

Bond University
Research Repository



Enhancing stockmarket trading performance with ANNs

Vanstone, Bruce; Finnie, Gavin

Published in:
Expert Systems with Applications

DOI:
[10.1016/j.eswa.2010.02.124](https://doi.org/10.1016/j.eswa.2010.02.124)

Licence:
CC BY-NC-ND

[Link to output in Bond University research repository.](#)

Recommended citation(APA):
Vanstone, B., & Finnie, G. (2010). Enhancing stockmarket trading performance with ANNs. *Expert Systems with Applications*, 37(9), 6602-6610. <https://doi.org/10.1016/j.eswa.2010.02.124>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

For more information, or if you believe that this document breaches copyright, please contact the Bond University research repository coordinator.

9-1-2010

Enhancing stockmarket trading performance with ANNs

Bruce Vanstone

Bond University, bruce_vanstone@bond.edu.au

Gavin Finnie

Bond University, Gavin_Finnie@bond.edu.au

Follow this and additional works at: http://epublications.bond.edu.au/infotech_pubs



Part of the [Systems Architecture Commons](#)

Recommended Citation

Bruce Vanstone and Gavin Finnie. (2010) "Enhancing stockmarket trading performance with ANNs" *Expert systems with applications*, 37 (9), 6602-6610.

http://epublications.bond.edu.au/infotech_pubs/118

Enhancing Stockmarket Trading Performance with ANNs

Bruce Vanstone
Bond University
Gold Coast, Queensland,
Australia
bruce_vanstone@bond.edu.au

Gavin Finnie
Bond University
Gold Coast, Queensland,
Australia
gavin_finnie@bond.edu.au

ABSTRACT

Artificial Neural Networks (ANNs) have been repeatedly and consistently applied to the domain of trading financial time series, with mixed results. Many researchers have developed their own techniques for both building and testing such ANNs, and this presents a difficulty when trying to learn lessons and compare results. In a previous paper, Vanstone and Finnie have outlined an empirical methodology for creating and testing ANNs for use within stockmarket trading systems. This paper demonstrates the use of their methodology, and creates and benchmarks a financially viable ANN-based trading system. Many researchers appear to fail at the final hurdles in their endeavour to create ANN-based trading systems, most likely due to their lack of understanding of the constraints of real-world trading. This paper also attempts to address this issue.

1. INTRODUCTION

This paper uses the empirical methodology outlined by Vanstone and Finnie [1] to create and benchmark ANNs for use within stockmarket trading systems. For brevity, Vanstone and Finnie's approach will be referred to within this paper as the 'empirical methodology', and, for the sake of clarity, as little of that paper will be repeated as is necessary.

The objective of this paper is to provide a demonstration of the 'how-to' embodied in their empirical methodology, by leading the reader through the selection of inputs and outputs for the ANNs, the construction and testing of ANN architectures, and the final benchmarking of the ANN. It will then demonstrate how the ANN is used within a trading system, and will further demonstrate how to benchmark the final ANN-based trading system.

2. LITERATURE REVIEW

A detailed review of the types of analysis and variables used in stockmarket trading systems has already been presented in the empirical methodology. For this reason, this review will focus on the specific literature and variables which support the case study system to be developed in this paper.

Primarily, the inspiration for this case study trading system comes from the work of Guppy [2], which in turn springs from a solid base of research focused on the use of moving averages.

A brief review of the academic literature supporting the use of moving averages within trading systems follows.

2.1. MOVING AVERAGES

Moving averages have a history as long as Technical Analysis itself. The field of modern technical analysis dates from the work of Charles Dow, who in 1884 drew up an average of the daily closing prices of 11 important stocks. Between 1900 and 1902, Dow wrote a series of articles in the Wall Street Journal documenting stock price patterns and movements he observed in the average. These articles were the first to describe systematic phenomena in the stock markets.

The majority of the academic literature concerning technical analysis also concerns the testing of simple technical rules, such as moving averages.

According to Pring [3], there are three basic principles of Technical Analysis, namely:

- Prices move in trends,
- Volume goes with the trend,
- A trend, once established tends to persist

The moving average and its derivatives are designed to expose when a security has begun 'trending', and as such, deal with the first and third principles listed above. The idea of observing (and profiting from) trends has a long history, and is one of the core components of many present-day trading strategies.

Academic research in the area of moving averages dates from the work of Neftci and Policano [4], who studied moving averages, and the slope of price movements on the chart (named trendlines by technical analysts). They studied closing prices of gold and T-bills, and created buy-and-sell rules based on trendlines and moving averages. Although they described their results from the study of trendlines as inconclusive, they reported a significant relationship between moving average signals and prices. Of particular interest was the fact that a set of significant parameters for one commodity were often insignificant for another commodity. This difference in significant parameters is often termed a market's 'personality'.

Later, Neftci [5] examined the relationship of the 150 day moving average rule to the Dow-Jones Index. This research concluded that the moving average rule generated Markov times (no dependence on future information) and has predictive value.

