Trend Recalling Algorithm for Automated Online Trading in Stock Market

Simon Fong, Jackie Tai
Department of Computer and Information Science,University of Macau,Taipa, Macau SAR
Email: ccfong@umac.mo, ma56562@umac.mo

Pit Pichappan
School of Information Systems, Al Imam University,Riyadh, Saudi Arabia
Email: pichappan@dirf.org

***Abstract* —Unlike financial forecasting, a type of mechanical trading technique called Trend Following (TF) doesn't predict any market movement; instead it identifies a trend at early time of the day, and trades automatically afterwards by a pre-defined strategy regardless of the moving market directions during run time. Trend following trading has a long and successful history among speculators. The traditional TF trading method is by human judgment in setting the rules (aka the strategy). Subsequently the TF strategy is executed in pure objective operational manner. Finding the correct strategy at the beginning is crucial in TF. This usually involves human intervention in first identifying a trend, and configuring when to place an order and close it out, when certain conditions are met. In this paper, we presented a new type of TF, namely Trend Recalling algorithm that operates in a totally automated manner. It works by partially matching the current trend with one of the proven successful patterns from the past. Our experiments based on real stock market data show that this algorithm has an edge over the other trend following methods in profitability. The new algorithm is also compared to time-series forecasting type of stock trading, and it can even outperform the best forecasting type in a simulation.**

***Index Terms*—Trend Following Algorithm, Automated Stock Market Trading**

I. INTRODUCTION

Trend following (TF) [1] is a reactive trading method in response to the real-time market situation; it does neither price forecasting nor predicting any market movement. Once a trend is identified, it activates the predefined trading rules and adheres rigidly to the rules until the next prominent trend is detected. Trend following does not guarantee profit every time. Nonetheless over a long period of time it may probably profit by obtaining more gains than loses. Since TF is an objective mechanism that is totally free from human judgment and technical forecasting, the trends and patterns of the underlying data play a primarily influential role in deciding its ultimate performance.

It was already shown in [2] that market fluctuation adversely affects the performance of TF. Although financial cycles are known phenomena it is a controversy whether cycles can be predicted or past values cannot forecast future values because they are random in nature. Nonetheless, we observed that cycles could not be easily predicted, but the abstract patterns of such cycles can be practically recalled and used for simple pattern matching. The formal interpretation of financial cycle (or better known as economic cycle) refers to economy-wide fluctuations in production or economic activity over several months or years. Here we consider it as the cycle that run continuously between bull market and bear market; some people refer this as market cycle (although they are highly correlated). In general a cycle is made of four stages, and these four stages are: "(1) consolidation (2) upward advancement (3) culmination (4) decline" [3]. Despite being termed as cycles, they do not follow a mechanical or predictable periodic pattern. However similar patterns are being observed to be always repeating themselves in the future, just as a question of when, though in approximate shapes. We can anticipate that some exceptional peak (or other particular pattern) of the market trend that happen today, will one day happen again, just like how it did happen in history. For instance, in the "1997 Asian Financial Crisis" [4], the Hang Seng Index in Hong Kong plunged from the top to bottom (in stages 3 to 4); then about ten years later, the scenario repeats itself in the "2008 Financial Crisis" [5] with a similar pattern.

Dow Theory [6] describes the market trend (part of the cycle) as three types of movement. (1) The "primary movement", main movement or primary trend, which can last from a few months to several years. It can be either a bullish or bearish market trend. (2) The "secondary movement", medium trend or intermediate reaction may last from ten days to three months. (3) The "minor movement" or daily swing varies from hours to a day. Primary trend is a part of the cycle, which consist of one or several intermediate reactions and the daily swings are the minor movements that consist of all the detailed movements. Now if we project the previous assumption that the cycle is ever continuously rolling, into the minor daily movement, can we assume the trend that happens today, may also appear some days later in the future?

doi:10.4304/jetwi.4.3.240-251

Here is an example for this assumption; Figure 1 shows 31 (bottom) trend graphs of Hang Seng Index Futures, which are sourced from two different dates. Although they are not exactly the same, in terms of major upwards and downwards trends the two graphs do look alike. This is the underlying concept of our trend recalling trading strategies that are based on searching for similar patterns from the past. This concept is valid for TF because TF works by smoothing out the averages of the time series. Minor fluctuations or jitters along the trend are averaged out. This is important because TF is known to work well on major trending cycles aka major outlines of the market trend.

The paper is structured as follow: Details of the trend recalling algorithm are presented in Section 2, step by step. Simulation experiments are carried out for evaluating the performance of the Trend Recalling algorithm in automated trading in Section 3. In particular, we compare the Trend Recalling algorithm with a selected time series forecasting algorithm. Section 4 concludes the paper.

Figure 1. Intra-day of 2009-12-07 and 2008-01-31 day trend graphs.

II. Recalling Past Trends

An improved version of Trend Following algorithm called Trend Recalling is proposed in this paper, which looks back to the past for reference for selecting the best trading strategy. It works exactly like TF except that the trading rules are borrowed from one of the best performing past trends that matches most of the current trend. The design of a TF system is grounded on the rules that are summarized by Michael W. Covel, into the following five questions [7]:

1. How does the system determine what market to buy or sell at any time?
2. How does the system determine how much of a market to buy or sell at any time?
3. How does the system determine when you buy or sell a market?
4. How does the system determine when you get out of a losing position?
5. How does the system determine when you get out of a winning position?

There is no standard answer to the above questions; likewise there exists no definite guideline for how the trading rules in TF should be implemented. The first and second questions are already answered in our previous respectively two intra-day 2009-12-07 (top) and 2008-01-works [1][2]. The third question is rather challenging, that is actually the core decision maker in the TF system and where the key factor in making profit is; questions 4 and 5 are related to it. Suppose that we have found a way to identify trend signal to buy or sell, and we have a position opened. Now if the system along the way identifies another trend signal, which complies with the current opened position direction, then we should keep it open, since it suggested that the trend is not yet over. However, if it is counter to the current position, we should probably get a close out, regardless whether you are currently winning or losing, as it indicates a trend reversion.

Our improved TF algorithm is designed to answer this question: when to buy or sell. The clue is derived from the past most similar trend. It is a fact that financial cycles do exist, and it is hypothesized that a trend on a particular day from the past could happen again some days later. This assumption supports the Trend Recalling trading mechanism, which is the basic driving force that our improved trend following algorithm relies on. The idea is expressed as a process diagram in Figure 2. As it can be seen in the diagram there are four major processes for decision making. Namely they are Pre-processing, Selection, Verification and Analysis. Figure 2 shows the process of which our improved TF model works by recalling a trading strategy that used to perform well in the past by matching the current shape of the pattern to that of the old time. A handful of such patterns and corresponding trading strategies are short-listed; one strategy is picked from the list after thorough verification and analysis.

