Expert Systems with Applications 42 (2015) 212–222

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Stock trading rule discovery with an evolutionary trend following model

Yong Hu [a,*], Bin Feng [b], Xiangzhou Zhang [c], E.W.T. Ngai [d], Mei Liu [e,*]

[a] Institute of Business Intelligence and Knowledge Discovery, Guangdong University of Foreign Studies, Sun Yat-sen University, Guangzhou 510006, PR China
[b] School of Management, Guangdong University of Foreign Studies, Higher Education Mega Center, Guangzhou 510006, PR China
[c] School of Business, Sun Yat-sen University, No. 135, Xingang Xi Road, Guangzhou 510275, PR China
[d] Department of Management and Marketing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China
[e] University of Kansas Medical Center, Kansas City, KS 66160, USA

ARTICLE INFO

Article history:
Available online 8 August 2014

Keywords:
Evolutionary trend following algorithm (eTrend)
eXtended Classifier System (XCS)
Trading rule discovery
Concept drift

ABSTRACT

Evolutionary learning is one of the most popular techniques for designing quantitative investment (QI) products. Trend following (TF) strategies, owing to their briefness and efficiency, are widely accepted by investors. Surprisingly, to the best of our knowledge, no related research has investigated TF investment strategies within an evolutionary learning model. This paper proposes a hybrid long-term and short-term evolutionary trend following algorithm (eTrend) that combines TF investment strategies with the eXtended Classifier Systems (XCS). The proposed eTrend algorithm has two advantages: (1) the combination of stock investment strategies (i.e., TF) and evolutionary learning (i.e., XCS) can significantly improve computation effectiveness and model practicability, and (2) XCS can automatically adapt to market directions and uncover reasonable and understandable trading rules for further analysis, which can help avoid the irrational trading behaviors of common investors. To evaluate eTrend, experiments are carried out using the daily trading data stream of three famous indexes in the Shanghai Stock Exchange. Experimental results indicate that eTrend outperforms the buy-and-hold strategy with high Sortino ratio after the transaction cost. Its performance is also superior to the decision tree and artificial neural network trading models. Furthermore, as the concept drift phenomenon is common in the stock market, an exploratory concept drift analysis is conducted on the trading rules discovered in bear and bull market phases. The analysis revealed interesting and rational results. In conclusion, this paper presents convincing evidence that the proposed hybrid trend following model can indeed generate effective trading guidance for investors.

1. Introduction

Quantitative investment (QI) has become a hot topic in the field of finance, and numerous QI products (models/tools/systems) have been developed. One of the most popular methods for designing new QI products is evolutionary learning. This method can effectively and robustly handle optimization problems with a huge search space, which make it suitable for mining knowledge from huge, complex, and nonlinear data sets, such as stock market data.

Evolutionary learning techniques were first applied to trading rule discovery in (Allen & Karjalainen, 1999). However, their research demonstrated that there is no excess return on the S&P 500 after transaction cost. Many studies on trading rule discovery have been carried out since then. For example, based on the S&P 500 from 2000 to 2006, Kaucic (2010) found that her/his trading model can achieve excess return in the bull market, but can only reduce loss in the bear market. These studies have two distinct limitations: (1) stock investment theory is not incorporated for heuristic optimization, and (2) underlying decision-making/prediction algorithm is designed to handle one-step problems rather than multi-step ones (this issue is definitely distinguished in the area of reinforcement learning research) and is incapable of adapting automatically to market directions.

Trend following (TF) is a widely accepted investment strategy because of its simple principle, considering the difficulty of accurate stock prediction. To decide when to buy and when to sell a stock, TF adopts a rule-based trading mechanism based on (long-term and/or short term) market trends rather than on any price forecasting or information gathering (Fong, Tai, & Si, 2011). The

* Corresponding authors. Address: Guangdong University of Foreign Studies, Higher Education Mega Center, Guangzhou 510006, PR China. Tel.: +86 973 596 6597; fax: +86 973 596 5777.

E-mail addresses: henryhu200211@163.com (Y. Hu), fengbin_ace@foxmail.com (B. Feng), zhxzhou@mail2.sysu.edu.cn (X. Zhang), eric.ngai@polyu.edu.hk (E.W.T. Ngai), meiliu@kumc.edu (M. Liu).

http://dx.doi.org/10.1016/j.eswa.2014.07.059
0957-4174/

underlying assumption of TF is that someone has obtained the market information prior to us, and we can follow the person with considerably less cost for information collection (Covel, 2009). The stock industry has numerous successful TF-based QI products, e.g., the Turtle Trader (Fong, Si, & Tai, 2012). As demonstrated in (James, 2003), TF is a valid trading strategy in currency market, where the simple moving average indicator can gain excess return with adequate information ratio. Meanwhile, Szakmary, Shen, and Sharma (2010) examined the performance of TF trading strategies in commodity futures markets using a monthly dataset spanning over 48 years and 28 markets. Their method yielded positive mean excess returns after transaction costs in 22 markets. Despite the importance and effectiveness of TF, relatively few academic works have investigated the application of machine learning (ML) techniques to enhance TF strategy. The first attempt was carried out by Fong et al. (2012), who applied fuzzy logic to construct a dynamic model for long-term trend trading. The selected TF indicators, however, were hardly optimal ones because no feature selection or equivalent mechanism (e.g., evolutionary learning) was considered.

To solve the above mentioned problems, this paper proposes a hybrid long-term and short-term evolutionary TF algorithm (eTrend). Specifically, eXtended Classifier Systems (XCS) and TF investment strategies are combined in eTrend in a way that the identified long-term and short-term trends are integrated to evolve the XCS, thus achieving a novel constraint learning paradigm. The eTrend is designed in this way for several reasons: (1) long-term trend can reduce volatility whereas short-term trend enables rapid response, (2) XCS acts similarly to a group of robot investors by continuously and automatically adapting to the current stock environment, and (3) XCS can uncover explicit trading rules rather than black-box relationships.

In conclusion, the eTrend model has two distinguished advantages. First, the combination of stock investment strategies and evolutionary learning can significantly reduce computation effort for feature selection and improve model practicability. Second, and more importantly, eTrend can automatically adapt to market directions and the discovered trading rules are reasonable and understandable, further analysis on which can prevent average investors from irrational trading.

As the stock environment changes frequently, the underlying data distribution may change as well over time, making it difficult for any ML model that was built on old data to perform well on the new data. This problem is called concept drift (Tsymbal, 2004), which commonly exists in the stock market, especially between different market phases such as the bull and bear phases. However, few studies have been conducted on concept drift analysis. Thus, in the present paper, we analyze the concept drift problem of the Chinese stock market based on rules discovered through eTrend.

The rest of the paper is organized as follows. A brief survey on related works is presented in Section 2. The research methodology is depicted in Section 3, which describes the design of the quantitative trading model. The experiment results are presented in Section 4. A summary and conclusions are provided in Section 5.

