Intelligent Algorithmic Trading Using Neural Networks and Genetic Algorithms

Andrew Cachia
Department of Artificial Intelligence
University Of Malta
andrew.cachia.12@um.edu.mt

***Abstract*—The aim of this project was to create an autonomous system for automatic stock trading. The first approach was that of using traditional time series analysis techniques, in the form of technical indicators, to monitor the market index. The second approach used machine learning in the form of an Artificial Neural Network. The two approaches were then combined into a hybrid trading system, and the results compared to other systems that attempt to predict stock market price fluctuations or systems that attempt to use algorithmic trading techniques. Finally, the hybrid system was optimized using Genetic Algorithms.**

***Index Terms*—Artificial Neural Network, Machine Learning, Algorithmic Trading, Stock Market Prediction, Genetic Algorithms, Technical Indicators**

I. Introduction

Predicting stock markets is something which has captured the attention of many investors and mathematicians over the years. The primary goal of an investor is to beat the market in order to generate high returns, and knowing beforehand the movement of the market would definitely help in this.

Currently the question of whether or not stock markets are predictable is still a highly debatable one. There seems to be three general schools of thought regarding stock market predictions:

1) The first believes that predictions based on past data are impossible. These make use of theories such as Random Walk Hypothesis, or the Efficient Market Hypothesis [23].
2) The second view is that of performing fundamental analyses, where external factors influencing the markets in question must be taken into consideration. This includes assessing other financial as well as non-financial signals, determining correlations and deducing the impact these movements may have on stock market price movements [24].
3) The third view which has recently been gaining popularity is that of developing models that make use of artificial intelligence techniques, which examine historic and present data in an attempt to learn and recognize patterns [17].

Many different attempts at predicting stock markets have been tried and tested. Studies have shown that it seems to be possible - using machine learning techniques, statistical analysis and digital signal processing to be able to somewhat predict movements in stock markets.

A. Findings

Current studies seem to show that traditional algorithmic trading techniques are becoming less and less effective as the market becomes more efficient [4]. Many of the technical indicators used in trading are known by most of the financial world, and therefore huge financial firms with advantageous resources such as bandwidth and close proximity to exchanges can execute trades much faster once the signals are hit.

In fact, a lot of scientists and financial engineers are dedicating more of their time into using machine learning techniques to predict stock market movements, rather than traditional algorithmic trading techniques [4], [7], [9], [15], [17]. Machine learning prediction techniques seem to be easier to implement and more effective, however they do also have a number of limitations, including instability, long training times, and relatively low confidence levels [4].

B. Objectives

The aim of this project is firstly to create a trading system using traditional technical analysis. Secondly, an Artificial Neural Network will be created that will be able to learn the stock market movements and predict the next movement at each point in time. This Neural Network will then be applied to a trading system in order to determine whether any profits are being generated, and comparing this system to the traditional algorithmic trading system using technical indicators. The system should be able to process data and return results dynamically with each new value added, and when performing simulations with historic data it must be blind towards all future values.

Next, a hybrid trading system will be used that will utilize both the ANN and the traditional techniques side by side.

Finally, Genetic Algorithms will be used in an attempt to optimize and increase the accuracy of the hybrid trading system.

The system will attempt to optimize profit. The better the accuracy of the predictions, the more profitable the trades placed will become.

C. Instruments

The system will be tested on a number of instruments, including:

- 2 Indices (S&P500, FTSE)
- 4 Stocks (Netflix, Google, General Motors, Coca-Cola)

- 3 Crypto-Currencies (Bitcoin, Ethereum, Litecoin)

The reason for selecting 3 different categories of instruments (Indices, Stocks and Cryptocurrencies), is due to the vastly different volatility levels between these different categories, with indices being the least volatile and cryptocurrencies having very high volatility.

D. Approach

The signal processing techniques that are to be used include **Moving Averages**, the **Fischer Transform** and the **Relative Strength Index**. These indicators should each give their own indication at each point in time of whether it is currently a good time to open or close a buy or sell position. The moving averages will also make use of **regime switching**.

The machine learning algorithms will consist of an **Artificial Neural Network**, which will, given a large training set comprised of historical data, be able to identify and detect patterns by itself, and learn to predict up or down movements in the market. **Genetic Algorithms** will also be used to determine the optimal weightings to assign to each of the indicator signals within the hybrid system.

The reason that an ANN was used as one of the machine learning algorithms of choice is that it is a powerful classification algorithm, detecting hidden patterns throughout large datasets, and yet flexible enough to be used to solve a vast range of different problems.

Genetic algorithms were also chosen as they are a powerful tool for traversing huge sample spaces in order to determine the best possible combination or rules. Including a metaheuristic approach to this application could prove beneficial because of the large amount of variables and complexities assosciated with financial markets.

E. Other Work

Artificial Neural Network's (ANN) have been used by a number of different authors when attempting to predict stock market price movements or to create trading systems. Kara et al. attempted predicting the stock price index movement of the Istanbul Stock Exchange, receiving 75.75% accuracy with their ANN model [16]. Other authors tried combining a number of different ML techniques together with ANNs, such as Fuzzy Logic [6], [9], or Hidden Markov Models [7].

In [29], the author's present a Robust Genetic Programming approach for determining profitable trading rules, which are applied to a portfolio of stocks from the Spanish market, and claim to have generated a 32% return between 2009-2013 beating the IBEX35 index return of 2.67%. The authors in [30] also use genetic algorithms to determine the optimal set of indicators to be used in the system, resulting in the system outperforming the benchmark significantly.

F. Existing Gaps

One noticeable scenario that seems to occur is that most authors base their trading systems or prediction models either entirely on technical indicators, or entirely on machine learning models. When both are used, generally the technical indicators are used as input to the machine learning models.

What this system proposes is having machine learning and technical analysis signals working side-by-side to generate buy and sell signals. This will allow us to harness the power of machine learning, whilst reducing risk as the technical indicators can be deemed to be safer and more controlled.

G. Main Contributions

The main contribution of this study will be to run a number of experiments to determine whether machine learning techniques really can outperform traditional technical analysis in a number of different markets. It also aims to create a hybrid system that will harness the power of both these two systems to determine if an optimal system can be created. Finally, it will attempt to determine if Genetic Algorithms can be used to enhance the trading system.

II. Literature Review

Although some might argue that financial markets are unpredictable due to their random nature, one must take into account determinism; the possibility of prediction [1]. In mathematics, a deterministic system is one where even though the state of the system at a specific point in time may be difficult to describe, no randomness is involved in the resulting future states of the system. Therefore, given a specific initial state or condition, a deterministic system will always produce the same output.

Chaos theory suggests that although certain initial values might be unpredictable, the underlying long term effects may in fact be predicated. As summed up by Edward Lorenz [2], *Chaos: When the present determines the future, but the approximate present does not approximately determine the future*.

Another factor to take into consideration is financial theories such as the Dow Theory, which states that a financial market usually tends to follow a pattern [6]. In the 1922 book entitled The Stock Market Barometer, William Peter Hamilton explains how although stock market trends tend to go through slight derivations from time to time, he asserts that charting past fluctuations will indicate market movements [8].

Due to the difficulty in forecasting stock markets, a large number of techniques have been tried and tested over the years. Preethi and Santhi [4], in their paper compared and contrasted many of the different techniques used by various different authors. From the various studies carried out, it can be seen that the methods used fall under two general categories:

- Traditional time series analysis; which involves analyzing data to extract useful information, or creating models based on previously observed values [5],
- Prediction techniques; which are comprised of machine learning techniques used to discover hidden patterns from the historic data, and that can be used to help in decision making [4].