A. Pre-processing

In this step, raw historical data that are collected from the past market are archived into a pool of samples. The pool size is chosen arbitrarily by the user. Five years data were archived in the database in our case. A sample is a day trend from the past with the corresponding trading strategy attached. The trend is like an index pattern for locating the winning trading strategy that is in the format of a sequence of buy and sell decisions. Good trading strategy is one that used to maximize profit in the past given the specific market trend pattern. This past pattern, which is deemed to be similar to the current market trend, is now serving as a guidance to locate the strategy to be applied for decision making during the current market trade session.

Since the past day trend that yielded a great profit before, reusing it can almost guarantee a perfect trading strategy that is superior to human judgment or a complex time series forecasting algorithm. The past samples are referenced by best trading strategies on an indicator that we name it as "*EDM*" (exponential divergence in movement). *EDM* is a crisp value indicator that is based on two moving average differences.

$$EDM_{(t)} = f\left(EMA_{s(t)} - EMA_{l(t)}\right), \tag{1}$$

$$EMA_{(t)} = \left(price_{(t)} - EMA_{(t-1)} \times \frac{2}{n+1}\right) + EMA_{(t-1)} \tag{2}$$

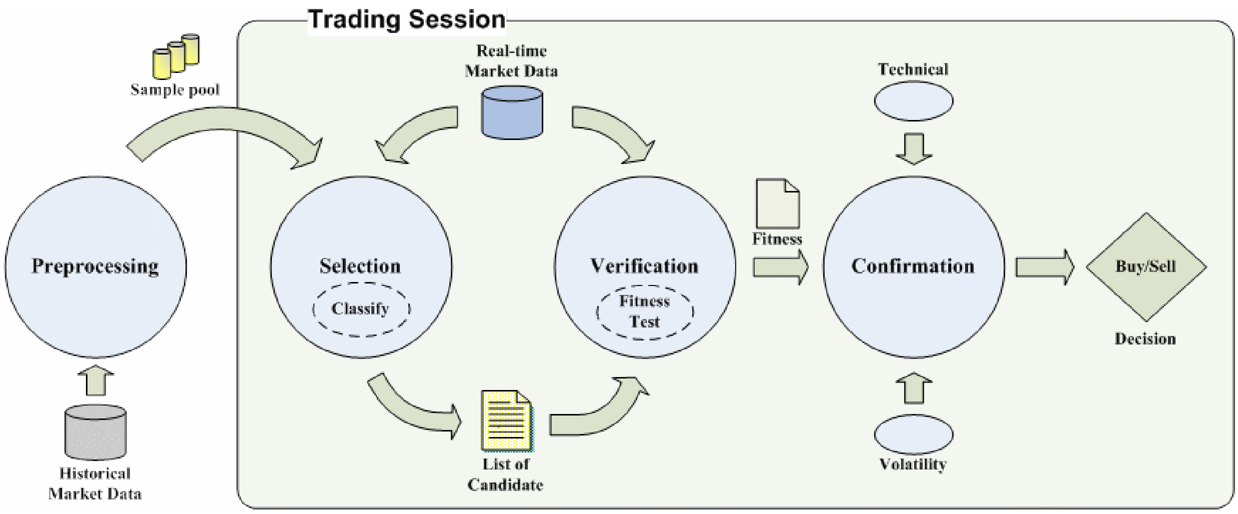

Figure 2. Improved TF process with Recalling function.

Where *price*(t) is the current price at any given time t, n is the number of periods, s denotes a shorter period of Exponential Moving Average, *EMA*(t) at time t, l represents a longer period *EMA*(t), $f(.)$ is a function for generating the crisp result. The indicator sculpts the trend; and based on this information, a TF program finds a list of best trading strategies, which can potentially generate high profit. The following diagram in Figure 3 is an example of pre-processing a trend dated on 2009-12-07 that shows the *EDM*. As indicated from the diagram the program first found a long position at 10:00 followed by a short position at around 10:25, then a long position at 11:25, finally a short position around 13:51 and closes it out at the end of the day, which reaps a total of 634 index points. Each index point is equivalent to \$50 Hong Kong dollars (KKD). In Hong Kong stock market, there is a two hours break between morning and afternoon sessions. To avoid this discontinuation on the chart, we shift the time backward, and joined these two sessions into one, so 13:15 is equivalent to 15:15.

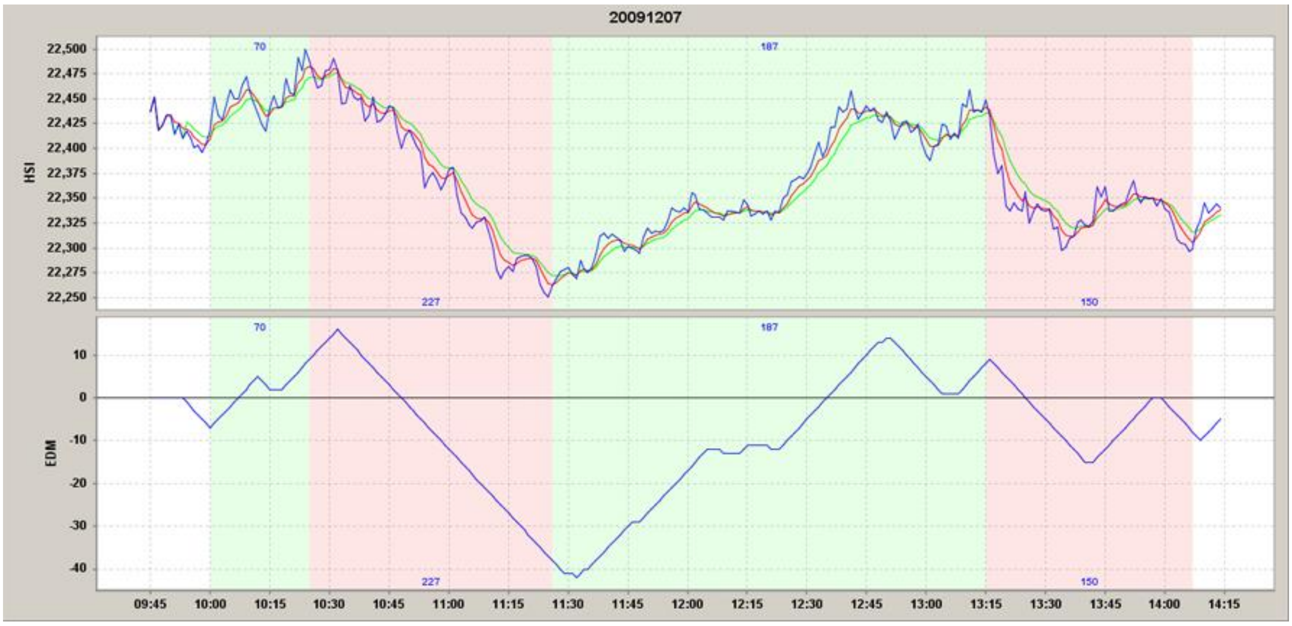

Figure 3. Example of EDM and preprocessed trend of 2009-12-07.

