# Daily Stock Prediction Using Neuro-genetic Hybrids

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 neuro-genetic daily stock prediction model. Traditional indicators of stock prediction are utilized to produce useful input features of neural networks. The genetic algorithm optimizes the neural networks under a 2D encoding and crossover. To reduce the time in processing mass data, a parallel genetic algorithm was used on a Linux cluster system. It showed notable improvement on the average over the buy-and-hold strategy. We also observed that some companies were more predictable than others.

## 1 Introduction

It is a topic of practical interest to predict the trends of financial objects such as stocks, currencies, and options. It is not an easy job even for financial experts because they are mostly nonlinear, uncertain, and nonstationary. There is no consensus among experts as to how well, if possible, financial time series are predictable and how to predict them.

Many approaches have been tried to predict a variety of financial time series such as portfolio optimization, bankruptcy prediction, financial forecasting, fraud detection, and scheduling. They include artificial neural networks [16] [19], decision trees [3], rule induction [5], Bayesian belief networks [20], evolutionary algorithms [9] [12], classifier systems [13], fuzzy sets [2] [18], and association rules [17]. Hybrid models combining a few approaches are also popular [15] [7].

Artificial neural networks (ANNs) have the ability to approximate well a large class of functions. ANNs have the potential to capture complex nonlinear relationships between a group of individual features and the variable being forecasted, which simple linear models are unable to capture. It is important in ANNs how to find the most valuable input features and how to use them. In [22], they used not only quantitative variables but also qualitative information to more accurately forecast stock prices. Textual information in articles published on the web was also used in [21]. In addition, the optimization of weights is another important issue in ANNs. The backpropagation algorithm is the most popular one for the supervised training of ANNs, but yet it is just a local gradient technique and there is room for further optimization.

In this paper, we try to predict the stock price using a hybrid genetic approach combined with a recurrent neural network. We describe a number of input variables that enable the network to forecast the next day price more accurately. The genetic algorithm operators provide diverse initial weights for the network.

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 Preliminaries

### 2.1 The Problem and Dataset

We attack the automatic stock trading problem. There are a number of versions such as intraday, daily, weekly, and monthly trading, depending on behavioral interval. In this work, we consider the daily trading.

In the problem, each record of dataset includes daily information which consists of the closing price, the highest price, the lowest price, and the trading volume. We name them at day $t$ as $x(t)$, $x_h(t)$, $x_l(t)$, and $v(t)$, respectively. The trading strategy is based on the series of $x(t)$; if we expect $x(t+1)$ is 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 that can be usually first tried with delay coordinates as follows:

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

But, it is a simple and weak model. We transform the original time series to another that is more suitable for neural networks. First, $\frac{x(t+1)-x(t)}{x(t)}$ is used instead of $x(t+1)$ as the target variable. For the input variables, we use technical indicators or signals that were developed in deterministic trading techniques. To achieve it, we construct the model as follows:

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

where $g_k$ $(k = 1, \ldots, m)$ is a technical indicator or signal.

### 2.2 The Objective

There can be a number of measures to evaluate the performance of the trading system. In our problem, we simulate the daily trading as close as possible to the actual situation and evaluate the profit. We have a cash balance $C_t$ and stock $S_t$ at day $t$ $(t = 1, \ldots, N)$. We start with $C$, i.e, $C_1 = C$ and $S_1 = 0$. Figure 1 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 strategy, the constant $B$ is the upper bound of stock trade per day. We have the final property ratio $P$ as follows:

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

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

**Fig. 1.** Investing strategy and change of the property

## 3 The Suggested System

### 3.1 Processing Data

As mentioned, 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 [10]. We describe some of them in the following:

- Moving average ( $MA$ )
  - The numerical average value of the stock prices over a period of time.
  - $MA_S$, $MA_L$ : short-term and long-term moving average, respectively.
- Golden-cross and dead-cross
  - States that $MA_S$ crosses $MA_L$ upward and downward, respectively.
- Moving average convergence and divergence ( $MACD$ )
  - A momentum indicator that shows the relationship between $MA_S$ and $MA_L$.
  - $MACD = MA_S - MA_L$
- Relative strength index ( $RSI$ )
  - An oscillator that indicates the internal strength of a single stock.
  - $RSI = 100 - \frac{100}{1 + U/D}$,
    $U, D$ : An average of upward and downward price changes, respectively.
- Stochastics
  - An indicator that compares where a stock price closed relative to its price range over a given time period.
  - $\%K = \frac{x(t) - L}{H - L} \times 100$,
    $H, L$ = The highest and the lowest price in a given time period.
  - $\%D$ = Moving average of $\%K$

We generate 64 input variables using the technical indicators. Figure 2 shows some representative variables. The others not shown in this figure include variables in the same forms as the above that use trading volumes in place of the prices. 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.

