PREDICTING S&P500 PRICE MOVEMENTS USING NEURAL NETWORKS

COMP 4107

PROFESSOR ALAN TSANG

GROUP 11

Ethan Prentice
101070194
Carleton University
Ottawa, ON
ethanprentice@cmail.carleton.ca

James Va-Chao
101084995
Carleton University
Ottawa, ON
jamesvachao@cmail.carleton.ca

Ghiath Majjanne
101090915
Carleton University
Ottawa, ON
ghiath.majjanne@carleton.ca

April 25, 2021

Contents

1 Introduction 3

2 Related Works 3

3 Data 4

4 Methodology 5

5 Results 6

6 Discussion and Conclusion 8

7 Figures 10

Bibliography 17

1 Introduction

This paper will cover the following topic. Firstly, the usage of neural networks to predict a stock market index (S&P500). Secondly, the effect of the recency of time series data for predictions. Lastly, we will explore different ways to manipulate the input data to predict market indexes.

This project pulls from real life stock data to see if neural nets can identify any underlying trends and how accurately time-series models can be used in real life problems.

The problem domain will be the various factors that contribute to stock index price movement and the analysis of various technical indicators and techniques to achieve an accurate future prediction of that index's price and/or direction.

2 Related Works

We researched S&P500 prediction techniques during the proposal stage of the project to choose the datasets that we would use. Some of these consisted of just predicting the direction of the index, while others focused on predicting the actual stock price on any given day.

The first related-work proposed by Jinho Lee and Jaewoo Kang [LK], uses historical individual stock data from the S&P500 to predict the future price of the index using MLP and CNN. This method was able to focus on more recent data as its data input shape was daily indicators multiplied by the amount of companies in the index. This method allows the use of multiple features (closing price, volume) and multiple companies to predict the final S&P closing price and as a result allowed the usage of a weighted sum. This general technique of using individual companies to predict and index will be explored in this report.

The second related-work uses convolutional and recurrent layers in the same network to remember short-term trends in the datasets. This helps the network learn both long-term trends and short-term trends and can hopefully identify if the price movement in a short time period will have a high impact on price. This related model's architecture is shown in Figure 7.1

3 Data

Reference to datasets

Dataset & Where we are acquiring it from	
S&P 500 index with daily basic info	Kaggle
S&P 500 index with more advanced monthly stats	Datahub
The individual S&P500 companies daily basic info	Kaggle

Dataset Contents

The first dataset contains daily information about the S&P500 index from 1950 to 2018 containing:

- Price
- Highs
- Open price
- Volume
- Lows
- Close price

The second dataset contains monthly information about the S&P index from 1870 to 2018 containing:

- Price
- Consumer Price Index
- Real dividend
- Dividend
- Long interest rate
- Real earnings
- Earnings
- Real price
- P/E10 ratio

The third dataset contains daily information about the companies included in the S&P500 index from 2013-2018:

- Date
- Close
- Lowest price of day
- Open
- Volume

Dataset Reasoning

We chose our data sets to explore our topics of interest related to index predictions. Firstly, the major difference between dataset 1 and dataset 2 was the frequency of the data. Dataset 1 contains daily info while dataset 2 contains monthly info. We theorized that using daily data would be better as it is more recent and to get a decent amount of data points we would not have to use data that could be old and irrelevant which would have to be done for monthly data.

Secondly, the major difference between the first and the third data set was the inclusion of individual company data in the third data set while the first data set only had the S&P500 index data. We theorized that by creating individual predictions of all the individual companies underlying the index, it would be possible to combine these predictions to better predict the price index. Our rational was that we had the indicators for price and volume for these individual companies and with that information, our neural net would isolate key companies that would have a greater effect on predicting the index.

4 Methodology

The first dataset contains less columns but has a higher granularity compared to the second dataset, with daily values rather than monthly. We experimented with a simple LSTM model and hybrid convolutional and recurrent models based off of those proposed by Yaping Hao and Qiang Gao [HG20]. These model architectures can be seen here: Figure 7.2.

The non-sequential hybrid model worked the best out of these approaches due to it's ability to incorporate LSTMs on different "windows" that are inputted into it by the convolutional network. This allows us to "remember" what trends looked like in small pieces, so we can extract the trends of the important features using the convolutional layers, allowing us to remember these trends in the LSTM layers. Concatenating these LSTM layers should give us a much better result than the sequential hybrid model.

