

# Deep Stock: training and trading scheme using deep learning

Sungwoo Kang

**Abstract**—Despite the efficient market hypothesis, many studies suggest the existence of inefficiencies in the stock market, leading to the development of techniques to gain above-market returns, known as alpha. Systematic trading has undergone significant advances in recent decades, with deep learning emerging as a powerful tool for analyzing and predicting market behavior. In this paper, we propose a model inspired by professional traders that look at stock prices of the previous 600 days and predicts whether the stock price rises or falls by a certain percentage within the next  $D$  days. Our model, called DeepStock, uses Resnet’s skip connections and logits to increase the probability of a model in a trading scheme. We test our model on both the Korean and US stock markets and achieve a profit of  $N\%$  on Korea market, which is  $M\%$  above the market return, and profit of  $A\%$  on US market, which is  $B\%$  above the market return.

**Index Terms**—deep learning, machine learning, trading, back-testing, quantitative trading, stock

## I. INTRODUCTION

The stock market has always been a challenging arena to navigate due to its complexity and unpredictability. Although it is thought that it is impossible to make an above-the-market profit because of the efficient market hypothesis, it seems this is not the case. Many researches show there’s some inefficiency in the market [1]–[3], and many techniques have been developed to gain this marginal gain, called the alpha.

Systematic trading has undergone significant advances in recent decades, with new technologies and approaches transforming the field. Early approaches use simple indicators such as EMA, bollinger bands, RSI, etc. More sophisticated approaches came into play such as factor models using value, momentum, volatility, growth as factors, and complicated linear (AR, MA, ARIMA) and non-linear algorithms (ARCH, GARCH). Machine learning techniques were also used to find these factors.

In recent years, deep learning has emerged as a powerful tool for analyzing and predicting market behavior, leveraging the capabilities of artificial neural networks to learn complex patterns and relationships in financial data. Almost all deep learning techniques have been applied to gain above market returns. In the belief that social media precedes future events, attempts to predict stock price by analyzing social media using combining Natural Language Processing exist [4]–[6]. Since stock prices are time series in nature, approaches using LSTMs [7]–[9] have been tried. [10]–[12] tries to find a trading rules using Reinforcement Algorithms. Approaches using time

series values as CNN inputs have been studied [13], [14], as well as providing image of the stock chart to the CNN model instead of numerical values [15].

However, professional traders somehow make consistent profit without these sophisticated approaches. By just looking at the charts, these traders find out major and minor trends as well as support and resistance levels, buy and sell if that trend is expected to continue or break.

Inspired by these traders, we try to mimic their process. Our model using deep learning looks at stock prices of the previous 600 days and predicts whether the stock price rises or falls by 10% (20% for US stocks) within the next  $D$  days. We believe that discretizing returns is important to facilitate learning and avoid overfitting. We use 600 days because we think this period is enough to find effective long term and short term trends. Using Resnet’s skip connections [16], we expect the model will be able to effectively find which trends are more important. In order to increase the probability of success in our trading scheme and to reduce the number of stocks that need to be traded, we only consider stocks whose softmax output from our model exceeds a certain threshold. By setting this threshold, we are able to focus on stocks that the model predicts with higher confidence, potentially increasing the chances of achieving profitable trades. A style of trading that attempts to capture short- to medium-term gains in a stock over a few days to several weeks is commonly known as swing trading. Thus we name our model DeepStock.

Both Korean and US stock markets have been tested, giving a profit of  $N\%$  on Korea market, which is  $M\%$  above the market return, and profit of  $A\%$  on US market, which is  $B\%$  above the market return.

Contributions are followings.

- training and trading scheme using deep learning models
- Use of logits to increase the probability of a model in a trading scheme
- simple model architecture that predicts whether the stock will go up or down by a given percentage.

The paper is organized as follows. The first section is an introduction to the subject. In the second section, we give a brief explanation and differences from previous research. In the third section, we explain our approach and on the forth section we show the following results, explaining the limits and what might go wrong when employing this model in real-life trading at the last.

