**Proceedings Paper:**

Kazakov, Dimitar Lubomirov and Butler, Matthew Richard (2012) Testing Implications of the Adaptive Market Hypothesis via Computational Intelligence. In: Computational Intelligence for Financial Engineering & Economics (CIFEr), 2012 IEEE Conference on. 2012 IEEE Computational Intelligence for Financial Engineering and Economics (CIFEr 2012), 29-30 Mar 2012 IEEE, USA, pp. 1-8.

https://doi.org/10.1109/CIFEr.2012.6327799

# Testing Implications of the Adaptive Market Hypothesis via Computational Intelligence

Matthew Butler and Dimitar Kazakov

***Abstract*—This study analyzes two implications of the Adaptive Market Hypothesis: variable efficiency and cyclical profitability. These implications are, inter alia, in conflict with the Efficient Market Hypothesis. Variable efficiency has been a popular topic amongst econometric researchers, where a variety of studies have shown that variable efficiency does exist in financial markets based on the metrics utilized. To determine if non-linear dependence increases the accuracy of supervised trading models a GARCH process is simulated and using a sliding window approach the series is tested for non-linear dependence. The results clearly demonstrate that during sub-periods where non-linear dependence is detected the algorithms experience a statistically significant increase in classification accuracy. As for the cyclical profitability of trading rules, the assumption that effectiveness waxes and wanes with the current market environment, is tested using a popular technical indicator, Bollinger Bands (BB), that are converted from static to dynamic using particle swarm optimization (PSO). For a given time period the parameters of the BB are fitted to optimize profitability and then tested in several out-of-sample time periods. The results indicate that on average a particular optimized BB is profitable, active and able to outperform the market index up to 35% of the time. These results clearly indicate the cyclical nature of the effectiveness of a particular trading model and that a technical indicator derived from historical prices can be profitable outside of its training period.**

## I. INTRODUCTION

The Adaptive Market Hypothesis (AMH) of Lo [14][15] offers an alternative market theory to Fama's Efficient Market Hypothesis (EMH) [5] that has several conflicting assumptions. These include the issues of bounded rationality of individual investors, path dependence of the equity-risk premium and variable market efficiency. The last assumption, that of variable efficiency, has been a popular topic amongst econometric researchers, where a variety of studies have shown that it does exist [2] [13] [20] in the financial markets for the metrics considered. These studies have also revealed that market efficiency is not a convergence but is in fact cyclical. This evidence supports the AMH and implies that a non-zero probability exists for creating trading strategies that outperform the market. Given that markets appear to exhibit non-linear correlations there still remains the question whether or not active trading strategies or technical analysis can take advantage of these inefficient market periods. The observation that market efficiency is cyclical is dependent on the robustness of the statistical test. From a forecasting point of view the most important question, assuming a cyclical nature to market efficiency, is whether or not these periods of non-linear dependence can be used to improve forecasting accuracy and therefore lead to more profitable trading models. The previous work on market efficiency was mainly concerned with demonstrating that efficiency was episodic and that a relationship existed between the maturity of the market and its degree of market efficiency. The results from each of the studies [20] [13] revealed that emerging markets tended to be less efficient than mature markets. In 2009, Todea et al. [18] analyzed if the profitability of an optimal moving average (MA) strategy was contingent on the market period. The results were obtained for six Asia-Pacific financial markets and in five of the markets the MA strategy was more profitable in periods that exhibited non-linear dependencies. These results however do not reflect any out-of-sample testing as an optimal strategy was determined a priori for a particular market and the results do not reveal if any advantages exist for forecasting future price trends. This is the motivation behind this research, to determine if the *presence* of non-linear dependencies in a time series offers any benefits to forecasting models developed from machine learning techniques. The word presence is emphasized as the actual data generating process is not known and any dependencies identified are contingent on the robustness of the statistical test.

In relation to the cyclical nature of market efficiency this study also assesses the validity of cyclical profitability. Due to the non-stationary nature of the stock market it is a valid assumption that trading models have to continually adapt to new environments. Though this may be true it does not necessarily imply that previously effective models do not contain any useful information. If trading models exhibit cyclical effectiveness then maintaining and consulting previous models may improve forecasting performance. In essence this would be a passing on of knowledge from older generations to new ones. This positive impact of older generations is seen in the natural world where the emergence of grandparents in human society led to an explosion of sophisticated tools and art [3]. There are various methods which could be explored to test the validity of cyclical profitability of technical analysis. There are several technical trading rules, such as the moving average convergence divergence (MACD) or momentum indicators (MOM), which could be easily implemented and their effectiveness monitored through time. A potential drawback is that the trading rule may never be desirable and although its profitability varies in time, the overall effectiveness may be sub-par to that of the market index and therefore rendering the

Matthew Butler and Dimitar Kazakov are with the Artificial Intelligence Group in the Department of Computer Science, The University of York, Deramore Lane, Heslington, York, UK (email: {mbutler, kazakov}@cs.york.ac.uk).

experimental results moot. This conclusion, of course, is based on the fact that an active technical trading rule that cannot outperform the passive buy and hold approach is irrelevant and is evidence against the AMH. Alternatively, we could use an active learning approach where an optimal trading strategy can be constructed for the majority of market environments. This approach would ensure that each trading model tested was at one time profitable and able to outperform the passive buy-and-hold approach. In section III-A we discuss the exact methodology used for choosing BBs fitted using PSO and how the results are evaluated.

