An automatic stock trading system using Particle Swarm Optimization

Conference Paper · June 2017

DOI: 10.1109/ECTICon.2017.8096283

Chukiat Worasucheep
King Mongkut's University of Technology Thonburi

Sirapop Nuannimnoi
National Chiao Tung University

An Automatic Stock Trading System using Particle Swarm Optimization

Chukiat Worasucheep, Sirapop Nuannimnoi, Ratchanon Khamvichit, Papon Attagonwantana
Applied Computer Science Program, Department of Mathematics
Faculty of Science, King Mongkut's University of Technology Thonburi, Thailand

Abstract—**This paper proposes a trading strategy based on a learning method to combine a set of technical trading signals. The learning employs a modified Particle Swarm Optimization to optimize the weights of signals. The set of weighted signals is then used to determine trading decisions, i.e. to buy, to sell or to hold. A trading simulation is conducted using historical daily stock prices of twenty stocks from NYSE and SET markets. The performance is evaluated using the return on investment with the testing subset of such data. The results are compared with buy-and-hold strategy and the signal follow strategy of each individual signal.**

Keywords : Particle Swarm Optimization, Trading strategy, Technical indicators

I. INTRODUCTION

Algorithmic trading has gained a very high interest in the domain of computational finance for several decades. Generally, it is very difficult to for an investor to make trading decisions without financial advices [1]. Recently, in order to support them, many techniques have been proposed and used to either predict prices or discover trading rules based on technical analysis approach. A number of technical indicators are used for generating trading signals (buy, sell or hold) [2]. Unfortunately they often create inconsistent signals for a particular stock. Many computational algorithms have been developed to apply these technical indicators for automatic trading of stocks or other financial data, most of them employ genetic programming, genetic algorithm or similar algorithms [3]-[9].

To resolve the trading signal inconsistencies, this paper proposes a new trading strategy built from signals generated from technical indicators. The proposed algorithm weights a set of signals, each of which can be *to buy*, *to sell*, or *to hold*, based on an evolutionary learning from historical data. The weighted signal set then will be used as trading strategy. The algorithm requires daily closing, opening, highest and lowest prices and volumes of a stock to be traded as inputs. These inputs are used to calculate widely used technical indicators and generate trading signals. Then a modified Particle Swarm Optimization (PSO) algorithm [10] is used to optimize the weights of indicators for an optimal trading strategy. PSO is a well-known swarm intelligence technique that is based on the social behavior of a group of computing agents [11]. Finally, the rates of return from the proposed trading strategy are compared with the ones from the buy-and-hold (B&H) strategy which is widely used by most passive mutual funds, and from each indicator's signal-follow trading strategy.

The remaining of this paper are as follows. Section II reviews some recent trading systems based on technical analysis. Section III describes the proposed algorithm. Section IV explains the experimental setups as well as results and discussion. At last, Section V concludes this paper with some possible future works.

II. LITERATURE REVIEW

Over the past twenty years, many academic research have been conducted to develop or improve technical trading systems. Many of them apply computational intelligence techniques to identify optimal trading rules [3]. A genetic programming is used by Allen and Karjalainen [4] for trading daily S&P500 prices from 1928 to 1995, however without a consistent beat on the buy-and-hold (B&H) strategy. Núñez-Letamendia [5] demonstrates the robustness of genetic algorithms-tuned trading strategy on 25 stocks from the Madrid Stock Exchange. Kwon and Moon [6] presents a hybrid neurogenetic system with a context based ensemble method and tested on 36 stocks in NYSE and NASDAQ, showing on average a notable improvement over the B&H strategy. Massimiliano [3] proposes a multi-objective genetic algorithm, boosting and statistical learning methods together for trading strategy of S&P 500 index. Genetic network programming with sarsa learning is employed for efficient stock trading decision-making using technical indicators and candlestick charts [7]. A multi-objective particle swarm optimization is proposed for end-of-day historical stock trading [8]. The system optimizes the weights of several technical indicators over two objective functions, i.e. percent profit and Sharpe ratio. More recent research survey can be reached from Yong et.al. [9] in 2015 which provided a systematic review on the state-of-the-art application of evolutionary computation techniques for rule discovery in stock algorithmic trading.

