Can You Make Money Investing In Stocks?

Jesse Wang

Yale University

Advisor: Professor Stephen Slade

May 8, 2017
Abstract

Solving the stock markets has been a centuries old problem that has puzzled many. Yet, despite the heavy competition of the smartest professionals that have recently come into the financial markets, every average joe still seems to want to be able to make money investing. In this paper, I explore whether it is possible to be successful in the stock markets as a retail investor, using machine learning to capture the intuitions of what the retail investor could think about while investing. I investigate some papers that claim it is still possible, and show that with some work, there is still merit to some of their ideas, even as recently as in the financial markets of 2016. These discussed methods include the ANTICOR algorithm, a new way of Pairs Trading, a dividends trade, decision trees with common technical indicators, and using Fama-French Factors. I use features and concepts found in the papers such as certain technical indicators, and the idea of using portfolio optimization rather than optimizing for individual stocks so that information on different stocks can be used as features for each other. I then build on these ideas in a grand neural network regression model that predicts the returns of 34 of the most liquid stocks found in the NASDAQ and NYSE. This model tries to capture all the information that is intuitively there in the papers, and has some more data as input such as currency rates and interest rates. Using very simple execution methods, I present in one case a model that has almost 40% returns when tested on out-of-sample 2016 stock data.
1 Introduction

How to make money in the stock markets has been one of the biggest questions everyone has asked since the introduction of stock exchanges to the world in 1602. (Petram, 2011) It has puzzled some of the greatest minds of our time; even Isaac Newton once tried his hands in the stock markets, and lost what is equivalent to $3 million of today’s money. (Holodny, 2016) With so many minds working at it over the past centuries, the fact that we still have no clear answers on how to invest consistently and successfully is proof that even a genius can’t figure it out.

Or is it?

It’s actually much more complicated than that. The narrative I was told throughout all my summer internships in finance was that without the structural advantages of high trading speeds and low latency of time to get information, as well as a team of smart Ph.Ds working for you and incredible technology infrastructure, investing in the stock market is all just about luck. If there were just 100 people out of the roughly 80 million traders interacting in the market daily who invest in wildly volatile portfolios, we’d expect to hear maybe 50 stories of traders who “did an amazing trade” and became insanely rich. We also commonly see hedge funds do exceptionally for a couple years, only to crash and never been seen again.

On the other hand, we have the likes of Warren Buffett, who once in 1999 said that he could guarantee a 50% return annually on $1 million. (Stone, 1999) We also have some recent literature that claim there still are money making opportunities, (although that begs the question of why the paper was published in the first place – why didn’t the authors just go make money for themselves?).
In this paper, I will first do an overview of some literature; what they are really saying, how good those results actually are, and what we can learn from them. Then I will take these results, build on them, and present my neural network model and the results from that. Finally, we conclude and give some final words on potential future work.

2 Reviews of Some Literature

There are a lot of papers out there exploring the profitability of different models in financial markets, but few that are good. (Again, if they were good, the authors would probably go out to make money for themselves… instead of publishing). However, there are some that capture a lot of the ideas behind some of the most common trading strategies used by retail investors and professionals alike, so I will run experiments to test their merit.

2.1 Anticor

To briefly summarize the algorithm, ANTICOR tries to compute optimal portfolio weightings. At its core, the algorithm bets on mean-regressing movements in the markets – when one stock goes up in one period, you make your portfolio invest less heavily on that stock and more on stocks that are least correlated. It is described by the authors as very aggressive in its reweighting between periods, and only considers one universe of stocks throughout its lifetime. The full details can be found in the paper by Borodin, El-Yaniv, and Gogan.

This paper was intriguing in that it boasted such superior performance – in particular, the “ANTICOR” algorithm described in the paper seemed to perform very well in all the data given. In particular, it earned $238.8 million for every dollar invested in the NYSE data set from 1962 to 1984. It also boasted extremely high returns in the same data sets when you reverse time, and is algorithmically very fast relative to machine learning models, as it didn’t need any training time.
Instead of trying to predict the returns of one stock, it also has the very different framework of trying to optimize portfolio weights, rather than trying to estimate the returns of individual securities. This falls in line with the advice I’ve heard in the industry, which is that if you don’t use multiple symbols in your algorithms you’re missing a lot of information. There’s also been a huge paradigm shift in our financial world recently of much lower brokerage fees for retail investors, which makes a frequently reweighting algorithm like ANTICOR very attractive.