The second dataset has more columns but less data points as it only has 1764 data points. Because of the lack of data, we have decided to only try to predict the price direction as predicting the actual price requires more data. For the model we tried a sequential LSTM model with dropouts layers Figure 7.4. We also tried a simple CNN model with 2 CNN layers and one normal Dense layer Figure 7.5 to compare the the results but we saw no difference between them. We then tried to balance the classes to prevent the model from only guessing one direction all the time but it didn't really change the results.

For the third dataset, the paper proposed by Jinho Lee and Jaewoo Kang[LK] used a CNN and MLP model that used individual stock indicators to predict the index price which was then compared to a baseline CNN and MLP model that just used the index. For our paper, in order to illustrate the core concept of using individual stock predictions to create an index prediction, we decided to keep it simple with a simple RNN model with a single LSTM and dropout layer Figure 7.6. Various learning rates, dropout percentage, and LSTM hidden layers were modified to produce our results. Performance wise, LSTM hidden layers above 256 would take too long to train and would result in similar performance. We opted to go with 256 LSTM hidden layers with a 0.2 dropout and a learning rate of 0.1.

5 Results

For the first dataset, we used two different approaches for the model. As mentioned in the Methodology section, these were a simple LSTM model with dropout, and hybrid variations proposed by Hao and Gao. These experiments were run to predict price, as well as direction.

For the first approach on this dataset, when using a simple LSTM model, the validation accuracy is very unstable and depending on the run, ranges from just 2% direction accuracy to 48%. It's validation loss is also near-flat as seen in the ipnyb.

The hybrid models performed much better than the simple LSTM. As expected, the non-sequential hybrid model performed much better than the sequential hybrid model. This is due to how it incorporates LSTMs on multiple window-sizes that are fed into them from the convolutional and max pooling layers.

For the models used on the first dataset, these were the results:

Data	Epoch Convergence	MSE (Test)	Direction Accuracy (Test)
Simple LSTM Model	60	0.2409	2.33% - 48.67%
Sequential Hybrid Model	35	0.2370	42.67%
Non-Sequential Hybrid Model	37	0.0307	67.11%

For the second dataset, both the LSTM and the CNN models converged at around 0.63 validation accuracy while the loss kept decreasing steadily during all epochs. It would probably be worth it to look for a different dataset with the same columns but with daily data instead of monthly which we believe would increase the accuracy of our model significantly.

For the third dataset, our model converged at the following points:

Data	Epoch Convergence	Mean_squared_error
Individual S&P500 companies (window_size 3)	10	0.0300
Individual S&P500 companies (window_size 10)	8	0.0289
Individual S&P500 companies (window_size 30)	5	0.0281

Since we produced a price prediction based on averaging out the individual companies' prices, we can also graph and visualize the info in Figure 7.7.

Based on our results, we can see the the model views the S&P as undervalued from days 0 - 500 (Our data's earliest date is 2013-02-08) and believes its overvalued from days 900 - 1200 Figure 7.7.

Based on this prediction data we can simulate a person buying when they believe the stock is undervalued (prediction > s&p price) and selling when its overvalued (s&p price > prediction). We can then compare this to what would happen if you had just bought and held the entire duration of 2013-2018. The results can be seen at Figure 7.8.

Our results show that buying and holding was the weaker strategy and would've netted a 75% return on investment while buying according to our algorithm would've resulted in 26 trades and an overall return of 86%.

These findings were similar to the results found by Jinho Lee and Jaewoo Kang[LK] who achieved a 5%-16% better return using an MLP and CNN model.

6 Discussion and Conclusion

Overall, we expected to see some significant differences in the different datasets. We expected to see higher granularity datasets, such as the one that uses all companies in the index perform better, however there were still issues/improvements/biases that could be address.

To summarize, the first dataset performed well, specifically when following Hao and Gao's techniques that were noted in their paper. This dataset was expected to do well due to daily granularity and abundance of data to work with. While the two sequential models tend to overfit or veer off, the non-sequential hybrid network always turned a profit. For more info on the expected profit from these models, refer to the ipynb associated with the first dataset.

We expected the second dataset to not perform well because of the lack of data but surprisingly it performed well considering the size of it. However, the high accuracy pushes me to think that the models are overfitting the data even though we introduced dropout layers to combat this and tried to balance the classes to prevent the model from predicting that the price is going to go up all the time due to the natural of the index to just always increase in value.

