CS680 Project: Stock Market Prediction

Robert Allan; r5allan@uwaterloo.ca

Abstract

Stock market prediction is a complex problem and one that is a common area of research for machine learning. For comparison of Machine Learning methods in this area this paper compares the performance of Support Vector Regression, Multi-Layer Perceptron, one dimensional Convolutional Neural Network, Long Short Term Memory Neural Network (LSTM) and Bi-directional LSTM. The data selected are stock price and volum data for nine stocks from the S&P 500, the data was normalized for each stock and MACD, RSI and EMA were calculated. After hyperparameter tuning the models were compared in terms of test MSE and MAPE. We found that the Bi-Directional LSTM consistently outperforms the other methods presented on the test set. Since the complexity of this network was not high it could also be used with additional data outside price and volume related measures in the future as a possible tool for short term trading.

1. Introduction

Stock market prediction is a lucrative field for investors and researchers who want to generate excess returns on investment. But historically it has been a difficult problem to solve. Even as early as 1933 research into the field in Cowles 3rd 1933 showed that the market could not be predicted. Following this efficient market theory in Fama 1970 also implies that it is not possible to beat the market . Despite this it is still a very active area for research, particularly more recently in the field of machine learning. Determining the timing to enter into a trade or exit a trade is considered a particularly valuable skill among traders, and accurate predictions of future prices are a valuable tool for trading.

There are a number of different ways to approach this in terms of data used and to define what is to be predicted and the forecast time frame. There is the more common technical analysis approach which usually uses combinations of price and volume data often with the same metrics that traders have traditionally used as seen in Henrique, Sobreiro, and Kimura 2018,Ince and Trafalis 2007, and also the more broad approach which uses other data sources to augment price data (such as sentiment analysis of business news). Likewise there are forecast time frames that are short term (seconds to $<$ 1 year time frame), medium term (1-2 years time frame) and long term (2+ years time frame). The most common problem studied is short term, it is also one of the easier problems to tackle due to less variance. For our purposes we are curious to predict the close price of a stock one day into the future (i.e. predict at closing price of day 2 at the close of day 1) using only price and volume data and technical indicators derived from this data for the particular stock. The data selected is a dataset of daily close and volume of companies listed on the S&P 500 from 2010 to the end of 2016.

2. Techniques

The first technique that was considered was the support vector regression (SVR) used in Henrique, Sobreiro, and Kimura 2018 and Ince and Trafalis 2007 . It uses an Support vector machine to obtain optimal regression function (1).

$$f(x, w) = w^T x + b \tag{1}$$

thus it seeks to minimise the following function

$$\min \frac{1}{2}\|w\|^2 \tag{2}$$

Subject to

$y_i - wx_i - b \leq \epsilon$

$wx_i + b - y_i \leq \epsilon$

commonly introduced kernels used are the linear, radial and polynomial equations (3),(4) and (5) respectively .

$$K(x_i, x_j) = x_i^T x_j \tag{3}$$

$$K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}, for \gamma > 0 \tag{4}$$

$$K(x_i, x_j) = (x_i^T x_j + 1)^d \tag{5}$$

This function has been used to predict index prices in Lu, Lee, and Chiu 2009 and outperforms a baseline random walk. It will be a useful technique for comparison as it is not based on neural networks, but can still perform quite well. For the feature set articulated later in the paper, its computation is relatively fast especially when compared with some of the more complex techniques examined.

Neural networks are now becoming one of the most common techniques for many machine learning problems. In Di Persio and Honchar 2016 the authors introduce 3 different techniques based on neural networks to predict stock prices, a Multilayer Perceptron (MLP), a 1 dimensional Convolutional Neural Network (CNN) and an Recurrent Neural Network (RNN) based on LSTM cell architecture. All three of their networks were implemented in Tensorflow Keras library in Python. Their MultiLayer Perceptron is quite a simple Neural Network by current standards and, in the case of Di Persio and Honchar 2016, contained 2 full connected hidden layers of 500 and 250 neurons respectively, using a high dropout of 0.75 and 0.6 for both layers. The MLP was the least effective of the three neural networks tested, but it is a useful comparison point for the performance of a simplier neural network with no time series specific architectural features. A 1 dimensional CNN is also proposed, comprising 2 convolution and max pooling layers both with a filter size of 64, filter length of 2 and pool length of 2, before being passed to one dense fully connected hidden layer of 250 units.

The Convolution Neural Network is often proposed for models which have image data, but when used in a one dimensional fashion it can also be used for processing sequence data such as time series data. They generally rely on stacks pairs of convolution layers and pooling layers in the manner of Figure 2. They found the CNN to be the most effective of the 3 surveyed however the margin was small (0.2491 for CNN MSE vs 0.2498 for RNN (LSTM) MSE).

