Predicting Stock Market Movements Using Public Sentiment Data and Sequential Deep Learning Models

Lyuang Fu, Junge Zhang, Zhewen Song,Yichao Shen
Center for Data Science, New York University
lf1664@nyu.edu,jz3502@nyu.edu,zs1356@nyu.edu,ys3197@nyu.edu

ABSTRACT

For interest of predicting stock price in the near future, our team focused on finding the most effective algorithm to get a decent result. The stock market is very dynamic due to various features. Cooperated with Accern, a financial information company that automates research and data analysis processes in the financial market, our team took important features from news, social network and the stock market, and fed them into different machine learning algorithms including Lasso Regression and Long short-term memory(LSTM) recurrent neural network. Our team also applied many different data preprocessing methods such as clipping and sliding window to improve the performance of model. For evaluation, due to the specialty of stock prediction, not only the value of stock price change but also the direction was considered in the process of analysis. Our team used three different metrics to test all models with mean absolute error (MAE), direction accuracy and Sharpe ratio. The best model is highlighted in the end with some future thoughts.

1 INTRODUCTION

It is a general interest to predict stock market movement which changes erratically every day. The financial industry desires a reliable reference to check their thoughts about the future stock price movement. Under such background, our team cooperated with Accern, a financial information company, to design a prediction model that can predict the stock movement in a daily basis. Meanwhile, we believe stock price is affected by information from media including news and social network. Accessible to time series data from news and social network (including sentiment, impact of news stories, etc.), this project aims to use news analytics to forecast the stock market. We design various models and evaluation metrics to find a model that can be put into practical use to help investors make better decisions in the stock exchange.

2 LITERATURE REVIEW

Forecasting the movements of financial products such as stock market has gained its popularity among researchers due to its potential for profits. In tradition, a lot of studies have focused on linear statistical time series models such as autoregressive integrated moving average(ARIMA). However, the noisy and non-linear dynamic natures of the system could limit the performance of such models, as stated by Pai and Lin (2005). At the same time, neural network has been drawing increasing attention and is more applied in the prediction. Quite a few studies were taken to compare and discuss the superiority of linear time series models like ARIMA and traditional neural networks on predicting financial movements. From most of the literatures our team have reviewed, the traditional neural network has a better performance than ARIMA model in terms of accuracy and mean square errors on price predictions, but the difference is not that significant. (Kohzadi et al, 1996)(Ayodele, Aderemi & Charles, 2014).

In the recent few years, long short-term memory (LSTM) based recurrent neural networks (RNN) has been regarded as one of the best models on such sequential prediction problem because of its ability to update the input through LSTM instantaneously. In other words, it remembers the previous input and can output according to the current input and the remembered values. As an example, in the study of Liu (2018), the best LSTM model reached more than 66% accuracy for all companies and 72.06% max accuracy, which outperformed all

other methods such as support vector machine (SVM) and convolutional neural networks(CNN) in both Standard Poor's 500 index and individual companies stock price. From above, we decided to use LSTM as our core model. In addition, when reviewing the study of Selvin et al. (2017), our team noticed that they had applied sliding window approach to pre-process datasets in order to predict future values on a short term basis, which inspired us on dealing with data sparsity.

3 DATA SETUP

The original dataset contains two parts. The first dataset includes price information of 19 representative stocks, such as Apple and Google, from year 2014 to 2018 with 6 numerical features and more than 23,000 rows. The price dataset is provided in daily format. The other dataset contains hourly information collected from everyday news and social network. The information includes evaluations of news sentiment and impact, and is assumed to have potential influence on the stock market, with 141 numerical features and more than 200,000 rows. For each training row, we used data within 24 hours (9am to 9am) of the current day as training features while the open to open price (9am – 9am) change (in percentage) between the next two consecutive days was the target. Our team intentionally split the train, validation and test datasets in order through the timeline to avoid the leak by predicting the past with future data. Specifically, the training data was from January 2nd, 2014 to May 3rd, 2017. The validation data was from May 4th, 2017 to April 4th, 2018. The test data was from April 5th, 2018 to September 12, 2018, The proportion between training, validation and test data was roughly 8:2:1.

3.1 Aggregated Preprocessing

Our baseline model, lasso regression, could not take hourly features and daily target at the same time, so only for the baseline benchmark, we aggregated the news data from hourly to daily format by first forward filling the missing hourly data (As an example, if data on 5am was missing, data at 4am was filled on that spot). Then considering a day from 9am to 9am next day, we aggregated the 24-hour news information data by average for this day.

