# Evolutionary Ensemble for Stock Prediction

Yung-Keun Kwon and Byung-Ro Moon

School of Computer Science & Engineering, Seoul National University
Shilim-dong, Kwanak-gu, Seoul, 151-742 Korea
{kwon, moon}@soar.snu.ac.kr

**Abstract.** We propose a genetic ensemble of recurrent neural networks for stock prediction model. The genetic algorithm tunes neural networks in a two-dimensional and parallel framework. The ensemble makes the decision of buying or selling more conservative. It showed notable improvement on the average over not only the buy-and-hold strategy but also other traditional ensemble approaches.

## 1 Introduction

Stock prediction is a historically hot topic. There were a variety of studies on this topic [10] [14] [18] [19] [24] [26] [28] [29]. Early studies were mostly about deterministic measures to help the prediction [1] [11] [16] [20]. Since about the early nineties many approaches based on stochastic or heuristic models have been proposed. They include artificial neural networks [26] [29], decision trees [4], rule induction [7], Bayesian belief networks [30], evolutionary algorithms [14] [18], classifier systems [25], fuzzy sets [3] [28], and association rules [27]. Hybrid models combining a few approaches are also popular [10] [24].

Kwon and Moon [19] proposed neuro-genetic hybrids for the stock prediction and showed notable success. However, since they did not consider the cost of trading, the performance can be overestimated when the trading occur too often. Although the trading cost is not very high these days, their results more or less took benefit from the zero trading cost. To overcome the problem, we need a longer term trading model.

In this paper, we propose a new neuro-genetic approach. It is an evolutionary ensemble of recurrent neural networks which is an extended model of the system in [19]. An ensemble learning is to aggregate multiple subsystems to solve a complex problem and expect stable performance, and since genetic algorithms produce many solutions it is natural to make an ensemble model in GAs. The basic evolutionary ensemble is one that chooses some best solutions and makes a decision by the opinion of the majority or average output of the ensemble. In this paper, we apply a different ensemble model. It does not use the same members of ensemble for each test data. It dynamically chooses the members that perform well for a set of training data with similar to each test data. In this model, the ensemble consists of the members performing best for the days with the most similar contexts to today's.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 1102–1113, 2004.

The rest of this paper is organized as follows. In Section 2, we explain the problem and present the objective. In Section 3, we describe our hybrid genetic algorithm for predicting the stock price. In Section 4, we provide our experimental results. Finally, conclusions are given in Section 5.

## 2 Objective

We have a database with years of daily trading data. Each record includes daily information which consists of the closing price, the highest price, the lowest price, and the trading volume. We name those at day $t$ as $x(t)$, $x_h(t)$, $x_l(t)$, and $v(t)$, respectively. If we expect $x(t+1)$ is considerably higher than $x(t)$, we buy the stocks; if lower, we sell them; otherwise, we do not take any action. The problem is a kind of time-series data prediction as follows:

$$x(t+1) = f(x(t), x(t-1), x(t-2), \ldots).$$

As in [19], we transform the original time series to another that is more suitable for neural networks. Instead of $x(t+1)$, $\frac{x(t+1)-x(t)}{x(t)}$ is used as the target variable as follows:

$$\frac{x(t+1)-x(t)}{x(t)} = f(g_1, g_2, \ldots, g_m),$$

where $g_k$'s $(k = 1, \ldots, m)$ are technical indicators or signals that were developed in previous studies.

We have four daily data, $x$, $x_h$, $x_l$, and $v$, but we do not use them for the input variables as they are. We utilize a number of technical indicators being used by financial experts such as moving average, golden-cross, dead-cross, relative strength index, and so on [15].

We describe some of them which were not considered in [19] in the following:

- Rate of change ( *ROC* )
  - A ratio of price difference between the current price and the price a period of time ago.
  - $ROC = \frac{x(t)-x(t-K)}{x(t-K)}$
- Money flow index ( *MFI* )
  - A momentum indicator that measures the strength of money flowing in and out of a security.
  - $MFI = MFI^+/MFI^-$
  - $MFI^+$, $MFI^-$ : the sum of MF of days when TP is greater or smaller than that of the previous day over a period of time, respectively.
  - $MF = TP \times v$
  - $TP = (x_h + x_l + x)/3$
- Ease of movement ( *EOM* )
  - An indicator that explains a relationship between price and volume.
  - $EOM = (MP(t) - MP(t-1))/BR$
  - $MP = (x_h + x_l)/2$
  - $BR = v/(x_h - x_l)$

$$
\begin{aligned}
X_1 &= \frac{MA(t) - MA(t-1)}{MA(t-1)} \\
X_2 &= \frac{MA_S(t) - MA_L(t)}{MA_L(t)} \\
X_3 &= \frac{x(t) - x(t-1)}{x(t-1)} \\
X_4 &= \text{the profit while the stock has risen or fallen continuously} \\
X_5 &= \text{\# of days for which the stock has risen or fallen continuously} \\
X_6 &= \frac{x(t) - x_l(t)}{x_h(t) - x_l(t)} \\
X_7 &= ROC(t) - ROC(t-1) \\
X_8 &= \frac{x(t) - MA(t)}{MA(t)} \\
X_9 &= \text{whether } MFI(t) \text{ crosses } MFI(t-1) \text{ or not} \\
X_{10} &= \text{whether } EOM(t) \text{ crosses zero value or not}
\end{aligned}
$$

**Fig. 1.** Some examples of input variables

In [19], 64 input variables were generated using the technical indicators. We add 11 input variables so totally generate 75 input variables. Figure 1 shows some representative variables including the added variables. In the figure, $MA$ means a numerical average value of the stock prices over a period of time. $MA_S$ and $MA_L$ are short-term and long-term moving average, respectively. After generating the new variables, we normalize them by dividing by the maximum value of each variable. It helps the neural network to learn efficiently.

There can be a number of measures to evaluate the performance of the trading system. In our problem, we use almost the same measure as the one in [19]. The only difference in them is that we consider the transaction cost in this work. Figure 2 shows the investing strategy and change of property at day $t+1$ according to the signal at day $t$ of the trading system. In the figure, $C_t$ and $S_t$ mean the cash and stock balances at day $t$ ($t = 1, \ldots, N$), respectively. We start with $C$, i.e, $C_1 = C$ and $S_1 = 0$. In the strategy, the constant $B$ is the upper bound of stock trade per day and $T$ is the transaction cost. The transaction cost was set to 0.3% in this work. We have the final property ratio $P$ as follows:

$$
P = \frac{C_N + S_N}{C_1 + S_1}.
$$

# 3 Evolutionary Ensemble

## 3.1 Artificial Neural Networks

We use a recurrent neural network architecture which is a variant of Elman's network [6]. It consists of input, hidden, and output layers as shown in Figure 3.