Two popular technical trading rules were tested by Brock et al. [6], namely, moving averages and trading

range breaks (known by technical analysts as Support and Resistance trading). Using data from the start of the DJIA in 1897 to the last trading day of 1986, the authors test a variety of combinations of moving averages, using a 1% band around predictions to eliminate whipsaws. They find support for the use of moving averages, and report that the differences in utility are not readily explained by risk. They conclude their results are consistent with the technical rules tested having predictive power.

Inspired by Brock et al [6] above, Mills [7] tests the same two trading rules in the London Stock Exchange, using FT30 data from 1935 – 1994. Mills' results are remarkably similar to Brocks, with Mills concluding that the trading rules could predict stock prices, and are thus profitable in periods when the market is inefficient.

Levich and Thomas [8] test currency futures contracts in five currencies over the period 1976 to 1990. They report persistent trading profits over the 15 year period using a variety of commonly researched moving average rules. Levich and Thomas concluded 'the profitability of trend following rules strongly suggest some form of serial dependency in the data, but the nature of the dependency remains unclear'.

LeBaron [9] provided more support for moving averages, by using moving average rules as specification tests for foreign exchange rates. He concluded that exchange rates do not follow the random walk, and that the deviations are detected by simple moving average rules.

2.2. TRADING SYSTEMS

According to Chande [10], a trading system consists of three major functions, namely:

- Rules to enter and exit trades,
- Risk Control, and,
- Money Management

Each of these functions is further described below.

2.2.1. RULES TO ENTER AND EXIT TRADES

The case study trading system to be developed is based on the work of Guppy [2, 11, 12], and uses his GMMA as a simple mechanical signal generator.

ANNs will be trained in support of the GMMA driven signal. The ANNs will be trained to forecast the likely strength of price movement, and will therefore provide an additional level of confidence in the signals used for initiating or exiting trading positions.

The GMMA (Guppy Multiple Moving Average) is defined as:

$$\left(\begin{array}{l} ema(3) + ema(5) \\ + ema(8) + ema(10) \\ + ema(12) + ema(15) \end{array} \right) - \left(\begin{array}{l} ema(30) + ema(35) \\ + ema(40) + ema(45) \\ + ema(50) + ema(60) \end{array} \right)$$

Equation 1 GMMA definition

Where $ema(n)$ is the n -period exponential moving average of closing prices.

For the case study system developed in this paper, rules to enter and exit trades are based on the combination of the GMMA signal and the strength of the ANN output signal.

2.2.2. RISK CONTROL

In the context of stock market trading, a trader is typically concerned with downside risk, which describes how much money is at risk on an individual trade-by-trade basis. This method of approaching risk leads to traders placing orders to sell/buy securities to cover open long/short positions when losses cross pre-determined thresholds. These are known as stop-loss orders.

As investors are typically preoccupied with return, it is also appropriate to consider risk to be appropriately controlled by trade risk within the confines of a trading system. After all, this is the entire purpose of a trading system. This method of considering risk is growing in popularity, see for example Kaufman [13], Longo [14], and Pocini [15].

A general framework for considering the issue of risk control is the TOPS COLA approach described by Chande [10]. TOPS COLA is an acronym for "take our profits slowly, cut off losses at once". In effect, it describes the traders approach to risk.

Trend following systems, particularly those based on moving averages, will typically have more losing trades than winning trades. In financial terms, this still leads to a viable system, as long as the value of losing trades is quite low, and/or the value of winning trades is high. Typically, according to Chande, about 5% of the trades made by a trend following system are the 'big ones'. In light of this information, it is easy to see how the TOPS COLA approach can work.

A detailed analysis of stop-setting methods is provided in the empirical methodology.

The stop-loss threshold in this implementation is selected by the study of the in-sample MAE as described by Sweeney [16], and later by Tharp [17]. The MAE studies the Maximum Adverse Excursion (MAE) of a set of trades, in an effort to determine the extent to which favorable (profitable) trades range into unprofitable territory before closing out profitably. This method of risk management allows traders to study the MAE characteristics of a set of trades, to identify preferred stop-loss points.

2.2.3. MONEY MANAGEMENT

Money management, aka position sizing, refers to the actual size of the trade to be initiated, taking into consideration the account equity and potential trade risk.

To simplify the complexities of Money Management, this paper suggests using a fixed percentage of equity per

trade (as suggested by Elder [18]) for testing and benchmarking. Not only is this simple to implement, but it also avoids having to determine how much of any profit effect observed is attributable to the neural network developed, and how much is attributable strictly to money management. Given the goal of this paper, this choice seems appropriate.

More advanced choices for money management, such as Risk Position Sizing, are excellent areas for future work. A summary of money management strategies is provided in the empirical methodology.

3. NEURAL NETWORK CREATION

The ANNs in this paper are being trained to provide a price movement strength forecast, to support the primary GMMA signals.