## II. Variable Efficiency

To analyze the effect of non-linear dependence in a time-series, on the forecasting accuracy of Supervised Learning, a generalized autoregressive conditional heteroskedasticity (GARCH) model is used to simulate a financial time-series. A GARCH model, as the name suggests, allows for conditional variance that is not constant through time (a characteristic that is commonly observed in financial time series). The form of a GARCH(1,1) process for a series of discrete observations $\{Y_t\}$ is given below:

$$Y_t = \sigma_t \epsilon_t \quad (1)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 Y_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \quad (2)$$

where $\epsilon_t$ is standard Gaussian white noise and the condition that $\alpha_1 + \beta_1 < 1$. Equations 1 and 2 return a white noise process with non-constant conditional variance, where the variance depends on the previous return. Equation 2 can be easily extended to include more lags. For the purpose of this study a GARCH(2,2) model was chosen. In the next section the methodology for identifying episodic non-linear dependence is explained.

### A. Non-linear Dependence

The methodology for this study is based on [13] [2] where a sliding window approach is used to partition the time series into subsamples that exhibit random walk behaviour and non-linear dependence. For a time series $\{Y_t\}_1^T$ and a window of size $d$ an initial sub-sample is created consisting of observations $\{Y_t\}_1^d$, the appropriate tests are run and then the window shifts by one day to cover $\{Y_t\}_2^{d+1}$ and so forth until the end of the sample $\{Y_t\}_{T-d}^T$. The window size used in this study is the same as [18] which is 200 observations.

Within each sliding window the sample is tested for non-linear dependence using the Hinich Portmanteau bi-correlation (H) test [6]. Prior to applying the Portmanteau tests the data within the sliding window undergoes two stages of pre-processing. First, the series $\{Y_t\}_1^T$ is considered to be a non-stationary stochastic process and to aid with the analysis the series is transformed to stationary by converting the series to continuously compounded percentage returns, as follows:

$$r_t = log(y_t / y_{t-1}) * 100 \quad (3)$$

where $r_t$ is the daily percentage return for time $t$. The second step is to standardize the data within each window to have a sample mean of zero and a sample standard deviation of one, as follows:

$$Z(t) = \frac{R(t) - m_R}{\sigma_R} \quad (4)$$

where Z(t) is the standardized series, $m_R$ is the sample mean and $\sigma_R$ is the sample standard deviation. The null hypothesis of the test is that $\{Z(t)\}$ is a realization of a white noise process with null bi-correlations. The Portmanteau test for non-linear correlations is calculated as follows:

$$H = \sum_{s=2}^{L} \sum_{r=1}^{s-1} G^2(r, s) \quad (5)$$

where,

$$G(r, s) = (n - s)^{1/2} C_{RRR}(r, s) \quad (6)$$

and,

$$C_{RRR}(r, s) = (n - s)^{-1} \sum_{t=1}^{n-s} Z(t)Z(t + r)Z(t + s) \quad (7)$$

where $r$ and $s$ satisfy $0 < r < s < L$. The $H$ statistic is distributed according to a $\chi^2$ law of probability with ($L$-1)($L$/2) degrees of freedom. The number of lags ($L$) is specified as L = $n^b$, with $0 < b < 0.5$ and $n$ is the window size. Previous work by Hinch and Patterson [6] recommend a value of 0.4 for $b$.

In addition to the pre-processing performed above; the series $\{Z(t)\}$ undergoes one additional step of pre-whitening before being supplied to the H bi-correlation test. The pre-whitening step entails filtering away the linear component and therefore any autocorrelation structure of $\{Z(t)\}$ by means of an autoregressive $AR(p)$ fit. The order $p$ is chosen between 0-10 as the smallest value for which the Ljung-Box Q(10) statistic is insignificant at the 10% level.

### B. Supervised Learning

We are interested in the effect, if any, non-linear correlations have on the forecasting abilities of trading models developed from supervised learning (SL). There is no shortage of literature of SL techniques being developed and applied to the financial domain. The dynamic and non-stationary nature of the financial markets makes them a challenging and attractive system to model using complex methods. This study focuses on six well established learning paradigms that are widely available for use. The algorithms considered are:

1) Multilayer Perceptron (MLP)
2) Support Vector Machine (SVM)
3) Artificial Immune System (AIS)
4) J48 Decision Tree (J48)
5) $k$-Nearest Neighbour (kNN)
6) Naïve Bayes

The forecasting task for each of the algorithms is classification. Each tuple of information supplied to the various SL techniques will have a class attribute ($C_i$) where $C_i \in \{0, 1\}$. 0 signifies a market contraction and 1 signifies a market expansion. Using the described above methodology two sub-samples are

TABLE I

THE RESULTS FROM TRAINING AND TESTING THE SL ALGORITHMS ON THE GARCH SUBSAMPLE DATA. NLD REPRESENTS SAMPLES WITH NON-LINEAR DEPENDENCE AND RW REPRESENTS SAMPLES ADHERING TO A RANDOM WALK. *, ** SIGNIFIES THE INCREASE IN ACCURACY IS STATISTICALLY SIGNIFICANT AT THE 5% AND 1% LEVELS RESPECTIVELY.

| | RW | | | NLD | | |
|---|---|---|---|---|---|---|
| Algorithm | Acc. | Min. | Max. | Acc. | Min. | Max. |
| MLP | 0.587 | 0.347 | 0.755 | 0.622** | 0.367 | 0.796 |
| SVM | 0.625 | 0.510 | 0.796 | 0.656** | 0.531 | 0.775 |
| AIS | 0.569 | 0.327 | 0.796 | 0.580* | 0.388 | 0.755 |
| J48 | 0.617 | 0.469 | 0.755 | 0.656** | 0.429 | 0.755 |
| kNN | 0.629 | 0.428 | 0.775 | 0.633 | 0.367 | 0.861 |
| NB | 0.617 | 0.429 | 0.796 | 0.651** | 0.490 | 0.775 |

created from the simulated GARCH process. One subsample consists of data that contains non-linear dependencies and the other contains data that adheres to a stochastic random walk. The SL algorithms are then applied to the separate samples, where 75% is allocated for training and 25% for testing.