However, as one reads the paper, it becomes apparent that there may be some overfitting. The time periods for the data sets seemed all over the place, and it’s unclear how the sets of stocks from each exchange were chosen for testing. In addition, only final returns are presented, without any presentation of how the algorithm performed over time (i.e. whether there were wild swings in profits over time in any given time period).

So I decided to explore this myself. In order to avoid overfitting, I decided beforehand that I would run this algorithm on the 48 stocks with the largest market cap in the US, on the roughly two year period from Jan. 2015 to Feb. 2017.
Figure 2.1.1: Performance of ANTICOR\textsuperscript{2}, where blue is the ANTICOR model and black is the baseline of a portfolio with constant uniform weighting of all 48 stocks. The y-axis is the cumulative return, and the x-axis is days since Jan. 1, 2015. As in the original paper, we average all time period sizes.

This makes the ANTICOR algorithm seem like a great success (albeit not as great as the original paper boasted). However, if we were to reverse time here, we get a very different picture:
Figure 2.1.2: Performance of ANTICOR®, where blue is the ANTICOR model and black is the baseline of a portfolio with constant uniform weighting of all 48 stocks, except here time is going backwards. The y-axis is the cumulative return, and the x-axis is days since Jan.1, 2015. As in the original paper, we average all time period sizes.

Although the intuition of the algorithm shouldn’t change when time moves backwards, it seems like ANTICOR’s performance takes a major hit. This begs the question of whether there’s something special about time moving forward rather than backward that the original authors didn’t capture intuitively in the paper, or whether this algorithm’s performance just can vary wildly depending on different regimes (which points to overfitting as the explanation for the incredible results reported in the original paper).
Regardless, this algorithm almost certainly has to perform differently in different regimes (in particular, when the market isn’t mean-regressing, the intuition of why this algorithm works doesn’t even hold anymore). So, the idea of this algorithm working no matter tested on what time period might be a bit silly. Instead, a possible way to improve this algorithm is to have an algorithm learn when it is better to reweight portfolios based on a market regime of mean-regression, and when not to. However, the intuition behind the ANTICOR algorithm, as well as the framework of optimizing portfolio weights instead of individual stock returns, was very interesting to explore and good to keep in mind as we build our own model later.

### 2.2 Pairs Trading

There has been a lot of literature around pairs trading, and through the grapevine you can hear about a lot of different hedge funds various different methods of pairs trading, with varying amounts of success. The idea behind pairs trading is to find a pair of stocks that behave very similarly (differently), and to bet on the convergence (divergence) of their returns.

In particular, one paper by Smith and Xu entitled “A good pair: alternative pairs-trading strategies” explore alternative ways to find pairs of stocks that are good candidates to bet on the convergence of in prices. They have two methods, one of which – the cointegration approach – has a fascinating and more theoretical foundation where pairs of stocks are chosen based on the mean reversion of their spread (determined by a test for cointegration). (Another method they use is the distance approach, where the literal difference in normalized prices is measured over some time period and stocks are paired in that way. While this was interesting and had good results, it seemed like it might have less theoretical foundation and bring less new insight into our eventual model). The authors go through extensive efforts to avoid data-snooping (or in other words, overfitting in a way). The authors showed the possibility of excess returns on average of about 2%
trading in the 2000s. The method can be more fully understood in the original paper (TODO citation).

The numbers looked good, and because the authors did so much to avoid overfitting, there wasn’t much to test. However, one thing I looked at was how well these methods actually found pairs of stocks. I ran the cointegration test on all NASDAQ stocks from 2000-2016 with daily average volumes higher than 1,000,000, and found that sometimes I would get seemingly good pairs like

![Figure 2.2.1: Here are the stock prices of AAL (blue) and BRCD (green) over time, from Jan. 2014 (the IPO of AAL) to Dec. 2016. The x-axis is days since Jan. 2014, and the y-axis is normalized stock price.]
Which could work. Perhaps American Airlines (AAL) is somehow heavily dependent on the communication systems of Brocade (BRCD)? Perhaps it doesn’t matter – if there is a relationship then great, but if you believe that pair trades, if random, are a Martingale (i.e. they yield 0 return over time), then you’re fine even if some of the pairs you find have spurious relationships. Remember, you almost certainly want to find new relationships that professionals haven’t already touched on and profited from (where there is no money left to be made), so trading pairs like this could be profitable. I once met an employee from Renaissance Technologies who argued that not all stories are simple, so if you’re sure your algorithm is good and it’s telling you there’s profitable trade, it is a positive expectancy trade.

