Algorithmic Trading

Calvin Hew

Background

The use of Algorithmic Trading. (AT) as a tool for automating the process of trading stocks remains dominant since it was first utilised over two decade ago. The main appeal of algorithmic trading is the ability to trade stocks reliably without human intervention (Chaboud et. al., 2014). The key factor being **reliable** as systems have only become more sophisticated with the advances in AI techniques (Pothumsetty, 2020). Moreover, a major argument for using AT is the benefit of eliminating human biases, which typically hinders trading decisions (Das and Kadapakkam). As a result, the aim for this project is to propose algorithmic trading system that can reliably make decisions to gain a profit.

Existing Systems

The notable success of Neural Networks (NN) predictions when used with time-series data (Fan et. al., 2024) suggests that it may perform well when presented with trading data. This is confirmed by Evans et. al. (2013) who proposed a strategy implementing a NN with a Genetic Algorithm (GA) which achieved an impressive 23.3% for annualised net profit.

Trading Indicators

Equally important to forecasting prices is the consideration of indicators as part of the process of formulating trading rules. A popular indicator is the MACD (Moving Average Convergence/Divergence indicator) which utilises the relationship between 2 moving averages (MACD + signal lines) to identify changes in momentum (Dolan, 2024). There are a number of prominent MACD trading strategies with the most popular being: the Crossover Strategy and the Zero-Cross Strategy (Schlossberg, 2024). As implied, the Crossover strategy looks at the intersection between the signal and MACD lines - generating either a bullish signal (for buying) or bearish signal (for selling). The Zero-cross sends a bullish signal whenever it inspects the crossover of the MACD line from below the zero line to above.

Approach

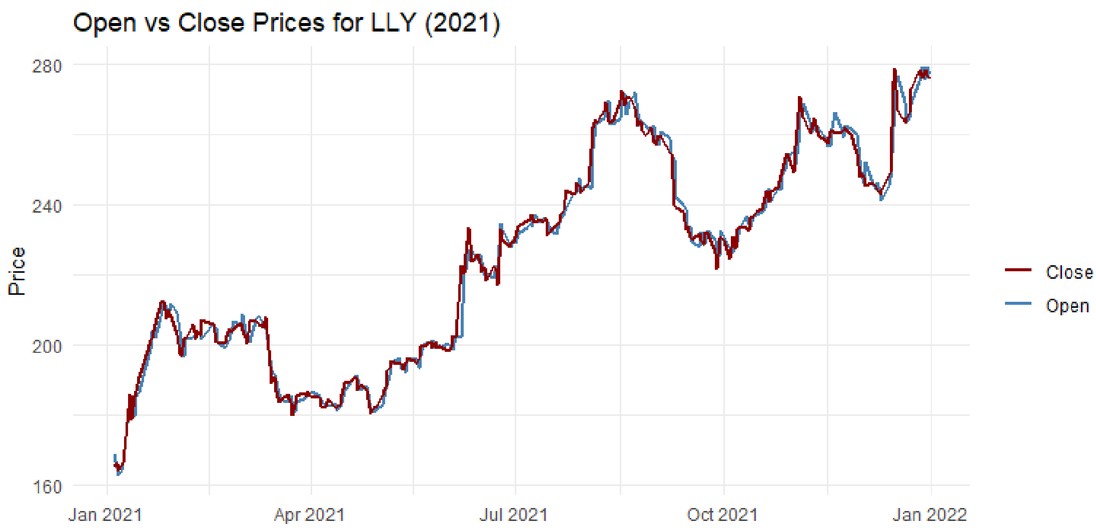
The technique proposed in this study is one of price prediction-based trading. Justified by the previous discussions, this project will implement a NN for forecasting prices. Based on these predictions, the MACD indicator and trading rules, the Zero-Cross strategy will be implemented. The proposed system and strategy will automate the decision process of entering or exiting a market for a particular stock. Finally, its performance will be evaluated against a competing strategy based on filter rules.

Data

```
set.seed(100)

asset <- "LLY"
data <- getSymbols(Symbols = asset, src = "yahoo", from = "2021-01-01",
                   to = "2022-01-01", period = "daily")
```

The stock of Eli Lilly and Company (LLY) demonstrated a consistent upward trajectory in the year 2021, reflective of robust financial outcomes and favorable investor confidence. Such a trend renders LLY a suitable candidate for the empirical assessment of trading strategies, thereby providing a substantial case for evaluating the robustness of forecasting algorithms amidst its steady market ascent.