---

**if** ( signal is $SELL$ ) {
  $C_{t+1} \leftarrow C_t + \min(B, S_t) \times (1 - T)$
  $S_{t+1} \leftarrow S_t - \min(B, S_t)$
}
**if** ( signal is $BUY$ ) {
  $C_{t+1} \leftarrow C_t - \min(B, C_t)$
  $S_{t+1} \leftarrow S_t + \min(B, C_t)$
}
$S_{t+1} \leftarrow S_t \times \dfrac{x_{t+1}}{x_t}$

---

**Fig. 2.** Investing strategy and change of the property

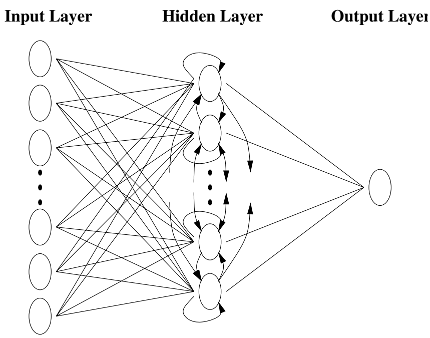



**Fig. 3.** The recurrent neural network architecture

Each hidden unit is connected to itself and also connected to all the other hidden units. The network is trained by a backpropagation-based algorithm.

It has 75 nodes in the input layer corresponding to the variables described in Section 2. Only one node exists in the output layer for $\frac{x(t+1)-x(t)}{x(t)}$.

### 3.2 Parallel Genetic Algorithm

We use a parallel GA to optimize the weights. It is a global single-population master-slave [2] and the structure is shown in Figure 4.

In this neuro-genetic hybrid approach, the fitness evaluation is dominant in running time. To evaluate an offspring (a network) the backpropagation-based algorithm trains the network with a set of training data. We distribute the load of evaluation to the clients (slaves) of a Linux cluster system. The main genetic parts locate in the server (master). When a new ANN is created by crossover and mutation, the GA passes it to one of the clients. When the evaluation is completed in the client, the result is sent back to the server. The server communicates with the clients in an asynchronous mode. This eliminates the need to synchronize every generation and it can maintain a high level of processor utilization, even if the slave processors operate at different speeds. This is possible because we use a steady-state GA which does not wait until a set of offspring

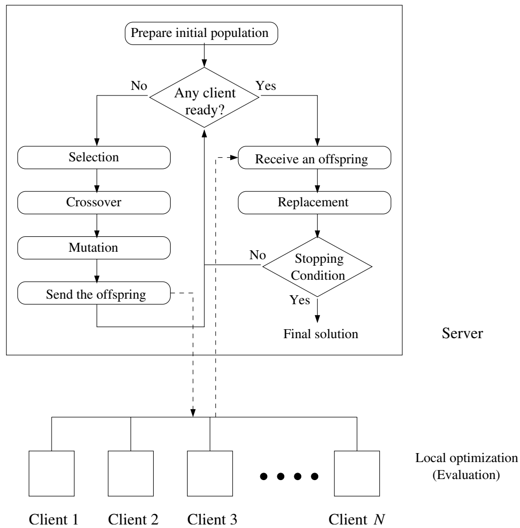
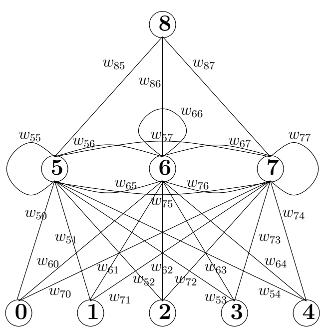



**Fig. 4.** The framework of the parallel genetic algorithm




A neural network

| | *input* | | | | | *hidden* | | | *output* |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 5 | $w_{50}$ | $w_{51}$ | $w_{52}$ | $w_{53}$ | $w_{54}$ | $w_{55}$ | $w_{56}$ | $w_{57}$ | $w_{85}$ |
| 6 | $w_{60}$ | $w_{61}$ | $w_{62}$ | $w_{63}$ | $w_{64}$ | $w_{65}$ | $w_{66}$ | $w_{67}$ | $w_{86}$ |
| 7 | $w_{70}$ | $w_{71}$ | $w_{72}$ | $w_{73}$ | $w_{74}$ | $w_{75}$ | $w_{76}$ | $w_{77}$ | $w_{87}$ |

Corresponding 2D chromosome

**Fig. 5.** Encoding in the GA

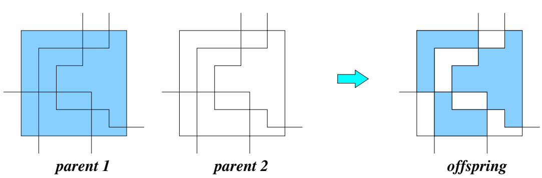



**Fig. 6.** An example of 2D geographical crossover

is generated. All these are achieved with the help of MPI (Message Passing Interface), a popular interface specification for programming distributed memory systems.

As shown in Figure 4, the process in the server is a parallel variant of traditional steady-state GA. In the following, we describe each part of the GA.

- *Representation:* We represent a chromosome by a two-dimensional weight matrix. In the matrix, each row corresponds to a hidden unit and each column corresponds to an input, hidden, or output unit. A chromosome is represented by a 2D matrix of $p \times (n+p+q)$ where $n$, $p$, and $q$ are the numbers of input, hidden, output units, respectively. In this work, the matrix size is $20 \times (75+20+1)$. We should note that most GAs for ANN optimization used linear encodings [9] [21]. We take the 2D encoding suggested in [17]. Figure 5 shows an example neural network and the corresponding chromosome.
- *Selection, crossover, and mutation:* Roulette-wheel selection is used for parent selection. The offspring is produced by geographic 2D crossover [13]. It is known to create diverse new schemata and reflect well the geographical relationships among genes. It chooses a number of lines, divides the chromosomal domain into two equivalent classes, and alternately copies the genes from the two parents as shown in Figure 6. The mutation operator replaces each weight in the matrix with a probability 0.1. All these three operators are performed in the server.
- *Local optimization:* After crossover and mutation, the offspring undergoes local optimization by backpropagation which helps the GA fine-tune around local optima. The result of local optimization provides the quality of the offspring. As mentioned, it is performed in the client and the result is sent back to the server.
- *Replacement and stopping criterion:* The offspring first attempts to replace the more similar parent to it. If it fails, it attempts to replace the other parent and the most inferior member of the population in order. Replacement is done only when the offspring is better than the replacee. The GA stops if it does not find an improved solution for a fixed number of generations.

