Expert Systems with Applications 38 (2011) 11347–11354

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Intelligent stock trading system based on improved technical analysis and Echo State Network

Xiaowei Lin, Zehong Yang *, Yixu Song

State Key Laboratory of Intelligent Technology and System, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Keywords:
Stock trading system
Technical analysis
Genetic Algorithm
Echo State Network

ABSTRACT

Stock trading system to assist decision-making is an emerging research area and has great commercial potentials. Successful trading operations should occur near the reversal points of price trends. Traditional technical analysis, which usually appears as various trading rules, does aim to look for peaks and bottoms of trends and is widely used in stock market. Unfortunately, it is not convenient to directly apply technical analysis since it depends on person's experience to select appropriate rules for individual share. In this paper, we enhance conventional technical analysis with Genetic Algorithms by learning trading rules from history for individual stock and then combine different rules together with Echo State Network to provide trading suggestions. Numerous experiments on S&P 500 components demonstrate that whether in bull or bear market, our system significantly outperforms buy-and-hold strategy. Especially in bear market where S&P 500 index declines a lot, our system still profits.

1. Introduction

Data mining in stock market has been a hot topic for a long time due to its potential profits. Unfortunately, stock market is a complex and dynamic system with noisy, non-stationary and chaotic data series (Peters, 1994). Stock movement is affected by complicated factors, which can be divided into two groups: one is determinant, such as gradual power change between buying and selling side; the other is random factors, such as emergent affairs or daily operation variations (Bao & Yang, 2008). Therefore, data mining in stock market is very difficult and challenging. Recently, advances in artificial intelligence have led to a number of interesting new approaches to stock data mining, based on non-linear and non-stationary models. Among them, soft computing techniques, such as fuzzy logic, neural networks and probabilistic reasoning draw most attention because of their ability to handle uncertainty and noise in stock market (Vanstone & Tan, 2003, 2005). Applications range from time series prediction, classification to rule induction.

Although past studies have attained remarkable achievement in stock data mining, especially price prediction, they seldom directly guide trading. Future price forecast is not enough to suggest ideal trading operation to get profit as much as possible. An ideal trading operation should occur at the peak or bottom of price trend, that is, a good investor will sell stocks near the top of the trend and buy them close to the bottom. Thus, it is important to predict not only the future price but also when the price trend will hit the peak or bottom. In real market, technical analysis is widely used to assist decision-making. Its central idea is to look for peaks, bottoms, trends and indicators to estimate the possibility of current trend reversal and then make buy/sell decisions based on technical indicators which are some statistics derived from recent historical data (Bao & Yang, 2008). However, traditional technical analysis suffers from some shortcomings. First, it is difficult to directly apply technical analysis on individual stocks, especially for green hand. Technical analysis usually appears in a form as a trading rule. Take the popular "Golden Cross" and "Dead Cross" for example, if the sigh of (long-term moving average) – (short-term moving average) changes from positive to negative, it is called "Golden Cross" which indicates to buy stocks; if the sign of (long-term moving average) – (short-term moving average) changes from negative to positive, it is called "Dead Cross" which suggests to sell stocks. In the above description, it is hard to decide the time spans for both long-term and short-term moving average (MA) because each stock should have its own appropriate time spans. Investors usually choose those parameters according to their experience. Second, there are various technical analysis approaches, such as moving average approach, relative strength indicator (RSI) approach and stochastic indicator approach. Not all of them are effective for every stock. How to choose proper technical analysis methods for individual stock is also difficult for ordinary investors.

In this paper, we propose an intelligent stock trading system based on enhanced technical analysis and neural network. Genetic

* Corresponding author. Tel.: +86 10 62796828; fax: +86 10 62782266.
E-mail addresses: xiaow.lin@gmail.com (X. Lin), yangzehong@sina.com (Z. Yang), songyixu@sohu.com (Y. Song).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.03.001

Algorithm (GA) is utilized to improve traditional technical analysis by learning appropriate parameters for each trading rule. Then, the improved trading rules behave as experts together to give trading suggestions with a novel neural network-Echo State Network (ESN). The experiments demonstrate that whether in bull or bear market, our system will gain more income than buy-and-hold strategy. Particularly, it can still earn in bear market.

The rest of the paper is organized as follows: Section 2 describes the application of GA to improve traditional technical analysis; Section 3 introduces ESN and our system; Section 4 shows the experiments and results. Finally, we make a conclusion and suggest for further research.

2. Technical analysis enhancements

Technical analysis tends to forecast future price movements based on the study of past markets. It assumes that history will repeat itself and tries to identify archetypal patterns which have appeared in the past to predict what is likely to happen in the future. Although it has been recognized as one of the most reliable techniques for dealing stocks (Baba, Kawachi, Nomura, & Sakatani, 2004), it is not convenient to utilize technical analysis directly because it often appears as trading rules with parameters which have to be determined through experience. In this section, we improve traditional technical analysis with GA.

2.1. Genetic Algorithm

GAs are heuristic search techniques that are based on the theory of natural selection and evolution (Holland, 1992). They are particularly suitable for multi-parameter optimization problems in which an object function is subject to numerous hard and soft constraints (Kim, Min, & Han, 2006; Kim & Shin, 2007). In this paper, GA helps to enhance traditional technical analysis by generating a combination of parameters with which the corresponding trading rule will identify optimal trading points as close as possible to real reversal points of trends.