A time series signal represents the values of a particular data set over time. It is generally depicted by a line chart, and shows the fluctuations in values throughout a certain period. For example, a financial time series, such as the S&P 500 signal, would show the price fluctuations over time.

Fig. 1. Showing S&P 500 signal over 10 years. Source: https://finance.yahoo.com/echarts

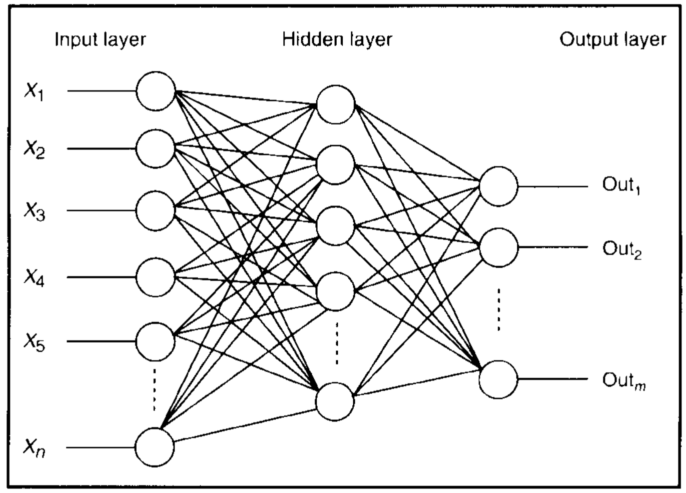

Fig. 2. Artifical Neural Network Design

Traditional time series analysis, while it can be effective, has proven to be difficult, especially when dealing with such large sets of seemingly random data. Machine learning prediction techniques seem to be easier to implement and more effective, however they do also have their limitations, including instability, long training times, and relatively low confidence levels [4].

One of the most popular machine learning algorithms in use is an Artificial Neural Network (ANN). A neural network is a type of machine learning strategy that is used to identify patterns across large spans of data. This technique makes use of black box technology, in the sense that the algorithm is not able to describe the pattern or why it occurs, but can simply identify patterns based on the inputs and outputs. The network first undergoes training through the use of a dataset, where a set of inputs and their corresponding outputs are fed into the network, until the ANN is able to successfully identify a pattern and converge. Converging means that with the current set up of the network, every input set in the training set that is entered will result in an output that is equal to (or very close to, based on the error threshold) the required output [22].

The author in [4] proposes that the limitations present in machine learning approaches could be overcome by combining different prediction techniques together, as well as with time series analysis techniques to improve the accuracy and efficiency of the prediction models.

One common proposal is that of combining Neural Networks with Hidden Markov Models. HMM are a valuable choice for such a task, due to their pattern recognition capabilities and proven stability for modeling dynamic systems [7].

Another is that of using Neuro-Fuzzy Architecture [9]. Adaptive Neuro-Fuzzy Inference Systems (ANFIS) combine fuzzy logic reasoning and the pattern-recognition capabilities of neural networks, to create and optimize the membership functions. A relationship between input and output is established. Using these optimized membership functions, the system will then use fuzzy logic to analyze the current market momentum and conditions [9]. The system can also be used for real-time high frequency trading [6]. The previous 3 price observations and the current price observation are analyzed, and the system will return either an up or down value, indicating whether the next price will go up or down.

An interesting technique to note that is used by traders is that of Moving Averages. The idea behind this is to smoothen out the volatile and irregular time series, allowing for a more consistent curve, and can also help better show market trends. In practice, buy and sell signals are generated by the crossing of two moving averages, which tend to be a long-period and short-period average respectively [10], [11].

The moving average works by using the current value in the time series, as well as a number of past values, depending on the window size (for example, with a window size of 10, the current value and past 9 values will be used). The average is then found using the respective method, and the result plotted or recorded.

This technique, however, does also contain a few flaws. One prominent one is that of lag. Since the moving average uses a certain number of past values, it will lag slightly behind the current value in the time series. The larger the window size, the larger the lag. Another problem that arises is that of loss of detail. The smoother the curve, the more detail will be lost, which might include certain details such as maximum or minimum points, or the volatility of fluctuations [10], [11].

Another issue to be considered is that of determining the right window size, as well as the right method of moving averages, since many have been created. A few of these include the Simple Moving Average (MA), Triangular MA, Exponential MA, Weighted MA, Mesa MA, Kaufman MA, Geometric MA, Quadratic MA, amongst many others [10].

[10] states that results show that the best Moving Average method used (in terms of smoothness-lag ratio), is the Triple Exponential Moving Average (TRIX). However, another interesting method to consider is the Kaufman Adaptive Moving Average (KAMA). The power in KAMA lies in its ability to determine market volatility and adapt accordingly. Therefore,

if the market is advancing steadily, you would want the MA to closely track he trend (i.e. a short look-back period). However, if the market is moving sideways with high fluctuations and volatility, the MA should have a longer look-back period so as to filter out the noise and smoothen the curve [11].

Another analytic technique whose application is proposed by Ehlers in his book [12] is the Fisher Transform. The purpose of the fisher transform is to bring the given signal to fluctuate around zero, to emphasize the signals cyclic nature and determine at which point of the cycle the signal currently is. Knowing the stage of the cycle as well as the cyclic turning points will be of great use when attempting to predict movements in price.

The Relative Strength Index (RSI) might also be useful when trying to predict market price movements. The RSI determines the current momentum in the market, and may generally indicate whether the price of a stock is currently overvalued or undervalued. It measures the current markets price compared to the price of a number of periods ago, taking into consideration the average value of daily prices in between [13].

The idea to use Neural Networks to solve complex prediction problems arose in 1964 [18], whose attempt however was unsuccessful due to the lack of a learn method for multi-layer networks. In 1988, however, a study by Werbos implemented a back-propagation algorithm into his Neural Network, and demonstrated the power of neural networks in solving prediction problems[19].

In [15], Nayak et al. demonstrate the importance of pre-processing data before trying to forecast the behavior of time series. They claim that certain regularities are often masked by noise, and that data normalization as well as a proper network architecture are fundamental steps in order for the ANN to learn from the data. They go on to describe and compare different techniques of normalizing data, including Min-Max Normalization, Sigmoid Normalization, as well as the importance of price smoothing.

Kara et al. attempted predicting the stock price index movement of the Istanbul Stock Exchange. They created two models, one making use of an ANN and the other of support vector machines (SVM), and compared the efficiency of each[16]. Their results showed that the average performance of the ANN model (75.75%) was higher than that of the SVM (71.52%). The ANN was set up as being a 3 layered feed forward network, with the best combination of parameters including 30 hidden neurons, momentum coefficient of 0.7 and learning rate of 0.1. As inputs, ten technical indicators were used, including a 10-day SMA, a 10-day weighted moving average, momentum, RSI, Accumulation/Distribution Oscillator, and more, and the output being either a 0 or 1 to reflect a price increase or decrease for the following day.

In [17], Niaki and Hoseinzade also make use of an ANN to attempt to predict the S&P 500 index. In their approach, they examine a number of other financial as well as non-financial signals to check for correlations between those and the S&P 500 signal. These signals were also split into different categories. The signals whose correlation to the S&P 500 signal was greater than 0.5 was chosen to be another input to the ANN, with a maximum of one signal per category. A simulation was then run, and although it resulted in a net loss, the ANN still far outperformed a standard passive buy and hold strategy.

In [20], Hamid and Iqbal also make use of a number of related signals as inputs to the neural network, including a combination of futures contracts and spot indexes. Futures contracts include commodities like gold or oil, treasury obligations such as bonds and bills, and foreign currencies. The spot indexes used are the Dow Jones Industrial Average, NYSE Composite Index, and the S&P 500 Index. The predictions included a 55-day forecast, a 35-day forecast and a 15-day forecast. Their results show that volatility forecasts from neural networks provide superior results when compared to implied standard deviations.

