Expert Systems with Applications xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Evaluation approach to stock trading system using evolutionary computation

I-Cheng Yeh [a,*], Che-hui Lien [b], Yi-Chen Tsai [a]

[a] Department of Information Management, Chung-Hua University, Hsin Chu 30067, Taiwan
[b] Department of Management, Thompson Rivers University, Kamloops, BC, Canada

ARTICLE INFO

Keywords:
Genetic algorithms
Neural networks
Decision system
Stock market
Over-learning

ABSTRACT

The past researches emphasize merely the avoidance of over-learning at the system level and ignore the problem of over-learning at the model level, which lead to the poor performance of the evolutionary computation based stock trading decision-making system. This study presents a new evaluation approach to focus on evaluating the generalization capability at the model level. An empirical study was provided and the results reveal four important findings. First, the decision-making system generated at the model design stage outperforms the system generated at the model validation stage, which shows over-learning at the model level. Secondly, for the decision-making system generated either at the model design stage or at the model validation stage, the investment performance in the training period is much better than that in the testing period, exhibiting over-learning at the system level. Third, employing moving time-frame approach is unable to improve the investment performance at the model validation stage. Fourth, reducing the evolution generation and input variables are unable to avoid the over-learning at the model level. The major contribution of this study is to clarify the issue of over-learning at the model and the system level. For future research, this study developed a more reliable evaluation approach in examining the generalization capability of evolutionary computation based decision-making system.

1. Introduction

The stock market is a high-risk market. The reason is that too many factors affect the stock prices, making the stock prices hard to predict. Nearly a dozen years, many artificial intelligence methods, such as the artificial neural networks (Haykin, 2004), genetic algorithms (Booker, Goldberg, & Holland, 1989; Holland, 1975), and the hybrid approach (Lien & Yeh, 2008), have been applied to the stock price forecast and the buy–sell decision-making. These studies (Armano, Marchesi, & Murru, 2005; Armano, Murru, & Roli, 2002; Enke & Thawornwong, 2005; Hayward, 2004; Kim & Han, 2000; Kwon & Moon, 2003; Lam, 2004; Lee, 2004; Massimiliano, Rushi, Oliver, & Mark, 2004; Olson & Mossman, 2003; Phua, Ming, & Lin, 2001) first attempt to establish a stock price forecast system, and then use the forecast system to develop the buy–sell decision system. Because of the unpredictability of the stock prices, the effect of such a study has its limitations.

Recently, some studies (Allen & Karjalainen, 1999; Kuo, Chen, & Hwang, 2001) adopted new ways of thinking to directly construct the buy–sell decision system, instead of developing a stock price forecast system, and indicated that this strategy can effectively enhance investment performance. This thinking may seem unreasonable, but from the perspective of control theory, not necessarily. Just like most people can learn driving through practice, but very few people really understand vehicle dynamics. In other words, most people can't accurately predict the movement of the vehicle, but still capable of making decision (control the steering wheel).

The genetic neural networks (GNN) are decision-making systems combining the non-linear model of neural networks and the global optimization of genetic algorithms. GNN use the strategy of reinforced learning to skip the construction of stock price forecast system and directly develop the buy–sell decision system (Lien & Yeh, 2008). The rationale is to employ the neural networks as the decision system. The technical indicators based on daily stock prices and trading volume are employed as the system inputs, and the buy–sell decision indicator as the system output. When this decision indicator is greater than an upper threshold, a buying decision is recommended; when this decision indicator is smaller than a lower threshold, a selling decision is suggested. This study uses evolution theory (the survival of the fittest) to produce the optimized decision-making system. The steps of evolution model are detailed as follows:

1. Producing the initial generation.
   Randomly generate the first generation's decision-making systems (i.e., randomly set the connection weights of neural networks).
2. Evaluating each individual (decision-making system).
   Make these systems to trade (buy/sell) under the simulated environment of using the historical data of the stock market

* Corresponding author. Tel.: +886 3 5186052; fax: +886 3 5186546.
E-mail address: icyeh@chu.edu.tw (I.-C. Yeh).

0957-4174/$ - see front matter © 2010 Published by Elsevier Ltd.
doi:10.1016/j.eswa.2010.07.035

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

at a particular period (during training). Then evaluate the performance of the systems based on their profit.
3. Reproducing systems.
Reproduce next generation's systems based on their performance. High performance systems have greater probability to be reproduced.
4. Implementing genetic operations.
Apply the genetic operations (e.g., mutation, crossover) on the reproduced systems to generate a new generation (i.e., reorganize connection weights of neural networks).
5. Alternation of generations.
Replace the systems of old generation with the systems of the new generation.
6. To Check conditions for the termination.
If the termination meets the conditions (e.g., the default number of generations), proceed to step (7), or else return to the step (2).
7. Output the best individual.
Output the best decision-making system of the last generation (i.e., a set of neural network's connection weights).

The decision-making system generated from the above approach can only be guaranteed for a good profit in the training period, but does not guarantee that the use of this system will generate a good profit in the future. Therefore, the stock market historical data must be divided into two parts, the former used for "training" and the latter for "testing". In the second step above, the final profits of the systems were calculated in the training and testing period individually, and in step three, only the profit during the training period affected the reproduction probability. The profit during the testing period only used to evaluate whether the individual (decision-making system) is generalized to the testing periods.