GA usually consists of four stages: initialization, selection, crossover and mutation. In the initialization stage, a population of genetic structures, called chromosomes that are randomly distributed in the solution space, is selected as the starting point of the search (Kim & Shin, 2007). Then, each chromosome, which represents a potential solution of the target problem, is evaluated by a user-defined fitness function. Through selection, the chromosomes with high performance will be preserved and propagate from generation to generation. The crossover forms a new offspring between two randomly selected "good parent" (Kim & Shin, 2007). And the mutation guarantees that it is possible to reach any point in the search space.

For real-world applications of optimization problems, choosing fitness function is the most critical step (Kim & Shin, 2007). In this paper, we design the fitness function to measure how close the suggested trading points are to those turning points of price trends. Suppose that there is an expected trading point sequence $T = \{T_1, T_2, \ldots, T_n\}$, in which buying and selling signals are staggered. For every expected trading point T_i, we search for operation signal S_j given by a specified technical analysis approach between its last and the next expected trading point ($T_{i-1} < S_j < T_{i+1}$).

(1) If T_i is an expected buying point, there are three cases:

(a) If S_j is a suggested buying point, the value of fitness function at T_i is computed as follows:

$$fitness(T_i) = close(S_j) - close(T_i) \tag{1}$$

in which $close(S_j)$ and $close(T_j)$ represent closing prices at S_j and T_i, respectively. The closer the closing price of S_j is to that of T_i, the smaller the value of fitness function is.

(b) If S_j is a suggested selling point and its price is close to T_i (satisfying $|close(S_j)\text{-}close(T_i)|/close(T_i) < 0.05$), then the value of the fitness function at T_i is

$$fitness(T_i) = 2 \times (\max(close(T_{i-1} : T_{i+1})) - close(T_i)) \tag{2}$$

in which $\max(close(T_{i-1}:T_i))$ means the maximum closing price between T_{i-1} and T_{i+1}. Since it is wrong to sell near the bottom of trend, such operation will be punished seriously.

(c) If no suggested operations are found between $T_{i-1} + 1$ and $T_{i+1} - 1$, it should also be punished due to missing opportunity. The fitness function is

$$fitness(T_i) = \max(close(T_{i-1} + 1 : T_{i+1} - 1) - close(T_i) \tag{3}$$

(2) Similarly, if T_i is an expected selling point, the fitness function is designed as follows:

(a) If S_j is a suggested selling signal, the fitness function at T_i is

$$fitness(T_i) = close(T_i) - close(S_j) \tag{4}$$

(b) If the price at S_j is close to that at T_i and S_j is misjudged as a buying point, the fitness function is

$$fitness(T_i) = 2 \times (close(T_i) - \min(close(T_{i-1} : T_{i+1})) \tag{5}$$

in which $\min(close(T_{i-1}:T_i))$ means the minimum closing price between T_{i-1} and T_{i+1}.

(c) If trading opportunity has been missed between $T_{i-1} + 1$ and $T_{i+1} - 1$, the fitness function is

$$fitness(T_i) = close(T_i) - \min(close(T_{i-1} + 1 : T_{i+1} - 1) \tag{6}$$

Finally, the fitness function of a suggested trading series $S = \{S_1, S_2, \ldots, S_m\}$ is defined as follows:

$$fitness(S) = \sum_{i=1}^{n} fitness(T_i) \tag{7}$$

GA is expected to find an optimal combination of parameters which will make the value of the fitness function minimum.

2.2. Moving average system

Simple MA is a popular technical indicator which calculates the mean price in a specified period. There are several methods to apply MA. Here, we improve two of them: one is the famous "Golden Cross" and "Dead Cross"; the other is MA envelope approach.

2.2.1. "Golden Cross" and "Dead Cross"

Baba et al. (2004) proposes a method to detect "Golden Cross" and "Dead Cross" with GA to reduce chance losses, which is also adopted in our system. Assume $Z_t = MA(N) - MA(n)$, in which $MA(N)$ means long-term MA while $MA(n)$ means short-term MA.

(a) If $z_t \geqslant 0$, find a number t_1 which is the closest to t ($t_1 < t$), and satisfies $z_{t_1 - 1} \leqslant 0$ and $z_{t_1} > 0$. Let $\overline{Mz_t} = \max(z_{t_1}, z_{t_1+1}, \ldots, z_t)$, if conditions (8) and (9) are met, it is time to buy

$$\overline{Mz_t} > b \times c \tag{8}$$

$$z_t < \min(\overline{Mz_t}/a, c) \tag{9}$$

Note that a, b, c are parameters.

(b) If $z_t < 0$, find the number k_1 which is closest to k ($k_1 < k$), and satisfies $z_{k_1-1} \geqslant 0$ and $z_{k_1} < 0$. Let $w_k = -z_k (k = k_1, \ldots, k)$, $\overline{Mw_k} = \max(w_{k_1}, w_{k_1+1}, \ldots, w_k)$, if conditions (10) and (11) are met, it is time to sell.

$$\overline{Mw_k} > \bar{b} \times \bar{c} \tag{10}$$
$$w_k < \min(\overline{Mw_k}/\bar{a}, \bar{c}) \tag{11}$$

Note that $\bar{a}$, $\bar{b}$, $\bar{c}$ are parameters.