The third dataset performed well but we would attribute this to the data used to train the model. The data had a bullish upward trend and the model probably picked up on this. When simulating the buying/selling process, the algorithm choose to sell when momentum was extremely bearish before major dips but always bought at the bottom of these dips as it constantly expected a rebound. This buying and selling compounded to produce better returns then just selling. The buy and selling

process can be visualized in Figure 7.9. However, it should be noted in Figure 7.8 that 26 trades were performed which can quickly eat into profits via trading fees/price slippage.

Furthermore, while the algorithm was bullish, it failed to be as bullish near the end of the dataset which resulted in a lot of lost profit. Had the data continued for 100 to 200 more days, the buy and hold strategy would've out performed the buy/sell algorithm.

Additional shortcomings which would've resulted in biases were that the data used to train the model did not include companies that joined/left the S&P500 during this 5 year data period. This would've created a model that was optimized for slow, steadily performing companies that could stay in the S&P500 and would fail to pickup the effects of fast growth companies or companies experience decline. This is further shown by the fact the our prediction seemed to model a consistent upwards trend that ignores short term price spikes/dips.

Additional research which we would've liked to have explored was the effect on days in terms of determining stock prices. For instance, had we used closing and opening prices, there would potentially be better days to buy/sell on. It would also be interesting to work with data that was more varied such as data during a financial crash. Furthermore, it would be interesting to test out more advanced over-fitting techniques and to work with more complex RNN structures/layers which we were exposed to in research papers but lacked the expertise and time necessary to fully understand and implement them.