3.1.1. CHARACTERISTICS OF GMMA-BASED TRADING SYSTEMS

The trading system to be developed in this paper will have the following characteristics:

1. Medium-term timeframe: position duration will be measured in weeks and months,
2. Market orders: positions will be acquired using t+1 market orders, ie. the trading system being developed is an eod (end-of-day) trading system, where orders are placed prior to next days market open.

3.1.2. CHARACTERISTICS OF DATA & TOOLS

This paper uses data for the ASX200 constituents of the Australian stockmarket. Data for this study was sourced from Norgate Investor Services [19]. For the in-sample data (start of trading 1994 to end of trading 2003), delisted stocks were included. For the out-of-sample data (start of trading 2004 to end of trading 2008) delisted stocks were not included. The ASX200 constituents were chosen primarily for the following reasons:

1. The ASX200 represents the major component of the Australian market, and has a high liquidity – a major issue with previous published work is that it may tend to focus on micro-cap stocks, many of which do not have enough trading volume to allow positions to be taken, and many of which have excessive bid-ask spreads,
2. This data is representative of the data which a trader will use to develop his/her own systems, and is typical of the kind of data the system will be used in for out-of-sample trading

It is important to train ANNs on data which includes delisted securities, to enable the neural network access to data which described the real world environment.

Software tools used in this paper include Wealth-Lab Developer, and Neuro-Lab, both products of Wealth-Lab Inc (now Fidelity) [20].

3.1.3. SELECTING INPUTS

Input variables need to be selected which can be expected to have some influence in the given timeframe.

Considering the desired timeframe for this case study is measured in weeks and months, it is likely that technical variables will be most appropriate. A great deal of published research is presented in the empirical methodology paper which supports the use of the following set of inputs for forecasting price return strength. The function profiles for these variables are discussed in detail in Vanstone [21].

The inputs chosen are:

1. EMA(close,3) / EMA(close,30)
2. EMA(close,15) / EMA(close,60)
3. HPR
4. LPR
5. SMA(volume,3) / SMA(volume,15)
6. ATR(3) / ATR(15)
7. ADX(3)
8. ADX(15)
9. STOCHK(3)
10. STOCHK(15)
11. RSI(3)
12. RSI(15)
13. MACD

Selected statistical properties of these variables (from the in-sample dataset) follows:

Variable	Min	Max	Mean	StdDev
1	0.85	2.04	1.04	0.06
2	0.84	1.91	1.04	0.06
3	0.07	1.00	0.89	0.13
4	0.02	1.00	0.72	0.17
5	0.01	1.67	0.99	0.32
6	0.00	3.71	1.00	0.30
7	3.59	100.00	53.44	19.54
8	6.05	99.71	25.13	10.77
9	0.00	100.00	54.56	36.62
10	0.00	100.00	65.02	27.73
11	0.43	100.00	58.08	24.99
12	32.70	98.03	58.64	8.46
13	-0.11	7.14	0.06	0.15

Table 1 Basic Statistical properties of in-sample variables

The formulas used to compute these variables are standard within technical analysis, except for LPR and HPR, which are also defined in Vanstone [21].

There is no well-defined set of inputs which suit all occasions, and it is important for the researcher to continually study published research and create function profiles to assess the suitability of likely input variables.

There is also no reason to assume that these inputs are likely to be the best for this purpose, they are simply culled from previous research by the same authors. There is a great deal of published academic and practitioner research which can be used to help refine the search for relevant variables; it is comprehensively reviewed in the empirical methodology paper.

3.1.4. SELECTING OUTPUTS

Again considering the timeframe is measured in weeks and months, it is important that the output forecast be for a similar period of time.

Essentially, there is no ‘correct’ timeframe to use. However, as a choice must be made for implementation, the forecast period was chosen as 20 days (about 1 trading month). Smaller or larger timeframe values which are consistent with the desired trading timeframe would also be appropriate choices.

$$\left(\frac{(\text{highest}(close_{i+20\dots i+1}) - close_i)}{close_i} \right) \times 100$$

Equation 2 Calculation of output variable

The calculation of the return variable allows the ANN to focus on the highest amount of change that occurs in the next 20 days, which may or may not be the 20-day forward return. For example, the price may spike up after 5 days, and then decrease again, in this case, the 5-day forward price would be used. Therefore, perhaps a better description of the output variable is that it is measuring the maximum amount of price change that occurs within the next 20 days.

The basic statistical properties of the output target variable follow:

Variable	Min	Max	Mean	StdDev
Target	0.00	100.00	10.48	24.84

Table 2 Basic Statistical properties of in-sample output target variable

3.1.5. PARTITIONING AVAILABLE DATA

For training and testing an ANN, data needs to be logically (or physically) partitioned into a minimum of 2 sets, a training set and a testing set. In essence, the main principle is to capture as much diverse market activity as possible (with a long training window), whilst keeping as long a testing window as possible (to increase shelf life and model confidence). This issue is discussed in detail in the empirical methodology. This paper splits the data into the following two sets:

Data from 1994 up to and including 2003 (in-sample) is used to predict known results for the out-of-sample period (from 2004 up to the end of 2008).