Different from Baba's design in which the 26 week MA is fixed as long-term MA and the 13 week MA is taken as short-term MA, the parameters which need to be learned through GA contain not only a, b, c, $\bar{a}$, $\bar{b}$, $\bar{c}$, but also the time spans of long-term and short-term MA N and n in this paper. Fig. 1 shows the analysis results using the two methods, from which we find Baba's design proposes too many operation signals but many of them are wrong; by using GA to improve it, there is no wrong operation signals although some opportunities are missing.

2.2.2. Moving average envelope

MA envelope forms a channel or zone of commitment around a MA. If price breaks the upper band in downtrend, then it is time to buy; if it breaks through the lower band in uptrend, then it is time to sell (see Fig. 2). We learn the time span of MA and the upper and lower bandwidths (p_1 and p_2 respectively in Fig. 1) of MA envelope through GA. Note that in traditional technical analysis, $p_1 = p_2$. But in this paper, we drop this limitation and allow $p_1 \neq p_2$.

2.3. Relative strength index system

RSI is an extremely useful and popular momentum oscillator developed by Welles Wilder. It compares the magnitude of recent gains to recent losses in an attempt to determine overbought and oversold conditions of an asset. It is calculated as follows:

$$\mathrm{RSI} = 100 - \frac{100}{1 + \mathrm{RS}} \tag{12}$$
$$\mathrm{RS} = \frac{\text{Average of } x \text{ days up closes}}{\text{Average of } x \text{ days down closes}} \tag{13}$$

RSI ranges from 0 to 100. Generally, if the RSI rises above overbought level (usually 80), it indicates a selling signal; if it falls below oversold level (usually 20), it indicates a buying signal. However, in real market, the RSI is too sensitive and often breaks through overbought or oversold level ahead of time and then return after trend reversal happens. Hence, we introduce an oblique line to assist trading. When the RSI rises above overbought level, we draw an oblique line ab with slope p (see Fig. 3). If the RSI falls below ab, it is time to sell. Similarly, when the RSI falls below oversold level, we introduce another oblique line $a'b'$ with slope q.

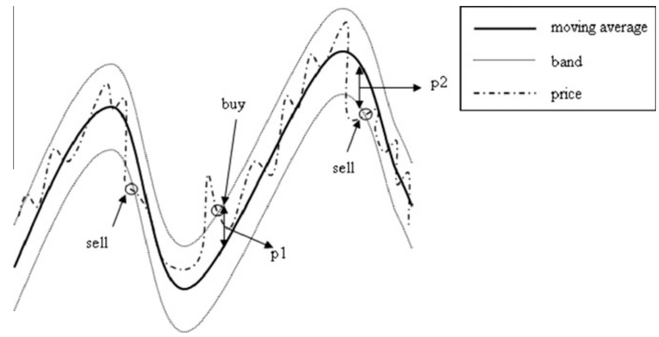

Fig. 2. Moving average envelope approach.

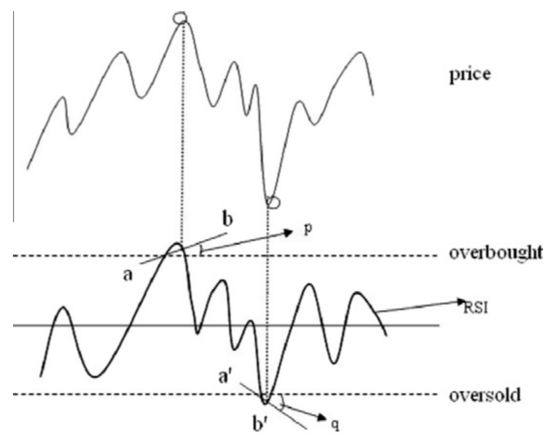

Fig. 3. RSI analysis approach.

If the RSI rises above $a'b'$, it is time to buy. GA helps to determine several parameters: the time span – x of RSI, the values of overbought and oversold bars and the slope p and q.

2.4. Rate of change system

Rate of change (ROC) is a common momentum indicator that measures the difference between current price and the past price to show how rapidly the price is moving. ROC can be obtained according to the following equation:

$$\mathrm{ROC} = \frac{close(i)}{close(i-n)} \times 100 \tag{14}$$

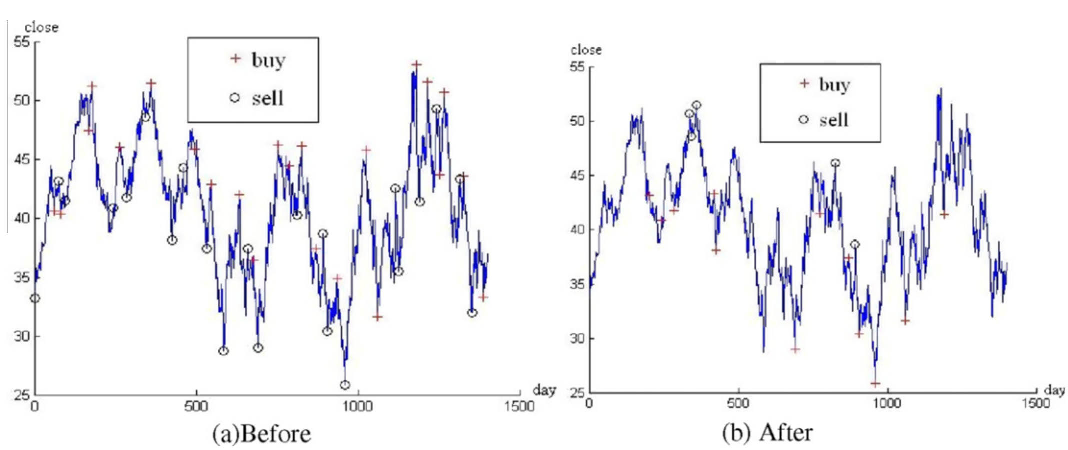

Fig. 1. Analysis result given by "Golden Cross" and "Dead Cross" approach.

