Algorithmic Trading

Francis Sunny & Geremy Bantug

Background of the Approach

Algorithmic Trading is the approach by which, trading of stocks is executed with predefined set of rules or rules generated using sophisticated algorithms. The approach here is to make use of the evolutionary techniques such as Genetic Programming to predict the price of a stock and then with the help of several technical indicators like moving averages and relative strength indices to generate trading rules in such a way to maximise the profit over a period. Also genetic algorithm is applied to optimise the parameters of the technical indicators.

The approach used here uses a sliding window approach and Grammatical Evolution to generate a formula that fits the trend of the stock and using the same formula to predict the future values of the stock. The stock values will be predicted for a certain period and used to apply the trading rules implemented using the technical indicators such as Moving Averages, RSI and MACD.These trading rules will be applied to generate signals that indicate when to buy and sell stock.

The literature "System for foreign exchange trading using genetic algorithms and reinforcement learning" 2007 by A. Hryshko & T. Downs employs a similar trading strategy to the one explored in this report. This literature focuses on employing machine and reinforcement learning along with technical analysis with optimisation of technical trading rules through genetic algorithms. Instead of approximating future trades using genetic programming, the literature focuses on technical analysis to make these predictions.

Dataset Chosen

The stock selected for analysis is: NFLX.

Period for which preliminary analysis done is from 2019-01-01 to 2021-01-01

The Netflix (NFLX) stock was chosen as the dataset for this experiment as it has a general upwards trend and the reason for only choosing one stock is to decrease the run time. However, this strategy could be applied to a portfolio instead of focusing on one stock which could be explored in future work. The data range was chosen so that there is plenty data from which the GrammaticalEvolution function can use to test and train that captures the general trend of the stock.

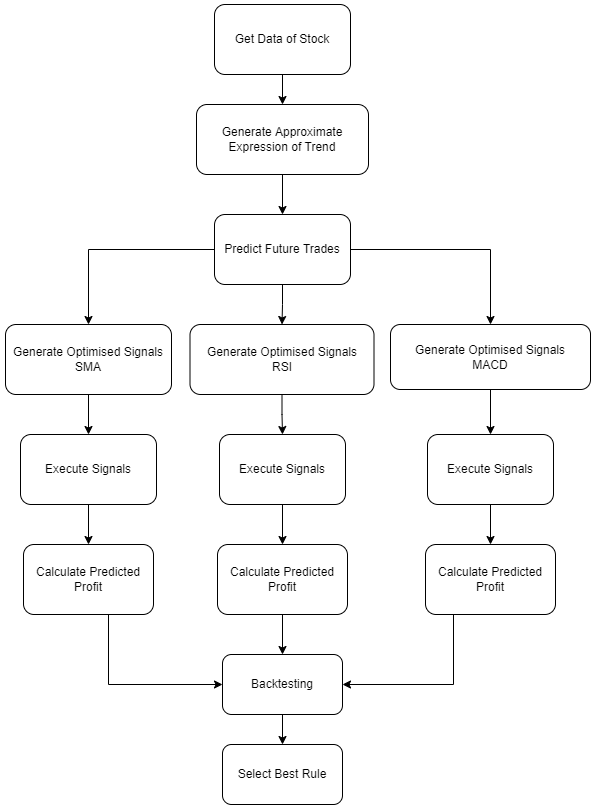
Trading Strategy

The approach designed here is to find the estimated profit that can be made on a single stock (NFLX) by predicting the future prices of the stock data up-to a particular number of days. Strategy here is purely based on technical analysis of the few of the technical indicators. In an ideal scenario this technical analysis should be combined with further fundamental analysis on the stock and market conditions to make proper predictions and building the trading rules.

The strategy designed here could be further improved by combining using other technical indicators, combing the signals from different technical indicators, splitting the capital available to different strategies and combining the signals to include more trades and get more profit. Instead of making such complex rules, approach taken here just finds the rule which gives max profit and apply that for the prediction period. After which, the model can be retrained to predict the next set of predictions.

Since this approach is only dealing with one single stock, it only considers buying the maximum number of stocks that can be bought with amount in hand and selling the same number of stocks. It doesn't consider buying again in between this buy and sell period.