3.2 Non-aggregated Preprocessing

While LSTM itself can do a "many to one" neural network, which means it can automatically process hourly features to predict the daily based target, we considered different data preprocessing variations to improve the performance of LSTM. Among different techniques, we applied window sliding to manuelly add more data into the training model, clipped data to reduce the impact of outliers and added gaussian noise to prevent the model from overfitting.

3.2.1 Sliding Window

Generally, neural network need large amount of data to train. To expand our training dataset in LSTM, sliding window technique was applied on the hourly collected features. With the width of 24 hours, we shift the window one hour earlier each time to crop previous 24 hours training data for the same target. Through the process, the sliding window gained 23 more training feature sets for every day, ending with 24 times more training data than LSTM without a sliding window. Figure 1 is an example of the sliding window our team used.

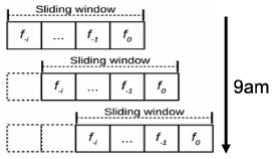

Figure 1: Sliding Window

3.2.2 Clipping

While news data do not necessarily provide related information every hour, the hourly data has high sparsity which means there are relatively extreme values that may affect the performance of LSTM. To reduce the impacts of these outliers, our team clipped top and bottom data that were out of range of 3 standard deviations.

3.2.3 Noise

To prevent the model from overfitting and let the model have the ability to learn new stuff every epoch, we added randomness by applying a Gaussian noise layer in the LSTM model. It aimed to prevent the model from overfitting because of learning the same thing every epoch.

4 METHODS

To predict the future price changing percentage of each stock, our team applied the Lasso Regression as the baseline benchmark. The idea was to get a basic sense of data with a simple and straightforward linear machine learning model. The second method our team used was long short-term memory(LSTM). "LSTM is an artificial recurrent neural network(RNN) architecture widely used in the field of deep learning." (Wikipedia, 2019). LSTM generally works well with time series data with its property processing entire sequence data. By recursively passing sequence information in the training process, LSTM can connect and integrate information through 24-hour data to better predict the stock price change. Our team expected LSTM to perform better than the Lasso baseline.

4.1 Lasso Regression

Lasso regression is a type of linear regression, as defined, performs L1 regularization, which adds a penalty that equals to the absolute value of the magnitude of coefficients (Stephanie, 2015). The final model will be sparse with few coefficients by controlling coefficients of some features to be zero if they are less informational to the model. With its linear structure and selective property on more informative features, it is the ideal as baseline model with simple training process dealing with high dimensional feature space. The use of L1 regularization also secures that the model does not memorize every data point to improve the training performance. To realize the Lasso Regression, scikit-learn package was utilized for the convenience and efficiency. In addition, different pairs of hyper-parameters such as the degree of regularization and learning rate were tested to ensure the final baseline model had the best performance.

4.2 LSTM

As defined before, long short-term memory (LSTM) is a widely used deep learning model, especially for time series data. The advantage of LSTM is that it can take previous output into consideration and establish a bridge between past data and current data recursively. In our project, all data points lied on an explicit timeline. obviously, to predict the future price change of stock, it would be more informative if we considered the entire sequential messages within 24 hours. Compared with lasso regression that had to aggregate hourly data into daily data beforehand, LSTM kept hourly data information completely and read the data points in much more depth before making final prediction decisions. The drawback is that it is more difficult to understand the process in the "black box" due to the complicated structure. However, the final performance will benefit from such structure that can process entire sequential information. Figure 2 shows a simple demonstration of LSTM.