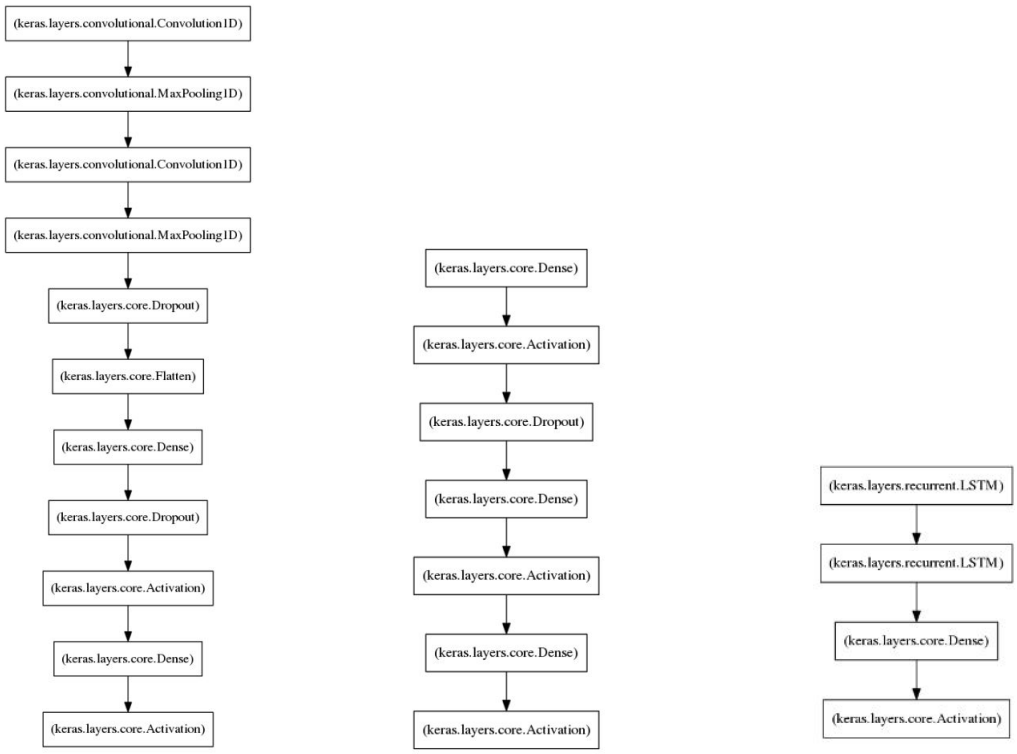

Figure 1: CNN, MLP and LSTM network architecture proposed in Di Persio and Honchar 2016

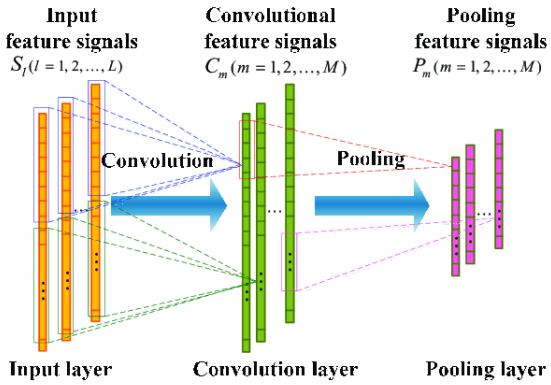

Figure 2: 1d CNN Convolution and pooling. Yang et al. 2019

Lastly they introduced a RNN based on a long short term memory (LSTM) architecture. This is a recurrent network using an Long Short Term Memory cell in its architecture as per Figure 3. It can process data sequentially but performs better at storing long term patterns than the traditional RNN architecture. They used a relatively simple LSTM design

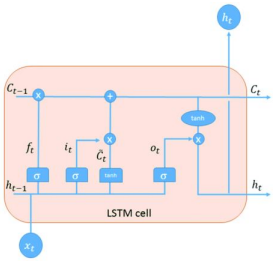

Figure 3: LSTM Cell. Rundo 2019

with just two hidden layers of LSTM cells with 100 neurons each. Diagrams of the exact architectures proposed can be found in Figure 1 The RNN architecture was designed for time series problems such as stock prediction so even though it lost to the CNN technique in their comparison, we believe that this technique is worth including for comparison.