7 Figures

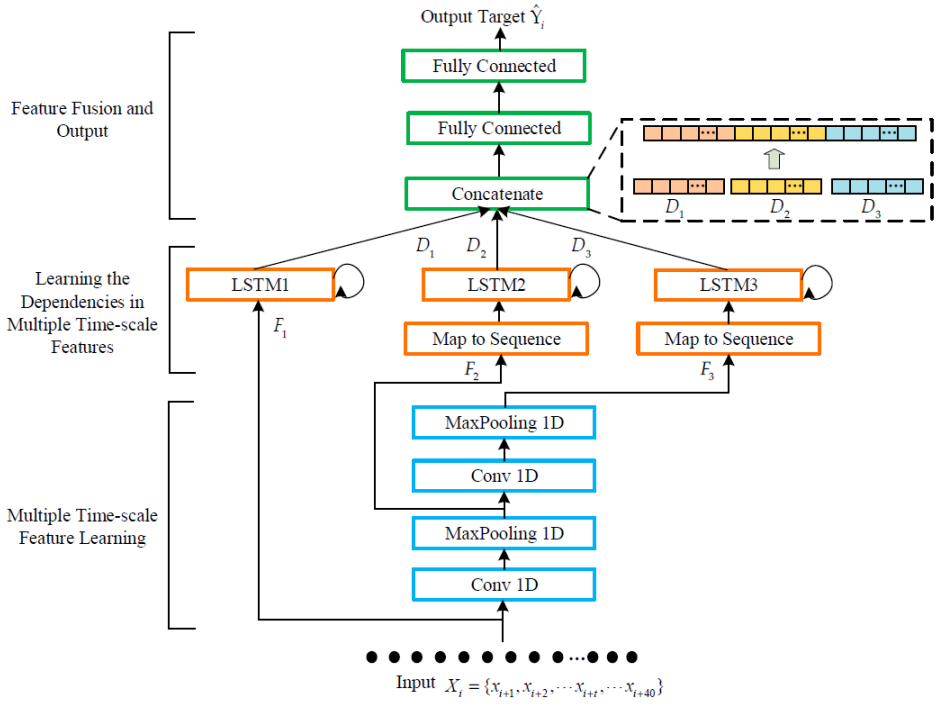

Figure 7.1: Hybrid CNN and RNN Model by Yaping Hao and Qiang Gao

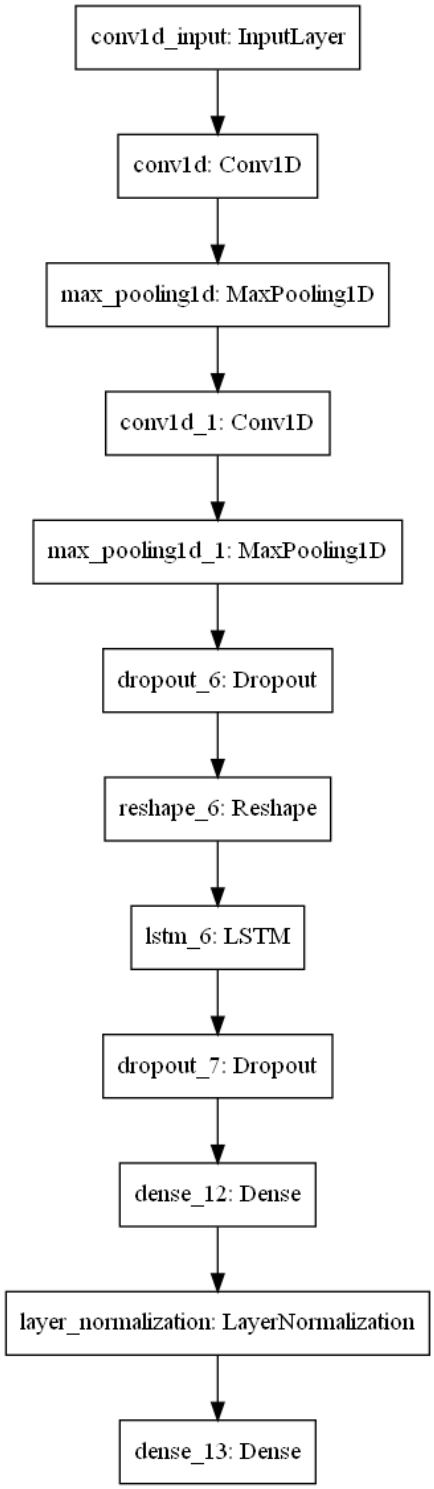