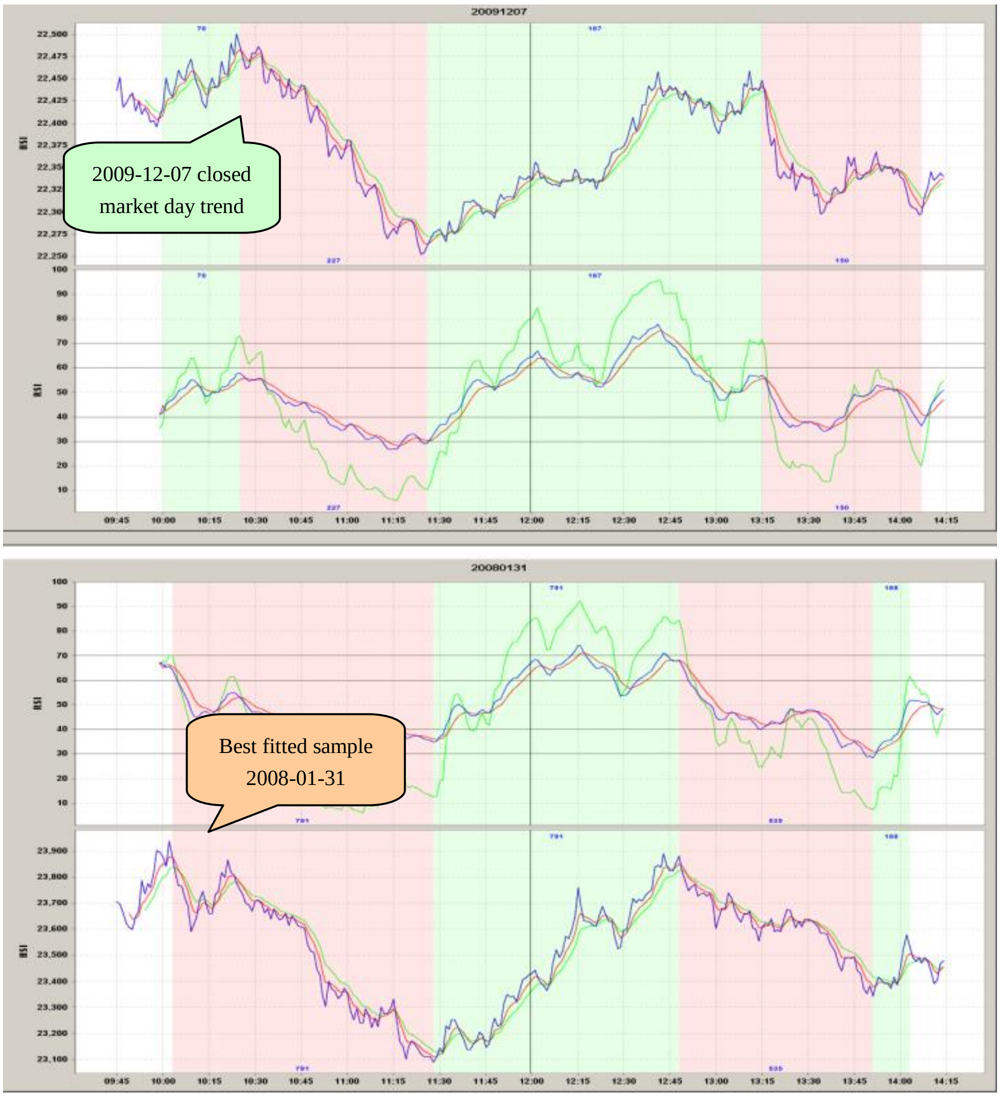

Figure 4. Example of 2009-12-07 sample (above) and its corresponding best fitness (2008-01-31) day trend and RSI graph (below).

B. Selection

Once a pool of samples reached a substantial size, the Trend Recalling mechanism is ready to use. The stored past samples are searched and the matching ones are short-listed. The goal of this selection process is to find the most similar samples from the pool, which will be used as a guideline in the forthcoming trading session. A foremost technical challenge is that no two trends are exactly the same, as they do differ from day to day as the market fluctuates in all different manners. Secondly, even two sample day trends look similar but their price ranges can usually be quite different. With consideration of these challenges, it implies that the sample cannot be compared directly value to value and by every interval for a precise match. Some normalization is necessary for enabling some rough matches. Furthermore the comparison should allow certain level of fuzziness. Hence each sample trend should be converted into a normalized graph, and by comparing their rough edges and measure the difference, it is possible to quantitatively derive a numeric list of similarities. In pattern recognition, the shape of an image can be converted to an outline like a wire-frame by using some image processing algorithm. The same type of algorithm is used here for extracting features from the trend line samples for quick comparison during a TF trading process.

In our algorithm, each sample is first converted into a normalized graph, by calculating their technical indicators data. A popular indicator Relative Strength Index (RSI) has a limited value range (from 1 to 100), which is suitable for fast comparison, and they are sufficient to reflect the shape of a trend. In other words, these indicators help to normalize each trend sample into a simple 2D line graph. We can then simply compare each of their differences of shapes by superimposing these line graphs on top of each other for estimating the differences. This approach produces a hierarchical similarity list, such that we can get around with the inexact matching problem and allows a certain level of fuzziness without losing their similarity attributes. Figure 4 shows an example of two similar sample trend graphs with the RSI displayed. The blue line is the original market trend, red line is the moving average and green line is the RSI.

C. Verification

During this process, each candidate from the list will be tested against the current market state. Ranking from the top smallest number as the most similar, they will be passed through fitness test. Each trend sample corresponds to a specific trading strategy (that was already established in the pre-processing step). Each trading strategy will be extracted and evaluated against historical data. The strategy is then tested on how well it performed as a trial. Each of their performances will be recorded. The trial performance will be used as a criterion to rearrange the list. Here we have an example before and after the fitness test, which was run on the 2009-12-07 during the middle of simulated trade session. The comparison is done based solely on the indicator EDM of the moving market price.

Verification is needed because the selection of these candidates is by a best effort approach. That is because the current and past market situations may still differ to certain extent.

