Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

“Build Your Own Trading Robot”

M.TECH Software Engineering (5YEARS)—2016

Soft Computing-SWE1011

Review 3

Submitted By

K.V.PRASANNA KUMAR-16MIS0254

M SAI DEEPU-16MIS0133

MALLELA NIKHIL -16MIS0256

FOR PRACTICAL FULFILLMENT OF THE INTEGRATED COURSE/FALL SEMESTER-2018

SLOT- **B2+TB2**

FACULTY IN CHARGE

PROF Senthil Kumar P

Department of

School of Information Technology

ABSTRACT:

An algorithmic trading robot is a computer code that has the ability to generate and execute buy and sell signals in financial markets. The neural network concept which we are going to use in algorithmic trading robot is **Adaline.** In adaline we can see there is only one input node nad output node.The main components of such a robot include entry rules that signal when to buy or sell, exit rules indicating when to close the current position, and position sizing rules defining the quantities to buy or sell. Giving the user workspace to work all time. Being human it is not possible to work relentlessly on the other hand, trading is a system where the tables can turn upside down in a fraction of seconds, the implementation of the automatic robot software can turn the tables in to our favour. Hence, this automated trading approach eases the way of trading and the proper logic can fetch a lot in terms of the trading perspective.

Introduction:

ADALINE (**Adaptive Linear Neuron** or later **Adaptive Linear Element**) is an early singlelayer artificial neural network and the name of the physical device that implemented this network. The network uses memistors. It was developed by Professor Bernard Widrow and his graduate student Ted Hoff at Stanford University in 1960. It is based on the McCulloch– Pitts neuron. It consists of a weight, a bias and a summation function.

The difference between Adaline and the standard (McCulloch–Pitts) perceptron is that in the learning phase, the weights are adjusted according to the weighted sum of the inputs . In the standard perceptron, the net is passed to the activation function and the function's output is used for adjusting the weights.

Technology has played an increasingly important role in the development of securities markets since the 1990s. It was readily embraced in the backend functions of clearing and settlement at exchanges, but it has played a more controversial role in the trading process. Earlier, in the 1970s, there was much debate about moving from open outcry markets to electronic limit order book markets. The latter became accepted as the dominant form of trading only in the last decade. A similar controversy now marks the debate on the role of algorithmic trading in exchanges, where computer algorithms directly place orders to trade. Policy makers, who largely encouraged the use of technology by mandating best execution practices for investors in the 1990s, are now exploring interventions to curb high frequency trading, in the 2010s. How algorithmic trading (AT) affects the quality of securities markets has been extensively analysed previously. These analyses, however, faced challenges in establishing causal linkages between changes in AT and changes in market quality (Biais and Foucault, 2014). Using a novel dataset and market setting, we set out to address some of these challenges. One of the abovementioned challenges is the preponderance of fragmented trading. In markets such as those in the U.S. which is the focus of most of the research work in this field, trading takes place at numerous venues, each with varying market access and microstructure. This makes it hard to understand the causal impact of any single microstructure feature, such as algorithmic trading, on any one trading venue. In contrast, the setting in this paper is the National Stock Exchange in India, where most of equity spot trading and all the derivatives trading is concentrated at a single exchange, for the duration of the analysis. A second challenge is the lack of clear identification of orders and trades as being generated by algorithms. Much of the existing research is based on proxies of algorithmic trading which leads to weak identification (Hendershott et al., 2011; Hasbrouck and Saar, 2013). Where there is better identification, the datasets are restrictive. Either the sample of securities is limited, or the period under study is short (Hendershott and Riordan, 2013). In contrast, the data in this paper has every order, and the counter-party order on every trade, flagged by the exchange as being AT or not, for all the securities that traded, for five years. A third challenge is in establishing causality. The problem of endogeniety 3 arises because other unobserved factors can be the common cause for high algorithmic trading and high levels of market quality on a security. This paper has three advantages in establishing a causal link between AT and changes in market quality. The first advantage is an exogenous identification event when the exchange commissioned co-location facilities (co-lo). Such an event directly affects the level of algorithmic trading, but not market quality. The second advantage is wide span of data which permits the use of matching techniques to select a sample of dates in the pre co-lo and post co-lo periods that have similar macroeconomic conditions. This ensures comparability without requiring assumptions about functional forms to be used as regression-style controls. The last advantage is the comprehensive coverage on the securities traded on the exchange, that can be used to control for endogeniety bias. A propensity score matching algorithm is used to identify pairs of securities that are matched on firm characteristics such as size, price and returns volatility but differ on the amount of AT. The securities which have a large change in the level of AT activity after co-lo are the treated group. The control group are securities with AT activity that was similar to the treated security before co-lo, but did not show a significant change in AT activity after co-lo.

A difference-in-differences regression is used to estimate the change in market quality of the treated relative to control securities. Any significant differences between the two can be attributed to AT. The estimated coefficients show that, on average, higher AT causes better market quality. This includes lower impact costs, larger number of shares available for trade, lower imbalance between the number of shares available to buy and sell, and a sharp drop in price volatility. The depth (measured by the monetary value available to trade) is not significantly affected by higher AT at the touch (best bid and offer). This paper adds new evidence to the literature about the causal impact of AT on the stability of market price and liquidity. Policy makers and regulators often voice concerns that the higher level of liquidity is transient because AT exits the market rapidly when there is unexpected news. Their main criticism is that AT causes a higher probability of extreme drops and reversals over a very short period of time during the trading day. The estimates in this paper show that AT lowers intraday liquidity risk. It also shows that higher AT leads to a lower incidence of extreme price movements during the trading day.

