Comparative Analysis Between Technical Indicators and Reinforcement Learning Methods for Algorithmic Trading

A Thesis
Presented to
the Faculty of the College of Computer Studies
De La Salle University Manila

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science by

ABOY, William Dominique
LUIS, Mariel
PROMENTILLA, Jose Mikhael Uriel
YAP, Mike Jaren

Duke Danielle DELOS SANTOS
Adviser

August 28, 2019

Acknowledgements

We would like to express sincere gratitude to our panelists, Dr. Florante Salvador and Dr. Nelson Marcos as well as our thesis adviser, Duke Danielle Delos Santos who gave us the golden opportunity to pursue this paper on the topic "Comparative Analysis Between Technical Indicators and Reinforcement Learning Methods for Algorithmic Trading." It also helped us in doing a lot of research as we came to learn more about so many new insights through this endeavour.

Abstract

The purpose of this study is to compare and analyze the performances between Reinforcement Learning (RL) methods and technical indicators contextualized as trading algorithms. Technical indicators such as EMA, MACD, RSI and OBV were implemented to serve as the baseline for the study. Q-learning and Dyna-Q learning were the methods used for RL. Quantopian platform was utilized as the trading simulation environment where total returns and Sharpe ratios were measured. RL methods were tested in phases such that in each phase, the best parameter is identified and used in the succeeding test phases. As a result, Dyna-Q had the best outcome when trading daily for 15 years within an uptrend stock, significantly outperforming the technical indicators, with Sharpe ratio as its objective function. The best setup of hyperparameters was a learning rate of 0.1, a discount rate of 0.5, and an epsilon of 0.1. However, Dyna-Q performed inconsistently when it was tested on other stock types. Nevertheless, the results of this study show that RL methods have the potential to outperform technical indicators in terms of total returns and Sharpe ratio.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of the Current State of Technology

Stated by Investopedia (2018), every growing company requires a massive amount of capital. Mitchell (2018) states that in order to raise huge amount of capital to hire employees, buy equipments, raw materials and other necessities of the company, is through selling shares or what is commonly referred to as Stocks. Stocks as defined by Investor.gov (2018), is referred to as a type of security that gives stakeholders a share of the ownership of the company. Stocks can also be referred as "equities". These stocks represent a share of the company, or an ownership of the company that gives you the right to vote for choices that would affect the company's track in the future. Stocks are bought by investors. While the said company is benefited by the money given by the investor, the investor gets hold of a percentage of the said company and its assets.

The concept of the stock market alone is incidentally overwhelming at first. The reality that comes with the stock market is that it carries a lot of risk, especially considering the use of real money, where you have no assurance whether you would get back your money doubled or lessened. According to Investopedia (2018), the Stock Market itself carries a lot of risk, one where the potential of the whole market would decline, or the value of one stock would decline independently of the stock market as a whole. This shows even though how much investment or how much money was put into stocks, it could immediately decline and at the same time with your investment as well. However, despite these risks that accompany investing in the stock market, many investors still choose to be a part of this endeavor. As a benefit, the investors gain huge capital when an increase in its price occur. Dividends also occur especially whenever the company would

distribute some of its earning to its stockholders, and lastly it also provides the stockholders the ability to vote shares and to influence the said company.

According to Hur (2016), the stock market we know today is not the same stock market that it used to be before. With the rise of the renowned New York Stock Exchange, although the first genuine stock was dated back in the early 1500s, the concept of 'the stock market' was not new to the public. The concept was evident at the time of the Romans, starting from the second B.C. as stated by Smith (2003). Early signs of trading, as done by Romans,were whenever they are by the Forum near the Temple of Castor. Smith (2003) also states that the classical historian, Mikhail Rostovtzeff, described the place as "crowds of men bought and sold shares and bonds of tax framing companies, various goods for cash and on credit, farms and estate in Italy and the provinces, house and shops in Rome and elsewhere, ships and storehouses, salves and cattle." Back in the day, "traders at the New York Stock Exchange yelled out orders to each other, creating a raucous din. When a stock traded on the strength of a news story, traders gathered in the stock's trading area and started shouting matches that sounded like brawls," as described by Johnston (2011), whereas in comparison to today's modern era, trading still goes on, but without the shouting, and offer investors a more efficient way to high-tech trading to research, the purchase of stocks and high paced trading through the help of technology.