## II. RELATED WORK

Using a sliding window and assigning labels with discretized return rate is not new. By discretizing the return rate,

Sungwoo Kang is with the Department of Electrical and Computer Engineering, Korea University, Seoul 02841, Republic of Korea e-mail: xxx@korea.ac.kr.

we can overcome small price change sensitivity and facilitate model training. [17] discussed the effect of N-Period Min Max labeling instead of just predicting whether the price will go up or down and used XGBoost model to trade accordingly. While they assign labels according to which side of the upper or lower limit of the N-day period is hit first, we assign labels whether the price hits fixed return rate 10% (or 20%) for a given period. We use fixed return rate of 10% because VI is activated for Korea stocks when price rises by 10%. We set 20% for US stocks since US stocks are more volatile. [13] used a 2-dimensional CNN to predict the next day's return of commonly used ETFs. 28 features with 28-day window was passed to it which include close price, volume, several technical indicators such as RSI, SMA, MACD. [14] used autoencoder composed of stacked restricted Boltzmann machines. They used 12 monthly returns and 20 daily returns as features and predict whether the stock's return will be above or below median return of whole stocks. Our approach differs from these approaches in that we use only ohlcv price data without any human-made features, uses long window of 600 days to capture price trends, assign labels by deciding whether stock will reach given percentage. Our model also differs in that we use sample weighting to improve label imbalance and utilize softmax logits for better accuracy and cumulative return. Hence, our model significantly outperforms the market.

### III. ALGORITHM

#### A. Dataset Preparation and Preprocessing

We use daily stock prices scraped from Naver Finance [18] for Korean stocks, and Stooq [19] historical data for US stocks. We took log-price since using the original price led to exploding gradients. We look at the stock price over the previous 600 days and determine the label as the stock price rises or falls by N% within the next D day compared to the close price. N was set to 10% for Korean stocks and 20% for US stocks. D was tested for 3,5,10,20,30. In the 5-day 10% scheme, a label of 1 is given if the highest price of the next 5 days is 10% or higher, -1 if the lowest price is less than -10%, and 0 if both the highest and lowest prices are in the range of -10% to 10% during the next 5 days. The 5-day 10% model and 10-day 10% model will have the same label of 1 if the price reaches 10% on day 3 and falls -10% on day 4. Even if both the highest and lowest prices reached the set limits on the same day, they were treated as label 1 in training.

A well-chosen back-testing period is important to avoid over-fitting. We choose the training period to be 2006-2015, and the validation period of 2016-2019, both of which include strong bear and bull markets, and markets that show no trend. The model was tested from 2020 to 2022. For US dataset, tickers that have close price above 2000\$ and below 2\$ were excluded in training.

Dataset statistics for Korea 10% and US 20% dataset are shown on table I and II. Each item in the table represents a percentage of 10% rise, 10% fall, and no significant rise or fall separated by '/'. It can be seen that the longer the labeling period, the higher the rise/fall ratio and the lower the sideways ratio. This can be interpreted as establishing a trend

period	train	val	test
3	0.12/0.08/0.80	0.09/0.04/0.87	0.13/0.07/0.81
5	0.18/0.13/0.70	0.14/0.08/0.78	0.19/0.11/0.70
10	0.28/0.21/0.50	0.23/0.16/0.61	0.29/0.21/0.50
15	0.35/0.27/0.38	0.29/0.22/0.48	0.35/0.27/0.38
20	0.40/0.32/0.29	0.34/0.27/0.39	0.39/0.31/0.29
30	0.45/0.37/0.18	0.40/0.33/0.26	0.44/0.36/0.20

TABLE I: dataset statistics for Korea 10% label

period	train	val	test
3	0.02/0.01/0.97	0.01/0.01/0.98	0.03/0.02/0.95
5	0.03/0.02/0.95	0.03/0.01/0.96	0.05/0.03/0.91
10	0.06/0.05/0.89	0.05/0.03/0.92	0.10/0.07/0.83
15	0.09/0.07/0.84	0.08/0.05/0.88	0.14/0.11/0.75
20	0.11/0.10/0.79	0.10/0.07/0.84	0.18/0.14/0.69
30	0.16/0.14/0.71	0.14/0.10/0.76	0.23/0.18/0.58

TABLE II: dataset statistics for us 20% label

over a longer period of time even for stocks that have been sideways. Also, there is a significant class imbalance between label 0 and label 1, up to 9:1. Hence we trained another model using sample weighting.