III. THE PROPOSED ALGORITHM

This section briefly explains the proposed trading algorithm, namely *PSOTrader*. The required inputs are daily closing prices, open prices, volumes, high prices and low prices of the stock in consideration. The steps of the algorithm are described below.

1. Nine chosen technical trading signals are calculated from the inputs for each historical data record.
2. The vectors of trading signals and closing prices are fed into the optimization process, in which our modified PSO optimizes the weights for all signals.
3. The weighted signals are summed up to determine a trading decision for each day.
4. The return of the investment, as a percentage of final portfolio value per initial portfolio value, is then calculated for performance comparison.

A. Trading Signals from Technical Indicators

There are many technical indicators developed and widely used, but they often provide inconsistent trading signals. In this work, the indicators below are considered for the creation of trading signal vectors [2], which the algorithm uses to calculate the weights for each indicator's signal.

978-1-5386-0449-6/17/$31.00

1. Simple Moving Average (SMA). Here, we use 100 days moving average. When the closing price moves above the 100-day SMA, a *buy* signal is generated. A *sell* signal is given when price moves below the SMA line.
2. Two moving averages are commonly used for "crossover" buy or sell indications. Three pairs of (20, 50), (50, 100), and (20, 100) are chosen to be our crossovers. The *buy* signal happens when the shorter line moves above the longer line, and the *sell* signal occurs when vice versa.
3. Moving Average Convergence Divergence (MACD). The *sell* signal occurs when the MACD signal line moves above the zero line, while the *buy* signal occurs when the line moves below the zero line.
4. Relative Strength Index (RSI) value ranges in 0 to 100. Movements above 70 are considered overbought and provides *sell* signal, while an oversold condition giving a *buy* signal would be a move below 30.
5. Stochastic Oscillator (STO) help identify trend velocity and movement. *Sell* signal occurs when the %D line moves above value 80, while the *buy* signal occurs when the %D moves below value 20.
6. Commodities Channel Index (CCI) identifies new trend or detect risk condition. CCI over +100 considers a *sell* signal while CCI below -100 considers a *buy* signal.
7. William Percent Range (%R) identifies the reversal point. Readings over 80 or under 20 identify overbought (*sell* signal) and oversold (*buy* signal), respectively.

B. Optimization with PSO

Particle Swarm Optimization (PSO) is a swarm intelligence technique developed for solving an optimization problem of real numbers [10]. A swarm consists of a set of particles, each of which represents a possible solution. Each particle consists of a position vector $x_i = (x_{i1}, x_{i2},..., x_{iD})$ and a velocity vector $v_i = (v_{i1}, v_{i2},..., v_{iD})$ where D is dimension or the number of decision variables to be optimized. PSO algorithm searches for the optimal solution in an iterative manner. In a common practice, in each iteration the velocity of each particle is updated by using the latest velocity, the cognitive information of the particle and the social information from the whole swarm. Then the updated velocity is used for changing the position of such particle. This iterative updates move the swarm to the optimal solution of the problem. Algorithmic details of PSO can be obtained from literature [10][11][12].

For this problem, the vector of weights for trading signals is represented in the PSO as the position vector of each particle. In this work, we use PSO to search for optimal values of the weights from the decision equation shown below.

$$Decision_d = (w_1 s_1 + w_2 s_2 + \cdots + w_n s_n) / \sum_i^n w_i \quad \text{.... (1)}$$

where trading signals $s_1, s_2, \ldots, s_n$ are generated from each indicator as described in subsection A. $w_1, w_2, \ldots, w_n$ are weights for each corresponding signal and n is the number of trading signals in use. These weights are to be optimized by a modified PSO and thus they are the decision variables of the optimization problem. Each $Decision_d$ determines the trading decision for day d, and its meaning is described in the next subsection. The PSO used in this paper was modified from a version with time-varying acceleration coefficients proposed in [12]. In that version of PSO, the cognitive coefficient (c_1) and the social coefficient (c_2) were set to vary linearly from the beginning to the end. Parameter inertia weight factor also decreases linearly with time.