Biais, Foucault, and Moinas (2014) believe fast trading technology entails high-speed market connections which enables the ability to search for attractive quotes and market information within seconds. Consequently, prices are now more accessible to different communities that time calculations between price updates have significantly decreased to fractions of a second (Cumming, 2015). Because of such change, many traders cannot cope with the fast-paced fluctuations of trades which then overwhelm their decision-makings and sentiments towards their process. "Investors must process very large amounts of information, in particular about trades and quotes, which are relevant both for the valuation of securities and the identification of trading opportunities," Biais et al. argues. Varon and Soroka (2016) explains that many economists adhere to the Efficient-Market Hypothesis, advocating that it is impossible to "beat the market" on a risk-adjusted basis – it cannot be helped that the stock market follows a random walk.

In 1988, Lo and MacKinlay claim to have stumbled upon results which uncovered empirical evidence, suggesting that stock returns contain predictable components. Further proof that debunks the random walk theory in the stock market was simply the use of specification test based on variance estimator. Fama and French (1988) concurs to the mounting evidence of predictable stock returns to both long and short return horizons. "The estimates for industry portfolios suggest that predictable variation due to mean reversion is about 35 percent of 3

to 5 year return variances. Returns are more predictable for portfolios of small firms. Predictable variation is estimated to be about 40 percent of 3 to 5 year return variances for small-firm portfolios. The percentage falls to around 25 percent for portfolios of large firms," they stated. Nevertheless, traders continuously seek consistent success in their managed funds, which provides motivation for a profitable strategy in algorithmic trading.

Algorithmic trading is the process of utilizing computers that are programmed for placing trades in order to generate profits at a very high speed. In other words, it is the translation of trading techniques into code, known as trading algorithms, that are used for trading in the market. Ratnaparkhi (2017) supports such idea as "84% of trades that happened in NYSE, 60% in LSE and 40% in NSE were done using algorithmic trading." It would allow traders to conveniently execute tens and thousands of trades per second. According to Boehmer, Fong, and Wu (2015), algorithmic trading has experienced an exponential growth over the past years despite not being a recent phenomenon due to its intensity, volume and especially speed. Chaboud, Chiquoine, Hjalmarssonh, and Vega (2014) highlight in their literature the two main differences between computer and human traders. First, computers are much faster than humans, both in processing and acting on information. Second is the potential for higher correlation in computers' trading actions than in those of humans, since computers need to be pre-programmed. Due to the growing interest, an increasing number of asset managers are already using computers as means to buy and sell stocks/shares automatically. The decisions made by these machines are based on algorithms, derived from statistical model (Chu, 2018). In addition, since this is based from historical records, algorithmic trading effectively eliminates the emotional factors which, for human investors, has an effect to the decision-making process. Yet Chu argues as well that when failures occur, algorithm is to blame especially if it does not meet the human common sense to share valuation. Some sensitive traders are skeptical to its relatively unusual behaviours that they deem unacceptable.

While algorithmic trading is generally allowed and used by many modern investors, there is no denying that it lacks the extensive flexibility and adaptability in dealing with rare occurrence of price fluctuations. Likewise, Chaboud et al. (2014) inform the concern of traders towards higher adverse selection and excess volatility.

Reinforcement Learning on the other hand, a branch of Machine Learning, enables "an agent to learn how to behave in a stochastic and possibly unknown environment, where the only feedback consists of a scalar reward signal," Necchi (2012) states. But since the environment (i.e. the stock market) evolves in a stochastic manner, there is a need for the agent to balance immediate rewards with that of future opportunities.

Many hypothesize that (RL) in the context of algorithmic trading could have the potential to rival, if not outperform, the trading algorithm strategies. An example would be the work of Du, Zhai, and Lv in 2009 where a comparative analysis between two RL techniques was conducted namely, Q-Learning and Recurrent Reinforcement Learning (RRL). Their findings suggest that RRL with policy iteration performs better than Q-Learning due to its flexibility in choosing objective functions to optimize using stochastic batch gradient ascent. While there have been attempts to do research on this particular field, Varon and Soroka (2016) are still convinced that there are only limited available resources of the said field despite the growing attention to the technology. Nevertheless, Varon and Soroka's findings suggest there is work needed to be done because in the current state of research, RL still trails behind existing trading algorithms. However, it does not necessarily mean algorithmic trading always outperforms RL.

1.2 Research Objectives

1.2.1 General Objective

To compare and analyze the performance of RL methods and Technical Indicators. The stocks will be selected according to their observable trends.