In this study, only ordinary shares are considered.

3.1.6. IN-SAMPLE BENCHMARKS

As explained in the empirical methodology, a number of hidden node architectures need to be created, and each one benchmarked against the in-sample data.

The method used to determine the hidden number of nodes is described in the empirical methodology. After the initial number of hidden nodes is determined, the first ANN is created and benchmarked. The number of hidden nodes is increased by one for each new architecture then created, until in-sample testing reveals which architecture has the most suitable in-sample metrics.

The empirical methodology uses the filter selectivity metric for longer-term systems, and Tharp’s expectancy [17] for shorter term systems. This paper also introduces the idea of using absolute profit per bar for medium term systems. This method assumes unlimited capital, takes every trade signaled, and measures how much average profit is added by each trade over its lifetime. This figure is then refined to the amount of profit added by trades on a daily basis.

3.1.7. DETERMINING ARCHITECTURE

A detailed review of the methods available for determining ANN architecture is provided in the empirical methodology.

This paper uses an approach described by Tan [22], which is to start with a small number of hidden neurons and increase the number of hidden neurons gradually. Tan’s procedure begins with 1 hidden layer, containing the square root of N hidden nodes, where N is the number of inputs. Training the network takes place until a pre-determined number of epochs have taken place without achieving a new low in the error function. For example, ANNs can be trained until no new low had been achieved for at least 2000 epochs. At this point the network would be tested against the in-sample set, and benchmarked using the appropriate in-sample metric described above. A new neural network is now created with the number of hidden nodes increased by 1, and the training and in-sample testing is repeated. After each test, the metric being used for benchmarking is assessed, to see if the new network configuration is superior. This process continues while the networks being produced are superior, that is, it terminates at the first network produced which shows inferior in-sample results.

This approach to training is an implementation of the early stopping method, which aims to preserve the generalization capabilities of neural networks. It is based on the observation that validation error normally decreases at the beginning of the training process, and begins to increase when the network starts to over-fit. Lack of generalization is caused by over-fitting. In an over-fit (over-trained, over-learned) situation, the network begins to memorize training examples, and loses the ability to generalize to new situations.

For this case study, ANNs were trained using the selected inputs and the architecture methodology described above.

3.1.8. SETTING SIGNAL THRESHOLDS

Each neural network developed will fit itself to the characteristics of the market which the training data represents, within the constraints of its architecture. A simple way to observe this fit is with the use of a function profile. From inspection of the function profiles for each neural network, the threshold at which the neural network output signal begins to signal profitable trades can be easily established.

Therefore, for the in-sample testing, the buy signal should take account of the individual neural networks threshold, and also take account of whether the signal is increasing in strength, or decreasing in strength from its previous forecast. Naturally, the sell signal should also take account of the threshold, and also take account of whether the signal is increasing in strength, or decreasing in strength from its previous forecast. It is also often considered a desirable property of a trading system if the rules for exiting a trade are the contra to the rules for entering it.

Therefore, a general buy and a general sell rule can be explicitly stated, and then applied to each trading system. Where x is the signal strength threshold chosen from the function profile, then the entry and exit rules become:

Buy: Buy tomorrow when neural signal output(today) > x , and neural signal output(today) > neural signal output(yesterday)

Sell: Sell tomorrow when neural signal output(today) ≤ x , and neural signal output(today) < neural signal output(yesterday)

These simple buy and sell rules take account of the threshold signal strengths, and using the same generic buy and sell rule for each network gives greater confidence of the generalization of the results.

This paper suggested that for each neural network, the output is a signal strength rating, scaled between 0 and 100. It is then to be expected that, in general, as the numeric value of the signal increases, so should the expected returns to this signal strength. This general principle can be seen by examining a function profile of the signal output of each neural network. The function profile for the 4 hidden node ANN is presented below.

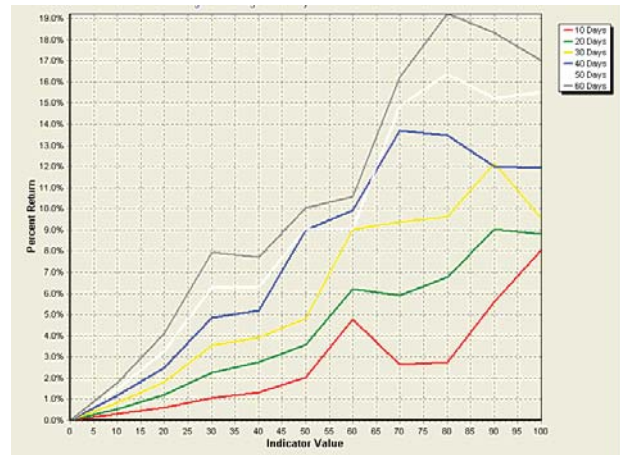


Figure 1 Function Profile for 4 hidden node ANN

The function profile clearly reveals that the ANN signal strength is rising as the actual percentage returns are rising. What the function profile doesn't reveal is the number of actual observations for each indicator value. It is important that there be a reasonable number of observations at any levels that the trading system is likely to rely on. The information necessary to make this judgement is shown in Table 3 below. It is clear that the number of observations falls away rapidly after the cutoff value of 40, hence, a trading system using this ANN would only trade when the signal strength is between 0 and 40.