### 3.3 Instance-Based Ensemble Model

An ensemble learning is to aggregate multiple subsystems to solve a complex problem. A number of approaches have been developed for ensemble learning [5] [8] [12] [23] [31]. The method is based on the fact that a solution with the smallest training error does not necessarily guarantee the most generalized one.

It is usual to select the best individual as the final solution in genetic algorithms. However, there is room for improving the performance with the help of other individuals in the population. Evolutionary ensemble approaches select a subset of the population as ensemble. It consists of some best individuals or representative ones from the whole population. In the latter, a clustering algorithm such as $k$-means algorithm [22] is used and a representative solution for each cluster is selected. In this paper, we devised an instance-based ensemble which is different from traditional ensembles. Traditional ensemble models do not consider the relationship or difference between data; the members of the ensemble are chosen with respect to a fixed set of instances. The basic idea of our instance-based ensemble is that it does not fix the members of ensemble but dynamically chooses the members that perform well for the days with similar contexts to today's. The instance-based ensembles are determined as follows:

– Obtain a set of NNs by the genetic algorithm described in Section 3.2.
– For each test day, select a set $K$ of instances among the training days that are the most similar to the test day in terms of Euclidean distance.
– Construct a subset of NNs, as ensemble, that predict relatively well on the days in $K$.

The trading decision depends on the majority decision in the ensemble. The final decision is one of the following three signals: *BUY*, *SELL*, and *KEEP*. The signal *KEEP* means no action. In the course, we extract three opinions from the ensemble, D1, D2, and D3. D1 is about the direction of the price at the next day. D2 and D3 are about the directions of the price at the day after tomorrow in the cases that tomorrow's price goes up or down, respectively. The ensemble gives the signal of *BUY* when both D1 and D2 are *"up"* and gives the signal of *SELL* when both D1 and D3 are *"down."* Otherwise, it gives the signal of *KEEP*. By this strategy, the trading becomes more conservative and too light an action can be avoided.

## 4 Experimental Results

We tested our approaches with the stocks of 36 companies in NYSE and NASDAQ. We evaluated the performance for 11 years from 1992 to 2002. We got the entire data from YAHOO (http://quote.yahoo.com). The GA was trained with two consecutive years of data and validated with the third year's. The solution was tested with the fourth year's data. This process was shifted year by year. Thus, totally 14 years of data were used for this work.

Table 1 shows the experimental results. The values mean the final property ratio $P$ defined in Section 2. *I-Ensemble* is the instance-based ensemble described in Section 3.3 and *Winner* is the version that uses the best solution. *A-Ensemble* is the version that selects a set of best NNs in the population and makes the decision from the average output of them, and *M-Ensemble* is the version that selects a set of best NNs in the same way as *A-Ensemble* and makes the decision by the the majority opinion of them. They are average results over 10 trials.

For quantitative comparison, we summarized the relative performance in Table 2. It represents the relative performance of each approach over the *buy-and-hold* strategy which buys the stock at the first day and holds it all through the year. Since there are 36 companies tested for 11 years, we have 394 cases except two cases with deficient data. In the table, *Up*, *Down*, and *NC* represent the situation of the stock market in each year. The *Up* and *Down* mean that the closing price has risen or fallen, respectively, over the year's starting price by 5% or more. *NC* means no notable difference. *Better* and *Worse* mean the number of cases where the $P$ value of the learned strategy was at least 5% higher or lower than that of the *buy-and-hold*, respectively. The *I-Ensemble* performed better

