LSTM & CNN Neural Network to Stock Prediction

Wenyu Huang, Xinyue Chang

Biostatistics Division

Dalla Lana School of public Health, University of Toronto

August 3rd, 2020

Github Repo Link: https://github.com/callmeawen/CNN_LSTM_CHL7001

Contents

Abstract

The financial market is tremendously impacting our daily lives in many perspectives. People invest in exchange-traded funds against the inflation rates. Netflix produced TV series to reveal Wall Street's life. Investors, research, traders and people from a large variety of domains talk about Financial News because financial stability is highly related to our modern society. Time series forecasting is one of the most challenging missions by deep learning. In this research, our goal is to create a profit-maximizing trading strategy. In fact, the trading system is considerably affected by a reliable prediction with high qualities. Therefore, we make a comparison between long short term memory and convolution neural network techniques on the targets of stock price of two corporations: The Procter & Gamble Company and Bank of America. Final results show that CNN before LSTM did successfully tackle random noise problems and uncertain information in time series but a single CNN model expresses the best performance. A code of conduct is added to the GitHub through the link provided on the first page.

Keywords: Machine learning, deep learning, neural network, Long Short Term memory (LSTM), Convolution Neural Network (CNN), Yahoo! Finance, The Procter & Gamble Company (PG), Bank of America (BAC).

1. Introduction

Stock is an essential component in the economy. A good prediction of future market movements plays a vital role in economic growth. Some traditional concepts for making predictions are logistic regression, random forests and so on. They always fail to generate precise results on stocks since they simply assume a linear process. Instead, modern deep learning techniques have been approved to have a higher predictive accuracy due to its property in overcoming nonlinear, discontinuous and chaotic characters and capturing deep features (Jialin Liu, 2019). In this project, we aim to compare the prediction of future prices of The Procter & Gamble Company (PG) and Bank of America (BAC). The process is done by grid search on different parameters for three models: 1) LSTM 2) CNN 3) combined CNN+LSTM using a diverse set of variables. By understanding the behaviors of the stock, investors may improve their investment decisions by determining an appropriate timing and volumes for buying or selling a stock. Generally, "buy low, sell high" is the oldest and the most famous criterion in the market. However, the real world is more complicated. In order to mimic transactions as close as the reality, four trading strategies are developed. In other words, the ultimate objective of this research is to maximize overall returns based on the prediction module and the trading module.

2. Material and Method

2.1. Dataset

Our raw data is the historical stock values of PG and BAC downloaded from a python package Yahoo! Finance. For each particular stock, it includes a daily open price, the daily highest price, the daily lowest price, a close price, an adjusted close price and the volume from 01/01/2017 to 01/01/2019. The closing price is demonstrated in Figure 1 and Figure 2. Moreover, several other market attributes such as technical indicators are studied and visualized in Figure 3 and Figure 4. They are able to offer various characteristics of the security from which to analyze the price movements. For example, the 5 days and 100 days moving averages are applied to smooth temporary and random price fluctuations over time. A buy signal happens when the short-duration MA crosses above the long-duration MA. In professional terms, this is called a "golden cross". On the contrary, the trend of price drops and generates a sell signal when two lines cross the other way. This is known as a "dead cross" (MITCHELL, 2020). Also, by taking into account the direction and extent of a movement and trading volume, a force index that assesses the power of a price change is displayed in Figure 1. To be specific, a decreasing FI below zero forecasts a decreasing price while an increasing FI above zero forecasts an increasing price (PALMER, 2019). Some other technical indicators tracked are Average True Range, Bollinger Bands, Rate of Change, Williams percentage Range and Moving Average Convergence Divergence. The third dataset is the Standard & Poor's 500 Index in Figure 3. The reason we consider this feature is because it consists of 500 corporations with a large market capitalization from different industries. S&P 500 is widely used by investors as a benchmark to represent overall economy and general market conditions. All the datasets are aggregated in an amount of 14 features, normalized by Min-Max scaling (Eq. (1)), and divided into a training set of 400 days, a validation set of 50 days and a test set of 50 days. Lastly, input features are passed into three models. Several parameters assumed in training are a batch size of 10, an epoch of 300, a learning rate of 0.0001 and time steps of 4 days, 14 days and 24 days.