2. Related works

The academe has had a long-standing discussion on the effectiveness of technical analysis in stock trading. Some argued that stock prices are not predictable because all relevant public information is already reflected in the prices. However, recent studies such as those by Brock, Lakonishok, and LeBaron (1992) and Blume, Easley, and O'Hara (1994) have presented positive empirical evidence on the effectiveness of technical analysis (Kaucic, 2010).

Various ML algorithms have been adopted for technical analysis, among which the most popular ones are Artificial Neural Network (ANN) (Cao, Leggio, & Schniederjans, 2005) and Support Vector Machine (SVM) (Huang, Nakamori, & Wang, 2005). However, ANN and SVM are both black-box models, which can hardly provide any insight into the nature of interactions between technical indicators and stock market fluctuations (Lai, Fan, Huang, & Chang, 2009). Michael W. Covel, one of the most famous fund managers in the US, stressed that "black-box makes me uncomfortable, in these trading, the situation which I cope with is the indigestion algorithm."(Covel, 2009). This opinion implies that a trading model should only be put into practice when it is understandable by investors (John, Miller, & Kerber, 1996); otherwise, the risk is unlimited.

Thus, many studies have been carried out to uncover intelligible stock trading rules. Allen and Karjalainen (1999) were the first to implement the generation of automatic trading rules by applying genetic programming (GP). Soon afterward, numerous other researchers recognized the importance of intelligible rule discovery. Table 1 lists several representative studies.

Genetic algorithm (GA) and GP are widely used for rule discovery, such as those in (Allen & Karjalainen, 1999; Esfahanipour & Mousavi, 2011; Gorgulho, Neves, & Horta, 2011; How, Ling, & Verhoeven, 2009; Kaucic, 2010; Mehta & Bhattacharyya, 2004; Núñez-Letamendia, 2007; Tsang, Yung, & Li, 2004). The process of rule construction is based on evolutionary learning, which aims to adapt the rules to the current environment and searches for the global optimum rules in the huge search space. Other popular ML methods include decision tree (DT) (Wu, Lin, & Lin, 2006), fuzzy logic (Bekiros, 2010; Chiung-Hon, Liu, & Wen-Sung, 2006), rough set (Shen & Loh, 2004), and grid search (Chong & Lam, 2010). In particular, explicit rules can be extracted from the black-box model. For example, Lam (2004) proposed a novel rule extraction method called GLARE to reveal the prediction logic and procedure of a black-box ANN.

Most of the above studies adopted different algorithms to perform statistical deduction. For example, DT represents the highest entropy, and ANN converges to the minimal mean squared error. However, these methods lack rigorous economic and financial interpretability; in other words, they are not suitable for guiding actual investment behaviors.

XCS is a special method for discovering automatic trading rules. GA and reinforcement learning are embedded in XCS, enabling XCS with the potential to act like a robot investor. Although there exist several studies on the application of XCS in trading rule discovery, such as those in (Hsu, Chen, & Chang, 2011), the research direction is still at its preliminary stage.

3. Methodology

3.1. XCS

Fig. 1 shows a brief development history of XCS from the original idea of Holland's learning classifier system (LCS) (Holland, 1976) to Wilson's XCS (Wilson, 1995). Many aspects of XCS are derived from ZCS, a "zeroth-level" classifier system intended to simplify Holland's canonical framework while retaining the essence of the classifier system idea. The main differences between XCS and ZCS are derived from the definitions of classifier fitness function, GA mechanism, and the more sophisticated action selection in XCS. XCS was designed to solve complex problems that have numerous optimal solutions and are difficult to choose with only a few attempts. It has demonstrated promising performance in maze and multiplexing problems (Wilson, 1995).

Table 1
Representative studies on trading rule discovery.

Authors (Year)	Methodologies	Sliding Window	Better than B&H	Trans action cost	Risk measure	Data sets
Allen and Karjalainen (1999)	GP	☐		☐		S&P 500
Mehta and Bhattacharyya (2004)	GA			☐		S&P 500
Tsang et al. (2004)	GP					
Lam (2004)	NN(rule extraction by GLARE)	☐				364 S&P companies
Shen and Loh (2004)	Rough Set, SOM				Sharpe Ratio	S&P500, MATIF-CAC, EUREXBOND, CBOT-US
Chiung-Hon et al. (2006)	Fuzzy	☐				TAIEX
Wu et al. (2006)	DT			☐		41 Taiwan stocks and 248 NASDAQ stocks
Núñez-Letamendia (2007)	GA			☐		25 stocks In Madrid Stock Exchange, IBEX-35, General Index
Chavarnakul and Enke (2009)	Fuzzy NN, GA	☐	☐	☐		S&P 500
Kaucic (2010)	Rule set selection by GA, PVC, BMA, BOOST voting for rules		☐	☐		S&P 500
How et al. (2009)	GP	☐		☐		Russell 1000 index, Russell 2000 index, Russell 3000 index
Bekiros (2010)	Fuzzy NN	☐	☐	☐	Sharpe Ratio	KLCI Composite, Stock Exchange Weighted, HangSeng, Jakarta Stock Exchange Composite, Straits Times (New), SET 100 Basic Industries, SP500, NYSE, FTSE100, CAC40
Chong and Lam (2010)	Grid Search	☐		☐		NASDAQ, NYSE, S&P500, DJIA
Esfahanipour and Mousavi (2011)	GP		☐	☐	Conditional Sharpe Ratio	PKOD1, JOSH1, NBEH1, SAKH1, PETR1, SIPA1, IKCO1, PARK1, CHML1, DSOB1
Gorgulho et al. (2011)	GA	☐	☐	☐	Sharpe Ratio, Sortino Ratio	30 stocks from Dow Jones index
Hsu et al. (2011)	XCS	☐	☐	☐		TX, MSCI

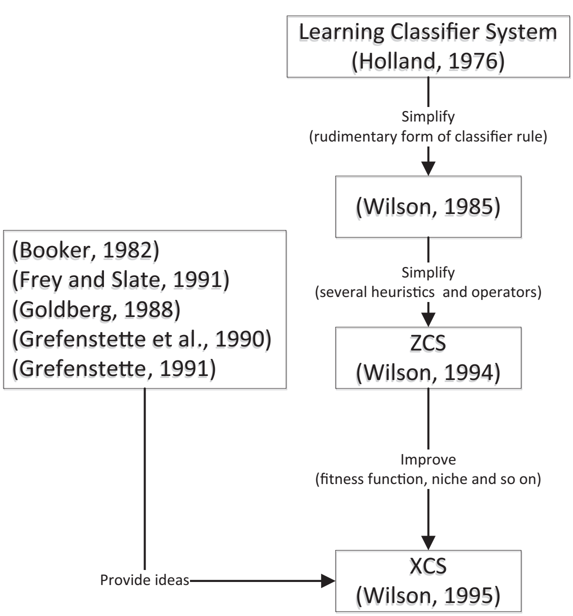

Fig. 1. The development history of XCS.