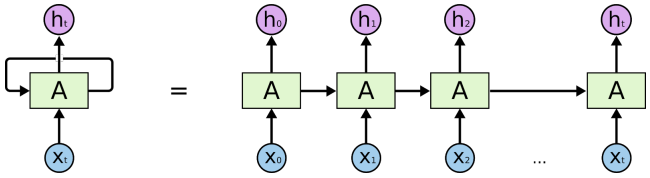

Figure 2: Structure of LSTM

5 EVALUATION

Our team considered the model more than a regression model in terms of evaluation metrics for practical business use. To predict the stock market movement, a decent model should not only be able to predict the value of movement but also the correct direction. The reason is that even though we predict a very close value for the stock price movement, different signs will lead to different choices of investment. For example, if the model predicts that stock A will experience a small drop of 0.01%, while it actually will have a small increase of 0.01% in price. As a result, the investor may sell the stock to prevent further loss as indicated by the prediction, instead of holding the stock for an actual higher price before selling. The different signs of the predicted value cause the investor to lose money even though the value of the prediction is very close to the real value. Therefore, besides the traditional regression evaluation metric, we also tried to give the model the power to optimize the direction accuracy with related evaluation metrics. In addition, from the practical usage of the model in the stock market, our team tried to give an intuitive evaluation on the model's ability in gaining benefits. We have added special financial evaluation metric that directly demonstrated the expected return of investments of models. All evaluation metrics were considered together to identify the model with the best performance both theoretically and practically.

5.1 Training Evaluation

For training evaluation, MSE (mean squared error), and LSTM piecewise MSE were used as metrics to train our models. In particular, MSE was the basic metric used for the Lasso benchmark model and piecewise MSE was a special and customized metric to improve our LSTM model's performance.

5.1.1 Lasso MSE

Our baseline model was Lasso regression. MSE was a reasonable metric to train on. MSE is the average of the squared error that is used as the loss function for least squares regression:

$$MSE = \sqrt{\Sigma_{i=1}^{n} \frac{1}{n}(w^T x_i - y_i)^2}$$

It is the sum of the square of the difference between the predicted and actual target variables over all the data points , divided by the number of data points(Nick, Manpreet Rajdeep 2019). From definition, lower MSE means the overall predictions are closer to the target and therefore a better Lasso regression model.

5.1.2 LSTM piecewise MSE

In order to get a better performed model that both can do well in value prediction and direction accuracy, we customized a special piecewise loss function named LSTM piecewise MSE. For this piecewise loss function, the training loss for LSTM was MSE when the prediction and the target had the same sign while it was four times MSE for different signs. The piecewise MSE aimed to penalize more when the prediction was completely wrong in stock price increasing or decreasing (e.g. make the negative prediction for the actually increasing stock price). Our team expected this LSTM piecewise MSE metric to optimize our model in two targeted directions: the accuracy of both value and direction of our prediction. Figure 3 to Figure 6 show the LSTM validation piecewise MSE versus epoch based on different non-aggregated preprocessing.

Figure 3 shows the basic LSTM validation piecewise MSE loss versus epoch.
Figure 4 shows the LSTM with sliding window validation piecewise MSE loss versus epoch.
Figure 5 shows the LSTM validation with clipping and sliding window piecewise MSE loss versus epoch.
Figure 6 shows the LSTM with sliding window and noise validation piecewise MSE loss versus epoch.

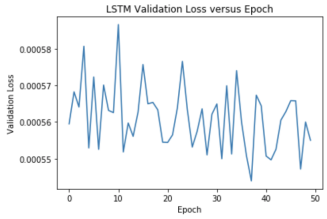

Figure 3: LSTM Result

Figure 4: LSTM with Slding Window Result

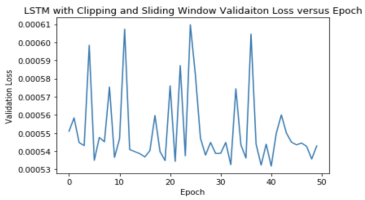

Figure 5: LSTM with Clipping and Sliding Window Result

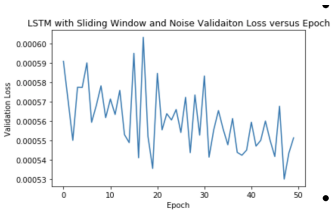

Figure 6: LSTM with Sliding Window and Noise Result

5.2 Test Evaluation

For test evaluation, as discussed above, our team paid attention to the accuracy of both value and direction. Basically, mean absolute error (MAE), direction accuracy and Sharpe ratio were used to represent the overall quality of our models.

5.2.1 MAE

Mean absolute error is one of the common metrics used to show the errors between the truth and prediction in regression model. The reason to use MAE instead of other lost function is that MAE can directly reflect the difference between true value and prediction value. The error is clearer to compare in the case that our price change prediction and target were small numbers in percentage. MAE was the first and basic metric our team used to test the performance of all models. While MAE showed the model performance in terms of value prediction accuracy, it was not enough in terms of direction accuracy and practical financial application in our scenario.