where $close(i)$ represents current closing price and $close(i-n)$ means closing price of n days before.

ROC can be plotted using different time spans. The divergence of different ROCs can indicate possible reversal of price trend. Generally, when long-term ROC reaches a new high while short-term ROC locates near the equilibrium line (usually with the value of 100), the price will possibly fall down (see point A in Fig. 4); similarly, when long-term ROC reaches a new low while short-term ROC is near the equilibrium line, the price may ascend (see point B in Fig. 4). GA helps to determine the following parameters: the time spans of both long-term and short-term ROC, the upper bound exceeding which means a new high, the lower border falling down which means a new low, and the upper and lower bandwidths of the equilibrium line which measures whether the ROC is close to that line.

2.5. Stochastic system

Stochastic oscillator developed by George C. Lane is a momentum indicator that can warn of strength or weakness in the market. In the up-trend, it tries to measure when the closing price would get close to the lowest price in the given period; in the down-trend, it means when the closing price would get close to the highest price in the given period. The original stochastic oscillator is plotted as two lines called %K and %D which is calculated according to the following equations:

$$\%K = 100 \times (H_3/L_3) \tag{15}$$

where H_3 represents the sum of three $(C - LL)$ and L_3 represents that of three $(HH - LL)$, in which C means current close; LL means the lowest low price in a specified period; and HH means the highest high price in a specified period.

$$\%D = 100 \times (H_3'/L_3') \tag{16}$$

where H_3' represents the sum of three H_3 and L_3' represents that of three L_3.

Generally, %K line is more sensitive than %D. Therefore, the crossover of %K and %D lines may indicate meaningful reversal. Take the left-hand crossover for example (see Fig. 5), in our system, when %K rises above %D and satisfies $\%K < a$ and $\%K - \%D < b$, it is time to buy. Conversely, when %K falls below %D and satisfies $\%K > c$ and $\%D - \%K < d$, it is time to sell. All the parameters a, b, c, d should be learned through GA.

2.6. Candle chart system

Candle chart, which comes from Japan, provides an easy-to-decipher picture of current price action. Usually a candlestick is composed of a body, an upper and a lower wick. The wick illustrates the highest and lowest price of an individual share in a certain day. The body illustrates the opening and closing price of that

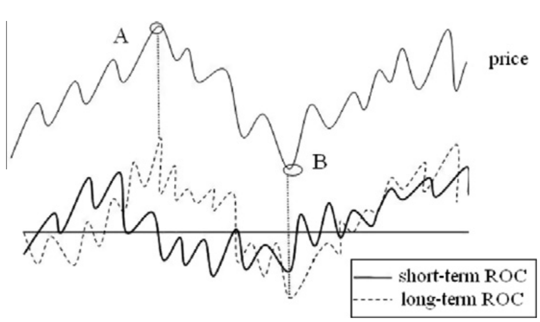

Fig. 4. ROC divergence analysis.

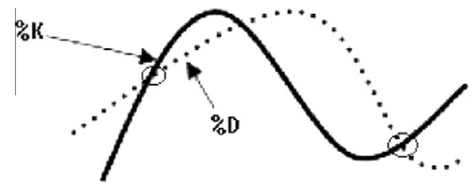

Fig. 5. Left hand crossover of stochastic oscillators.

day – if the closing price is higher than the opening, the body is white with the opening price at the bottom of the body and the closing price at the top; if the closing price is lower than the opening, the body is black with the opening price at the top and the closing at the bottom. Market turns are suggested by identifying candle patterns. The following patterns are used in our system: hammers/hanging man, dark cloud cover, piercing line and engulfing pattern (see Fig. 6). We use some parameters to quantitatively describe those patterns.

2.6.1. Hammer and hanging man

If $\frac{\text{the length of wicks or tails}}{\text{the length of body}} > a$, the candlestick is called hammer or hanging man, which indicates price reversal in the future.

2.6.2. Dark cloud cover

The pattern of dark cloud cover is composed of two candlesticks in which a black one follows a white one and $\frac{|\text{next day's close} - \text{previous day's open}|}{\text{the length of previous candlestick's body}} < b$. It is an indication of a future bearish trend.

2.6.3. Piercing line

Different from dark cloud cover, the pattern of piercing line is composed of two candlesticks in which a white one follows a black one and $\frac{|\text{previous day's open} - \text{next day's close}|}{\text{the length of previous candlestick's body}} < c$. It is an indication of a future bullish trend.