The execution process of XCS includes four steps: perception, action, feedback, and learning. Fig. 2 depicts the overall architecture of XCS and shows an interaction process with the environment.

(1) Perception. The animat system embedded in XCS communicates with environment in the way that the environment states are encoded by 0/1 values according to specific indicators. Each of these values represents the status of an indicator of the underlying problem. Then the rudimentary coding sequence is delivered to the detectors for the next step of action selection.
(2) Action selection. The rule bank (i.e., population) labeled with [P] stores classifiers in the form of "if <condition> then <action>" and other properties of these classifiers such as the precision, fitness, and error. For example, the first classifier is "#011:01", and, among this "1" stands for meeting the corresponding indicator, "0" stands for without meeting, #" stands for do not care, and the binary sequence after the colon is the code of action. Set [P] is the robot's memory, at first all rules in the rule bank are created randomly. After a few steps of learning, [P] would be filled with appropriate rules adapting to the specific environment. Then classifiers that match the current environment state received by detector are selected as a match set labeled with [M]. If no matched classifier exists, the covering process is triggered to produce a new classifier with the current condition and random action. Based on both the associated action and the properties of each classifier in the match set [M], a prediction array [PA] of expected rewards for each action is calculated. Finally, the appropriate action is determined by selecting the action with maximum expected reward.
(3) Feedback. The effectors execute the determined actions that may lead to positive/negative rewards provided by the environment, which yields a guidance of adjustment on current knowledge (rule bank) in XCS.
(4) Learning. Internal classifiers of previous action set are then adjusted according to this reward by the reinforcement learning process. The adjusted classifiers are then saved in XCS's memory, and sometimes out-of-date classifiers are removed. This procedure enables XCS with more powerful adaptive capability.

Compared to Holland's LCS, XCS stands out in several aspects. (1) The GA procedure is periodically invoked in the niches defined by the match set [M] according to the environment rewards, which enables XCS to eliminate the situation where high-reward

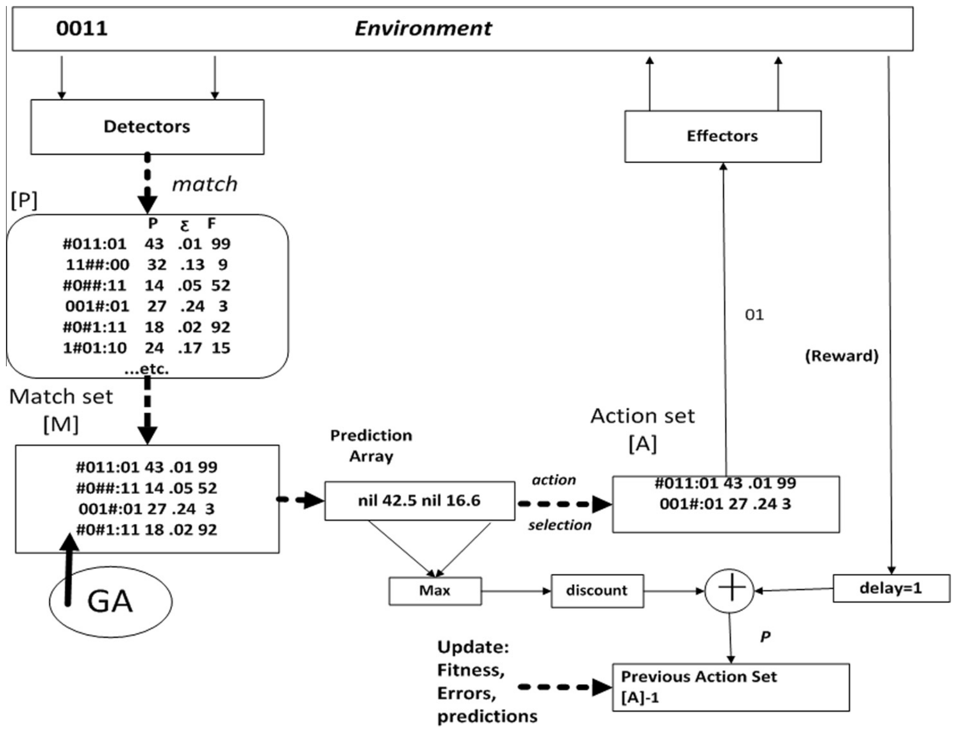

Fig. 2. A schematic illustration of XCS (Wilson, 1995) (pp. 152).

//The update process on value of p in [A]

$$p \leftarrow p + \beta(\text{actual payoff} - p) \qquad (1)$$

// The update process on value of ε in [A]

$$\varepsilon \leftarrow \varepsilon + \beta(|\text{actual payoff} - p| - \varepsilon) \qquad (2)$$

// The update process on value of fitness in [A]:

accuracy sum $\leftarrow 0$

Initialize accuracy vector k for each classifier in [A]

If (prediction error$<\varepsilon 0$) $\quad k \leftarrow 1$

Else $\quad k \leftarrow \exp[(\ln \alpha)(\varepsilon - \varepsilon 0)/\varepsilon 0] \qquad (3)$

accuracy sum $\leftarrow$ accuracy sum $+k$

For each classifier in [A]

fitness$\leftarrow$fitness$+\beta \times k \div$(accuracy sum- fitness) $\qquad (4)$

Notes: α , $\varepsilon 0$ are the parameters that indicates the updating rate of p, ε, and fitness.

Fig. 3. The reinforcement learning process of XCS.

classifiers seize the learning population. (2) The measurement of the classifiers' survival probability in the population is not only the return but also the accuracy based fitness. (3) XCS delicately absorbs the popular Q-learning technique in the reinforcement learning process.

In Fig 3, the formulas of reinforcement learning process are shown. Same as the Q-learning process, the formulas parameters affect the prior step experience learning and next step decision making process.

Applying XCS in trading rule discovery has two main advantages. First, compared to traditional algorithms for rule discovery, XCS is dynamic, evolutionary, and intelligent. Three rule learning modes are used: genetic operation, covering (the completely new environment states), and generalization (of more specific rules based on a subordinate relationship). The detection of rules and adaption to environmental variation are instant. Second, from the concept drift perspective, the prediction accuracy is continuously optimized to achieve the self-adaption to the new environment, which results in the capacity of concept drift discovery and analysis between different market phases.