Function Range	Observations	% Return
Overall	645,710	0.4643
0 – 10	476,553	0.2950
11 – 20	85,782	0.5870
21 – 30	36,992	1.0589
31 – 40	44,157	1.3173
41 – 50	509	2.0466
51 – 60	284	4.7836
61 – 70	212	2.6353
71 – 80	141	2.7292
81 – 90	206	5.6124
91 – 100	874	8.0398

Table 3 Number of observations at each function level - 4 hidden node ANN

Approximately two-thirds of the observations are in the range 0 – 10 (476553 observations) out of a total 645710 observations. The average return over the total observations is 0.4643%. The average return of the majority of the observations (range 0 – 10) is 0.2950%, below the overall average. The average return of every other range is higher than the overall observation average of 0.4643%. For this reason, the in-sample cutoff value for the signal threshold for this particular ANN was chosen as 10.

Similar analysis can be applied to the Function Profiles and Number of observations data for the 5 hidden node ANN, and the 6 hidden node ANN, shown in Figure 2 and Figure 3, and Table 4 and Table 5 below.

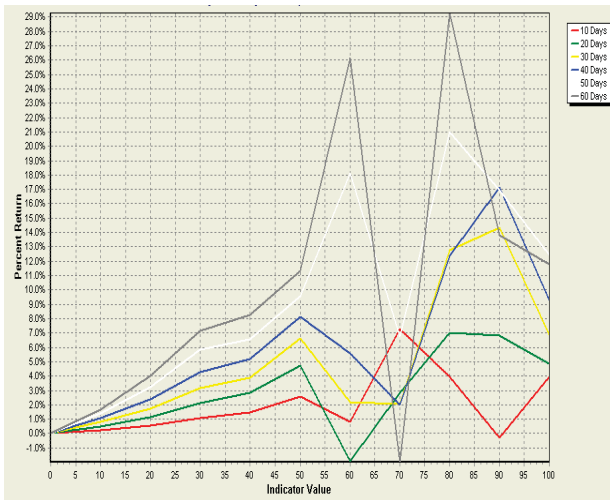


Figure 2 Function Profile for 5 hidden node ANN

Function Range	Observations	% Return
Overall	645,710	0.4643
0 – 10	417,169	0.2540
11 – 20	178,384	0.5705
21 – 30	16,629	1.1047
31 – 40	11,534	1.4592
41 – 50	21,477	2.5584
51 – 60	30	0.8429
61 – 70	27	7.2624
71 – 80	19	3.9233
81 – 90	28	-0.3063
91 – 100	413	3.9755

Table 4 Number of observations at each function level - 5 hidden node ANN

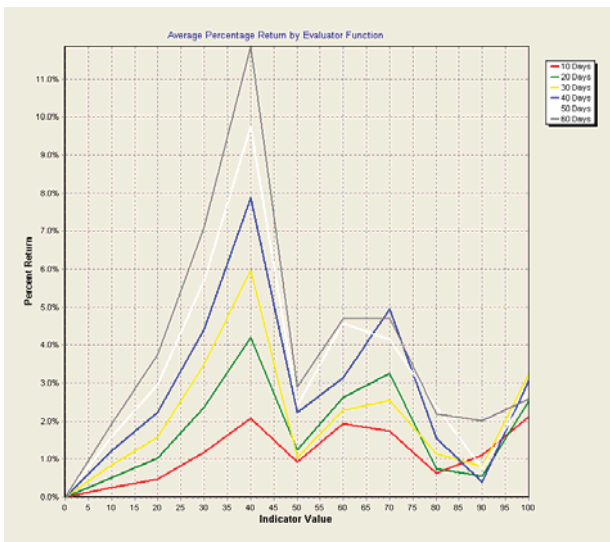


Figure 3 Function Profile for 6 hidden node ANN

Function Range	Observations	% Return
Overall	645,710	0.4643
0 – 10	454,526	0.2618
11 – 20	120,129	0.4651
21 – 30	22,649	1.1794
31 – 40	25,778	2.0739
41 – 50	886	0.9284
51 – 60	657	1.9351
61 – 70	565	1.7286

71 – 80	631	0.6400
81 – 90	753	1.1103
91 – 100	19,136	2.1136

Table 5 Number of observations at each function level - 6 hidden node ANN

The following table shows the architecture (number of hidden neurons) and the relevant in-sample benchmarks, as well as the same benchmark values for the buy-and-hold naïve approach, and the non-ANN GMMA based approach.