Although the application of genetic neural networks on the stock market may lead to good investment performance, but because of the changing nature of stock markets, the future performance of this original decision-making system to generate a good profit is suspicious. Although the latest data can be loaded into the evolution model to generate the new optimized decision-making system (neural network), its performance depends on the evolution parameters which can be adjusted subjectively by the model builder(s). The evolution parameters include evolution entity (e.g., neural network architecture), evolution operations (e.g., the number of evolution generation), and evolution environment (e.g., the length of training period and testing period). Therefore, if we want to prove objectively that the decision-making system generated by the evolution model in the future has a good performance, the evolution model's parameters must be fixed and only the data be allowed to update.

Table 1
Illustration of model design and model validation.

Model stage	Training period	Testing period	Model parameters
Model design	1982–1991	1992–1994	Adjustable
Model validation 1	1985–1994	1995–1997	Fixed
Model validation 2	1988–1997	1998–2000	Fixed
Model validation 3	1991–2000	2001–2003	Fixed
Model validation 4	1994–2003	2004–2006	Fixed

For example, assume that the data are separated into 10 years interval with 1991–2000 representing a "training period", from 2001 to 2003 a "testing period." Although the system shows good investment performance during both the training and testing period, the only conclusion we can draw is that the decision-making system achieves the generalization of system level. To testify generalization capability of the evolution model, we have to fix the evolution model's parameters, replace the old data by the new ones, and then generate a new decision-making system. If the new system shows a good investment performance during both the new training and testing interval, then we can verify that the evolution model with the specific parameters achieves the generalization at the model level. For example, by adding the latest data (2004–2006), we redefine the time period of training (1994–2003) and testing (2004–2006). If the evolution model can produce a new decision-making system with a good profit during both the new training and testing period, the evolution model shows the generalization capability at the model level.

This process can be performed many times as shown in Table 1 and Fig. 1. Only at the model design stage trial and error can be used to adjust evolution model's parameters to strengthen the evolution model and produce high performance decision-making systems. In the subsequent model validation stage, the model parameters must be fixed and only the data used in the training and testing period can be updated. If the decision-making systems do not have a good profit at the model validation stage, then overfitting of model parameters occurs at the model design stage, which leads to producing a good performance at the model design stage only but not at the model validation stage.

Our study aims at examining whether genetic neural networks have the generalization capability at the model level. Section 2 describes the rationale of stock trading decision system based on GNN. Section 3 discusses the investment performance of GNN-

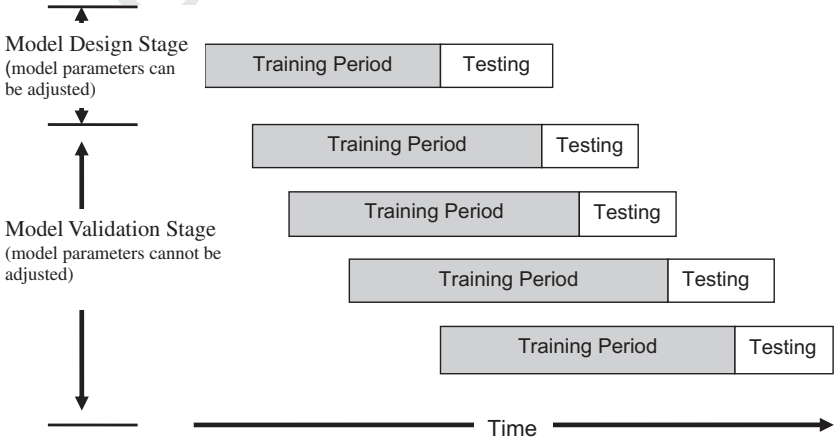

Fig. 1. Model design and model validation stage.

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

based decision-making system generated at the model design stage and at the model validation stage, followed by the discussion of investment performance of GNN-based decision-making system generated at the model validation stage. In Section 5, we present some approaches to improve GNN's generalization capability. Section 6 contains concluding remarks.

2. GNN-based stock trading decision-making system

2.1. The calculation of technical indicators

1. Trend indicator (Durbin Watson, DW)

$$\mathrm{DW}_n = \frac{\sum_{t=1}^{n}(\Delta C_t - \Delta C_{t-1})^2}{\sum_{t}^{n}(\Delta C_t)^2} \tag{1}$$

C_t = the closing price on the tth day; $\Delta C_t = C_t - C_{t-1}$.

2. Relative Strength Indicator (RSI)

$$\mathrm{RSI}_n = \frac{\sum_{t=1}^{n} Max(\Delta C_t, 0)}{\sum_{t=1}^{n}|\Delta C_t|} \tag{2}$$

3. Price Volume Indicator (PVI)

$$\mathrm{PVI}_n = \frac{\sum_{t=1}^{n}\Delta C_t \times V_t}{\sum_{t=1}^{n}|\Delta C_t| \times V_t} \tag{3}$$

V_t = the trading volume on the tth day

4. Price Volume Change Indicator (PVC)

$$\mathrm{PVC}_n = \frac{\sum_{t=1}^{n}\Delta C_t \times \Delta V_t}{\sum_{t=1}^{n}|\Delta C_t \times \Delta V_t|} \tag{4}$$

$\Delta V_t = V_t - V_{t-1}$.

5. Moving Average Indicator (MAI)

$$\mathrm{MAI}_{m,n} = \frac{MA_m}{MA_n} \tag{5}$$

MA_n = n-day moving average of closing prices; MA_m = m-day moving average of closing prices.

