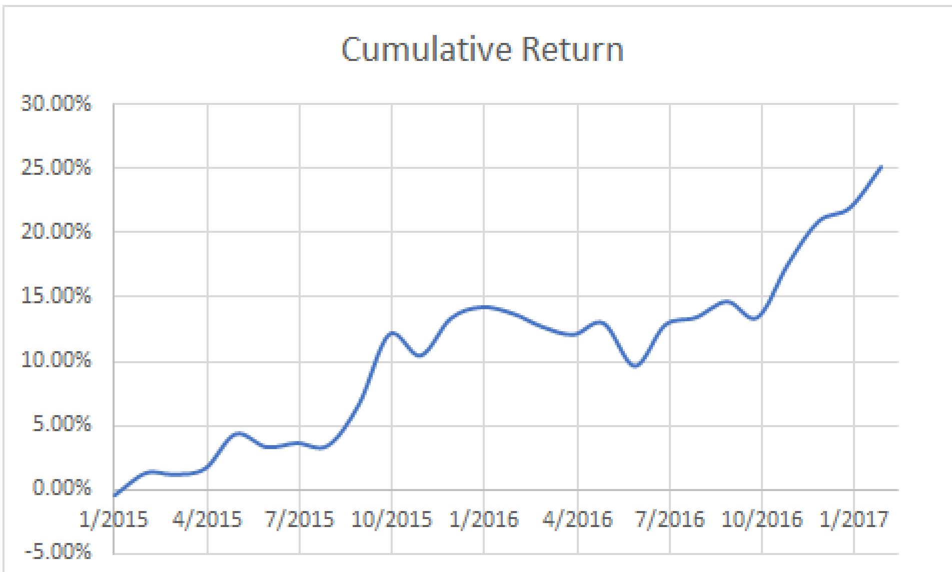
Cumulative return with the clustering approach

Individual-stock Training

Since there is not sufficient data to produce an accurate general model, we train an LSTM model for 50 individual randomly chosen stocks as an exploration of the accuracy of the LSTM approach in predicting stock prices. The robustness of this approach will be proven in *Part 5 - Robustness*.

With this approach, the model achieved the following performance. More detailed data on each month's returns are provided in the Appendix. :

Annualized Return	Annualized Volatility	Annualized Sharpe Ratio	Skewness	Maximum Drawdown
10.57%	6.46%	1.637	0.267	4.01%



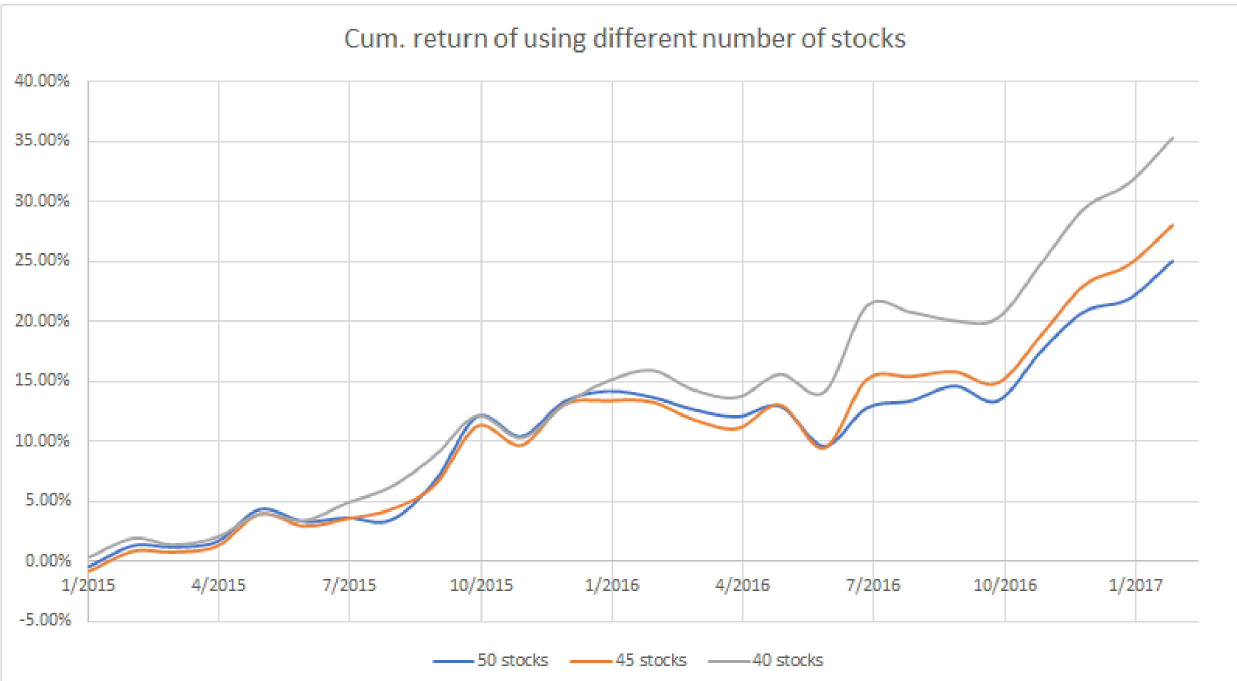
Cumulative return with the individual stock approach

From the cumulative return graphs and the backtesting parameters, we can confirm that constructing the portfolio with LSTM model on individual stocks are better than the constructing LSTM models based on clusters with the given data set. Our strategy outperforms the market portfolio.

