Portfolio Investment Based on Neural Networks*

Jia-Hong Liu, Chang-Biau Yang† and Hung-Hsin Chen

Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung 80424, Taiwan

ABSTRACT

In this paper, we combine the trading signals generated by the gene expression programming (GEP) method of Lee *et al.* and the portfolio rebalance generated by the convolutional neural network (CNN) method of Jiang *et al.* to form a stock investment method with portfolio management. The investing targets are 213 listed companies existing during 1995/1/5 through 2017/12/31 in Taiwan stock market. The sliding window scheme is adopted, and the trading began on 2002/1/2 and ended on 2017/12/31, totally 3965 days.

We perform three experiments, trading top-100 stocks with the top-100 stock features, trading top-100 stocks with the 213 stock features, and trading 213 stocks with the 213 stock features. The annualized returns of the three experiments are 25.00%, 26.52% and 27.32%, respectively, which is better than buy-and-hold strategy (the annualized returns of top-100 and 213 stocks are 12.36% and 12.21%, respectively.) and using only the method of Lee *et al.* (the annualized returns of top-100 and 213 stocks are 12.94% and 12.67%, respectively.).

***Keywords*-stock investment; portfolio; conventional neural network (CNN); gene expression programming (GEP)**

I. INTRODUCTION

The stock investment problem can be divided into two main parts, the trading strategy for deciding the timing of buying and selling, and the portfolio rebalance for deciding the weight of buying or selling each stock in the portfolio.

The portfolio rebalance is a fundamental issue in the financial field, and extensive research has been conducted. In the survey of Li and Hoi [11], the portfolio rebalance techniques can be divided into four categories: follow-the-winner, follow-the-loser, pattern-matching approaches, meta-learning algorithms.

The financial market condition varies over the time changing. Searching profitable trading and suitable portfolio rebalance on the rapid changing market are difficult. To overcome these problems, people usually use the historical features to profile condition and to make decision. After collecting enough historical information, several evolutionary algorithms, including the *genetic algorithm* (GA) [5], [15], *genetic programming* (GP) [4], [6], [9], and *gene expression programming* (GEP) [1], [3], [10], can be used to learn profit strategies from these features.

*This research work was partially supported by the National Science Council of Taiwan under contract MOST 104-2221-E-110-018-MY3.

†Corresponding author: cbyang@cse.nsysu.edu.tw

The *artificial neural network* (ANN) has been widely used for solving the investment problems. Deng *et al.* [2] used the *deep neural network* to produce stock trading signals, Ticknor [14] used the *Bayesian regularized neural network* to predict stock prices, and Jiang *et al.* [7] used the *convolutional neural network* (CNN) to select cryptocurrency portfolio. These researches mainly focus on how to predict the price correctly, instead of earning profit in the investment. Our method in this paper is aimed at the profit maximization of trading with the CNN.

Our method hybridizes the GEP method of Lee *et al.* [10] for evolving profitable trading strategy, and the CNN method of Jiang *et al.* [7] for the portfolio rebalance to stock investment. The method of Lee *et al.* used 27 technical indicators in the GEP for evolving profitable strategies, and then the trading signals are generated. The method of Jiang *et al.* [7] does a cryptocurrency trade every 30 minutes, and their method rebalances the portfolio each transaction according to the CNN. The CNN takes the highest price, the lowest price, and closing price in the past 30 minutes as input features, and then it exports *return on investment* (ROI) of the next (subsequent) 30 minutes as output. The objective function of the CNN is to optimize the ROI with the input features.

The investment targets of this paper are 213 listed companies existing during 1995/1/5 through 2017/12/31 in Taiwan stock market. The trading interval began on 2002/1/2 and ended on 2017/12/31, totally 3965 days. There are three experiments, trading 100 stocks with the 100-stock features, trading 100 stocks with the 213-stock features, and trading 213 stocks with the 213-stock features. The annualized returns for the three experiments are 25.00%, 26.52% and 27.32%, respectively. The returns of our method are better than buy-and-hold strategy and the method of Lee *et al.* [10].

The rest of the paper is organized as follows. In Section II, we introduce the GEP method of Lee *et al.* and the *convolutional neural network* (CNN) method of Jiang *et al.* In Section III, the details of our method are described. The experimental results of our method are shown in Section IV, and the conclusions and future works are given in Section V.

II. PRELIMINARIES

A. The Trading Strategy Method of Lee et al.

The GEP method of Lee *et al.* [10] produces trading strategies according to the *portfolio index* (PI), which is

average cumulative daily return of the given 100 stocks. The 100 stocks, as shown in Table I, are the listed companies with the largest capitalization on 1995/1/5 in Taiwan stock market.

Table I
THE TOP 100 STOCKS WITH THE LARGEST CAPITAL ON 1995/1/5 IN TAIWAN STOCK MARKET.