Block Diagram of Trading Strategy



Prediction of Future Data

In this section the prediction of future data is explored through an implementation of genetic programming using the R library gramEvol and a function called GrammaticalEvolution. The prediction is done through a sliding window approach, where a window size of 10 is used to lag the input data. With this lag of data for date ranges previously selected when gathering the data, is then split into test and train samples in which the GrammaticalEvolution function can learn the data and identify an approximate expression that fits the trend of the data. To build the expression another functionality of gramEvol is used, to state the parameters of the rules which is called CreateGrammar. These parameters include constants, variables, operands and functions which can be used to build the expression that fits the trend of the data. The variables used for creating the grammar is the lag data generated from the window sizes.

With the rules created using CreateGrammar, evaluate the expression generated and find its fitness through RMSE which is the fitness function. On iterating through the search space for possible expressions GrammaticalEvolution helps to identify the best expression with minimum cost.

Once the best expression is found, the next step is to predict the future trade data. For this approach only the Open and Close prices of the day are required.

Technical Indicators

SMA Crossover

To identify when to buy or sell stock technical indicators such as SMA Crossover, RSI and MACD were implemented and were optimised using genetic algorithm with the R library GA. To obtain signals for SMA Crossover an SMA function was used from the R library TTR, in which the two parameters considered where the windows for the short and long SMA of the combined initial train data and predicted closing price values. This is so that the SMA can fully capture the general trend of the stock and consider the predicted future data. A buy signal is generated when the short SMA crosses above the long SMA and a sell signal is generated when the short SMA crosses below the long SMA.

The graph explains the SMA with a short moving average of 20, represented with blue line, and long moving average of 50 represented in red line. In the approach here these short and long values may not be used to generate signals and execute the trades, but instead these values would be replaced with the optimal parameters obtained from the GA optimiser designed.

Executing Trade With Optimal Parameters for SMA

To optimize the values of long and short moving averages, GA is utilised to identify these parameters which generates maximum profit from the range of trades in the predicted data. The GA is designed to iterate through a low and upper bounds for each short and long moving average in the range of values from 2 to 100 and take the best combination from these values. The upper bound was selected as 100 because this approach uses only a small prediction window.

The best values from the range of upper and lower bounds are determined by the fitness function designed. The fitness function is designed to maximise the profit obtained for each set of SMA signals and minimising the fitness whenever an unlikely condition arises. Unlikely conditions identified the fitness function are:

1. Short moving average greater than long moving average.
2. Difference between long and short less than a value of 10. Or else the two moving averages would be very close to each other.
3. Profit is less than 0 or a null value or previous profit value.

Optimal Parameters for short and long SMAs: 8, 43.

Profit from SMA Crossover Trading Rule: 745.2203425

Relative Strength Index (RSI)

To generate the RSI values a similar approach to the SMA Crossover is implemented but with the use of the RSI function of the TTR library. As a rule it is good to buy stock when RSI is less than 30 and sell when greater than 70 [source].

Executing Trade With Optimal Parameters for RSI

To optimize the low and high RSI values and window of days, a similar GA previously designed, is utilised to identify these parameters which generates maximum profit from the range of trades in the predicted data.

The best values from the range of upper and lower bounds are determined by the fitness function designed. The fitness function is designed to maximise the profit obtained for each set of RSI signals and minimising the fitness whenever an unlikely condition arises. Unlikely conditions identified the fitness function are:

1. Low RSI greater than high RSI.
2. Difference between low and high RSIs less than a value of 10. Or else, the two signals would be very close to each other.
3. Profit is less than 0 or a null value or previous profit value.

Optimal Parameters for low and high RSIs: 42, 55.

Optimal Parameter for window size: 9.

Profit from RSI Trading Rule: 1221.7989457

Moving Average Convergence Divergence (MACD)

The MACD implementation also follows a similar approach to both RSI and SMA Crossover with the use of the MACD function of the TTR library. MACD signals is gathered from the crossover of EMA and SMA trend lines. MACD implemented using the TTR library uses SMA to calculate long moving average, EMA to calculate short moving average and signal is calculated using exponential smoothed line [source].https://bookdown.org/kochiuyu/technical-analysis-with-r-second-edition/macd.html

Executing Trade With Optimal Parameters for MACD

To optimize the long and short moving averages and signal values, a similar GA previously designed is utilised to identify these parameters which generates maximum profit from the range of trades in the predicted data.

The best values from the range of upper and lower bounds are determined by the fitness function designed. The fitness function is designed to maximise the profit obtained for each set of MACD parameters and minimising the fitness whenever an unlikely condition arises. Unlikely conditions identified the fitness function are:

1. Short EMA greater than long SMA.
2. Difference between short and long MAs less than a value of 10. Or else, the two parameters would be very close to each other.
3. Profit is less than 0 or a null value or previous profit value.