$$x_{sc} = \frac{x - x_{min}}{x_{max} - x_{min}} - Eq.(1)$$

2.2. Long Short Term Memory: LSTM

The framework is firstly analyzed by long short term memory, which is similar to Recurrent Neural Network (RNN). By introducing the concept of memory, LSTM has the potential to deal with a vanishing gradient problem in time series (SRIVASTAVA, 2017) and transfers previous information to the present. The diagram in Figure 6 illustrates the architecture of LSTM. As data flows through different gates (the input gate z^i, the forget gate z^f and the output gate z^o in memory blocks, it is read, forgotten and stored. Then, the cell state and the hidden state are updated and transferred to the next cell. For instance, previous cell state c^{t-1} is used to store the information kept from the last step: an increasing trend of the stock price in the past. Previous hidden state h^{t-1} utilizes a feedback loop to receive outputs of some layers from last cells and feed them back into the inputs (Guanting Chen, 2017): the closing price of the stock yesterday. Next, they are combined with the current input state at x^t, which can fresh information: an unexpected major personnel change or a new policy today. By considering long term dependencies in stocks, LSTM shows a successful computing power in this experiment.

2.3. Convolution Neural Network: CNN

CNN or Convolutional Neural Network is also a big innovation in machine learning, with most applications in image recognition, image classification and natural language processing. However, with recent breakthroughs in data science, some studies show a better performance of convolutional neural networks in stock prices modeling compared with RNN. The advantage is especially reflected in "automatically and adaptively learning in spatial hierarchies of features through a backpropagation algorithm" (Rikiya Yamashita, 2018). Specifically, if features are not informative enough, they may hinder the extraction. A total number of 6 layers in CNN are constructed:

1. Input layer
2. Conv1d — The convolution operation is conducted to measure how much changes are caused by applying a weighted filter to an input (Ehsan Hoseinzade, 2019). All of the weights are summed to one.
3. Maxpooling1d with a pool size of 2— The output is a 1-dimensional tensor of size (input channels). This step has the ability to reduce the size of inputs. As a result, it will "decrease the computational cost of the learning process" (Ehsan Hoseinzade, 2019).
4. Conv1d — The second convolutional layer.
5. Flatten — Flatten function has the ability to combine the pooled features to a single column.
6. Dense — Because of its computational simplicity (Xavier Glorot, 2011), the default Rectified Linear Unit (ReLU) in Eq. (2) as a nonlinear activation function is considered to pass the output to the next layer.

Overfitting is likely to occur if a trained model fits too much to the training set. It will cause the reduction of generalization of future unseen information (Ehsan Hoseinzade, 2019). After taking transformations through each layer, the number of parameters that must be learned are decreased. Therefore, we can reduce the risk of overfitting and improve forecasting performance.

$$f(x) = \max(0, x) - Eq. (2)$$

2.4. CNN+LSTM

Traditional methods such as weighted moving average are largely introduced to smooth and de-noise datasets. The one-dimensional convolution is defined in Eq. (3), where f is the input vector with length n and g is the kernel with length m. According to the formula, the convolution operation can be viewed as a smoothing operator if the parameters are all positive, and hence, we propose to employ a CNN as a deeper input gate before LSTM to learn smoothing parameters from the inputs. Note that previous work already shows that by reducing the dimension, an "important" input gate before the LSTM will benefit the modeling of temporal structures of LSTM (Graves, Mohamed, & Hinton, 2013).

$$(f \times g)(i) = \sum_{j=1}^{m} g(j) \cdot f\left(i - j + \frac{m}{2}\right) - Eq.(3)$$

2.5. Trading Strategy

Without loss of generality, we assume that a better predicting performance generates a higher rate of returns. As a result, the second phase is to compare the total profits by designed models and parameters. Four trading algorithms are developed with an initial capital of $1000:

1. All in/all out:
   - If the prediction increases, invest all money.
   - If the prediction decreases, sell all stocks.
2. Buy or sell by a weight = rate of change of price in Eq. (4):
   - If the prediction increases, invest a proportion (weights) of money.
   - If the prediction decreases, sell a proportion (weights) of stocks.
3. Daily investment with one share:
   - If $\hat{y}_{t+1} > \hat{y}_t$, buy one share.
   - If $\hat{y}_{t+1} < \hat{y}_t$, sell one share.
4. Buy and hold:
   - Invest all on the first day of the test period.
   - Sell all in the end.