C. Trading Decision

During the simulation process, the $Decision_d$ from equation (1) is used to determine whether to buy, to sell or to hold the particular stock on that day d with the conditions below.

- If $Decision_d$ is higher than t_d, then buy.
- If $Decision_d$ is lower than $-t_d$, then sell.
- If $Decision_d$ is in the range of $(-t_d, t_d)$, hold the position.

Parameter t_d is the *decision threshold* value for making buy/sell decision and $t_d \in$ (-0.1, 1.0). If t_d approaches 1.0, the algorithm suggests buying. In contrast, if t_d approaches -1.0, the algorithm suggests selling.

Later in the end of each simulation process, the return of investment calculated from *return* = (*final portfolio value* - *initial portfolio value*) / *initial portfolio value* is used as PSO's objective function value. That is the goal is to find the optimal set of weights for the particular stock that gives the highest return on investment during the considered period.

IV. EXPERIMENTATION

To verify the performance of the proposed algorithm, ten highly active stocks from different industries in NYSE, and ten stocks from The Stock Exchange of Thailand (SET) are used in an experiment. Historical data of daily stock prices from 2015 to 2016 (about 505 days or records) are downloaded from *finance.yahoo.com* for used in the simulation. These 20 stocks have different trends within the tested period; some are sideway, while others are downward or upward.

A. Parameters

In this experiment, parameters of PSO are set as widely used. c_1 and c_2 are set to be changed through time from 2.5 to 0.5, and from 0.5 to 2.5, respectively [12]. The inertia weight, $i\omega$, is decreased linearly from 0.9 to 0.4 for more exploitation ability at the beginning, and then more exploration ability at the end. We set the number of particles at 20 and 25 to investigate the difference, and the maximum number of allowable objective function calls at 150,000 per run as stopping criterion.

B. Performance Comparison

After the algorithm learns from the training dataset, the set of optimized weights are used for trading with the testing dataset. Then the return on investment during the testing period is calculated as described in subsection III.C. The obtained return on investment is compared to those from the simple buy-and-hold strategy (B&H), and from the signal-follow strategy for each individual indicator's. The signal-follow strategy means that trading decision (buy, sell or hold) is determined by the only indicator in consideration. Here in this experiment, we compare with the following most common trading signals. SMA100, EMA20/50, EMA20/100, EMA50/100, MACD, RSI, STO, CCI and Williams %R. The buy and sell signals are determined as discusses in subsection III.A.

The experiment also studies the effects of some parameters in the proposed *PSOTrader* algorithm. Swarm size of PSO generally has some effects on the optimization result and is normally set between 20 and 30. So we tests the swarm size (n) of both 20 and 25. From subsection III.C, parameter t_d (decision threshold) affects the frequency of trading. More frequent trading may suffer higher cost of trading commission fee. In this experiment, we study 3 different values of t_d: 0.1, 0.3 and 0.5. The initial investment is set to 1,000,000 baht for simulating each stock. The trading commission fee is set at 0.2% of every trading amount of both buying and selling orders. For each buying order, a quarter amount of the whole money is used to buy the stock. For each selling order, a quarter volume of stocks (if bought using the whole amount) is sold. This follows the basic concept of gradual investment, not buying the stock in just one time using all the money.

For each stock, proportion of training data subset and testing data subset is 90:10, meaning that the first 90% of dataset is used for modeling the PSO trading equation (1), which is then used for testing the performance with the remaining 10% dataset. For other trading strategies, the trading signals are calculated for the whole dataset, but the trading is simulated only with the 10% testing subset, for a fair comparison.