Approach	No of Trades	Profit per bar (day)
Naïve Buy-and-Hold	362	\$4.55
Non-ANN GMMA	11,690	\$3.63
4 Hidden node ANN	6,532	\$6.32
5 Hidden node ANN	4,862	\$7.76
6 Hidden node ANN	5,570	\$7.54

Table 6 In-sample benchmarks

From the analysis of the data provided in Table 6 and Table 4, it is clear that the 5 hidden node ANN should be selected as the winning architecture from in-sample testing.

This is the ANN which will be selected to continue forward to out-of-sample testing.

3.1.9. SETTING TRADING STOPS

In this paper, the MAE technique discussed in the empirical methodology is used to set trading stops. This technique can be used to identify an appropriate stop-loss percentage for the in-sample set of trades. This stop-loss percentage is then used to control trading risk for the out-of-sample trades.

By building a histogram of the actual (in-sample) trade data, split according to trades that were eventually won (were profitable), and trades that were eventually lost (were unprofitable), a visual inspection can be made of a useful stop threshold. This information is very valuable to a trader, as it also gives an indication of how the profit/loss percentages will be affected when the stop is introduced. In this approach, the stop percentage value determined from the in-sample data will be then used as the stop value in the out-of-sample testing data.

Figure 4 shows a detailed histogram of the MAE of the set of trades harvested from the in-sample data, as selected by the 5 hidden node ANN architecture. The simulations used to produce this histogram assume an unlimited amount of trading capital, and a fixed investment of \$10,000 per trade. This assumption is necessary to ensure the histogram shows all trades which have been signalled by the ANN.

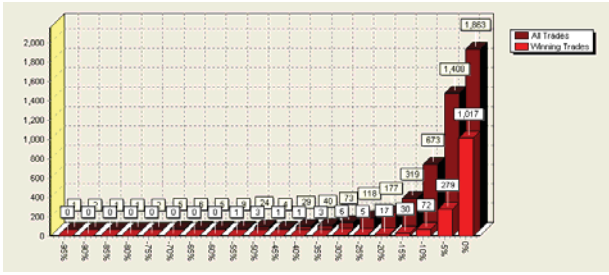


Figure 4 MAE histogram for 5 hidden node architecture

The histogram shows the maximum adverse excursions of the trades, and also the number of trades at that level which went on to become profitable. For example, 673 trades lost 10%, of which 72 went on to become profitable.

Typically, the stop level is chosen by ‘eye-balling’. The goal is to select a stop loss threshold which balances the number of trades straying into adverse territory and not recovering, with those straying into adverse territory yet still recovering to a profitable conclusion. From the MAE histogram, values of either 5% or 10% would seem appropriate. Remembering the advice of Chande [23], it is wiser to err on the side of a wider stop than a tight stop. For this reason, the value of 10% is chosen.

4. REAL-WORLD CONSTRAINTS

All trades initiated from end-of-day data must be day+1 long market orders. This means that after a signal is given, then the trade takes place on the next day the market is open, at market open price. For example, after the market has closed on day t , the trading system would be run, and any buy (sell) signals generated are queued for opening positions (closing positions) for the start of the next days trading, day $t+1$. In this way, there is no possibility of acting on information which is not publicly available to all traders. In essence, this is similar to the issue of displacing fundamental data by at least 6 months, again, to ensure that the trading system is not being tested on data which was not available in the market.

All trading simulations must account for transaction costs, and it is advised that these be over-estimated for historical testing. Traditionally, the cost of brokerage for retail traders has been falling, therefore, using today's transaction costs to simulate historical trading results as of 10 years ago is very misleading, particularly if the strategy being tested generates a large number of trades. It is reasonable to compensate for cost reductions by inflating the transaction costs for the entire simulation. In this way, the bias is overestimated against the trader.

Another realistic simulation constraint is slippage. Although a trade may be initiated at market open, this does not mean the trade will be opened (closed) at market open price. There will inevitably be slippage due to the fact that at market open there may be a great many trades scheduled. Naturally, the price can move around quite considerably in the early part of trading, and slippage is the method to account for this cost. Slippage settings of 0.5% would be reasonable.

It is also important when developing and benchmarking systems of this type that simulations respect volume constraints. It is not realistic to assume that there is an unlimited amount of stock available for purchase. Historical technical data includes the volume data item. When training and testing, it is realistic to assume that the positions sizes acquired be some smallish factor of the overall trade volume available. A suitable factor might be 5% - 10 %, or perhaps even more dependant on the market cycle. Depending on the type of market behaviour being exploited, the amount of stock available to buy in the market may be less than the traders desirable position size. In this case, tests need to be run to ensure the required line of stock can be acquired within a realistic timeframe. In the case of slow gaining, moving average style systems, this rarely presents a problem.

Finally, it is unwise in historical simulations to refer directly to cut-off values for variables such as price. For example, it would be unrealistic to include a condition that price must be less than \$5 to initiate a trade. Historic price data is adjusted for splits etc, therefore, historically a price may be shown as \$5, but at the actual date that stock was traded in the market, it could well have been a very different price.