$$rate\ of\ change = \frac{\hat{y}_{t+1} - \hat{y}_t}{\hat{y}_t} - Eq.(4)$$

3. Results

3.1 Test data against Predicted values in PG and BAC

As shown in Figure 7 – 12, We tuned the time steps of 4, 14 and 24 days to see if the memory time would affect the predictions on three methods. The black line indicates our test data, and the predictive curves for each market index are represented by solid lines in different colors. According to these plots, we observe that the curve of the CNN model is much closer to our actual values than that of the other two approaches. For BAC, the curve of LSTM occasionally derives far from the test values during the test period, while CNN+LSTM performs the worst in most cases. As for PG, the observations are opposite. In terms of price trend prediction, the performance of all three models is generally good. On the other hand, there is no obvious difference in outputs when adjusting memory times.

In conclusion, unlike traditional parametric models that assume a linear process of financial time series, it is proved that machine learning models have been regarded as noise-resistant models with more effective solutions. This found is especially reflected in the analysis of CNN model. Recall that in the third model, we employ a CNN before passing into LSTM. Even though CNN successfully brings the benefits of overfitting reduction and data de-noising, its forecast still turned worse in the end. One of the reasons may be the performance instability of the long short term memory model when analyzing different stocks.

3.2 Evaluation methodology

The first evaluation metric used for comparison is MSE. From Table 1, the lowest MSE, on average, was captured in CNN model, and the second best one occurred on the LSTM model for the index BAC. CNN+LSTM has the largest MSE around 0.0125, while its performance ranks the second for the index PG. Furthermore, a loss or a summation of the errors on each case is calculated on training. With regard to the loss plots in appendix, the green, red and blue lines are the respective loss curves for the CNN, LSTM and CNN+LSTM models. We see that CNN has a significant advantage as the minimum loss for the proposed model is the lowest among all. In contrary, the loss for the combined model has a slowest dropping speed for both stocks.

In some cases, a closer predictive curve does not equal a higher prediction accuracy. Yet, the accuracy and the correlation are generally positively correlated. That is to say, the better the evaluation is, the shorter the distance between the prediction and real data is.

3.3 Earnings using different models and strategies

Table 2 illustrates the rate of return using 4 indicated strategies above. Most of them are positive around 11% using all in/all out strategy in PG stock, but this trading strategy is risky because people are likely to lose the majority of principles in an early stage if the model prediction is not accurate enough. On the contrary, by looking at BAC rate of returns, most of them tend to have a deficit. The second and the third algorithm are both conservative, but the daily investment method can test our model prediction on trends as it highly depends on price movements each day. As for the last buy & hold trading algorithm without using any models, it generate profits if and only if the overall trend is increasing.

Overall, from the model specific rate of returns, CNN and LSTM tend to have better performance in trading with more earnings and less financial loss. The cumulative returns dependent on the actual price trends. If a particular index has a rising tendency during the test period on the whole, investors may have a higher probability to earn money regardless of which trading systems are used. That is possibly because our trading methods are not quick enough to response to declines, compared with growths.

4. Conclusion and Discussion

In summary, we address the implementation and the comparison of convolution neural network and long short memory to financial time series prediction. As discussed above, the trading system based on the prediction of a single CNN outperforms with a relatively higher cumulative returns compared to LSTM and CNN+LSTM. One of the reasons that impacts of CNN as a deeper input gate are not obvious is a lack of features and noises in this experiment. Moreover, our study length is only 2 year daily close price, therefore, one of the further improvements is the extension of study length and depth (i.e weekly, hourly trading). Due to the computational limitation, a limited number of model parameters is trained. Thus, future study will also introduce more random noises and parameters values. At the same time, we can take into account the effects of frequent transaction costs, build more professional trading algorithms with prior knowledge to create profitable portfolios, then step up some API calls to create real accounts to perform daily trading in the real market.

Lastly, we want to remind all that buy stocks more thoughtfully and rationally. Our model is just for reference. We are not responsible for any of your personal loss in the market.

Reference

Ehsan Hoseinzade, S. H. (2019). *CNNpred: CNN-based stock market prediction using a diverse set of variables.* ResearchGate.

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). *Speech recognition with deep recurrent neural networks.* IEEE Xplore.