Advantages:

ZFreedom

Expert Advisors give you independence from your computer screens. An EA can execute your trading systems without any human intervention and therefore you can be sleeping (assuming you set some alerts), working a full-time job or tanning at the beach while the expert advisor is doing all the trading for you.

Speed

EAs have the ability to execute orders at an extremely rapid speed. If there is a 100 pip market movement in 2 minutes it is very unlikely that you will have the time to do as many things as a computer could. Given the ideal trading environment, an EA is able to open, close or modify positions in split seconds making the execution of the trading strategy much more reliable than what it would for you. An EA can watch an almost unlimited number of data streams. You of course, are unable to check 20 indicator variables on 10 currencies at the same time. Well, an expert advisor is capable of doing all these things for you. As you can see, what it means to have an expert advisor is to have an extremely large number of possibilities. The fact of having a computer script execute our trading system allows us to explore trading techniques that were previously impossible to do with manual trading, it also gives us the possibility to use an unmentionable number of data streams and it makes trading independent from us, so to speak.

Capturing all targeted moves

EAs operate 24/7 and thus catch many market moves you would be unable to execute due to you being a human being (you have to sleep, eat, have fun etc). This is also a major advantage since the expert advisor is able to "watch" the market with far more efficiency and diligence than any human being.

Systematic

EAs executed their rules religiously with no emotions whatsoever. Hence, you can be sure that your strategy will be run exactly as you designed it. A manual trader on the other hand may be "flexible" with his or her trading rules and not follow the them exactly. This is especially so after a huge loss or win.

Disadvantages:

Inflexible

Being systematic can also be seen as a disadvantage. The disadvantage of automated trading systems is that they are inflexible. An automated trading system cannot do anything outside of its coding and is therefore limited to the logic you fix it to. The main problem with this is that the expert advisor may stop working profitably if market condition change against it, how far the expert advisor can adapt to changing market conditions depends mainly on the logic you give it.

Black Box

They can be used as black boxes! This is a great disadvantage of automated trading systems because it means that people who has no idea of how a system actually works can load it on a trading platform and make it trade. What you get then is someone who does not understand what the system does or if it is profitable or not running a system on their computer. This of course is the area commercial expert advisor sellers exploit and an area you should be careful about. Do NOT trade any automated trading system you do NOT fully understand.

Emotions

People are deceived into thinking that there are no emotions in automated trading. This is a huge mistake and something that has made automated trading more difficult for the people who want to profit from it. Although the EA has no emotions, the trader that controls its workings does have emotions. The nature of the emotions experienced during the use of an automated trading system are different than those experienced when trading manually but they can be as or even more destructive. The trader may be tempted to temper with the EA's rules or shut it down altogether unnecessarily.

EA testing can be unreliable

Expert advisor testing can be unreliable. A great problem arises when you combine the unreliability of some of the tests with the fact that black boxes can be sold. What you get is people buying systems that they think will perform great only to find out that their accounts get wiped out.

In other case, users create their own EAs with irrational design/testing methods. Designing Robots is both an art and a science. It is the reason why this course was created. Poor design methodologies will lead to poor understanding and management of the robots. This will eventually lead to capital ruin.

Literature review:

1. The rapidly expanding literature on algorithmic trading (AT) focuses on whether such trading enhances the ability of markets to improve long term investor welfare and capital efficiency for firms. Theory suggests that high frequency trading, a subset of AT, can have both positive and negative contributions. The positive contribution is in transmitting information more rapidly into market prices (Jovanovic and Menkveld, 2010; Martinez and Rosu, 2013), and improving market liquidity (Hoffmann, 2012; Foucault, 1999). The negative contribution is in increasing adverse selection costs for existing (non-algorithmic) traders which can have negative

externalities (Biais et al., 2013; Cartea and Penalva, 2012). Empirical research finds more consensus. A higher presence of AT is correlated with lower costs of liquidity as well as lower short term volatility (Hendershott et al., 2011; Hasbrouck and Saar, 2013). Others find higher price efficiency and liquidity with higher levels of HFT, particularly around times of market stress (Menkveld, 2013; Carrion, 2013; Brogaard et al., 2012; Chaboud et al., 2009), and that AT demands liquidity when it is cheap and supplies it when liquidity is scarce (Hendershott and Riordan, 2013; Carrion, 2013). But this literature comes with well documented limitations (Biais and Foucault, 2014). One limitation is that much of the empirical analysis is done without explicit identification of AT. Recent data has better identification but are restricted to either very few securities or a short period of time.

2. The first important piece of work related to the Efficient Market Hypothesis (EMH), comes in the 1900 as a dissertation of Louis Bachelier. In this thesis, the author well describes the general principles of options and forwards and concludes with a couple of groundbreaking ideas.
   Firstly, Bachelier asserts after inspecting the price changes of Rentes on Paris Bourse, that ”The mathematical expectation of a speculator is nil” (Bachelier 1900, p. 10). Since the author considers the ultimate outcome and the expectation of it, rather than distribution of prices as such, we talk about a fair game defining the series. Those were later on followed by a many other authors. (Sewell 2011) states that Bachelier’s work was lost for around 50 years before rediscovered by Leonard J. Savage who brought it to the attention of Paul Samuelson and others. To put it a bit sentimentally, it was Savage’s postcard who led several academics such as Fisher Black1, Myron Scholes and Robert Merton, directly or indirectly, to win a Nobel Prize in Economics, according to (Davis et al. 2011).

