CS230

# Predicting stock price dynamics using stacked GRU's and LSTM's

**Raul Girbal**
Department of Mathematics
Stanford University
rgirbal@stanford.edu

**Glib Stronov**
Department of Economics
Stanford University
gstronov@stanford.edu

## Abstract

In this paper, we aim to explore the predictive power of different Recurrent Neural Networks (RNN) on the S&P 500 daily closing price over different time periods. Recognizing the importance of price prediction for financial markets, we investigate how well LSTM and GRU RNN's stacked in different architectures predict prices for the S&P 500 stock index. Our data set includes daily S&P 500 prices from 2000 onward, and manually selected features that include the classic OHLC feature set as well as manually chosen and tailored technical indicators as well commodity and economic indicators. Our stacked LSTM and GRU model achieved the lowest error rate, an average of 5% difference in returns over different time periods tested.

## 1 Introduction

One of the most famous results in finance is the Efficient Market Hypothesis, which states that it should be impossible to predict short-term price fluctuations in the stock market. It assumes that investors are perfectly rational. However, recent research suggests that it may not be the case. There are a lot of restrictions on investors and recent research points to a variety of psychological biases that investors exhibit, which suggests that it might be possible to distinguish patterns in market behaviour and thus predict future prices.

In this paper, we will attempt to create such a model to predict price movements of S&P 500. The Standard & Poor's 500 (S&P 500) is an index formed of the 500 companies listed in the NYSE and NASDAQ with the largest market capitalization. It can serve as a proxy for the performance of the US stock market and can say a lot about the performance of the US and world economic system as a whole. For this purpose, we will be using Recurrent Neural Networks (RNN's) with unique architectures for "recalling" longer term information. Using RNN's, this project can shed light on how different stacking methods affect performance as well as analyzing how well our models can predict longer term price movements of the S&P 500.

To be more specific, we will analyze daily price information from the S&P 500 and will be predicting the closing price of the index using information up to the closing price of the day before. We will use the classic OHLC (Open, High, Low, Close) price features, as well as commonly available indicators such as the Relative Strength Index (RSI), the Moving Average Convergence Divergence indicator, daily volume, West Texas Intermediate (WTI) price, and European Brent prices. Using this feature space, we analyze LSTM's and GRU's stacked together as well as dropout implementations and see how they perform compared to a linear regression and other standard vanilla models on the data.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

## 2 Related work

Some of the most popular price prediction techniques used by researchers today are Linear Regression, Support Vector Machines, Random Forests, Case-based Reasoning, Neural Networks, and even Clustering Algorithms.

Out of these techniques, Linear Regression performs the worst by far. Most researchers conclude that a simple Linear Regression model can't capture the complex non-linear nature of stock markets. Case-based reasoning and Neural Networks approaches outperformed the other models by a significant margin. Case-based reasoning was good at screening out the potential stocks and influential factors to use in its predictions and using a dynamic time window allowed it to make even more accurate predictions by using most relevant information. Neural Networks, specifically Convolutional Neural Networks and Recurrent Neural Networks, were good at dealing with time series and picking up the prevalent market trends. One major issue with some of the Neural Network models was that they seemed to over-fit the data in many cases, so researchers had to be careful during the training period to make sure their models generalized well.

When it comes to the data used in these models, researchers tend to heavily rely on technical data, such as price or trading volume. The main advantage of using technical data instead of fundamentals data is that technical data is available on a daily, or in some cases even hourly, basis, while fundamentals data is only available every quarter, limiting the amount of data they can use.

## 3 Dataset and Features

We were able to obtain daily data for every trading day on NYSE from January 2000 to September 2018, which is equivalent to 4679 data points. Our dataset contains 10 distinct features – Opening Price, High Price, Low Price, Closing Price, Trading Volume, Moving Average Convergence Divergence, Moving Average Convergence Divergence Signal, Relative Strength Index, West Texas Intermediate, and European Brent. The first 4 features directly relate to S&P 500 price. Since we are trying to predict what the price of S&P 500 will be after a few days, it makes sense to include the price as a feature. We also chose to use 4 price indicators instead of just 1 to observe how much price volatility there is and if that has any predictive power on the future price.

The next feature we use is trading volume. It is defined as the total amount traded in dollars within a single trading day. We include this feature because volatility can be an indicator of rapid change in the price, reflecting increased interest and potential re-evaluation by many investors at the same time.