Lastly we will evaluate the Bidirectional LSTM neural network as introduced in Jia et al. 2019. This is a special form of LSTM network previously mentioned whereby a forward LSTM layer processing the timeseries from 1 to t and another backwards layer performs the LSTM calculation from t to 1 the outputs of the forward and backward LSTM layers and then combined in an output layer a diagram of this cell architecture can be seen in Figure 4 . This will potentially make better use of historical information in time series such as

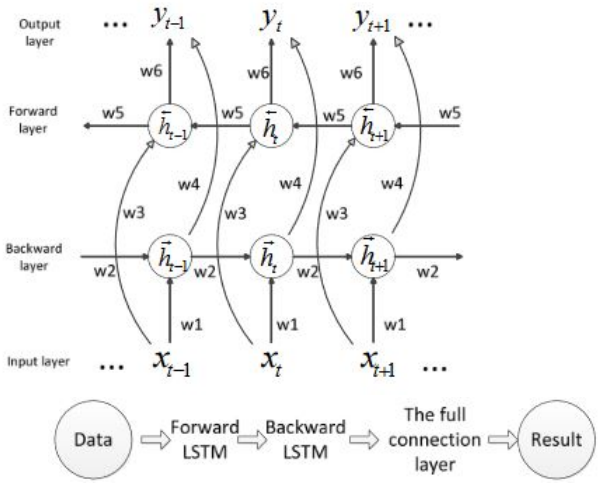

Figure 4: Bi-directional LSTM cell architecture as introduced in Jia et al. 2019

stock data. In their paper they see an improvement of 19.4% on mean average error vs a traditional uni-directional LSTM. For this reason I will include it in the comparison as it may be one of the best performing models available for time series data such as stock data.

The data selected is a dataset of daily close and volume of companies listed on the S&P 500 from 2010 to the end of 2016. In addition to this we will also create some traditional stock metrics such as Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI) and Exponential Moving Average (EMA) as used in Ince and Trafalis 2007

and Henrique, Sobreiro, and Kimura 2018. Volume and Close Price for each stock will be normalised between 0 and 1 over the dataset, as this will improve the accuracy of SVR in particular. We will use a 90 day window prior to the target close value as our training data, and will split the data into a training and test set with the last 35% of the data being for testing.

3. Empirical Evaluation

The primary metrics used in model comparison were Mean Squared Error (MSE) and Mean Average Percentage Error (MAPE). For comparison only the error on the test set was considered. A number of comparisons on a single stock were undertaken to compare hyper-parameters of a given model. In terms of the MLP 3 hidden layers and relu activations were used. The optimum width of the hidden layers was found to be 256 (out of 256, 512 and 1024 widths tested see Table 1) . As early stopping criteria looking at test MAPE was used to prevent overfitting rather than dropout. In the one dimensional CNN two 1d

Hidden layer units	Test MSE	Test MAPE
256	0.001029	3.602534
512	0.001144	3.848567
1024	0.001029	3.535700

Table 1: Relative performance of MLP hidden layer widths

Convolution and Max pooling layers were used in the same manner as show in Figure 1. However during testing it was found that removing the fully connected hidden layer after flattening increased the performance of this network so it was removed. A filter size of 64 and kernel size of 1 was found to perform best across a few different stocks.

In the uni-directional LSTM the same architecture as shown in Figure 1 was used. A number of hyper-parameters were tested on a stock's dataset. In this case dropout was included but as with previous models early stopping was also implemented to prevent overfitting. We can see from Table 2 that the best performing model was the one with LSTM unit size of 512 and a dropout of 0.1. In the bi-directional LSTM the same architecture

LSTM Size	Dropout	Test MSE	Test MAPE
512	0.1	0.000530	2.400171
256	0.1	0.000671	2.682860
512	0.2	0.000673	2.704331
256	0.2	0.000626	2.803674
512	0.5	0.001041	3.332096
256	0.5	0.000798	3.029570

Table 2: Performance of different hyperparameters on a single stock using one way LSTM

as shown in Figure 1 was used but this time with the LSTM layers using the bidirectional wrapper as implemented in Tensorflow Keras. Again hyper-parameters were tested in the

same way as the unidirectional LSTM. We can see from Table 3 that the best performing model was the one with LSTM unit size of 512 and a dropout of 0.1.

LSTM Size	Dropout	Test MSE	Test MAPE
256	0.1	0.000488	2.421743
256	0.2	0.000641	2.733346
256	0.5	0.000990	3.605036
512	0.1	0.000351	1.909292
512	0.2	0.000403	2.106355
512	0.5	0.000741	2.964597

Table 3: Performance of different hyperparameters on a single stock using Bi-directional LSTM

For the SVR models the Radial and Polynomial basis functions were tested alongside linear kernels, they did not appear to result in an improvement of performance so only a linear kernel SVR has been used, a regularisation parameter of c = 1 was used. The SVR was implemented using the sklearn package. All other neural network based models were implemented in Tensorflow Keras in Python, which allowed for GPU acceleration.