3. High Frequency Trading (HFT) is a specific type of algorithmic trading. Before understanding HFT, then, it would be beneficial to define algorithmic trading. These strategies attempt to discover underlying temporary but recurring pricing phenomena that can generate trading opportunities. By and large, these strategies focus on stock price relationships, such as those between pairs or groups of stocks, or of an instantaneous stock price to that security’s historic price, or price/volatility relationships. In general, they do not involve fundamental analysis of the company. Algorithmic trading strategies may include microsecond price movements that allow a trader to benefit from market-making trades, several minute-long strategies that trade on momentum forecasted by market microstructure theories, and several hourlong market movements that surround recurring events and deviations from statistical relationship (Aldridge (2010)). HFT strategies have attracted much attention from investors, regulators, policy makers, academics, the press and the public broadly. In part, this is because of the disruptive nature of the technology used, and in part because of the widespread perception that high frequency traders are “gaming” the system, as express most notably in the bestselling book, Flash Boys. According to the U.S. Securities and Exchange Commission, high-frequency traders are “professional traders acting in a proprietary capacity that engage in strategies that generate a large number trades on daily basis.”1. The SEC characterizes HFT itself to include:
   1. The use of high-speed and sophisticated computer programs for generating, routing, and executing orders;
   2. Use of co-location services and individual data feeds offered by exchanges and others to minimize network and other types of latencies. “Co-location means placing a proprietary trading computer adjacent to the exchanges’ order-taking system. Latency, generally, is a measure of the time lag between a stimulation and a response; in terms of HFT it is generally used to mean how long it takes an HFT system to place a trade order once it senses a trading opportunity;
   3. Very short timeframes for establishing and liquidating positions;
   4. The submission of numerous orders that are canceled shortly after submission; and

5. Ending the trading day in as close to a zero position as possible (that is, not carrying significant positions over night).

METHODOLGY:

STEP BY STEP PROCEDURE:

1. Installing MQL4:

Download MT4 from their website and install it on computer.Once MT4 is running, you will need an account with a broker that supports MT4.They will then be able to give you your MT4 login credentials.

2. Writing robot:

There are many languages which would be suitable for writing an algorithm trading robot from scratch, but the complications that you will encounter are in fact with the API to direct market access that your stock broker will provide—that is how you will actually enable yourself to place the trade orders in the financial markets.

Processing market data could be achieved in a plethora of languages, and probably to a faster extent than the MT4 MQL4 can run back tests (back tests are a way of testing your algorithm robot; more on this later). For the reasons of ease of functionality and general support for financial software, I highly recommend using MQL4 (MetaQuotes Language 4), the native language of MetaTrader 4, to write your algorithm trading robot.

3. Editing MQL4 Code in the MetaEditor:

We recommend using the built-in MetaEditor IDE that comes with the MT4 trading platform. To enter the editor, just right-click on an existing expert advisor in the left-hand navigator pane and select modify from the menu. The MetaEditor of MT4 will then open, and you will be able to create a new file.

4. OurAlgorithm:

For our strategy, we will begin using a basis of support and resistance from the SMA (Smoothed Moving Average) indicator. The SMA can predict a bullish or bearish entry/exit signal. The smoothed moving average helps us cut out the noise from the market, giving us a clearer view of the direction of the price action. In either an uptrend or downtrend, the SMA can behave as a support or resistance, depending on the orientation. When the price is going up, the SMA will behave as a floor of support, and when the price is declining, vice versa as a resistance/ceiling.

a. Basis for Entry:

1. When we plot two SMAs of different periods—one of a 40 period and one of a 10 period—we can then see how they cross over and create a basis for entry. As we mentioned before, when the SMA is below the price (less than the close price), we consider it a basis for support, and when the SMA is above the price (greater than the close price), we consider this a line of resistance.
2. So, in code, we do the following first to create a method for checking the cross-over of inputs for later determining our basis for entry:
3. Now we can calculate our SMA using the iMA technical indicator method provided by MQL4 and run that through our CheckForCross function to see if there has been a cross like so:
4. Here we are using the MODE_SMMA to return us the Smoothed Moving Average from the iMA technical indicator method for our chosen smoothing method.

5. If you wish to use another smoothing method, there are several choices such as Simple, Exponential, and Linear-weighted.

6. As with all support and resistance, the standard trading methodology works here: buy support and sell resistance!

7. So, for our algorithm, we are going to do just that. When there is a cross in either direction, we are going to apply the appropriate direction of trade and enter the market.

8. Here we check for the return of the CheckForCross function we defined prior, after loading it with our two SMAs defined by the iMA technical indicator.