A plot of this asset's open and close prices shows the overall rise over the trading period:



The period of 2021-01-01 to 2021-12-31 was selected as it acts a better indicator of the current and future market trends. Moreover, stable stocks with few price oscillations (such as Google and Microsoft) were avoided as it is not effective in testing the robustness of the Neural Network.

Processing data involves lagging and scaling the data. A lag of 3 is applied - which is important for a price-prediction approach as it can help to capture trends and dependencies within the data which are valuable for detecting future patterns. NaN values also needs to be addressed before feeding to the NN.

```
lag = c(1:3)
```

Trading Strategy

Neural Network

The neural network (NN) will be utilized to forecast the opening and closing prices of selected assets. The accuracy of the NN's predictions will be measured using the common evaluation metric of Root Mean Squared Error (RMSE).

```
train_nn <- function(train_data) {
  col_name <- colnames(train_data)
  core_cols <- col_name[1:2]


  f <- as.formula(paste("close + open ~", paste(col_name[!col_name %in%
core_cols],
                                               collapse = " + ")))
  hidden_layers <- c(8,5,2)
  threshold <- 0.006

  set.seed(100)
  nn <- neuralnet(f, data=train_data,  hidden=hidden_layers,
                  threshold = threshold, stepmax=1e6)

  return(nn)
}
```

The NN predicts both Open and Close prices for the selected asset. There are arguments justifying either approach of using Open or Close prices in the context of MACD trading strategies. Closed is typically favoured for minimising risks as it better reflects trade activity. Open gives insight to potential early trends but suffers from more volatility.

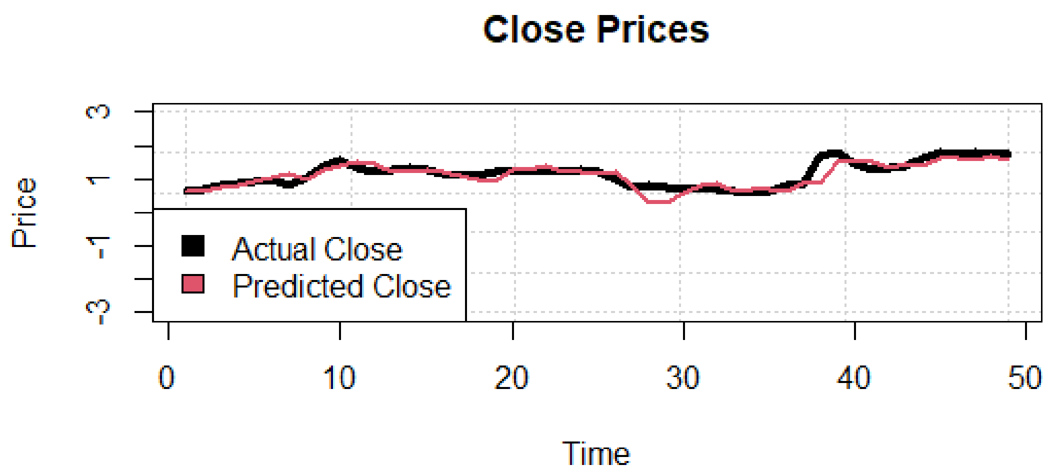
The formula parameter when expressed in full is: f = close + open ~ c1 +c2 +c3 + o1 +o2 +o3 which allows the network to separately predict both open and close prices.