5. BENCHMARKING

Once the appropriate in-sample architecture has been decided, the architecture and training must be frozen, and the network can proceed to out-of-sample benchmarking. At the same time, all the parameters of signal strength threshold, stop-loss threshold and money management values used in the in-sample testing must also be frozen for out-of-sample benchmarking.

For the case study system developed in this paper, the relevant information is as follows:

Parameter	Value
ANN chosen	5 Hidden Node architecture
Signal threshold	10
Stop-loss threshold	10% initial stop
Money Management value	5% equity per trade

Table 7 Parameters for out-of-sample system

A detailed discussion of trading metrics is presented in the empirical methodology. In this paper, Table 8 presents the values for each of the out-of-sample metrics, for both the case study system, and the buy-and-hold benchmark. For further reference on appropriate values for the metrics, and their exact construction and interpretation, the reader should consult the empirical methodology paper.

It should be remembered that the factors which determine whether a system is acceptable or not are ultimately the choice of the trader. No system should be chosen if it displays undesirable characteristics; however, individual traders would differ on their choice of system, dependant on such issues as their tolerance to

risk, their amount of starting capital, and their trading horizon.

Metric	5 Hidden node ANN	Buy-and-Hold benchmark
Net Profit (%)	338.10%	12.60%
Annualized Gain (%)	34.37%	2.40%
Number of Trades	446	1 (index)
Exposure (%)	91.37%	100.00%
Winning Trades (%)	29.60%	100.00%
Average Profit (%)	10.21%	12.77%
Losing Trades (%)	70.40%	0.00%
Average Loss (%)	7.28%	N/A
Max. Drawdown (%)	50.91%	50.58%
Profit Factor	1.59	N/A
Recovery Factor	0.79	0.12
Payoff Ratio	7.12	N/A
Sharpe ratio	1.12	0.22
Ulcer Index	14.64	13.22
Luck Coefficient	12.53	N/A
Pessimistic Rate of Return	2.59%	0.00%
Equity Drop Ratio	0.18	2.47

Table 8 Trading System Metrics

Once out-of-sample benchmarking has been completed, the trader has a realistic model of the trading system, which can then be realistically assessed. From this model, the trader can make accurate judgements about whether this particular trading system meets the trader's specific individual trading requirements.

Consistency is one of the most important areas for a trader to focus on, and the level of system consistency can be determined by comparing the figures for the in-sample model to the figures for the out-of-sample model. Clearly, the smaller the amount of variation between the two models, the greater the likelihood that the neural network has captured the generalities of the profit-generating phenomena. However, it is to be expected that there will be some differences between the values from in-sample and out-of-sample testing. Generally, these can be explained by observation of the market cycle. Where this is not the case, the trader must treat the finished model with caution. This is because an accurate model also serves another purpose; it gives the trader guidelines within which to operate.

Table 9 and Table 10 allow the trader to compare the relative consistency of the behaviour of the 5 Hidden Node architecture with each of the non-neural approaches. In essence, the trader can see how consistently the non-neural approaches have performed out-of-sample, and use this information to judge how consistently the neural approach has performed out-of-sample.

Approach	No of Trades	Profit per bar (day)
Naïve Buy-and-Hold	362	\$4.55
Non-ANN GMMA	11,690	\$3.63
5 Hidden node ANN	6,532	\$6.32

Table 9 In-Sample benchmarks

Approach	No of Trades	Profit per bar (day)
Naïve Buy-and-Hold	200	\$9.54
Non-ANN GMMA	3,941	\$9.26
5 Hidden node ANN	1,651	\$19.08

Table 10 Out-of-Sample comparison figures

Should the trader decide to trade using this model, then it will be clear going forward whether the model is operating within the expected guidelines, and, more importantly, it will give early warning if the model unexpectedly deviates from expectations. This could happen if some underlying characteristic of the market changed, and it is important for the trader to realize this as soon as possible.

It is clear that the ANN developed as a case study in this paper has outperformed the market in spectacular style. However, it is also clear that some underlying mechanics of the market have changed during the development of the Financial Crisis, and this ANN is no more immune from these changes than any other trading approach.

This change in market behaviour demonstrates one of the most pronounced benefits to developing a computational model. It can be determined in advance when the model is deviating from its expected behaviour. At this point, the trader can determine whether to suspend trading using the model, until it returns to its expected behaviour. As most traders are focused on drawdown, one simple way to determine whether the model is acting within expected tolerance is to predefine a level of drawdown at which the trader decides to withdraw funds from the market. For most traders, this is an extension of the stop-loss mechanism they use to manage their individual trades.

6. CONCLUSION

This paper has worked through a case study of the methodology for designing and testing stockmarket trading systems using soft computing technologies, specifically artificial neural networks, which was developed by Vanstone and Finnie [1].

For other examples of the Vanstone & Finnie methodology in practice, the reader may wish to pursue other papers written by the authors in this area, for example Vanstone et al. [24, 25]. These papers step through the process of selecting input variables, designing artificial neural networks for trading, and benchmarking of the trading results.