Symbol	Name	Symbol	Name	Symbol	Name	Symbol	Name	Symbol	Name
1101	台泥	1326	台化	1605	華新	2010	春源	2601	益航
1102	亞泥	1402	遠東新	1608	華榮	2014	中鴻	2603	長榮
1103	嘉泥	1409	新纖	1609	大亞	2015	豐興	2605	新興
1104	環泥	1414	東和	1704	榮化	2103	台橡	2606	裕民
1108	幸福	1416	廣豐	1710	東聯	2105	正新	2608	大榮
1109	信大	1417	嘉裕	1712	興農	2107	厚生	2609	陽明
1110	東泥	1418	東華	1718	中纖	2201	裕隆	2610	華航
1201	味全	1419	新紡	1802	台玻	2204	中華	2704	國賓
1210	大成	1434	福懋	1810	和成	2303	聯電	2801	彰銀
1216	統一	1440	南紡	1902	台紙	2308	台達電	2809	京城銀
1218	泰山	1444	力麗	1903	士紙	2311	日月光	2812	台中銀
1229	聯華	1446	宏和	1904	正隆	2315	神達電腦	2820	華票
1301	台塑	1449	佳和	1905	華紙	2330	台積電	2903	遠百
1303	南亞	1456	怡華	1907	永豐餘	2371	大同	2905	三商行
1304	台聚	1503	士電	1909	榮成	2501	國建	2913	農林
1305	華夏	1504	東元	2002	中鋼	2504	國產	2915	潤泰全
1310	台苯	1507	永大	2006	東鋼	2505	國揚	9906	欣巴巴
1312	國喬	1513	中興電	2007	燁興	2506	太設	9907	統一實
1313	聯成	1603	華電	2008	高興昌	2511	太子	9908	大台北
1314	中石化	1604	聲寶	2009	第一銅	2515	中工	9945	潤泰新

There are five main steps in the method of Lee *et al.* to produce consensus signals, including buying, selling, or waiting, on the trading day, described as follows [10].

Step 1: Profitable strategy training
Historical data are divided into many template intervals of L-day length. Each template interval with the previous template interval has a 10-day distance. The GEP evolves 10 profitable strategies in each template interval.

Step 2: Similar template searching
The current trading day is denoted as t. Then the leading interval, with length L, is from day $t-L+1$ to t. The DTW is utilized to search the template intervals similar to the leading interval.

Step 3: Validation checking
Each profitable strategy got from Step 2 is checked whether it is suitable or not in the validation interval, from day $t-L_v+1$ to the current day t. A trading strategy is suitable if it satisfies one of the following two conditions.

1) The return of BAH (buy-and-hold) strategy is greater than 0, and the return of the profitable strategy is greater than 80% of BAH.
2) The return of BAH strategy is less than or equal to 0, and the return of the profitable strategy is greater than or equal to 0.

Step 4: Consensus signal voting
After the suitable strategies are collected, the majority vote is utilized to decide the consensus trading signal (buying, selling or waiting signal). The consensus trading signal is determine by the three proportion values α_{all}, α_{buy} and α_{sell} as follows.

$$\begin{aligned} \alpha_{all} &= \frac{V^{buy}+V^{sell}}{V^{buy}+V^{sell}+V^{wait}}, \\ \alpha_{buy} &= \frac{V^{buy}}{V^{buy}+V^{sell}}, \\ \alpha_{sell} &= \frac{V^{sell}}{V^{buy}+V^{sell}}, \end{aligned} \tag{1}$$

where V^{buy}, V^{sell} and V^{wait} are the voting counts for buying, selling and waiting signal, respectively. The final consensus trading signal is determined as follows.

$$Signal = \begin{cases} Buying & \text{if } \alpha_{all} > \theta_{all} \text{ and} \\ & \alpha_{buy} > \theta_{buy-and-sell}, \\ Selling & \text{if } \alpha_{all} > \theta_{all} \text{ and} \\ & \alpha_{sell} > \theta_{buy-and-sell}, \\ Waiting & \text{otherwise,} \end{cases} \tag{2}$$

where θ_{all} and $\theta_{buy-and-sell}$ are the predefined thresholds. When a buying signal is got, all the capital is used to buy PI. When a selling signal is got, PI is sold. When a waiting signal is got, nothing is done.

Step 5: Computing return
Steps 2 to 4 are repeated on every trading day until the end of the testing interval. Then, the overall return of the testing interval is computed.

B. Convolutional Neural Network Method Of Jiang' et al.

The CNN method of Jiang *et al.* for trading cryptocurrencies contains several neural layers. The input layer is composed by a tensor, which is made of three price matrices, including the closing price matrix $\mathbf{P}_t^{(cl)}$, the highest price matrix $\mathbf{P}_t^{(hi)}$ and lowest price matrix $\mathbf{P}_t^{(lo)}$.

These three price matrices are composed of price vectors on various trading periods. Assuming that the current period is t, the length of the feature interval is L_f, and there are m assets. The dimension of the three price matrices are all $m \times L_f$. So, the dimension of the input tensor $X_t = (\mathbf{P}_t^{(cl)}, \mathbf{P}_t^{(hi)}, \mathbf{P}_t^{(lo)})$ is $m \times L_f \times 3$.