D. Confirmation

After the verification process is done, the candidate list is re-sorted according to the fitness test results. The fittest one will be used as the reference of subsequent trading strategy during the TF decision making. In order to further improve the performance on top of the referencing to the past best strategy, some technical analysis is suggested to be referenced as well. By the advice of Richard L. Weissman from his book [8], the two-moving average crossover system should be used as a signal confirmation. Cross-over means a rise on the market price starts to emerge; it must cross over to its averaged trend. The two-moving average crossover system entails the rise of a second, shorter-term moving average. Instead of using simple moving average, however, EMA - exponential moving average with RSI should be used, that is a short-term RSI EMA and a long-term RSI EMA crossover proceed with the trading only when the market fluctuation is neither too high nor too low. When the reading of this fuzzy system is too high or too low, the system closes out the position then. There are many ways to calculate volatility; the most common one is finding the standard deviation of an asset closing price over a year. The central concept of volatility is finding the amount of changes or variances in the price of the asset over a period of time. So, we can measure market volatility simply by the following equation:

$$Volatility_{(t)} = SMA_n\left(\left(\ln(price_{(t)}) - \ln(price_{(t-1)})\right) \times C\right) \quad (3),$$

$$SMA_{(t)} = \frac{Close_{(t)} + Close_{(t-1)} + ...Close_{(t-n+1)}}{n} \quad (4)$$

Where ln(.) is a natural logarithm, n is the number of periods, *t* is the current time, *C* is a constant that enlarges the digit to a significant figure. SMA is Simple Moving Average that is the average stock price over a certain period of time. By observing how the equation responds to historical data, we can find the maximum volatility as ±15 percent. Base on the previous fluctuation test result, we can define it as the following fuzzy membership.

system. When a changing trend is confirmed and it appears as a good trading signal, the crossover system must also be referenced and check if it gives a consistent signal. Otherwise the potential change in trend is considered as a false signal or intermittent noise. For example in our case the trading strategy from the recalled sample hints a long position trade. We check if RSI crossover system shows a short-term EMA crossing over its long-term EMA or not.

List of Candidate

Rank	Sample
0	2008-01-25
1	2008-03-07
2	2009-08-18
3	2008-01-23
4	2008-11-10
5	2007-08-29
6	2008-04-16
7	2007-09-13
8	2008-12-15
9	2008-02-01
10	2009-11-25
11	2008-01-31
12	2008-02-21
14	2007-12-05
15	2009-03-17

List After Fitness Test

Sample	Fitness	Rank
2009-03-17	373	15
2007-09-13	357	7
2008-12-15	341	8
2008-02-01	339	9
2008-01-31	326	11
2007-12-05	305	14
2008-01-25	300	0
2008-02-21	296	12
2009-08-18	294	2
2008-01-23	273	3
2008-11-10	270	4
2007-08-29	179	5
2009-11-25	165	10
2008-03-07	143	1
2008-04-16	3	6

Figure 5. Fitness test applied on 2009-12-07 at the time 14:47.

In addition to validating the hinted trading signals from past strategy, market volatility should be considered during decision making. It was found in our previous work [2] that the performance of TF is affected mostly by the market fluctuation. It resulted in losses because frequent wrong trading actions were made by the TF rules when the market fluctuates too often. The market fluctuation is fuzzified as a fuzzy volatility indicator. This fuzzy volatility indicator is embedded in the TF mechanism is to automatically monitor the volatility, and percent. Base on the previous fluctuation test result, we can define it as the following fuzzy membership.

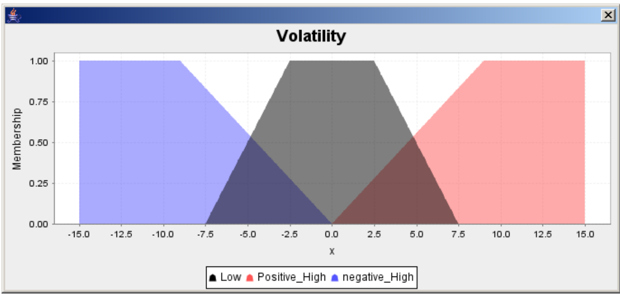

Figure 6. Fitness test applied on 2009-12-07 at the time 14:47.

During the trading session, volatility will be constantly referenced while the following rules apply at the TF system:

IF volatility is too positive high and long position is opened THEN close it out
IF volatility is too positive high and no position is opened THEN open short position
IF volatility is too low THEN do nothing

IF volatility is too negative high and short position is opened THEN close it out
IF volatility is too negative high and no position is opened THEN open long position

These rules have a higher priority over the trade strategies, such that when the condition has met any of these rules, it will take over the control regardless of what decision that the trade strategies has made. In other words other conditions are not considered but only the volatility factor. The four processes are summarized as pseudo codes shown in the Appendix. Though the model is generic, which should be able to work on any market with varying patterns, a new pool sample is recommended to be created for different market as in the Pre-Processing Section.

III. EXPERIMENTS

Two experiments are conducted in this project. One is for verifying the efficacy of Trend Recalling algorithm in a simulated automated trading system. The other is to compare the performances in terms of profitability yielded by Trend Recalling algorithm and time series forecasting algorithms. The objective of the experiments is to investigate the feasibility of Trend Recalling algorithm in automated trading environment as an alternative to time-series forecasting.

A. Performance of Trend Recalling in Automated Trading

The improved TF algorithm with Trend Recalling function is programmed into an automated trading simulator, in JAVA. A simplified diagram of the prototype is shown in Figure 7. It essentially is an automated system, which adopts trading algorithms for deciding when to buy and sell based on predefined rules and the current market trend. The system interfaces with certain application plug-ins that instruct an online-broker to trade in an open market. The trading interval is per minute. Two sets of data are used for the experiment for avoiding bias in data selection. One set is market data of Hang Seng Index Futures collected during the year of 2010, the other one is H-Share also during the same year. They are basically time-series that have two attributes: timestamp and price. Their prices move and the records get updated in every minute. The two datasets however share the same temporal format and the same length, with identical market start and end times for fair comparisons of the algorithms. The past patterns stored in the data base are collected from the past 2.5 years for the use by Trend Recalling algorithm. All trials of simulations are run and the corresponding trading strategies are decided by the automated trading on the fly. At the end of the day, a trade is concluded by measuring the profit or loss that the system has made. The overall performance of the algorithms is the sum of profit/loss averaged by the number of days. In the simulation each trade is calculated in the unit of index point, each index point is equivalent to HKD 50, which is subject to overhead cost as defined by Interactive Broker unbundled commission scheme at HKD 19.3 per trade. The Return-of-Investment (ROI) is the prime performance index that is based on Hong Kong Exchange current Initial margin requirement (each contract HKD 7400 in year of 2010).