#### B. Model Architecture

We expect the Resnet [16] architecture's skip connections to capture major and minor trends, lower layers to capture short-term trends and higher layers the long-term. We used one-dimensional Resnet with a kernel size of 5, and a depth of 6. Fully Connected Layers were added to the end. Binary Cross Entropy (BCE) was used for loss function. TSAI framework [20] was used for implementing the model. The same model using varying kernel sizes 3,5,10,15,20,30 (TrendU) and 30,20,15,10,5,3 (TrendD) were also trained. We thought that these varying kernel sizes would capture various short and long-term trends, but they were shown to be ineffective compared to fixed kernel size. Because a significant class imbalance exists, we trained another Resnet model with a weighted loss function, weighting  $1/(\text{label 1 percentage})$  to label 1, other labels weighted the same with 1 (rwResnet). All models were trained for 50 epochs, with 16-bit precision. Training took 1.5 days for Korea stock and 5 days for US stock.

### IV. RESULTS

#### A. Test Set Results

Test accuracy of all labels, and accuracy of label 1 of four models (Resnet, TrendU, TrendD, rwResnet) on Korea dataset are shown. The accuracy of all labels tends to decrease as the labeling period increases, while label 1 accuracy tends to increase. We believe this is due to a decrease in the proportion of stocks moving sideways (label 0) and an increase in proportion of stocks rising (label 1). All models except rwResnet show similar results. Using varying kernel size

compared to fixed size kernel shows no significant difference. rwResnet shows lower accuracy at all labels but significantly outperforms others in the label 1 classification, hence we used this model for later backtesting and training on US dataset. For US dataset, rwResnet all label accuracy seems to decrease as labeling period increases. No significant tendencies were shown for label 1 accuracy. Proportion of the label 1 of the whole dataset is shown beside splitted by a slash.

period	Resnet	rwResnet	TrendU	TrendD
3	0.8424	0.6732	0.8408	0.8418
5	0.7374	0.5424	0.7298	0.7365
10	0.5674	0.4701	0.5552	0.5663
15	0.4958	0.4531	0.4608	0.4943
20	0.4665	0.4501	0.4488	0.4771
30	0.4717	0.4735	0.4484	0.4579

TABLE III: all label accuracy of Korea stocks

period	Resnet	rwResnet	TrendU	TrendD
3	0.2116	0.7107	0.2172	0.2409
5	0.2583	0.8118	0.2277	0.2019
10	0.3587	0.881	0.3332	0.3025
15	0.5145	0.9173	0.3824	0.4222
20	0.5781	0.9358	0.436	0.5614
30	0.6969	0.923	0.4964	0.5548

TABLE IV: label 1 accuracy of Korea stocks

period	all labels	label 1
3	0.8034/1.0000	0.8366/0.0265
5	0.7579/1.0000	0.8238/0.0452
10	0.7098/1.0000	0.7841/0.0893
15	0.6940/1.0000	0.6907/0.1280
20	0.6501/1.0000	0.7452/0.1619
30	0.6052/1.0000	0.6610/0.2178

TABLE V: rwResnet acuracy of US stocks

### B. Thresholding

The softmax output (logits) of a model is commonly believed to indicate the level of confidence in the result of an image recognition task, such as identifying 90% happiness and 5% surprise in emotion recognition. However, can this concept be applied to trading domains? By setting a certain threshold value for the softmax output, we can predict results with higher confidence and increase the probability of success, leading to improved cumulative profit. Our research shows that this is indeed the case.

We present the label accuracy of rwResnet on Korea and US datasets, demonstrating how the accuracy increases as the threshold value is raised. Each column represents a labeling period, while each row represents a given threshold value, with "base" representing a threshold of 0. The accuracy and

proportion of the dataset are displayed for each item, divided by a slash and rounded to 4 decimal places. A value of 0 indicates that there is no corresponding output.

The same trend is observed when applying thresholding to the label one accuracy of rwResnet, as well as when thresholding on different models like TrendD and TrendU. You can find the full results in the appendix.