**Table 1.** $P$ values

| Symbols | Strategies | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AA | *I-Ensemble* | 1.206 | 1.093 | 1.091 | 1.211 | 1.239 | 1.181 | 1.081 | 1.577 | 0.797 | 1.190 | 0.674 |
| | *Winner* | 0.992 | 0.993 | 1.079 | 1.108 | 1.045 | 1.002 | 1.280 | 1.203 | 0.678 | 1.030 | 0.715 |
| | *A-Ensemble* | 1.141 | 1.115 | 1.282 | 1.272 | 1.305 | 1.214 | 1.279 | 1.359 | 0.797 | 1.155 | 0.754 |
| | *M-Ensemble* | 1.003 | 1.180 | 1.227 | 1.093 | 1.227 | 1.116 | 1.066 | 1.476 | 0.797 | 1.200 | 0.794 |
| AXP | *I-Ensemble* | 0.999 | 1.021 | 1.120 | 1.388 | 1.365 | 1.674 | 1.196 | 1.629 | 1.060 | 0.690 | 0.973 |
| | *Winner* | 0.995 | 1.050 | 0.983 | 1.330 | 1.245 | 1.583 | 1.147 | 1.355 | 1.084 | 0.690 | 0.807 |
| | *A-Ensemble* | 0.965 | 1.169 | 1.114 | 1.419 | 1.428 | 1.564 | 1.240 | 1.711 | 1.169 | 0.690 | 1.020 |
| | *M-Ensemble* | 1.026 | 1.030 | 1.114 | 1.456 | 1.237 | 1.599 | 1.176 | 1.558 | 1.034 | 0.690 | 0.883 |
| AYP | *I-Ensemble* | 1.074 | 1.156 | 0.825 | 1.325 | 1.041 | 1.065 | 1.060 | 0.765 | 1.330 | 0.850 | 0.192 |
| | *Winner* | 1.001 | 1.102 | 0.810 | 1.298 | 1.075 | 0.943 | 1.012 | 0.723 | 1.242 | 0.993 | 0.217 |
| | *A-Ensemble* | 1.070 | 1.113 | 0.825 | 1.384 | 1.039 | 0.991 | 1.062 | 0.740 | 1.282 | 1.050 | 0.157 |
| | *M-Ensemble* | 1.087 | 1.074 | 0.813 | 1.265 | 1.035 | 1.011 | 1.070 | 0.729 | 1.347 | 0.920 | 0.201 |
| BA | *I-Ensemble* | 0.783 | 1.102 | 1.114 | 1.668 | 1.300 | 0.950 | 0.713 | 1.260 | 1.018 | 0.614 | 0.864 |
| | *Winner* | 0.830 | 1.026 | 1.019 | 1.599 | 1.458 | 0.948 | 0.684 | 1.257 | 0.988 | 0.519 | 1.056 |
| | *A-Ensemble* | 0.860 | 1.012 | 0.983 | 1.709 | 1.300 | 0.973 | 0.720 | 1.329 | 1.097 | 0.615 | 0.856 |
| | *M-Ensemble* | 0.832 | 0.998 | 1.089 | 1.605 | 1.294 | 0.953 | 0.731 | 1.279 | 1.129 | 0.615 | 0.864 |
| C | *I-Ensemble* | 1.233 | 1.778 | 0.817 | 1.709 | 1.440 | 1.841 | 0.934 | 1.634 | 1.245 | 1.010 | 0.749 |
| | *Winner* | 0.824 | 1.672 | 0.817 | 1.211 | 1.387 | 1.698 | 0.960 | 1.861 | 1.367 | 0.979 | 0.775 |
| | *A-Ensemble* | 1.238 | 1.878 | 0.817 | 1.639 | 1.441 | 1.848 | 0.983 | 1.695 | 1.406 | 1.010 | 0.803 |
| | *M-Ensemble* | 1.210 | 1.833 | 0.817 | 1.435 | 1.441 | 1.773 | 0.980 | 1.636 | 1.317 | 1.010 | 0.748 |
| CAT | *I-Ensemble* | 1.330 | 1.327 | 1.244 | 1.181 | 1.487 | 1.135 | 0.967 | 1.228 | 0.883 | 1.009 | 0.928 |
| | *Winner* | 1.219 | 1.159 | 1.302 | 1.185 | 1.549 | 0.994 | 0.967 | 1.098 | 0.795 | 0.959 | 0.904 |
| | *A-Ensemble* | 1.060 | 1.644 | 1.231 | 1.280 | 1.245 | 1.351 | 0.967 | 1.499 | 0.829 | 1.115 | 0.962 |
| | *M-Ensemble* | 1.355 | 1.386 | 1.174 | 1.223 | 1.377 | 1.234 | 0.967 | 1.232 | 0.780 | 1.002 | 0.914 |
| DD | *I-Ensemble* | 1.037 | 1.031 | 1.053 | 1.270 | 1.330 | 1.349 | 1.097 | 1.269 | 0.900 | 0.966 | 1.369 |
| | *Winner* | 1.039 | 1.005 | 1.019 | 1.095 | 0.853 | 1.276 | 1.110 | 1.366 | 0.758 | 0.982 | 1.004 |
| | *A-Ensemble* | 1.038 | 0.989 | 0.996 | 1.233 | 1.075 | 1.577 | 1.205 | 1.331 | 1.005 | 0.996 | 1.334 |
| | *M-Ensemble* | 1.065 | 0.994 | 1.103 | 1.178 | 1.272 | 1.262 | 1.119 | 1.431 | 0.827 | 0.927 | 1.207 |
| DELL | *I-Ensemble* | 3.047 | 0.667 | 1.838 | 1.540 | 2.016 | 3.304 | 3.461 | 1.568 | 0.343 | 1.633 | 1.147 |
| | *Winner* | 3.440 | 0.917 | 1.992 | 1.717 | 2.799 | 1.882 | 3.461 | 1.331 | 0.341 | 1.259 | 0.871 |