$\mathbf{P}_t^{(cl)}$, $\mathbf{P}_t^{(hi)}$ and $\mathbf{P}_t^{(lo)}$ are calculated as follows.

$$\begin{aligned} \mathbf{P}_t^{(cl)} &= \left[\mathbf{p}_{t-L_f+1}^{(cl)} \odot \mathbf{p}_t^{(rec)} \mid \mathbf{p}_{t-L_f+2}^{(cl)} \odot \mathbf{p}_t^{(rec)} \mid \cdots \mid \mathbf{p}_{t-1}^{(cl)} \odot \mathbf{p}_t^{(rec)} \mid \mathbf{1}\right], \\ \mathbf{P}_t^{(hi)} &= \left[\mathbf{p}_{t-L_f+1}^{(hi)} \odot \mathbf{p}_t^{(rec)} \mid \mathbf{p}_{t-L_f+2}^{(hi)} \odot \mathbf{p}_t^{(rec)} \mid \cdots \mid \mathbf{p}_{t-1}^{(hi)} \odot \mathbf{p}_t^{(rec)} \mid \mathbf{p}_t^{(hi)} \odot \mathbf{p}_t^{(rec)}\right], \\ \mathbf{P}_t^{(lo)} &= \left[\mathbf{p}_{t-L_f+1}^{(lo)} \odot \mathbf{p}_t^{(rec)} \mid \mathbf{p}_{t-L_f+2}^{(lo)} \odot \mathbf{p}_t^{(rec)} \mid \cdots \mid \mathbf{p}_{t-1}^{(lo)} \odot \mathbf{p}_t^{(rec)} \mid \mathbf{p}_t^{(lo)} \odot \mathbf{p}_t^{(rec)}\right], \\ \mathbf{p}_t^{(k)} &= \left[p_{1,t}^{(k)}, p_{2,t}^{(k)}, \cdots, p_{m,t}^{(k)},\right]^\top, \ k \in \{cl, hi, lo\} \\ \mathbf{p}_t^{(rec)} &= \left[1/p_{1,t}^{(cl)}, 1/p_{2,t}^{(cl)}, \cdots, 1/p_{m,t}^{(cl)}\right]^\top, \end{aligned} \tag{3}$$

where $\mathbf{p}_t^{(cl)}$, $\mathbf{p}_t^{(hi)}$, $\mathbf{p}_t^{(lo)}$, and $\mathbf{p}_t^{(rec)}$ are the closing price vector, highest price vector, lowest price vector, and the reciprocal of closing price vector of the m assets on period t, respectively; $p_{i,t}^{(cl)}$, $p_{i,t}^{(hi)}$ and $p_{i,t}^{(lo)}$ are the closing price, lowest price and highest price of the asset i on period t, respectively; $\odot$ is element-wise product operator of two vectors. Each column vector of $\mathbf{P}_t^{(cl)}$, $\mathbf{P}_t^{(hi)}$ and $\mathbf{P}_t^{(lo)}$ is element-wise divided by $\mathbf{p}_t^{(cl)}$ on period t for normalization.

The CNN model of Jiang *et al.* contains three hidden layers, where each neuron in the hidden layers uses $ReLU(x) = max(0, x)$ as its active function. The portfolio rebalance is the process for reallocating proportion of assets to keep back the original objective of investment.

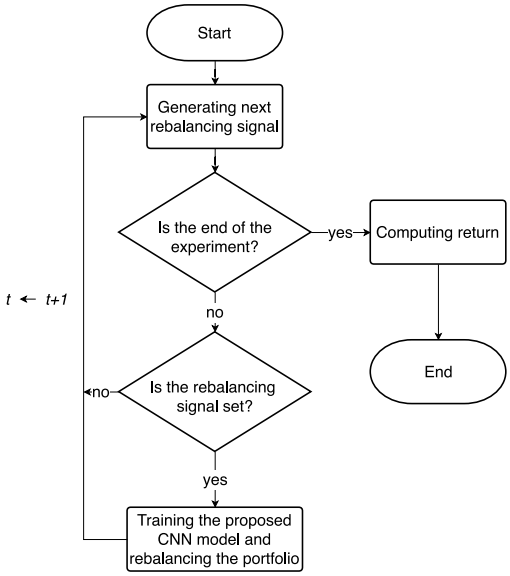

Figure 1. The flowchart of our method.

Considering the cost of rebalancing portfolio, the previous portfolio is also viewed as input to the network. The output layer of the CNN is composed of the *softmax* function, and the output values represent the weight vector of investing capital in $m+1$ assets (m risky assets and a risk-free asset) on period t, also called the *portfolio*.

The dimension of the filter in the CNN model is $channel \times length \times weight$. The filter with length 1 implies that the price of each asset is considered independently with each other before reaching the output layer. The independent structure considers only the features on same asset, so the model can be trained more easily than the large amount of features for multiple assets.