3.2. eTrend

This study established a profitable model based on the hybrid of XCS and TF trading strategy. Fig. 4 illustrates the main procedure flow of eTrend. The training and testing periods are divided using the sliding window method. We treat all the trading data of $t-1$ year as the training data and the data of t year as the testing

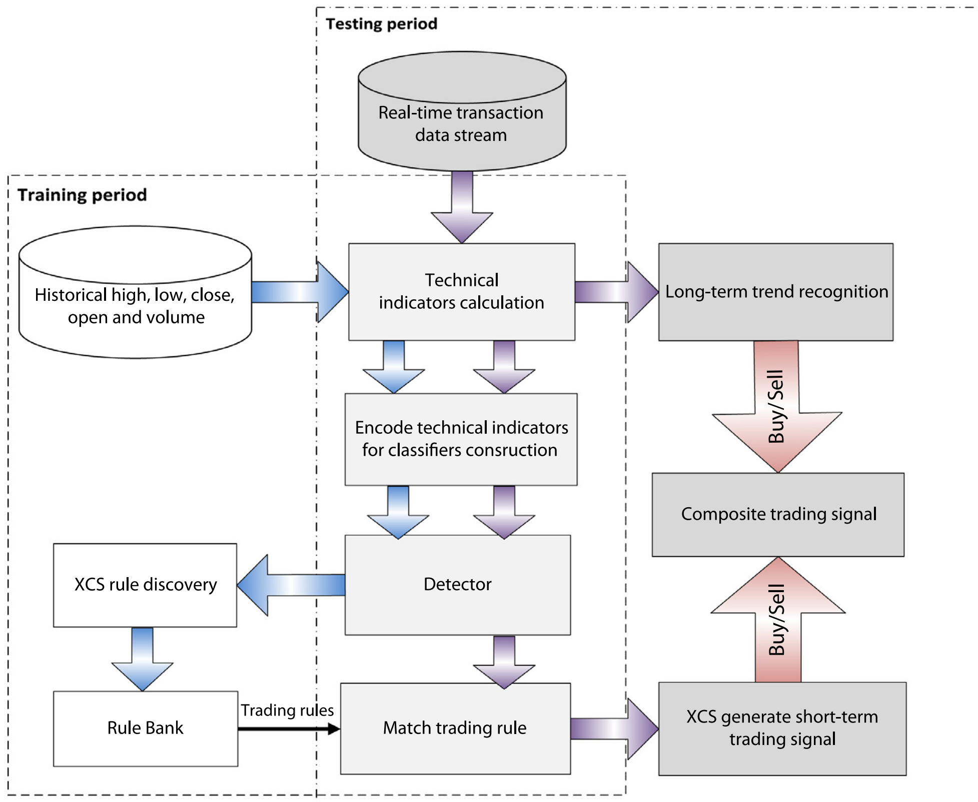

Fig. 4. The main procedure flow of eTrend.

data. In the next period the data of t year is used as the training data the $t+1$ year as the testing data. Eventually, if t equal to 2013, the training–testing process will be terminated. The input data are daily trading information stream, including the open, high, low, and closing prices and volume. Because of the difficulty in analyzing these irregular raw data, following common practice, these data are converted to technical indicators before both the training and testing periods. The selected technical indicators include moving average (MA) and volume moving average (VMA), as well as other indicators. These indicators are encoded as a string of 0/1 s, which represents the current stock state and is handed over to the detector of XCS.

The main difference of testing period and training period is embodied in three aspects:

(1) Data used in the testing period are real-time transaction data (note that out-of-sample data are used in the subsequent experiments) while the training period uses historical data.
(2) The training period is to learn the best trading rules to fill the rule bank, i.e., the knowledge of a robot investor. In contrast, the testing period is to put this knowledge into practice. XCS aims to identify the most accurate and matching classifiers (rules) with the current stock market condition and generate the corresponding action after "thinking".
(3) Several technical indicators are used for long-term trend recognition to determine the long-term trading signal. XCS is used to generate short-term trading signal. Afterward the composite trading signal is generated, in other words, eTrend is based on a hybrid of long-term and short-term trend recognitions.

Fig. 5 shows the trading logic of eTrend. The decision-making procedure is that if a long-term up-trend (down-trend) is identified, then the long-term trend transaction lock will be opened (closes). As the transaction lock is opened, the best rules generated by XCS are continuously matched for precise short-term trend buy timing. Whenever a buy signal is generated by XCS, the corresponding stock would be contained in a portfolio. The entire model waiting for the next sell timing is also handled in a similar manner. Specifically, the long-term trend guides the investment direction and the short-term trend determines the buy/sell timing.

4. Experiment

4.1. Data

Three indexes from the Shanghai Stock Exchange were chosen to evaluate the eTrend model. These indexes are the Composite Index, the Industrial Index, and the Business Index, all of which cover the period from January 1, 2001 to July 31, 2013. The raw trading information was daily stock trading data, including open, high, low, and closing (price) as well as volume.

4.2. Long-term and short-term indicators

Due to the difficulty of analyzing the irregular daily trading information stream, including the open, high, low, and closing price and volume, these raw data was converted to the technical indicators required in this paper.

For long-term trend analysis, the MA-based method was selected as the long-term trend identifier. This is based on the idea in James (2003), in which this simple trading strategy was verified to be superior in the currency market. In stock trading practice, MA-method is simple but can clearly and smoothly figure out the trend. This unique characteristic contributes to the effectiveness and popularity of the MA-method. Table 2 lists the MA-based long-term trend indicator adopted in this study.

```
Repeat until end of market
If long-term up-trend is identified (price is advancing)
        Open the transaction lock
        Detect Inputs (current trading state on raw data)
        Convert inputs into Bit string (Encoding raw data according to formula in Fig. 5)
        If (Inputs are in Population)
                Generate Match set [M]
        Else
                Covering
        For (Each action in March Set)
        Compute system prediction
        Obtain prediction array [PA]
        Select Action
        If XCS generate buy signal (Short-TF timing verified)
                If no position has been opened
                        Generate buy signal
        Else
                NOP
Else if long-term down-trend is identified (price is declining)
        Close the transaction lock
        Detect Inputs (current trading state on raw data)
        Convert inputs into Bit string (Encoding raw data according to formula in Fig. 5)
        If (Inputs are in Population)
                Generate Match set [M]
        Else
                Covering
        For (Each action in March Set)
        Compute system prediction
        Obtain prediction array [PA]
        Select Action
        If XCS generate sell signal
                If one position has been opened
                        Generate Sell signal
                Else
                        NOP
Else
        Waiting
If end of market
        Close all opened positions
```

Fig. 5. Pseudo-codes of eTrend.

As for the short-term trend analysis, according to Fasanghari and Montazer (2010), the combination of scientific methodology and personal experience is the vital point to success. In eTrend, XCS learns short-term TF trading rules in terms of seven indicators adopted from (Anbin & Junfu, 2005), all of which are technical indicators widely accepted by investors. These technical indicators are 0/1 variables, as defined in Table 3.

Table 2
Long-term trend analysis indicators-encoding for XCS.