Apart from improving the probability of success, there is another reason to use thresholding in trading. Currently, there are around 4000 (Korea) and 5000 (US) tradable tickers available. Even trading 1% of the dataset would still result in about 40-50 tickers daily, which is not practical and would incur hefty fees. By setting a threshold value, we can narrow down the list of stocks to be traded, reducing the number of trades and associated fees.

Now, the question arises: does trading only in stocks with high threshold values result in a high cumulative return? Our research shows that this is indeed the case.

	base	0.9	0.95	0.995	0.9995
3	0.6732	0.9609/0.1190	0.9780/0.0435	0.9926/0.0148	0.9970/0.0034
5	0.5424	0.8968/0.0543	0.9318/0.0305	0.9806/0.0136	0.9974/0.0021
10	0.4701	0.8300/0.0359	0.9021/0.0212	0.9627/0.0024	1.0000/0.0000
15	0.4531	0.8212/0.0251	0.9131/0.0135	0.9731/0.0004	1.0000/0.0000
20	0.4501	0.6903/0.0366	0.8090/0.0118	0.9231/0.0001	1.0000/0.0000
30	0.4735	0.6902/0.0241	0.7908/0.0062	0.8630/0.0000	0

TABLE VI: all label accuracy Thresholding rwResnet on Korea dataset

	base	0.9	0.95	0.995	0.9995
3	0.8034	0.9310/0.5023	0.9593/0.3602	0.9932/0.1000	0.9975/0.0369
5	0.7579	0.9021/0.4373	0.9417/0.2884	0.9942/0.0700	0.9979/0.0225
10	0.7098	0.8839/0.3415	0.9348/0.2000	0.9859/0.0464	0.9919/0.0170
15	0.6940	0.8468/0.3695	0.8926/0.2235	0.9562/0.0486	0.9687/0.0177
20	0.6501	0.8152/0.2670	0.8655/0.1385	0.9351/0.0270	0.9474/0.0115
30	0.6052	0.7712/0.2424	0.8063/0.1208	0.8387/0.0116	0.8502/0.0025

TABLE VII: all label accuracy Thresholding rwResnet on US dataset

### C. Backtest Results

We use the label reweighted Resnet model (rwResnet) for backtesting. The model was tested for 2020-2022 rebalanced every three months with original cash of 10 million won for Korea stocks and ten thousand dollars for US stocks. Trading commissions and taxes were taken into account. We assume we can sell whole amount when the given percentage was reached, hence only +/-10(20)% of P&L (Profit and Loss) will occur during single trade. Threshold of softmax logits were chosen where 0.003 to 0.005 percent of dataset proportion was reached by thresholding. Entry ratio were set to 0.1 mostly and 0.3 if the dataset proportion was below 0.001. Postfix 's' (for sidecut) was assigned if the stock was sold at the closing price of that day if the stock go sideways during the labeling peridiod. Postfix 'w' was assigned if the stock was held even if it passed the labeling period until it reaches the given percentage (+/-10% for Korean stocks, +/-20% for US).

rwResnet\_3s(0.9995)(0.1) refer to model trained for labeling period of 3 days, threshold 0.9995, entry ratio 0.1, and sells at the close price if the stock went sideways during the labeling period. In real trading scenario, 10% (20%) drop followed by a 10% (20%) increase will result in loss, and vice versa as a gain. However since we don't have access to minute data of all tickers, we don't know whether it profited if the price fluctuate twice the given percentage in the given period. Thus we provide two backtest results one treating all stocks that fluctuated twice the given percentage as profit (base backtest), and another which treat these as loss (conservative backtest). Actual returns will lie somewhere in between. Full table is shown in the appendix.

	simple sum	compund
KOSPI/KOSDAQ	26.02/34.52	1.14/1.18
rwResnet_3s(0.9995)(0.1)	109.23	2.63
rwResnet_3w(0.9995)(0.1)	126.37	2.89
rwResnet_5s(0.9995)(0.1)	151.17	3.86
rwResnet_5w(0.9995)(0.1)	118.28	2.95
rwResnet_10s(0.995)(0.1)	128.36	2.95
rwResnet_10w(0.995)(0.1)	135.59	3.31
rwResnet_15s(0.99)(0.1)	118.40	2.75
rwResnet_15w(0.99)(0.1)	124.93	2.85
rwResnet_20s(0.99)(0.3)	84.14	2.03
rwResnet_20w(0.99)(0.3)	63.15	1.61
rwResnet_30s(0.95)(0.1)	70.38	1.71
rwResnet_30w(0.95)(0.1)	97.57	2.35