III. Our Method

Our stock investment method hybridizes the GEP method of Lee *et al.* [10] for evolving profitable trading strategy to decide the trading signals, and the CNN method of Jiang *et al.* [7] for rebalancing the capital weights of the portfolio. The flowchart of our method is shown in Figure 1.

Figure 2 illustrates the relationship among the intervals in our method. The whole experiment interval starting from $t=1$ to L_e contains the historical and testing intervals, and the length of the historical interval varies with the τth rebalancing signal sg_τ. The variable τ represents that there are totally τ rebalancing signals are set from the first day of the testing interval to the current trading day $t = s[\tau]$, and there are N_g rebalancing signals in the testing interval on the current trading day $t = s[N_g]$. The historical interval of the τth rebalancing signal starts from $t=1$ to $s[\tau]$, and the time-varied historical interval is divided into training and validation intervals with length $L_{t,\tau}$ and $L_{w,\tau}$, respectively. Assume that N_s sampling intervals extracted from the training interval are used to train our CNN model, where the length of a sampling interval is L_s. The sliding window scheme is adopted

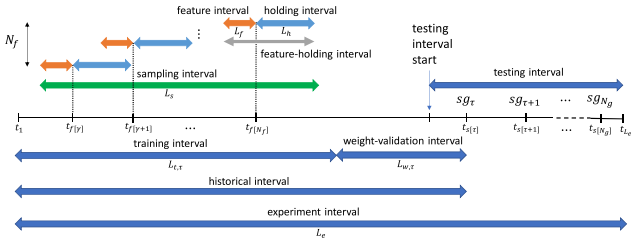

Figure 2. The relationship among the intervals in our method.

to divide a sampling interval into several feature and holding intervals with length L_f and L_h, respectively. The variable γ denotes the γth feature-and-holding interval and $t = f[\gamma]$ is the first day of the corresponding holding interval. If there are N_f feature-and-holding intervals, then we have $L_s = L_f + N_f \times L_h$, since we use the sliding window with shifting L_h days.

There are three main steps of our method, as described in the following.

Step 1: Generating the rebalancing signal
Firstly, the GEP method of Lee *et al.* is utilized to generate consensus trading signal on each trading day in the testing interval. The rebalancing signal is determined when two consecutive consensus signals change. More precisely, the value of a rebalancing signal is $True$ if the consensus signal is changed from buying to selling, or selling to buying; otherwise it is $False$. Assuming that there are totally N_g rebalancing signals in the testing interval.

Step 2: Deciding whether the rebalancing is set or not
If the value of the current rebalancing signal is $False$, the capital weights of the portfolio is not changed until the next rebalancing signal is set. When the value of a rebalancing signal is $True$, the capital weights of the portfolio is adjusted according to the CNN model.

Step 3: Training the CNN model
We propose two convolutional neural network (CNN) models for adjusting the capital weights of portfolio, which are modified from the CNN model of Jiang *et al.* [7]. The main difference of the two proposed models is the dimensions of the input and output vectors. The first CNN model focuses on the features of all candidate stocks, while the second CNN model focuses on the selected stocks in the candidate set.

Figure 3 shows our first CNN model, whose structure is almost the same as the CNN model of Jiang *et al.*, except that the cash bias (the risk-free asset) is removed. The variables m and L_f in the input layer represent the number of stocks in the candidate set and the length of historical prices of an stock, respectively. In the first CNN model, the number of stocks in the output layer is the same as the input layer. Since the cash bias is removed, this CCN model considers only the maximal profit.

Figure 4 shows our second CNN model. In the second model, several stocks in the candidate set are selected in advance, and the all information in the candidate set is utilized to train the model. The number of the pre-selected

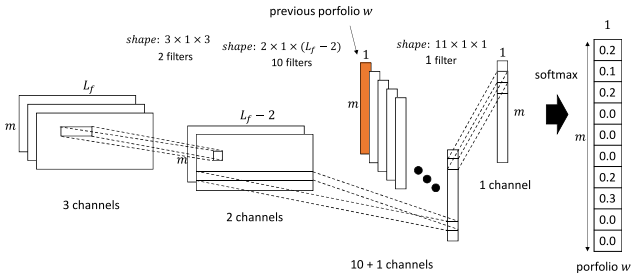

Figure 3. Our first CNN model.

stocks and the stocks in the candidate set are denoted as l and m, respectively.

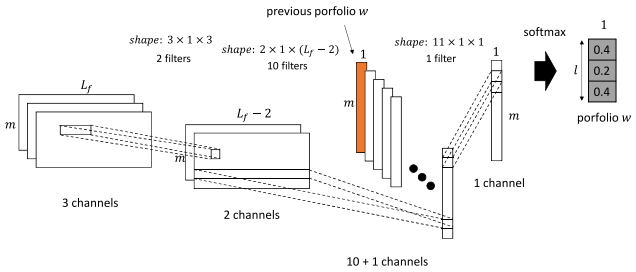

Figure 4. Our second CNN model, which outputs the portfolio of the specified l stocks.