But other times we would get pairs like this:

![Graph showing stock prices of AAPL and CPST from Jan. 2000 to Dec. 2016](image)

*Figure 2.2.2: Stock prices of AAPL (blue) and CPST (green) from Jan. 2000 – Dec. 2016. The x-axis is days since Jan. 2014, and the y-axis is normalized stock price.*
This latter one seems ridiculous: AAPL has been growing at incredible rates while CPST fell massively in the dot com bubble and never quite recovered, and to think they’re somehow related and mean-reverting in prices as a pair seems somewhat ridiculous. So it seems as if there needs to be some cleaning/refining process in this cointegration approach (the original paper does hint at this, saying there were many pairs to choose from as a result of this method). This is possibly one way of improving on those results.

Nevertheless, the findings of this paper are interesting, and the idea of how stocks interacted in pairs (or in general, together) gave something to think about in building our overall model.

2.3 Dividends

This isn’t actually part of any formal literature – it’s just something I have always wanted to explore. It has always been fascinating to me as to why everyone wants a stock that pays good dividends, because that’s simply just cash coming out of the company (so once you receive a dividend, theoretically a company’s stock price should drop by the dividend amount the next day). Maybe sending out a dividend is a good sign of company health, and that cash flows are good? But can’t it also mean that the company has nothing better to do with the cash (i.e. research and development, or any other investments in the future), and growth has topped out?

Here we explore the simple strategy of buying a stock before the ex-dividend date (i.e. at the close price), and selling it immediately after securing the dividend (i.e. the open price the next day). We build in a proxy for the cost of the spread by paying a premium of the closing (ask price – bid price). The idea is that because successful dividend handouts are a sign of healthy cash flow, a stock will not drop by as much as its dividend amount (which should theoretically happen).
To test this, out of the over 6000 symbols in the NYSE and NASDAQ universe, I considered all the symbols that had an average daily volume of more than 1,000,000. I also took out any dividends that had huge sizes (a payout of more than 10% of the company itself), so to not skew the results one way or another. This left 4726 dividend opportunities in the NASDAQ, and 18412 in the NYSE. Below are the profits from each of these trades (calculated as subtracting how much the stock price dropped overnight from the dividend payout, and dividing by the capital put in to buy the stock. So the numbers in the charts below are overnight returns!)

Figure 2.3.1: Histogram of the returns of this dividend trade on the NYSE
Amazingly, the NASDAQ trades averaged a 0.131% return per trade, and the NYSE trades averaged a 0.110% return per trade, for an overall average of 0.114% return per trade. Taxes will make this profit go down, but it will still stay positive, because we’ve already accounted for trading fees very accurately.

This seems odd, so one might wonder if this opportunity is going away. With brokerage fees going down over time, this does seem to be a dwindling opportunity as more people take advantage of it: in 2016, the overall average return was only 0.030% over 2640 trades on both exchanges. However, this still seems like it’s quite the opportunity because of how easy it is to execute, if one has the capital to ride out the risk.
2.4 Technical Indicators

Technical indicators have been at the heart of day trading for decades now. Many of these indicators, such as the 12-day exponential moving average of the price, or Bollinger Bands (how many standard deviations away a stock’s current price is from its moving average) try to capture the visual intuition of what a human trader might think about looking at price charts, or other relevant stock data. Although there is no evidence of true theoretical foundation behind these very visual statistics, it is widely believed that the very intuitiveness of these indicators can capture some information about a stock. Regardless, a lot of stock market participants look at them, which may make it self-fulfilling that technical indicators are very important. (An overview of the technical indicators I use throughout this paper can be found in Appendix B).

In *Automated Trading with Boosting and Expert Weighting*, Creamer and Freund test the effectiveness of a vast array of technical indicators, the ones they claim are most popular, and implement a trading system using Alternating Decision Trees. The authors use 100 randomly chosen stocks from the S&P 500 for this study, and without trading fees, boast up to a 15% return out-of-sample in the period from mid-2003 to the end of 2004. (Creamer and Freund, 2009) The full details can be found in the paper.

It is worth noting, however, that within this time period, the S&P 500 rose roughly 30%. So maybe this trading system wasn’t so good if you adjust it to market returns. Regardless, to capture the explanatory power of the model, I took roughly a generating subset of the technical indicators from Creamer and Freund’s paper (there is clearly a lot of multicollinearity in the original paper) as well as some others, and used a Random Forest Regression to try and predict stock returns. This model trained on data of 34 highly liquid stocks from 2000-2015 and tested on
the data for 2016. (Appendix C shows how I chose the 34 highly liquid stocks and what they are, and Appendix D shows how I got profits from the returns).