### C. Experiment Results

After applying the above methodology the GARCH process which was 1000 data points long, yielded 799 samples using a 200 data point sliding window. The class distribution within the simulated series as a whole was a 34/64 split in favour of class 0; meaning more market contractions. These samples were then partitioned into 534 samples which adhered to a random walk and 265 samples that exhibited non-linear dependence. Figure 1 provides some example plots of the GARCH subsamples that exhibited random walk behaviour (right) and non-linear dependence (left). The results from training and testing the algorithms are presented in table I and figure 2.

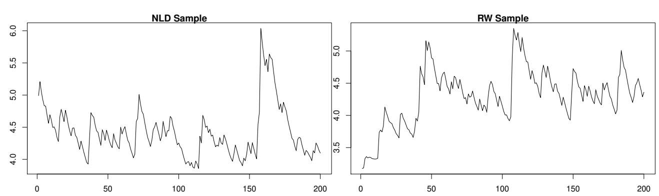


Fig. 1. Example plots of the GARCH process when it is exhibiting non-linear dependence (NLD) (left) and random walk (RW) behaviour (right).

The results in table I show that all 6 algorithms achieved a higher directional accuracy in the subsamples that exhibited non-linear dependence and in 5 of the 6 cases the increase was statistically significant based on a one-sided t-test. The only exception was the kNN algorithm where only a small incremental gain was realized, however the overall accuracy was comparable to the other algorithms. These results indicate that when non-linear dependence is present the SL algorithms tested were able to take advantage of this deterministic component of the signal.

## III. CYCLICAL PROFITABILTIY

The area of computational intelligence (CI) offers several algorithms that can learn and adapt to noisy and non-stationary

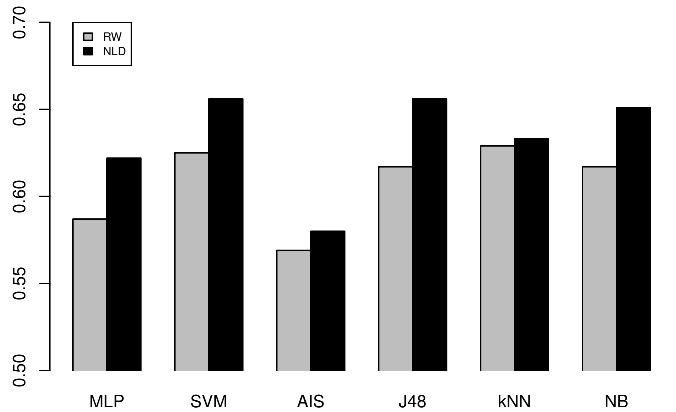


Fig. 2. A Histogram of the testing accuracy results from the Random Walk (RW) and non-linear dependent (NLD) subsamples.

environments. Concerning financial time-series analysis, several studies have shown that CI algorithms have been effective at learning and forecasting, producing results suggesting that the markets are not perfectly efficient. From this we have to decide what the primary objectives of the study are and which algorithms can accommodate. The list of primary objectives is provided below.

1) Optimal - able to outperform the market benchmark,
2) Flexible - adapt to changing market conditions, and
3) Interpretable - surmise what the agent is doing and determine market conditions from agent structure

For the purpose of this study we are asserting that the simple waxing and waning of a trading policy is not strict enough to test this implication of the AMH and acquire a meaningful result. Thus we are testing if whether an optimal strategy formed in one time-period is ever *effective* again. A strategy will be considered effective if the following criteria are satisfied:

$$R_t(TM) > R_t(M), \tag{8}$$
$$R_t(TM) > 0, \text{ and} \tag{9}$$
$$T_t > 0 \tag{10}$$

where $R_t(TM)$ and $R_t(M)$ are the returns of the trading model and the market in time period $t$ respectively and $T_t$ is the corresponding number of trades in time period $t$. These criteria state that a trading model is effective if it is able to outperform the market index benchmark, while producing a positive return and is active in the market.

The first of the primary objectives is to ensure that the results are meaningful. Secondly, for a trading model to be profitable in a range of market conditions that model needs to be flexible. Rigid trading rules will not produce above average returns at all times, which is precisely why technical analysis is difficult. Thirdly, the model should be white box. The results from the analysis would be more meaningful if we could interpret what the agent has learned, and if we could surmise what type of market conditions are suitable for a particular agent. For example, can we determine if the market was trending or more volatile based on the agent's structure?

Let us start with one of the most popular learning paradigms from CI for time-series analysis, Artificial Neural Networks (ANN), where studies have shown that they are arguably among the most robust [19] [9]. In the context of the three primary objectives we can determine that ANNs are able to outperform the market during training, that they are flexible but represent a black-box model, and that it would be difficult to extract domain knowledge from the topology and connection weights. Support Vector Machines have also become popular in the financial forecasting literature and offer a robust and flexible modelling approach, however, they also suffer from a lack of interpretability just as the ANNs. Evolutionary Computation (EC) is also an active area of research in financial forecasting and encompasses a variety of techniques from genetic algorithms (GA), genetic programming (GP), Artificial Immune Systems (AIS) and hybrid algorithms, to name a few. Once again, in the canonical use of these techniques we can easily accommodate the objectives of flexibility and optimality but the models will generally be black box. Moving back to traditional technical analysis, certain trading rules could be more effective in trending markets (moving averages) and others when the market is moving sideways (Bollinger Bands) and although it is possible to interpret these rules, they are, by construction, static.