2.6.4. Engulfing pattern

It is characterized by a candlestick with a small body followed by another candlestick whose body completely engulfs the previous one. If an engulfing pattern whose second candlestick is a white one appears at the end of a downtrend and in the meantime satisfies |next day's close − previous day's open| > d and |previous day's close − next day's open| > e, it suggests a bullish trend in the future. Conversely, if an engulfing pattern whose second candlestick is a black one appears at the end of an uptrend and satisfies |next day's open − previous day's close| > f and |next day's close − previous day's open| > g, it suggests a bearish trend in the future.

3. Stock trading system

Our stock trading system is based on a novel recurrent neural network (RNN)-Echo State Network (ESN). Each technical analysis

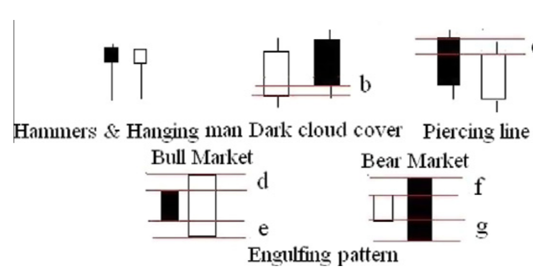

Fig. 6. Candle patterns.

system behaves as an expert and gave their forecast about whether critical trend turning will happen according to current closing price. If it is time to buy, the system will give a buying signal with value of −1; if it is time to sell, it will give a selling signal with value of 1; or it will give a signal with value of 0 which suggests no operation. Then all the signals offered by each system are fed into an ESN to evaluate whether it is appropriate to buy or sell the specified stock at current closing price. Investors can make their trading decisions by referring to the system's output which is a score showing how close current price is to future's ideal trading point.

3.1. Echo State Network

ESN is a novel RNN recently proposed by Jaeger and Haas (2004). Its basic idea is to use a large "reservoir" RNN as a supplier of interesting dynamics from which the desired output is combined (Jaeger, 2002). Compared with other conventional neural networks, the training of ESN is very simple and it does not need to worry about local convergence that traditional neural networks often confront with. ESN can be taken to all basic tasks of signal processing and control, including time series prediction, inverse modeling, pattern generation, event detection and classification, modeling distributions of stochastic processes, filtering and nonlinear control (Jaeger & Haas, 2004). It has been applied in wireless communications (Jaeger & Haas, 2004), robot control (Ishii, van der Zant, Becanovic, & Ploger, 2004) and speech recognition (Jaeger, Lukosevicius, & Popovici, 2007; Skowronski & Harris, 2007) and achieved good results.

A standard ESN is composed of input, hidden (also called "reservoir") and output layers (see Fig. 7). Note that connections directly from the input to the output layer and connections between output neurons are also allowed. Assume that $u(n+1)$ is an input vector at time step $(n+1)$, the activation of internal state $x(n+1)$ is updated according to

$$x(n+1) = f(W^{in}u(n+1) + Wx(n) + W^{back}y(n)) \tag{17}$$

where $f = (f_1, \ldots, f_n)$ are the internal unit's activation functions (typically sigmoid functions) (Jaeger, 2001); W^{in}, W and W^{back} are input–hidden, hidden–hidden and output–hidden connections' matrices, respectively and $y(n)$ is the output at time step n. The network output is obtained through the following equation:

$$y(n+1) = f^{out}(W^{out}(u(n+1), x(n+1), y(n))) \tag{18}$$

where $f^{out} = (f_1^{out}, \ldots f_L^{out})$ are the output unit's output functions; W^{out} represents output connections; and $(u(n+1), x(n+1), y(n))$ is the concatenation of the input, internal, and previous output activation vectors. ESN differs from previous recurrent neural network in that only W^{out} should be modified during learning process and any regression method is available.

In this paper, a standard ESN with 100 neurons in the reservoir is adopted. For each input channel, the input connection weights are randomly chosen to be −1, 1 and 0 with probabilities 20%, 20% and 60%.

The feedback connection weights are randomly sampled from the uniform distribution between [−1, 1].

The activation of internal state is computed according to

$$x(n+1) = 1/(1 + \exp(-2 \times (W^{in}u(n+1) + Wx(n) + W^{back}y(n)) + \upsilon)) \tag{19}$$

where υ is noise data randomly sampled from $[-5 \times 10^{-6}, 5 \times 10^{-6}]$.

Note that in our application, $y(n)$ is not the actual output of last time step. To avoid error iteration, $y(n)$ is a coarse judgment about whether the last price stands for a possible reversal or not with the methodology mentioned in Bao and Yang (2008). If last price is a candidate bottom, $y(n) = -1$; if it is a candidate peak, $y(n) = 1$; or $y(n) = 0$. The output of ESN is calculated with linear function.

The current design of ESN relies only on the selection of spectral radius (the largest eigenvalue) of the reservoir's weight matrix. In order to have echo states, the spectral radius of the internal connection weight matrix W must satisfy $|\lambda_{\max}| < 1$. However, recent studies demonstrate that the echo state constraint as a design principle is too weak to build a rich enough reservoir. There are many possible weight matrices with the same spectral radius, but unfortunately they do not all perform at the same level of mean square error (MSE) for functional approximation (Ozturk, Xu, & Príncipe, 2007). Researchers proposed various approaches to overcome this shortcoming. In this paper, we apply Ozturk et al. (2007)'s design of reservoir which improves the richness of ESN dynamics.