Guanting Chen, Y. C. (2017). *Application of Deep Learning to Algorithmic Trading.*

Jialin Liu, F. C.-C.-M. (2019). *Stock Prices Prediction using Deep Learning Models.*

MITCHELL, C. (2020, August 5). *How to Use a Moving Average to Buy Stocks.* Retrieved from Investopedia: https://www.investopedia.com/articles/active-trading/052014/how-use-moving-average-buy-stocks.asp

PALMER, B. (2019, March 4). *Defining the Force Index.* Retrieved from Investopedia: https://www.investopedia.com/articles/trading/03/031203.asp

Rikiya Yamashita, M. N. (2018, June 22). *Convolutional neural networks: an overview and application in radiology.* Retrieved from SpringLink: https://link.springer.com/article/10.1007/s13244-018-0639-9

SRIVASTAVA, P. (2017, December 10). *Essentials of Deep Learning : Introduction to Long Short Term Memory.* Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/

Xavier Glorot, A. B. (2011). *Deep Sparse Rectifier Neural Networks.* Proceedings of Machine Learning Research.

Appendix

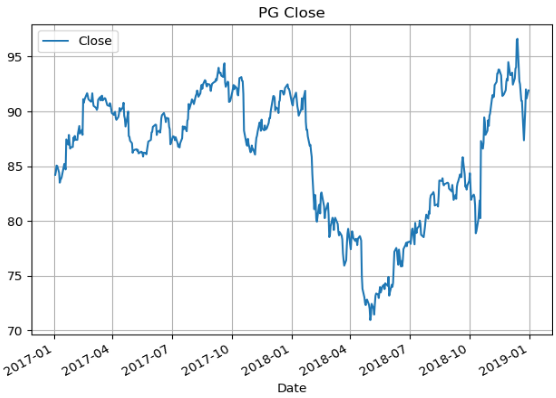

Figure 1 ：Closing price of PG

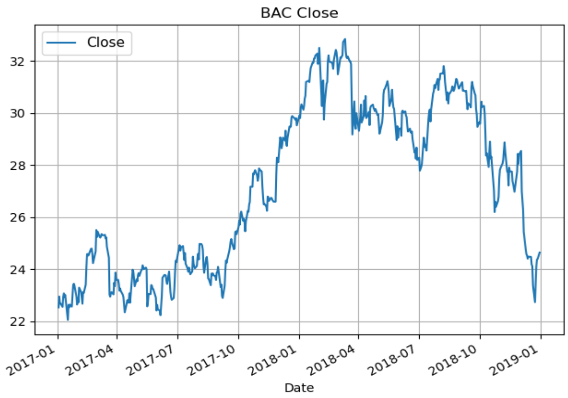