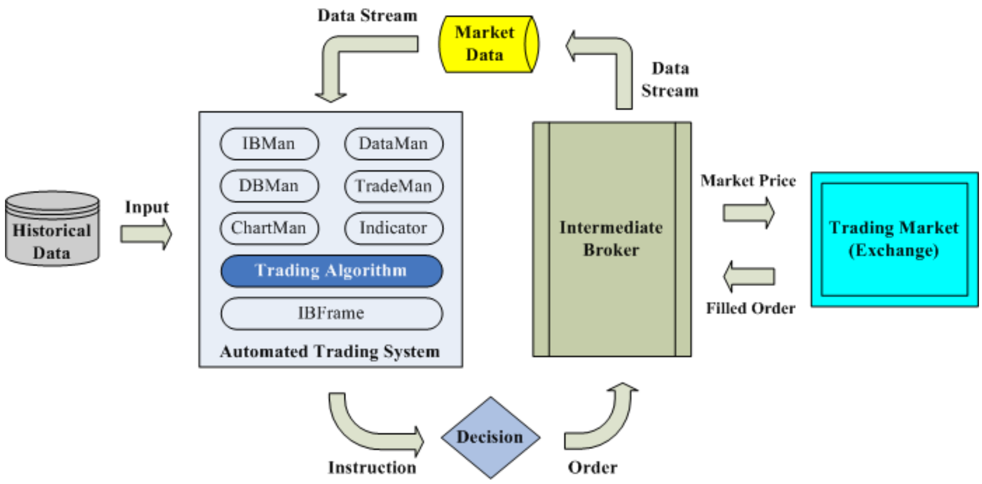

Figure 7. A prototype diagram that shows the trading algorithms is the core in the system.

A time-sequence illustration is shown in Figure 8 that depicts the essential 'incubation' period required prior to the start of trading. The timings are chosen arbitrarily. However, sufficient time (e.g., 30 minutes was chosen in our experiment) should be allowed since the beginning of the market for RSI to be calculated. Subsequently another buffer period of time followed by the calculation of the first RSI_0 would be required for growing the initial trend pattern to be used for matching. If this initial part of the trend pattern is too short, the following trading by the Trend Recalling algorithm may not work effectively because of inaccurate matching by short patterns. If it is stalled for too long for accumulating a long matching pattern, it would be late for catching up with potential trading opportunities for the rest of the day. The stock market is assumed to operate on a daily basis. A fresh trade is started from the beginning of each day. In our case, we chose to wait for 30 minutes between the time when

RSI_0 is calculated and when the trading by Trend Recalling started. The Trend Recalling steps repeat in every interval that periodically guides the buying or selling actions in the automatic trading system. As time progresses the matching pattern lengthens, matching would increasingly become more accurate and the advices for trading actions become more reliable. In our simulation we found that the whole process by Trend Recalling algorithm that includes fetching samples from the database, matching and deciding the trading actions etc., consumes a small amount of running time. In average, it takes only 463.44 milliseconds to complete a trading decision with standard deviation of 45.16; the experiment was run on a PC with a CPU of Xeon QC X3430 and 4Gb RAM, Windows XP SP3 operating system.

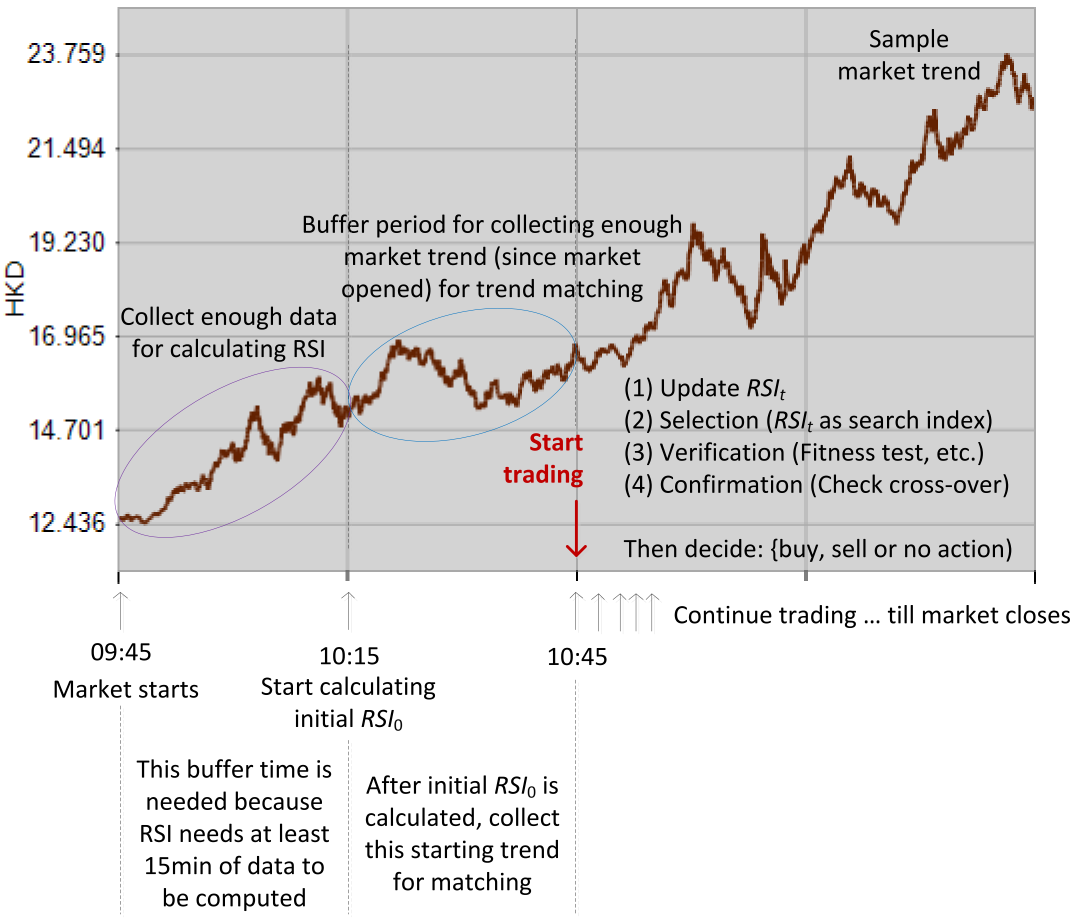

Figure 8. Illustration of the incubation period in market trading by Trend Recalling.

The simulation results are shown in Table 1 and Table 2 respectively, for running the trading systems with different TF algorithms over Hang Seng Index Futures data and H-Share data. The Static TF is one that has predefined thresholds *P* and *Q* whose values do not change throughout the whole trade. *P* and *Q* are the bars when over which the current price goes beyond, the system will automatically sell and buy respectively. Dynamic TF allows the values of the bars to be changed. Fuzzy TF essentially fuzzifies these bars, and FuzzyVix fuzzifies both the bars and the volatility of the market price. Readers who are interested in the full details can refer to [2]. The Tables show the performance figures in terms of ROI, profits and losses and the error rates. Overhead costs per trade are taken into account for calculating profits. The error rate is the frequency or the percentage of times the TF system made a wrong move that incurs a loss. As we observed from both Tables, more than 400% increase in ROI by Trend Recalling algorithm is achieved at the end of the experimental runs. This is a significant result as it implies the proposed algorithm can reap more than four folds of whatever the initial investment is, annually. The trading pattern of Trend Recalling algorithm is shown in Figure 9 for Hang Seng Index Futures data and Figure 10 for H-Share. The same simulation parameters are used by default. The trading pattern of Trend Recalling algorithm is compared to that of other TF algorithms proposed earlier by the authors. Readers who are interested in the other TF algorithms can refer to [2][3] for details. From the Figures, the trading performance by Trend Recalling strategy is always winning and keeps improving in a long run. Figure 11 shows a longitudinal view of trading results over a day; one can see that TF does not guarantee profits at all times, but overall there are more profits than losses.