$$
\begin{aligned}
X_1 &= \frac{MA(t) - MA(t-1)}{MA(t-1)} \\
X_2 &= \frac{MA_S(t) - MA_L(t)}{MA_L(t)} \\
X_3 &= \text{\# of days since the last golden-cross} \\
X_4 &= \text{\# of days since the last dead-cross} \\
X_5 &= \frac{x(t) - x(t-1)}{x(t-1)} \\
X_6 &= \text{the profit while the stock has risen or fallen continuously} \\
X_7 &= \text{\# of days for which the stock has risen or fallen continuously} \\
X_8 &= MACD(t) \\
X_9 &= \%K(t) \\
X_{10} &= \%D(t) \\
X_{11} &= \frac{x(t) - x_l(t)}{x_h(t) - x_l(t)}
\end{aligned}
$$

**Fig. 2.** Some examples of input variables

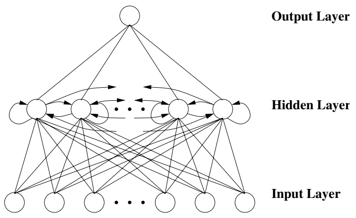


**Fig. 3.** The recurrent neural network architecture

## 3.2 Artificial Neural Networks

We use a recurrent neural network architecture which is a variant of Elman's network [4]. It consists of input, hidden, and output layers as shown in Fig. 3. 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 64 nodes in the input layer corresponding to the variables described in Section 3.1. Only one node exists in the output layer for $\frac{x(t+1)-x(t)}{x(t)}$.

## 3.3 Distribution of Loads by Parallelization

There have been many attempts to optimize the architectures or weights of ANNs. In this paper, we use a GA to optimize the weights. Especially, we parallelize the genetic algorithm since it takes much time to handle data over a long period of time. The structure of the parallel GA is shown in Fig. 4.

Parallel genetic algorithms can be largely categorized into three classes [1]: (i) global single-population master-slave GAs, (ii) single-population fine-grained,

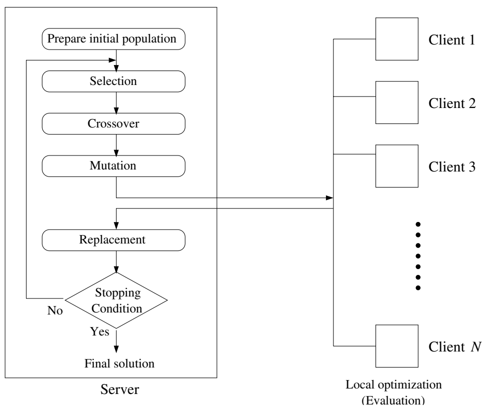
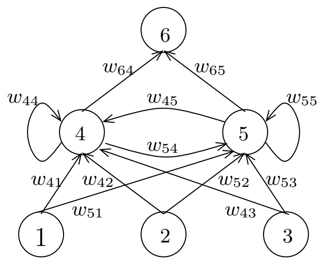


**Fig. 4.** The framework of the parallel genetic algorithm



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 4 | $w_{41}$ | $w_{42}$ | $w_{43}$ | $w_{44}$ | $w_{45}$ | $w_{64}$ |
| 5 | $w_{51}$ | $w_{52}$ | $w_{53}$ | $w_{54}$ | $w_{55}$ | $w_{65}$ |

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

and (iii) multiple-population coarse-grained GAs. We take the first model for this work. 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 training data.

In a measurement with *gprof*, the evaluation part took about 95% of the total running time. 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, it sends the result 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 is generated. All these are achieved with the help of MPI (Message Passing Interface), a popular interface specification for programming distributed memory systems. In this work, we used a Linux cluster system with 46 CPUs.

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

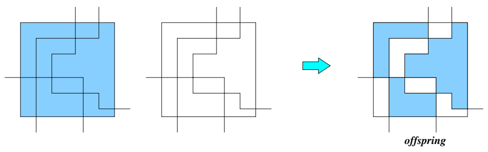


**Fig. 6.** An example of 2D geographical crossover

- *Representation:* Most GAs used linear encodings in optimizing ANN's weights [6] [14]. Recently, a two-dimensional encoding has proven to perform favorably [11]. We represent a chromosome by a two-dimensional weight matrix as shown in figure 5. 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 $p \times (n+p+q)$ where $n$, $p$, and $q$ are the numbers of input, hidden, output units, respectively. In this work, the matrix size is $20 \times (64 + 20 + 1)$.
- *Selection, crossover, and mutation:* Roulette-wheel selection is used for parent selection. The offspring is produced through geographic 2D crossover [8]. 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 low probability. All these three operators are performed in the server.
- *Local optimization:* After an offspring is modified by a mutation operator, it is locally optimized by backpropagation which helps the GA fine-tune around local optima. Local optimization is mingled with quality evaluation. As mentioned, it is performed in the client and the result is sent 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 generations.