If a rebalancing signal is set, which means that the capital weights in the portfolio need to be rebalanced, then the CNN models will be retrained. Training a CNN model requires a very large amount of data, but the available data in the historical interval are not so many. Hence, N_s sampling intervals, each with the same length L_s, before the current rebalancing signal sg_τ are sampled randomly from the corresponding training interval. The training method for the CNN model is shown in Algorithm 1.

Algorithm 1 The training method in the training interval on the current rebalancing signal. ForwardPropagation() is the forward propagation process with the CNN and the sampling interval. UpdateNetworkWeights() is the forward propagation process which updates the CNN weights by maximizing the objective function $\overline{R}$.

Require: The number of sampling interval N_s; the trading day of the τth rebalancing signal sg_τ;
Ensure: CNN;
1: **for** $i = 1$ To N_s **do**
2: A sampling interval SI_i selected from the corresponding training interval;
3: $output \leftarrow$ ForwardPropagation(SI_i, CNN)
4: $\overline{R} = \frac{1}{N_f} \sum_{\gamma=1}^{N_f} \ln(u_{t_{f[\gamma]}} \mathbf{y}_{t_{f[\gamma]}} \cdot \mathbf{w}_{t_{f[\gamma-1]}})$
5: $gradient \leftarrow maximize\, \overline{\mathrm{R}}$
6: UpdateNetworkWeights($gradient$, $output$, CNN)
7: **end for**
8: return(CNN)

The *stochastic gradient descent* [8], [12] and *backpropagation* [13] algorithms with the N_{sp} sampling intervals are adopted for training the CNN. Whenever a rebalancing signal is set, N_s sampling intervals of length L_s are sampled from the corresponding training interval to train the CNN models. Once the CNN model is trained, the corresponding weight-validation interval is used to monitor the training condition.

The length of a holding interval in a sampling interval is L_h days, which means that after the current capital weights of the portfolio are rebalanced, the capital weights are not changed until L_h trading days later. The objective function of our CNN models is to maximize the average logarithmic return $\overline{R}$, with considering the trading fee c. The return $\overline{R}$ represents the average logarithmic return in a sampling interval according to the capital weights adjusted by the CNN model in N_f feature-and-holding intervals. Assume there are m stocks and N_f rebalancing intervals in a sampling interval. The objective function $\overline{R}$ is given as follows.

$$\begin{aligned}
\max \overline{\mathrm{R}} &= \tfrac{1}{N_f} \textstyle\sum_{\gamma=1}^{N_f} \ln\left(u_{t_{f[\gamma]}} \mathbf{y}_{t_{f[\gamma]}} \cdot \mathbf{w}_{t_{f[\gamma-1]}}\right), \\
\mathbf{y}_{t_{f[\gamma]}} &= \left[\frac{p_{1,t_{f[\gamma]}}}{p_{1,t_{f[\gamma-1]}}}, \frac{p_{2,t_{f[\gamma]}}}{p_{2,t_{f[\gamma-1]}}}, \cdots, \frac{p_{m,t_{f[\gamma]}}}{p_{m,t_{f[\gamma-1]}}}\right]^\top, \\
\mathbf{w}_{t_{f[\gamma]}} &= \left[w_{1,t_{f[\gamma]}}, w_{2,t_{f[\gamma]}}, \cdots, w_{m,t_{f[\gamma]}}\right]^\top, \\
& \textstyle\sum_{i=1}^{m} w_{i,t_{f[\gamma]}} = 1, \\
u_{t_{f[\gamma]}} &= 1 - \left|\left|\mathbf{w}_{t_{f[\gamma]}} - \mathbf{w}_{t_{f[\gamma-1]}}\right|\right|_1 \times c,
\end{aligned} \tag{4}$$

where $t_{f[\gamma]}$ is the day for rebalancing portfolio in the γth feature-and-holding interval, u_t is the reduced factor for adjusting portfolio on day $t_{f[\gamma]}$ with considering the trading fee $c \in [0, 1]$, the dot $(\cdot)$ is the standard inner product operator of two vectors, $y_{t_{f[\gamma]}}$ is the price relative vector of the portfolio on day $t_{f[\gamma]}$, $p_{m,t_{f[\gamma]}}$ is the price of the mth asset on day $t_{f[\gamma]}$, and $w_{m,t_{f[\gamma]}}$ is the weight of capital invested in the mth asset on day $t_{f[\gamma]}$.