Figure 7.2: Sequential Hybrid CNN and RNN Model

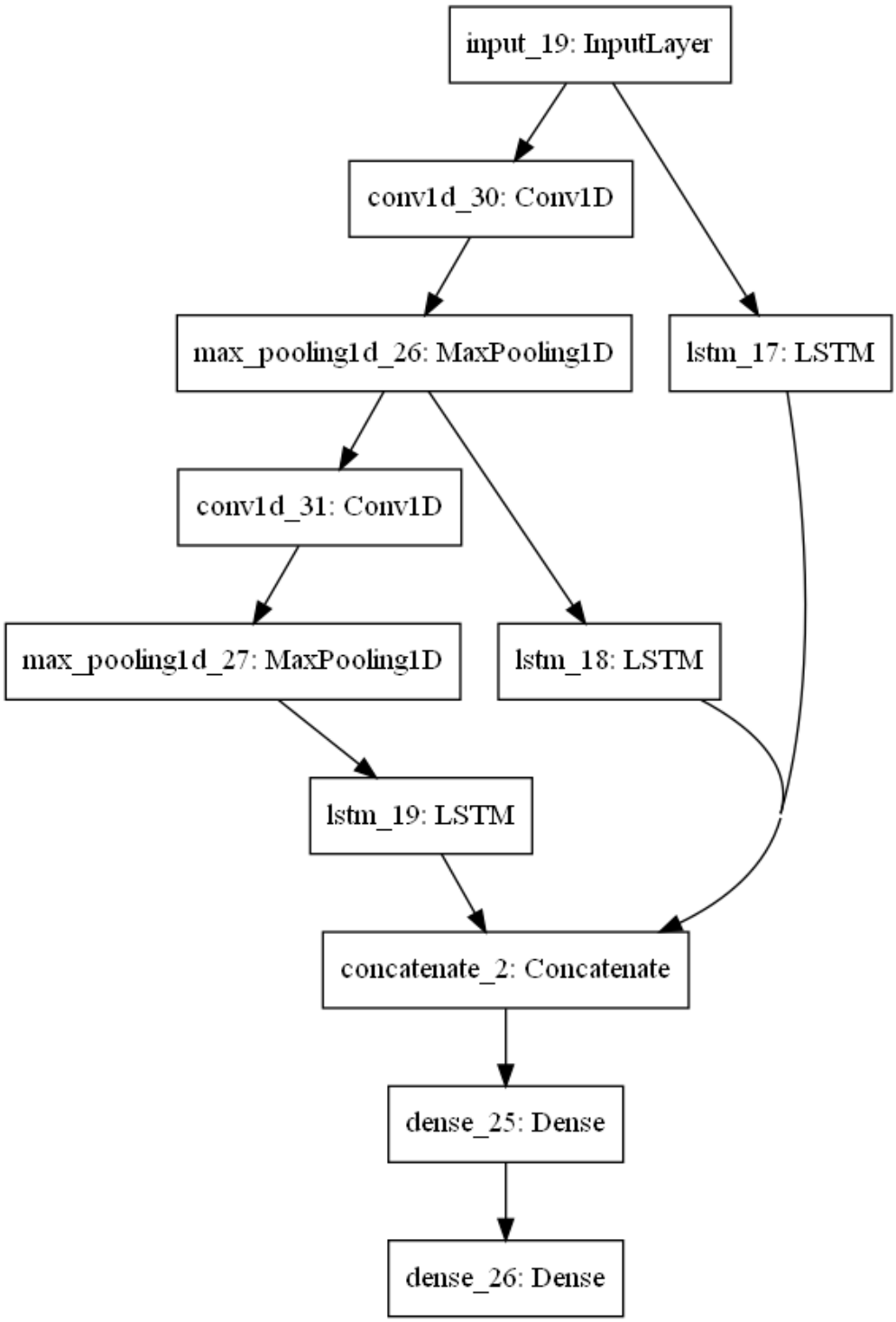

Figure 7.3: Non-Sequential Hybrid CNN and RNN Model (not including dropout or batch norm layers)