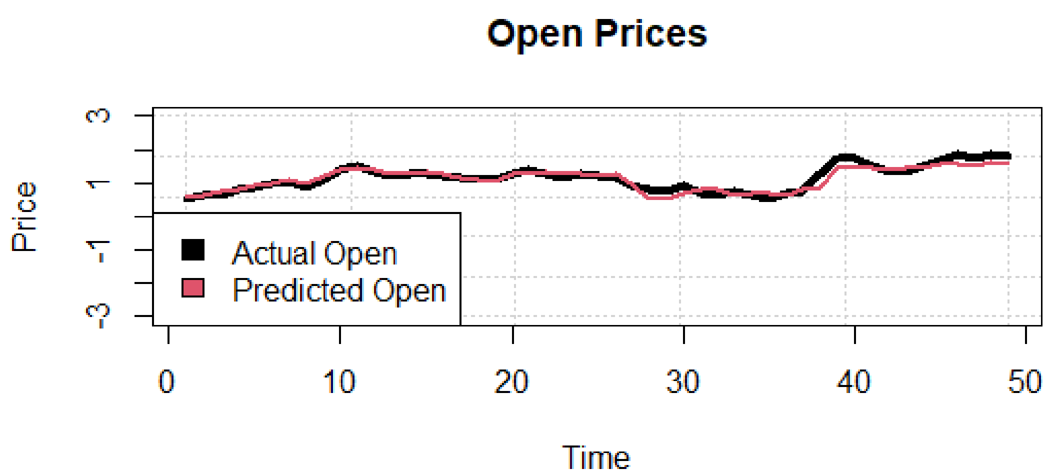
```
### training
train_size <- 200
price_data <- processPrices(price_close, price_open, lag, train_size)
train_data <- price_data[[1]]
test_data <- price_data[[2]]
```

Though the obvious approach may seem to be investing in the best performing stocks from the previous year, trading experts often argue against this naive strategy (Ermey, 2023). Thus, instead of training over an annualised trading period, a less risky approach justifies using a test-train split. A train size of 200 is used to ensure there is enough test data to be used for back-testing.

To evaluate the performance of the NN predictions, the metric of Root Mean Squared Error (RMSE) can be calculated. The RSME compares the predicted value against the real data for quantifying the error in the NN - therefore a smaller value is more desirable. The results presented show the respective RMSE of train vs. test data for both open and closed prices. Though the predicted test data for close price contains the highest error, values between 0.2 and 0.5 are generally acceptable as accurate predictors in ML contexts.

```
nn = train_nn(train_data)
### prediction
scale_params <- price_data[[3]]
results = predict_nn(nn, train_data, test_data, scale_params)
```






```
## [1] "RMSE Close Price (Train): 0.098206629611806"
## [1] "RMSE Close Price (Test): 0.204264782428813"
## [1] "RMSE Open Price (Train): 0.06057764048108"
## [1] "RMSE Open Price (Test): 0.139488512653992"
```

Trading Rules

The `macdTrading` function serves as an automated trading system with a $10,000 starting fund, trading with the "LLY" stock and using the MACD Zero-line cross strategy. Based on the NN's predicted prices, it scans for when the MACD line crosses above the zero line, indicating an uptrend and a cue to buy 100 units of stock, and identifies crosses below the zero line as downtrends, signaling to sell and secure positions.

Typically the MACD parameters are 12-26-9 for fast, slow and signal respectively. However, the recommended setting for day trading is suggested to be 3-10-16 (AvaTrade, 2024). This will be used for the proposed approach as short term trading is more conducive of the limited data with a train/test split.

```
macdTrading <- function(asset, prices) {
  # reset strategy
  rm.strat(name = 'macd', silent = TRUE)

  # initial funds
  init_eq = 10000
  currency('USD')
  stock(asset,currency='USD',multiplier=1)
  portfolio.st <- 'macd'
  account.st <- 'macd'

  # initialise portfolio, account and orders
  initPortf(name=portfolio.st,
            symbols=asset)
  initAcct(name=account.st,
           portfolios=portfolio.st,
           initEq=init_eq)
  initOrders(portfolio=portfolio.st)

  strat <- init_macd_strategy(prices, 3, 10)

  out<-try(applyStrategy(strat, portfolio.st, parameters=list(nFast=3,
nSlow=10, nSig=16, maType="EMA"), verbose=TRUE))
  updatePortf(Portfolio=portfolio.st)

  return(portfolio.st)
}
```

The zero-line cross strategy offers significant advantages by leveraging precise timing for market entries and exits, aiming to amplify the initial investment through astute predictions of price movements. Its design emphasizes efficiency and simplicity, automating trading decisions to focus on profitability and risk management.

In the proposed system, a bullish signal is generated if the MACD is identified to cross the zero line from below whilst a bearish signal is provided if the zero line is crossed from above. This can be achieved through the use of the `quantstrat` package.

The trading rules then dictate a stock to be purchased in the case of bullish signals and sold under bearish signals. Finally, performance of the trading strategy can be evaluated by plotting the trading positions and analysing the trading statistics (such as Net P/L) using `tstats <- tradeStats(portfolio)`.