Trend types	Indicator
Up-trend	Current day's close price > MA (price in 20 days before)
Down-trend	Current day's close price < MA (price in 20 days before)

4.3. Experiment scheme

Risk control and transaction cost were considered in our experimental analysis. The parameter settings of these issues are listed in Table 4. In trading practices, the transaction cost is an important expense in technical analysis based investment strategies. In this paper, transaction costs are calculated in accordance with the reality of Shanghai Stock Exchange. Moreover, some risk control methods were implemented in this study. Specifically, the stop loss is a specific threshold used to prevent eTrend from continuous huge losses. The high retreat (ratio) is to confirm the current revenue and reduce the loss of gained profit.

In addition, Table 5 lists the important parameter settings of the eTrend model. For example, in leaning process of XCS, the value of Theta_GA, is a threshold that indicates when the learning population is over 25, GA should be applied on the action set for learning new classifier. In GA, the other parameter pM represents the probability of mutating one allele and the action in an offspring classifier.

Common evaluation indicators include return, excess return, and annualized return. However, return should not be the only

Table 3
Short-term trend analysis indicators-encoding for XCS.

#.	Indicator definition
1	a. Stock price breakout on a new high 10
2	b. Volume of stock breakout on a new low
3	c. MA5 > MA60
4	d. VMA5 > VMA22
5	e. DIF golden cross DEA
6	f. MACD 12 cross MACD26
7	g. 9 K > 9D

Note: DIF: MA26-MA12; DEA: the 9 days moving average of DIF; MACD: Exponential Smoothing Moving Average; RSV = (Close-Low) ÷ (High-Low) × 100; K: 2/3 × Kt-1 + 1/3 × RSVt; D: 2/3 × Dt-1 + 1/3Kt.

evaluation indicator considered. Risk is also important and should receive more attention. To better evaluate both return and risk, we adopt the Sortino ratio, which is different from the famous Sharpe ratio in that the former distinguishes between the rise and fall of fluctuation. The underlying idea is that the upside movement is desirable to investors so that this kind of fluctuation is not regarded as a risk. The higher the Sortino ratio, the higher the return that can be obtained with lower downside risk. The definition of Sortino ratio is defined as follows:

$$Sortino = [mean(Data) - MAR]/square\ [lpm\ (Data, MAR, 2)],$$

where *Data* is the series of portfolio return, *MAR* is the scalar minimum acceptable return (default $MAR = 0$), and *lpm* is the lower partial moments.

4.4. Experiment results

As sliding window method was applied, the eTrend was trained for twelve times. In each training period, the rule bank was updated. For example, Table 6 lists the most accurate rules discovered in the testing years 2001 and 2002. These rules are simple and highly accurate, with some rules achieving over 80% accuracy. These technical trading rules are promising that high reward is expected.

On the perspective of return of investment, performance comparison between three indexes are shown in Fig. 6. The eTrend is capable of obtaining excess return in almost all the market phases regardless of whether it is compared with buy and old or with the Composite Index. This result indicates that eTrend is suitable for not only the general investors but also the hedge fund managers. Furthermore, in the bull market phase (2006–2008), the model can generate excess return continuously, meanwhile, in bear market phase (2008–2009) it can avoid falling too low as the market environment does. It indicates that eTrend can not only evade the down-side risk in the bear market phase but also make several winning decision. For further comparison to other algorithms, the DT and ANN were used to replace the XCS as the core algorithm in eTrend. For a fair comparison, both long term trend and risk control are used on experiment of DT and NN as well. The result demonstrates that the advantage of XCS is obvious in comparison to DT and ANN, because the return series of XCS is smooth and fruitful.

As shown in Table 7, eTrend significantly outperforms the buy-and-hold strategy with high Sortino ratio. The annualized returns are approximately 10% over the three indexes after the transaction cost. Over all indexes, the Sortino ratios are close to 1, which indicate that eTrend gets a high return within a low downside risk.

Fig. 7 shows the performance of every trading decision on the three indexes. The wining rate is approximately 50%. However, returns of the winning trading decisions are, on average, much higher than those of the losing ones. Thus, eTrend can naturally achieve high end return. This result is consistent with the nature of the TF strategy, that is, it is not concerned about the immediate gain every time but it is expected to win in the end of accumulative return.

Table 4
The parameter settings of transaction cost and risk control.

Parameters		Descriptions	Values
Transaction cost			
	Buy cost	The cost of buy operation	0.001
	Sell cost	The cost of sell operation	0.002
Risk control			
	Stop loss	If the price is lower than the buy price by a given threshold, then the sell operation is executed	0.05
	High retreat	If the price is lower than the highest price in hold periods by a given threshold, then the sell operation is executed	0.05

Table 5
Parameter settings of the eTrend model.

Parameters	Descriptions	Values
For leaning process of XCS		
Condition_len	The length of condition	7
Alpha	The fall rate in the fitness evaluation	0.1
Beta	The learning rate for updating fitness, prediction, prediction error, and action set size estimate in the XCS classifiers	0.2
Gamma	The discount rate in multi-step problems	0.95
Delta	The fraction of the mean fitness of the population below which the fitness of a classifier may be considered in its vote for deletion	0.1
Nu	Specifies the exponent in the power function for the fitness evaluation	5
Theta_GA	The threshold for the GA application in an action set	25
Epsilon	The error threshold under which the accuracy of a classifier is set to one	10
Theta_del	Specifies the threshold over which the fitness of a classifier may be considered in its deletion probability	20
P_dontcare	The probability of using a do not care symbol in an allele when doing the covering step	0.5
PredictionErrorReduction	The reduction of the prediction error when generating an offspring classifier	0.25
FitnessReduction	The reduction of the fitness when generating an offspring classifier	0.1
Theta_sub	The experience of a classifier required to be a subsumer	20
For GA		
pX	The probability of applying a crossover in an offspring classifier	0.8
MaxPopSize	The maximum population of GA	800
pM	The probability of mutating one allele and the action in an offspring classifier	0.04

Table 6
The most accurate rules discovered in 2001 and 2002.

Year	Conditions	Actions	Accuracy
2001	###1010	1 (Buy)	0.51
	0##1##0	0 (Sell)	0.51
	##100#0	1 (Buy)	0.98
	#0#00#0	0 (Sell)	0.52
	##0####	1 (Buy)	0.72
	0##0###	0 (Sell)	0.61
2002	##00###	0 (Sell)	0.80
	0####0#	1 (Buy)	0.77
	000#00#	0 (Sell)	0.58
	10##000	1 (Buy)	0.74
	#001##0	0 (Sell)	0.60
	#1#####	0 (Sell)	0.87
	##0##10	1 (Buy)	0.75
	0#1#010	0 (Sell)	0.51
	00110#0	1 (Buy)	0.74
	#010#10	1 (Buy)	0.74
	001####	0 (Sell)	0.70

4.5. Concept drift analysis

The concept drift phenomenon is common in the stock market. Between different market phases, such as bear and bull markets, the characteristics of up-trend stocks are not the same. The following analysis is based on the intelligible rules detected by the eTrend model on the Business Index.