9. We use OrderSend, which is provided by MQL4, to open the trade. As a best practice, the result is stored in the ticket variable and later checked for a positive return value so as to handle any error that may have been reported from the broker's side. ***b. Basis for Exit***

1. Like the basis for entry (except in the inverse case), when the SMA creates a death cross, we can use this signal for closure of our trade, if any trades are open. The logic for this would be written as so:
2. Here we enter a for loop to iterate over all the open orders, although currently we will only trade one trade at a time—this allows us to expand in the future and have multiple open trades if we feel we need it.
3. This also makes usage of the OrderSelect method easier, as we can use the cnt variable from our for loop.
4. Once inside the iteration, we simply look at the current OrderType (checking for a Buy OP_BUY or Sell OP_SELL order**)** and then apply the conditional logic from the return of our CheckForCross method.

c. Adding Money Management

Right now our algorithm is simple with a basis for entry and exiting of trades, but still quite dangerously stupid when it comes to fund management. For us to keep the risk under some control, we will only place one-fifth of the tradable equity on each trade, so now we need to factor that into our algorithm.

We use AccountFreeMargin and NormalizeDouble to generate this lot size, and if it's calculated at below the minimal lot size of 0.1, we will set it to 0.1.

As the account grows (or shrinks!), the exposure is only going to be set at one-fifth of the account equity. This means that over-exposure of a fixed amount (e.g. trading a specific fixed lot on any size account) theoretically cannot happen, ergo the risk of margin call from overtrading is removed or greatly reduced. ***d. Break Even:***

Additionally, we will attempt to perform some break-even situations, meaning that if the market has changed against us to create a loss substantially from opening, we look to exit with minimal loss and at least break even so as to retain our capital.

This can be achieved by monitoring the loss of our trade and relation to the open price. If the direction changes and our trade is left out of the money, we can attempt to close out the trade as close to the entry price as possible:

The above uses OrderModify to attempt to set the stop loss to the order open price. This is a crude but simple method to ensure that we step out when the price comes back to our entry price. This is only applied when the current unrealised profit is in a loss.

5. Testing It Out

We can test the previous days, months, or even years of market data for a financial instrument with ease in the MT4 (Meta Trader 4) environment. However, traders are encouraged not to trust back testing alone, but to use it as a guide to steer their trading strategy and see how effective an idea is. Back testing enables traders to check if their thinking holds water, so to speak, before committing more effort and moving forward—writing more of their algorithm into code.

6. Placing the Expert Files:

If you have been using an external text editor to write your advisor, you will need to load your expert advisor into MT4 trading platform to compile it and error check. Simply add the .mq4 file to your MetaTrader installation in the Expert directory, e.g. /MetaTrader4/MQL4/Experts.The Expert Advisor will then be available inside your MT4 software from the Navigator menu on the left-hand side under the experts section.

7. Back Testing:

Use the shortcut Control-R to open the Strategy Tester or select it from the View menu. The tester pane will open in the bottom of your window.From here, you can select the algorithm to run in the first dropdown menu, so choose the filename of the Expert advisor you have created here. Next, you can choose the symbol (financial instrument). We can also select the time period resolution to run on (15minute period, 1-hour period, 1-day period, and so on). I will be using the 30-minute setting.

Finally, we have an option for modelling on Every Tick, Control Points, or Open Prices only. While writing your script, you can opt for the Open Prices only, as it will rapidly execute your test—although the results won't be worth banking real money on yet!For this, when you are looking for a real test before going to forward testing, it is recommended to run on Every Tick. This will take a lot longer to process! (You can leave it running in an MT4 VPN online or of course overnight on your local machine if you wish.)

Modifying the Parameters

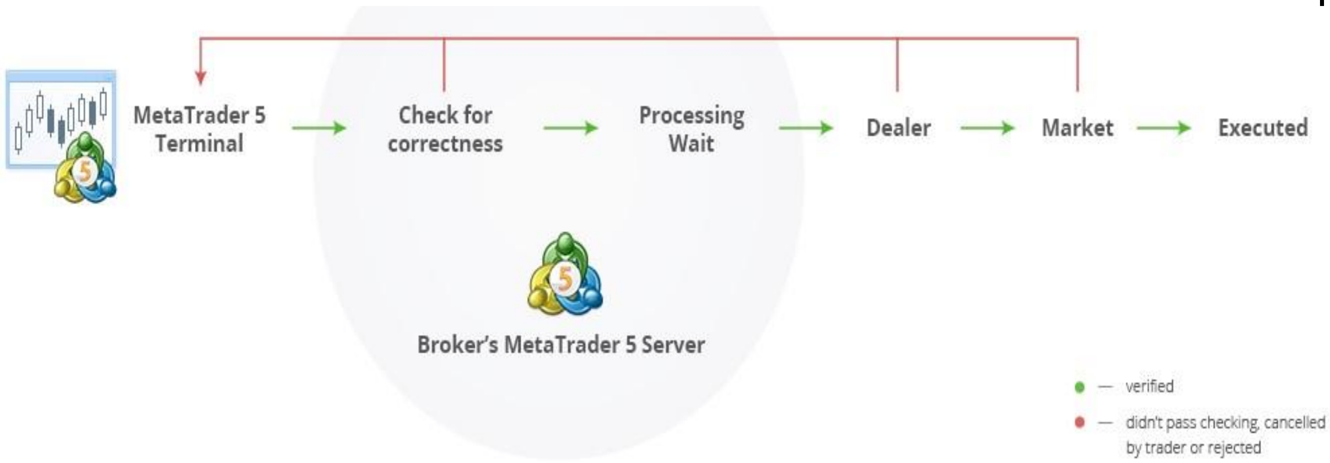
We set a couple parameters (PeriodOne and PeriodTwo) for our expert advisor so that the time periods of the two Moving Averages can be modified. These can be accessed via the Strategy tester by clicking the Expert Properties button and viewing the input tab. A numerical value can be given for each parameter here—the defaults are PeriodOne = 40 and PeriodTwo = 10.