TABLE I.

PERFORMANCE OF ALL TF TRADING ALGORITHMS ON HANG SENG INDEX FUTURES 2010

Simulation	Static	Dynamic	Fuzzy	FuzzyVix	Recalling
Total Index Point	1115	1142	1476	2981	7201
Net Worth (HKD)	55750	57100	73800	149050	360050
Total Trade	1088	987	1578	1149	1186
Cost	41997	38098	60911	44351	45780
Error (%)	44.12	40.36	36.99	37.01	35.11
P&L	13753	19002	12889	104699	314270
Total ROI	19%	26%	17%	141%	425%

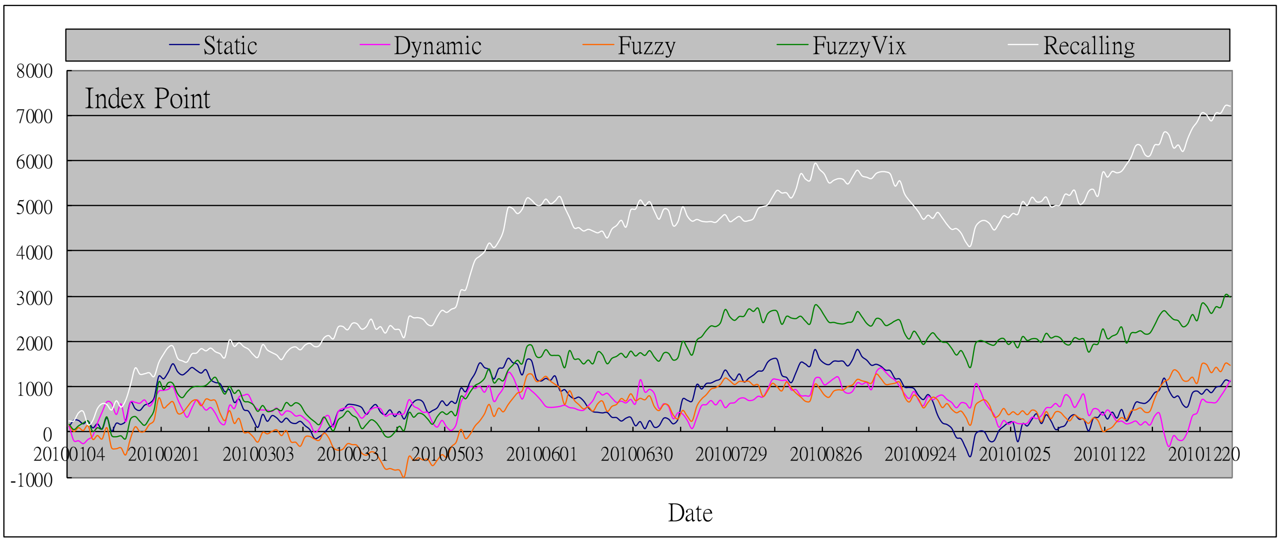

Figure 9. Simulation of all TF trading algorithms on Hang Seng Index Futures during 2010.

TABLE II.

PERFORMANCE OF ALL TF TRADING ALGORITHMS ON H-SHARE 2010

Simulation	Static	Dynamic	Fuzzy	FuzzyVix	Recalling
Total Index Point	917	1471	1926	2606	4816
Net Worth (HKD)	45850	73550	96300	130300	240800
Total Trade	835	1408	1021	1068	1140
Cost	32231	54349	39411	41225	44004
Error (%)	48.53	42.73	41.07	40.23	35.36
P&L	13619	19201	56889	89075	196796
Monthly ROI	2.70%	3.81%	11.29%	17.67%	39.05%
ROI	32%	46%	135%	212%	469%

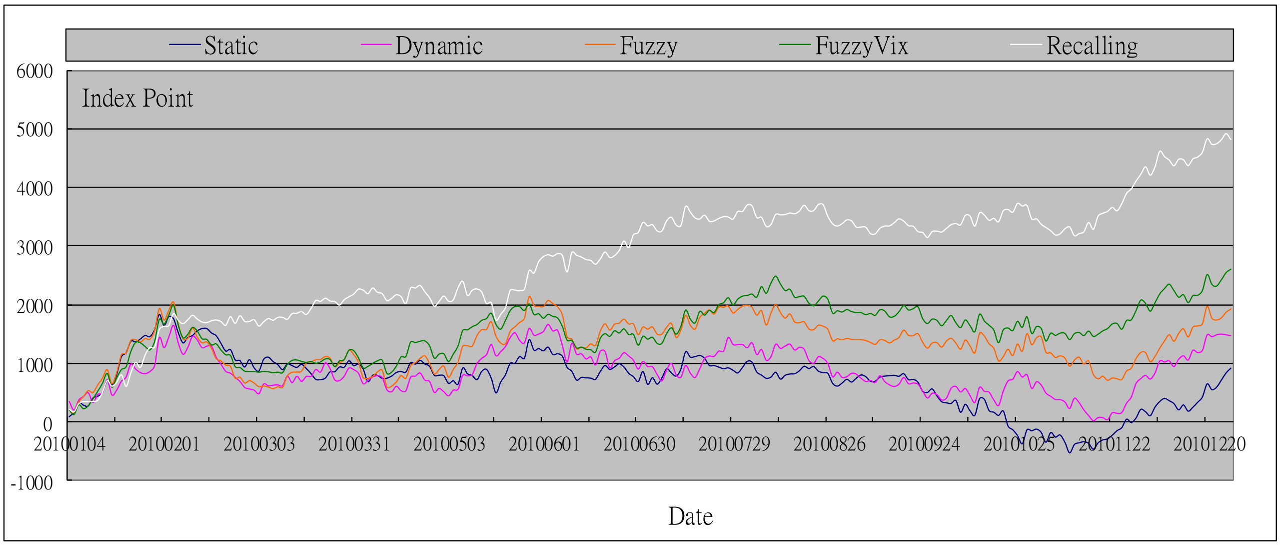

Figure 10. Simulation of all TF trading algorithms on H-Share during 2010.