6. Moving Average Volume Indicator (MVI)

$$\mathrm{MVI}_{m,n} = \frac{MV_m}{MV_n} \tag{6}$$

MA_n = n-day moving average of trading volume; MA_m = m-day moving average of trading volume.

2.2. Trading rules and the fitness function

This research considers the profit or loss in the training period and in the testing period. Because the objective of this study is to develop an evolution model to produce decision-making system with high investment performance in Taiwan's stock market; therefore, the transaction costs refer to the practical stock transaction costs in Taiwan. These costs include transaction fees of purchasing accounting for 0.1425% of the buying price; the transaction fees and trading tax of selling account for 0.1425% and 0.3% of the selling price, respectively.

The fitness function of the individual (decision-making system) in each evolution generation is the proceeds (capital) at the end of the investment period (training or testing period). If the stock was bought and held until the end of the investment period, we will calculate its market value (stock shares multiplied by the market price); otherwise, the value of the stock sold before the end of the period is used directly as the capital at the end of the period.

2.3. Genetic neural networks (GNN)

2.3.1. Input variables of neural networks

In the input layer, the input variables are composed of those technical indicators. Because it is not consistent regarding the range of each indicator, we have to normalize the indicators first and the transform formula is illustrated as follows:

$$V_{new} = \frac{V_{old} - \mu}{\sigma} \tag{7}$$

where V_{old} = technical indicator before normalization; V_{new} = technical indicator after normalization; μ = mean of the technical indicator; σ = standard deviation of the technical indicator.

2.3.2. Output variable of neural networks

We set one hidden layer in the GNN model. The relationship between the input layer and hidden layer nodes is:

$$h_k = f(net_k) = \frac{1}{1+\exp(-net_k)} \tag{8}$$

$$net_k = \sum_{i=1}^{N_{inp}} W_{ik} \times x_i - \theta_k \tag{9}$$

where net_k is a weighted sum of the input values multiplied by the weights of the respective connections on the kth hidden layer node; x_i is the ith normalized input variable; W_{ik} represents weights connecting node i in the input layer with node k in the hidden layer; θ_k is called the bias of node k in the hidden layer.

The relationship between hidden layer nodes and the output node is:

$$y = f(net) = \frac{2}{1+\exp(-net)} - 1 \tag{10}$$

$$net = \sum_{k=1}^{N_{hid}} W_k \times h_k - \theta \tag{11}$$

where net is a weighted sum of the input values multiplied by the weights of the respective connections on the output node; h_k is the output of node k in the hidden layer; W_k represents weights connecting node k in the hidden layer with the output node; θ is called the bias of the output node.

The y is the output value representing the buy–sell decision. The higher the value y, the more representative of decision to buy; the smaller the value y, the more representative of decision to sell. Therefore,

(1) When y is greater than an upper threshold α, the system shows a signal of buying.
(2) When y is smaller than a lower threshold β, the system shows a signal of selling.

The α and β, like the weights and biases of the neural networks, are determined in the evolution computation.

2.3.3. The objective function of genetic algorithms

Because there is no target value of this output variable, it is not available to employ the traditional steepest decent method to modify the neural network weights. We therefore use genetic algorithms to optimize the weights. The objective function is the fitness. Instead of using the sum of square error, we use capital at the end of the investment period as the fitness. Through evolution computation, the decision-making systems (neural networks) with maximized fitness would be generated. In other words, the weights and biases of the neural networks as well as the buying threshold α ($0 < \alpha < 1$) and selling threshold β ($-1 < \beta < 0$) are optimized to produce maximum profit.

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

2.3.4. Setting the parameters of genetic algorithms

We set the parameters of genetic algorithms as follows: (1) population size: 100, (2) crossover rate: 0.9, (3) mutation rate: 0.01, (4) evolution generations: 30 and (5) the range of network weights: −3 to +3.

In neural network architecture, we employ six nodes standing for six technical indicators in the input layer, six nodes in the hidden layer, and one node standing for the buy–sell decision indicator in the output layer.

2.4. Evaluation of performance of trading system

In order to measure the performance of the GNN-based trading system, our research presents the "coefficient of profitability" defined as follows:

$$\text{Coefficient of profitability} = \frac{\sqrt[n]{\frac{M_e}{M_s}}}{\sqrt[n]{\frac{C_e}{C_s}}} \quad (12)$$

where n represents the length (years) of stock trade; M_s and M_e are the capital at the start and end of the investment period, respectively; C_s and C_e are the closing stock market index at the start and end of the investment period, respectively. When the coefficient of profitability is greater than one, the decision system defeats the market and generates extra profits.

3. The investment performance of decision-making system generated at the model design stage

We used daily data of the Taiwan Stock Market Index (TAIEX) from 5 August 1989 to 12 July 2004. Totally 4060 individual data were collected. The data were divided into two parts by timing and are shown in Fig. 2:

1. Training period: 3000 daily observations were used as training data from 5 August 1989 to 6 April 2000.
2. Testing period: 1060 daily observations were used as testing data from 7 April 2000 to 12 July 2004.

Fig. 2 exhibits the stock market fluctuating in the training and testing period, which shows the representativeness of the data. The evolution model first employs training data to generate a high performance buy–sell decision-making system, and then the system is applied on the testing data to evaluate its performance.