8. Forward Testing:

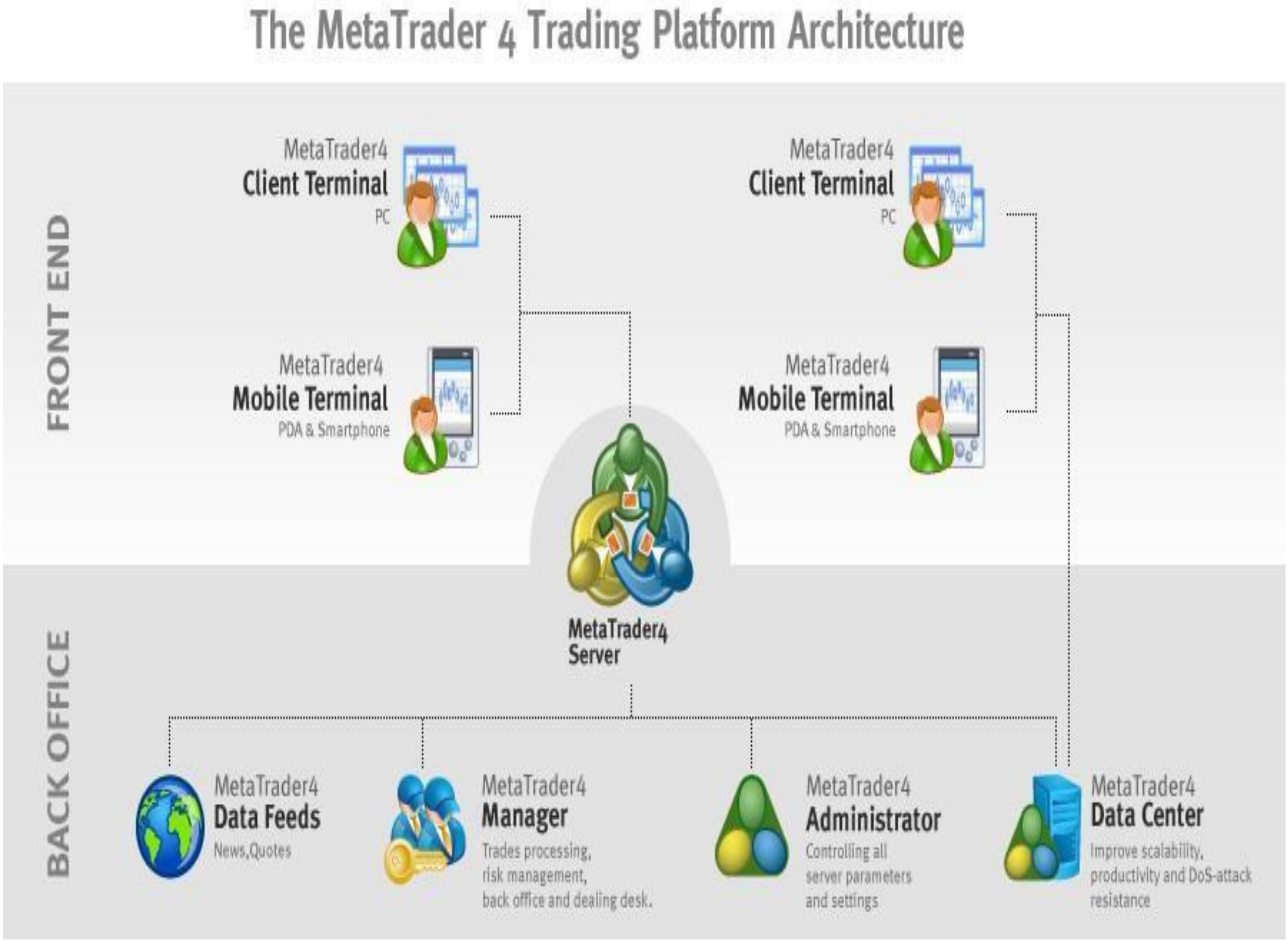
Once you have tested over historical data, you can start to test with the live market as you have already seen if your ideas weathered the storm from the back test—and (hopefully) have found some confidence in what you believe to be a winning strategy! In the live market, you may find your strategy falls flat on its face due to elements you had not factored in your prior back tests. Remember the market is always right. Your code is never smarter than the market, and there is no such thing as being lucky in trading.

A forward test really is the acid test to see if the strategy will be profitable for you to put real money on it.The forward test is usually best performed on a dedicated VPN for MT4 EA (Expert Advisors) which is often provided free by most FX brokers. The script will run 24/5 whilst the market is open, and you will be able to monitor the trades live by logging in to the account from a terminal as it will run on your mobile device via the MT4 app—or desktop machine via the MT4 software.