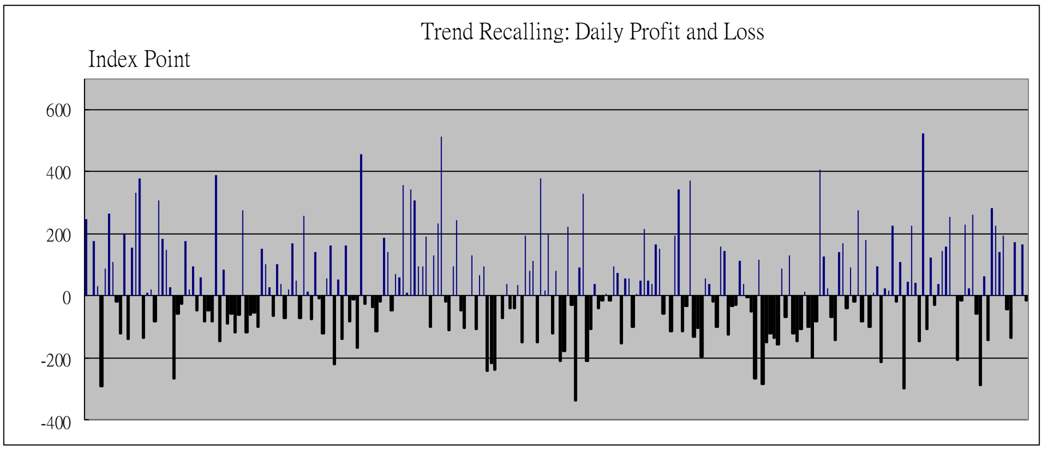

Figure 11. Profit and loss diagram.

B. Comparison of Trend Recalling and Time Series Forecasting

Time series forecasting (TSF) is another popular technique for stock market trading by mining over the former part of the trend in order to predicting the trend of near-future. The major difference between TSF and TF is that, TSF focuses on the current movements of the trend with no regard to history, and TSF regresses over a set of past observations collected over time. Some people may distinguish them as predictive and reactive types of trading algorithms. Though the reactive type of algorithms have not been widely studied in research community, there are many predictive types of time series forecasting models available, such as stationary model, trend model, linear trend model, regression model, etc. Some advance even combined neural network with TSF [9].

In our experiment here, we want to compare the working performance of TSF and TF, which is represented by its best performer so far – Trend Recalling algorithm. For a fair comparison, both types of algorithms would operate over the same dataset, which is the Hang Seng Index Futures. We simulate their operations and trading results over a year, under the same conditions, and compare the level of profits each of them can achieve. The profit or loss for each trade would be recorded down, and then compute an average return-of-investment (ROI) out of them. ROI will then be the common performance indicator for the two competing algorithms. It is assumed that ROI is of prime interest here though there may be other technical performance indictors available for evaluating a trading algorithm [10]. For examples, Need to Finish, Price Sensitivity, Risk Tolerance, Frequency of Trade Signals and Algorithmic Trading Costs etc.

In the TSF, future values are predicted continuously as trading proceeds. If the predicted value is greater than the closing value, the system shall take a long position for the upcoming trade. And if it is lower than the previous value, it takes a short position; anything else it will do nothing. Instead of testing out each individual algorithm under the TSF family, a representative algorithm will be chosen based on its best prediction accuracy for this specific set of testing data. Oracle Crystal Ball [11] that is well-known prediction software with good industrial strength is used to find a prediction model that offers the best accuracy. For comparison the “best” candidate forecasting algorithm is selected by Oracle Crystal Ball that yields the lowest average prediction error. Oracle Crystal Ball has built-in estimators that calculate the performance of each prediction model by four commonly used accuracy measures: the mean absolute deviation (MAD), the mean absolute percent error (MAPE), the mean square error (MSE), and the root mean square error (RMSE). Theil’s U statistic is a relative accuracy measure that compares the forecasted results with a naive forecast. When Theil’s U is one the forecasting technique is about as good as guessing; more than one implies the forecasting technique is worse than guessing, less than one means it is better than guessing. Durbin–Watson statistic is a test statistic used to detect the presence of autocorrelation in the prediction error. The value always lies between 0 and 4. If the Durbin–Watson statistic is substantially smaller than 2, there is evidence of positive serial correlation. In general if Durbin–Watson is smaller than 1, there may be cause for alarm. Small values of Durbin–Watson statistic indicate successive error terms are, on average, close in value to one another, or positively correlated. If it is greater than 2 successive error terms are, on average, much different in value to one another, i.e., negatively correlated. In regressions, this can imply an underestimation of the level of statistical significance. Table 3 lists the prediction accuracies in terms of the error measures. The best performer is Single Exponential Smoothing prediction model for the chosen testing dataset.

TABLE III.

PERFORMANCE OF PREDICTIVE MODELS GENERATED BY ORACLE CRYSTAL BALL

Methods	Rank	MAPE	RMSE	MAD	Theil's U	Durbin-Watson
ARIMA(0,1,0)	2	0.95%	252.97	203.29	1.00	1.9861
Double Exponential Smoothing	4	0.95%	253.09	203.40	1.0005	1.9665
Double Moving Average	5	1.21%	325.81	258.90	1.2924	1.4262
Single Exponential Smoothing	**1**	**0.95%**	**253.00**	**203.25**	**1.0001**	**1.9528**
Single Moving Average	3	0.95%	252.97	203.29	1.00	1.9861

With this optimal prediction model suggested by Oracle Crystal Ball for the given data, we apply the following trade strategies for the prediction model: For a long position to open, the following equation should be satisfied, $Pv_{t+1} - P_t > 0$. For a short position to open, the following equation should be satisfied, $Pv_{t+1} - P_t < 0$ where Pv_{t+1} is the predictive value, and P_t is the closing price at the time t.

The two trading models, one by TF and the other by TSF, are put vis-à-vis in the simulation. The simulation results are gathered and presented in Table 4 and their corresponding performance curves are shown in Figure 12. The results show that Trend Recalling consistently outperformed Single Exponential Smoothing algorithm in our experiment.

TABLE IV.

SIMULATION RESULTS OF "PREDICTIVE MODEL" AND "REACTIVE MODEL"

Simulation	Predictive Model (Single Exponential Smoothing)	Reactive Model (Recalling)
Total Index Point	1995	7201
Net Worth (HKD)	99750	360050
Total Trade	246	1186
Cost (Commission)	9496	45780
P&L	90254	314270
Monthly ROI	10.16%	35.39%
ROI (Year)	122%	425%

IV. CONCLUSION

Trend following has been known as a rational stock trading technique that just rides on the market trends with some preset rules for deciding when to buy or sell. TF has been widely used in industries, but none of it was studied academically in computer science communities. We pioneered in formulating TF into algorithms and evaluating their performance. Our previous work has shown that its performance suffers when the market fluctuates in large extents. In this paper, we extended the original TF algorithm by adding a market trend recalling function, innovating a new algorithm called Trend Recalling Algorithm. Trading strategy that used to make profit from the past was recalled for serving as a reference for the current trading. The trading strategy was recalled by matching the current market trend that was elapsed since the market opened, with the past market trend at which good profit was made by the strategy. Matching market trend patterns was not easy because patterns can be quite different in details, and the problem was overcome in this paper. Our simulation showed that the improved TF model with Trend Recalling algorithm is able to generate profit from stock market trading at more than four times of ROI. The new Trend Recalling algorithm was shown to outperform the previous TF algorithms as well as a time-series forecasting algorithm in our experiments.