Moving Average Convergence Divergence is a trend-following indicator that shows the relationship between two moving averages of the same security but over a different length of time. In our case, it compares the moving average of S&P 500 over the past 12 days and the moving average over the past 26 days. The assumption that goes into this indicator is that the longer-term average represents a more stable growth rate of the security, while the shorter-term average represents the prevalent market trend and that this trend is going to persist for some time. If the short-term growth rate is above the longer-term average, we assume that the stock has been growing faster than usual and that this trend indicates a buy signal. If the short-term growth rate is below the longer-term average, we assume that the stock is under-performing, which indicates a sell signal.

Moving Average Convergence Divergence Signal measures the changes in changes of consecutive Moving Average Convergence Divergence estimators and allows us to forecast when the market trends are going to revert, therefore solving the issue of time lag. We calculate Moving Average Convergence Divergence Signal according to this formula:

$$MACD\,Signal_t = .2 * MACD_t + 0.8 * MACD\,Signal_{t-1}$$

The last technical indicator is the Relative Strength Index. It also measures the change in momentum of the price change but it separates gains and losses, which is not the case in Moving Average Convergence Divergence. This difference in calculation allows Relative Strength Index to better identify overbought and oversold conditions and not just the momentum shifts. Relative Strength Index can be calculated like this:

$$RSI = 100 - \frac{100}{1 + \frac{AverageGain}{AverageLoss}}$$

Finally, we also include oil prices as features. More specifically, we include West Texas Intermediary and European Brent. Oil is an incredibly important input for the U.S. economy and its prices can significantly change within a short span of time. Given its importance, we think that including oil prices can be a good source of additional information about the market trends. The reason why we include both West Texas Intermediary and Brent is because the difference in growth rates can help us identify positive or negative effects on the U.S. economy as a whole. Since the U.S is a major exporter of oil, increased in the price of exported oil favors the companies that are in that sector, which is what fluctuations in West Texas Intermediary price should measure. At the same time, Brent prices can represent a global trend in input prices that U.S. companies outside of the oil sector depend on.

# 4 Methods

As previously stated, our objective is to predict the price changes of the S&P 500. We do that by attempting to predict the returns of the index over different time periods. We use over four thousand data points from the S&P 500, from the year 2000, and several related features to predict the return on the index 1, 2, 3, 4, 5, 10, and 50 days after. For our stacking test we use a more sparse day selection of 1, 3, 5, 10, 25, and 50 days. We split our data so that 90% is in our training set and 10% is in our test set. We have approached the problem of predicting S&P 500 index prices over 1-5, 10, and 50-day periods by using Recurrent Neural Networks with a dedicated structure for recalling longer term information. Specifically, we used LSTM's and GRU's and stacked them together in different orderings.

## 4.1 Leveraging the structure of RNNs

While there are metrics that rely on past price movements to identify overbought or oversold conditions or momentum, they are unlikely to be sufficient on their own to generate good predictions. This information is available to all investors and is already likely priced in. Since we want to discover non-obvious patterns that other players in the market may be missing, we want to structure our model so that it can effectively incorporate past data into its predictions.

For this purpose, Recurrent Neural Networks seem like a good fit because they were created with the purpose of picking up patterns in a series of data points instead of treating each case independently of others. Specifically, we want to use LSTMs and GRUs to be able to look at both long-term and short-term patterns to determine the stock price.

## 4.2 Choosing the depth of the network

In order to create a neural network that accurately recalls information from macroeconomic variables and does not over-fit to our data, we want to achieve an appropriate balance between depth and out-of-sample performance. For our project we decided to test multiple stackings of GRU's and LSTM's, along with an addition of dropout, in order to see what model would be most robust. We define a block to be a combination of an LSTM module followed by a GRU, both with the same width. We decided on this basic block structure because our testing suggested that an LSTM-GRU structure would function better than a GRU-LSTM structure (with an average error of 0.09 vs 0.11).

Using this basic block structure, we wanted to understand how the number of blocks, added with a 20% dropout rate in between each block, affected our models performance in using 20 days of data to predict the price of the S&P 500 1, 3, 5, 10, 25, and 50 days ahead. In our testing we look at how 1, 2, 3, 5, and 10 stacked block networks perform in our testing. We will compare these results to the standard Linear regression and 3 other models detailed in Figure 1.