C. Results and Discussions

Table I and Table II report returns on investments for stocks in NYSE and SET respectively, from trading by using each strategy. The average returns (for all stocks) are summarized in Table III for the sake of clarity. Excluding our proposed algorithm, SMA100 signal provides the lowest returns on average while Williams %R provides the highest returns on average. Thus also shown in Table I and Table II are the numbers of buying and selling executed by using SMA100 and Williams %R signals as well as the proposed *PSOTrader* algorithm, for investigation.

As seen in the tables, trading signals are not reliable to every stock. In some stocks, some trading signals work well, whereas in other stocks, those trading signals underperform. The performance of most trading signals for NYSE stocks and for SET stocks are mixed.

From Table III, on average, B&H strategy performs badly (-0.95%), but SMA100 performs the worst (-1.01%) for this set of tested stocks. The best performer is our *PSOTrader* with swarm size = 25 and t_d = 0.1, yielding a profit of 3.81%. A simulation with swarm size = 25 works better than 20 for every t_d value. The lower t_d value, the more frequently the algorithm trades (buys and sells) as seen in Table I. This takes more chance of trading that is enough to earn a higher profit, as seen in Table III.

However a high frequency of trading does not guarantee a high profitability. Look at the cases of SMA100 and Williams %R. SMA100 makes more frequent trading than Williams %R does, but SMA100 gives less returns. The reason should be that both employ only one technical signal which is not reliable. In contrast, the trading decision in our *PSOTrader* depends upon a combined set of technical signals rather than only one signal. The proposed strategy successfully learns the optimal weights for each component signal. The obtained weighted signal thus provides a more efficient trading strategy and results in the highest profit on average.

V. CONCLUSION AND FUTURE WORKS

Many technical trading signals are not very reliable and sometimes even conflicting. This paper proposes a trading strategy that learns to follow an optimally-weighted set of trading signals instead of following only one signal. The optimal weights are obtained from a learning using a modified PSO algorithm. Then, the decision to either buy, sell or hold the current volume of stocks is determined using those weighted signals. The experimental results demonstrate that the proposed algorithm clearly outperforms both buy-and-hold strategy and all individual-signal following strategies in terms of return on investment.

Possible directions for future works include adaptive setting of the buying and selling proportion for each trading, further study of other technical signals for a more profitable trading decision, and improved optimization algorithms for optimizing the weights. Risk on investment such as the Sharpe ratio or the maximum drawdown, defined as the maximum loss from a peak to a trough of a portfolio before a new peak is attained, should also be considered in the experiment. Also different trading period length (other than 2 years) should be tested.

REFERENCES

[1] J.M. Berutich, F. Lopez, F. Luna, and D. Quintana. "Robust Technical Trading Strategies Using GP for Algorithmic Portfolio Selection," Expert Systems with Applications, vol. 46, 2016, pp. 307-315.

[2] S. Leigh. Essential Technical Analysis: Tools and Techniques to Spot Market Trends. New York, NY: Wiley, 2002.

[3] K. Massimiliano, "Investment using evolutionary learning methods and technical rules," European Journal of Operational Research, vol. 207, December 2010, pp. 1717-1727.

[4] F. Allen, R. Karjalainen, "Using genetic algorithms to find technical trading rules," Journal of Financial Economics, vol. 51, pp. 245–271.

[5] L. Núñez-Letamendia, "Fitting the control parameters of a genetic algorithm: an application to technical trading systems design," European Journal of Operational Research, vol. 179, pp. 847–868.

[6] Y.K. Kwon, B.R. Moon, "A hybrid neurogenetic approach for stock forecasting," IEEE Transactions on Neural Networks, vol. 18, 2007, pp. 851–864.

[7] Y. Chen, et al., "A genetic network programming with learning approach for enhanced stock trading model," Expert Syst. Appl., vol. 36, 2009, pp.12537–12546.

[8] A. C. Briza, Prospero C. Naval Jr., "Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data," Applied Soft Computing, vol. 11, January 2011, pp. 1191-1201.