TABLE VIII: Korea base backtest result

	simple sum	compund
KOSPI/KOSDAQ	26.02/34.52	1.14/1.18
rwResnet_3s(0.9995)(0.1)	67.65	1.75
rwResnet_3w(0.9995)(0.1)	64.78	1.59
rwResnet_5s(0.9995)(0.1)	136.10	3.31
rwResnet_5w(0.9995)(0.1)	114.30	2.82
rwResnet_10s(0.995)(0.1)	110.83	2.48
rwResnet_10w(0.995)(0.1)	111.73	2.67
rwResnet_15s(0.99)(0.1)	111.02	2.57
rwResnet_15w(0.99)(0.1)	121.26	2.74
rwResnet_20s(0.99)(0.3)	84.14	2.03
rwResnet_20w(0.99)(0.3)	63.15	1.61
rwResnet_30s(0.95)(0.1)	48.36	1.34
rwResnet_30w(0.95)(0.1)	70.97	1.76

TABLE IX: Korea conservative backtest result

What stocks does the model buy? It tends to buy stocks that have broke through the resistance area of 200-300 days with a long white body candle, regardless of labeling period it was trained on. Example charts are shown in the appendix with 5, 10, 20, 60, 120, 240 exponential moving average.

#### D. Discussions and limits

Slippage was not taken into account. We assumed that all the stocks held can be sold within one tick. Although this might

be difficult for institutions and super ants, but there will be no problem for individual investors. Stock gaps were not taken into account, the same fixed profit or loss was taken even when the day's open price was formed above or below the fixed percentage. In real-life trading, the investor may get profit/loss more than the specified percentage. Delisted tickers were not taken into account. About 2% to 3% of Korean and US stocks are delisted annually. This percentage might not be few, but we think this risk can be reduced by abstaining from trading during the applicable compliance period. Our model buys stock at exact close price at the end of the market. OTC transactions were not taken into account. In real trading scenario, one might have to buy before market end, or place an order the next day at the previous day's close price.

#### REFERENCES

- [1] N. Jegadeesh and S. Titman, "Returns to buying winners and selling losers: Implications for stock market efficiency," *The Journal of finance*, vol. 48, no. 1, pp. 65–91, 1993.
- [2] C. S. Asness, T. J. Moskowitz, and L. H. Pedersen, "Value and momentum everywhere," *The journal of finance*, vol. 68, no. 3, pp. 929–985, 2013.
- [3] E. F. Fama and K. R. French, "Dissecting anomalies," *The journal of finance*, vol. 63, no. 4, pp. 1653–1678, 2008.
- [4] Y. Xu and V. Keselj, "Stock prediction using deep learning and sentiment analysis," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 5573–5580.
- [5] H. Maqsood, I. Mehmood, M. Maqsood, M. Yasir, S. Afzal, F. Aadil, M. M. Selim, and K. Muhammad, "A local and global event sentiment based efficient stock exchange forecasting using deep learning," *International Journal of Information Management*, vol. 50, pp. 432–451, 2020.
- [6] Y. Xu and V. Keselj, "Stock prediction using deep learning and sentiment analysis," in *2019 IEEE international conference on big data (big data)*. IEEE, 2019, pp. 5573–5580.
- [7] Z. Zou and Z. Qu, "Using lstm in stock prediction and quantitative trading," *CS230: Deep Learning*, Winter, pp. 1–6, 2020.
- [8] P. Ghosh, A. Neufeld, and J. K. Sahoo, "Forecasting directional movements of stock prices for intraday trading using lstm and random forests," *Finance Research Letters*, vol. 46, p. 102280, 2022.
- [9] M. Fabbri and G. Moro, "Dow jones trading with deep learning: The unreasonable effectiveness of recurrent neural networks," in *Data*, 2018, pp. 142–153.
- [10] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, and H. Fujita, "Adaptive stock trading strategies with deep reinforcement learning methods," *Information Sciences*, vol. 538, pp. 142–158, 2020.
- [11] X.-Y. Liu, Z. Xiong, S. Zhong, H. Yang, and A. Walid, "Practical deep reinforcement learning approach for stock trading," *arXiv preprint arXiv:1811.07522*, 2018.
- [12] H. Yang, X.-Y. Liu, S. Zhong, and A. Walid, "Deep reinforcement learning for automated stock trading: An ensemble strategy," in *Proceedings of the first ACM international conference on AI in finance*, 2020, pp. 1–8.
- [13] M. U. Gudelek, S. A. Boluk, and A. M. Ozbayoglu, "A deep learning based stock trading model with 2-d cnn trend detection," in *2017 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2017, pp. 1–8.
- [14] L. Takeuchi and Y.-Y. A. Lee, "Applying deep learning to enhance momentum trading strategies in stocks," in *Technical Report*. Stanford University Stanford, CA, USA, 2013.
- [15] R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua, "Using deep learning neural networks and candlestick chart representation to predict stock market," *arXiv preprint arXiv:1903.12258*, 2019.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [17] Y. Han, J. Kim, and D. Enke, "A machine learning trading system for the stock market based on n-period min-max labeling using xgboost," *Expert Systems with Applications*, vol. 211, p. 118581, 2023.
- [18] <https://finance.naver.com/>.
- [19] <https://stoq.com/>.