With each of these techniques possessing weaknesses with respect to the primary objectives, it is a natural succession to entertain the combination of two or more of them. There has been documented success in combining population based optimization techniques with technical trading models, such as GAs with moving averages [10]. This would entail determining the length of windows for calculating the moving averages via profitability based fitness functions. Another recent study combined Bollinger Bands with Particle Swarm Optimization (PSO) [1] to tune the parameters to current market conditions. The experiments implied that the effectiveness of the indicator could be enhanced beyond that of just using the default parameters. In the context of the primary objectives the hybrid models are the most suitable. Using an architecture from traditional technical analysis allows for interpretable models; additionally the benefit of flexibility from the CI algorithms is retained, and finally the comparability between models is possible as the technical trading rules have a finite set of attributes, which allows for comparisons in a relatively small $n$-dimensional space.

For this study the optimal trader for each market segment will be determined using Adaptive Bollinger Bands (ABB) [1], which are based on a technical indicator created by John Bollinger in the 1980's.

### A. Adaptive Bollinger Bands

The ABBs were initially developed because, despite their popularity, the recent academic literature had shown Bollinger Bands (BB) to be ineffective [11] [12]. However, through PSO-based parameter fine tuning the indicator could be improved and outperform the market index under certain market conditions. The three main components of BBs are:

1) An N-day moving average (MA) for a price series $\{P_i\}$, which creates the middle band, equation 11,

$$MA_n(t) = \frac{\sum_{i=t-N+1}^{t} P_i}{N} \quad (11)$$

2) an upper band, which is the MA plus $k$ times the standard deviation of the middle band, and
3) a lower band, which is the MA minus $k$ times the standard deviation of the middle band.

The default settings for using BBs are a moving average window of 20 days and a value of $k$ equal to 2 for both the upper and lower bands. When the price of the stock is trading above the upper band, it is considered to be overbought, and conversely, an asset which is trading under the lower band is oversold. The trading rules that can be generated from using this indicator are given by equations 12–13:

$$Buy : P_i(t-1) < BB_n^{low}(t-1) \& P_i(t) > BB_n^{low}(t) \quad (12)$$

$$Sell : P_i(t-1) > BB_n^{up}(t-1) \& P_i(t) < BB_n^{up}(t) \quad (13)$$

Essentially, the above rules state that a buy signal is initialized when the price ($P_i$) crosses the lower band from below, and a sell signal when the price crosses the upper band from above. Using the BBs in their canonical form, in both cases the trade can be closed out when the price crosses the middle band. As such, a trader will be taking long/short positions in the market; a long/short position is a trading technique which profits from increasing/decreasing asset prices.

To allow for efficient online optimization of the BBs we define two new forms of the traditional indicator, running and exponential BBs, that make use of estimates of the $1^{st}$ and $2^{nd}$ moments of the time series.

*1) Running and Exponential Bollinger Bands:* We define a BB as:

$$BB = MA_n \pm k \times \sigma(n_{period}) \quad (14)$$

where $MA_n$ is an $n$-day moving average and $\sigma$ is the standard deviation. Then a Running Bollinger Band that makes use of estimates of the $1^{st}$ and $2^{nd}$ moments is:

$$BB = A_n \pm k \times J_n(B_n - A_n^2)^{1/2} \quad (15)$$

where,

$$A_n = \frac{1}{n}\sum_{i=1}^{n} Y_i \quad , \quad B_n = \frac{1}{n}\sum_{i=1}^{n} Y_i^2 \quad (16)$$

$$J_n = \frac{n}{n-1}^{1/2} \quad (17)$$

where the normalization factor $J_n$ allows for an unbiased estimate of the $\sigma$ and $Y_i$ is $i^{th}$ data point. From this, recursive updates of the BBs can be performed as follows:

$$A_n = \frac{1}{n}Y_n + \frac{n-1}{n}A_{n-1} \quad (18)$$

$$B_n = \frac{1}{n}Y_n^2 + \frac{n-1}{n}B_{n-1} \quad (19)$$

TABLE II

THE PARAMETERS THAT THE PSO ALGORITHM OPTIMIZED. MA STANDS FOR MOVING AVERAGE. THE PARTICELS ARE THE NUMBER OF PARTICLES FROM EACH INDIVIDUAL IN THE SWARM ALLOCATED FOR THAT PARAMETER.

| Description | Particles |
|---|---|
| The value for calculating the upper/lower band. | 2/2 |
| Window size for the upper/lower band MA. | 5/5 |
| The type of ABB to use for upper/lower band. | 1/1 |
| The stop loss for short-sells/buys. | 2/2 |

TABLE III

THE RANGE OF FEASIBLE VALUES FOR EACH PARAMETER AND ITS CORRESPONDING MAXIMUM VELOCITY FOR NAVIGATING THE SOLUTION SPACE. WHERE $\oplus$ SIGNIFIES THAT THE MA TYPE CAN ONLY TAKE ON VALUES OF 0 OR 1.

| Parameter | Range | Max Velocity |
|---|---|---|
| Upper Band | {-4,4} | 0.10 |
| Lower Band | {-4,4} | 0.10 |
| MA Type | $\{0 \oplus 1\}$ | 0.10 |
| Stop Loss | {-0.99,0} | 0.10 |
| Window Size | {5,500} | 20 |

For the exponential form we define the BB on a time scale $\eta^{-1}$. Where incremental updates of the estimates are:

$$A_n = \eta Y_i + (1-\eta)A_{n-1} \tag{20}$$

$$B_n = \eta Y_i^2 + (1-\eta)B_{n-1} \tag{21}$$

and the normalization factor becomes:

$$J_n = \frac{1-\eta/2}{1-\eta} \tag{22}$$

This implementation of the ABBs was written in JAVA and optimizes eight parameters, as displayed in table II. A result from [1] concluded that BBs are ineffective at generating profits when the market is trending. This shortcoming of the BBs was mainly due to the exiting of profitable trades prematurely. To counteract this consequence of using the middle band (the $N$ day moving average) to initiate the closing out of a trade, this implementation uses trailing stop-losses to determine exit points. A trailing stop-loss is a popular trading technique that essentially allows a set amount to be lost from the maximum profit achieved.