An example of (4) with two feature-and-holding intervals in a sampling interval is shown in Figure 5. In the example, the cumulative return of the ith stock with $N_f = 2$ feature-and-holding intervals in a sampling interval, and the rebalancing occurs on days 30 and 60 are shown in (5). $R^{cum}_{i,c=0}$ and $R^{cum}_{i,c\neq 0}$ represent the cumulative return of the ith stock in the sampling interval without and with trading fee c, respectively.

$$\begin{aligned}
R^{cum}_{i,c=0} &= \frac{p_{i,t_{90}}}{p_{i,t_{30}}} \times w_{i,t_{30}} \times \frac{p_{i,t_{150}}}{p_{i,t_{90}}} \times w_{i,t_{90}}, \\
R^{cum}_{i,c\neq 0} &= (1 - |w_{i,t_{30}} - 0| \times c) \times \\
& \quad \frac{p_{i,t_{90}}}{p_{i,t_{30}}} \times w_{i,t_{30}} \times \\
& \quad (1 - |w_{i,t_{90}} - w_{i,t_{30}}| \times c) \times \\
& \quad \frac{p_{i,t_{150}}}{p_{i,t_{90}}} \times w_{i,t_{90}}
\end{aligned} \tag{5}$$

where $p_{i,t}$ is the price of asset i on day t, $w_{i,t}$ is the capital weight of asset i on day t.

IV. Experimental Results

Our investment target is the 213 companies in Taiwan stock market which were listed on 1995/1/5 and never delisted during 1995/1/5 through 2017/12/31. The 100 companies of the 213 companies with the largest capital on 1995/1/5 form the candidate set are shown in Table I, and the remaining 113 companies are shown in Table II. The

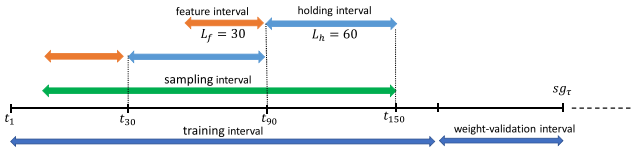

Figure 5. The example with a rebalancing interval of 60 days, a feature interval of 30 days and the number of feature-holding intervals being 2.

historical interval starts from 1995/1/5 to 2016/9/8, totally 5554 days, and the testing interval starts from 2002/1/2 to 2017/12/31, totally 3965 days. Since the algorithm operates with the sliding window, each trading day for testing never overlaps with the historical interval.

Table II
THE REMAINING 113 STOCKS ON 1995/1/5 IN TAIWAN STOCK MARKET WHICH WERE NOT DELISTED BEFORE 2017/12/31.

Symbol	Name	Symbol	Name	Symbol	Name	Symbol	Name	Symbol	Name	Symbol	Name
1203	味王	1437	勤益控	1701	中化	2106	建大	2514	龍邦	5601	台聯櫃
1213	大飲	1438	裕豐	1702	南僑	2108	南帝	2516	新建	9902	台火
1215	卜蜂	1439	中和	1707	葡萄王	2302	麗正	2520	冠德	9904	寶成
1217	愛之味	1441	大東	1708	東鹼	2305	全友	2524	京城	9905	大華
1219	福壽	1442	名軒	1709	和益	2312	金寶	2540	愛山林	9910	豐泰
1220	台榮	1443	立益	1711	永光	2313	華通	2607	榮運	9911	櫻花
1225	福懋油	1445	大宇	1713	國化	2314	台揚	2611	志信	9912	偉聯
1227	佳格	1447	力鵬	1714	和桐	2316	楠梓電	2612	中航	9914	美利達
1307	三芳	1451	年興	1717	長興	2317	鴻海	2701	萬企	9917	中保
1308	亞聚	1452	宏益	1805	寶徠	2321	東訊	2702	華園	9918	欣天然
1309	台達化	1453	大將	1806	冠軍	2323	中環	2705	六福	9919	康那香
1315	達新	1454	台富	1808	潤隆	2324	仁寶	2706	第一店	9921	巨大
1316	上曜	1455	集盛	1906	寶隆	2325	矽品	2816	旺旺保	9933	中鼎
1319	東陽	1457	宜進	2012	春雨	2327	國巨	2901	欣欣		
1410	南染	1459	聯發	2013	中鋼構	2328	廣宇	2904	匯僑		
1413	宏洲	1506	正道	2017	官田鋼	2329	華泰	2906	高林		
1423	利華	1512	瑞利	2020	美亞	2331	精英	2908	特力		
1432	大魯閣	1514	亞力	2101	南港	2332	友訊	5301	寶得利		
1435	中福	1611	中電	2102	泰豐	2373	震旦行	5302	太欣		
1436	華友聯	1612	宏泰	2104	中橡	2509	全坤建	5304	鼎創達		

The top-100 stocks were used by Lee's *et al.* [10] for computing PI, and the remaining 113 stocks provide the additional information for investing in our method. We perform three experiments as follows.

First experiment: The portfolio contains the top-100 stocks and the features are generated by the top-100 stocks. That is, $l = 100$ and $m = 100$. It is performed in our first CNN model.

Second experiment: The portfolio contains the top-100 stocks and the features are generated by the 213 stocks. That is, $l = 100$ and $m = 213$. It is performed in our second CNN model.

Third experiment: The portfolio contains the 213 stocks and the features are generated by the 213 stocks. That is, $l = 213$ and $m = 213$. It is performed in our first CNN model.