5.2.2 Direction accuracy

Direction accuracy was the metric used to test the sign prediction accuracy of predicted stock price change. For a given list of prediction value of stock price movement, direction accuracy computed the percentage of correct-sign prediction among all predictions compared with ground truth. The evaluation metric cared about the ability whether the model could accurately forecast the increasing or decreasing trend of the stock correctly.

5.2.3 Sharpe ratio

"The Sharpe ratio was developed by Nobel laureate William F. Sharpe and is used to help investors understand the return of an investment compared to its risk." (Nick Pentreath, Manpreet Singh Ghotra and Rajdeep-Dua, 2019). In our case, this ratio is useful to represent the value of all models we trained in the real investment scenario by intuitively showing the expected returns with these prediction models. The formula was as follows:

$$\text{Return} = \frac{\sum prediction * target}{\sum prediction}$$

$$\text{Sharpe} = \frac{mean\,of\,return}{std\,of\,return}\sqrt{n}$$

Basically for return, we calculated it for all stocks in one day. It could be understood as the normalized return across all stocks in one day. Then, we used the average returns within the testing time frame and corresponding standard deviation to get the value of Sharpe ratio. Finally we multiplies the square root of stock market opening days(n, about 250) in a year to represent the annualized Sharpe ratio.

The Sharpe ratio has its practical meaning in the field of investment. Investors care about how the combination of stocks perform during a specific time range, which is measured by mean of return. If the mean is larger, it means the combination performs well. However, investors also want to spread the risk in the combination instead of just concentrate all risks in several stocks. This part is measured by the standard deviation of returns. If standard deviation is larger, it means the combination does not spread the risk ideally. Overall, after considering the above two factors, we can get the final Sharpe ratio. When the Sharpe ratio is higher, it means that the model performs better in the real investment scenario. Generally, a Sharpe ratio above 2 is considered as good. Our team applied Sharpe ratio as a proof of the model's ability in gaining actual profit through stock exchange.

6 RESULTS

For the final results, all information was summarized in Figure 7.

	MAE	Direction Accuracy	Sharpe ratio
LSTM	0.0094206	0.5309211	0.9847754
LSTM Clipping + Window	**0.0093911**	0.5654605	1.8502489
LSTM Noise + Window	0.0094107	0.5093421	0.1014509
LSTM Window	0.0093949	0.5404605	**3.0202048**
Lasso	0.0096155	**0.5716247**	2.3129036

Figure 7: Result

6.1 Lasso Regression

Lasso regression is designed as the benchmark for all models but the result is a bit surprising. It had the largest direction accuracy and second best Sharpe ratio. However, when our team checked the prediction of Lasso regression on the test set, it was found that Lasso benchmark could only predict same positive values for every day. It turned out Lasso benchmark was not bad because the data were extracted from a bull-market. Holding stocks every day still gave good results but this strategy would not generalize very well in the future stock market. Our team plotted a graph to check the cumulative performance of investment return for the baseline model in Figure 8. After about 5 months'virtual investment, the final return was about 1.13.

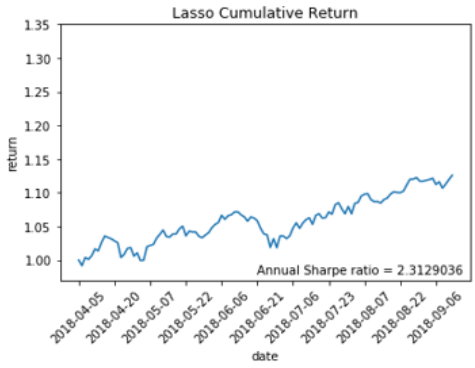

Figure 8: Lasso Cumulative Return

6.2 LSTM

For the LSTM, our team trained four different models. The basic LSTM without using any special data preprocessing methods ranked last in most metrics except the Sharpe ratio. LSTM model with clipping and sliding window strategy acted stably among all LSTM models with the best MAE and direction accuracy but Sharpe ratio below 2. LSTM model with window and noise strategy did not performs well, especially in Sharpe ratio with just 0.1. LSTM model with only sliding window strategy was the model with the highest Sharpe ratio and relatively low MAE, which was considered as the most desirable model in our scenario. Our team also plotted the graph for the best LSTM model to check the cumulative performance of investment return in Figure 9. After about 5 months'virtual investment, the final return was about 1.35, much higher than the baseline Lasso model.