3.2. Price series transformation and trading strategy

Stock price often periodically fluctuates and appears as cycles in which up and down intertwines. As trading assistant system, we are only concerned about trend changes or reversal points, not the actual prices. Therefore, we propose a method to simply transform original price series to static time series in which only the information about trend reversal is reserved.

As mentioned before, ideal trading operations should occur at trend reversal points. However, turning points are usually determined from a subjective view. It depends on trading cycles and strategy. In our experiment, we measure reversal points with criteria mentioned in Bao and Yang (2008): Given a price sequence $(x_1, y_1), \ldots, (x_n, y_n)$, ($x_i$ and y_i are the time point and price data of the ith day respectively), a minimal time interval T and a minimal vibration percentage P, if $x_{i+1} - x_i < T$ and $\frac{|y_{i+1} - y_i|}{(y_i + y_{i+1})/2} < P$, remove (x_i, y_i) and (x_{i+1}, y_{i+1}) (see Fig. 8). Following this criterion, we can smooth price series and obtain reversal points both offline and online.

After recognizing turning points, we set the value of buying point as −1 and that of selling point as 1. A pair of buying and selling point is connected with a straight line and points between them are set to the values in the oblique line. Hence, all the prices are transformed to values between [−1, 1] (see Fig. 9). The raw

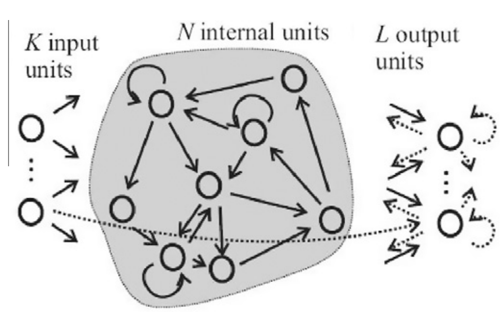

Fig. 7. The architecture of standard ESN (Jaeger & Haas, 2004).

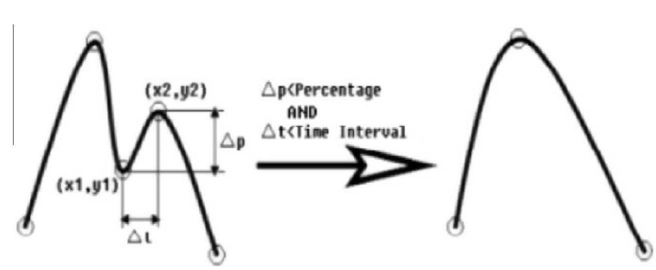

Fig. 8. Criterion for reversal selection (Bao & Yang, 2008).

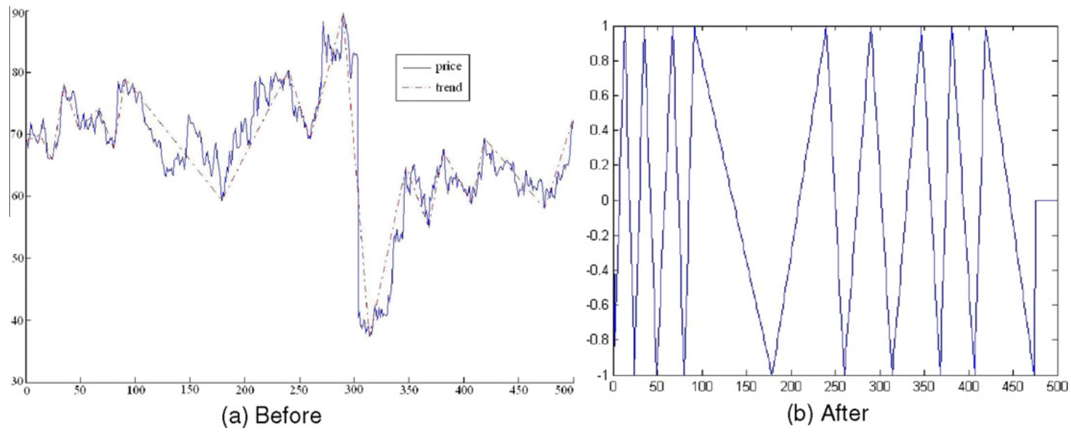

Fig. 9. Price series transformation.

price sequence is converted to a series of triangle functions in which the trend information is maintained. The closer a value is to -1, the more suitable the corresponding price is for buying; the closer a value is to 1, the more suitable the corresponding price is for selling.

In stock transaction, investors should set a threshold $\theta\,(\theta > 0)$ for trading operation first. If the system's output is less than $-\theta$, it is time to buy the stock; if the system's output is more than θ, it is time to sell it. Because it is probable that too many trading signals appear near price reversal points, we adopt the strategy that we operate after the real reversal appears.: we continue smoothing the price online with smaller parameters using the same turning point selection method to obtain candidate reversal points. When trading suggestion comes up, we should first judge whether there exists a possible turning point. If so, follow the suggestion at once; otherwise, we'd better wait until a candidate turning point emerges and execute suggested operation.

4. Experiments and results

To evaluate the performance of our system extensively, we test gains and losses of nearly all stocks of S&P 500 components which cover data over 3000 points and compare the average profits with S&P 500 index and buy-and-hold strategy in bull and bear markets respectively. Each stock is trained and tested alone. Suppose $10,000 initial fund for each of them, we trade all funds/stocks at each operation and consider transaction cost as 0.5%.