Optimal Parameters for short EMA and long SMA: 43, 54.

Optimal Parameter for exponential smoothed : 10.

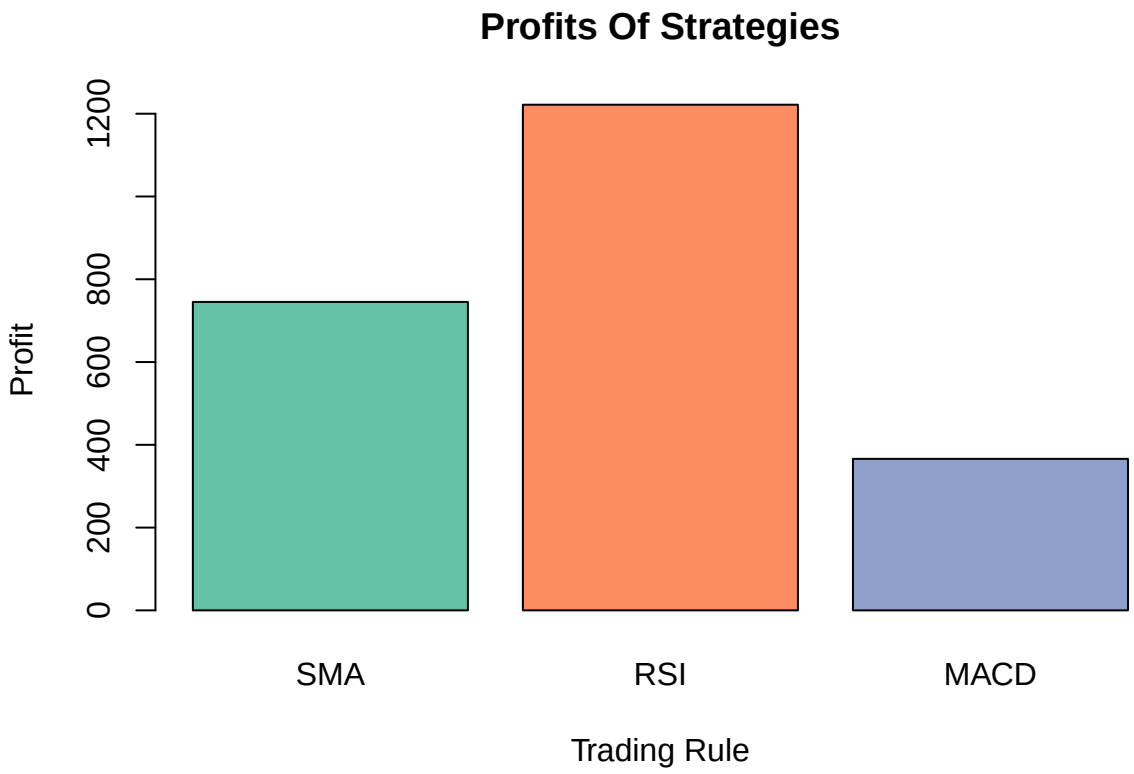
Profit from MACD Trading Rule: 1221.7989457