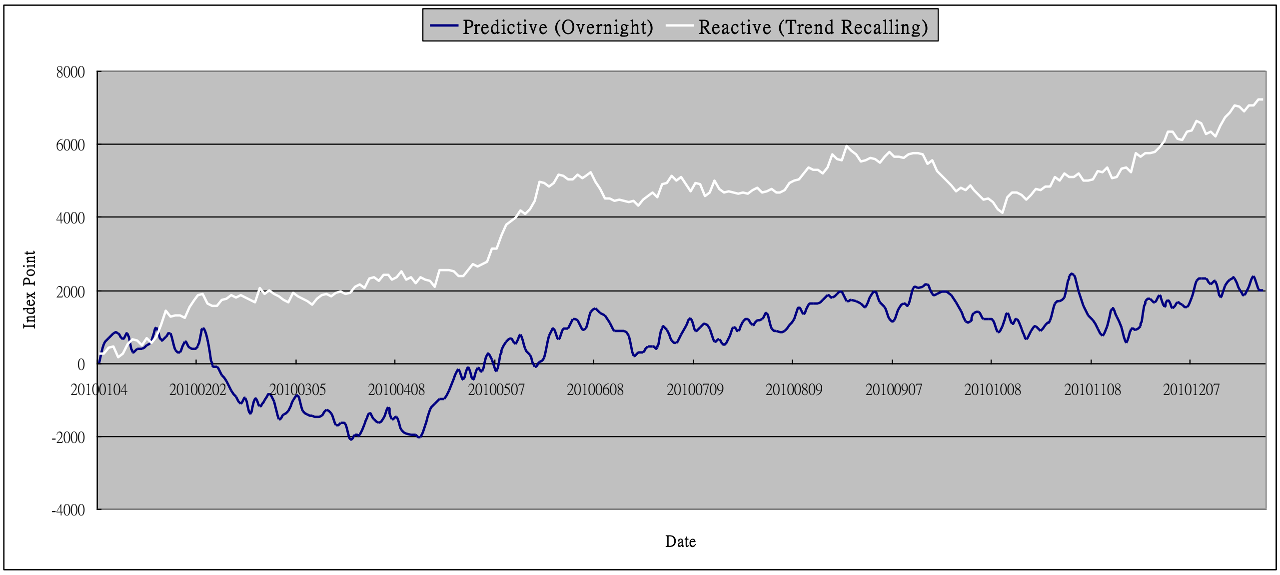

Figure 12. Simulation trade result of predictive model and reactive model on HSI futures contracts in 2010.

REFERENCES

[1] Fong S. and Tai J., " Improved Trend Following Trading Model by Recalling Past Strategies in Derivatives Market", The Third International Conferences on Pervasive Patterns and Applications (PATTERNS 2011), 25-30 September 2011, Rome, Italy, pp. 31-36

[2] Fong S., Tai J., and Si Y.W., "Trend Following Algorithms for Technical Trading in Stock Market", Journal of Emerging Technologies in Web Intelligence (JETWI), Academy Publisher, ISSN 1798-0461, Volume 3, Issue 2, May 2011, Oulu, Finland, pp. 136-145.

[3] Stan Weinstein's, Secrets for Profiting in Bull and Bear Markets, pp. 31-44, McGraw-Hill, USA, 1988.

[4] Wikipedia, 1997 Asian Financial Crisis, Available at http://en.wikipedia.org/wiki/1997_Asian_Financial_Crisis, last accessed on July-3-2012.

[5] Wikipedia, Financial crisis of 2007–2010, Available at http://en.wikipedia.org/wiki/Financial_crisis_of_2007, last accessed on July-3-2012.

[6] Schannep J., "Dow Theory for the 21st Century: Technical Indicators for Improving Your Investment Results", Wiley, USA, 2008.

[7] Covel M.W., "Trend Following: How Great Traders Make Millions in Up or Down Markets", New Expanded Edition, Prentice Hall, USA, 2007, pp. 220-231.

[8] Weissman R.L., "Mechanical Trading Systems: Pairing Trader Psychology with Technical Analysis", Wiley, USA, 2004, pp. 10-19.

[9] Mehdi K. and Mehdi B., "A New Hybrid Methodology for Nonlinear Time Series Forecasting", Modelling and Simulation in Engineering, vol. 2011, Article ID 379121, 5 pages, 2011.

[10] Domowitz I. and Yegerman H., "Measuring and Interpreting the Performance of Broker Algorithms", 2005, Techical Report, ITG Inc., August 2005, pp. 1-12.

[11] Oracle Crystal Ball, a spreadsheet-based application for predictive modeling, forecasting, simulation Available at http://www.oracle.com/technetwork/middleware/crystalball/overview/index.html, last accessed on July-3-2012.

Appendix – The Pseudo Code of the Trend Recalling Algorithm

```
Loop all historical data
        Loop each minute within each day
                Compute and save technical data
                Compute EDM

                Found all the turning point according to EDM
                For each two connected turning point
                        Found their respective position
                        Calculate P&L according to the position
                        If P&L is positive
                                Save position and each point time line
                        Else
                                Adjust each point time line
                End-loop
        End-loop
End-loop
```

```
Loop each minute until end of market
        Compute technical data
        Compute volatility

        For each sample in pool
                Compare sample technical data with current market technical data
                Save their similarity into list
        End-loop

        Sort the list with most similar on the top

        For 1 To 20 in the list
                Apply fitness test on each sample with current market trend
                Save their performance into list
        End-loop

        Sort the list with most fitness on the top

        Extract strategies from the top fitness sample

        Reference RSI crossover system

        Apply fuzzy sets

        If fuzzy sets not fire
                If no position is opened
                        If strategy shows long position and confirm with RSI crossover
                                Open long position
                        Else if strategy shows short position and confirm with RSI crossover
                                Open short position
                Else if long position is opened
                        if strategy shows short position and confirm with RSI crossover
                                Close long position
                                Open short position
                Else if short position is opened
                        if strategy shows long position and confirm with RSI crossover
                                Close short position
                                Open long position

        If end of the day
                Close any opened position
End-loop
```