An additional advantage to using BBs as the underlying technical analysis tool is that we are able to tap into a common heuristic used by active traders of identifying turning points in stock movements. The identification of an overbought or oversold security signals a correction and therefore a change in directional movement. However, choosing a turning point is very difficult as a trader will be taking positions that are contrary to the current market trend.

### *B. Particle Swarm Optimization*

Particle Swarm Optimization (PSO) [8] is a population based algorithm inspired from swarm intelligence commonly used in optimization tasks. PSO has had success with searching complex solution spaces, similar to the abilities of genetic algorithms (GA). PSO was chosen for the original study as it had been shown to be as effective as GAs when modelling technical trading rules, as in Lee et al. [10], yet it had a much simpler implementation and arrived at a global optimum with fewer iterations. Each particle in the swarm represented an $n$-dimensional position vector that maps to the various parameters displayed in table II. In its canonical form the swarm is governed by the following:

$$\begin{aligned} \upsilon_{i,j} = \omega \times \upsilon_{i,j} + c_1 r_1 \times (local_{best\, i,j} - x_{i,j}) \\ + c_2 r_2 \times (global_{best\, j} - x_{i,j}) \end{aligned} \tag{23}$$

Here $\upsilon_{i,j}$ is the velocity of $j^{th}$ dimension of the $i^{th}$ particle, $c_1$ and $c_2$ determine the influence on a particular particle by its optimal position previously visited and the optimal position obtained by the swarm as a whole, $r_1$ and $r_2$ are uniform random numbers between 0 and 1, and $\omega$ is an inertia term (see [17]) chosen between 0 and 1.

$$x_{i,j} = x_{i,j} + \upsilon_{i,j} \tag{24}$$

Here $x_{i,j}$ is the position of the $j^{th}$ dimension of the $i^{th}$ particle in the swarm. To encourage exploration and limit the speed with which the swarm would converge, a maximum velocity was chosen for each dimension dependent on its range of feasible mappings. In table III the range and maximum velocity for each parameter is displayed. The type of ABB to use was mapped using a wrapper function which evaluated to a running BB if the particle had a value greater than or equal to 0.5 and mapped to an exponential BB if the particle had a value less than 0.5.

*1) Heterogenus Particle Swarm Optimization:* This study used a more sophisticated version of PSO called Dynamic Heterogeneous Particle Swarm Optimization (dHPSO) [4] which has been shown to outperform the canonical from of PSO on a variety of optimization problems. With dHPSO the position update remains the same but the calculation of the velocity update is expanded to allow for alternatives. The swarm becomes heterogeneous as each particle in the swarm will have one of five possible velocity update profiles and the swarm is dynamic as the velocity update profile will change if a particle becomes stagnant. The additional velocity updates are as follows:

$$\upsilon_{i,j} = \omega \times \upsilon_{i,j} + c_1 r_1 \times (local_{best\, i,j} - x_{i,j}) \tag{25}$$

$$\upsilon_{i,j} = \omega \times \upsilon_{i,j} + c_2 r_2 \times (global_{best\, j} - x_{i,j}) \tag{26}$$

$$\upsilon_{i,j} \sim N\left(\frac{local_{best\, i,j} + global_{best\, j}}{2}, \sigma\right) \tag{27}$$

$$\upsilon_{i,j} = \begin{cases} local_{best\, i,j} & \text{if U(0,1)} < 0.5 \\ N\left(\frac{local_{best\, i,j}+global_{best\, j}}{2}, \sigma\right) & \text{otherwise} \end{cases} \tag{28}$$

where, $N$ and $U$ are normal and uniform distributions respectively. Equation 25 is the cognitive only profile where the social component has been removed. This promotes exploration as each particle becomes a hill-climber. Equation 26 is the social only profile where the cognitive component has been removed. In effect the entire swarm becomes one large hill-climber. Equation 27 is the Barebones PSO where the position update is the velocity update, so:

$$x_{i,j} = v_{i,j}, \quad and \tag{29}$$
$$\sigma = |local_{best\,i,j} - global_{best\,j}|. \tag{30}$$

Finally, equation 28 is the modified Barebones profile. One additional improvement has been made to the dHPSO algorithm where particles that continue to be stagnant after velocity profile changes will be re-initialized randomly in the solution space. This modification was shown to improve the algorithms ability to find solutions that outperform the market index.

*2) Fitness Function:* The goal of the experiment is to create an optimal trader, determined by profitability, for each market segment. As such, it would seem obvious that training the swarm with a fitness function based on profit would be the most appropriate. Although other literature, Moody et al. [16], has found that optimal performance was arrived at with fitness functions which have a risk to reward payoff, the previous study which developed the ABBs concluded that a fitness function which simply maximizes profitability was the most effective and therefore will be used in this study. The fitness function is as follows:

$$fitness_i = \sum_{t=1}^{T} capital_t \times \frac{(P_{1,t} - P_{0,t})}{P_{0,t}} - (\tau \times capital_t) \tag{31}$$

where $fitness_i$ is the fitness of the $i^{th}$ particle in the swarm, $\tau$ represents the transaction costs, $T$ is the total number of trades, and $P_0$ and $P_1$ are the entering and exiting price for the underlying asset. The profit for each trade is the rate of return multiplied by the capital invested minus the transaction cost which is also a function of the amount of capital invested. It is important to keep in mind that the number of trades does not reflect the amount of time invested in the market. Once an ABB enters the market, either short or long, the trade is maintained until the end of the test period or the stop-loss criteria is satisfied.

### C. Data and Experiment Setup

The data used for testing cyclical profitability were the daily closing prices for the S&P 500 for a 10 year time period spanning 2001-2010. The first 5 years were allocated for training the ABBs with the remaining 5 years for testing. A benefit of using BBs (as well as other technical analysis techniques) is that no pre-processing of the data is required as the indicators do not make any assumptions of normality or stationarity.