Performance Evaluation

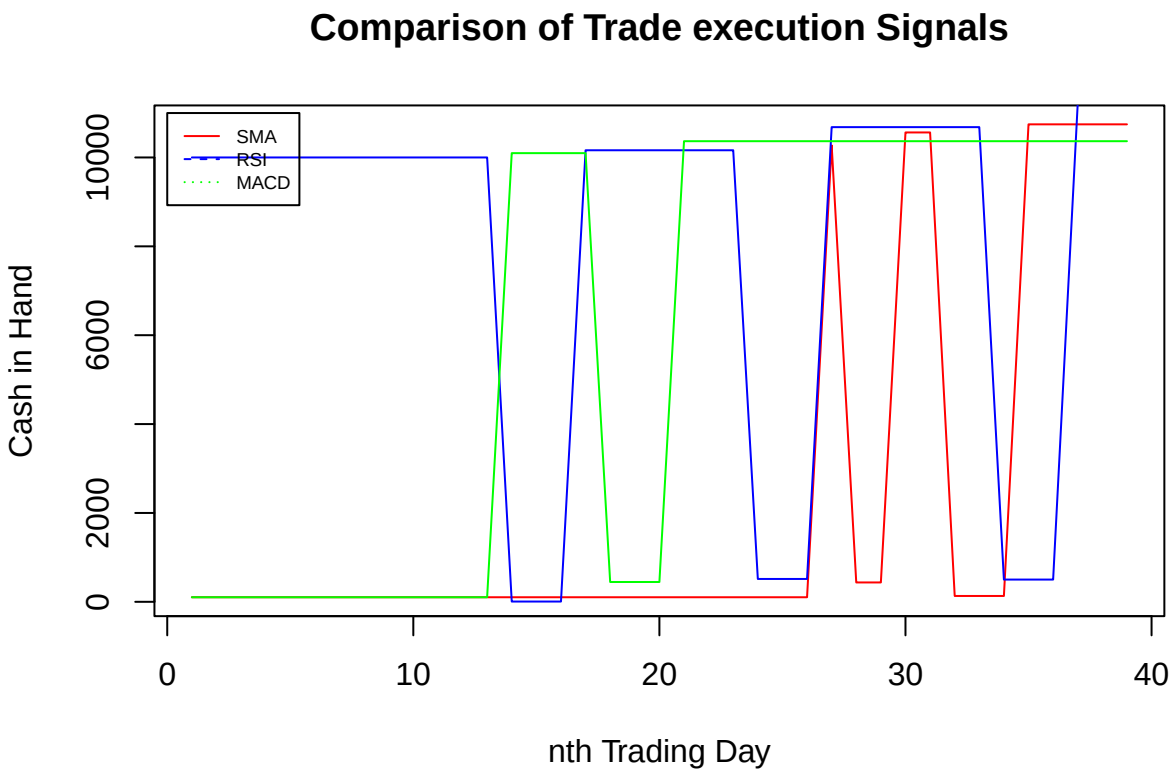
Buy and Sell Function

Once the signals have been generated from the technical indicators, they are then passed into a function as an data frame input consisting of signals, predicted closing and opening prices. The holding signals for the position taken is generated inside this function. This holding signal is then used for executing the trade ie. Buying and selling of the stock. This approach assumes to have a capital of $10000 at the beginning. The indicators used here are built using the closing price of the training stock data, so the buy and sell function uses these signals to buy and sell the maximum number of stock possible at the next opening price of each signal.

Comparison of Technical Indicator Rules



Barplot indicates that the technical trading indicator that produced the highest profit is RSI. The profit made from the technical indicator using the predicted trade data is 1221.7989457.



The line plot above explains the variation of signals of the holding position for each technical indicator. SMA is represented using red line, RSI with blue and MACD with green line.

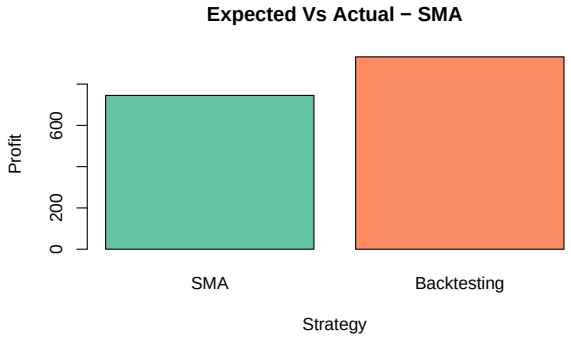
Backtesting

In back testing the signals obtained from each technical indicator is used to execute the trades for the actual data and the profit at the end is compared with the profit obtained from predicted data.

To carry out back testing, getSymbols is used to retrieve the actual trade data in the range of the predicted data and the signals from each technical trading rules are applied to the actual data. A comparison of the profits are then made to identify the percentage error of the predicted profits to the actual profits.

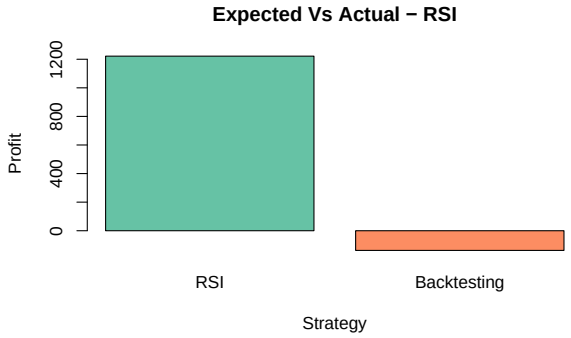
Backtesting On SMA Signals

The signal obtained from the executing the SMA trade is used for actual data and the profit is calculated. The plot below shows the difference between expected and actual profits with the signal generated.