In interesting combination is that of combining an ANN with a genetic algorithm, as described in [14]. Yudong & Lenan make use of a genetic algorithm based on bacterial chemotaxis, which is used to update the ANNs weights in order to obtain the function which is accurate enough to define the relationship between the input variables. They claim that this strategy offers less computational complexity, better accuracy and less training time.

Genetic algorithms are implemented in various different ways by various authors. In [29], the authors present a Robust Genetic Programming approach for determining profitable trading rules, which are applied to a portfolio of stocks from the Spanish market. The authors state how one of the biggest challenges with genetic algorithms is over-fitting. They propose overcoming this by rather than calculating the fitness over the whole dataset, calculating it on randomly selected segments, which helped create a much more robust system.They also claim to have generated a 32% return between 2009-2013 beating the IBEX35 index return of 2.67%.

Another set of authors combined genetic algorithms with rough sets analysis [30]. A rule discovery approach is created using Genetic Algorithms, with the GA determining the optimal set of indicators and their rules to find the most profitable approach. When performing this experiment on a number of Korea Composite Stock Price Index 200 (KOSPI 200) futures market, it was shown to significantly outperform the benchmark.

A. Literature Review Critical Analysis

One important observation to note is that a large number of the above studies, especially those related to genetic algorithms, perform their experimentation and testing against indices. The issue with this is that indices are generally quite stable due to the diverse amount of equities within. It would be interesting to see how these algorithms fare against more volatile assets such as equities or even cryptocurrencies. In our experiment, a number of different financial markets will be used for testing.

Another observation is that a number of authors tend to use a "long-only approach" when carrying out experiments and running simulations. Although this is not a bad approach, it will work better during bull markets and worse during bear markets, meaning that the testing period used will make a large difference.

One noticeable scenario that seems to occur is that most authors base their trading systems or prediction models either entirely on technical indicators, or entirely on machine learning models. When both are used, generally the technical indicators are used as input to the machine learning models.

What this system proposes is having machine learning and technical analysis signals working side-by-side to generate buy and sell signals. The GA will then be used to assign different weightings to the different signals, to determine whether more importance should be given to the signals generate by the AI system, or to the technical indicators (and which ones). This will hopefully help increase profits whilst reducing risk as the technical indicators can be deemed to be safer and more controlled.

III. Methodology

A detailed description of the techniques and algorithms used in this system are given, as well as an overview of the systems architecture and design.

The trading system is comprised of a number of different components, each giving their own indication of the direction the stock market should be heading. The system will then use a combination of these indicators in order to determine whether to open or close a buy or sell position.

It can be summarized into two different sections. One uses traditional statistical and analytic techniques to analyze the time series and determine the direction of movement. The other makes use of Machine Learning to detect patterns and learn how the market will fluctuate based on current and historic data.

The traditional analytic methods will be described first, since some of them are used when pre-processing the input data before feeding it to the machine learning algorithm.

A. Analytic Indicators

A number of different technical indicators were used in this project, each with their own method of analyzing different features or properties of the market. Their definition and structure are listed below:

1) Moving Averages Indicator: Moving Averages (also known as rolling averages) are a common technique used in statistics when analyzing time series data [10], [11]. Given a fixed subset size (window size), the value of the moving average is calculated by obtaining the current value in the time series as well as the past amount of values as determined by the window size, and calculating the respective average of those values. This is performed at each value of the series (where past data is available) in order to create a new, smoother time series which closely reflects the actual time series.

The advantage of this is that it smoothens out short term price fluctuations and highlights long-term trends. A disadvantage could be that it creates an element of lag. The size of the window chosen will greatly determine both the smoothness of the curve, as well as the lag. Therefore, care must be taken when selecting a window size.

Crossing of two different moving average signals, however, could be a good indication of the direction in which the market is heading, and would therefore be a good indicator of whether to open or close a buy or sell position.

As previously mentioned, quite a variety of moving average types exist [10]. In this system, four types of Moving Averages were used, namely:

- **Simple Moving Average:** Takes a simple average of the values in the time period window.
- **Exponential Moving Average:** Similar to an SMA, but more weight is given to the most recent data.
- **Triangular Moving Average:** A double-smoothened moving average, where most of the weight is given to the values in the middle of the time period window.
- **Bipolar Moving Average:** Structured according to Pascals triangle.

For both buy and sell positions, different types of Moving Averages are used, each with their own set of window sizes. When two moving average lines cross, a signal is generated, being either a buy open, buy close, sell open or sell close.

Therefore, for the Buy signal, whenever the value of the EMA is greater than that of the SMA, the buy position will open, and the system will indicate that its a good time to buy. For the Sell signal, whenever the BMA is greater than the TMA, a sell position will open.

The system makes use of **regime switching** in order to determine the best pair of moving averages to use. A set of pre-defined window sizes are set as shown below.

Buy Signal							
SMA Window Sizes:	5	15	25	40	50	75	100
EMA Window Sizes:	3	5	9	11	15	20	25
Sell Signal							
TMA Window Sizes:	10	20	35	45	55	90	
BMA Window Sizes:	4	7	10	12	18	22	

During a regime switch, each and every SMA window size would be used along with every EMA window size and tested over the past 100 values. The same is done for the TMA and BMA window sizes. The combination which would have returned the highest profit will then be selected and used in real time over the system for the next 15 input values (in this case, the next 15 days). After that, another regime switch will be performed.

The chart in Figure 3 demonstrates a sample of the input signal along with the two moving average signals. In this example, the lines are as follows:

- The red line represents the input signal, i.e. the daily values of the S&P 500
- The blue line represents an Exponential Moving Average of changing window sizes.

Fig. 3. Moving Averages Graph of the S&P 500 Index

- The green line represents a Simple Moving Average of changing window sizes.

The black circles highlight the points where a Buy signal is being generated, since the value of the EMA would be greater than that of the SMA.

```
Code snippet:
IF EMA_value > SMA_value
        RETURN  'BUY'
ELSE
        RETURN  'NULL'

IF TMA_value > BMA_value
        RETURN 'SELL'
ELSE
        RETURN 'NULL'
```

2) Fischer Transform Indicator: One problem when it comes to analyzing financial time series is that many people automatically assume a Gaussian probability density function (PDF) of the data when performing their calculations (the typical bell shaped curve). This assumes that the majority of occurrences in the time series fall clustered around the mean. In reality, however, stock markets tend to be cyclic by nature, with most of the occurrences actually falling closer to the extreme values. Therefore, the nature of the time series exhibited is very similar to that of a sine wave [12].

The purpose of the Fisher Transform is to change the PDF of the distribution from Gaussian to almost-Gaussian. This allows much easier identification of the cyclic turning points, as the PDF of the distribution now places values near the extremes.

Therefore if the system is able to recognize a turning point, it will be able to predict the direction of the market.

The Fisher Transform equation is:

$$y = 0.5 \cdot \ln\left(\frac{1+x}{1-x}\right) \tag{1}$$

Where x is the input, y is the transformed output, and *ln* the natural logarithm.

The transformed signal now fluctuates around 0. The upper and lower bounds are set to 0.1 and -0.1 respectively. When the signal crosses 0 heading upwards, an open buy signal will be generated. When the signal passes the upper bound, we know that it is soon about to turn downward and therefore would be a good time to close the buy positon. On the other hand, when crossing 0 and heading downwards, a sell signal will be generated, closing the sell position as the lower bound is reached.