The reason I chose to use Random Forest is because of the way it captures the behavior of how traders use technical indicators. For most technical traders, they usually look at an indicator (say Bollinger Bands) and say that if the value passes through some threshold, then they would perform some action (e.g. if Bollinger Bands exceed 2, then sell). At the core of the Random Forest, it consists of many decision trees that are essentially trying to optimally find a similar cutoff for a discontinuity in behavior. Here is how this model performed in 2016:

![Figure 2.4.1: Blue is my model, Green is the baseline uniform portfolio](image-url)
As we can see, this model actually outperformed a baseline uniform portfolio (a portfolio which consists of all the highly liquid stocks, but in constant uniform weightings). In particular, the positive drift of the model’s results seems very stable. However, no trading fees have been considered here, (i.e. taxes, bid-ask spreads), which is very likely to bring the profitability down, perhaps lower than the baseline. Regardless, this is strong evidence that, even with the perhaps suboptimal weighting mechanism of my model’s portfolio (described in Appendix D), there is a lot of signal to be found in technical indicators, even as recently as 2016.

5 Fama French

Although the Fama-French paper on the 3-factor model is perhaps not directly applicable, this section probably wouldn’t complete without this famous paper. In this paper, Fama and French describe five factors that could affect the stock and bond markets in the aggregate, and in particular how they affect companies of different market caps. They find very strong explanatory power in the long-term horizon. The full details can be found in the paper. (Fama and French, 1993)

These factors don’t say anything about individual stocks, as most of the other features I use do, but can have some explanatory power over the whole market of large market cap stocks. So I will add these factors to my model.

3 Data

The data used in this study was mostly taken from Wharton’s WRDS database. We used daily stock prices of all the ticker names in NASDAQ and NYSE from 2000-2016, although in the actual studies and models we usually only used symbols that were nontrivial in some way (say having a volume of over 10,000,000 on average over some time period, among other criteria for example. The exact criteria is described in Appendix C). In the same time period, we also use the
normal basic information such as daily volume or closing bid/ask prices, interest rates, some data on currencies (which, intuitively, could affect these huge symbols we’re looking at since they probably have operations overseas, and are vulnerable to currency rates), and the Fama-French factors.

4 The Overall Model

To build on the literature, we use some of the ideas that were intriguing and the data we have to build an immense feature space, and use a neural network to try building a profitable trading model. The assumption here is that the neural network is both a lower bound on what a retail investor could potentially do (I’ve used nothing that isn’t public information), and an upper bound on what retail investors usually think of (since it could probably capture the intuition of whatever the retail trader is thinking, so long as I’m feeding in the same data).

5 Results

I built and tuned a neural network to predict the returns of the 34 stocks (described in Appendix C). I trained on data from 2000-2015, and tested on the 2016 data. As with any financial data, the results are very noisy. Here’s a scatter plot of the returns I predicted for AAPL against the actual realized returns:
Figure 5.1: Returns by my model vs. realized returns of AAPL. It’s hard to tell from this if there’s any actual real signal.

So we need a different approach to measure the success of this. Using the approach (again described in Appendix D, as we did for the technical indicators section of the literature reviews) to convert these predictions into a profit over time, we get this:
We see we get some positive returns over time (approximately 11% returns, which actually is very good). If we compare this to the Random Forest model earlier or the baseline constant uniform portfolio however, it seems bad. But this is due to the underlying market movements: from the section with the Random Forest, it seemed like that model was very dependent on the underlying stocks to be going up to have exceptional returns (the two lines in that figure followed each other very tightly). However, we’ll see that there is strong evidence the neural network model is robust to different regimes, since it still does well, as we’ll see, even when it’s forced to only buy or only sell stocks.
Figure 5.3: Returns over time. Here my model is restricted so it can only buy stocks. Blue is my model, and Green is a uniform constant portfolio of the 34 highly liquid stocks (i.e. the baseline).

As we can see, when the neural network is forced to only buy stocks (which is actually a constraint faced by a lot of retail investors), the neural network follows the baseline portfolio very closely (but also outperforms it).
Figure 5.4: Returns over time. Here my model is restricted so it can only sell stocks. Blue is my model, and Green is a uniform constant portfolio of the 34 highly liquid stocks (i.e. the baseline).