Now that the models and their hyper-parameters have been selected we will compare the performance of the various models. On the assumption that stock patterns may differ between sectors, nine stocks from five different sectors are proposed. They are: DOW - Dow Chemical and BA - Boeing from the Industrials sector, AAPL - Apple Inc. and NFLX Netflix from the Information Technology sector, WFC - Wells Fargo and JPM JP Morgan Chase from the Financial sector and PFE - Pfizer Inc. and JNJ - Johnson and Johnson from the Health Care Sector, and AMZN - Amazon from consumer discretionary sector. Each observation row comprised of 90 days of normalized close prices, EMA (10 days) datapoints, MACD, RSI and normalized volumes.

For techniques such as CNN and LSTM methods that were more suited to time series data these were presented as a 3 dimensional dataset, but for the remainder such as the MLP and SVR these were "flattened" to become 2 dimensional datasets. For each stock training was done separately on the 1054 training observations (of previous 90 days of data prior to the target close price) and target close prices. Test data comprised of 526 observations.

4. Results and Conclusion

We can see from Table 4 and Table 5 that the LSTM models perform the best on average, but surprisingly the SVR with a linear kernel is very close to the performance of the LSTM depending on the stock in question. The bi-directional LSTM comes out on top for almost all tests, but there is a price in run time, as we can see from Table 6 it is 5-6 times slower than the SVR or uni-directional LSTM models.

Another interesting performance measure is the percentage of the time the regression prediction chose the right direction for the stock movement from the future close from the previous close. We looked at this metric for the predicted future close price of our neural

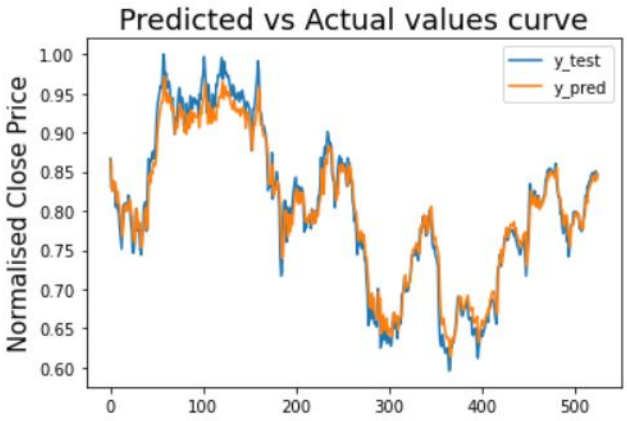

Figure 5: AAPL Stock Actual Close Price vs Predicted for Bi-Directional LSTM

Stock Ticker	CNN MSE	MLP MSE	LSTM MSE	BD LSTM MSE	SVR MSE
DOW	0.003023	0.002753	0.000493	**0.000422**	0.000664
AAPL	0.002999	0.001953	0.000529	**0.000295**	0.000507
WFC	0.001876	0.002069	0.000615	**0.000396**	0.000462
PFE	0.001324	0.001394	0.000412	**0.000337**	0.000543
NFLX	0.004265	0.003553	0.001346	**0.000725**	0.001061
AMZN	0.001396	0.001803	0.000577	**0.000299**	0.000428
BA	0.001926	0.001487	0.000580	**0.000549**	0.000594
JPM	0.000854	0.000882	0.000656	0.000420	**0.000349**
JNJ	0.004491	0.000859	0.000438	**0.000383**	0.000521

Table 4: Performance of Models on our test set for the selected stocks (MSE)

Stock	CNN MAPE	MLP MAPE	LSTM MAPE	BD LSTM MAPE	SVR MAPE
DOW	5.445118	5.327034	2.117862	**1.959733**	2.579952
AAPL	5.555139	4.298248	2.243054	**1.629289**	2.208483
WFC	4.256979	3.983184	2.264416	**1.798474**	1.954965
PFE	3.515359	3.634950	1.951703	**1.733275**	2.225309
NFLX	7.08869	7.39527	3.98807	**3.13713**	3.49169
AMZN	5.47037	6.55812	3.50263	**2.34823**	2.49752
BA	4.25537	3.79438	2.34695	**2.26781**	2.31406
JPM	3.86312	3.63829	3.02148	2.59847	**2.35191**
JNJ	7.18647	3.39613	2.30667	**2.18772**	2.67865