## 4 Experimental Results

We tested our approaches with 36 companies' stocks in NYSE and NASDAQ. We evaluated the performance for ten years from 1992 to 2001. 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 13 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 2.2. In the table, the *Hold* strategy buys the stock at the first day and holds it all the year through. *RNN* and *GA* are the average results

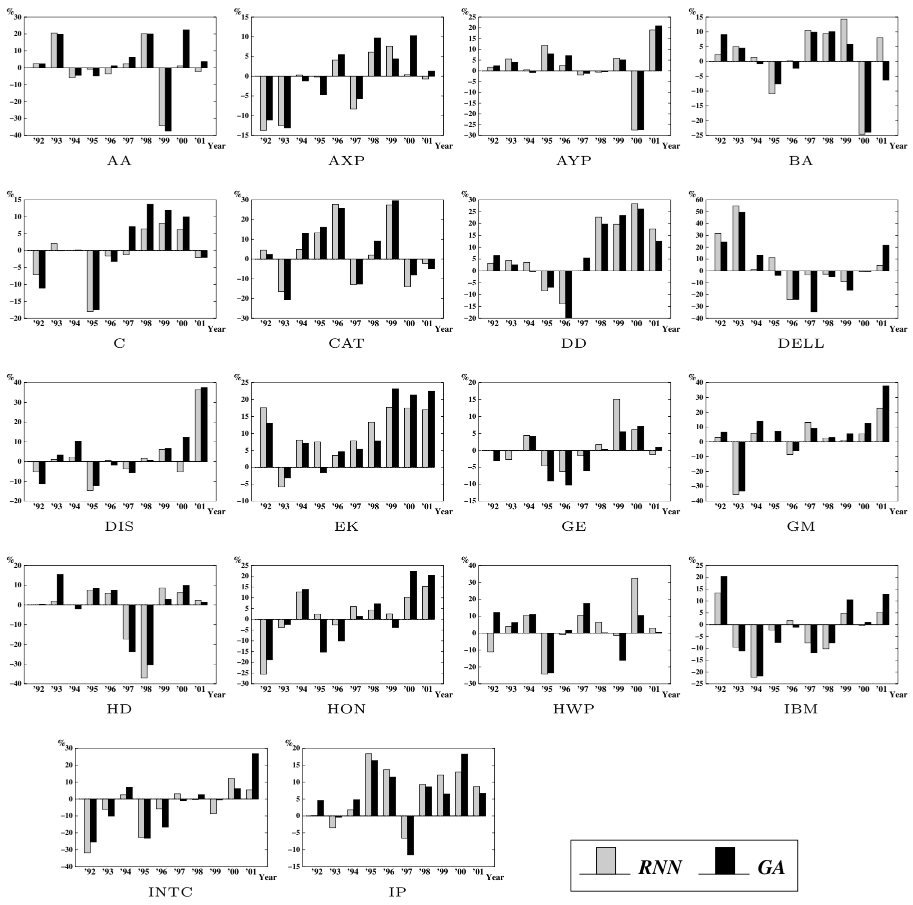


**Fig. 7.** Improvements of *RNN* and *GA* over *Hold* in the first half of the companies

by the recurrent neural network (20 trials) and the hybrid genetic algorithm (20 trials), respectively. For qualitative comparison, we summarized the relative performance in table 2. In the table, *Up*, *Down*, and *NC* represent the states of the stock market in each year. Since there are 36 companies tested for 10 years, we have 359 cases except one case with deficient data. The *Up* and *Down* mean that the closing price has risen or fallen over the year's starting price by 5% or more, respectively. *NC* means no notable difference. *Better*, *Worse*, and *Even* represent the relative performance of *RNN* and *GA* over the *Hold*. *Better* and *Worse* mean that *P* value of the learned strategy is at least 5% higher or lower than that of *Hold*, respectively. The GA performed better than the *Hold* in 153 cases, worse in 88 cases, and comparable in 118 cases. *RNN* and *GA* showed a significant performance improvement over *Hold* on the average. The *GA* considerably improved the performance of *RNN*.