At the model design stage, this study tried to use two different evolution models (GNN network architectures):

1. Short-term indicators GNN employing the following six indicators: DW(10), RSI(10), PVI(10), PVC(10), MAI(5, 20), and MVI(5, 20).
2. Long-term indicators GNN using the following six indicators: DW(20), RSI(20), PVI(20), PVC(20), MAI(1, 50), and MVI(1, 50).

Because of randomness of evolution computation, these two models are re-implemented ten times individually and the results are shown in Table 2. Comparing the short-term and long-term indicators system reveals that, in the training period, the coefficient of profitability (=1.17) of the short-term indicators system is slightly less than that (=1.21) of the long-term indicators system; however, in the testing period, the coefficient of profitability (=1.09) of the short-term indicators system is almost equivalent to that (=1.08) of the long-term indicators system. Therefore, the long-term indicators do not outperform the short-term indicators.

In order to explore the impacts of evolution model's parameters on the performance of the system, this research tried the following different parameters:

1. The range of network weights: −1 to +1 or −3 to +3.
2. Evolution generations: 30 or 100.

This combination of two parameters generated four groups of parameters, which were re-implemented ten times based on the long-term indicators system. The results (shown in Table 3) reveal:

1. *The range of network weights:* During the training period, under the range of network weights between −1 and +1, the resulting coefficients of profitability are 1.17 and 1.20 for 30 generations and 100 generations, respectively. Under the range of network weights between −3 and +3, the resulting coefficients of profitability are 1.20 and 1.23 for 30 generations and 100 generations, respectively. The large range of network weights slightly outperforms the small one. However, during the testing period, the small range of network weights slightly outperforms the large range of network weights.

Fig. 2. TAIEX of training and testing period at the model design stage.

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

Table 2
Comparing the performance of short-term and long-term indicators system at the model design stage.

Serial number	Short-term indicators system				Long-term indicators system			
	Annual profitability (%)		Coefficient of profitability		Annual profitability (%)		Coefficient of profitability	
	Training period	Testing period	Training period	Testing period	Training period	Testing period	Training period	Testing period
1	17.75	6.85	1.17	1.15	18.91	4.22	1.19	1.08
2	18.17	7.05	1.18	1.16	19.80	6.72	1.21	1.11
3	17.52	1.81	1.17	1.10	19.30	−1.19	1.20	1.03
4	17.52	1.74	1.17	1.10	18.87	10.79	1.20	1.15
5	17.14	−1.57	1.17	1.06	20.31	11.58	1.21	1.16
6	16.36	0.68	1.16	1.09	21.4	0.76	1.22	1.05
7	17.14	−12.09	1.17	0.95	20.33	6.85	1.21	1.11
8	17.22	6.04	1.17	1.14	22.14	−13.79	1.23	0.90
9	16.95	4.03	1.17	1.12	20.25	8.92	1.21	1.13
10	17.25	−0.88	1.17	1.07	21.73	8.30	1.23	1.12
Mean	17.30	1.37	1.17	1.09	20.30	4.31	1.21	1.08
Standard deviation	0.5	5.6	0.00	0.06	1.15	7.56	0.01	0.078

Table 3
Comparisons of the performance of evolution model at the model design stage.

Serial number	Range of network weights: (−1, +1)				Range of network weights: (−3, +3)			
	30 generations		100 generations		30 generations		100 generations	
	Training period	Testing period	Training period	Testing period	Training period	Testing period	Training period	Testing period
1	1.19	0.93	1.22	0.90	1.19	0.97	1.22	0.92
2	1.14	1.10	1.19	1.15	1.20	0.92	1.28	0.79
3	1.20	0.94	1.21	0.88	1.21	0.96	1.23	0.97
4	1.16	1.02	1.22	0.91	1.19	0.92	1.23	0.92
5	1.17	1.11	1.18	1.10	1.20	1.09	1.20	1.12
6	1.16	1.18	1.19	1.12	1.19	1.14	1.20	1.16
7	1.19	1.11	1.19	1.12	1.21	1.18	1.25	1.07
8	1.18	1.11	1.19	1.14	1.21	0.86	1.25	0.90
9	1.19	1.11	1.20	1.12	1.19	1.11	1.20	1.14
10	1.17	0.94	1.20	0.89	1.19	1.06	1.22	1.07
Mean	1.17	1.06	1.20	1.03	1.20	1.02	1.23	1.01
Standard deviation	0.02	0.09	0.01	0.12	0.01	0.10	0.02	0.12

2. *Evolution generation*: The coefficient of profitability for 100 generations is slightly higher than the one for 30 generations during the training period. However, in the testing period, the coefficient of profitability for 30 generations is slightly higher than the one for 100 generations.

Combining the discussions above, the long/short term indicators, the different range of network weights, and the different numbers of generations do not have significant impacts on the performance of system during the period of testing.

Fig. 3 shows the time of the buy–sell decision for the testing period. The "white triangles" represent the time of buying and the "black diamonds" stand for the time of selling. It shows that GNN can escape the sharply downward period to avoid losses, but appears too conservative in the rising period.