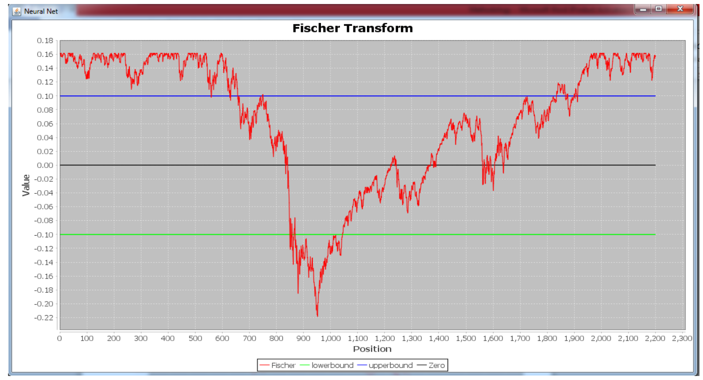

Fig. 4. Fischer Transform of the S&P 500 Index

```
Code Snippet:

IF input_value > highest
        THEN highest = input_value
IF input_value < lowest
        THEN lowest = input_value

Point = (input-lowest)/(highest-lowest)
ScaledPoint = 0.33 * 2 * Point - 0.5;
Fisher = 0.5 * log((1 + ScaledPoint)/(1-ScaledPoin

IF Fisher > 0 AND Fisher < upperbound
        RETURN 'BUY'
IF Fisher < 0 AND Fisher > lowerbound
        RETURN 'SELL'
```

3) Relative Strength Index Indicator: The Relative Strength Index (RSI) is considered a momentum oscillator, used to measure the velocity and magnitude of the change in the current market price. It is a useful indicator to determine whether the price of a stock is currently overvalued or undervalued. It compares the magnitude of recent gains and losses to determine whether the momentum of the signal is too high. If so, then there generally is a good chance that the stock is being over hyped and thus overvalued, meaning that after the initial soar; the price will come shooting back down. The RSI was developed by J. Welles Wilder in 1978 and since has become a popular indicator.

To calculate the RSI value at each point, the past 10 values are analyzed. Whenever the value increases in relation to the previous value, it is marked as up, whilst when it decreases, it is marked as down. The average for all the values marked as up, as well as of those marked as down is calculated, and entered into the following formula:

$$RS = \frac{average(up)}{average(down)} \tag{2}$$

The value of this is then used to calculate the RSI:

$$RSI = 100 - \frac{100}{1 + RS} \tag{3}$$

Whenever the RSI value is above the upper bound, meaning that the stock is overvalued, a sell signal is generated. On the other hand, when below the lower bound a buy signal is issued.

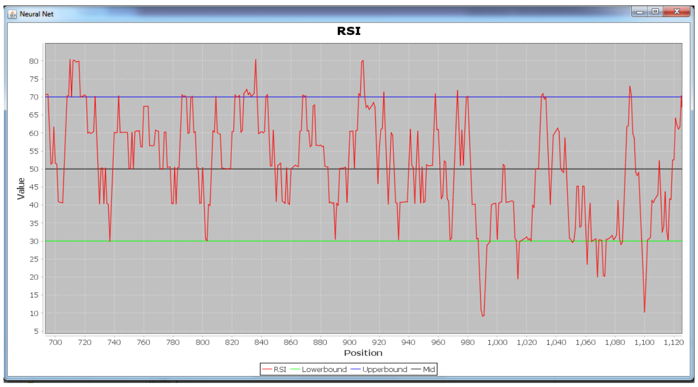

Fig. 5. RSI of the S&P 500 Index

```
Code snippet:

FOR int I = 1 TO 10
          IF (value(i) > value(i-1))
                    THEN up(i) = value (i)
          IF (value(i)<value(i-1))
                    THEN down(i)=value(i)
ENDFOR

RS = average(up) / average(down)
RSI = 100     100/(1 + RS)

If (RSI > 70)
          THEN RETURN    BUY
IF (RSI < 30)
          THEN RETURN    SELL
```

B. Machine Learning Algorithms

Machine learning was used in this system in the form of an Artificial Neural Network (ANN) and a Genetic Algorithm (GA). The objective was to determine whether or not Artificial Intelligence could learn to predict stock market fluctuations based on historic data, and improve the accuracy of the trading system.

1) Neural Network Architecture: For the purpose of this project, a Neural Network toolbox was created using Java in order to allow efficient switching between different types of neural network setups.

The primary function of the Neural Network toolbox is to recognize and detect patterns in a dataset using machine learning in the form of an Artificial Neural Network. The function accepts a two dimensional array as the inputs, and another two dimensional array defining the target outputs. The network consists of 3 layers, being the input, hidden and output layers. A 4 layer approach was also attempted; however the results were very unsatisfactory.

The function will begin a loop, where it iterates through each input set in the dataset. For each input, it will perform the first forward pass by calculating the sum at each hidden node in the hidden layer of the input values connected to it, with each input value first multiplied by the weight of the connection between the respective nodes. This can be summarized in the formula below.

$$S = \sum_{j=1}^{N} x_j w_j \tag{4}$$

Where S is the resulting value, *Xj* is the input value of neuron *j*, and *Wj* is the value of the weights in the connection between input *j* and the hidden neuron.

The resulting value will then be passed through a sigmoid function, also known as a squashing function, whose purpose is for the output to lie between zero and one.

$$S(t) = \frac{1}{1 + e^{-t}} \tag{5}$$

Where t is the current value in the hidden neuron, and S is the resulting value after performing the sigmoid function.

The second forward pass is performed in exactly the same manner. The output generated represents the output value from the neural network. This value is then compared to the target value, and if the difference is greater than the error threshold, the network will perform a backward pass to adjust the weights.

To perform a backward pass, the delta for each output neuron must be computed using the following formula:

$$\Delta = OUT_j(1 - OUT_j)(Target - OUT_j) \tag{6}$$

Where *OUTj* is the value of the output neuron *j*. The backward pass is then performed and the value of each weight is changed:

$$\Delta w = n \cdot hidden_p \cdot \Delta_j \tag{7}$$

where Δw is the change in the value of the weight, $hidden_p$ is the value of the hidden neuron p and Δ_j is the delta value for output neuron j.

To calculate the delta values for the hidden layer, a slightly different method needs to be used, as shown by the formula below:

$$\Delta_p = OUT_{pj}(1 - OUT_{pj})(\sum \Delta_{qk} - W_{pqk}) \tag{8}$$

First, each hidden neuron should find the sum of the delta values for each output neuron attached to it multiplied by the connecting weight. The result should then be input in the above formula, with OUT being the value of the hidden neuron. The weights of the first layer are then changed in the same manner as previously described for the second layer.

The ANN also makes use of momentum analysis. Momentum analysis was implemented in the neural network to speed up convergence, as well as to avoid local minima [26]. During the backpropagation each time a weight is to be changed, the momentum coefficient is multiplied by the value the weight was changed in the previous epoch, and the result would then be added to the current value the weight will be changed by.

The toolbox allows a variable number of training parameters to be set, including momentum coefficient, learning rate, error threshold, as well as number of hidden neurons. An early stopping rule can also be set if need be, which defines the amount of facts needed to be learnt before the algorithm can stop attempting to converge. For this experiment, the stopping rate was set at 5000 epochs.

2) ANN Training Method: In order to train the neural network, a training set needs to be fed to it which contains both the input data and the target output, so as to allow the algorithm to learn and detect patterns.

The data to be analyzed was the historical prices of the various financial markets. Daily closing prices from 2005 2017 were taken as training data. The problem with feeding raw data to a neural network is that due to its high volatility, the algorithm will find it difficult to detect patterns.

Therefore, the first step of data preprocessing was that of applying the fisher transform to the series. This would de-trend the data, and simply show at what point of the cycle the series currently is, i.e. whether at the top or bottom of a cycle. After this, the data was smoothened using a Low Pass Filter in the form of a Simple Moving Average (SMA) of window size 5, in order to reduce the volatility and simply show the general direction in which the market is heading.