In training set, each technical analysis approach is improved using GA and they cooperate to provide input for ESN. Then ESN is trained to approximate a series of triangle functions that is converted from the original price series. In testing set, each enhanced technical analysis method offers their judgment about whether current point is a buy (with value of -1), sell (with value of 1) or no operation (with value of 0) point according to current closing price, which is fed into the trained ESN to give an integrated evaluation about how close current price is to real trend reversal. Investors make decisions according to output of the system and their previous setting of trading threshold.

4.1. Experiments in bull market

Our system performs simulated trading on 438 stocks of S&P 500 components from December 2003 to November 2005 (with over 500 daily prices) for testing and adopts price data of the previous 2000 days for training. In the testing period, S&P 500 index gains about 18.9%; buy-and-hold strategy gains 20.5% in average and our system profits up to 41.6% in average. Statistically, among the 438 stocks, buy-and-hold strategy gains in 280 stocks but losses in 158 stocks; our trading system gains in 413 stocks, losses in 10 stocks and suggests no operation in 15 stocks. Fig. 10 shows the overall market performance of buy-and-hold strategy and our system.

Further observation finds that among the 280 stocks that buy-and-hold strategy profits, our system gains more income in 202 stocks; among the 158 stocks that buy-and-hold strategy deficits, our system still profits in 136 stocks and losses less in other 20 stocks. Table 1 lists the profit margin of buy-and-hold strategy and our trading system in some randomly selected stocks. Trading log of stock CAG (Fig. 11) shows that our trading system can approximately catch reversal points and give corresponding advice for dealing stock.

In a word, our trading system can outperform buy-and-hold strategy in bull market.

4.2. Experiments in bear market

Our system also performs simulated trading on 324 stocks from September 2000 to September 2002 (with over 500 daily prices) for testing and adopts data of the previous 2000 days for training. In the testing period, S&P 500 index losses about 26.5%; buy-and-hold strategy losses 20.3% in average but our system is still able to gain 26.5% in average. Among the 375 stocks, buy-and-hold strategy gains in 116 stocks but losses in 259 stocks; our trading system gains in 324 stocks, losses in 28 stocks and suggests no operation in 23 stocks. Fig. 12 shows the overall market performance of our system and buy-and-hold strategy.

Further observation finds that among the 116 stocks that buy-and-hold strategy profits, our system obtains more income in

Fig. 10. Overall performances in bull market.

Table 1
Profits of individual stock in bull market.

Stock	Model	
	Buy-and-hold (%)	Trading system (%)
AIG	8.96	18.12
ANDW	−22.56	16.34
BEN	98.98	79.59
CAG	1.29	45.23
CCU	−22.32	12.41
EDS	16.59	21.52
HNZ	0.17	14.57
KO	−9.72	4.51
MKC	3.07	16.00
PDCO	−46.41	45.28
PNC	15.50	27.48
SANM	−64.04	−0.54
TUC	57.25	77.50
TXT	52.82	30.60
WY	5.79	35.38
XRX	16.80	36.59

Table 2
Profits of individual stock in bear market.

Stock	Model	
	Buy-and-hold (%)	Trading system (%)
ADI	−78.09	−17.68
BJS	−63.4	76.00
CD	−14.4	64.5
CCU	−22.32	12.41
DOW	−15.3	46.4
FPL	−33.09	22.44
HBAN	1.06	12.58
HPQ	−91.44	−4.64
KRI	5.18	18.25
LTD	−34.93	10.89
MYG	−43.36	−2.68
PMTC	−85.99	−14.51
SLB	−55.61	−7.0
TMO	−34.42	20.51
WYE	−35.79	19.74
ZION	−6.81	18.59

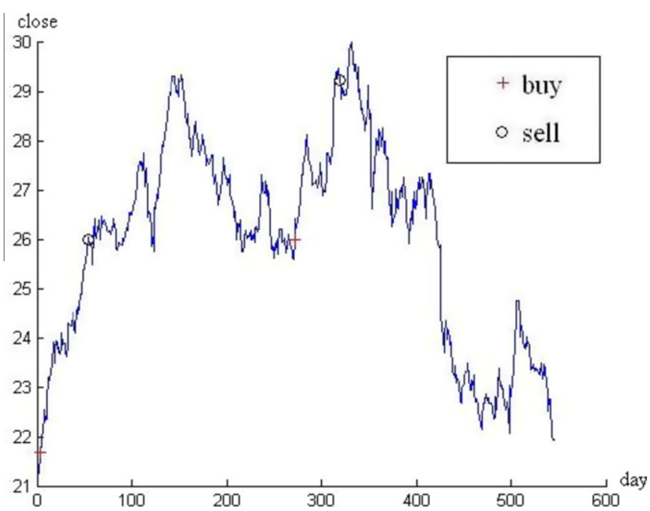

Fig. 11. Trading log of CONAGRA FOOD INC (CAG), profit 45.23%.