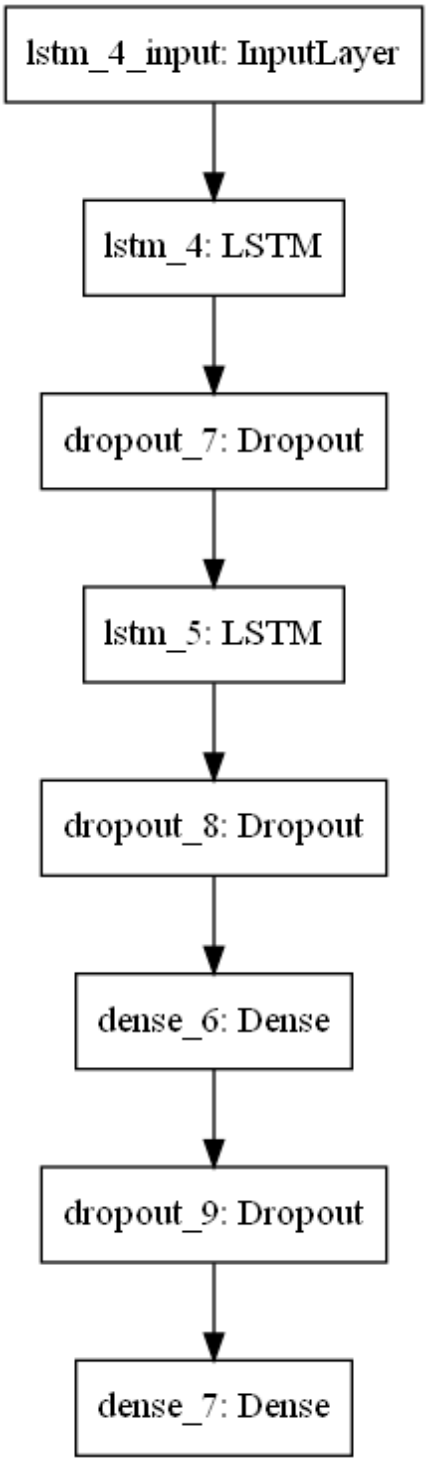

Figure 7.4: Sequential LSTM Model with Dropouts

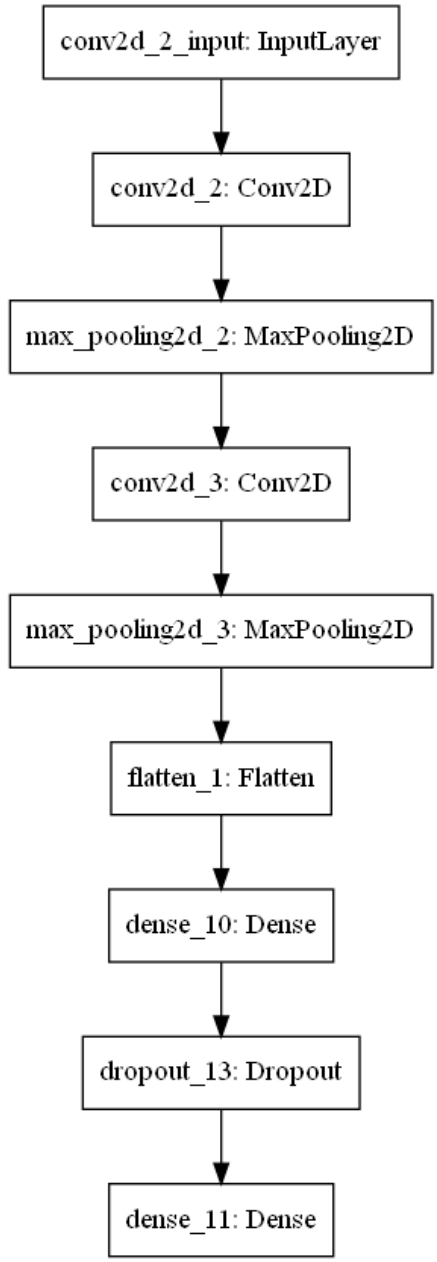

Figure 7.5: Simple CNN Model

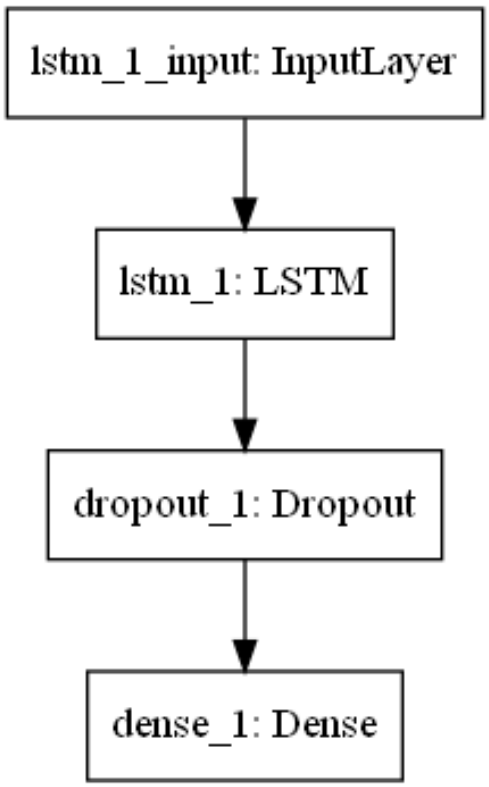

Figure 7.6: Sequential LSTM Model for Individual Companies to Index Prediction (Dataset 3)