- [20] I. Oguiza, “tsai - a state-of-the-art deep learning library for time series and sequential data,” Github, 2022. [Online]. Available: <https://github.com/timeseriesAI/tsai>

APPENDIX A  
FULL THRESHOLD RESULT

Resnet	3	5	10	15	20	30
base	0.2116/0.1265	0.2583/0.1852	0.3587/0.2871	0.5145/0.3515	0.5781/0.3939	0.6969/0.4412
0.9	0.4605/0.0438	0.7315/0.0299	0.9077/0.0154	0.7801/0.0014	0.9568/0.0028	0.6364/0.0001
0.95	0.6820/0.0266	0.9219/0.0204	0.9846/0.0095	0.9119/0.0001	0.9979/0.0006	0.0455/0.0000
0.99	0.9680/0.0106	0.9996/0.0064	0.9980/0.0017	0	0	0
0.995	0.9945/0.0062	1.0000/0.0023	0.9920/0.0003	0	0	0
0.9995	1.0000/0.0001	0	0	0	0	0

TABLE X: Thresholding Resnet on Korea dataset

	base	0.9	0.95	0.99	0.995	0.9995
3	0.6732	0.9609/0.1190	0.9780/0.0435	0.9897/0.0172	0.9926/0.0148	0.9970/0.0034
5	0.5424	0.8968/0.0543	0.9318/0.0305	0.9741/0.0173	0.9806/0.0136	0.9974/0.0021
10	0.4701	0.8300/0.0359	0.9021/0.0212	0.9588/0.0068	0.9627/0.0024	1.0000/0.0000
15	0.4531	0.8212/0.0251	0.9131/0.0135	0.9613/0.0022	0.9731/0.0004	1.0000/0.0000
20	0.4501	0.6903/0.0366	0.8090/0.0118	0.8902/0.0008	0.9231/0.0001	1.0000/0.0000
30	0.4735	0.6902/0.0241	0.7908/0.0062	0.8399/0.0004	0.8630/0.0000	0