| | *A-Ensemble* | 3.415 | 0.958 | 1.796 | 1.483 | 2.007 | 3.346 | 3.461 | 1.253 | 0.352 | 2.111 | 1.122 |
| | *M-Ensemble* | 2.235 | 0.513 | 1.713 | 1.660 | 2.013 | 3.346 | 3.461 | 1.455 | 0.344 | 1.868 | 1.424 |
| DIS | *I-Ensemble* | 1.286 | 1.024 | 1.113 | 1.106 | 1.164 | 1.462 | 0.877 | 1.002 | 1.024 | 0.828 | 1.125 |
| | *Winner* | 1.397 | 0.956 | 1.145 | 1.036 | 1.136 | 1.330 | 0.892 | 1.018 | 1.089 | 0.971 | 1.184 |
| | *A-Ensemble* | 1.345 | 1.009 | 1.024 | 0.931 | 1.107 | 1.149 | 0.843 | 1.050 | 1.073 | 0.908 | 1.301 |
| | *M-Ensemble* | 1.246 | 1.019 | 1.096 | 1.046 | 1.141 | 1.466 | 0.923 | 1.026 | 0.807 | 0.948 | 1.196 |
| EK | *I-Ensemble* | 0.891 | 1.173 | 1.078 | 1.548 | 1.193 | 0.842 | 1.140 | 0.978 | 0.655 | 0.777 | 1.232 |
| | *Winner* | 0.861 | 1.287 | 0.918 | 1.392 | 0.947 | 0.840 | 1.187 | 0.976 | 0.588 | 0.702 | 1.022 |
| | *A-Ensemble* | 0.889 | 1.187 | 1.076 | 1.227 | 1.149 | 0.810 | 1.096 | 1.023 | 0.694 | 0.761 | 1.097 |
| | *M-Ensemble* | 0.893 | 1.279 | 1.054 | 1.353 | 1.115 | 0.826 | 1.134 | 1.111 | 0.608 | 0.762 | 1.198 |
| GE | *I-Ensemble* | 1.119 | 1.179 | 1.011 | 1.312 | 1.376 | 1.448 | 1.350 | 1.600 | 0.954 | 0.934 | 1.152 |
| | *Winner* | 1.161 | 1.201 | 0.982 | 1.162 | 1.075 | 1.484 | 1.315 | 1.606 | 0.912 | 0.895 | 0.906 |
| | *A-Ensemble* | 1.116 | 1.141 | 1.006 | 1.234 | 1.328 | 1.429 | 1.359 | 1.316 | 0.982 | 0.874 | 0.989 |
| | *M-Ensemble* | 1.156 | 1.136 | 0.996 | 1.178 | 1.266 | 1.487 | 1.386 | 1.597 | 0.981 | 0.853 | 1.075 |
| GM | *I-Ensemble* | 1.065 | 1.244 | 0.755 | 1.283 | 0.957 | 1.309 | 1.150 | 1.304 | 0.840 | 0.963 | 0.993 |
| | *Winner* | 0.970 | 1.024 | 0.708 | 1.251 | 0.940 | 1.085 | 1.088 | 1.376 | 0.630 | 0.971 | 0.850 |
| | *A-Ensemble* | 0.969 | 1.030 | 0.754 | 1.251 | 0.995 | 1.526 | 1.284 | 1.419 | 0.809 | 1.324 | 1.040 |
| | *M-Ensemble* | 1.122 | 1.050 | 0.741 | 1.251 | 1.023 | 1.183 | 1.203 | 1.316 | 0.708 | 0.929 | 0.905 |
| HD | *I-Ensemble* | 1.437 | 0.787 | 1.167 | 0.874 | 1.036 | 1.211 | 1.425 | 1.661 | 0.697 | 1.110 | 0.485 |
| | *Winner* | 1.250 | 0.831 | 1.074 | 1.150 | 1.164 | 1.411 | 1.850 | 1.703 | 0.683 | 1.139 | 0.493 |
| | *A-Ensemble* | 1.437 | 0.825 | 1.214 | 1.049 | 1.037 | 1.210 | 1.640 | 1.549 | 0.615 | 1.106 | 0.516 |
| | *M-Ensemble* | 1.437 | 0.817 | 1.204 | 0.882 | 1.038 | 1.382 | 1.476 | 1.679 | 0.716 | 1.106 | 0.471 |
| HON | *I-Ensemble* | 1.347 | 1.303 | 0.886 | 1.287 | 1.378 | 1.238 | 1.103 | 1.138 | 0.817 | 0.799 | 0.672 |
| | *Winner* | 0.958 | 1.171 | 1.014 | 1.038 | 1.149 | 1.158 | 1.148 | 1.232 | 0.859 | 0.682 | 0.682 |
| | *A-Ensemble* | 1.289 | 1.289 | 0.990 | 1.189 | 1.365 | 1.312 | 1.103 | 1.164 | 0.878 | 0.809 | 0.642 |
| | *M-Ensemble* | 1.391 | 1.243 | 0.912 | 1.222 | 1.286 | 1.249 | 1.090 | 1.142 | 0.791 | 0.763 | 0.637 |
| HWP | *I-Ensemble* | 1.299 | 1.334 | 1.387 | 1.207 | 1.235 | 1.337 | 1.145 | 1.755 | 0.887 | 0.716 | 1.040 |
| | *Winner* | 1.004 | 1.203 | 1.305 | 1.276 | 1.334 | 1.353 | 1.215 | 1.293 | 0.724 | 0.716 | 0.807 |
| | *A-Ensemble* | 1.297 | 1.078 | 1.407 | 0.990 | 1.358 | 1.351 | 1.235 | 1.220 | 0.883 | 0.716 | 0.945 |
| | *M-Ensemble* | 1.577 | 1.133 | 1.389 | 1.193 | 1.320 | 1.390 | 1.088 | 2.013 | 0.806 | 0.716 | 0.933 |