This approach not only simplifies the trading process but also ensures that every decision is data-driven and aligned with overarching investment goals, all within a framework that meticulously tracks and evaluates the strategy's performance over time.

The alternative crossover strategy offers the advantage of capturing significant trends, potentially leading to profitable trades when these stocks begin a new trend. However, the inherent drawback is the lagging signals that might result in entering or exiting trades later than optimal. For "LLY", this could mean missing the initial movements of a promising trend or holding onto a position too long when the trend reverses, affecting the overall profitability and risk management of your trading strategy.

Thus, the proposed strategy offers a systematic approach to trading by leveraging the MACD indicator to automate the identification of potential buy and sell signals in predicted price data. While such strategies hold promise for investors seeking to capitalize on market trends, it's crucial to recognize the inherent risks involved in trading and to conduct thorough testing and evaluation before deploying strategies in live trading environments.

Performance on Training Data

Initially, the trading strategy's performance should be evaluated in the context of close and open prices.

The `tradeStats()` function provides helpful statistics of the trades conducted with the MACD strategy for both open and close prices. The main metric of interest, undoubtedly, is the net profit/loss. However, it can be helpful to analyse the range of statistics available.

Open Prices

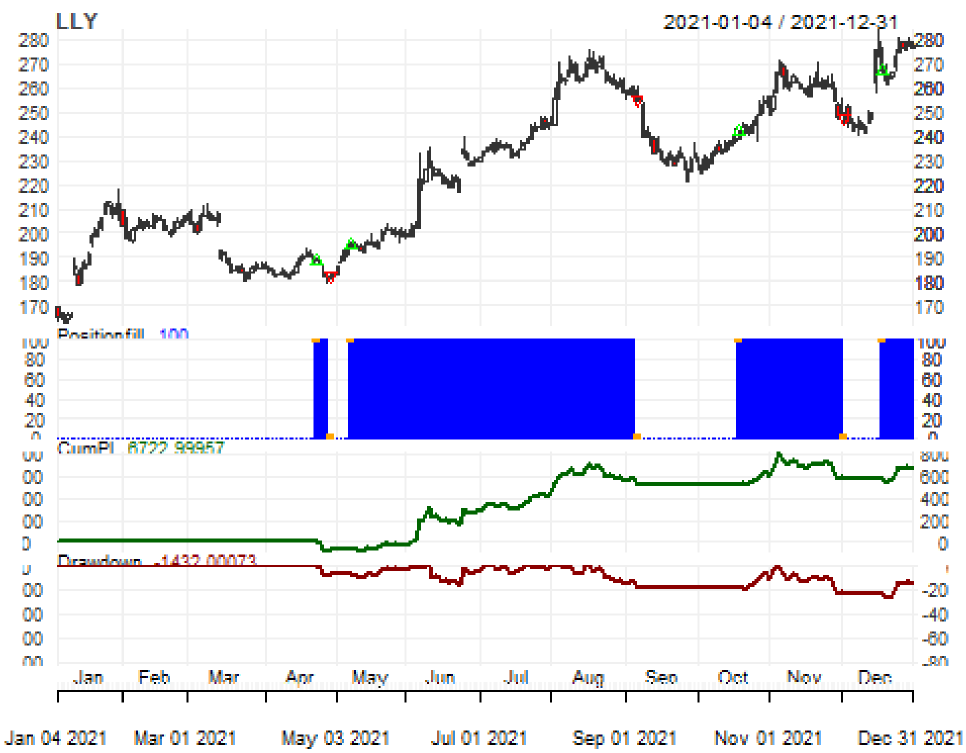
```
## [1] "Net Profit from MACD (Train Op Price): $5407"

## [1] "Avg Profit from MACD (Train Op Price): $1377.67"

## [1] "No. of transactions from MACD (Train Op Price): 7"

## [1] "Largest Winner from MACD (Train Op Price): $4458"
```

Close Prices

```
## [1] "Net Profit from MACD (Train Cl Price): $6723"

## [1] "Avg Profit from MACD (Train Cl Price): $1947.67"

## [1] "No. of transactions from MACD (Train Cl Price): 7"

## [1] "Largest Winner from MACD (Train Cl Price): $5938"
```

Comparing the results of the open vs close price, it is unsurprising that the close prices performed better overall. Both conducted 7 transactions in total but close prices ended up with more profit. This reinforces the argument to use close prices with MACD approaches as they are less susceptible to volatile market conditions. Open prices may be considered better for identifying early trends for a more stable asset. However, as LLY has seen some dips and rises throughout the selected trading year, it can explain the comparatively worse performance of open prices. As a result, it is decided that open prices will be dropped moving forward with MACD trading using test data.