Here instead we see that the neural network still performs very well, considering that it must sell securities in a strongly bull market. The baseline portfolio gains about 25% in value, whereas the only selling neural network only lost about 8%.

Putting these two pieces of information together, we can expect that the model is making positive expectancy bets on both its buying and selling, which implies it is consistently making expected profits under any kind of bull or bear market regime.
Interestingly, if you believe that the markets generally go up over time, as we’ve seen in the baseline, (which is contrary to the neural network’s idea that perhaps the market is a Martingale that stays constant period over period in expectancy), then maybe you should go with the model that only buys. This would generate a return of almost 40% in 2016. This is actually not a far-fetched belief, with the S&P 500 almost always consistently rising. Everyone is in some ways long the economy (everyone want the economy to do well, so they work towards though), and so there’s probably always some natural force pushing all stock prices generally up.

6 Conclusion

Given enough data, thought, and computational power, it does seem like it is possible to make positive returns in the market still, even now when the financial markets have become so competitive with all the professionals. Many of the pieces of literature reviewed which claimed to generate positive returns seemed to, at least with some work or reworking, have some merit and could possibly have made money even as recently as 2016. In particular, the ANTICOR algorithm, certain combinations of technical indicators with decision trees, and trading dividends, all seemed to potentially be profitable methods. Putting these all together, I was able to build a robust neural network that was also profitable, generating a return of roughly 11% in 2016 (or almost 40%, if one is, like retail investors, constrained to only buy stocks, or believes that markets generally go up).

How to make money in the markets is generally a hard problem, but possible to solve. Hopefully this gives the reader the perspective that stock trading is not all just luck, and maybe, with some hard work, you could be the next Warren Buffett.
7 Future Work

There is a lot to be done to improve this. More data could be added, such as company financials, options prices, market sentiments, and the trades of successful traders (such as Warren Buffett’s most recent trades) (data input was unfortunately constrained by the computational power available to me). We could also perform more portfolio optimization when we calculate the profitability of stock return predictions, taking things like standard deviation of returns into consideration, and maximizing for something more sophisticated like Sharpe Ratio (expected return divided by standard deviation of returns).
Works Cited


Appendix A: Running the Actual Code

To run the code, go to the folder “RealModel”. Then open getNames.py to change the parameters of how to choose your set of NYSE/NASDAQ stocks. Then run

```python
python2 getNames.py

python2 combineTickers.py

python2 combineAll.py

python2 model1.py

python2 profits.py
```

Each of these steps are customizeable. If you are happy with your universe of stocks, you can just run the last two steps. The experiments with the other papers are in the other folders.

Appendix B: Parameters of Neural Network

Neural networks have a lot of parameters, so it is very hard to tune. However, after some tinkering, the model I built had a hidden layer of 68 nodes in between the input and output layers, with a sigmoid activation function. It was run with a batch size of 32, and 300 epochs, with a cross validation split of 0.2. I tried using deeper nets, but that seemed to start overfitting the data, and performance suffered.

Remember that the test of 2016 data was done out of sample, and so the tinkering of hyperparameters to the neural network cannot have caused overfitting.

Appendix C: The Ticker Names and How They Were Chosen

Here’s the list of all the symbols:
From NYSE:

AA, BAC, BSX, C, CHK, DOW, F, FCX, GE, GLW, HAL, JPM, KEY, KO, MRK, MU,
ORCL, PFE, RIG, SWN, TSM, WFC, WFT, XOM, XRX

From NASDAQ:

AAPL, AMAT, CMCSA, CSCO, EBAY, GILD, INTC, MSFT, QCOM, YHOO

These symbols were chosen because in 2015, they had an average daily volume of over
10,000,000, had an average bid-ask spread of less than $0.03, had an average price of over $10,
and they all existed throughout 2000-2015.

**Appendix D: Going from Predictions to Profits**

Because this was not the focus of the thesis, this process was done very simply. For every
day of the predicted returns in 2016, I gave weights to each of the 34 stocks that were weighted
linearly according to the distance between the predicted returns and 0. (Of course, those with
negative predicted returns were given a negative weight) and I normalized the weights so that the
sum of the absolute values was 1. From this, a portfolio return was calculated every day, showing
profits.

If this were optimized to take standard deviation into consideration to maximize some kind
of portfolio risk-adjusted return, this would have probably vastly improved performance.

If this were optimized, the results would have been even better.