TABLE XI: all label accuracy Thresholding rwResnet on Korea dataset

	base	0.9	0.95	0.995	0.9995
label_3_tp10_ls10	0.6821/1.0000	0.9189/0.2618	0.9701/0.1525	0.9853/0.0513	0.9854/0.0284
label_5_tp10_ls10	0.6340/1.0000	0.9228/0.1822	0.9665/0.1052	0.9792/0.0305	0.9814/0.0130
label_10_tp10_ls10	0.5507/1.0000	0.8409/0.1234	0.9201/0.0630	0.9529/0.0084	0.9524/0.0017
label_15_tp10_ls10	0.5193/1.0000	0.7991/0.1006	0.8815/0.0399	0.8602/0.0007	0.8670/0.0001
label_20_tp10_ls10	0.4880/1.0000	0.7769/0.0686	0.8818/0.0244	0.9001/0.0010	0.7764/0.0001
label_30_tp10_ls10	0.4751/1.0000	0.7366/0.0431	0.8101/0.0148	0.7479/0.0007	0.5968/0.0001
label_3_tp20_ls20	0.8034/1.0000	0.9310/0.5023	0.9593/0.3602	0.9932/0.1000	0.9975/0.0369
label_5_tp20_ls20	0.7579/1.0000	0.9021/0.4373	0.9417/0.2884	0.9942/0.0700	0.9979/0.0225
label_10_tp20_ls20	0.7098/1.0000	0.8839/0.3415	0.9348/0.2000	0.9859/0.0464	0.9919/0.0170
label_15_tp20_ls20	0.6940/1.0000	0.8468/0.3695	0.8926/0.2235	0.9562/0.0486	0.9687/0.0177
label_20_tp20_ls20	0.6501/1.0000	0.8152/0.2670	0.8655/0.1385	0.9351/0.0270	0.9474/0.0115
label_30_tp20_ls20	0.6052/1.0000	0.7712/0.2424	0.8063/0.1208	0.8387/0.0116	0.8502/0.0025

TABLE XII: all label accuracy Thresholding rwResnet on US dataset

	0.0	0.9	0.95	0.995	0.9995
3	0.8366/0.0265	0.9615/0.0115	0.9736/0.0073	0.9954/0.0019	0.9948/0.0004
5	0.8238/0.0452	0.9622/0.0168	0.9781/0.0093	0.9951/0.0016	0.9987/0.0004
10	0.7841/0.0893	0.9438/0.0217	0.9584/0.0086	0.9841/0.0008	0.9984/0.0003
15	0.6907/0.1280	0.8537/0.0305	0.8899/0.0136	0.9575/0.0011	0.9891/0.0002
20	0.7452/0.1619	0.9228/0.0319	0.9480/0.0121	0.9654/0.0005	0.9249/0.0001
30	0.6610/0.2178	0.8574/0.0371	0.8992/0.0148	0.9787/0.0011	0.9958/0.0002

TABLE XIII: label 1 accuracy Thresholding rwResnet on US dataset

APPENDIX B  
FULL BACKTEST RESULT

*A. Conservative backtest using rwResnet on Korea Stocks*

	2020.1-3	2020.4-6	2020.7-9	2020.10-12	2021.1-3	2021.4-6	2021.7-9	2021.10-12	2022.1-3	2022.4-6	2022.7-9	2022.10-12	simple sum	compound
KOSPI	-19.33	38.01	10.50	21.86	3.97	6.78	-6.50	-1.38	-7.75	-14.86	-6.50	1.22	26.02	1.14
KOSDAQ	-15.57	53.71	16.57	12.82	-2.19	6.65	-3.13	5.16	-8.45	-20.75	-7.79	-2.51	34.52	1.18
5s(0.9995)(0.1)	1.59	9.85	26.07	24.19	10.97	41.69	-2.02	12.59	7.98	-13.12	2	14.31	136.10	3.31
5w(0.9995)(0.1)	5.86	20.94	20.84	18.89	9.98	25.79	5	3.99	-0.16	-11.87	2.07	12.97	114.30	2.82
10s(0.995)(0.1)	-9.98	34.82	28.29	40.47	2.33	16.05	-14.21	2.86	6.68	-1.62	-11.36	16.5	110.83	2.48
10w(0.995)(0.1)	-0.98	37.68	15.45	15.93	12.81	15.1	-12.94	10.98	4.97	-4.01	-4.11	20.85	111.73	2.67
15s(0.99)(0.1)	-6.88	44.64	10.05	20.39	10.15	27.07	-9.72	3.77	4.6	-12.4	4.08	15.27	111.02	2.57
15w(0.99)(0.1)	-5.03	51.67	10.85	23.92	11.89	29.72	-16.88	7.05	11.98	-6.97	2.04	1.02	121.2	2.74
20s(0.99)(0.3)	0.03	35.98	11.55	28.18	17.62	-3.85	-11.92	6.08	-6.01	-9.81	1.21	15.08	84.14	2.03
20w(0.99)(0.3)	0.03	35.98	0.05	27.01	14.94	3.02	-11.92	3.01	-6.01	-23.98	3	18.02	63.15	1.61
30s(0.95)(0.1)	-15.17	44.24	8.7	4.37	2.09	11.79	-26.98	15.45	10.24	-18.42	-1.03	13.08	48.36	1.34
30w(0.95)(0.1)	-13.88	38.21	15.13	7.06	0.96	3.49	-15.76	23.82	9.94	-14.86	4.84	12.02	70.97	1.76