PROCESS:

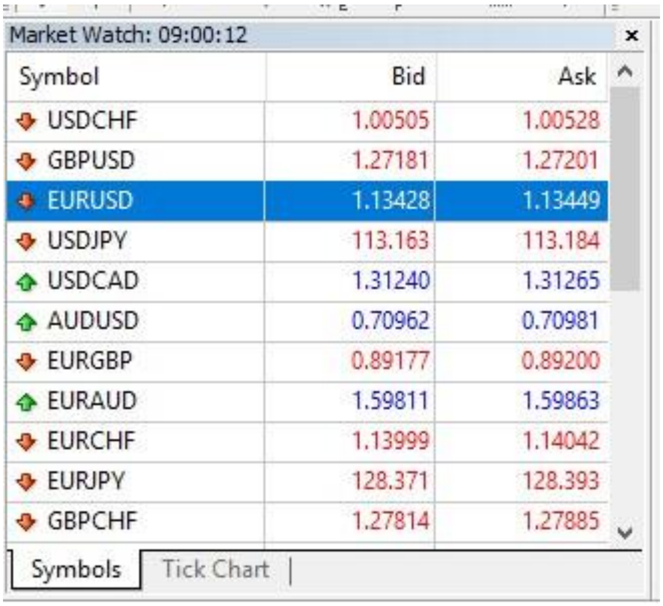


ARCHITECTURE:


The MetaTrader 4 Trading Platform Architecture
FRONT END
BACK OFFICE
MetaTrader4 Client Terminal PC
MetaTrader4 Client Terminal PC
MetaTrader4 Mobile Terminal PDA & Smartphone
MetaTrader4 Mobile Terminal PDA & Smartphone
MetaTrader4 Server
MetaTrader4 Data Feeds News,Quotes
MetaTrader4 Manager Trades processing, risk management, back office and dealing desk.
MetaTrader4 Administrator Controlling all server parameters and settings
MetaTrader4 Data Center Improve scalability, productivity and DoS-attack resistance

RESULT ANALYSIS:

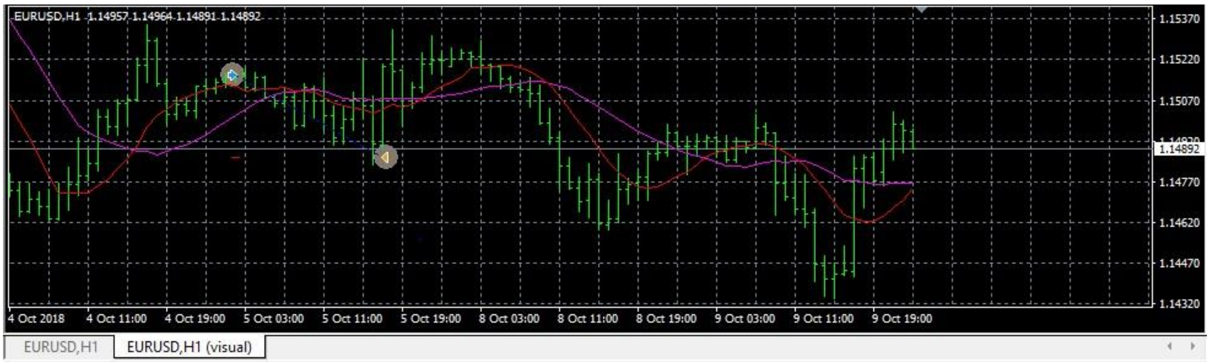
INPUT:



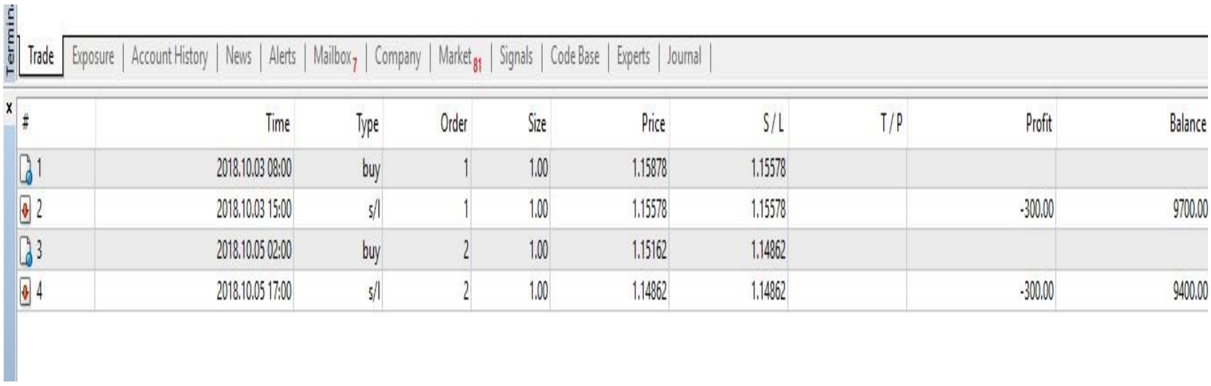
Market Watch: 09:00:12

Symbol	Bid	Ask
USDCHF	1.00505	1.00528
GBPUSD	1.27181	1.27201
EURUSD	1.13428	1.13449
USDJPY	113.163	113.184
USDCAD	1.31240	1.31265
AUDUSD	0.70962	0.70981
EURGBP	0.89177	0.89200
EURAUD	1.59811	1.59863
EURCHF	1.13999	1.14042
EURJPY	128.371	128.393
GBPCHF	1.27814	1.27885