# 5 Experiments/Results/Discussion

We tested the predictive power of the four models listed above by testing their ability to predict prices 1, 2, 3, 4, 5, 10, and 50 days ahead. Furthermore, we tested how different number of blocks affect the predictive power of our network. Our results were promising as they suggested that the models, especially those with an intermediate level of stacking complexity, were learning higher level

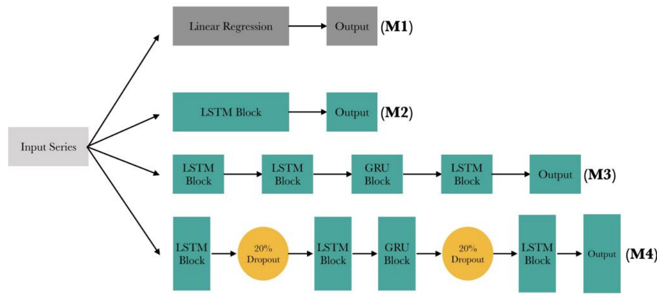


Figure 1: The 4 different models we tested.

| | Model | | | |
|---|---|---|---|---|
| Days | M1 | M2 | M3 | M4 |
| 1 | 0.243 | 0.026 | 0.022 | 0.023 |
| 2 | 0.630 | 0.041 | 0.040 | 0.044 |
| 3 | 0.185 | 0.076 | 0.052 | 0.061 |
| 4 | 0.128 | 0.084 | 0.100 | 0.092 |
| 5 | 0.167 | 0.115 | 0.100 | 0.122 |
| 10 | 0.280 | 0.262 | 0.195 | 0.200 |
| 50 | 0.836 | 0.917 | 0.656 | 0.558 |

Figure 2: Table showing performance of the different standard models.

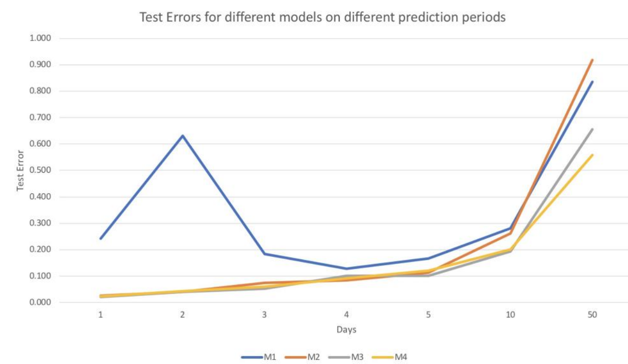


Figure 3: Test Errors of the different models over different prediction periods.

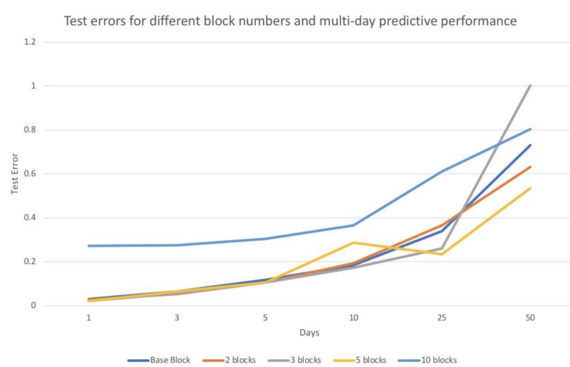


Figure 4: Test Errors of the different block numbers over different predictive periods.

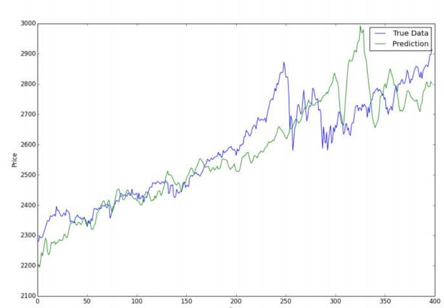


Figure 5: (M4) LSTM + GRU stacked with dropout 50-day predictions

information from the data.Figure 2 shows the final test error rates of each model on around 400 days of S&P 500 data and Figures 3 and 4 show test errors for the different models.

Looking at Figures 2 and 3 we can gather a few interesting patterns. The first is a trend we expected: the fact that at first linear regression performs far worse than our other models and then converges as we attempt to predict more and more days ahead. This reflects the fact that as we attempt to look further into the future, the information in the data relevant to price prediction decreases and more stochastic events can occur that dislodge the price from the trends that our model expected to hold. A second trend to note is that up until 5-day prediction our model without dropouts outperforms our model with dropouts. This is probably due to the fact that 4-day predictions can rely more on the recent information from the price data, which LSTM can fit very well. Thus, in the specific times where the model is wrong, it is when there are more extreme events where other models also make equivalently large mistakes. After 5 days we see that our model with dropout begins to outperform the version without, and especially in the 50-day prediction scenario we see that the model with dropouts outperforms all other models by at least 12% in terms of accuracy. This points to the fact that the model with dropouts has blocks that are better trained to deal with stochastic volatility and picks out more robust sub-features from the data with which to base predictions on.