[9] H. Yong, K. Liu, X. Zhang, L. Su, E.w.t. Ngai, and M. Liu. "Application of Evolutionary Computation for Rule Discovery in Stock Algorithmic Trading: A Literature Review." Applied Soft Computing, vol.36, 2015, pp. 534-51.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," In IEEE International Conference on Proceedings Neural Networks, Nov-Dec. 1995, vol. 4, pp. 1942–1948.

[11] Y. del Valle, G.K. Venayagamoorthy, S., Mohagheghi, J.-C. Hernandez, R.G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems," IEEE Trans. on Evol. Comp, vol. 12, pp. 171 – 195, April 2008.

[12] A. Ratnaweera, S.K. Halgamuge, and H.C. Watson. "Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients," IEEE Transactions on Evolutionary Computation, vol. 8, 2004, pp. 240-255.

TABLE I THE RETURNS ON INVESTMENTS FOR THE NYSE STOCKS

Strategy		AAPL	AMZN	CVS	GM	GOOGL	HSBC	KO	PG	TSLA	YHOO
B&H		-1.51%	-8.66%	-9.05%	8.97%	-4.52%	5.83%	-1.80%	-2.10%	4.56%	-9.88%
Signal follow	SMA100	-0.90%	-0.58%	0.13%	-3.16%	-0.09%	1.80%	-1.25%	-0.33%	0.76%	-1.98%
	#Buy/#Sell	9/9	12/11	13/13	14/14	11/11	9/8	14/13	16/15	12/12	12/12
	EMA20/50	0.61%	-0.19%	-2.83%	1.03%	0.10%	1.26%	0.26%	-0.89%	1.14%	0.91%
	EMA20/100	0.36%	-0.13%	-3.10%	1.44%	0.41%	1.22%	0.24%	0.42%	-1.38%	-0.40%
	EMA50/100	0.31%	-0.79%	-2.62%	1.40%	-0.03%	1.02%	0.27%	-0.95%	1.30%	-0.56%
	MACD	0.20%	-2.60%	2.26%	-0.10%	0.12%	-1.72%	0.59%	-0.24%	2.41%	-1.45%
	RSI	3.36%	0.00%	0.00%	0.00%	1.58%	3.70%	0.00%	0.00%	0.00%	-5.03%
	STO	-0.31%	-3.64%	-3.76%	4.23%	3.85%	1.67%	0.43%	-0.78%	4.16%	-2.77%
	CCI	0.00%	-0.90%	0.39%	1.38%	0.00%	0.00%	0.21%	-0.04%	3.34%	0.12%
	Williams %R	0.30%	0.36%	-3.14%	5.70%	2.81%	2.28%	2.21%	2.20%	10.59%	0.76%
	#Buy/#Sell	4/4	5/2	4/4	5/3	5/4	5/2	4/3	5/3	5/5	7/3
PSOTrader0.1 *n* = 20	Training	2.68%	8.35%	0.60%	7.47%	3.43%	0.06%	2.85%	0.26%	7.51%	1.40%
	Testing	8.35%	-0.68%	1.46%	16.23%	4.77%	4.96%	2.46%	-0.66%	0.59%	-4.72%
	#Buy/#Sell	14/14	11/8	11/11	11/11	8/8	11/5	14/13	23/17	14/14	16/11
PSOTrader0.1 *n* = 25	Training	2.13%	18.25%	0.31%	8.30%	3.18%	-2.47%	2.23%	-2.75%	6.96%	0.15%
	Testing	10.18%	-0.40%	-2.07%	3.66%	3.90%	3.70%	3.89%	0.06%	19.38%	-6.62%
	#Buy/#Sell	14/14	14/14	12/12	11/10	11/9	9/3	12/11	9/8	12/12	13/4
PSOTrader0.3 *n* = 20	Training	0.51%	16.33%	0.13%	11.04%	2.92%	0.05%	0.52%	1.51%	4.99%	0.69%
	Testing	2.24%	-0.63%	-1.54%	-0.44%	0.75%	1.22%	0.50%	0.69%	12.72%	0.57%
	#Buy/#Sell	5/2	3/1	4/1	1/1	5/5	2/1	6/3	2/0	6/4	2/1
PSOTrader0.3 *n* = 25	Training	1.67%	0.23%	0.00%	10.06%	2.40%	0.99%	-0.61%	-0.05%	6.07%	1.39%
	Testing	1.50%	-0.73%	-2.50%	-0.52%	1.65%	2.87%	0.16%	1.79%	13.36%	-0.25%
	#Buy/#Sell	3/3	3/3	5/4	2/2	3/3	2/2	4/3	10/6	6/4	3/1
PSOTrader0.5 *n* = 20	Training	0.00%	0.94%	1.21%	7.35%	0.00%	1.16%	-0.54%	1.51%	0.49%	0.00%
	Testing	0.00%	0.06%	-1.38%	2.26%	0.00%	0.00%	-0.01%	0.25%	7.21%	0.00%
	#Buy/#Sell	0/0	3/3	1/0	3/1	0/0	0/0	3/2	1/1	4/2	0/0
PSOTrader0.5 *n* = 25	Training	0.00%	0.00%	0.00%	3.54%	0.00%	-0.57%	-0.61%	1.51%	4.98%	0.00%
	Testing	0.00%	-2.61%	0.00%	0.00%	0.00%	0.24%	-0.03%	0.25%	9.57%	0.00%
	#Buy/#Sell	0/0	1/0	0/0	0/0	0/0	1/1	2/0	1/1	3/1	0/0