**Table 1.** Continued

| Symbols | Strategies | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IBM | *I-Ensemble* | 0.566 | 1.010 | 0.994 | 1.150 | 1.738 | 1.378 | 1.644 | 1.276 | 0.734 | 1.459 | 0.841 |
| | *Winner* | 0.598 | 0.935 | 0.969 | 1.103 | 1.567 | 1.202 | 1.567 | 1.420 | 0.734 | 1.439 | 0.695 |
| | *A-Ensemble* | 0.622 | 1.150 | 0.988 | 1.063 | 1.759 | 1.378 | 1.668 | 1.341 | 0.734 | 1.543 | 0.836 |
| | *M-Ensemble* | 0.631 | 1.109 | 1.001 | 1.232 | 1.686 | 1.378 | 1.515 | 1.361 | 0.734 | 1.518 | 1.081 |
| INTC | *I-Ensemble* | 1.436 | 1.443 | 1.041 | 1.573 | 2.200 | 1.116 | 1.668 | 1.432 | 0.757 | 1.066 | 0.985 |
| | *Winner* | 1.068 | 1.358 | 1.078 | 1.564 | 2.015 | 0.985 | 1.623 | 1.084 | 0.678 | 1.008 | 0.975 |
| | *A-Ensemble* | 1.370 | 1.304 | 1.041 | 1.640 | 2.224 | 1.114 | 1.707 | 1.120 | 0.609 | 1.111 | 1.079 |
| | *M-Ensemble* | 1.142 | 1.275 | 1.041 | 1.353 | 2.228 | 1.114 | 1.729 | 1.425 | 0.710 | 1.113 | 1.000 |
| IP | *I-Ensemble* | 0.945 | 1.018 | 1.148 | 1.136 | 1.135 | 1.034 | 0.932 | 1.333 | 0.734 | 1.114 | 0.969 |
| | *Winner* | 0.910 | 0.967 | 1.130 | 1.164 | 1.222 | 0.938 | 0.880 | 1.024 | 0.831 | 1.057 | 0.882 |
| | *A-Ensemble* | 0.946 | 1.026 | 1.376 | 1.210 | 1.013 | 0.988 | 0.932 | 1.251 | 0.711 | 1.077 | 0.937 |
| | *M-Ensemble* | 0.946 | 1.049 | 1.223 | 1.163 | 1.087 | 1.013 | 0.932 | 1.124 | 0.718 | 1.145 | 1.053 |
| JNJ | *I-Ensemble* | 0.882 | 0.929 | 1.117 | 1.562 | 1.164 | 1.333 | 1.318 | 1.184 | 1.282 | 1.110 | 0.882 |
| | *Winner* | 0.888 | 0.933 | 0.977 | 1.556 | 1.016 | 1.388 | 1.431 | 1.436 | 1.288 | 1.043 | 0.912 |
| | *A-Ensemble* | 0.882 | 0.914 | 1.095 | 1.546 | 1.192 | 1.199 | 1.156 | 0.999 | 1.307 | 0.957 | 0.934 |
| | *M-Ensemble* | 0.882 | 0.950 | 1.164 | 1.599 | 1.227 | 1.311 | 1.299 | 1.110 | 1.226 | 1.217 | 0.927 |
| JPM | *I-Ensemble* | 0.944 | 0.976 | 0.814 | 1.442 | 1.173 | 1.207 | 0.939 | 1.173 | 1.220 | N/A | N/A |
| | *Winner* | 0.942 | 0.977 | 0.815 | 1.384 | 1.099 | 1.011 | 0.987 | 1.164 | 1.359 | | |
| | *A-Ensemble* | 0.959 | 0.987 | 0.814 | 1.163 | 0.959 | 1.184 | 1.012 | 1.133 | 1.118 | | |
| | *M-Ensemble* | 0.940 | 0.974 | 0.814 | 1.570 | 1.248 | 1.153 | 0.905 | 1.133 | 1.267 | | |
| KO | *I-Ensemble* | 1.054 | 1.060 | 1.064 | 1.457 | 1.401 | 1.297 | 0.948 | 0.836 | 1.035 | 0.880 | 0.943 |
| | *Winner* | 1.040 | 1.060 | 1.032 | 1.503 | 1.329 | 1.222 | 0.899 | 0.834 | 0.816 | 0.838 | 0.921 |
| | *A-Ensemble* | 1.047 | 1.060 | 1.042 | 1.449 | 1.381 | 1.279 | 0.940 | 0.762 | 1.056 | 0.781 | 0.974 |
| | *M-Ensemble* | 1.037 | 1.060 | 1.073 | 1.413 | 1.386 | 1.251 | 1.029 | 0.839 | 1.061 | 0.993 | 0.997 |
| MCD | *I-Ensemble* | 1.248 | 1.160 | 1.048 | 1.642 | 1.024 | 1.070 | 1.719 | 1.029 | 0.845 | 0.943 | 0.619 |
| | *Winner* | 1.166 | 1.031 | 1.132 | 1.465 | 1.113 | 1.044 | 0.994 | 1.042 | 0.845 | 0.566 | 0.644 |
| | *A-Ensemble* | 1.248 | 1.220 | 1.051 | 1.348 | 0.992 | 1.117 | 1.246 | 1.030 | 0.845 | 0.791 | 0.592 |
| | *M-Ensemble* | 1.292 | 1.244 | 1.059 | 1.498 | 1.049 | 1.092 | 1.662 | 1.034 | 0.845 | 0.946 | 0.579 |
| MMM | *I-Ensemble* | 1.056 | 1.157 | 1.053 | 1.308 | 1.227 | 1.021 | 0.894 | 1.176 | 1.125 | 0.938 | 1.049 |
| | *Winner* | 1.050 | 1.022 | 1.089 | 1.176 | 1.071 | 1.034 | 0.894 | 0.996 | 1.283 | 0.808 | 1.003 |
| | *A-Ensemble* | 1.102 | 1.093 | 1.011 | 1.254 | 1.179 | 0.936 | 0.894 | 1.133 | 1.062 | 0.982 | 1.050 |
| | *M-Ensemble* | 1.092 | 1.100 | 1.073 | 1.210 | 1.150 | 0.961 | 0.894 | 1.221 | 1.235 | 0.884 | 1.050 |
| MO | *I-Ensemble* | 0.959 | 0.755 | 1.030 | 1.448 | 1.214 | 1.337 | 1.157 | 0.564 | 1.281 | 0.969 | 0.802 |
| | *Winner* | 0.959 | 0.726 | 0.777 | 1.221 | 1.198 | 1.300 | 1.139 | 0.595 | 1.494 | 0.905 | 0.827 |
| | *A-Ensemble* | 0.959 | 0.755 | 1.000 | 1.277 | 1.214 | 1.299 | 1.136 | 0.629 | 1.382 | 0.974 | 0.796 |
| | *M-Ensemble* | 0.959 | 0.756 | 0.985 | 1.328 | 1.214 | 1.383 | 1.151 | 0.652 | 1.396 | 0.921 | 0.839 |
| MRK | *I-Ensemble* | 0.790 | 0.800 | 1.206 | 1.636 | 1.243 | 1.316 | 1.310 | 0.958 | 1.333 | 0.817 | 0.999 |
| | *Winner* | 0.856 | 0.803 | 1.063 | 1.124 | 1.251 | 1.163 | 1.417 | 0.875 | 1.374 | 0.928 | 0.970 |
| | *A-Ensemble* | 0.791 | 0.812 | 0.986 | 1.680 | 1.243 | 1.220 | 1.245 | 0.905 | 1.265 | 1.065 | 1.058 |
| | *M-Ensemble* | 0.787 | 0.803 | 1.150 | 1.489 | 1.267 | 1.264 | 1.495 | 0.892 | 1.346 | 0.813 | 0.987 |
| MSFT | *I-Ensemble* | 1.130 | 0.965 | 1.241 | 1.350 | 1.819 | 1.594 | 2.012 | 1.653 | 0.372 | 1.540 | 1.258 |
| | *Winner* | 1.150 | 1.027 | 1.268 | 1.243 | 1.914 | 1.377 | 1.926 | 1.837 | 0.381 | 1.431 | 0.912 |
| | *A-Ensemble* | 1.115 | 0.998 | 1.150 | 1.345 | 1.819 | 1.142 | 1.829 | 1.653 | 0.394 | 1.762 | 1.238 |
| | *M-Ensemble* | 1.120 | 0.974 | 1.213 | 1.186 | 1.817 | 1.536 | 1.972 | 1.690 | 0.372 | 1.510 | 1.135 |
| NMSB | *I-Ensemble* | 4.926 | 4.054 | 1.291 | 3.055 | 1.561 | 1.963 | 1.230 | 1.108 | 1.344 | 1.315 | 1.249 |
| | *Winner* | 6.671 | 4.190 | 4.315 | 4.395 | 3.051 | 2.876 | 1.861 | 1.407 | 2.403 | 1.398 | 1.210 |
| | *A-Ensemble* | 7.167 | 4.481 | 4.447 | 4.235 | 2.772 | 3.180 | 2.067 | 1.540 | 2.166 | 1.442 | 1.112 |
| | *M-Ensemble* | 6.822 | 4.199 | 3.875 | 4.745 | 2.932 | 2.909 | 1.927 | 1.470 | 2.255 | 1.390 | 1.270 |
| ORCL | *I-Ensemble* | 1.597 | 1.821 | 1.504 | 1.519 | 1.687 | 0.855 | 1.820 | 1.431 | 0.893 | 0.557 | 0.820 |
| | *Winner* | 1.520 | 1.546 | 1.579 | 1.151 | 1.686 | 0.949 | 1.490 | 1.386 | 1.021 | 0.611 | 0.904 |
| | *A-Ensemble* | 0.978 | 2.019 | 1.515 | 1.652 | 1.797 | 0.869 | 1.373 | 1.346 | 0.946 | 0.530 | 0.981 |
| | *M-Ensemble* | 1.317 | 1.887 | 1.537 | 1.513 | 1.700 | 0.989 | 1.870 | 1.723 | 0.895 | 0.530 | 0.633 |
| PG | *I-Ensemble* | 1.180 | 1.056 | 1.133 | 1.362 | 1.258 | 1.446 | 1.112 | 1.285 | 0.722 | 1.025 | 1.021 |
| | *Winner* | 1.066 | 0.908 | 1.131 | 1.215 | 1.182 | 1.315 | 1.146 | 1.177 | 0.958 | 0.837 | 1.014 |
| | *A-Ensemble* | 1.115 | 1.027 | 1.208 | 1.370 | 1.265 | 1.483 | 1.112 | 1.358 | 0.681 | 1.049 | 1.036 |
| | *M-Ensemble* | 1.111 | 1.048 | 1.129 | 1.294 | 1.082 | 1.496 | 1.112 | 1.301 | 0.672 | 1.019 | 1.002 |
| RYFL | *I-Ensemble* | 5.341 | 8.777 | 13.169 | 5.974 | 6.145 | 2.837 | 1.417 | 1.925 | 2.137 | 3.067 | 1.569 |
| | *Winner* | 6.780 | 11.990 | 15.244 | 5.312 | 7.467 | 4.640 | 1.824 | 2.945 | 3.854 | 5.264 | 2.055 |
| | *A-Ensemble* | 9.031 | 12.164 | 15.898 | 6.318 | 7.424 | 4.747 | 1.554 | 3.690 | 3.952 | 4.807 | 2.033 |
| | *M-Ensemble* | 9.035 | 12.399 | 13.660 | 6.269 | 6.834 | 4.348 | 1.550 | 2.679 | 3.678 | 5.111 | 1.789 |