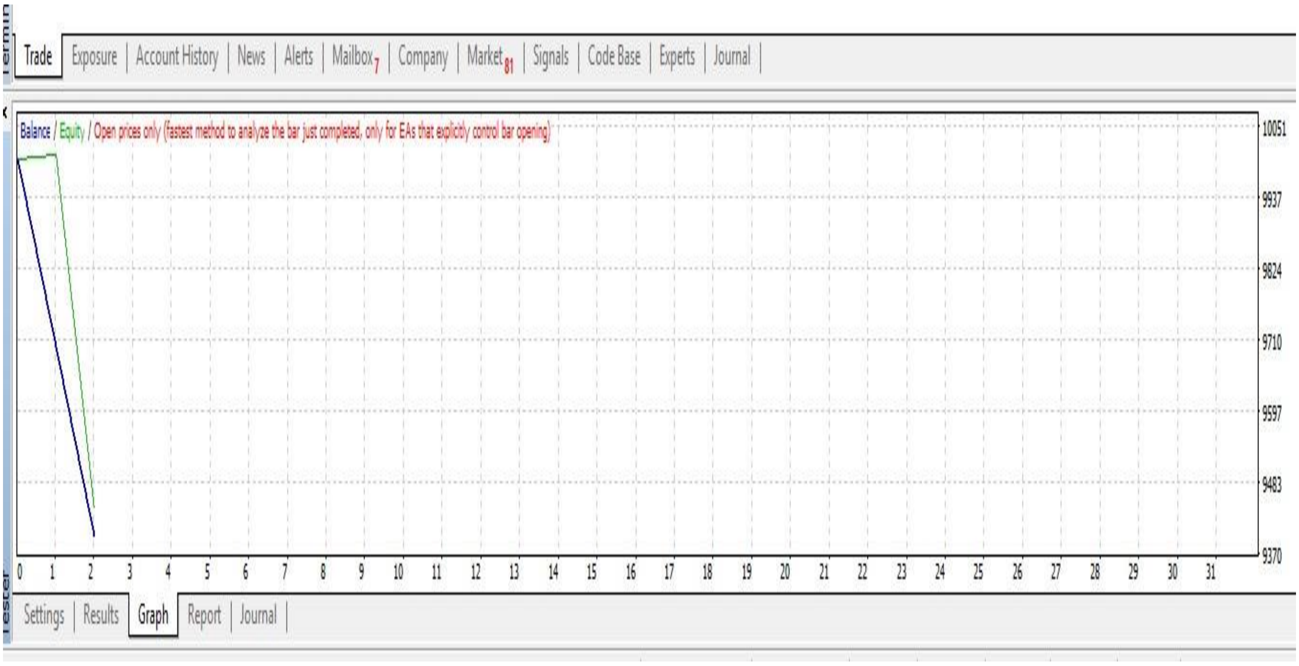
OUTPUT:

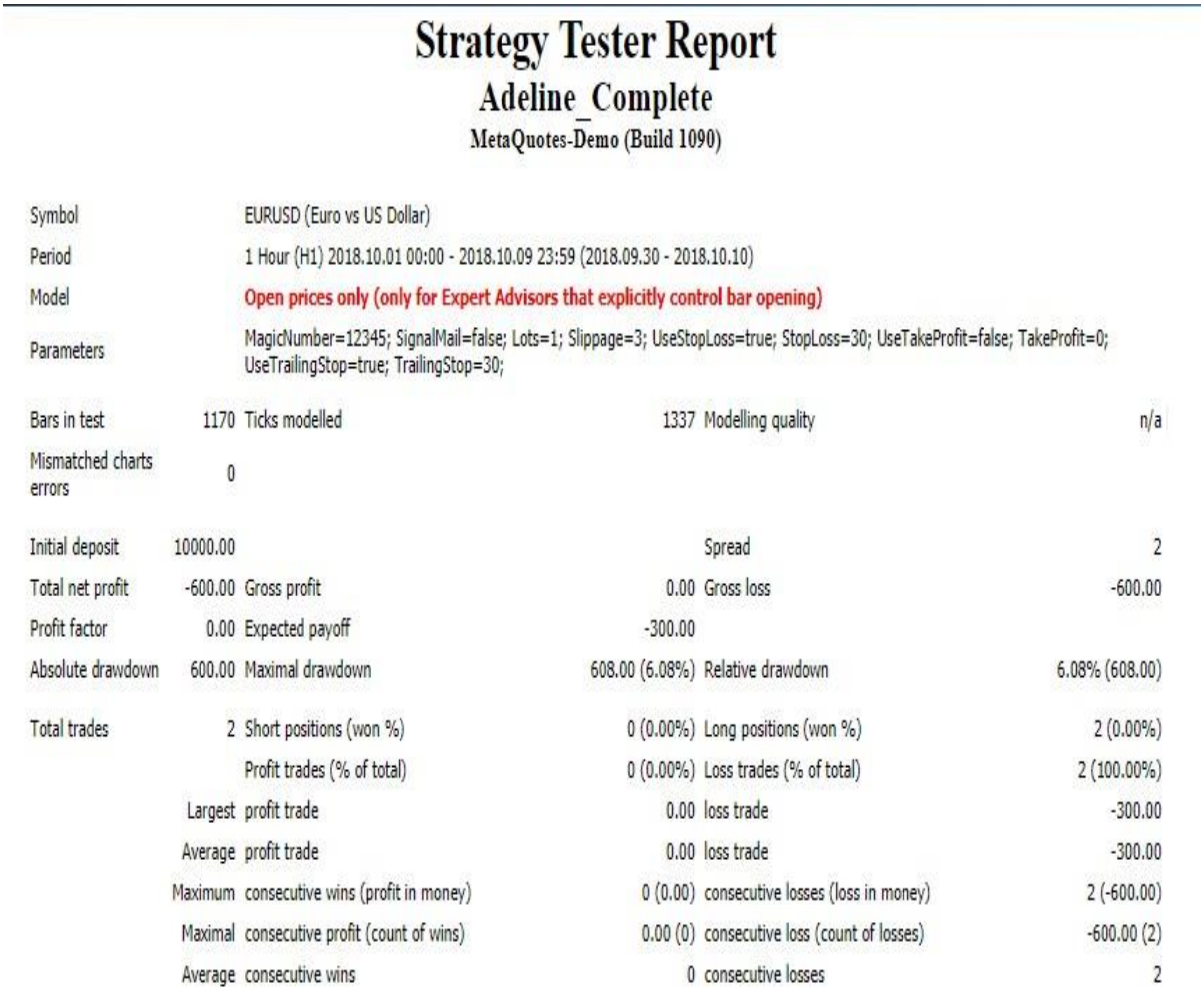


Report:



#	Time	Type	Order	Size	Price	S / L	T / P	Profit	Balance
1	2018.10.03 08:00	buy	1	1.00	1.15878	1.15578			
2	2018.10.03 15:00	s/l	1	1.00	1.15578	1.15578		-300.00	9700.00
3	2018.10.05 02:00	buy	2	1.00	1.15162	1.14862			
4	2018.10.05 17:00	s/l	2	1.00	1.14862	1.14862		-300.00	9400.00





Strategy Tester Report

Adeline_Complete

MetaQuotes-Demo (Build 1090)

Symbol	EURUSD (Euro vs US Dollar)				
Period	1 Hour (H1) 2018.10.01 00:00 - 2018.10.09 23:59 (2018.09.30 - 2018.10.10)				
Model	Open prices only (only for Expert Advisors that explicitly control bar opening)				
Parameters	MagicNumber=12345; SignalMail=false; Lots=1; Slippage=3; UseStopLoss=true; StopLoss=30; UseTakeProfit=false; TakeProfit=0; UseTrailingStop=true; TrailingStop=30;				
Bars in test	1170	Ticks modelled	1337	Modelling quality	n/a
Mismatched charts errors	0				
Initial deposit	10000.00			Spread	2
Total net profit	-600.00	Gross profit	0.00	Gross loss	-600.00
Profit factor	0.00	Expected payoff	-300.00		
Absolute drawdown	600.00	Maximal drawdown	608.00 (6.08%)	Relative drawdown	6.08% (608.00)
Total trades	2	Short positions (won %)	0 (0.00%)	Long positions (won %)	2 (0.00%)
		Profit trades (% of total)	0 (0.00%)	Loss trades (% of total)	2 (100.00%)
	Largest	profit trade	0.00	loss trade	-300.00
	Average	profit trade	0.00	loss trade	-300.00
	Maximum	consecutive wins (profit in money)	0 (0.00)	consecutive losses (loss in money)	2 (-600.00)
	Maximal	consecutive profit (count of wins)	0.00 (0)	consecutive loss (count of losses)	-600.00 (2)
	Average	consecutive wins	0	consecutive losses	2

CONCLUSION:

We've laid down some of the basics of writing a trading algorithm and introduced many new ideas. From walking through the code, I hope you can now see the inner workings of an algorithm in MQL4, and see how a technical indicator like the moving average is used for generating an entry and exit signal. In terms of money management, we have gone over the possibility of including break-even conditions and dynamic lot sizing to use one-fifth of the available equity. Feel free to tweak these parts of the code to your desires for risk. We have gone over the backtesting strategy tester of MT4 and opened the doors for the potential of forward testing and even Zscore testing in the future, all of which are vital before going live.With more fine tuning and research, you can possibly have a very profitable experience in the markets one day soon. Now that you have a good foundational expert advisor as a base script, you can start writing your own strategies into the MQL4 language—and testing out if they really work as well as you feel they ought to, or as some trading books and mentors may allude to!

Going forward, you will definitely want to test your algorithm more before committing real money to it. Once you feel confident enough that you have a good performing script, you may also want to join a community to take your ideas further and help brainstorm, or take apart other algorithms to see their workings and how you could incorporate them into your strategy. Sharing expert advisors with other traders is a great way to collaborate and see how other people are writing their algorithm for the plethora of technical indicators out there such as MACD, RSI, CCI, ADX, Bollinger Bands and Ichimoku... the list goes on and on.For doing more testing, I really recommend you download all the previous market data and load it into MT4 so as to be able to do a more thorough back test.

References:

1. HIGH-FREQUENCY TRADING: a white paper; Khaldoun Khashanah, Ionut Florescu, Steve Yang
2. The causal impact of algorithmic trading on market quality; Nidhi Aggarwal and Susan Thomas
3. http://www.igidr.ac.in/pdf/publication/WP-2014-023.pdf
4. Application of technical analysis on algorithmic trading; Jan S´ıla; PhDr. Ladislav Kriˇstoufek Ph.D.
5. Biais B, Foucault T (2014). “HFT and Market Quality.” Bankers, Markets and Investors.