Additionally, we note that there are interesting trends in Figure 4 as well. We note that the 10 block model performs worse than all other models, probably because the additional blocks are only causing more over-fitting despite the inclusion of 20% dropouts between each block. Secondly, we note that the ideal blocking number seems to be somewhere in between 3 and 5 blocks, especially given the large divergence in performance in the 10 and 50 predictions.

Overall,we can also observe that our models are able to deduce a lot of information about how markets work. We note that 50-day price predictions in Figure 5 are surprisingly accurate. We see that not only does the model pick up on trend data, but it is also picking out interesting patterns from the data which it is then representing in its price predictions. The distance between the two peaks is roughly the lag of 50 days which it takes the model to receive new information. We note that the model after receiving the information of the decline makes the necessary adjustments, but then it predicts that the price will skyrocket back up and then decline once more. This suggests that the LSTM + GRU with a 20% dropout M4 model is picking up on market overreactions and volatility swings in a semi-representative way. This is extremely promising given that we would like our models to better understand trading movements and the psychology of the market.

## 6 Conclusion/Future Work

In this project, we successfully implemented a Linear Regression Model as a baseline and 3 variants of Recurrent Neural Networks that were able to predict S&P 500 returns over a short period of time into the future. Of all models, we found that the model that combines LSTM layers with GRU layers but does not use dropout layers is the most accurate over short-term periods, but the model that includes dropout layers has the best performance over longer prediction periods. We also tested that different stacking architectures have very material affects on the accuracy of our models. Our results point to the fact that the optimal balance between stacking depth and accuracy seems to be between 3 and 5 consecutive LSTM-GRU blocks for this amount of data.

Unsurprisingly, we observed that the accuracy for all models goes down when the length of the prediction period becomes larger, but our models still retain predictive capacity when they are predicting the price in 50 trading days. We also found that neural networks outperform linear regression for any prediction time period, but the difference in accuracy goes down with the length of the prediction period.

This supports our original belief that neural networks are well-suited to pick up non-linear market trends and leverage them to make accurate predictions about the short-term changes in the price of S&P 500.

## 7 Code

Our code can be accessed at https://github.com/gstronov/cs230-project

## 8 Contributions

The project was created by Raul Overdijk Girbal and Glib Stronov. Raul and Glib worked on all parts of the project together, so that both of them would know how the model is organized and how it is performing. However, Raul focused more on implementing the model and performing quantitative analysis, while Glib focused on data collection and pre-processing along with performing qualitative analysis.

## References


[1] Aungiers, Jakob. LSTM Neural Network for Time Series Prediction.

[2] Dissanaike, Gishan. Do Stock Market Investors Overreact?

[3] Shen, Guizhu, et al. Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions.

[4] Schuermann, Lucas. Notes on LSTMs for Time Series Prediction in Finance

[5] Helmbold, David P., and Philip M. Long. "Surprising Properties of Dropout in Deep Networks." Proceedings of Machine Learning Research

[6] Masoud, Najeb. "Predicting Direction of Stock Prices Index Movement Using Artificial Neural Networks: The Case of Libyan Financial Market." British Journal of Economics, Management & Trade, vol. 4, no. 4, pp. 597–619.

[7] Larsen, Jan Ivar. "Predicting Stock Prices Using Technical Analysis and Machine Learning."

[8] Chang, Pei-Chann, et al. "A Neural Network with a Case Based Dynamic Window for Stock Trading Prediction." Expert Systems with Applications, vol. 36, no. 3, 2009, pp. 6889–6898

[9] Chang, Pei-Chann, et al. "Integrating a Piecewise Linear Representation Method and a Neural Network Model for Stock Trading Points Prediction." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 39, no. 1, pp. 80–92.

[10] Behl, Smarth, et al. "A Machine Learning Based Stock Trading Framework Using Technical and Economic Analysis."

[11] Rehan Raja, and Chhapra, and Imran Umer. Behavioural Biases and Investor Decisions.