Fig. 4 is the historical capital chart for the testing period and the capital at the start are set as one million NT$ (NT$ 10,000/per point; 100 points = one million NT$). The darkened line and the fine line demonstrate the results of the GNN-based operation and the buy-and-hold strategy, respectively. The results show that

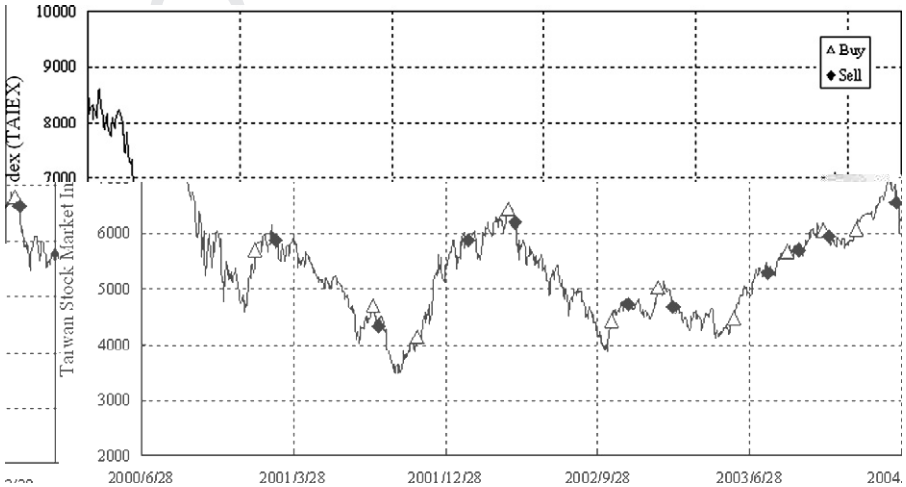

Fig. 3. The time of buy–sell decision during the testing period at the model design stage.

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

Fig. 4. Comparing the performance of TAIEX and the GNN-based strategy during the testing period at the model design stage.

the profitability of the GNN-based decision-making system is stable and a profit gap between the two lines is gradually formed in the testing period.

In the testing interval, TAIEX dropped 32% and, at average, the percentage of annual loss is 7.4%. However, the percentage of overall profitability generated by the GNN-based decision-making system achieved 49% and, at average, the percentage of annual profitability is 8.3%, showing remarkable gains.

4. The investment performance of decision-making system generated at the model validation stage

In Section 3, this study decided the evolution model's parameters. To validate whether the model have the generalization capability, we advanced the data period approximately 3 years and divided the data into two parts by timing (shown in Fig. 5):

1. Training period: 3000 daily observations (nearly 11-years of data) were used as training data from 28/1/1992 to 23/1/2003.
2. Testing period: 1060 daily observations (nearly 4-years of data) were used as testing data from 24/12003 to 17/5/2007.

Using these new data sets, the model developed in Section 3 was re-implemented ten times and the results are shown in Table 4. The mean coefficients of profitability of the training period and the testing period are 1.24 and 0.96, respectively. After advancing 3-years of data, compared to the performance at the model design stage, the investment performance at model validation stage is significantly deteriorated. Fig. 6 shows the time of the buy–sell decision for the testing period and reveals that GNN can't escape the sharply downward period to avoid losses; as well, it can't catch the rising trend to gain profit.

Fig. 7 is the historical capital chart for the testing period and the starting point is set as 100. The darkened line and the fine line demonstrate the results of the GNN-based decision-making system and the buy-and-hold strategy, respectively. Apparently, the buy-and-hold strategy outperforms the GNN-based strategy. In the testing period, TAIEX rose 92% but the overall profit generated by GNN gained 47% only.

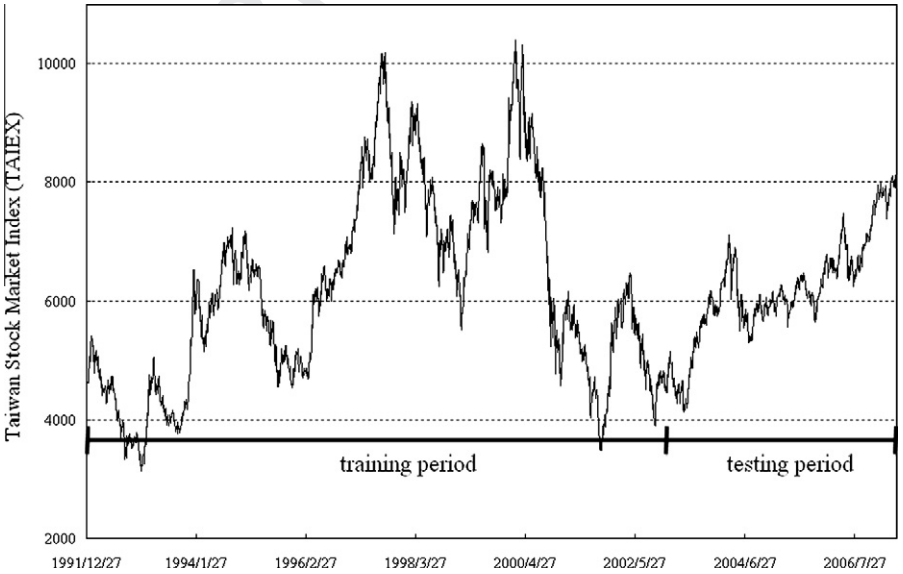

Fig. 5. TAIEX of training and testing period at the model validation stage.

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

Table 4
The investment performance of decision-making system at the model validation stage.