This methodology presented clearly separates the in-sample process of training neural networks and selecting parameters from the out-of-sample benchmarking process. It also aims to ensure that if the neural models developed during the in-sample training process are curve-fit, then that is clearly exposed during the out-of-sample benchmarking. This process of breaking up the development into a number of discrete, testable steps provides another advantage – it allows the developer to focus on correcting a specific part of the process if and when things go wrong.

The objective of developing viable mechanical stockmarket trading systems based on technologies such as neural networks is achievable. The key is to conduct the development process within a well-defined methodology, and as close to real-world constraints as possible.

REFERENCES

1. Vanstone, B. and G. Finnie, *An Empirical Methodology for developing Stockmarket Trading Systems using Artificial Neural Networks*. Expert Systems with Applications, 2009. 36: p. 6668-6680.
2. Guppy, D., *Trend Trading*. 2004, Milton, QLD: Wrightbooks.
3. Pring, M.J., *Martin Pring's Introduction to Technical Analysis*. 1999, Singapore: McGraw-Hill.
4. Neftci, S.N. and A.J. Policano, *Can Chartists outperform the Market? Market Efficiency tests for 'Technical Analysis'*. Journal of Futures Markets, 1984. 4(4): p. 465-478.
5. Neftci, S.N., *Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of 'Technical Analysis'*. Journal of Business, 1991: p. 549-571.
6. Brock, W., J. Lakonishok, and B. LeBaron, *Simple Technical Trading Rules and the Stochastic Properties of Stock Returns*. Journal of Finance, 1992. 47(5): p. 1731-1764.
7. Mills, T.C., *Technical Analysis and the London Stock Exchange: Testing trading rules using the FT30*. International Journal of Finance and Economics, 1997. 2: p. 319-331.
8. Levich, R. and L. Thomas, *The significance of Technical Trading Rule Profits in the Foreign Exchange Markets: A Bootstrap Approach*. Journal of International Money and Finance, 1993. 12: p. 451-474.
9. LeBaron, B., *Technical Trading Rules and Regime shifts in Foreign Exchange*, in *Advanced Trading Rules*, E. Acar and S. Satchell, Editors. 1997, Butterworth Heinemann.
10. Chande, T.S., *Beyond Technical Analysis: how to develop and implement a winning trading system*. 1997, New York: Wiley.
11. Guppy, D., *Exploiting Positions with Money Management*, in *Technical Analysis of Stocks and Commodities*, J.K. Hutson, Editor. 1999, Technical Analysis Inc.: Seattle, WA. p. 381-391.
12. guppytraders.com. *Guppy Multiple Moving Average*. [cited 04-05-2007]; Available from: www.guppytraders.com/gup329.shtml.
13. Kaufman, P.J., *Trading Systems and Methods*. Wiley Trading Advantage. 1998, New York: Wiley.
14. Longo, J.M., *Selecting Superior Securities: Using Discriminant Analysis and Neural Networks to differentiate between 'Winner' and 'Loser' stocks*, in *UMI Dissertation Services Number 9601890*. 1996, Rutgers University. Graduate School - Newark.
15. Pocini, M., *Momentum Strategies applied to Sector Indices*. Journal for the Colleagues of the International Federation of Technical Analysts, 2004. 2004 Edition: p. 3-10.
16. Sweeney, J., *Maximum Adverse Excursion: analyzing price fluctuations for trading management*. 1996, New York: J. Wiley.
17. Tharp, V.K., *Trade your way to Financial Freedom*. 1998, NY: McGraw-Hill.
18. Elder, A., *Trading for a Living*. 1993: John Wiley & Sons.
19. *Norgate Premium Data*. 2004 [cited 01-01-2004]; Available from: www.premiumdata.net.
20. *Wealth-Lab*. 2005 [cited; Available from: www.wealth-lab.com].
21. Vanstone, B., *Trading in the Australian stockmarket using artificial neural networks*. 2006, Bond University.
22. Tan, C.N.W., *Artificial Neural Networks: Applications in Financial Distress Prediction and Foreign Exchange Trading*. 2001, Gold Coast, QLD: Wilber Press.
23. Chande, T.S., *Beyond Technical Analysis (2nd Edition): How to develop and implement a winning trading system*. 2nd ed. 2001: John Wiley & Sons.
24. Vanstone, B., G. Finnie, and C.N.W. Tan. *Applying Fundamental Analysis and Neural Networks in the Australian Stockmarket*. in *International Conference on Artificial Intelligence in Science and Technology (AISAT 2004)*. 2004. Hobart, Tasmania.
25. Vanstone, B., G. Finnie, and C.N.W. Tan. *Evaluating the Application of Neural Networks and Fundamental Analysis in the Australian Stockmarket*. in *IASTED International Conference on Computational Intelligence (CI 2005)*. 2005. Calgary, AB, Canada: ACTA Press.