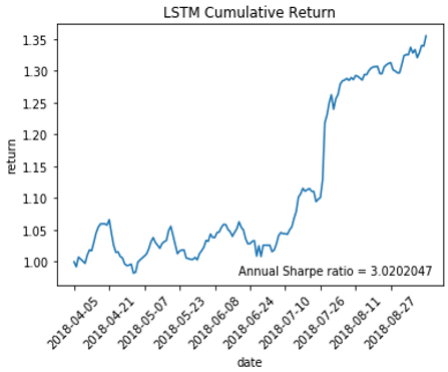

Figure 9: LSTM Cumulative Return

7 LIMIT AND FUTURE WORK

7.1 Volume of data

In order to obtain a better performance, we need to feed LSTM model with more data. For now, we only covered 19 stocks from 2014 to 2018 . Both of the time range and number of stocks could be expanded to get more data to train our model for more information put in the model.

7.2 Skewness, Models and Metrics

For now, our data had some skewness, which was a concern for our robustness of our model. Methods such as log transformation could be experimented in the future to avoid potential problem. In addition, one concern is how to increase the direction accuracy of the model. Other models or a new training metric may be designed and experimented to better optimize the direction accuracy and predict stocks in the context of inherent time series data. In real stock market, price fluctuation is much more complicated and unpredictable. More sophisticated framework could be tried to predict real market better.

8 CONCLUSION

Our team applied a Lasso benchmark and LSTM to predict the stock price change with time series data from news, social network and stock market. Meanwhile, we have tested our models with MAE, direction accuracy and Sharpe ratio to ensure the model performance both theoretically and practically. From our results, Lasso hit a good Sharpe ratio. However, Lasso benchmark could only predict same positive values for every day. It worked out coincidentally in this time frame but that strategy would not generalize very well. The reason why it predicted same positive value was the time frame of the data was in a bull-market. Thus, if you just held the stocks all the time, it could still give you easy and decent returns. However, this would not always hold true. The future market could be more fluctuating and the holding strategy would not work any more.
For our LSTM model, currently with limited amount of data, it showed its potential with reduced MAE and good Sharpe ratio compared to the baseline. However, the direction accuracy needed to be further optimized,

which was more important in some degrees because the prediction direction could directly determined profit gain or loss.

9 ACKNOWLEDGEMENT

Our team sincerely thanks Accern and our mentor from Accern data science team, Dr. Josua Krause for guidance, supports, time, and resources during the project.

10 REFERENCE

P.F. Pai, C.S. Lin 2005, 'A hybrid ARIMA and support vector machines model in stock price forecasting', *Omega*, vol. 3, no. 6, pp. 497-505.

Adebiyi, Ayodele Adewumi, Aderemi Ayo, Charles 2014, 'Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction', *Journal of Applied Mathematics*, vol. 10.1155/2014/614342, pp.1-7.

Nowrouz Kohzadi, Milton S Boyd, Bahman Kermanshahi, and Iebeling Kaastra 1996, 'A comparison of artificial neural network and time series models for forecasting commodity price', *Neurocomputing*, vol.10, no.2, pp.169–181.

Liu, Huicheng 2018 'Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network.' *ArXiv:1811.06173 [Cs, q-Fin]*, *arXiv.org*, http://arxiv.org/abs/1811.06173.

Selvin, Sreelekshmy R, Vinayakumar Gopalakrishnan, E. A Menon, Vijay Kp, Soman 2017, *Stock price prediction using LSTM, RNN and CNN-sliding window model*, 1643-1647. 10.1109/ICACCI.2017.8126078.

Marshall Hargrave 2019, *Sharpe Ratio*, Investopedia, viewed 8 December 2019, <https://www.investopedia.com/s/sharperatio.asp>/lasso-regression/>.

Nick, P, Manpreet, G Rajdeep, D 2019, *Mean Squared Error and Root Mean Squared Error*, viewed 10 Dec 2019, <https://www.oreilly.com/library/view/machine-learning-with/9781785889936/669125cc-ce5c-4507-a28e-065ebfda8f86.xhtml>

Stephanie 2015, *Lasso Regression: Simple Definition*, Statistics How To, Cancer Council, viewed 9 December 2019, <https://www.statisticshowto.datasciencecentral.com