**Table 1.** Continued

| Symbols | Strategies | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SBC | *I-Ensemble* | 1.115 | 1.217 | 0.979 | 1.280 | 0.901 | 1.432 | 1.603 | 1.056 | 1.077 | 0.956 | 0.729 |
| | *Winner* | 1.083 | 1.246 | 0.973 | 1.174 | 0.834 | 1.571 | 1.584 | 0.949 | 1.027 | 0.981 | 0.905 |
| | *A-Ensemble* | 1.184 | 1.301 | 0.979 | 1.167 | 0.901 | 1.432 | 1.404 | 1.151 | 0.926 | 0.792 | 0.783 |
| | *M-Ensemble* | 1.108 | 1.233 | 0.957 | 1.131 | 0.943 | 1.524 | 1.517 | 0.913 | 0.977 | 1.104 | 0.737 |
| SUNW | *I-Ensemble* | 1.176 | 0.913 | 1.193 | 1.445 | 1.241 | 1.595 | 1.754 | 3.269 | 0.672 | 0.514 | 0.918 |
| | *Winner* | 1.072 | 0.595 | 1.158 | 1.499 | 1.680 | 2.084 | 1.022 | 3.101 | 0.614 | 0.574 | 1.073 |
| | *A-Ensemble* | 1.350 | 0.995 | 1.146 | 1.108 | 1.366 | 1.722 | 1.858 | 3.398 | 0.671 | 0.505 | 1.083 |
| | *M-Ensemble* | 1.501 | 0.863 | 1.261 | 1.858 | 1.340 | 1.686 | 1.643 | 3.277 | 0.665 | 0.492 | 0.864 |
| T | *I-Ensemble* | 0.970 | 1.014 | 0.950 | 1.180 | 1.012 | 1.123 | 1.311 | 1.179 | 0.351 | 1.246 | 0.935 |
| | *Winner* | 0.972 | 1.044 | 0.950 | 0.968 | 0.847 | 0.949 | 1.190 | 0.825 | 0.440 | 0.786 | 1.003 |
| | *A-Ensemble* | 0.931 | 0.990 | 0.950 | 1.347 | 0.891 | 1.126 | 1.324 | 1.067 | 0.345 | 1.599 | 1.196 |
| | *M-Ensemble* | 0.937 | 0.932 | 0.950 | 1.102 | 0.965 | 1.287 | 1.296 | 1.073 | 0.342 | 1.238 | 1.001 |
| UTX | *I-Ensemble* | 1.034 | 1.139 | 1.079 | 1.474 | 1.346 | 1.120 | 1.627 | 1.173 | 1.204 | 0.812 | 0.995 |
| | *Winner* | 0.989 | 1.082 | 1.134 | 1.300 | 1.321 | 1.107 | 1.350 | 1.362 | 1.222 | 0.692 | 0.825 |
| | *A-Ensemble* | 0.920 | 1.007 | 1.048 | 1.225 | 1.400 | 1.110 | 1.461 | 1.157 | 1.204 | 0.744 | 0.847 |
| | *M-Ensemble* | 0.942 | 1.008 | 1.178 | 1.358 | 1.347 | 1.131 | 1.524 | 1.140 | 1.204 | 0.734 | 1.009 |
| WMT | *I-Ensemble* | 1.063 | 0.811 | 0.809 | 1.140 | 1.062 | 1.258 | 1.316 | 1.824 | 0.843 | 1.126 | 0.942 |
| | *Winner* | 1.072 | 0.801 | 0.723 | 1.092 | 0.947 | 1.147 | 1.180 | 1.755 | 0.875 | 0.998 | 0.963 |
| | *A-Ensemble* | 1.063 | 0.811 | 0.833 | 1.114 | 1.048 | 1.401 | 1.089 | 1.857 | 0.932 | 1.108 | 0.937 |
| | *M-Ensemble* | 1.063 | 0.811 | 0.782 | 1.159 | 1.098 | 1.220 | 1.148 | 1.834 | 0.806 | 1.143 | 0.961 |
| XOM | *I-Ensemble* | 1.118 | 1.119 | 0.951 | 1.114 | 1.244 | 1.303 | 1.207 | 1.110 | 1.454 | 0.939 | 0.922 |
| | *Winner* | 1.178 | 1.088 | 1.020 | 1.175 | 1.273 | 1.296 | 1.203 | 1.101 | 1.431 | 0.986 | 1.022 |
| | *A-Ensemble* | 1.130 | 1.108 | 1.024 | 1.039 | 1.220 | 1.380 | 1.294 | 1.075 | 1.460 | 1.063 | 0.963 |
| | *M-Ensemble* | 1.143 | 1.095 | 1.007 | 1.178 | 1.275 | 1.263 | 1.178 | 1.079 | 1.372 | 0.916 | 0.960 |

**Table 2.** Relative performance over *buy-and-hold* Strategy

(1) *I-Ensemble*

| | *Better* | *Worse* | *Even* | Total |
|---|---|---|---|---|
| *Up* | 45 | 60 | 130 | 235 |
| *Down* | 53 | 3 | 56 | 112 |
| *NC* | 14 | 3 | 30 | 47 |
| Total | **112** | **66** | **216** | 394 |