**Table 1.** $P$ values

| Symbols | Strategies | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AA | *Hold* | 1.141 | 0.967 | 1.211 | 1.272 | 1.198 | 1.094 | 1.035 | 2.195 | 0.797 | 1.102 |
| | *RNN* | 1.168 | 1.166 | 1.140 | 1.262 | 1.154 | 1.120 | 1.243 | 1.446 | 0.806 | 1.078 |
| | *GA* | 1.167 | 1.159 | 1.157 | 1.211 | 1.212 | 1.161 | 1.243 | 1.373 | 0.975 | 1.143 |
| AXP | *Hold* | 1.205 | 1.185 | 1.116 | 1.419 | 1.316 | 1.624 | 1.144 | 1.549 | 0.992 | 0.686 |
| | *RNN* | 1.039 | 1.037 | 1.119 | 1.416 | 1.370 | 1.488 | 1.214 | 1.667 | 0.996 | 0.682 |
| | *GA* | 1.071 | 1.030 | 1.103 | 1.353 | 1.389 | 1.531 | 1.256 | 1.617 | 1.094 | 0.696 |
| AYP | *Hold* | 1.070 | 1.108 | 0.825 | 1.328 | 1.039 | 1.077 | 1.062 | 0.756 | 1.757 | 0.795 |
| | *RNN* | 1.089 | 1.168 | 0.829 | 1.484 | 1.065 | 1.057 | 1.055 | 0.800 | 1.274 | 0.946 |
| | *GA* | 1.096 | 1.152 | 0.818 | 1.433 | 1.113 | 1.065 | 1.058 | 0.795 | 1.276 | 0.960 |
| BA | *Hold* | 0.839 | 1.101 | 1.072 | 1.709 | 1.300 | 0.941 | 0.674 | 1.231 | 1.528 | 0.625 |
| | *RNN* | 0.858 | 1.156 | 1.087 | 1.523 | 1.303 | 1.040 | 0.737 | 1.408 | 1.152 | 0.676 |
| | *GA* | 0.915 | 1.151 | 1.063 | 1.579 | 1.270 | 1.034 | 0.742 | 1.304 | 1.163 | 0.586 |
| C | *Hold* | 1.238 | 1.656 | 0.817 | 1.878 | 1.441 | 1.803 | 0.940 | 1.576 | 1.273 | 1.000 |
| | *RNN* | 1.150 | 1.691 | 0.817 | 1.540 | 1.418 | 1.782 | 1.001 | 1.702 | 1.352 | 0.979 |
| | *GA* | 1.101 | 1.655 | 0.819 | 1.548 | 1.394 | 1.931 | 1.070 | 1.764 | 1.401 | 0.980 |
| CAT | *Hold* | 1.246 | 1.644 | 1.231 | 1.082 | 1.227 | 1.342 | 0.967 | 1.026 | 0.952 | 1.128 |
| | *RNN* | 1.303 | 1.374 | 1.291 | 1.225 | 1.567 | 1.169 | 0.986 | 1.308 | 0.819 | 1.103 |
| | *GA* | 1.275 | 1.304 | 1.390 | 1.256 | 1.542 | 1.173 | 1.054 | 1.331 | 0.875 | 1.071 |
| DD | *Hold* | 1.038 | 1.026 | 1.131 | 1.286 | 1.326 | 1.272 | 0.909 | 1.176 | 0.738 | 0.886 |
| | *RNN* | 1.071 | 1.071 | 1.172 | 1.178 | 1.142 | 1.273 | 1.116 | 1.408 | 0.948 | 1.043 |
| | *GA* | 1.105 | 1.052 | 1.127 | 1.196 | 1.063 | 1.342 | 1.089 | 1.452 | 0.931 | 0.997 |
| DELL | *Hold* | 2.514 | 0.530 | 1.661 | 1.787 | 2.877 | 3.346 | 3.461 | 1.372 | 0.344 | 1.553 |
| | *RNN* | 3.312 | 0.822 | 1.680 | 1.987 | 2.182 | 3.233 | 3.364 | 1.248 | 0.343 | 1.624 |
| | *GA* | 3.130 | 0.793 | 1.880 | 1.720 | 2.184 | 2.186 | 3.288 | 1.148 | 0.342 | 1.890 |
| DIS | *Hold* | 1.476 | 1.009 | 1.081 | 1.306 | 1.107 | 1.479 | 0.890 | 1.011 | 0.935 | 0.742 |
| | *RNN* | 1.399 | 1.020 | 1.106 | 1.115 | 1.112 | 1.424 | 0.906 | 1.072 | 0.887 | 1.012 |