Fig. 8 lists the rules generated in different periods, including two bear phases and one bull phase. The result shows that the concept drift exists between different years and between different stock market phases, which results in variable rules.

Table 8 presents the detailed information on discovered rules. Among the two bear periods (2004 and 2008), the same rule "condition: ###1##0 action: 1" can be found. The value of the fourth indicator is 1, i.e., VMA5 > VMA22, implying that the stock was recently transacted frequently. The value of the seventh indicator is 0, i.e., 9 K < 9D, indicating that the price is not yet up. According to the financial implication, under these conditions in the bear market phase, the buy chance is forthcoming. The buy signal generated by the eTrend model is consistent with the financial analysis.

However, between the bull (2007) and bear (2008) market phases, the second rule in 2008 is "condition: ###1#0# action: 0" while the fourth rule in 2007 is "condition: ###1#0# action: 1." That is to say, even if the important indicators fall under the same conditions, opposite actions should be taken during different market phases. Specifically, the value of the fourth indicator is 1, i.e., VMA5 > VMA22. The value of the sixth indicator (MACD 12 cross MACD26) is 0. In this case, the interpretation is that in the bull market phase the price is not yet up (buy signal), whereas in the bear market phase the price is declining (sell signal).

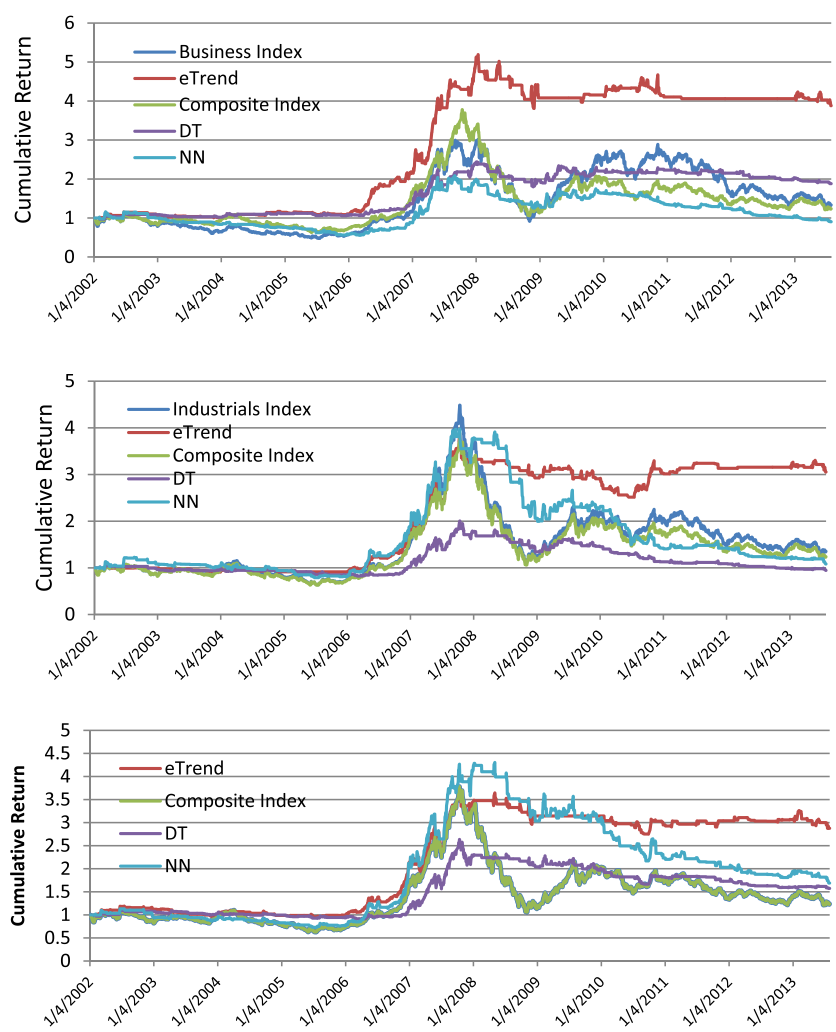

Fig. 6. Comparison of performance on three indexes.

Table 7
Detailed experiment results.

Stock	Annualized Return (%)	AR (%)	AR on B&H (%)	AR on DT (%)	AR on ANN (%)	SR
Shanghai Composite Index	9.89	187	23.7	57.3	68.9	0.94
Shanghai Industrial Index	10.48	205	34.7	−5.5	8.29	0.81
Shanghai Business Index	12.86	288	36.6	88	−10.7	1.05

Notes: AR indicates Accumulative Return; SR indicates Sortino Ratios.