Figure 2 ：Closing price of BAC

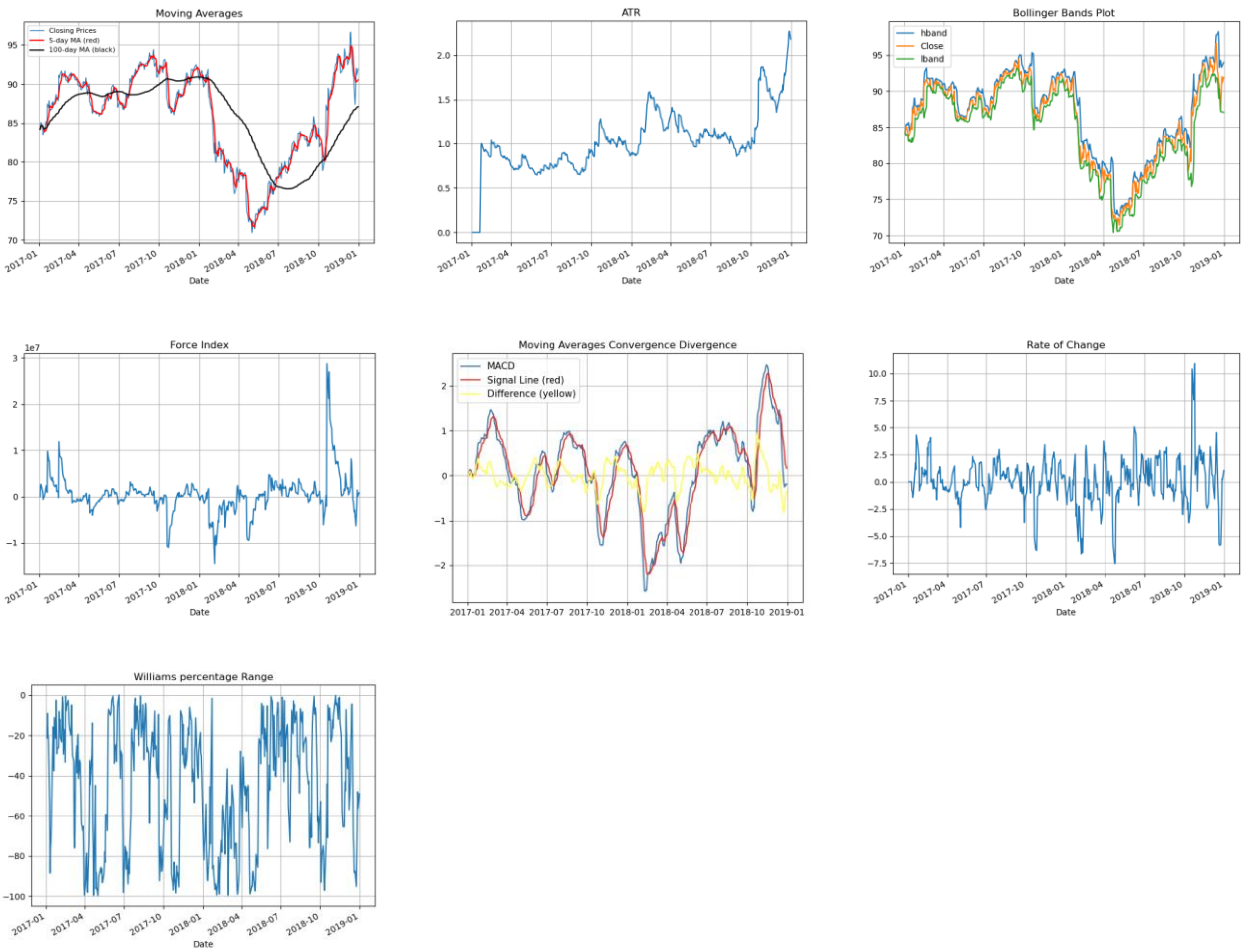

Figure 3 ：Technical indicators for PG

Figure 4 ：Technical indicators for BAC

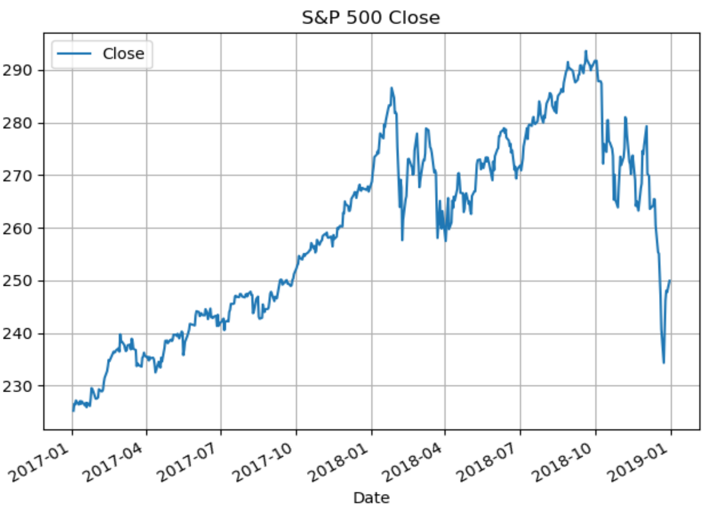

Figure 5: S&P 500 closing price

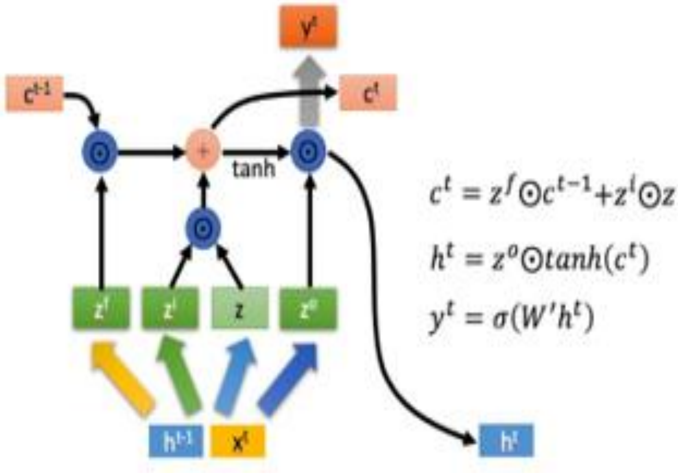

Figure 6: LSTM architecture

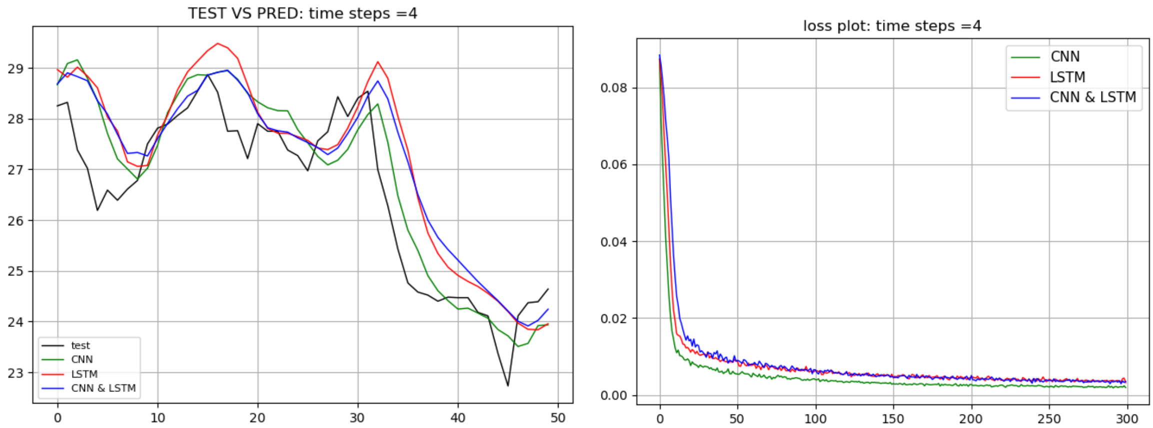

Figure 7: BAC Test against predict using 3 different models with time steps equal to 4

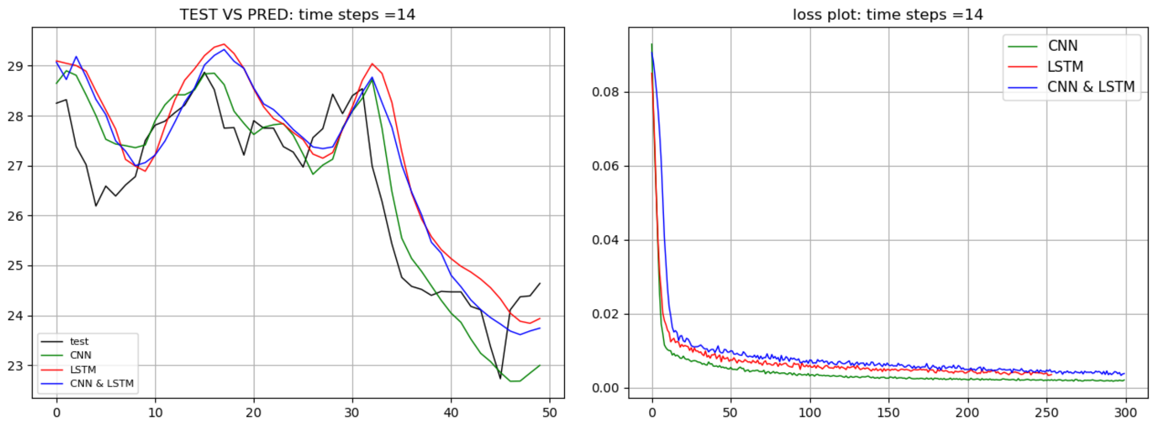

Figure 8: BAC Test against predict using 3 different models with time steps equal to 14