Serial number	Training period		Testing period	
	Percentage of annual profitability	Coefficient of profitability	Percentage of annual profitability	Coefficient of profitability
1	20.1	1.21	7.2	1.00
2	17.2	1.19	−2.4	0.92
3	19.3	1.20	0.2	0.94
4	17.9	1.19	5.1	0.99
5	53.0	1.55	−4.9	0.89
6	19.6	1.21	3.8	0.97
7	16.6	1.18	2.3	0.95
8	20.9	1.22	3.9	0.98
9	16.2	1.18	0.6	0.94
10	20.7	1.23	4.6	0.99
Mean	22.2	1.24	2.0	0.96
Standard deviation	11.0	0.10	3.7	0.03

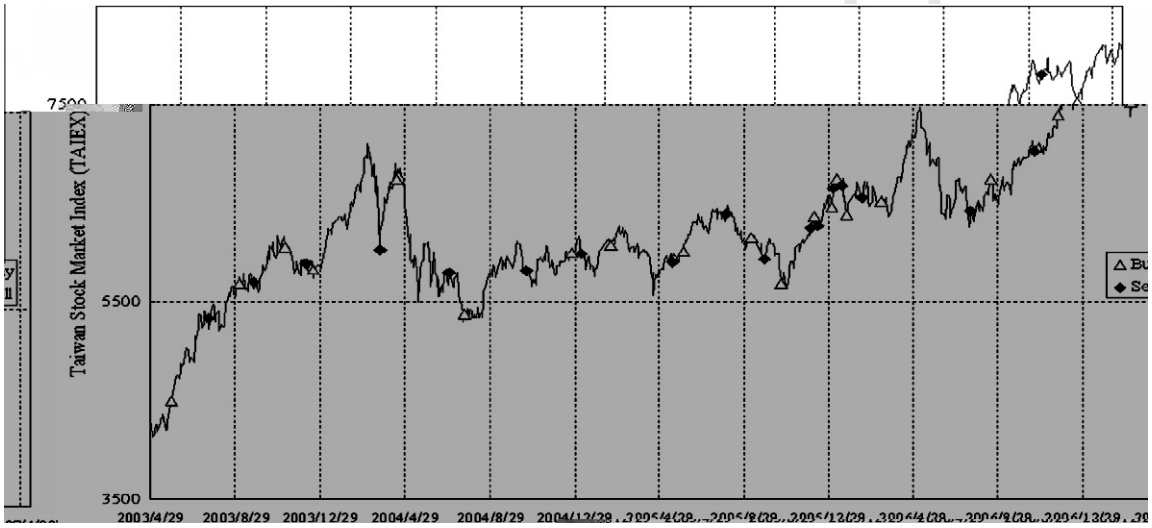

Fig. 6. The time of buy–sell decision during the test period at the model validation stage.

Fig. 7. Comparing the performance of TAIEX and GNN-based strategy during the testing period at the model validation stage.

To explore the influences of evolutionary computation parameters on the performance of system, we tried different range of network weights (−1 to +1 or −3 to +3) and evolution generations (30 or 100), and re-implemented the models ten times (shown in Table 5). Apparently, although the four mean coefficients of profitability in the training period are greater than one, but those in the testing

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

Table 5
Comparing the performance of model parameters at the model validation stage.

Serial number	Range of network weights: (−1, +1)				Range of network weights: (−3, +3)			
	30 generations		100 generations		30 generations		100 generations	
	Training period	Testing period	Training period	Testing period	Training period	Testing period	Training period	Testing period
1	1.15	0.96	1.18	0.96	1.14	0.95	1.17	0.93
2	1.11	0.91	1.16	0.94	1.14	0.93	1.18	0.90
3	1.16	0.95	1.17	0.93	1.16	0.98	1.19	0.96
4	1.15	0.93	1.18	0.94	1.16	0.93	1.17	0.92
5	1.14	0.95	1.16	0.91	1.17	0.89	1.19	0.90
6	1.13	0.95	1.15	0.94	1.15	0.99	1.17	0.95
7	1.15	0.95	1.16	0.95	1.16	0.96	1.18	0.92
8	1.12	0.95	1.19	0.95	1.14	0.93	1.17	0.89
9	1.14	0.93	1.16	0.95	1.13	0.91	1.17	0.93
10	1.14	0.93	1.17	0.93	1.14	0.93	1.18	0.91
Mean	1.14	0.94	1.17	0.94	1.15	0.94	1.18	0.92
Standard deviation	0.01	0.01	0.01	0.01	0.01	0.03	0.01	0.02

period are smaller than one, i.e., the buy-and-hold strategy outperforms the GNN-based strategy during the testing period.

5. Three approaches to improve the generalization of genetic neural networks

In section four, after advancing 3-years of data, compared to the performance at the model design stage, the investment performance at model validation stage is significantly deteriorated. The results showed that the model developed in Section 3 does not have generalization capability. To improve the model's generalization capability, this section tests three possible approaches.

5.1. Moving timeframe approach

The deterioration of the investment performance maybe results from the outdated data during the 12-year training period, which makes the buy/sell trading decision-making system developed by the evolution model does not fit the data of testing interval; or maybe the 5-year testing period is too long and the stock market structure was changed during the time, which also make the trading system developed by the model cannot be applied to the testing period.

Therefore, this study employs moving timeframe approach and uses 4800 daily observations from 7 June 1989 to 24 April 2007. The first 2160 daily observations (nearly 8-years of data) were used as the training data and the subsequent 240 daily observations (nearly 1-year of data) were used as the testing data. The above timeframe moves 11 times (=(4800 − 2160)/240 = 11).