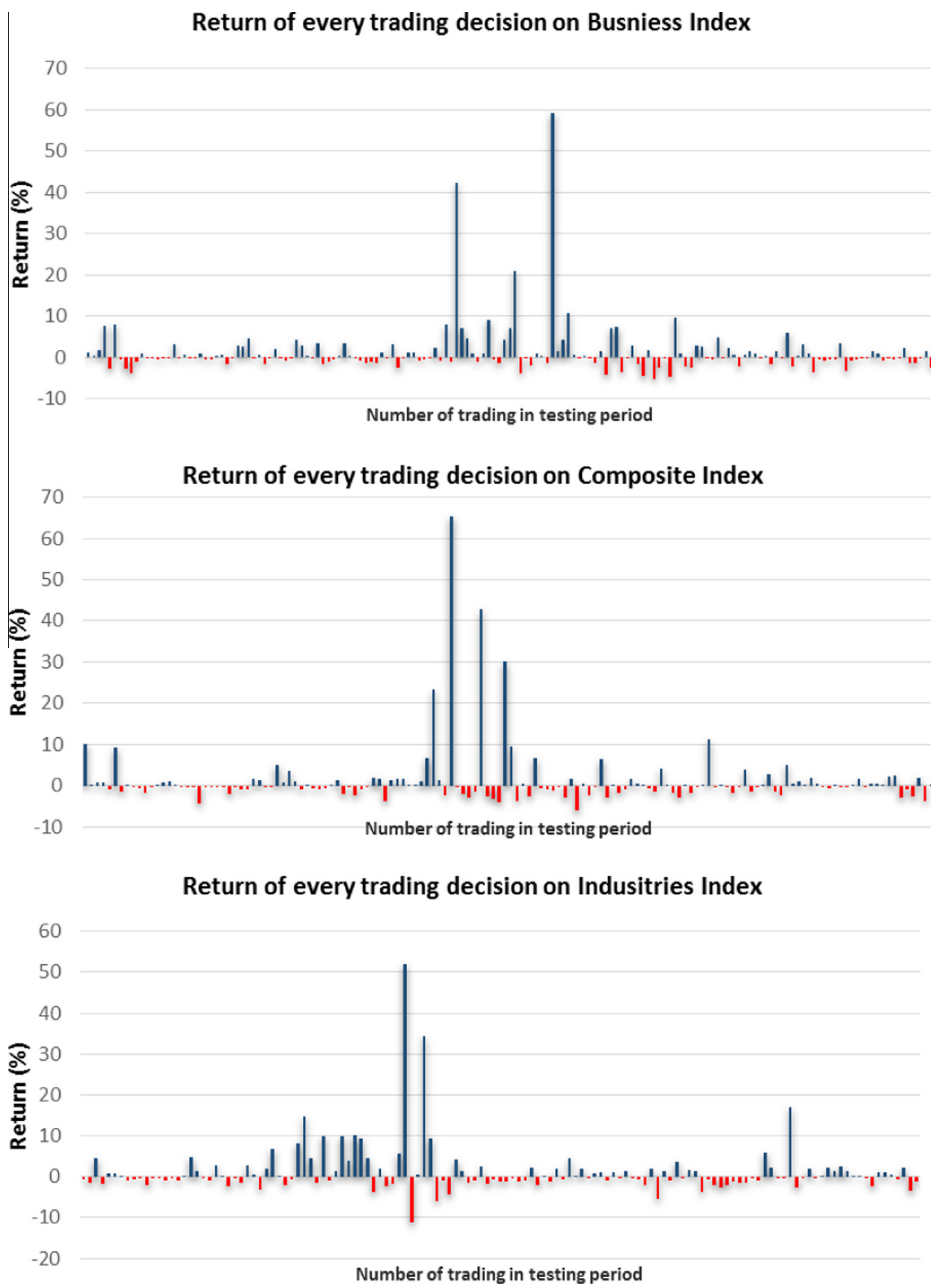

Fig. 7. The performance of each trading decision on three indexes.

In conclusion, the rules generated by our eTrend model in different market conditions are reasonable and in accordance with general financial knowledge. This knowledge can certainly help investors in stock investment with lower risk and higher return.

5. Conclusion and future work

The main contribution of this research is the combination of classical stock investment rules (i.e., trend following) and

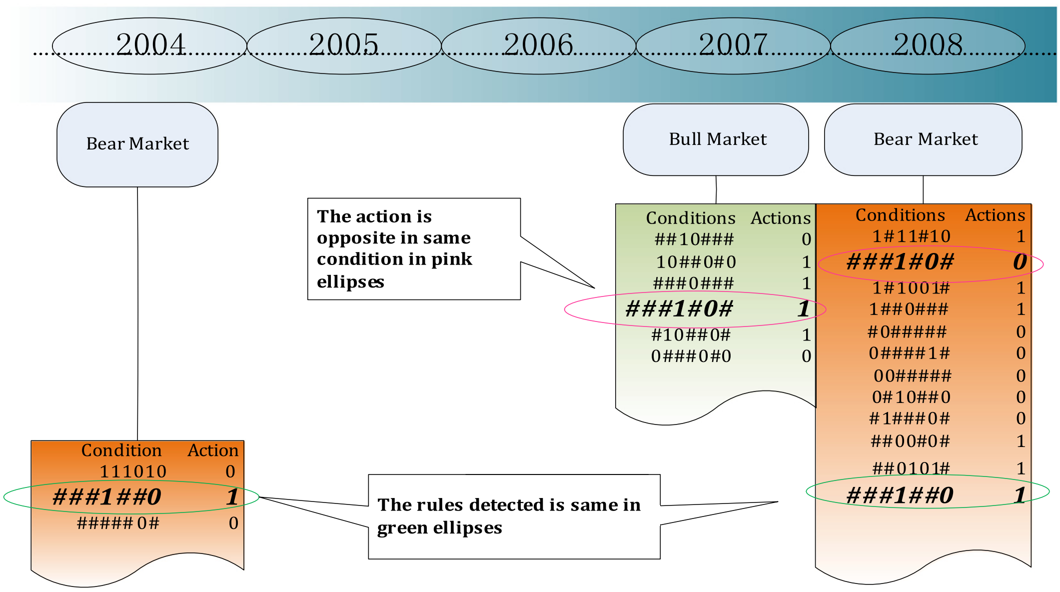

Fig. 8. Comparison of rules between different stock market phases.

Table 8
Detailed rules analysis.

No.	Conditions	Details	Actions	Notes
1	###1##0	VMA5 > VMA22;9 k ⩽ 9D; Others do not care	1(Buy)	Appeared in both 2004 and 2008 Bear Market
2	###1#0#	VMA5 > VMA22;MACD 12 without cross MACD26;Others do not care	1(Buy in Bull market) 0(Sell in Bear market)	The action is opposite in the same condition at different market phases

evolutionary learning (i.e., XCS), which can significantly improve the computation effectiveness and practicability of stock trading models. Moreover the eTrend can automatically adapt to market directions and uncover reasonable and understandable trading rules and have a good ability in automatically handling the concept drift problem.

The significance of this research is two folds. Firstly, for a long time, evolutionary learning is one of the most important algorithms, and trend following is one of most useful trading strategies in investment practice. However, evolutionary learning and trend following is incorporated in few academic research. Different from (Mabu, Hirasawa, Obayashi, & Kuremoto, 2013), which makes a random choice of technical indicators as model inputs, the inputs of our eTrend are from the practice of investment experts. This research fills up this gap and paves the way for future researches in this direction. Secondly, the trading strategy adopted in this paper is simple. By using evolutionary learning, many valuable trading rules are discovered at the running time. This model can effectively avoid the trading irrationality of general investors and the rules discovered are intelligible and reasonable, and suitable for further analysis and verification by expert. Our eTrend model improves the performance by incorporating expert knowledge with evolutionary learning method, which can avoid the subjective judgment of investors and thus directly reduce the risk.

Overall, this research is a good example of quantitative investing model. The eTrend proposed in this paper is verified as a novel QI product. Experimental results indicate that eTrend can obtain excess returns with a high Sortino ratio after the transaction cost. Moreover, eTrend surpassed the ANN and DT in almost all of the testing periods. In addition, XCS is identified as an effective solution for the concept drift, thus facilitating analysis between the rules discovered in different market phases.

Limitation of this study is that, the trading strategy rules selected from the industry are sub-optimal, which may lead to the fact that the rules discovered by eTrend are not global optimal.

In future studies, eTrend can be further improved by (1) integrating more sophisticated evolutionary learning approach, and (2) including more knowledge from investment experts as the input of the learning algorithm.

Acknowledgments

This research was partly supported by the National Natural Science Foundation of China (71271061, 70801020), Science and Technology Planning Project of Guangdong Province, China (2010B010600034, 2012B091100192), and The Ministry of Education Innovation Team Development Plan, Guangdong Natural Science Foundation Research Team, and Business Intelligence Key Team of Guangdong University of Foreign Studies (S2013030015737, IRT1224, TD1202).

References

Allen, F., & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of Financial Economics, 51*(2), 245–271. http://dx.doi.org/10.1016/S0304-405X(98)00052-X.