*1) Creating Optimal Agents:* To allow for a range of investment policies we analyze the optimal traders at different levels of granularity. Thus the experiments are conducted for an increasing number of data points within the sliding window. The use of the sliding window is the same as described in section II-A. Table IV displays the parameters and number of agents created for each of the experiment setups. To assess the profitability of the agents, the experiments are carried out with an initial starting capital of £1000.00 and a transaction rate (applied when entering and exiting the market) of 0.25%, i.e., a quarter of a percent of the amount of capital invested. We assume no transaction costs for investing in the risk-free rate ($R_f$) which is accrued daily and has AER of 2%. In this implementation the ABB fully invests all capital each day and whilst in a trade no other positions can be taken.

TABLE IV
THE PARAMETERS FOR THE VARIOUS EXPERIMENT SETUPS.

| Case | Window | # of Agents | ≈ time | # of test periods |
|---|---|---|---|---|
| 1 | 125 | 1132 | 6 mths | 1133 |
| 2 | 250 | 1007 | 1 yr | 1008 |
| 3 | 500 | 757 | 2 yrs | 758 |
| 4 | 1000 | 257 | 4 yrs | 258 |

The parameters for the PSO algorithm have the same settings for each experiment and are displayed in table V. In order to maximize the number of time-periods where an agent is identified that outperforms the market, the PSO algorithm will initially train for 100 epochs. If at that time an optimal agent is not found the algorithm is allowed to continue up to a maximum of 1000 epochs. The dimensions are a sum of the number of particles in each position vector allocated to each of the ABB parameters.

TABLE V
THE PARAMETER SETTINGS FOR THE PSO ALGORITHM.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| initial epochs | 100 | max epochs | 1000 |
| $c_1$ | 2 | $c_2$ | 2 |
| Particles | 30 | Dimensions | 20 |

### D. Cyclical Profitability Results

The results presented in this section are the average performance results for the ABBs over all test periods. There are three metrics considered: (1) the average number of ABBs that outperform the market (OM), (2) the average number of ABBs that produce a positive return (PR), and (3) the average number of ABBs that are effective (EF), where effective implies, from the above definition, that the ABB was profitable, active and outperformed the market index. The following sections will present tables and box plots of the results as well as a discussion.

*1) Case 1 through Case 4:* The results from training and testing the ABBs using sliding windows of 125, 250, 500 and 1000 days are presented in tables VI-IX and figures 4-7 are boxplots of the metric distributions.

### E. Discussion

The results presented in tables VI through IX reveal that at each level of granularity there were ABBs that were effective in the out-of-sample test data. Figure 3 plots the average number of trades and the percentage of effective ABBs against the window size. We see an increase in the percentage of the ABBs that are effective as the window size increases. This is due to overfitting, where the ABBs tuned to smaller amounts of

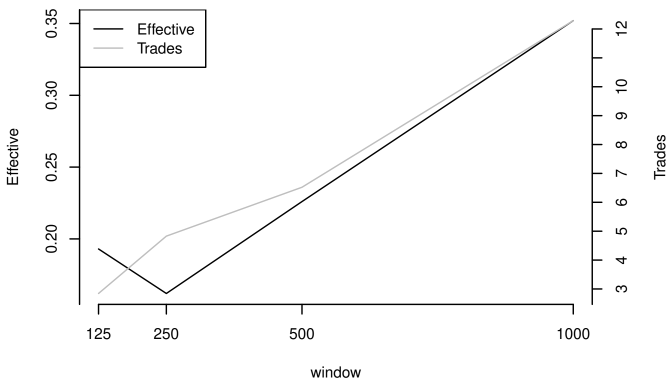


Fig. 3. Plots of the average number of trades for effective ABBs and the average percentage of ABBs that were effective.

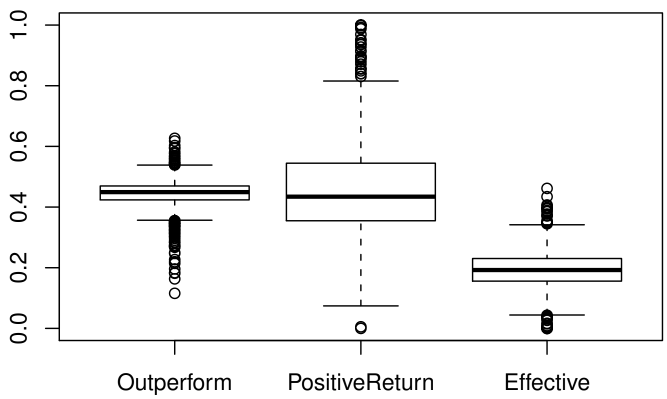


Fig. 4. Box plots of the distributions of the Outperform the Market (OM), Positive Return (PR) and EFfective (EF) metrics for case 1.

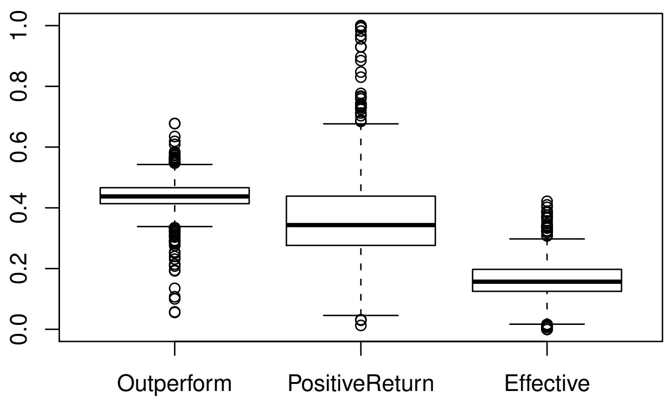


Fig. 5. Box plots of the distributions of the Outperform the Market (OM), Positive Return (PR) and EFfective (EF) metrics for case 2.