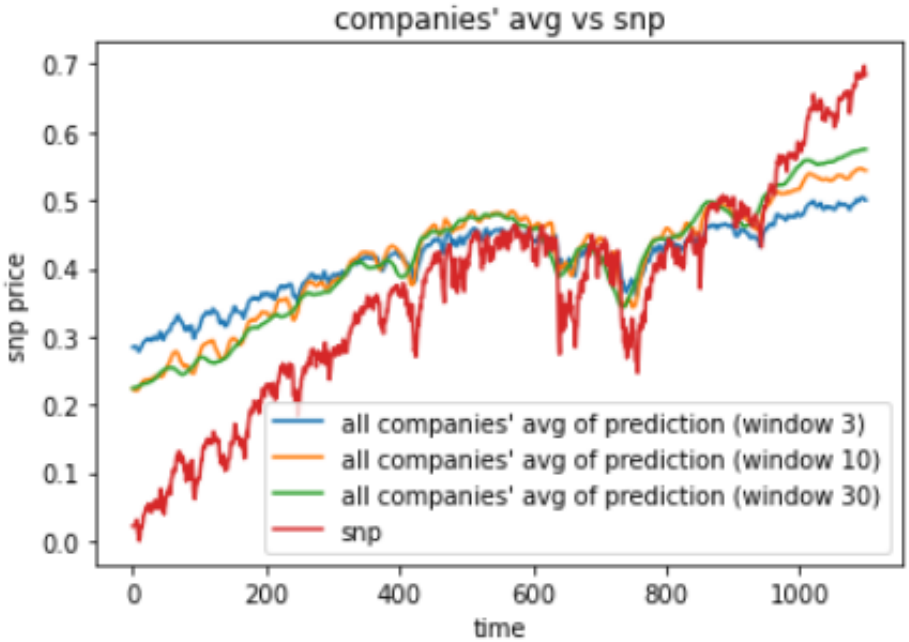

Figure 7.7: Individual companies' avg price vs S&P500 price

```
Starting SNP price 1517.93
paper_hands starting value 1517.93
diamond_hands starting value 1517.93
Final SNP Price 2664.11

Overall positions paper_hands (buy based on algo) =============
paper_hands starting value: 1517.93
paper_hands net value: 2826.45
paper_hands trades: 26
paper_hands shorts: 0
paper_hands return: 186.2%

Overall positions diamond_hands (buy and hold) =============
diamond_hands starting value: 1517.93
diamond_hands net value: 2664.11
diamond_hands trades: 2
diamond_hands shorts: 0
diamond_hands return: 175.51%
```

Figure 7.8: Simulating buying/selling using the prediction (paper_hands) vs buying and holding (diamond_hands)

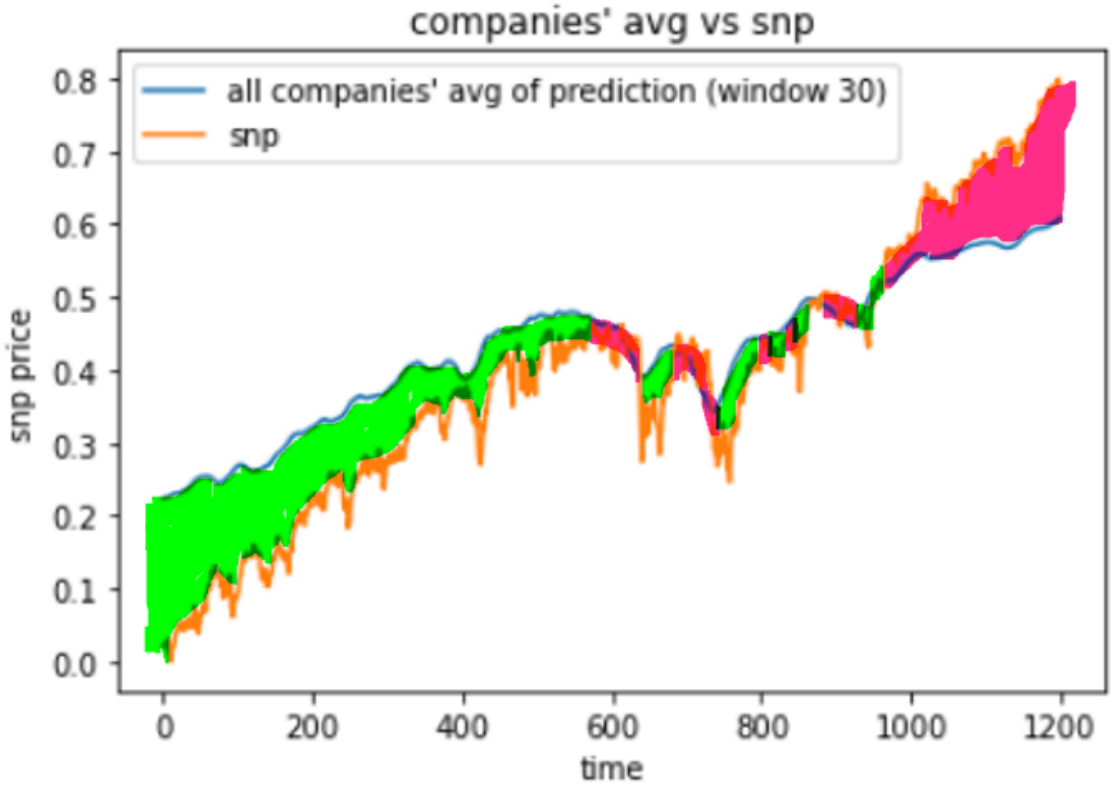

Figure 7.9: Simulating buying vs selling using graph visualization where green is buying/hold and red is selling

Bibliography

[HG20] Yaping Hao and Qiang Gao. “Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Time Scale Feature Learning”. In: *School of Electronic and Information Engineering, Beihang University, Beijing 100191, China;* (2020). Accessed: April 10th, 2021.

[LK] Jinho Lee and Jaewoo Kang. *Effectively training neural networks for stock index prediction: Predicting the SP 500 index without using its index data*. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0230635. Accessed: April 25th, 2021.