In our experiments, several trading strategies are also tested, including the buy-and-hold (BAH), the method of Lee *et al.*, rebalancing per month, rebalancing per quarter, and rebalancing signals decided by the method of Lee *et al.*

The parameters N_f, L_f and L_h are set as follows. The number of the feature-and-holding interval N_f is set to 2. The length of the feature interval $L_f \in \{30, 60, 90\}$ is considered. The length of the holding interval L_h is set according to the different trading strategies, $L_h = 30$ in rebalancing per month, $L_h = 90$ in rebalancing per quarter, and $L_h = 60$ in rebalancing signal decided by Lee *et al.*. Therefore, the total number of parameters for one rebalancing strategy is three (three values of L_f). On each parameter combination, when the first rebalancing signal is set, in order to get better initial internal weights of the CNN models, 30 CNN models are trained, and 20 models with the highest average returns according to the objective function (4) are selected to test the performance of our method. The return of our method is calculated by the average of the 20 models on each rebalancing signal.

The annualized returns of these strategies are shown in Table III, which are grouped by trading stocks and stock features. As one can see, in each group, the parameter of $(Ours, Lee, L_f = 60)$ usually gives a good annualized return (rank 1 or 2). They are denoted as $S(Ours, Lee, L_f = 60, 100, 100)$, $S(Ours, Lee, L_f = 60, 100, 213)$ and $S(Ours, Lee, L_f = 60, 213, 213)$, where the last two numbers are the values of l and m, respectively. Their annualized returns are 25.00%, 26.52% and 27.32%, respectively, which are all higher than *buy and hold* (BAH) and using only signals of Lee *et al.* [10]. The return of $S(Ours, Lee, L_f = 60, 213, 213)$ is the highest since it has more chances to select the stocks with higher profit. The cumulative returns from 2002 to 2017 for the above three cases with the BAH strategy and the method of Lee *et al.* are shown in Figure 6.

Table III
THE ANNUALIZED RETURNS FOR 2002 TO 2009, 2010 TO 2017 AND 2002 TO 2017.

Case	Parameter	2002 to 2009	Rank	2010 to 2017	Rank	2002 to 2017	Rank
Trading 100 stocks with 100 stocks	Buy-and-hold	19.98	7	5.09	10	12.36	8
	Lee	23.75	5	2.95	11	12.94	7
	Ours, Lee, L_f=30	34.8	2	8.48	7	20.92	2
	Ours, Lee, L_f=60	**38.29**	1	**13.31**	4	**25.0**	1
	Ours, Lee, L_f=90	24.1	4	14.25	3	19.05	3
	Month, L_f=30	9.87	9	7.48	8	8.75	10
	Month, L_f=60	11.98	8	14.85	2	13.52	6
	Month, L_f=90	5.91	10	15.07	1	10.46	9
	Quarter, L_f=30	29.39	3	6.22	9	17.29	4
	Quarter, L_f=60	21.09	6	9.1	6	15.04	5
	Quarter, L_f=90	2.38	11	11.51	5	6.95	11
Trading 100 stocks with 213 stocks	Buy-and-hold	19.98	8	5.09	8	12.36	8
	Lee	23.75	7	2.95	10	12.94	7
	Ours, Lee, L_f=30	40.17	3	7.74	6	22.71	3
	Ours, Lee, L_f=60	**44.14**	1	**11.35**	3	**26.52**	2
	Ours, Lee, L_f=90	41.94	2	15.04	2	27.77	1
	Month, L_f=30	-0.32	10	-2.29	11	-1.24	11
	Month, L_f=60	0.59	9	15.36	1	7.79	9
	Month, L_f=90	-5.31	11	9.85	4	2.04	10
	Quarter, L_f=30	28.18	4	4.91	9	16.1	6
	Quarter, L_f=60	27.56	5	6.82	7	16.83	5
	Quarter, L_f=90	26.62	6	8.09	5	17.08	4
Trading 213 stocks with 213 stocks	Buy-and-hold	18.13	5	6.5	10	12.21	6
	Lee	21.71	4	4.24	11	12.67	5
	Ours, Lee, L_f=30	30.92	2	12.95	5	21.3	2
	Ours, Lee, L_f=60	**37.34**	1	**18.33**	4	**27.32**	1
	Ours, Lee, L_f=90	14.89	6	12.03	6	13.54	4
	Month, L_f=30	-34.11	10	20.69	3	-10.82	10
	Month, L_f=60	-43.09	11	24.6	2	-15.79	11
	Month, L_f=90	-16.37	9	33.39	1	5.75	9
	Quarter, L_f=30	14.17	7	7.74	8	10.97	7
	Quarter, L_f=60	28.26	3	7.71	9	17.56	3
	Quarter, L_f=90	10.85	8	9.93	7	10.41	8

The L_f controls the amount of features input to the neural network. Theoretically, more historical features may predict the future more precisely. However, more features need more samples to well train the network. In the stock market, the amount of training samples is usually small, and the feature with farther distance is less related to the current time. So, how to balance the L_f is an importance issue. In addition, the large feature interval that leads to small amount of samples may make us train the network difficultly. In Table III, most parameters with L_f=60 and the method of Lee *et al.* have higher profit than L_f=30. When L_f changes from 60 to 90, the profit