TABLE II THE RETURNS ON INVESTMENTS FOR THE SET STOCKS

Strategy		ADVANC	AOT	BANPU	BEM	CPALL	KBANK	KTB	PTT	SCC	TRUE
B&H		-6.45%	2.96%	-0.40%	-3.01%	1.22%	0.17%	-0.40%	5.27%	-2.75%	2.47%
Signal follow	SMA100	-2.02%	-0.57%	-6.13%	0.00%	-2.04%	-0.51%	0.37%	0.18%	-1.62%	-2.28%
	#Buy/#Sell	12/12	11/10	14/13	16/15	13/12	9/9	12/12	12/12	11/11	15/15
	EMA20/50	-1.80%	-1.51%	-2.43%	-0.35%	-0.84%	-0.37%	0.53%	0.01%	-0.85%	0.28%
	EMA20/100	0.00%	-0.60%	0.00%	-0.74%	0.00%	0.44%	0.62%	1.10%	0.27%	-0.23%
	EMA50/100	-1.93%	-0.09%	1.10%	-0.21%	-1.64%	-0.76%	0.29%	-0.09%	-1.91%	0.73%
	MACD	-1.42%	0.88%	-2.93%	-1.72%	0.79%	-0.29%	-2.12%	-1.28%	-0.94%	-2.34%
	RSI	-2.00%	5.98%	2.00%	0.00%	0.00%	4.83%	2.49%	7.77%	0.69%	1.28%
	STO	1.31%	8.29%	2.27%	-0.65%	1.81%	0.32%	2.69%	2.33%	3.95%	0.22%
	CCI	-0.38%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.62%	2.63%	0.00%
	Williams %R	-2.31%	6.13%	3.05%	1.56%	3.79%	2.00%	2.61%	2.70%	2.88%	1.28%
	#Buy/#Sell	4/2	7/5	3/1	5/1	5/3	4/4	5/4	3/3	5/2	3/0
PSOTrader0.1 *n* = 20	Training	1.44%	-0.31%	9.91%	10.57%	-1.37%	1.58%	0.18%	-3.05%	1.81%	-0.43%
	Testing	1.05%	-2.11%	-0.76%	-0.70%	4.83%	7.96%	5.69%	6.95%	6.22%	1.91%
	#Buy/#Sell	16/5	8/6	17/9	11/1	21/15	21/15	19/13	18/18	14/8	12/7
PSOTrader0.1 *n* = 25	Training	1.53%	-2.32%	9.91%	7.55%	1.49%	3.48%	0.44%	-2.07%	1.30%	-3.53%
	Testing	-0.81%	6.58%	-0.76%	-0.98%	3.88%	7.85%	6.31%	8.88%	7.58%	1.88%
	#Buy/#Sell	9/3	12/10	17/9	13/5	16/12	15/10	22/15	17/13	16/8	9/3
PSOTrader0.3 *n* = 20	Training	4.47%	-2.36%	2.13%	9.62%	2.57%	3.56%	1.62%	1.48%	0.00%	2.35%
	Testing	3.70%	3.13%	2.20%	-0.57%	1.56%	2.30%	3.45%	0.69%	2.63%	2.63%