TABLE VI
THE RESULTS FROM TRAINING AND TESTING THE ABBS WITH A SLIDING WINDOW OF 125 DAYS. $??_{\bar{tr}}$ STANDS FOR THE AVERAGE NUMBER OF TRADES BY THE ABBS THAT SATISFIED THE CONCERNED METRIC.

| | OM | $OM_{\bar{tr}}$ | PR | $PR_{\bar{tr}}$ | EF | $EF_{\bar{tr}}$ |
|---|---|---|---|---|---|---|
| Average | 0.444 | 2.841 | 0.463 | 2.409 | 0.193 | 2.844 |
| Min | 0.116 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 |
| Max | 0.627 | 21.270 | 1.000 | 19.723 | 0.462 | 19.914 |
| Median | 0.449 | 1.941 | 0.434 | 1.549 | 0.192 | 1.984 |

TABLE VII
THE RESULTS FROM TRAINING AND TESTING THE ABBS WITH A SLIDING WINDOW OF 250 DAYS. $??_{\bar{tr}}$ STANDS FOR THE AVERAGE NUMBER OF TRADES BY THE ABBS THAT SATISFIED THE CONCERNED METRIC.

| | OM | $OM_{\bar{tr}}$ | PR | $PR_{\bar{tr}}$ | EF | $EF_{\bar{tr}}$ |
|---|---|---|---|---|---|---|
| Average | 0.438 | 5.128 | 0.371 | 4.588 | 0.162 | 4.829 |
| Min | 0.056 | 0.000 | 0.013 | 0.000 | 0.000 | 1.000 |
| Max | 0.678 | 41.695 | 1.000 | 38.571 | 0.422 | 38.147 |
| Median | 0.438 | 4.015 | 0.343 | 3.669 | 0.157 | 3.765 |

TABLE VIII
THE RESULTS FROM TRAINING AND TESTING THE ABBS WITH A SLIDING WINDOW OF 500 DAYS. $??_{\bar{tr}}$ STANDS FOR THE AVERAGE NUMBER OF TRADES BY THE ABBS THAT SATISFIED THE CONCERNED METRIC.

| | OM | $OM_{\bar{tr}}$ | PR | $PR_{\bar{tr}}$ | EF | $EF_{\bar{tr}}$ |
|---|---|---|---|---|---|---|
| Average | 0.601 | 8.558 | 0.311 | 6.689 | 0.226 | 6.521 |
| Min | 0.001 | 0.062 | 0.000 | 0.065 | 0.000 | 1.000 |
| Max | 0.959 | 92.201 | 0.997 | 92.028 | 0.858 | 92.028 |
| median | 0.617 | 7.258 | 0.289 | 5.463 | 0.219 | 5.123 |

TABLE IX
THE RESULTS FROM TRAINING AND TESTING THE ABBS WITH A SLIDING WINDOW OF 1000 DAYS. $??_{\bar{tr}}$ STANDS FOR THE AVERAGE NUMBER OF TRADES BY THE ABBS THAT SATISFIED THE CONCERNED METRIC.

| | OM | $OM_{\bar{tr}}$ | PR | $PR_{\bar{tr}}$ | EF | $EF_{\bar{tr}}$ |
|---|---|---|---|---|---|---|
| Average | 0.611 | 12.108 | 0.391 | 11.871 | 0.352 | 12.289 |
| Min | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 |
| Max | 1.000 | 51.171 | 1.000 | 48.952 | 1.000 | 48.952 |
| Median | 0.624 | 13.593 | 0.399 | 13.296 | 0.329 | 13.589 |

data are more likely to become overfitted and to not generalize as well. We also observe a monotonic increase in the average number of trades executed by the ABBs as the window size increases. From the OM and PR metrics we can observe that ABBs are not always active in the market and that the parameters which are optimal in one time period can lead to a technical indicator that does not execute any trades when the market environment is quite different. This is partly the reason for higher percentages of the ABBs producing positive returns but not being able to outperform the market. On average the ABBs made a trade every 3 to 4 months when they were effective, though there were instances where the ABBs were effective and extremely active in executing trades. In case 3 where the window size was 500 days we observe a maximum average trading activity of 92.028, which translates to about 4 trades a month. This is quite active for a technical indicator that is identifying turning points in stocks price behaviour.

The boxplots reveal that none of the metric distributions are normal (all rejected the null of normal from the Jarque-Bera test [7]) and that for the majority of the plots there are several outliers beyond the $1^{st}$ and $3^{rd}$ quartiles. With the exception of case 4 (1000 day window) all of the boxes are quite small indicating that 50% of the data is within close range of the median. This narrow interquartile range coincides with the large amount of outliers or suspected outliers.

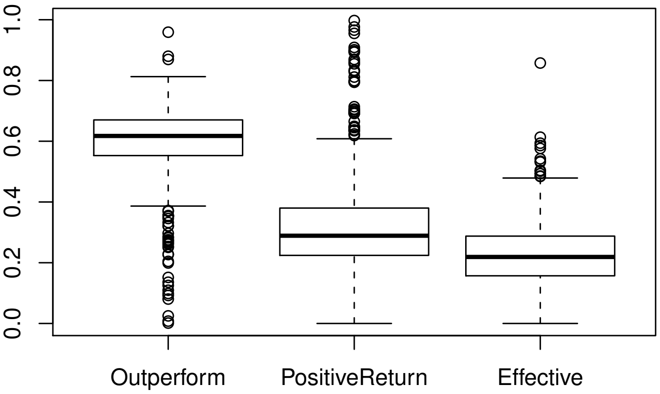


Fig. 6. Box plots of the distributions of the Outperform the Market (OM), Positive Return (PR) and EFfective (EF) metrics for case 3.