| | *GA* | 1.308 | 1.043 | 1.191 | 1.148 | 1.087 | 1.397 | 0.897 | 1.077 | 1.050 | 1.019 |
| EK | *Hold* | 0.830 | 1.370 | 1.076 | 1.419 | 1.149 | 0.807 | 1.119 | 0.907 | 0.596 | 0.764 |
| | *RNN* | 0.976 | 1.290 | 1.163 | 1.526 | 1.190 | 0.869 | 1.267 | 1.067 | 0.701 | 0.894 |
| | *GA* | 0.938 | 1.326 | 1.153 | 1.396 | 1.202 | 0.850 | 1.206 | 1.117 | 0.724 | 0.936 |
| GE | *Hold* | 1.116 | 1.216 | 0.981 | 1.441 | 1.328 | 1.516 | 1.359 | 1.492 | 0.875 | 0.916 |
| | *RNN* | 1.114 | 1.184 | 1.024 | 1.375 | 1.244 | 1.492 | 1.382 | 1.718 | 0.928 | 0.905 |
| | *GA* | 1.082 | 1.215 | 1.021 | 1.309 | 1.192 | 1.423 | 1.361 | 1.575 | 0.937 | 0.924 |
| GM | *Hold* | 1.060 | 1.684 | 0.754 | 1.251 | 1.120 | 1.107 | 1.162 | 1.245 | 0.699 | 0.931 |
| | *RNN* | 1.091 | 1.086 | 0.797 | 1.251 | 1.024 | 1.252 | 1.192 | 1.260 | 0.737 | 1.143 |
| | *GA* | 1.131 | 1.124 | 0.858 | 1.340 | 1.052 | 1.206 | 1.196 | 1.314 | 0.786 | 1.284 |
| HD | *Hold* | 1.437 | 0.787 | 1.171 | 1.044 | 1.037 | 1.786 | 2.011 | 1.663 | 0.699 | 1.120 |
| | *RNN* | 1.440 | 0.802 | 1.171 | 1.123 | 1.098 | 1.478 | 1.266 | 1.805 | 0.742 | 1.145 |
| | *GA* | 1.442 | 0.909 | 1.148 | 1.132 | 1.114 | 1.363 | 1.401 | 1.711 | 0.768 | 1.135 |
| HON | *Hold* | 1.437 | 1.288 | 0.875 | 1.449 | 1.365 | 1.175 | 1.103 | 1.301 | 0.781 | 0.764 |
| | *RNN* | 1.071 | 1.239 | 0.986 | 1.483 | 1.330 | 1.244 | 1.151 | 1.333 | 0.860 | 0.881 |
| | *GA* | 1.167 | 1.257 | 0.996 | 1.227 | 1.228 | 1.192 | 1.182 | 1.251 | 0.956 | 0.921 |
| HWP | *Hold* | 1.219 | 1.141 | 1.265 | 1.698 | 1.207 | 1.282 | 1.068 | 1.704 | 0.641 | 0.679 |
| | *RNN* | 1.084 | 1.185 | 1.399 | 1.286 | 1.197 | 1.416 | 1.136 | 1.680 | 0.849 | 0.699 |
| | *GA* | 1.368 | 1.212 | 1.405 | 1.298 | 1.229 | 1.507 | 1.070 | 1.430 | 0.708 | 0.683 |
| IBM | *Hold* | 0.555 | 1.150 | 1.280 | 1.232 | 1.686 | 1.378 | 1.733 | 1.264 | 0.734 | 1.426 |
| | *RNN* | 0.630 | 1.040 | 0.996 | 1.204 | 1.715 | 1.272 | 1.555 | 1.324 | 0.732 | 1.502 |
| | *GA* | 0.669 | 1.022 | 1.003 | 1.139 | 1.668 | 1.216 | 1.599 | 1.397 | 0.741 | 1.611 |
| INTC | *Hold* | 1.721 | 1.416 | 1.041 | 1.839 | 2.224 | 1.114 | 1.664 | 1.440 | 0.714 | 1.012 |
| | *RNN* | 1.175 | 1.330 | 1.067 | 1.422 | 2.094 | 1.148 | 1.659 | 1.316 | 0.802 | 1.068 |
| | *GA* | 1.285 | 1.272 | 1.113 | 1.413 | 1.856 | 1.103 | 1.708 | 1.432 | 0.759 | 1.284 |
| IP | *Hold* | 0.946 | 1.026 | 1.105 | 1.042 | 1.067 | 1.082 | 0.932 | 1.318 | 0.711 | 1.023 |
| | *RNN* | 0.948 | 0.990 | 1.125 | 1.233 | 1.214 | 1.010 | 1.019 | 1.477 | 0.803 | 1.112 |
| | *GA* | 0.990 | 1.022 | 1.158 | 1.212 | 1.190 | 0.959 | 1.012 | 1.402 | 0.841 | 1.092 |