Part 5 - Robustness

The robustness of each model has been ensured by the validation and test sets. In order to also test the robustness of our training approaches and the generality of the choice of stocks, we evaluate the return of portfolios using different number of stocks included in the strategy. The results are reported below:

Number of stocks	Annualized Return	Annualized Volatility	Annualized Sharpe Ratio	Skewness	Maximum Drawdown
50 (benchmark)	10.57%	6.46%	1.637	0.267	4.01%
45	11.70%	6.96%	1.680	0.285	3.47%
40	14.20	6.46%	2.199	0.657	1.93%



Cumulative return with different number of chosen stocks

The difference of backtesting parameters by Increasing the number of stocks from 40 to 45 is quite significant, yet they follow the same trend. By increasing the number of stocks from 45 to 50, the difference is negligible. This shows that the random picking of stocks does not affect the generality of the strategy, provided that the number of stocks included in the portfolio is sufficiently large.

Transaction costs were also considered. Specifically, we solved for the hypothetical fraction of transaction cost that would eliminate all our profit. After forcing the cumulative return to 0 in the testing set, we concluded that any transaction cost beyond 0.651% would erase all our profits.

Part 6 - Conclusion

Based on the idea that exposure to value, momentum and profitability can lead to excess returns and the belief that machine learning models can capture complex non-linear relationships between signals and returns while dynamically adjusting for changing market conditions, we designed and tested Long Short-Term Memory Recurrent Neural Network models and simulated trading according to the model outputs. The portfolio achieved 10.57% annualized return and a Sharpe ratio of 1.64.

Potential Improvement

A. Accuracy Improvement with More Data

LSTM requires a large amount of sequential data. In our dataset, we only have 108 months for training the model, which is far from enough. We believe that the accuracy could be further improved when more data is provided.

B. Time Delay Improvement with More Data

Without enough data, LSTM tends to shift the prediction by one to two periods. This reduces the performance of trading volatile stocks. We could also potentially tackle this problem by feeding the time delta into the model to make it more sensitive to frequent temporal changes.

Limitations of Model

A. Transaction as a Optimization Parameter

While transaction costs were considered in Part 5 of this report, it was not taken into account when designing the model. In Part 5 we considered hypothetical transaction cost, and found that with a transaction cost of 0.651%, the strategy's profits would be eliminated The actual transaction cost could have been modelled with data about the stock prices and bid/ask spreads. This would have given a more realistic picture of performance after taking implementation cost into account.

B. Restriction on Short-selling

It was assumed that there was no restriction on short-selling, and that there was zero borrowing cost. Future work could take the implementation impacts of short-selling into account.

C. Downside Risk Modeling

Due to the limited time window used, the model was not tested under turbulent conditions like a financial crisis. Thus, there may be some hidden downside risk associated with this strategy. Future research could be done to manage such risks, for example by including more recession data.

References

Cazalet, Z. and Roncalli, T. (2014). *Facts and Fantasies About Factor Investing*. Lyxor Asset Management, France.

Christopher O., Understanding LSTM Networks:
https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[Accessed: 8 May 2018]

Draper, D. (2016). *10 things investors need to know about smart beta.* CNBC:
https://www.cnbc.com/2016/01/19/10-things-investors-need-to-know-about-smart-beta.html
[Accessed: 6 May 2018]

Israel, R. and Villalon, D. (2013) *Building a Better Core Equity Portfolio A New Paradigm for Core Equity Investing*. AQR Capital Management.

Israel, R. and Ross, A. (2017) *Measuring Factor Exposures: Uses and Abuses,* The Journal of Alternative Investments, Vol. 20, No. 1.

Thompson, J. (2017). *Smart beta funds pass $1tn in assets.* Financial Times:
https://www.ft.com/content/bb0d1830-e56b-11e7-8b99-0191e45377ec
[Accessed: 6 May 2018]

Appendix: Results on the Testing Set Data of the 50-stock Portfolio

Month	Return	Turnover_ratio	Cumulative Return
1/2015	-0.40%	2	-0.40%
2/2015	1.74%	1.108225108	1.33%
3/2015	-0.12%	1.180952381	1.21%
4/2015	0.47%	1.318295739	1.68%
5/2015	2.64%	1.898785425	4.36%
6/2015	-0.97%	1.10989011	3.35%
7/2015	0.30%	0.987012987	3.66%
8/2015	-0.19%	0.904761905	3.46%
9/2015	3.13%	1.349206349	6.70%
10/2015	5.07%	0.939393939	12.12%
11/2015	-1.48%	1.282296651	10.46%
12/2015	2.62%	0.992481203	13.35%
1/2016	0.78%	0.824016563	14.23%
2/2016	-0.45%	1.797160243	13.72%
3/2016	-0.89%	1.128078818	12.71%
4/2016	-0.55%	1.338461538	12.09%
5/2016	0.80%	0.896969697	12.98%
6/2016	-2.95%	1.436363636	9.65%
7/2016	2.90%	1.877777778	12.84%
8/2016	0.53%	1.25	13.43%
9/2016	1.09%	1.1	14.66%
10/2016	-1.08%	1.361904762	13.42%
11/2016	3.70%	1.086834734	17.62%
12/2016	2.77%	1.358695652	20.88%
1/2017	0.86%	2.584980237	21.92%
2/2017	2.59%	1.37037037	25.08%