Visualising P and L: Train Data (Cl Prices)



Performance on Unseen Data

The approach taken for backtesting is utilising a test-train split over the trading period of 2021-01-01 to 2021-12-31.

Profits from Trading: Test Data

First, it may be helpful to be recall the evaluation of the NN training: train MSE was 0.098 vs. test MSE was 0.204. That being said, test data is expected to have a higher rate of error in its prediction when compared to the data the model was trained on.

Thus, it is expected that predictions from the test set would result in comparably lesser net profits vs. the train set. However, it may be surprising to learn that this is not the case:

Profits from Train Set

```
## [1] "Net Profit from MACD (Train): $6723"

## [1] "No. of transactions from MACD (Train): 7"
```

Profits from Test Set

```
## [1] "Net Profit from MACD (Test): $7117"

## [1] "No. of transactions from MACD (Test): 7"
```

At first it may seem surprising that the predictions from the test data resulted in a superior profit of $7117, surpassing the predicted train profits, which stood at $6723. Despite the perceived inferiority of the test predictions relative to the actual test data, it is conceivable that the forecasted test prices more accurately encapsulate forthcoming market trends.

This can be confirmed through running the MACD strategy against the real data (what the NN was originally trained on) and observing the profits returned.

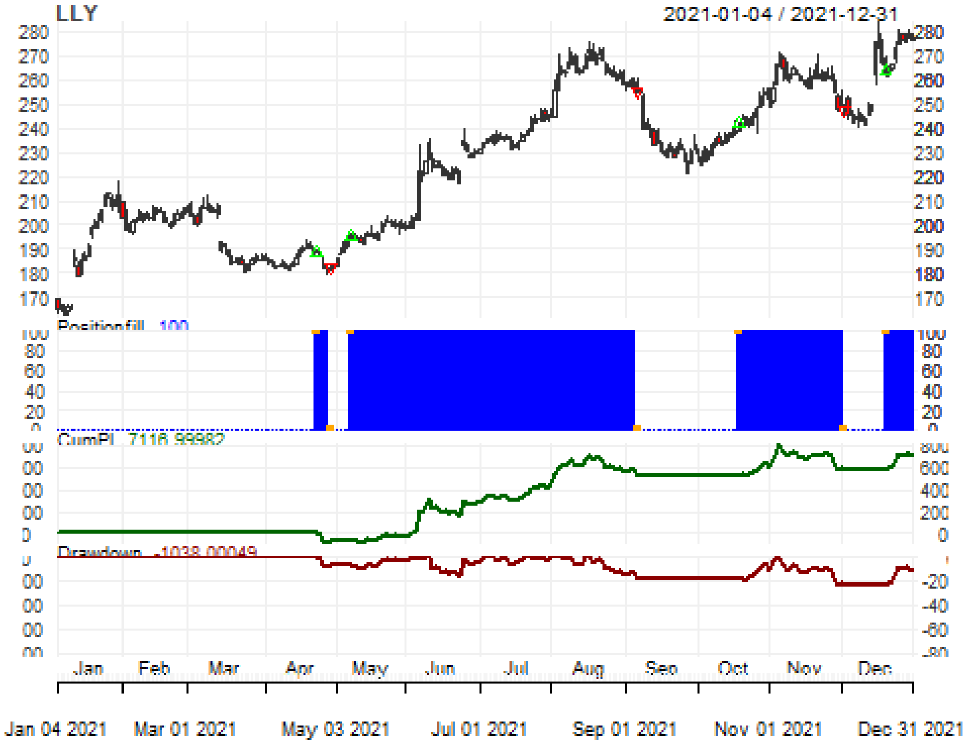
Profits from Real Set

```
## [1] "Net Profit from MACD (Real): $6894"

## [1] "No. of transactions from MACD (Real): 7"
```

As expected, the profits achieved from the MACD strategy using real data reflects the hypothesis that the predicted future data is a better indicator of profitable investments. Granted, the difference in profits is relatively minimal - this is also expected as the NN model was evaluated to be a reasonable predictor.