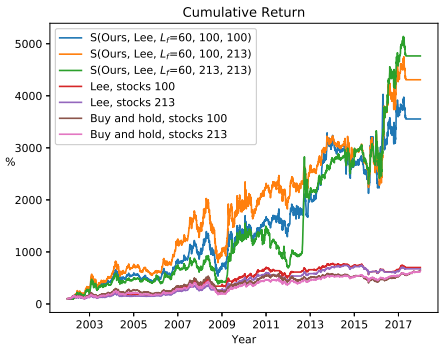

Figure 6. The cumulative returns from 2002 to 2017.

decreases. So, the $L_f = 60$ may be suitable for the amount of our training samples.

For almost cases, the performance of the trading time with a fixed period (month or quarter) is not good. Because at the end of a fixed period, it may not be a good portfolio rebalancing time. The method of Lee *et al.* uses the history features to learn the trading rule by evolution. The trading rule is helpful for finding the good trading time. In our experiment results, our portfolio strategy combined with the trading rules of Lee *et al.* can earn the higher profit.

Our method considers the rebalance of the weights in the portfolio, but does not reserve any cash. By the experimental results, the return of each year is high in the rising trend. However in the descending trend, the method can hardly earn profit. Although our method may lose more than buy-and-hold, it still outperforms that of Lee *et al.*

V. Conclusion

In this paper, we use the trading signals generated by the method of Lee *et al.* as portfolio rebalancing timing, and use the CNN models, inspired by the method of Jiang *et al.*, to adjust the weights of stocks in the portfolio. We propose two CNN models to produce portfolio for stock market. The first CNN model removes cash bias to adapt our objective function, and the second CNN model uses the candidate stocks features for training.

We perform three experiments, including trading top-100 stocks with the top-100 stock features, trading top-100 stocks with the 213 stock features, and trading 213 stocks with the 213 stock features. The annualized returns of the three experiments are 25.00%, 26.52% and 27.32%, respectively. As the experimental results show, our method is better than buy-and-hold and the method of Lee *et al.*. In addition, with the more stock features and the more stock in the candidate set, our method could earn more profit.

In our method, the trading signals for all stocks are the same. However, the suitable trading time may be different for different stocks. In the future, an investor may try to use the portfolio weight as the trading signal, where the weight can be viewed as the strength of a trading signal. Trading each stock at its suitable trading time could earn better profit than trading all stocks at the same time.

References

[1] C. Y. Chou, C. B. Yang, and H. H. Chen, "Portfolio investment based on gene expression programming," *Proc. of the 32nd International Conference on Computers and Their Applications*, Honolulu, Hawaii, USA, pp. 225–230, Mar. 2017.

[2] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Transatction on Neural Networks and Learning Systems*, Vol. 28, No. 3, pp. 653–664, 2017.

[3] C. Ferreira, "Gene expression programming: A new adaptive algorithm for solving problems," *Complex Systems*, Vol. 13, pp. 87–129, 2001.

[4] C. M. Hsu, "A hybrid precedure for stock price prediction by integrating self-organizing map and genetic programming," *Expert Systems with Applications*, Vol. 38, pp. 14026–14036, 2011.

[5] C. F. Huang, B. R. Chang, D. W. Cheng, and C. H. Chang, "Feature selection and parameter optimization of a fuzzy-based stock selection model using genetic algorithms," *International Journal of Fuzzy Systems*, Vol. 14, No. 1, pp. 65–75, 2012.

[6] S. M. Jhou, C. B. Yang, and H. H. Chen, "Taiwan stock forecasting with the genetic programming," *Proc. of the 16th Conference on Artificial Intelligence and Application (Domestic Track)*, Chungli, Taiwan, pp. 151–157, Nov. 2011.

[7] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem." https://arxiv.org/abs/1706.10059, 2017.

[8] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, Vol. 23, pp. 462–466, 1952.

[9] J. R. Koza, *Genetic Programming: On the programming of computers by means of natural selection*, Vol. 1. Cambridge, USA, MA: MIT press, 1992.

[10] C.-H. Lee, C.-B. Yang, and H.-H. Chen, "Taiwan stock investment with gene expression programming," *Procedia Computer Science*, Vol. 35, pp. 137–146, 2014.

[11] B. Li and S. C. H. Hoi, "Online portfolio selection: A survey," *ACM Computing Surveys*, Vol. 46, pp. 1–32, 2014.

[12] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, Vol. 22, pp. 400–407, 1951.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, Vol. 323, pp. 533–536, 1986.

[14] J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, Vol. 40, pp. 5501–5506, 2013.

[15] T. J. Tsai, C. B. Yang, and Y. H. Peng, "Genetic algorithms for the investment of the mutual fund with global trend indicator," *Expert Systems with Applications*, Vol. 38, pp. 1697–1701, 2011.