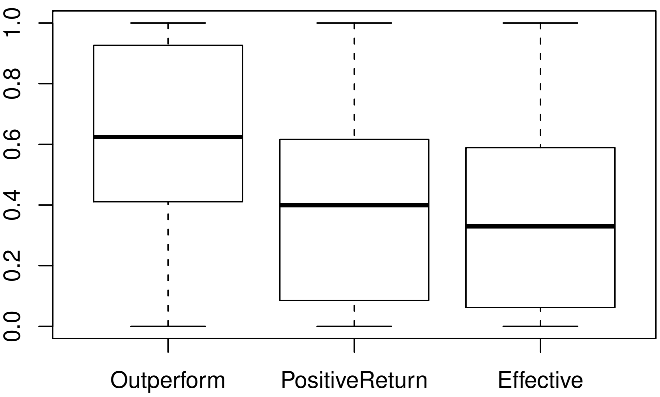


Fig. 7. Box plots of the distributions of the Outperform the Market (OM), Positive Return (PR) and EFfective (EF) metrics for case 4.

## IV. Conclusions and Future Work

This paper presented an analysis of two implications of the AMH from a computational intelligence perspective. The first was variable efficiency and whether the presence of non-linear dependence in a time-series offered any advantages for forecasting with supervised learning algorithms. The results clearly demonstrate that when non-linear dependence is present there is a statistically significant increase in the directional accuracy of the SL algorithms forecasts. This result was obtained using a simulated GARCH process but proves that if non-linear dependence can be reliably detected in a financial time-series then more accurate forecasts can be expected.

The second implication of cyclical profitability was shown to be quite abundant in the financial markets. Its more restrictive form, cyclical effectiveness, was also shown to be valid though not as abundant. This result demonstrates that trading models fitted to one time-period will have a non-zero probability of being effective again. The results also provide insight into overfitting and the information content in older previously learned models.

Future work concerns the development of a forecasting algorithm which can combine the signals produced by a population of optimized technical indicators to take advantage of cyclical profitability.

## References


[1] Matthew Butler and Dimitar Kazakov. Particle swarm optimization of bollinger bands. In *Proceedings of the 7th international conference on Swarm intelligence*, ANTS'10, pages 504–511, Berlin, Heidelberg, 2010. Springer-Verlag.

[2] Daniel O. Cajueiro and Benjamin M. Tabak. Ranking efficiency for emerging markets. *Chaos, Solitons and Fractals*, 22(2):349 – 352, 2004.

[3] Rachel Caspari. The evolution of grandparents. *Scientific American*, 305:44–49, August 2011.

[4] Andries Engelbrecht. Heterogeneous particle swarm optimization. In Marco Dorigo, Mauro Birattari, Gianni Di Caro, Ren Doursat, Andries Engelbrecht, Dario Floreano, Luca Gambardella, Roderich Gro, Erol Sahin, Hiroki Sayama, and Thomas Sttzle, editors, *Swarm Intelligence*, volume 6234 of *Lecture Notes in Computer Science*, pages 191–202. Springer Berlin / Heidelberg, 2010.

[5] Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2):383–417, May 1970.

[6] M. Hinich and D. Patterson. Detecting epochs of transient dependence in white noise. *Money, Measurement and Computation*, pages 61–75, 2005.

[7] Carlos M. Jarque and Anil K. Bera. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3):255 – 259, 1980.

[8] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942 –1948 vol.4, nov/dec 1995.

[9] Bjoern Krollner, Bruce Vanstone, and Gavin Finnie. Financial time series forecasting with machine learning techniques: A survey. In *European symposium on artificial neural networks: Computational and machine learning. Bruges, Belgium.Apr. 2010*. School of Information Technology at ePublications@bond, 2010.

[10] Ju-Sang Lee, Sangook Lee, Seokcheol Chang, and Byung-Ha Ahn. A comparison of GA and PSO for excess return evaluation in stock markets. In *IWINAC (2)*, pages 221–230, 2005.

[11] C. Lento and N. Gradojevic. The profitability of technical trading rules: a combined signal approach. *Journal of Applied Business Research*, 23(1):13–27, 2007.

[12] J. Leung and T. Chong. An empirical comparison of moving average envelopes and Bollinger Bands. *Applied Economics Letters*, 10(6):339–341, 2003.

[13] Kian-Ping Lim. Ranking market efficiency for stock markets: A nonlinear perspective. *Physica A: Statistical Mechanics and its Applications*, 376:445 – 454, 2007.

[14] Andrew W. Lo. The adaptive markets hypothesis: market efficiency from an evolutionary perspective. *Journal of Portfolio Management*, 30:15–29, 2004.

[15] Andrew W. Lo. Reconciling efficient markets with behavioral finance: the adaptive markets hypothesis. *Journal of Investment Consulting*, 7:21–44, 2005.

[16] J. Moody, L. Wu, Y. Liao, and M. Saffell. Performance functions and reinforcement learning for trading systems and portfolios. *Applied Financial Economics Letters*, 17:441–470, 1998.

[17] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 69–73, 1998.

[18] Alexandru Todea, Maria Ulici, and Simona Silaghi. Adaptive markets hypothesis - evidence from Asia-Pacific financial markets. *The Review of Finance and Banking*, 1(1):007–013, December 2009.

[19] Paul D. Yoo, Maria H. Kim, and Tony Jan. Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. In *CIMCA-IAWTIC'06*, pages 835–841, Washington, DC, USA, 2005. IEEE Computer Society.

[20] Luciano Zunino, Massimiliano Zanin, Benjamin M. Tabak, Daro G. Prez, and Osvaldo A. Rosso. Forbidden patterns, permutation entropy and stock market inefficiency. *Physica A: Statistical Mechanics and its Applications*, 388(14):2854 – 2864, 2009.