Anbin, C., & Junfu, S. (2005). Application of XCS classifier system to financial decision support: An empirical study of TAIEX. *Chinese Journal of Management, 2*.

Bekiros, S. D. (2010). Fuzzy adaptive decision-making for boundedly rational traders in speculative stock markets. *European Journal of Operational Research, 202*(1), 285–293. http://dx.doi.org/10.1016/j.ejor.2009.04.015.

Blume, L., Easley, D., & O'Hara, M. (1994). Market statistics and technical analysis: The role of volume. *The Journal of Finance, 49*(1), 153–181. http://dx.doi.org/10.1111/j.1540-6261.1994.tb04424.x.

Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *The Journal of Finance, 47*(5), 1731–1764. http://dx.doi.org/10.2307/2328994.

Cao, Q., Leggio, K. B., & Schniederjans, M. J. (2005). A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market. *Computers & Operations Research, 32*(10), 2499–2512. http://dx.doi.org/10.1016/j.cor.2004.03.015.

Chavarnakul, T., & Enke, D. (2009). A hybrid stock trading system for intelligent technical analysis-based equivolume charting. *Neurocomputing, 72*(16–18), 3517–3528. http://dx.doi.org/10.1016/j.neucom.2008.11.030.

Chiung-Hon, L. L., Liu, A., & Wen-Sung, C. (2006). Pattern discovery of fuzzy time series for financial prediction. *Knowledge and Data Engineering, IEEE Transactions on, 18*(5), 613–625. http://dx.doi.org/10.1109/TKDE.2006.80.

Chong, T. T., & Lam, T. (2010). Predictability of nonlinear trading rules in the U.S. stock market. *Quantitative Finance, 10*(9), 1067–1076. http://dx.doi.org/10.1080/14697688.2010.481630.

Covel, M. W. (2009). *Trend following (updated edition): Learn to make millions in up or down markets*. United States of America: Pearson Education LTD.

Esfahanipour, A., & Mousavi, S. (2011). A genetic programming model to generate risk-adjusted technical trading rules in stock markets. *Expert Systems with Applications, 38*(7), 8438–8445. http://dx.doi.org/10.1016/j.eswa.2011.01.039.

Fasanghari, M., & Montazer, G. A. (2010). Design and implementation of fuzzy expert system for Tehran Stock Exchange portfolio recommendation. *Expert Systems with Applications, 37*(9), 6138–6147. http://dx.doi.org/10.1016/j.eswa.2010.02.114.

Fong, S., Si, Y., & Tai, J. (2012). Trend following algorithms in automated derivatives market trading. *Expert Systems with Applications, 39*(13), 11378–11390. http://dx.doi.org/10.1016/j.eswa.2012.03.048.

Fong, S., Tai, J., & Si, Y. W. (2011). Trend following algorithms for technical trading in stock market. *Journal of Emerging Technologies in Web Intelligence, 2*(3), 136–145. http://dx.doi.org/10.4304/jetwi.3.2.136-145.

Gorgulho, A., Neves, R., & Horta, N. (2011). Applying a GA kernel on optimizing technical analysis rules for stock picking and portfolio composition. *Expert Systems with Applications, 38*(11), 14072–14085. http://dx.doi.org/10.1016/j.eswa.2011.04.216.

Holland, J. H. (1976). Adaptation. In R. Rosen & F. M. Snell (Eds.). *Progress in Theorerical Biology*. New York: Plenum (Reprinted).

How, J., Ling, M., & Verhoeven, P. (2009). Does size matter? A genetic programming approach to technical trading. *Quantitative Finance, 10*(2), 131–140. http://dx.doi.org/10.1080/14697680902773629.

Hsu, Y., Chen, A., & Chang, J. (2011). An inter-market arbitrage trading system based on extended classifier systems. *Expert Systems with Applications, 38*(4), 3784–3792. http://dx.doi.org/10.1016/j.eswa.2010.09.039.

Huang, W., Nakamori, Y., & Wang, S. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research, 32*(10), 2513–2522. http://dx.doi.org/10.1016/j.cor.2004.03.016.

James, J. (2003). Simple trend-following strategies in currency trading. *Quantitative Finance, 3*(4), C75–C77. http://dx.doi.org/10.1088/1469-7688/3/4/604.

John, G. H., Miller, P., & Kerber, R. (1996). Stock selection using rule induction. *IEEE Expert, 11*(5), 52–58. http://dx.doi.org/10.1109/64.539017.

Kaucic, M. (2010). Investment using evolutionary learning methods and technical rules. *European Journal of Operational Research, 207*(3), 1717–1727. http://dx.doi.org/10.1016/j.ejor.2010.07.008.

Lai, R. K., Fan, C., Huang, W., & Chang, P. (2009). Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications, 36*, 3761–3773. http://dx.doi.org/10.1016/j.eswa.2008.02.025 (2 Part 2).

Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems, 37*(4), 567–581. http://dx.doi.org/10.1016/S0167-9236(03)00088-5.

Mabu, S., Hirasawa, K., Obayashi, M., & Kuremoto, T. (2013). Enhanced decision making mechanism of rule-based genetic network programming for creating stock trading signals. *Expert Systems with Applications, 40*(16), 6311–6320.

Mehta, K., & Bhattacharyya, S. (2004). Adequacy of training data for evolutionary mining of trading rules. *Decision Support Systems, 37*(4), 461–474. http://dx.doi.org/10.1016/S0167-9236(03)00091-5.

Núñez-Letamendia, L. (2007). Fitting the control parameters of a genetic algorithm: An application to technical trading systems design. *European Journal of Operational Research, 179*(3), 847–868. http://dx.doi.org/10.1016/j.ejor.2005.03.067.

Shen, L., & Loh, H. T. (2004). Applying rough sets to market timing decisions. *Decision Support Systems, 37*(4), 583–597. http://dx.doi.org/10.1016/S0167-9236(03)00089-7.

Szakmary, A. C., Shen, Q., & Sharma, S. C. (2010). Trend-following trading strategies in commodity futures: A re-examination. *Journal of Banking & Finance, 34*(2), 409–426. http://dx.doi.org/10.1016/j.jbankfin.2009.08.004.

Tsang, E., Yung, P., & Li, J. (2004). EDDIE-Automation, a decision support tool for financial forecasting. *Decision Support Systems, 37*(4), 559–565. http://dx.doi.org/10.1016/S0167-9236(03)00087-3.

Tsymbal, A. (2004). The problem of concept drift definitions and related work. Technical Report TCD-CS-2004-15.

Wilson, S. W. (1995). Classifier fitness based on accuracy. *Evolutionary Computation, 3*(2), 149–175. http://dx.doi.org/10.1162/evco.1995.3.2.149.

Wu, M., Lin, S., & Lin, C. (2006). An effective application of decision tree to stock trading. *Expert Systems with Applications, 31*(2), 270–274. http://dx.doi.org/10.1016/j.eswa.2005.09.026.