**Table 1.** Continued

| Symbols | Strategies | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| JNJ | *Hold* | 0.882 | 0.914 | 1.204 | 1.546 | 1.181 | 1.308 | 1.271 | 1.115 | 1.106 | 1.159 |
| | *RNN* | 0.881 | 0.921 | 1.211 | 1.554 | 1.184 | 1.313 | 1.411 | 1.150 | 1.314 | 1.162 |
| | *GA* | 0.913 | 0.912 | 1.185 | 1.490 | 1.074 | 1.434 | 1.426 | 1.275 | 1.376 | 1.179 |
| JPM | *Hold* | 0.959 | 1.065 | 0.814 | 1.436 | 1.194 | 1.156 | 0.957 | 1.133 | 1.363 | |
| | *RNN* | 1.003 | 1.051 | 0.822 | 1.369 | 1.091 | 1.234 | 0.961 | 1.223 | 1.404 | N/A |
| | *GA* | 0.976 | 1.032 | 0.830 | 1.381 | 1.242 | 1.234 | 0.960 | 1.101 | 1.345 | |
| KO | *Hold* | 1.047 | 1.060 | 1.163 | 1.449 | 1.383 | 1.290 | 1.004 | 0.839 | 1.079 | 0.775 |
| | *RNN* | 1.058 | 1.060 | 1.082 | 1.464 | 1.385 | 1.256 | 1.012 | 0.870 | 1.154 | 1.007 |
| | *GA* | 1.114 | 1.059 | 1.054 | 1.430 | 1.352 | 1.243 | 0.955 | 0.916 | 1.164 | 0.982 |
| MCD | *Hold* | 1.248 | 1.148 | 1.036 | 1.562 | 0.992 | 1.055 | 1.615 | 1.030 | 0.845 | 0.790 |
| | *RNN* | 1.268 | 1.256 | 1.197 | 1.579 | 1.087 | 1.050 | 1.483 | 1.114 | 0.845 | 0.934 |
| | *GA* | 1.254 | 1.245 | 1.227 | 1.584 | 1.092 | 1.044 | 1.358 | 1.179 | 0.855 | 0.966 |
| MMM | *Hold* | 1.056 | 1.062 | 1.011 | 1.255 | 1.321 | 0.968 | 0.894 | 1.265 | 1.263 | 0.992 |
| | *RNN* | 1.161 | 1.112 | 1.075 | 1.310 | 1.245 | 1.048 | 0.916 | 1.100 | 1.121 | 0.995 |
| | *GA* | 1.180 | 1.135 | 1.081 | 1.318 | 1.221 | 1.101 | 0.903 | 1.138 | 1.052 | 0.949 |
| MO | *Hold* | 0.959 | 0.755 | 1.000 | 1.594 | 1.214 | 1.219 | 1.160 | 0.446 | 1.971 | 0.993 |
| | *RNN* | 0.959 | 0.750 | 1.006 | 1.356 | 1.164 | 1.450 | 1.201 | 0.678 | 1.440 | 1.005 |
| | *GA* | 0.960 | 0.779 | 0.980 | 1.280 | 1.204 | 1.455 | 1.244 | 0.690 | 1.662 | 0.993 |
| MRK | *Hold* | 0.791 | 0.803 | 1.085 | 1.680 | 1.243 | 1.347 | 1.391 | 0.903 | 1.375 | 0.632 |
| | *RNN* | 0.790 | 0.803 | 1.239 | 1.465 | 1.310 | 1.263 | 1.430 | 0.981 | 1.360 | 0.975 |
| | *GA* | 0.817 | 0.868 | 1.196 | 1.260 | 1.258 | 1.270 | 1.433 | 0.988 | 1.377 | 0.950 |
| MSFT | *Hold* | 1.120 | 0.941 | 1.502 | 1.491 | 1.819 | 1.606 | 2.151 | 1.653 | 0.372 | 1.527 |
| | *RNN* | 1.153 | 0.941 | 1.343 | 1.418 | 1.694 | 1.603 | 1.894 | 1.705 | 0.382 | 1.564 |
| | *GA* | 1.270 | 1.031 | 1.343 | 1.293 | 1.898 | 1.438 | 1.920 | 1.793 | 0.407 | 1.435 |
| NMSB | *Hold* | 0.893 | 1.360 | 1.000 | 1.618 | 1.273 | 1.514 | 0.896 | 1.063 | 0.842 | 1.379 |
| | *RNN* | 7.049 | 4.855 | 4.603 | 4.896 | 3.012 | 3.280 | 2.174 | 1.530 | 2.485 | 1.571 |
| | *GA* | 6.773 | 5.081 | 4.843 | 4.777 | 3.021 | 3.437 | 2.126 | 1.623 | 2.440 | 1.524 |
