Stock Trading Program Report

Ian Mays

# Abstract

This report explores the use of algorithms in stock market trading. A set of relatively simple algorithms, including hold-and-buy, 5-day moving average, 20-day moving average, and relative strength index, were tested on Apple, Amazon, Google, and Microsoft stock data spanning a year. It was found that the hold-and-buy technique showed the best performance by far.

Table of Contents

[Abstract 1](#_Toc153192435)

[Introduction 2](#_Toc153192436)

[Stocks 2](#_Toc153192437)

[Algorithms 2](#_Toc153192438)

[Results 3](#_Toc153192439)

[Holder Algorithm 3](#_Toc153192440)

[5-day Moving Average Algorithm 4](#_Toc153192441)

[20-day Moving Average Algorithm 4](#_Toc153192442)

[Relative Strength Index Algorithm 5](#_Toc153192443)

[Conclusion 5](#_Toc153192444)

# Introduction

With the proliferation of computers, an increasing amount of trading on the stock market is done using algorithms. Computers can quickly process large amounts of numerical information, and much research has been done into how to effectively harness that power to obtain the best results on the stock market. The most sophisticated algorithms often make use of machine learning techniques. However, this report will focus on a few extremely simple algorithms, comparing their performances on a handful of the most popular stocks.

# Stocks

|  |  |
| --- | --- |
|  |  |
|  |  |

The stocks examined in this report are the four largest companies by market capitalization in the United States as of 2023, those being Apple (AAPL), Amazon (AMZN), Google (GOOGL), and Microsoft (MSFT). The data for these stocks was obtained from Yahoo Finance, and covers the period from November 17, 2022 to November 17, 2023. All four of these stocks are technology companies, and all four have shown significant growth over the last year, with increases ranging from about 30% to about 50%.

# Algorithms

For each stock, four hypothetical traders were introduced, each starting with $10,000 to use for trading, and each one using a different algorithm to make decisions. On each trading day, each trader will decide whether to buy or sell shares on that day, and how many to trade.

The first algorithm tested, which effectively serves as a control group for the experiment, is referred to as the “Holder”. The Holder simply buys as many shares as possible on the first day and holds onto them for the entire year.

The second and third algorithms are both based on a moving average of the previous closing prices, one of them from the previous 5 trading days (approximately a week in real time) and the other from the previous 20 days (approximately a month in real time). If the current share price is lower than the moving average, the trader will buy an amount of stock worth 10% of its current balance, rounded to the nearest integer number of shares. If the current share price is higher, it will sell 10% of its currently owned stock, again rounded to the nearest integer number of shares.

The fourth and final algorithm is based on the Relative Strength Index, or RSI. RSI works by comparing the average magnitude of all day-to-day price increases within a given time period to the average magnitude of the day-to-day decreases. This algorithm uses a period of 14 days for this calculation, which is standard. The larger the increases are compared to the decreases, the higher the RSI. RSI is typically implicitly communicated as a percentage, i.e. a value between 0 and 100. In this case, this algorithm works by considering the value of its shares as a proportion of its total net worth. The RSI, as a percentage, determines how much of its net worth should be in the form of stock. Thus, the algorithm will typically buy more stocks when the RSI increases and sell when the RSI decreases.

# Results

In this section, net worth is represented using an area graph, where the blue region represents the account balance while the yellow region represents the total value of the trader’s owned shares.

## Holder Algorithm

|  |  |
| --- | --- |
|  |  |
|  |  |

As all four stocks have grown significantly during the time period examined, the simple strategy of buying and holding turned out to perform remarkably well. From an initial balance of $10,000 per stock, the trader ended up with a net worth of $12,934 when trading Apple stock, $14,736 for Amazon, $13,937 for Google, and $15,594 for Microsoft.

## 5-day Moving Average Algorithm

|  |  |
| --- | --- |
|  |  |
|  |  |

The 5-day moving average trader ended with $11,527 for Apple, $12,785 for Amazon, $12,852 for Google, and $12,347 for Microsoft.

## 20-day Moving Average Algorithm

|  |  |
| --- | --- |
|  |  |
|  |  |

The 20-day moving average trader ended with $11,792 for Apple, $12,944 for Amazon, $13,635 for Google, and $12,236 for Microsoft.

## Relative Strength Index Algorithm

|  |  |
| --- | --- |
|  |  |
|  |  |

The relative strength index trader ended with $11,720 for Apple, $12,067 for Amazon, $11,904 for Google, and $12,340 for Microsoft.

# Conclusion

In all cases, the primitive buy-and-hold strategy gave the best result by significant margins, including by over $3,000 in the case of Microsoft. The other algorithms all performed in roughly the same ballpark, earning a less significant but still positive profit. This does seem to show that just because an algorithm is more complex does not mean it is more effective at solving a given problem. In fact, the most complex algorithm used here, RSI, performed the worst overall, being at or near the bottom of the pack for all four stocks.

However, this is not to say that buying and holding is definitely the best method. For one, only a limited number of stocks over a limited time period were examined, and these stocks all have significant similarities with each other that may not apply in general. Also, it may be more effective to predict stock prices using other metrics which are not covered here, or different ways of using the same metrics. Moving average and relative strength are certainly useful to take into account, but the performance of algorithms that use them will depend on how exactly they are interpreted and acted on.