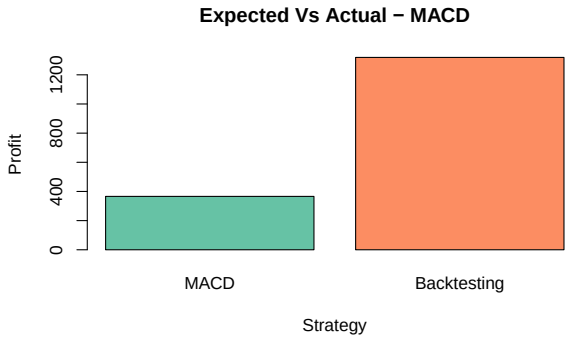
Backtesting On RSI Signals

The signal obtained from the executing the RSI trade is used for actual data and the profit is calculated. The plot below shows the difference between expected and actual profits with the signal generated.



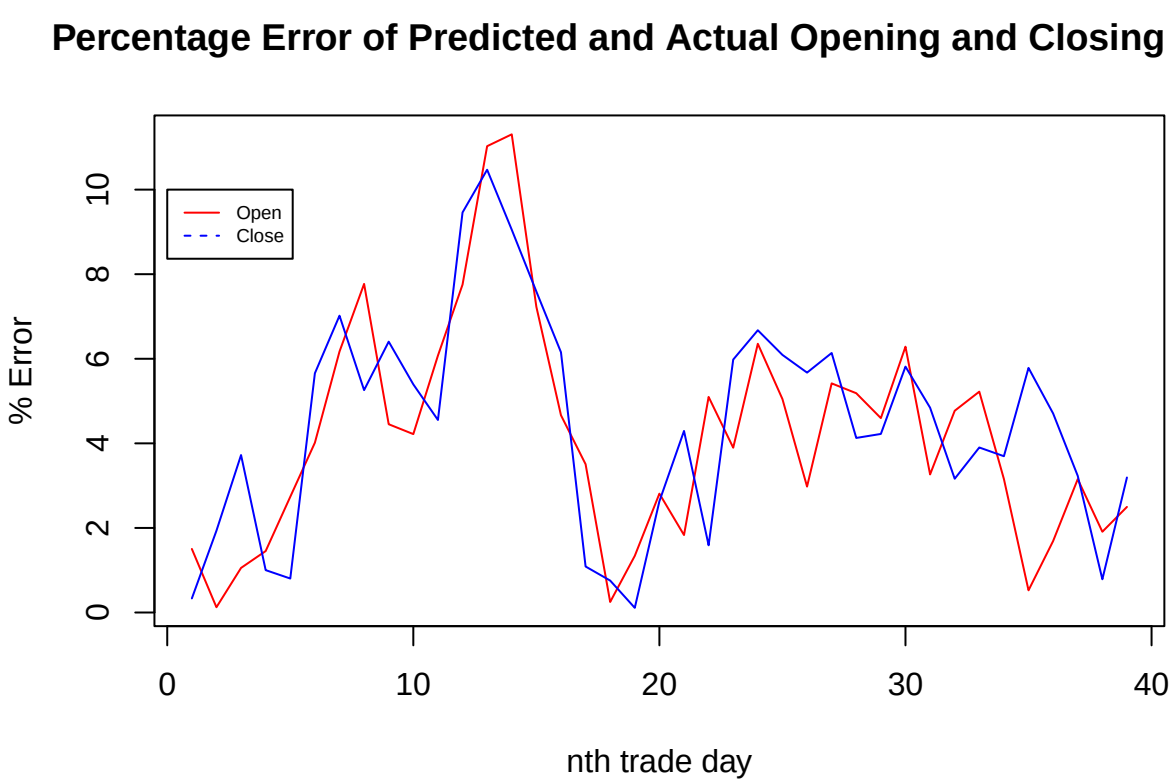
Backtesting On MACD Signals

The signal obtained from the executing the MACD trade is used for actual data and the profit is calculated. The plot below shows the difference between expected and actual profits with the signal generated.



Error Analysis

Barplot indicates that the technical trading indicator that has least percentage error, therefore being more accurate is SMA.



To identify how accurate the GrammaticalEvolution was at predicting the future nth trade, the percentage error of the predicted closing and opening price was calculated from the actual real historical data. These errors are plotted in a line graph which indicates that the errors increase after the 5th day. This means that the longer predictions will result in less accurate trades.

The average Error in Opening price prediction is 2.1898544.

The average Error in Closing price prediction is 2.2240841.