The next step was to create the target data, which was basically the next days movement. Assuming the current days value in the transformed time series to be t, if the value at t+1 increases, then the target output for day t would be a 1. If the value at t-1 is set to decrease, then the target output for day t would be a 0.

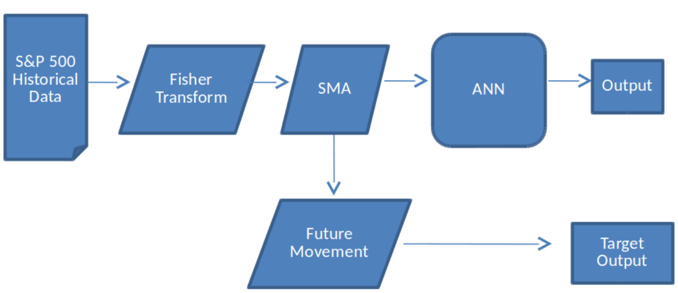

Fig. 6. ANN Sequence

The training set fed to the neural network was comprised of the current days value along with the past 19 values in the transformed signal, along with the target output. The neural network makes use of error back propagation in order to adjust the weights each time the resulting outputs error is larger than the error threshold permitted.

The training algorithm was run until the early stopping limit of 5000 epochs was met, so as to prevent overtraining and allow better generalization of results.

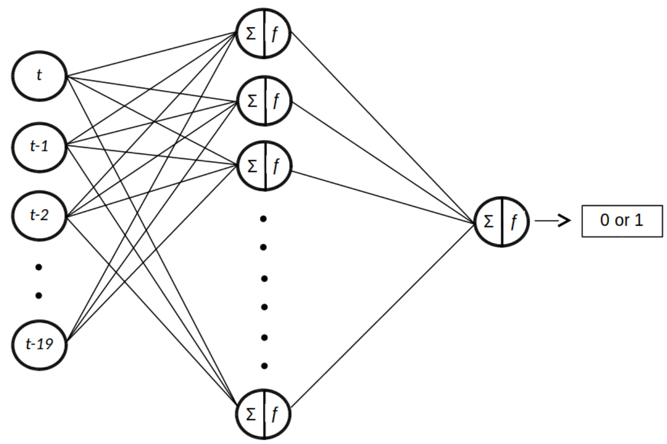

Fig. 7. ANN Architecture

In order to test the accuracy of the ANN after training was complete, a simulation was performed on new data. The input given was the daily closing prices, this time from 2018. The target output was calculated for each day in the time series.

The data was then fed to the neural network. No backpropagation was used to correct the error, the data was simply fed forward and the actual output compared to the target output. If the outputs error was greater than the error threshold, that epoch was marked as incorrect. Once this was done for the whole testing set, the system would compare the amount of incorrect results in relation to the total amount of values given, and thus the accuracy of the ANN would be calculated.

3) Genetic Algorithms: Genetic Algorithms is a type of heuristic algorithm that is inspired by the theory of natural selection, where the fittest individuals are selected for reproduction in order to produce offspring for the next generation [31].

A genetic algorithm is generally made up five phases:

1) Initial Population
2) Fitness Function
3) Selection
4) Crossover
5) Mutation

The **initial population** is a set of individuals (also known as chromosomes), where each individual defines a rule set for the particular problem which needs to be solved.

We then pass each of these individuals/chromosomes into a **fitness function**. This fitness function will perform some kind of function, using the parameters given in the chromosome, and will give a respective fitness score.

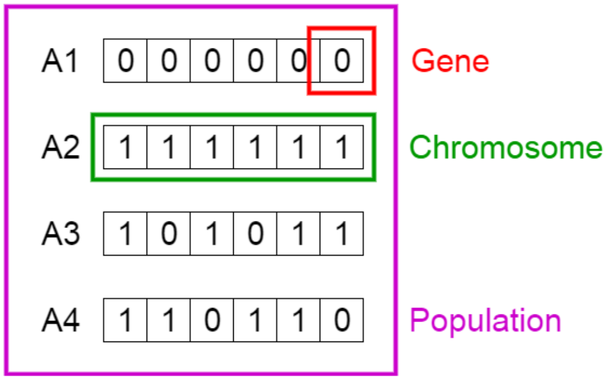

Fig. 8. Population of Chromosomes. *source: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3*

The individuals with the highest fitness score are then **selected**, and their genes are passed over onto the next generation.

The offspring are then created by passing the selected parents through a **crossover**, where portions of their genes are swapped with each other. This will help identify the particular genes, or rules, which are resulting in the highest fitness.

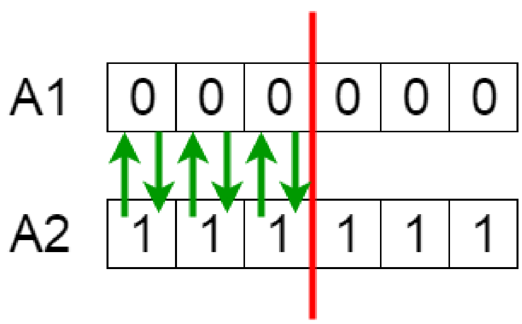

Fig. 9. GA Chromosome Crossover. *source: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3*

Every so often, a gene will be changed randomly in what is called a **mutation**. This will allow the algorithm to explore new genes which may not have yet appeared within the population.

4) Genetic Algorithm Training Method: The aim of the genetic algorithm in this scenario is to assign weightings to the different indicators within the hybrid trading system in order to identify which ones should be given more importance, and ideally optimize the accuracy of the predictions. The current strategy used by the hybrid trading system is a very simple strategy; if 2 or more of the indicators issue either a BUY or a SELL signal, then place a trade of the particular signal. If not, either close any open trades or take no action.

However, some indicators might be powerful enough to warrant opening a position by themselves. This will hopefully allow for better performance as positions can be opened or closed quicker.

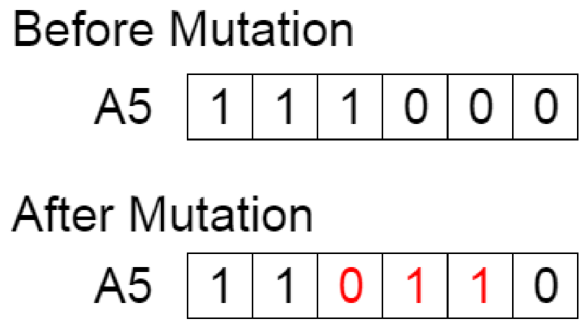

Fig. 10. GA Chromosome Mutation. *source: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3*

Each chromosome in the GA represents the set of weightings assigned to each indicator. The fitness function is a simulation performed on the training period of the financial market data. Whilst training the model, the GA will therefore continously modify the weightings to determine the set of weights which will generate the highest profit during the simulation.

An important point to note is that GAs are **extremely susceptible to over-fitting**. Most of the initial experiments using GAs performed very well during the training period, and very poorly during the testing period.

During the experimentation phase, a number of techniques were discovered that helped to prevent over-fitting and achieve a more robust result. These are as follows:

1) **Keep it as simple as possible**. Simplicity is key with Genetic Algorithms. Experiments showed that smaller sets of possible variables for each gene allowed better generalization and thus less over-fitting.
2) **Simple random sampling**. This idea was inspired by [29]. Instead of training the model over the whole training set at once, the training set was split up into a number of random groups with varying sizes. In the fitness function, each chromosome would be tested individually over all the different periods, and the total profit made over all the different periods returned as the fitness score. This greatly helped improve generalization.
3) **Re-training every 90 days**. Re-training helps the algorithm identify any new patterns that may be emerging and keep the model optimized.