Because of the randomness of evolution computation, the 11 timeframes are re-implemented three times for thirty generations individually and the results are shown in Table 6. Although the mean coefficients of profitability in the training period are greater than 1 in all eleven timeframes, those in the testing period are greater than 1 only in three timeframes. The overall mean coefficients of profitability in the training and testing period are 1.149 and 0.954, respectively. Apparently, the moving timeframe approach is unable to improve the generalization of evolution model.

5.2. Reducing evolution generation approach

To avoid the over-learning of GNN evolution model, it is necessary to suppress its learning capability. GNN's learning capability is positively correlated with the numbers of evolution generation, and therefore, this study tries to reduce the numbers of generation to one generation. The results are shown in Table 7. Although the mean coefficients of profitability in the training period are greater than 1 in all eleven timeframes, those in the testing period are greater than 1 only in one timeframe. The overall mean coefficients of profitability in the training and testing period are 1.092 and 0.943, respectively. Comparing the mean coefficients of profitability in the training period in Table 4 (1.24) and in Table 7 (1.092), this approach really suppressed the learning capability; however, it is still unable to avoid the over-learning of evolution model.

5.3. Reducing input variables approach

Generally, the more the complexity of the neural networks structure, the higher the possibility of over-learning. To reduce the complexity of the neural networks structure, our research tries to reduce the input variables to three main variables, including RSI(10), MVI(5, 20), and PVC(10). The major reason to select the

Table 6
The coefficients of profitability of moving timeframe approach.

Time-frame	Coefficient of profitability in the training period				Coefficient of profitability in the testing period			
	1	2	3	Mean	1	2	3	Mean
1	1.197	1.189	1.202	1.196	0.896	0.921	0.896	0.905
2	1.186	1.174	1.189	1.183	1.021	0.942	0.970	0.978
3	1.122	1.128	1.125	1.125	0.887	1.062	1.135	1.028
4	1.137	1.124	1.122	1.127	0.856	0.878	0.875	0.870
5	1.141	1.124	1.106	1.124	1.172	0.801	0.879	0.950
6	1.127	1.153	1.130	1.137	1.184	1.137	0.978	1.099
7	1.130	1.179	1.140	1.150	1.067	0.960	0.953	0.993
8	1.148	1.208	1.169	1.175	0.798	0.847	0.828	0.825
9	1.152	1.156	1.175	1.161	0.927	0.952	0.875	0.919
10	1.162	1.131	1.119	1.137	0.965	0.906	0.870	0.914
11	1.144	1.115	1.124	1.127	1.065	0.974	1.007	1.015
Mean	1.150	1.153	1.146	1.149	0.985	0.944	0.934	0.954

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

Table 7
The coefficient of profitability of reducing evolution generation approach.

Period	Mean profit coefficient in the training period				Mean profit coefficient in the testing period			
	1	2	3	Mean	1	2	3	Mean
1	1.133	1.110	1.164	1.136	1.001	0.954	0.987	0.980
2	1.090	1.102	1.133	1.108	1.013	0.988	0.992	0.997
3	1.064	1.039	1.056	1.053	0.936	1.043	0.861	0.947
4	1.073	1.066	1.090	1.076	0.906	0.916	0.964	0.929
5	1.090	1.053	1.086	1.077	1.312	0.846	0.776	0.978
6	1.079	1.147	1.121	1.116	1.089	0.968	0.978	1.011
7	1.094	1.109	1.087	1.096	0.934	0.888	0.995	0.939
8	1.123	1.105	1.122	1.117	0.761	0.782	0.843	0.795
9	1.056	1.078	1.124	1.086	0.901	0.949	0.896	0.915
10	1.080	1.045	1.119	1.082	0.954	0.905	0.958	0.938
11	1.059	1.088	1.058	1.068	0.922	0.889	0.998	0.936
Mean	1.085	1.086	1.105	1.092	0.976	0.920	0.932	0.943

Table 8
The coefficient of profitability of reducing input variables approach.

Period	Mean profit coefficient in the training period				Mean profit coefficient in the testing period			
	1	2	3	Mean	1	2	3	Mean
1	1.202	1.195	1.195	1.197	0.864	1.023	1.014	0.967
2	1.179	1.180	1.160	1.173	0.787	0.927	0.942	0.886
3	1.109	1.096	1.127	1.111	0.863	1.059	0.933	0.952
4	1.110	1.128	1.118	1.119	0.933	0.931	0.808	0.890
5	1.104	1.098	1.141	1.114	1.336	1.234	0.948	1.172
6	1.153	1.131	1.153	1.146	1.078	1.142	1.155	1.126
7	1.128	1.134	1.123	1.128	0.946	0.967	0.954	0.955
8	1.161	1.170	1.156	1.162	0.760	0.673	0.744	0.726
9	1.120	1.117	1.120	1.119	0.914	0.850	0.780	0.848
10	1.123	1.119	1.121	1.121	0.884	0.905	0.872	0.887
11	1.112	1.112	1.116	1.113	0.889	1.013	0.945	0.949
Mean	1.137	1.135	1.139	1.137	0.933	0.975	0.918	0.942

Table 9
The comparison of over-learning at the model and the system level.