| ORCL | *Hold* | 1.924 | 1.991 | 1.504 | 1.513 | 1.457 | 0.821 | 1.870 | 4.121 | 0.893 | 0.524 |
| | *RNN* | 1.445 | 1.840 | 1.647 | 1.517 | 1.667 | 1.024 | 1.416 | 1.576 | 0.916 | 0.528 |
| | *GA* | 1.377 | 1.558 | 1.613 | 1.491 | 1.554 | 0.974 | 1.540 | 1.633 | 0.977 | 0.735 |
| PG | *Hold* | 1.145 | 1.080 | 1.085 | 1.333 | 1.283 | 1.518 | 1.112 | 1.191 | 0.732 | 1.008 |
| | *RNN* | 1.136 | 1.069 | 1.115 | 1.399 | 1.224 | 1.455 | 1.111 | 1.398 | 1.115 | 1.151 |
| | *GA* | 1.137 | 1.058 | 1.091 | 1.354 | 1.237 | 1.417 | 1.113 | 1.395 | 1.010 | 1.188 |
| RYFL | *Hold* | 0.750 | 0.889 | 0.688 | 2.273 | 0.760 | 1.053 | 0.260 | 1.154 | 0.800 | 1.333 |
| | *RNN* | 8.658 | 12.406 | 16.073 | 7.267 | 7.670 | 5.034 | 1.770 | 3.627 | 3.983 | 5.753 |
| | *GA* | 9.037 | 12.718 | 15.765 | 6.346 | 8.001 | 5.191 | 1.827 | 2.950 | 3.967 | 5.655 |
| SBC | *Hold* | 1.121 | 1.137 | 0.979 | 1.437 | 0.901 | 1.424 | 1.404 | 0.897 | 1.068 | 0.778 |
| | *RNN* | 1.169 | 1.274 | 0.983 | 1.217 | 0.931 | 1.513 | 1.538 | 1.066 | 1.020 | 1.020 |
| | *GA* | 1.170 | 1.299 | 1.000 | 1.242 | 0.948 | 1.526 | 1.635 | 1.067 | 1.032 | 1.115 |
| SUNW | *Hold* | 1.165 | 0.851 | 1.245 | 2.512 | 1.196 | 1.551 | 2.170 | 3.398 | 0.665 | 0.484 |
| | *RNN* | 1.104 | 1.019 | 1.412 | 1.212 | 1.334 | 1.638 | 1.605 | 3.298 | 0.701 | 0.544 |
| | *GA* | 1.361 | 0.943 | 1.465 | 1.556 | 1.594 | 1.836 | 1.433 | 2.580 | 0.776 | 0.538 |
| T | *Hold* | 1.293 | 1.037 | 0.950 | 1.347 | 0.891 | 1.404 | 1.324 | 1.028 | 0.342 | 1.287 |
| | *RNN* | 1.031 | 0.987 | 0.950 | 1.188 | 1.102 | 1.336 | 1.317 | 1.207 | 0.395 | 1.655 |
| | *GA* | 1.095 | 0.981 | 0.950 | 1.169 | 1.102 | 1.219 | 1.231 | 1.123 | 0.444 | 1.417 |
| UTX | *Hold* | 0.920 | 1.228 | 1.048 | 1.491 | 1.400 | 1.114 | 1.477 | 1.157 | 1.204 | 0.859 |
| | *RNN* | 1.052 | 1.170 | 1.175 | 1.325 | 1.396 | 1.134 | 1.560 | 1.246 | 1.233 | 0.745 |
| | *GA* | 0.988 | 1.183 | 1.216 | 1.330 | 1.301 | 1.175 | 1.559 | 1.288 | 1.279 | 0.720 |
| WMT | *Hold* | 1.063 | 0.811 | 0.819 | 1.114 | 0.989 | 1.712 | 2.048 | 1.659 | 0.806 | 1.068 |
| | *RNN* | 1.060 | 0.813 | 0.771 | 1.035 | 1.038 | 1.262 | 1.166 | 1.700 | 0.943 | 1.076 |
| | *GA* | 1.107 | 0.857 | 0.791 | 1.158 | 1.097 | 1.265 | 1.217 | 1.771 | 0.990 | 1.111 |
| XOM | *Hold* | 1.023 | 1.039 | 0.951 | 1.330 | 1.220 | 1.258 | 1.174 | 1.077 | 1.140 | 0.882 |
| | *RNN* | 1.219 | 1.198 | 1.134 | 1.267 | 1.293 | 1.392 | 1.280 | 1.105 | 1.523 | 0.957 |
| | *GA* | 1.247 | 1.182 | 1.131 | 1.234 | 1.312 | 1.397 | 1.296 | 1.156 | 1.536 | 0.994 |