Visualising P and L: Test Data (Cl Prices)

Upon first look, it does appear that this graph looks almost identical to the visualisation of profits using Train Data. However, upon closer inspection, one can notice the slight dip in the Train graph towards the end of December as opposed to the continous incline identified in this graph for Test data.



Alternative Approach

Filter Rule Strategy

To compare against the performance of the original MACD Zero-Cross strategy, a simple filter rule strategy was implemented. The concept of a filter rule is fairly straightforward: send a signal to buy/sell based on the belief that stocks on the rise will continue to increase and stocks on the decline will continue to fall (Mitchell, 2022).

Moreover, as the MACD was traded with short-term trading fast, slow and signal parameters, this filter rule strategymakes for a good benchmark to compare the MACD against. However, there is no built in function so it was necessary to define a custom function to compare the % change from the previous day:

```
# define custom indicator
 custom_filter <- function(price) {
   lagprice <- lag(price,1)
   temp<-price/lagprice - 1
   colnames(temp) <- "filter"
   return(temp)
 }
```

Similar to the original approach, this strategy will utilise the same test-train split and close prices to ensure the comparisons are consistent. This strategy requires one additional parameter which acts as the threshold reflecting the % shifts which signal buy or sell.

In the process of deciding the threshold %, it was noticed that even a 1% change can have significant impacts on the signal generated. The 6% threshold reaps significantly worse profits of $3861 vs the 7% which generates $6695. An 8% threshold was too high as there were no price changes above 8% which lead to no trade. Thus, the sweet spot appears to be 7%.

Comparison

To conclude, the Zero-cross strategy implemented using the MACD indicator and NN has produced some promising results. As shown in the comparison below, the MACD approach using forecasted prices reaped the highest net profit when compared against another short term trading strategy based on filter rules. This holds true for both the test and train sets, but it is especially impressive for the test set which achieved the best overall results.

Train Set	MACD	Filter (6%)	Filter (7%)
Net Profits ($)	6723	5533	5533
Transactions	7	4	4

Test Set	MACD	Filter (6%)	Filter (7%)
Net Profits ($)	7117	3861	6695
Transactions	7	5	4

This project may be extended by attempting even more different trading strategies - perhaps one that favour long-term trading to compare how it fares against this study's proposed approach. Moreover, the scope of this project only covered one asset - LLY which was known to have performed well in 2021 and so it may be interesting to see how this strategy would perform trading a stock that did not see growth. Coverage of trading periods may also be a possible avenue to explore as this system performs well in short-term trading scenarios but not necessarily in the long-term.

References

AvaTrade, 2024. MACD Trading Strategies. Technical Analysis Indicators & Strategies. AvaTrade. https://www.avatrade.co.uk/education/technical-analysis-indicators-strategies/macd-trading-strategies

Chaboud, A.P., Chiquoine, B., Hjalmarsson, E. and Vega, C., 2014. Rise of the machines: Algorithmic trading in the foreign exchange market. The Journal of Finance, 69(5), pp.2045-2084.

Dolan, D., 2024. What Is MACD?. The moving average convergence/divergence indicator helps investors identify price trends. Investopedia. https://www.investopedia.com/terms/m/macd.asp

Das, S. and Kadapakkam, P.R., 2020. Machine over mind? Stock price clustering in the era of algorithmic trading. The North American Journal of Economics and Finance, 51, p.100831.

Ermey, R., 2023. Investing in last year's top 10 stocks is 'a recipe for disaster,' expert says. CBNC. https://www.cnbc.com/2023/01/14/best-performing-stocks-of-2022.html

Evans, C., Pappas, K. and Xhafa, F., 2013. Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation. Mathematical and Computer Modelling, 58(5-6), pp.1249-1266.

Fan, W., Zhang, R., He, H., Hou, S. and Tan, Y., 2024. A short-term price prediction-based trading strategy. Plos one, 19(3), p.e0294970.

Mitchell, C., 2022. Filter Rule: Meaning, Parameters, Example. Investopedia https://www.investopedia.com/terms/f/filterrule.asp

Pothumsetty, R., 2020. Application of artificial intelligence in algorithmic trading. Int. J. Eng. Appl. Sci. Technol., 4(12), pp.140-149.

Schlossberg, B., 2024. How to Trade the MACD. Investopedia. https://www.investopedia.com/articles/forex/05/macddiverge.asp#toc-types-of-macd-strategies