Type	Driver	Passive	Cause	Phenomenon
Model level	Model builder	Evolution model	Model builders continuously use the "trial and error" approach to adjust the evolution model parameters	Good investment performance in the model design stage, but not in the model validation stage
System level	Evolution model	Decision- making system	Evolution model continuously employs evolution operations to automatically adjust the system parameters	Good investment performance in the training period, but not in the testing period

three variables is that they are the technical indicators of price, volume, and a combination of price and volume, which may sufficiently represent the indicators of technical analysis. The results are shown in Table 8. The overall mean coefficients of profitability in the training period and testing period are 1.137 and 0.942, respectively. Comparing the mean coefficients of profitability in the training period in Table 4 (1.24) and in Table 8 (1.137), the approach really suppressed the learning capability; however, it is still unable to avoid the over-learning of evolution model.

6. Conclusion

Our study investigates the problem of evolution model's generalization capability and conducts an empirical study on Taiwan's stock market. The empirical conclusions are as follows:

1. The decision-making system generated at the model design stage outperforms the system generated at the model validation stage, which shows over-learning at the model level.
2. Whether the decision-making system was generated either at the model design stage or at the model validation stage, the investment performance in the training period is much better than that in the testing period, which exhibits over-learning at the system level.
3. This research employs three approaches, including moving timeframe, reducing the evolution generation, and reducing the input variables, to improve the evolution model's generalization capability. But the results are not satisfactory. Apparently, it is not easy to use the current framework to overcome the problem of over-learning at the model level. Researchers have to try overcoming this problem through different approaches.

The major reason of the over-learning at the model level is that the model builders continuously use the "trial and error" approach to adjust the model parameters. This manipulation includes:

- evolution entity: e.g., the short/long term indicators system, and the number of input variables, the number of hidden neurons, and the range of connection weights of neural networks;
- evolution operations: e.g., population size, crossover rate, mutation rate, and the number of evolution generation;
- evolution environment: e.g., the length of training period and testing period, the moving timeframe approach;

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035

Those can make the model generate decision-making systems with good investment performance in both training and testing period at the model design stage, but not at the model validation stage.

The reasons accounting for the over-learning at the system level are that the evolution model continuously employs evolution operations to automatically adjust the decision-making system parameters (e.g., the weights and biases of neural networks and the thresholds of buying and selling), which can make the decision-making system to generate good investment performance in the training period only, but not in the testing period. Most of the previous researches emphasize merely the avoidance of over-learning at the system level and ignore the problem of over-learning at the model level, causing the performance of the generated decision-making system in the real application to be far below than anticipated.

Although this study cannot develop a GNN evolution model that can produce decision-making systems (neural networks) with good profitability, but our paper contributes to clarifying the issue of over-learning at the model and at the system level (see Table 9). For future research, this study developed a more reliable evaluation approach in examining the generalization capability of evolutionary computation based decision-making system.

References

Allen, F., & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of Financial Economics, 51*, 245–271.

Armano, G., Murru, A., & Roli, F. (2002). Stock market prediction by a mixture of genetic-neural experts. *International Journal of Pattern Recognition and Artificial Intelligence, 16*(5), 501–526.

Armano, G., Marchesi, M., & Murru, A. (2005). A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences, 170*(1), 3–33.

Booker, L. B., Goldberg, D. E., & Holland, J. H. (1989). Classifier systems and genetic algorithms. *Artificial Intelligence, 40*, 235–282.

Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications, 29*(4), 927–940.

Haykin, S. (2004). *Neural networks: A comprehensive foundation*. NJ: Prentice Hall PTR.

Hayward, S. (2004). Setting up performance surface on an artificial neural network with genetic algorithm optimization: In search of an accurate and profitable prediction for stock trading. In *Proceedings, 2004 congress on evolutionary computation* (pp. 948–954).

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan.

Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications, 19*(2), 125–132.

Kuo, R. J., Chen, C. H., & Hwang, Y. C. (2001). An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. *Fuzzy Sets and Systems, 118*, 21–45.

Kwon, Y. K., & Moon, B. R. (2003). Daily stock prediction using neuro-genetic hybrids. In *Genetic and evolutionary computation conference 2003* (pp. 2203–2214).

Lam, M. (2004). Neural network techniques for financial performance prediction: Integrating fundamental and technical analysis. *Decision Support Systems, 37*(4), 567–581.

Lee, R. S. T. (2004). IJADE stock advisor: An intelligent agent based stock prediction system using hybrid RBF recurrent network. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, 34*(3), 421–428.

Lien, L.-C., & Yeh, I.-C. (2008). Building trading system for Taiwan stock market using genetic neural networks. *Journal of Information Management, 1*(1), 29–52.

Massimiliano, V., Rushi, B., Oliver, H., & Mark, S. (2004). Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks. *Expert Systems with Applications, 27*(3), 417–425.

Olson, D., & Mossman, C. (2003). Neural network forecasts of Canadian stock returns using accounting ratios. *International Journal of Forecasting, 19*(3), 453–465.

Phua, H. P. K., Ming, D., & Lin, W. (2001). Neural network with genetically evolution algorithms for stocks prediction. *Asia-Pacific Journal of Operation Research, 18*(1), 103–108.

Please cite this article in press as: Yeh, I.-C., et al. Evaluation approach to stock trading system using evolutionary computation. *Expert Systems with Applications* (2010), doi:10.1016/j.eswa.2010.07.035