(2) *Winner*

| | *Better* | *Worse* | *Even* | Total |
|---|---|---|---|---|
| *Up* | 45 | 125 | 65 | 235 |
| *Down* | 57 | 16 | 39 | 112 |
| *NC* | 14 | 12 | 21 | 47 |
| Total | **116** | **153** | **125** | 394 |

(3) *A-Ensemble*

| | *Better* | *Worse* | *Even* | Total |
|---|---|---|---|---|
| *Up* | 56 | 87 | 92 | 235 |
| *Down* | 58 | 9 | 45 | 112 |
| *NC* | 15 | 3 | 29 | 47 |
| Total | **129** | **99** | **166** | 394 |

(4) *M-Ensemble*

| | *Better* | *Worse* | *Even* | Total |
|---|---|---|---|---|
| *Up* | 47 | 68 | 120 | 235 |
| *Down* | 42 | 7 | 63 | 112 |
| *NC* | 15 | 7 | 25 | 47 |
| Total | **104** | **82** | **208** | 394 |

than the *buy-and-hold* in 119 cases, worse in 60 cases, and comparable in 216 cases. *Winner* uses the best NN and it is an approach with no ensemble. We note that *Winner* is the same model as the one in [19] except that it was evaluated under consideration of the transaction cost in this work. It did not show good performance when considering the transaction cost, primarily due to too often trades. *I-Ensemble* showed a significant performance improvement over not only *Winner* but also the other ensemble models on average.

## 5 Conclusion

In this paper, we proposed a GA-based evolutionary ensemble of recurrent neural networks for the stock trading. It showed significantly better performance than the "buy-and-hold" strategy and traditional ensemble models with a variety of companies on the data for the recent 11 years. In addition to the ensemble, we tried to make a conservative system not to trade too often. We have satisfiable profits after considering the transaction cost.

In the experiments, the proposed GA predicted better in some companies than in others. It implies that this work can be useful in portfolio optimization. Future study will include finding the stock trading strategy combined with portfolio. In addition, we believe this approach is not just restricted to the stock market.

**Acknowledgment.** This work was supported by Brain Korea 21 Project. The ICT at Seoul National University provided research facilities for this study.

## References

1. W. Blau. *Momentum, Direction, and Divergence.* John Wiley & Sons, 1995.
2. E. Cantu-Paz. A survey of parallel genetic algorithms. *Calculateurs Parallels*, 10(2):141–171, 1998.
3. O. Castillo and P. Melin. Simulation and forecasting complex financial time series using neural networks and fuzzy logic. In *Proceedings of IEEE Conference on Systems, Man, and Cybernetics*, pages 2664–2669, 2001.
4. Y. M. Chae, S. H. Ho, K. W. Cho, D. H. Lee, and S. H. Ji. Data mining approach to policy analysis in a health insurance domain. *International Journal of Medical Informatics*, 62(2):103–111, 2001.
5. H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6:1289–1301, 1994.
6. J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.
7. J. A. Gentry, M. J. Shaw, A. C. Tessmer, and D. T. Whitford. Using inductive learning to predict bankruptcy. *Journal of Organizational Computing and Electronic Commerce*, 12(1):39–57, 2002.
8. L. K. Hansen and P. Salamon. Neural network ensembles. 12:993–1001, 1990.
9. S. A. Harp, T. Samad, and A. Guha. Towards the genetic synthesis of neural networks. In *International Conference on Genetic Algorithm*, pages 360–369, 1989.
10. P. G. Harrald and M. Kamstra. Evolving artificial neural networks to combine financial forecasts. *IEEE Transactions on Evolutionary Computation*, 1(1):40–52, 1997.
11. F. Hochheimer. *Computerized Trading Techniques.* Merrill Lynch Commodities, 1982.
12. R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.
13. A. B. Kahng and B. R. Moon. Toward more powerful recombinations. In *International Conference on Genetic Algorithms*, pages 96–103, 1995.
14. M. A. Kanoudan. Genetic programming prediction of stock prices. *Computational Economics*, 16:207–236, 2000.

15. P. J. Kaufman. *Trading Systems and Methods*. John Wiley & Sons, 1998.
16. J. F. Kepka. Trading with ARIMA forecasts. *Technical Analysis of Stocks & Commodities*, 3:142–144, 1985.
17. J. H. Kim and B. R. Moon. Neuron reordering for better neuro-genetic hybrids. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 407–414, 2002.
18. K. J. Kim. 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, 2000.
19. Y. K. Kwon and B. R. Moon. Daily stock prediction using neuro-genetic hybrids. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 2203–2214, 2003.
20. P. E. Lafferty. The end point moving average. *Technical Analysis of Stocks & Commodities*, 13:413–417, 1995.
21. C. T. Lin and C. P. Jou. Controlling chaos by ga-based reinforcement learning neural network. *IEEE Transactions on Neural Networks*, 10(4):846–869, 1999.
22. J. MacQueen. *Some methods for classification and analysis of multivariate observation*. Berkely, University of California Press, 1967.
23. N. J. Nilsson. *Learning Machines: Foundations of Trainable pattern-classifying systems*. New Yourk: McGraw Hill, 1965.
24. K. N. Pantazopoulos, L. H. Tsoukalas, N. G. Bourbakis, Brün, and E. N. Houstis. Financial prediction and trading strategies using neurofuzzy approaches. *IEEE Transactions on Systems, Man, and Cybernetics–Part:B*, 28(4):520–531, 1998.
25. S. Schulenburg and P. Ross. Explorations in LCS models of stock trading. In *Advances in Learning Classifier Systems*, pages 151–180, 2001.
26. P. Tiňo, C. Schittenkopf, and G. Dorffner. Financial volatility trading using recurrent neural networks. *IEEE Transactions on Neural Networks*, 12:865–874, 2001.
27. R. Veliev, A. Rubinov, and A. Stranieri. The use of an association rules matrix for economic modelling. In *International conference on neural information processing*, pages 836–841, 1999.
28. Y. F. Wang. Predicting stock price using fuzzy gray prediction system. *Expert Systems with Applications*, 22(1):33–38, 2002.
29. I. D. Wilson, S. D. Paris, J. A. Ware, and D. H. Jenkins. Residential property price time series forecasting with neural networks. *Knowledge-Based Systems*, 15(5):335–341, 2002.
30. R. K. Wolfe. Turning point identification and Bayesian forecasting of a volatile time series. *Computers and Industrial Engineering*, 15:378–386, 1988.
31. X. Yao and Y. Liu. Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics–Part:B*, 28:417–425, 1998.