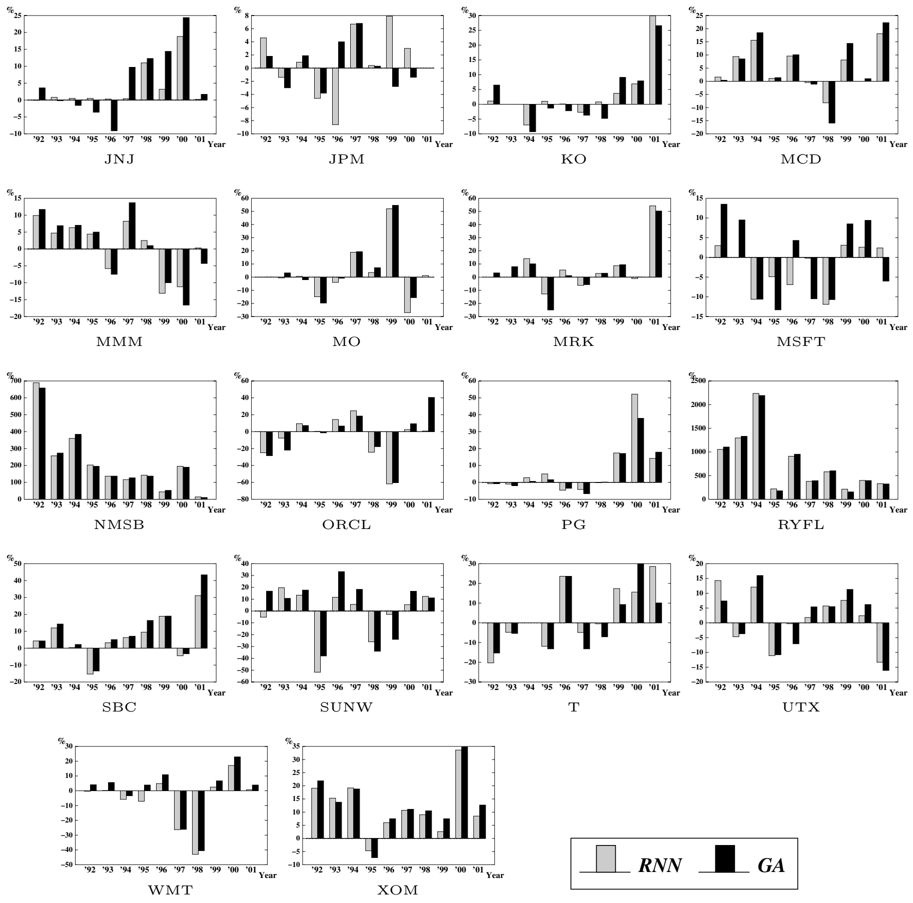


**Fig. 7.** Continued

**Table 2.** Relative performance of *RNN* and *GA* over *Hold*

(1) *RNN*

| | *Better* | *Worse* | *Even* | Total |
|---|---|---|---|---|
| *Up* | 63 | 64 | 103 | 230 |
| *Down* | 55 | 3 | 28 | 86 |
| *NC* | 21 | 1 | 21 | 43 |
| Total | **139** | **68** | **152** | 359 |

(2) *GA*

| | *Better* | *Worse* | *Even* | Total |
|---|---|---|---|---|
| *Up* | 79 | 76 | 75 | 230 |
| *Down* | 66 | 2 | 18 | 86 |
| *NC* | 28 | 2 | 13 | 43 |
| Total | **173** | **80** | **106** | 359 |

Figure 7 shows the relative performance graphically; the Y-axis represents the percentage improvement of *RNN* and *GA* over *Hold*. We can observe in the figure that some companies are more predictable than others. Practically, one can choose the stocks of companies that turned out to be more stably predictable by the GA.

## 5 Conclusion

In this paper, we proposed a genetic algorithm combined with a recurrent neural network for the daily stock trading. It showed significantly better performance than the "buy-and-hold" strategy with a variety of companies for recent ten years. For the input nodes of the neural network, we transform the stock prices and the volumes into a large number of technical indicators or signals. The genetic algorithm helps optimize the weights of neural network globally. This turned out to contribute to the performance improvement.

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, this approach is not just restricted to the stock market.

**Acknowledgment.** This work was partly supported by Optus Inc. and Brain Korea 21 Project. The RIACT at Seoul National University provided research facilities for this study.

## References

1. E. Cantu-Paz. A survey of parallel genetic algorithms. *Calculateurs Parallels*, 10(2):141–171, 1998.
2. 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.
3. 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.
4. J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.
5. J. A. Gentry, M. J. Shaw, A. C. Tessmer, and D. T. Whitford. Using inductive learning to predict bankrupcy. *Journal of Organizational Computing and Electronic Commerce*, 12(1):39–57, 2002.
6. 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.
7. P. G. Harrald and M. Kamstra. Evolving artificial neural networks to combine financial forecasts. *IEEE Transactions on Evolutionary Computation*, 1(1):40–52, 1997.
8. A. B. Kahng and B. R. Moon. Toward more powerful recombinations. In *International Conference on Genetic Algorithms*, pages 96–103, 1995.
9. M. A. Kanoudan. Genetic programming prediction of stock prices. *Computational Economics*, 16:207–236, 2000.
10. P. J. Kaufman. *Trading Systems and Methods*. John Wiley & Sons, 1998.
11. 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.
12. 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.

13. P. Y. Liao and J. S. Chen. Dynamic trading strategy learning model using learning classifier system. In *Proceedings of the Congresson Evolutionary Computation*, pages 783–789, 2001.
14. 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.
15. 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.
16. P. Tiňo, C. Schittenkopf, and G. Dorffner. Financial volatility trading using recurrent neural networks. *IEEE Transactions on Neural Networks*, 12:865–874, 2001.
17. 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.
18. Y. F. Wang. Predicting stock price using fuzzy grey prediction system. *Expert Systems with Applications*, 22(1):33–38, 2002.
19. 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.
20. R. K. Wolfe. Turning point identification and Bayesian forecasting of a volatile time series. *Computers and Industrial Engineering*, 15:378–386, 1988.
21. B Wuthrich, V. Cho, S. Leung, D. Permunetilleke, K. Sankaran, and J. Zhang. Daily stock market forecast from textual web data. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 2720–2725, 1999.
22. Y. Yoon and G. Swales. Predicting stock price performance: a neural network approach. In *Proc. 24th Annual Hawaii International conference on System Sciences*, pages 156–162, 1991.