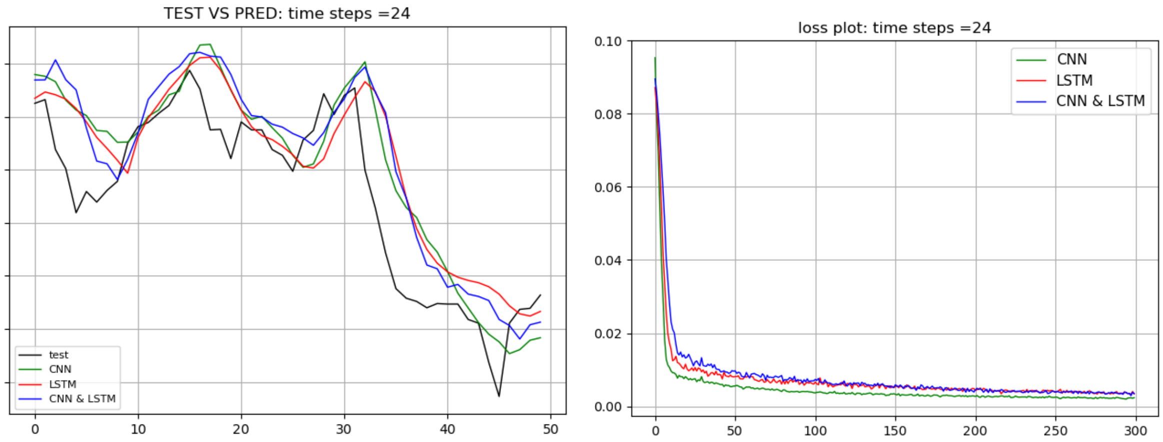

Figure 9: BAC Test against predict using 3 different models with time steps equal to 24

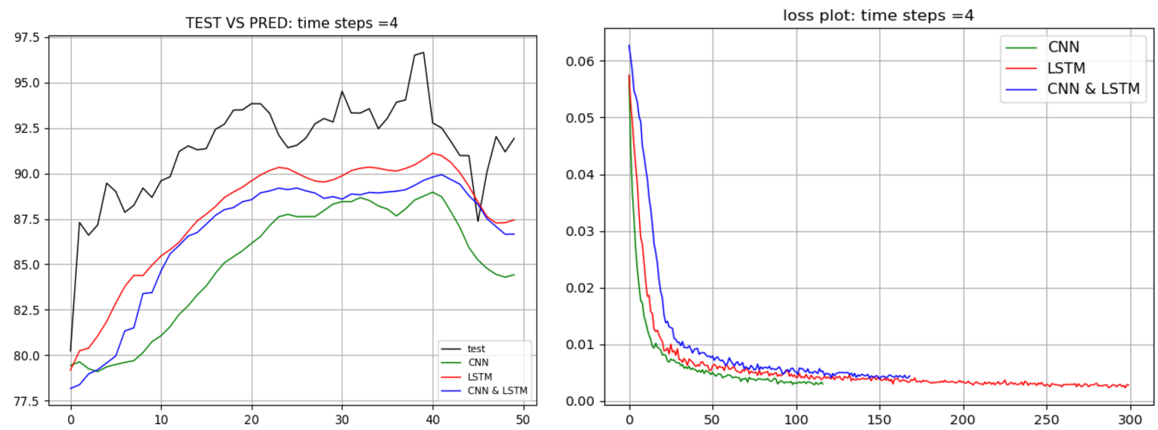

Figure 10: PG Test against predict using 3 different models with time steps equal to 24