	#Buy/#Sell	7/1	5/2	4/1	10/0	3/1	6/4	7/6	3/3	3/1	6/2
PSOTrader0.3 *n* = 25	Training	2.54%	-3.85%	0.00%	7.86%	2.10%	4.12%	0.88%	1.06%	1.53%	1.37%
	Testing	1.42%	10.61%	2.08%	2.31%	-0.55%	4.96%	3.73%	2.25%	3.32%	2.25%
	#Buy/#Sell	5/1	14/10	2/0	7/1	3/2	6/2	11/8	6/5	12/7	5/0
PSOTrader0.5 *n* = 20	Training	1.16%	0.00%	0.00%	0.00%	3.07%	0.49%	0.00%	0.92%	0.00%	0.00%
	Testing	1.86%	0.00%	0.00%	0.00%	0.15%	0.00%	0.93%	1.98%	0.00%	0.00%
	#Buy/#Sell	3/0	0/0	0/0	0/0	1/0	0/0	1/1	4/3	0/0	0/0
PSOTrader0.5 *n* = 25	Training	1.48%	0.00%	0.00%	-1.28%	4.09%	0.58%	3.49%	1.05%	0.00%	1.81%
	Testing	0.00%	0.00%	0.00%	0.00%	-0.05%	3.01%	4.71%	0.00%	0.66%	1.28%
	#Buy/#Sell	0/0	0/0	0/0	0/0	2/1	3/0	7/4	0/0	1/0	3/0

TABLE III THE AVERAGE RETURNS

Strategy		Average	NYSE Average	SET Average
B&H		-0.95%	-1.82%	-0.09%
Signal follow	SMA100	-1.01%	-0.56%	-1.46%
	EMA20/50	-0.30%	0.14%	-0.73%
	EMA20/100	0.00%	-0.09%	0.08%
	EMA50/100	-0.26%	-0.06%	-0.45%
	MACD	-0.59%	-0.05%	-1.14%
	RSI	1.33%	0.36%	2.30%
	Stoch	1.28%	0.31%	2.25%
	CCI	0.42%	0.45%	0.39%
	Wiiliams %R	**2.39%**	**2.41%**	**2.37%**

Strategy		Average	NYSE Average	SET Average
PSOTrader0.1 *n* = 20	Training	2.75%	3.46%	2.03%
	Testing	**3.19%**	**3.28%**	**3.10%**
PSOTrader0.1 *n* = 25	Training	2.70%	3.63%	1.78%
	Testing	**3.81%**	**3.57%**	**4.04%**
PSOTrader0.3 *n* = 20	Training	3.21%	3.87%	2.54%
	Testing	**1.89%**	**1.61%**	**2.17%**
PSOTrader0.3 *n* = 25	Training	1.99%	2.22%	1.76%
	Testing	**2.49%**	**1.73%**	**3.24%**
PSOTrader0.5 *n* = 20	Training	0.89%	1.21%	0.56%
	Testing	**0.67%**	**0.84%**	**0.49%**
PSOTrader0.5 *n* = 25	Training	1.00%	0.88%	1.12%
	Testing	**0.85%**	**0.74%**	**0.96%**