Table 5: Performance of Models on our test set for the selected stocks (MAPE)

network regressors vs the test set. The accuracy of this prediction is shown in Table 7. As we can see both LSTMs' superiority is less clear cut this this case compared with NLP and

Model	Run Time (seconds)
CNN	16
MLP	12
LSTM	110
SVR	161
Bi-directional LSTM	664

Table 6: Speed of model training on full dataset using GTX1660ti mobile GPU

Stock Ticker	CNN	MLP	LSTM	BD LSTM
DOW	**0.501901**	0.490494	0.496198	0.488593
AAPL	0.517110	0.507605	0.532319	**0.547529**
WFC	0.500000	**0.549430**	0.524715	0.545627
PFE	**0.522814**	0.492395	0.501901	0.513308
NFLX	0.515209	**0.547529**	0.513308	0.526616
AMZN	0.488593	**0.511407**	0.501901	0.465779
BA	0.488593	0.534221	0.528517	**0.543726**
JPM	**0.532319**	0.509506	0.511407	0.486692
JNJ	0.507605	**0.551331**	0.528517	0.513308

Table 7: Accuracy of Prediction of stock movement direction of NN techniques tested

CNN. Of the two LSTMs again Bi-directional LSTM seems to be the better performer in more cases. However its performance when looking at it in this way is casts doubts as to whether it could inform a successful trading strategy.

To prove conclusively the performance of the best model the Bi-directional LSTM in the real world is a more complex question, and starts to ask questions regarding the trading strategy that is employed and also what return can be generated in comparison with traditional trading strategies. Certainly the improvement of MSE over other techniques is encouraging but the accuracy of the prediction of direction is less so, being not signficantly better than a random selection (for any model presented here). Compared with other methods presented here the Bi-directional LSTM network does appear that it would be a more useful tool on the bases of the improved regression error. But in order to fully assess the effectiveness, more research should be done using techniques such as trading simulations similar to those presented in Gerlein et al. 2016 and also comparisons with common strategies such as buy and hold such as the comparisons presented in Żbikowski 2015 in terms of both return on investment and equity draw down.

There is also scope for further research to improve this model by using additional data to the stock's technical price and volume data alone. Since the winning Bi-directional LSTM model itself is still relatively low computational complexity, it certainly is not in the scale of Big Data where computational power becomes a limiting factor. Macroeconomic indicators for example such as interest rates and commodity prices or also Natural language processing related data such as sentiment from financial news websites could also be employed which could also augment and improve predictions.

References

Cowles 3rd, Alfred (1933). "Can stock market forecasters forecast?" In: *Econometrica: Journal of the Econometric Society*, pp. 309–324.

Di Persio, Luca and Oleksandr Honchar (2016). "Artificial neural networks architectures for stock price prediction: Comparisons and applications". In: *International journal of circuits, systems and signal processing* 10.2016, pp. 403–413.

Fama, Eugene F (1970). "Efficient capital markets: A review of theory and empirical work". In: *The journal of Finance* 25.2, pp. 383–417.

Gerlein, Eduardo A et al. (2016). "Evaluating machine learning classification for financial trading: An empirical approach". In: *Expert Systems with Applications* 54, pp. 193–207.

Henrique, Bruno Miranda, Vinicius Amorim Sobreiro, and Herbert Kimura (2018). "Stock price prediction using support vector regression on daily and up to the minute prices". In: *The Journal of finance and data science* 4.3, pp. 183–201.

Ince, Huseyin and Theodore B Trafalis (2007). "Kernel principal component analysis and support vector machines for stock price prediction". In: *Iie Transactions* 39.6, pp. 629–637.

Jia, Mingzhu et al. (2019). "Analysis and Research on Stock Price of LSTM and Bidirectional LSTM Neural Network". In: *3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)*. Atlantis Press.

Lu, Chi-Jie, Tian-Shyug Lee, and Chih-Chou Chiu (2009). "Financial time series forecasting using independent component analysis and support vector regression". In: *Decision Support Systems* 47.2, pp. 115–125. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2009.02.001. URL: http://www.sciencedirect.com/science/article/pii/S0167923609000323.

Rundo, Francesco (2019). "Deep LSTM with Reinforcement Learning Layer for Financial Trend Prediction in FX High Frequency Trading Systems". In: *Applied Sciences* 9.20, p. 4460.

Yang, Kai et al. (2019). "A blind spectrum sensing method based on deep learning". In: *Sensors* 19.10, p. 2270.

Żbikowski, Kamil (2015). "Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy". In: *Expert Systems with Applications* 42.4, pp. 1797–1805.