*B. Base backtest using rwResnet on Korea Stocks*

	2020.1-3	2020.4-6	2020.7-9	2020.10-12	2021.1-3	2021.4-6	2021.7-9	2021.10-12	2022.1-3	2022.4-6	2022.7-9	2022.10-12	simple sum	compound
KOSPI	-19.33	38.01	10.50	21.86	3.97	6.78	-6.50	-1.38	-7.75	-14.86	-6.50	1.22	26.02	1.14
KOSDAQ	-15.57	53.71	16.57	12.82	-2.19	6.65	-3.13	5.16	-8.45	-20.75	-7.79	-2.51	34.52	1.18
3w(0.9995)(0.1)	-18.92	35.26	28.21	27.02	7.98	20.05	-13.88	18.82	5.95	-4.97	0.92	19.93	126.37	2.89
5s(0.9995)(0.1)	1.59	15.48	26.07	24.19	10.97	41.69	0.62	12.59	7.98	-6.32	2	14.31	151.17	3.86
5w(0.9995)(0.1)	5.86	20.94	20.84	18.89	9.98	25.79	5.00	3.99	-0.16	-7.89	2.07	12.97	118.28	2.95
10s(0.995)(0.1)	-2.24	34.82	26.27	40.47	2.33	22.22	-14.21	3.51	6.68	1.38	-11.36	18.49	128.36	2.95
10w(0.995)(0.1)	8.01	37.68	23.54	15.93	11.81	20.88	-12.94	10.98	4.97	-3.01	-4.11	21.85	135.59	3.31
15s(0.99)(0.1)	-6.88	44.64	10.05	25.63	10.15	27.07	-6.59	3.77	4.6	-13.39	4.08	15.27	118.40	2.75
15w(0.99)(0.1)	-5.03	51.67	10.85	23.69	11.89	29.72	-13.89	9.95	11.98	-8.96	2.04	1.02	124.93	2.85
20s(0.99)(0.3)	0.03	35.98	11.55	28.18	17.62	-3.85	-11.92	6.08	-6.01	-9.81	1.21	15.08	84.14	2.03
20w(0.99)(0.3)	0.03	35.98	0.05	27.01	14.94	3.02	-11.92	3.01	-6.01	-23.98	3	18.02	63.15	1.61
30s(0.95)(0.1)	-17.16	35.02	10.67	4.37	6.37	16.9	-22.03	26.39	10.24	-12.44	-1.03	13.08	70.38	1.71
30w(0.95)(0.1)	-9.99	34.17	18.11	7.06	10.96	6.47	-10.94	21.81	9.94	-6.88	4.84	12.02	97.57	2.35

C. Conservative backtest using *rwResnet* on US Stocks

2020.1-3	2020.4-6	2020.7-9	2020.10-12	2021.1-3	2021.4-6	2021.7-9	2021.10-12	2022.1-3	2022.4-6	2022.7-9	2022.10-12	simple sum	compound

D. Base backtest using *rwResnet* on US Stocks

2020.1-3	2020.4-6	2020.7-9	2020.10-12	2021.1-3	2021.4-6	2021.7-9	2021.10-12	2022.1-3	2022.4-6	2022.7-9	2022.10-12	simple sum	compound



## APPENDIX C

## CHARTS THAT MODEL PREDICT STOCK PRICE WILL GO UP

Korea tickers consist of numbers, while US tickers consist of alphabets.



Fig. 1: ticker 05330



Fig. 2: ticker 063570



Fig. 3: ticker 004410



Fig. 4: ticker 101680