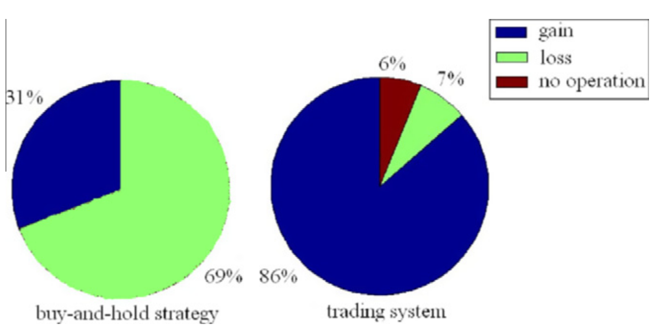

Fig. 12. Overall performances in bear market.

104 stocks; among the 259 stocks that buy-and-hold strategy losses, our system still profits in 208 stocks, suggests no trading operation in 23 stocks and losses less in the other 28 stocks. Table 2 lists the profit margin of buy-and-hold strategy and our system in some randomly selected stocks. Trading log of FPL (Fig. 13) shows that even in bear market, our system can master the fluctuation of price on the whole and capture opportunities to earn.

In a word, our trading system not only outperforms buy-and-hold strategy but also keeps profiting in bear market.

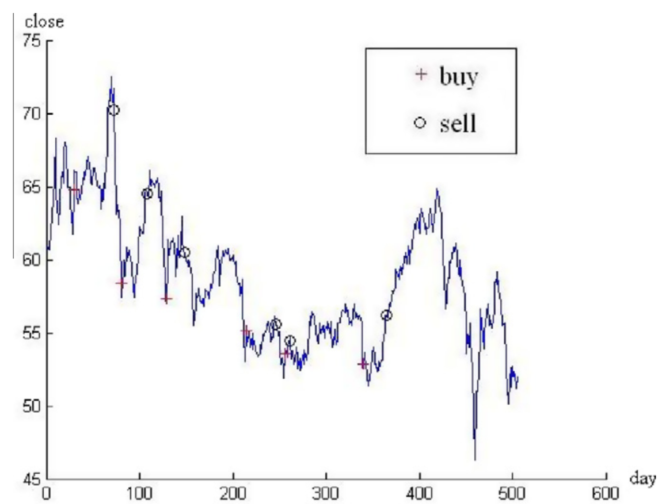

Fig. 13. Trading log of FPL GROUP INC (FPL), profit 22.44%.

5. Conclusions

In this paper, we propose a novel stock trading system based on ESN, which combines a variety of technical analysis approaches enhanced by GA. Simulated experiments in the whole market present that no matter in bull or bear markets, our trading system will obtain more income than buy-and-hold strategy. Especially in bear market in which S&P 500 index drops a lot, our system still profits.

Although our system can facilitate investors in trading decision, how to select appropriate parameters of our model for individual stock is worth further exploration. Parameters, such as the radius of ESN's reservoir and trading threshold, may influence the profits of our system. The results of the above simulated experiments are attained by manually selecting proper parameters. Unfortunately, it is very difficult to determine parameters. Stock market is a very complicated dynamic system. The parameters which are suitable for training set cannot be guaranteed to be proper for testing set.

Further work also includes introducing more enhanced technical analysis approaches and augmenting the system with other soft computing techniques.

References

Baba, N., Kawachi, T., Nomura, T., & Sakatani, Y. (2004). Utilization of NNs & GAs for improving the traditional technical analysis in the financial market. *SICE Annual Conference, 2*(2), 1409–1412.

Bao, D., & Yang, Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Systems with Applications, 34*(1), 620–627.

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Ann Harbor, MI: MIT Press.

Ishii, Kazuo, van der Zant, Tijin, Becanovic, Vlatko, & Ploger, Paul (2004). Optimization of parameters of echo state network and its application to underwater robot. In *SICE annual conference in Sapporo* (Vol. 3, pp. 2800–2805).

Jaeger, H. (2001). The ''Echo State'' approach to analyzing and training recurrent neural networks. GMD-German National Research Institute for Computer Science, vol. GMD Report 148.

Jaeger, Herbert. (2002). Short term memory in echo state networks. GMD-Report 152, GMD-German National Research Institute for Computer Science.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communications. *Science, 304*(5667), 78–80.

Jaeger, H., Lukosevicius, M., & Popovici, D. (2007). Optimization and application of echo state networks with leaky integrator neurons. *Neural Networks, 20*(3), 335–352.

Kim, M.-J., Min, S.-H., & Han, I. (2006). An evolutionary approach to the combination of multiple classifiers to predict a stock price index. *Expert Systems with Applications, 31*(2), 241–247.

Kim, H.-J., & Shin, K.-S. (2007). A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets. *Applied Soft Computing, 7*(2), 569–576.

Ozturk, M. C., Xu, D., & Príncipe, J. C. (2007). Analysis and design of echo state networks. *Neural Computation, 19*(11), 111–138.

Peters, E. E. (1994). *Fractal market analysis: Applying chaos theory to investment and economics*. New York: Willey & Sons, Inc.

Skowronski, M. D., & Harris, J. G. (2007). Automatic speech recognition using a predictive echo state network classifier. *Neural Networks, 20*(3), 414–423.

Vanstone, B., & Tan, C. (2003). A survey of the application of soft computing to investment and financial trading. In *Proceedings of the 8th Australian & New Zealand intelligent information systems conference (ANZIIS 2003), Sydney, Australia, 10–12 December*.

Vanstone, B., & Tan, C. N. W. (2005). Artificial neural networks in financial trading. In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology*. Idea Group.