Initially, the weights assigned to the indicators were decimal numbers between 0 and 1. Another 2 genes were added to the chromosone, which included the threshold to decide when to buy or sell. Using decimal numbers turned out to not perform well at all, as the infinite range of decimal numbers allows for a huge variety of different possibilities, which creates a lot of over-fitting. Including the threshold as a variable also added a lot more variability and complexity.

To make things simpler, integers were used instead of decimal numbers. This allowed better generalization and a much smaller sample space. The threshold was removed from the chromosome and instead set to a fix value of 3. Genes (i.e. weightings) could hold a value within the range of 0 - 3. This definately improved results, but issues of overfitting where still present.

To make the algorithm even simpler, the ranges were reduced between 1-2, and the threshold now set to 2. This basically meant that the only option for the trading system is that each indicator can either open a position by itself, or must wait for at least 1 other indicator to also give the signal for a position to be opened.

IV. Empirical Results and Evalutation

All the findings and empirical results pertaining to the different trading systems use will be demonstrated and explained. The accuracy of the Neural Network in generating predictions are also demonstrated.

A. Buy and Hold Strategy

First, we begin by performing a simple buy and hold strategy for the year 2018, giving us a benchmark to compare other results to.

TABLE I
Buy and hold strategy results for year 2018.

	ROI	Profit
LTC	-86.20%	-199.69
NFLX	33.12%	66.59
BTC	-72.60%	-10245.36
GOOG	-1.80%	-19.15
ETH	-82.73%	-638.93
GSPC	-7.01%	-188.96
KO	7.57%	3.3036
GM	-16.23%	-6.4532
FTSE	-12.03%	-919.97
Totals	**-237.91%**	**-12148.6196**

It turns out that 2018 was a very unprofitable year, with the markets experiencing heavy losses, espescially for cryptocurrencies.

B. Algorithmic Trading System

The trading system uses a combination of the three indicators to determine whether to open or close a buy position, as well as whether to open or close a sell position. The system runs through the data sequentially, entering the value into the system. The system will dynamically calculate the respective indicator values, update the graphs, and generate a signal, which may be of type BUY, SELL or NONE. It is important to note that at each point in time, the system is blind towards all future values.

If at least 2 out of 3 indicators generate a BUY or SELL, the system will open a buy or sell position respectively. Otherwise, it will close any open buy or sell position.

Each time a buy or sell position is opened, the system will add the value to the buy open or sell open variable respectively, whilst when closed add that days value to the buy close or sell close variable. After the run through is complete, the system will calculate the profit gained or lost as follows:

$$BuyProfit = BuyClose - BuyOpen \tag{9}$$

$$SellProfit = SellOpen - SellClose \tag{10}$$

TABLE II
Algorithmic trading strategy system results for year 2018.

	Return on Investment			Profit/Loss		
	Buy	Sell	Total	Buy	Sell	Total
LTC	-1.20%	3.10%	1.80%	-3.13	14.6	11.47
NFLX	1.10%	1.60%	2.70%	24.37	37.27	61.64
BTC	-3.20%	-0.50%	-3.70%	-915.06	-609.65	-1524.71
GOOG	-0.10%	1.20%	1.10%	-2.77	68.09	65.32
ETH	-0.50%	-1.00%	-1.60%	-7.27	-38.49	-45.76
GSPC	0.10%	0.40%	0.50%	3.8	56.74	60.54
KO	0.00%	0.00%	0.00%	0	0	0
GM	-1.60%	-1.90%	-3.60%	-2.771	-3.424	-6.195
FTSE	-0.40%	0.40%	0.00%	-31.79	90.78	58.99
Totals	**-5.80%**	**3.30%**	**-2.80%**	**-934.621**	**-384.084**	**-1318.705**

The test began by performing a simulation of the algorithmic trading system over the historic data of the different assets. The system performed reasonably well for most assets. Although cryptocurrences experienced losses, these losses were drastically mitigated and reduced compared to the buy and hold.

C. ANN Trading System

The Artificial Neural Network (ANN) was first trained over the training period of the historic data. It was trained to learn to identify market movements, and thus predict whether the next price movement would be up or down. A seperate ANN was trained for each of the different assets. When tested during 2018 (out-of-sample data), the following results were achieved:

TABLE III
ANN Training Accuracy

	Training	
	Number of Epochs	ANN Accuracy
LTC	5000	61.64%
NFLX	3723	69.72%
BTC	2334	72.60%
GOOG	1590	66.66%
ETH	3723	69.72%
GSPC	5000	75.70%
KO	3286	74.90%
GM	1922	70.63%
FTSE	3379	75.10%

The trading system was then built using the trained Artificial Neural Networks, and was tested out over the testing period of the historical data of the different assets. To trade, the following rules are given to the system:

- If the next predicted movement is **up**:
  - If no position is open, open a buy position
  - If a sell position is open, close the sell position
  - If a buy position is open, keep open
- If the next predicted movement is **down**:
  - If no position is open, open a sell position
  - If a buy position is open, close the buy position
  - If a sell position is open, keep open

The graph in Figure 11 is an example time series, showing the next predicted value at each point.

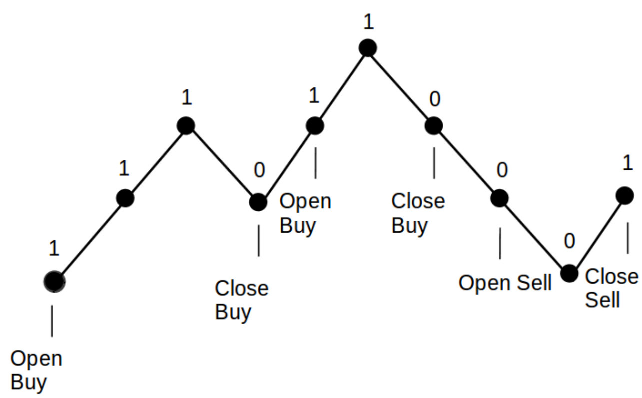

Fig. 11. AI Trading Strategy

TABLE IV
ANN TRADING SYSTEM RESULTS FOR YEAR 2018.

	Return on Investment			Profit/Loss		
	Buy	Sell	Total	Buy	Sell	Total
LTC	-55.40%	24.10%	-31.30%	-74.93	22.95	-51.98
NFLX	19.30%	30.50%	49.90%	72.62	106.72	179.34
BTC	-97.40%	38.60%	-58.80%	-5789.16	2764.72	-3024.44
GOOG	20.70%	33.90%	54.60%	234.19	381.56	615.75
ETH	-86.20%	7.20%	-79.00%	-496.54	56.29	-440.25
GSPC	-3.40%	8.30%	4.80%	-79.711	230.719	151.009
KO	9.50%	8.80%	18.30%	3.963	3.741	7.704
GM	-2.10%	5.10%	3.00%	-1.486	1.632	0.146
FTSE	-3.70%	7.60%	3.90%	-253.58	569.57	315.99
Totals	**-198.70%**	**164.10%**	**-34.60%**	**-6384.634**	**4137.902**	**-2246.731**

The ANN strategy seems to have been able to generate a few higher profits, but seems to have also taken a riskier approach at the expense of a few very high losses.

If we had to remove cryptocurrencies from the portfolio, the ANN strategy actually would have outperformed the algorithmic trading strategy. In fact, it still managed to generated a relatively high sell profit.

However, it seems to have struggled with determining patterns within these cryptocurrency markets. Pre-2018, these cryptocurrencies experienced large bull runs and therefore the system was trained on this. It thus continued to expect these markets to rise, which is why large buy losses were experienced by the system. Due to these large losses, it ended up underperforming during 2018 when compared to the traditional algorithmic trading approach.