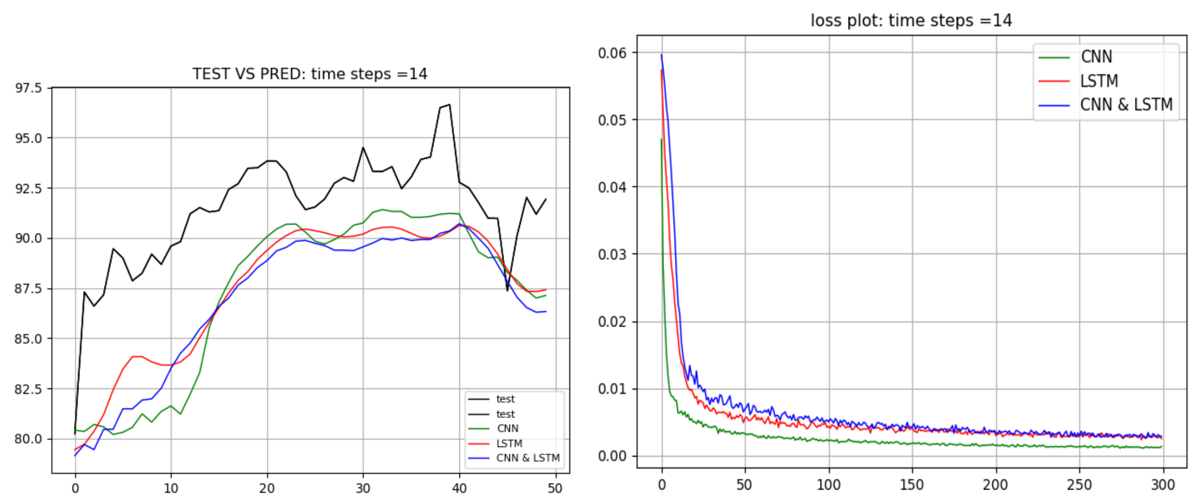

Figure 11: PG Test against predict using 3 different models with time steps equal to 24

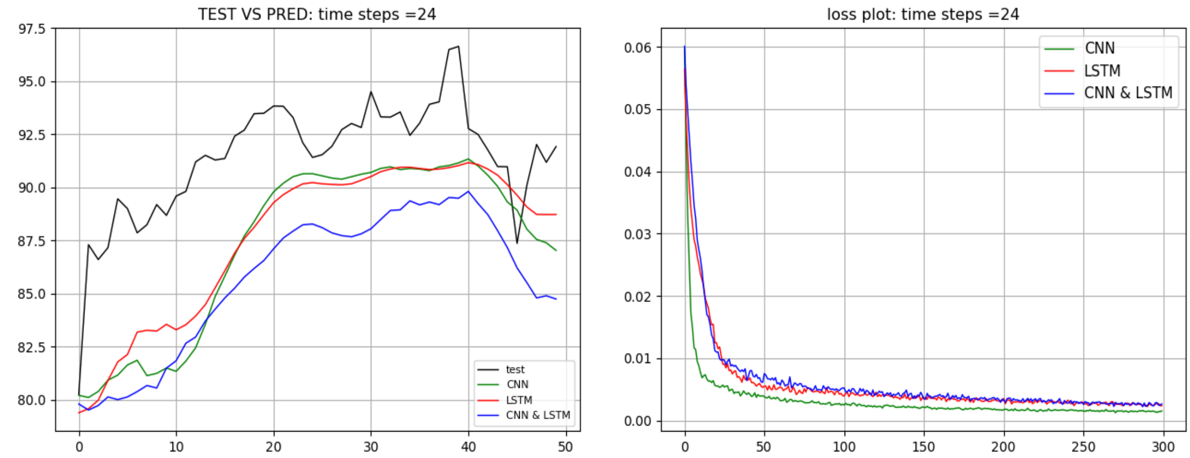

Figure 12: PG Test against predict using 3 different models with time steps equal to 24

	LSTM	CNN	CNN + LSTM
Average MSE for PG	0.03	0.014	0.026
Average MSE for BAC	0.0103	0.00945	0.0125

Table 1 ：average MSE table for two stocks using different models

	LSTM			CNN & LSTM			CNN			No model
Time steps	All in all Out	Weights	Buy/ Sell one unit	All in all Out	Weights	Buy/ Sell one unit	All in all Out	Weights	Buy/ Sell one unit	Buy & hold
PG										
4 days	12.70%	0.39%	9.94%	13.60%	0.32%	13.13%	15.00%	0.07%	16.00%	14.00%
14 days	11.34%	0.24%	9.11%	15.24%	0.26%	16.02%	5.20%	0.10%	3.10%	14.00%
24 days	11.19%	0.22%	8.85%	-0.35%	0.07%	15.45%	1.40%	0.09%	10.32%	14.00%
BAC										
4 days	-2.53%	-1.05%	9.36%	-1.28%	-1.41%	9.86%	4.63%	-1.74%	21.40%	-12.78%
14 days	-1.82%	-1.75%	10.62%	-1.45%	-1.68%	9.52%	4.30%	-1.47%	20.71%	-12.78%
24 days	-1.58%	-1.55%	11.12%	1.54%	-1.57%	15.42%	3.71%	-1.54%	19.63%	-12.78%

Table 2 ：Rate of return using different models and strategies