D. Hybrid Trading System

A hybrid trading system was created, which makes use of a combination of both the technical indicators, as well as the Artificial Neural Network. This system works similar to the algorithmic trading system, in the sense that a combination of signals are generated by the different indicators, and a decision on whether to open or close a buy or sell position will be generated.

For a position to be taken, at least two out of the four indicators should signal to open a BUY or SELL position. Otherwise, the system will close any open positions, or hold.

TABLE V
ORIGINAL HYBRID TRADING SYSTEM RESULTS FOR YEAR 2018.

	Return on Investment			Profit/Loss		
	Buy	Sell	Total	Buy	Sell	Total
LTC	0.01%	3.87%	3.88%	0.15	73.95	74.1
NFLX	0.32%	2.85%	3.17%	9.59	144.15	153.74
BTC	-0.37%	2.61%	2.24%	-721.9	3672.73	2950.83
GOOG	1.09%	0.76%	1.86%	111.92	152.23	264.15
ETH	-1.91%	3.49%	1.57%	-178.18	256.91	78.73
GSPC	0.21%	0.53%	0.74%	64.21	250.8	315.01
KO	0.75%	-0.08%	0.68%	5.53	-0.6297	4.7212
GM	0.29%	0.17%	0.46%	1.6595	1.7417	3.4012
FTSE	-0.19%	0.42%	0.22%	-321.44	758.08	436.64
Totals	**0.20%**	**14.62%**	**14.82%**	**-1028.4605**	**5309.962**	**4281.3224**

It is very evident that the hybrid system turned out to be the superior system. It managed to generate profitable returns whilst reducing the risks and minimizing the negative returns.

Considering that the standard industry benchmark, the S&P 500, made a loss of -7.01% during the year of 2018, an ROI of 14% on this portfolio is quite a satisfactory performance.

E. Hybrid Trading System with Genetic Algorithms

Genetic Algorithms are used to improve on the Hybrid Trading System, by optimizing the weightings assigned to each indicator. This will determine the importance given to the signal generated by each indicator, and thus could potentially open and close positions quicker, hopefully improving the amount of returns made.

TABLE VI
HYBRID TRADING SYSTEM WITH GENETIC ALGORITHMS RESULTS FOR YEAR 2018.

	Return on Investment			Profit/Loss		
	Buy	Sell	Total	Buy	Sell	Total
LTC	-5.13%	6.78%	1.65%	-67.07	38.11	-28.96
NFLX	0.65%	4.90%	5.55%	7.1	139.08	146.18
BTC	-9.97%	2.24%	-7.73%	-3517.43	487.11	-3030.32
GOOG	1.61%	0.04%	1.64%	89.49	10.1799	99.6699
ETH	-4.86%	21.85%	16.99%	-202.48	272.22	69.74
GSPC	1.14%	0.19%	1.33%	124.9802	68.5991	193.5793
KO	2.86%	-2.37%	0.49%	6.3485	-4.2225	2.126
GM	1.43%	0.42%	1.85%	5.8543	4.0393	9.8936
FTSE	-0.71%	0.44%	-0.27%	-481.78	738.76	256.98
Totals	**-12.98%**	**34.49%**	**21.50%**	**-4034.987**	**1753.8758**	**-2281.1112**

The results shown above are very interesting. Upon glancing at the results, one might quickly assume that the system did not perform well because of the large total loss made. However as usual, this large loss was due to the big losses in bitcoin.

When looking at the Return on Investment, in actual fact we can see that the system actually **outperformed** the original

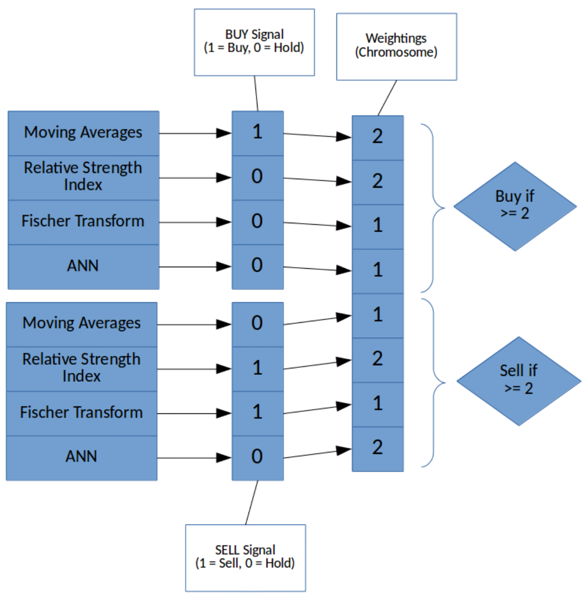

Fig. 12. Hybrid System using Genetic Algorithms, with sample parameters.

hybrid trading system, generating a total return of 21.5% as opposed to 14.82%. **This experiment clearly demonstrates the importance of examining the return on investment as opposed to absolute profits**.

We also have a clear example of this with Ethereum. When comparing the profit made in Ethereum in the original hybrid system, it might appear as if the Genetic Algorithms performed worse, since they generated a smaller profit. However, the ROI shows that the return was higher. This actually denotes that the Genetic Algorithms had better market timing, and managed to make a similar amount of profit by investing much less capital, which ultimately means taking less risk.

Another example to note is that of Litecoin. Although the results show a positive ROI, it also shows a negative profit (i.e. a loss). This is because the Returns on the buy and sell profits are calculated respective to the trades taken. Therefore, although the return of the sell trades was greater than the return loss of the buy, the absolute loss of the buy was still greater.

Therefore, overall Genetic Algorithms seems to be very promising and seem to have improved the trading performance. However, one can also note higher volatily and bigger losses, meaning it is still a more risky and unstable approach. The original hybrid trading system seemed to appear more stable and safer.

V. Conclusion

A. System Performance

Overall, all of the systems describe above showed a positive outcome, greatly surpassing the buy and hold strategy. The traditional signal processing analytic techniques worked quite well, generally being able to give a clear indication of the direction of the market. The combination of these indicators, including the Fisher Transform to detect cycles, the Relative Strength Index to determine over or undervalued stocks, as well as moving averages to show the trend, set a solid base for the trading system. The important concept to understand in this scenario is the volatility of the input signal, to determine whether a Low Pass Filter should be used or not to improve the results gained.

The Artificial Neural Network system, on the other hand, proved to be very promising. It was able to detect patterns and understand market movements to quite a satisfactory degree of accuracy, without being fed any rules or indications regarding market movements or conditions. The highest accuracy thus far achieved with the first approach to training the Neural Network was around 76%, performing relatively well in relation to the results gained from other systems. In [16], the authors gained an accuracy of 75.74% in their neural network prediction model, although the signal used was the Istanbul Stock Exchange. In [27], the author claims to have received a hit rate (or accuracy rate) of just over 50% when predicting stock market movements of the Oslo Stock Exchange 1 week in advance, also using an artificial neural network, which consists of 12 inputs. The author in [28] claims to have received an accuracy of around 88% when predicting stock market price fluctuations using an ANN, however the sample testing size used was merely 75 points, whereas the simulations run in this project were over around 2000 daily values. The historic testing data used was also pre 1996.

Genetic Algorithms also proved very promising, and managed to optimize the hybrid trading system by determining the size of the weights assigned to each indicator. This study however also clearly showed how susceptible they are to overfitting, and more research and experimentation should be performed before using such a system in live trading.

This prediction system in and of itself is quite commendable, but being applied practically in trading could prove to be effective especially in the long term. The system could continuously be trained and will learn to identify new patterns that might emerge over time. Stock markets are very dynamic by nature and therefore having a system that can adapt over time would be helpful.

The inclusion of AI into this trading system also clearly improved the accuracy of the system. The user may also decide whether to take a high or low risk approach when using the hybrid system, depending on the required criteria to open a position. Whilst most other studies feel the need to focus solely on one approach, the combination of factors, algorithms and indicators has shown to be effective.

B. Critical Analysis

The first factor that needs to be taken into consideration before applying this system in real-time is that of **transaction costs**. The system tends to open and close buy and sells more frequently than other systems, since it is dependent on 3 indicators at once. Therefore, taking transaction costs into consideration would give a slightly clearer and more realistic picture of what results would be obtained when trading live.

One important aspect that is missing from this evaluation is **risk**. When comparing the different trading systems, we are comparing solely based on profit, and not considering how much risk each trading system exposed itself to. This became very evident in the AI trading system, where it made decent returns for most assets but huge losses in cryptocurrencies. Calculating the **Sharpe Ratio** for each portfolio would allow us to make fairer comparisons and evaluations of the trading strategies.

Another factor that wasn't given much attention was that of the **trade sizes**. In our simulation, we simply used the current price as the trading size, meaning the trading system would have bought 1 share (or 1 coin), with each trade. This was done for simplicity's sake, as the ROI still allowed us to gain a good indication of the strategy's performance. However this of course does not make sense in terms of proper portfolio allocation, and exposes it to more risk.

C. Future Work

One factor that could potentially greatly improving the accuracy of the AI models, and thus improve the profitability of the system, is introducting another AI model that could automatically detect different market conditions. A number of conditions could be defined, such as 'Upwards Trending', 'Downwards Trending' and 'Sideways'. This would allow the models to be training specifically for each market condition. This system can then load the correct configuration based on the current market conditions, which would hopefully improve the accuracy of the predictions. Fuzzy logic is generally well suited for these kind of classifications. Decision trees might also come in handy for the trading system, so as to automatically switch between trading strategies depending on market conditions.

The Genetic Algorithm also has a lot of room for further experimentation and possible improvement. It's flexibility allows a wide range of different possible strategies to be tested. The biggest issue in this case was the long training times.

Another interesting approach would be to use Genetic Algorithms to enhance the regime switching operation. A similar approach could be taken to the one already used above. The GA chromosone could include both the types of Moving Averages to be used, as well as the window sizes. The GA will then be able to determine the optimal parameters to be used for the Moving Average crossover indicators.

Certain improvements on some of the analytic techniques have also been recorded and could be looked into. One example would be that of plotting the inverse fisher transform on the same graph as the fisher transform. Rather than solely relying on the upper and lower bounds as indications to close a buy or sell signal, the system might also use crossings between the two lines as indications to allow the buy or sell position to remain open slightly longer, as well as to close the position early in case of an unexpected fluctuation.

In order to improve the performance of the artificial neural network, it could be combined with other techniques. Possibilities might include clustering to spot outliers or general exceptions so as to filter them out and have a more accurate data sample.

References

[1] Etienne Ghys (2012). The Butterfly Effect. ICME-12,

[2] - http://mpe2013.org/2013/03/17/chaos-in-an-atmosphere-hanging-on-a-wall/

[3] - Hasselblatt, Boris; Anatole Katok (2003). A First Course in Dynamics: With a Panorama of Recent Developments. Cambridge University Press. ISBN 0-521-58750-6.

[4] G. Preethi, B. Santhi (2012). Stock Market Forecasting Techniques: A Survey. Journal of Theoretical and Applied Information Technology. ISSN: 1992-8645

[5] - Imdadullah. "Time Series Analysis". Basic Statistics and Data Analysis. itfeature.com. Retrieved 2 January 2014.

[6] - A. Kablan (2009). Adaptive Neuro-Fuzzy Inference System for Financial Trading using Intraday Seasonality Observation Model. World Academy of Science, Engineering and Technology

[7] - Md. Rafiul Hassan and Baikunth Nath, "Stock Market forecasting using Hidden Markov Model: A New Approach", Proceeding of the 2005 5th international conference on intelligent Systems Design and Application 0-7695-2286-06/05, IEEE 2005.

[8] S.Brown, W. Goetzmann, A. Kumar (1998). The Dow Theory William Peter Hamilton's Record Reconsidered. NYU Stern, Department of Finance, Working Paper Series 1998

[9] - A. Kablan, W. L. Ng. (2010) High Frequency Trading using Fuzzy Momentum Analysis. Proceedings of the World Congress on Engineering 2010 Vol I.

[10] A. Raudys, E. Malcius, V. Lenciauskas (2012). Moving Averages for Financial Data Smoothing. Vilnius University, Faculty of Mathematics and Informatics, Lithuania

[11] Currency Trader (2006), The Adaptive Moving Average.

[12] J. Ehlers (2004) Cybernetic Analysis for Stocks and Futures - Chapter 1. Wiley, USA

[13] The Technical Analyst (2010) Relative Strength Index. The Technical Analyst issue Jan Mar 2010

[14] Z. Yudong, W. Lenan (2008) Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. Expert Systems with Applications

[15] S. Nayak, B Misra, H Behera (2014) Impact of Data Normalization on Stock Index Forecasting. International Journal of Computer Information Systems and Industrial Management Applications

[16] Y. Kara, M. Boyacioglu, O. Baykan (2008) Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. Expert Systems with Applications

[17] S. Niaki, S, Hoseinzade (2013) Forecasting S&P 500 index using artificial neural networks and design of experiments. Journal of Industrial Engineering International

[18] - M.J.C. Hu (1964) "Application of the Adaline System to Weather Forecasting," Master Thesis, Technical Report 6775-1, Stanford Electronic Laboratories, Stanford, CA

[19] - P.J. Werbos, "Generalization of back propagation with application to a recurrent gas market model," Neural Networks, vol. 1, pp. 339-356, 1988.

[20] Investopedia, "Financial Market", http://www.investopedia.com/search/default.aspx?q=stock%20market [Accessed on 04/05/2019].

[21] Investorama, Yahoo Finance, "How Stock Markets Work", https://finance.yahoo.com/education/stocks/article/101017/How_Stock_Markets_Work [Accessed on 04/05/2019].

[22] The University of Wisconsin, "A Basic Introduction to Neural Networks", http://pages.cs.wisc.edu/˜bolo/shipyard/neural/local.html [Accessed on 04/05/2019]

[23] - Peters, E. (1996). Chaos & order in capital markets: A new view of cycles, prices, market volatility. John Wiley & Sons Inc.

[24] - R Majhi, & Panda, G., (2007). Stock market prediction of S&P 500 and DJIA using bacterial foraging optimization technique. In 2007 IEEE congress on evolutionary computation (CEC 2007) (pp. 25692579).

[25] Investopedia, "Standard & Poors 500 Index", http://www.investopedia.com/terms/s/sp500.asp [Accessed on 06/05/2019]

[26] P. Wasserman, “Neural Computing: Theory and Practice“, Chapter 3. Coriolis Group (Sd); First Edition edition (June 1989)
[27] R. Aamodt, “Using Artificial Neural Networks to Forecast Financial Time Series“, Norwegian University of Science and Technology
[28] A. Kulkarni, “Application of Neural Networks to Stock Market Prediction“, Technical Report, 1996.
[29] - Berutich, Lpez, Luna, and Quintana. “Robust Technical Trading Strategies Using GP for Algorithmic Portfolio Selection.“ Expert Systems With Applications, 2016, Vol.46, pp.307-315
[30] - Y. Kim, W. Ahn, K.J. Oh, D. Enke, “An intelligent hybrid trading system for discovering trading rules for the futures market using rough sets and genetic algorithms“, Applied Soft Computing Journal, 2017, Vol.55, pp.127-140
[31] - J.H. Holland, “Genetic Algorithms“, Scientific American, Vol. 267 (1), 1992, pp. 66-73