1.2.2 Specific Objectives

1. To identify the appropriate dataset/s that includes stocks, time frame and more, and tools for trading simulations and backtesting.
2. To determine a baseline algorithm or strategy that can be used for comparison and analysis.
3. To implement RL methods as trading algorithms.
4. To compare and analyze the performance of RL methods with the chosen technical indicator as the baseline.

1.3 Scope and Limitations of the Research

With the use of the Quantopian platform as also the source of this study's datasets, the dataset were limited to what Quantopian is able to offer. The dataset used are historical data contextualized in the United States, containing the necessary stock market values (i.e. price, volume, etc.). Quantopian is a web-based tool used to produce performance results of the RL methods and baseline algorithms/strategies due to the tool's provision of free backtesting and historical data. Stock indices like NASDAQ, NYSE, S&P500 and the like were unavailable to be utilized for trading. The stocks used for testing were Acacia Research Corporation (ACTG), Walmart Inc.(WMT), Cleveland-Cliffs Inc.(CLF), Altaba Inc (AABA), Vertex Pharmaceutical Inc. (VRTX), BRF S.A.(BRFS), China Southern Airlines Co Ltd (ZNH), Illumina, Inc.(ILMN), and China Eastern Airlines Corporation Ltd.(CEA). These stocks were picked because of its different movement, its duration. Additionally, these stocks have complete datasets from Quantopian directory.

In this research, the baseline algorithms/strategies used are technical indicators namely, Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI) and On Balance Volume (OBV).

RL methods was implemented such that its compatible with dataset and backtesting platform, Quantopian. The Python programming language was used because it was compatible with the Quantopian platform. The RL algorithms implemented are Q-learning and Dyna-Q learning. Since model-based learning algorithms may require neural networks and stochastic modeling techniques, it was out of the scope of this study. Dyna-Q learning was instead partly represent model-based learning methods. Q-learning and Dyna-Q learning represented the three kinds of RL methods: model-free learning, model-based learning, and dyna learning.

Furthermore, the measurements were based on profit gains and profit losses of the RL methods, disregarding the additional transaction costs. The study mainly focused on the total returns, common returns and Sharpe ratio as performance basis. Such metrics were calculated by Quantopian upon completion of every backtests. Other performance factors provided by the platform may also be included in the analysis.

1.4 Significance of the Research

Further applications of Reinforcement Learning may potentially be unlocked or discovered specifically in this research, the possible use of RL in the context of algorithmic trading. It will aid analyst in comprehending the fast-paced scenarios and behaviours of the stock market. Furthermore, this research may provide successful strategies and techniques that would yield profitable outputs especially for traders who wish to compete in the market.

Chapter 2

Review of Related Literature

The concept of this research mainly revolves around the domain of stock trading, in which comparative analysis between reinforcement learning contextualized to such environment and a trading algorithm shall be conducted. This chapter discusses papers, software, tools and technology related to stock markets and algorithmic trading, which are integral to this particular field of study.

2.1 Stock Trading

According to Tillier (2017), it is incorrect to imply that the phrase "the stock market" points to only one such market where there is, in fact, many. The stock market refers to the collection of markets and exchanges where the issuing and trading of stocks to and from various classes of securities take place. It is where a trade can be made through either formal exchanges or over-the-counter (OTC) marketplaces.

2.1.1 Stocks

Buying a stock/shRare means buying a piece of the company (Mitchell, 2017). Many traders have commonly defined stocks as representations of ownership towards a certain company's assets and earnings. However, Hayes (2017) defies the misconception as stockholders do not technically own the corporation, but own the shares issued by the corporation.

To "trade" in the jargon of the financial markets means to buy and sell stocks (Little, 2018). These two main transaction methods are what stimulates the very purpose of stock exchanges. Mitchell (2017) agrees that buying stocks is as important as selling stocks because in order to increase its worth, there must be a considerable volume of traders who are in agreement with a particular market. Likewise, Cumming (2015) states that if someone buys an stock, they do so with the desire for it to increase in value so that they can later sell it for a profit. In the perspective of selling stocks, Glassman (2013) explains that price is the reason for traders to sell their assets – either the price has gone up so much that it's time to take profits, or the price has gone down so much that it's time to cut losses.

In general, stock prices are commonly believed to react sensitively to economic activities which are influenced by a wide variety of unanticipated events, bearing diversified risks (Chen, Roll, & Ross, 1986).

2.1.2 Stock Table/Quotes

Table 2.1: Apple Inc. (AAPL) Stock Table in 2016, Yahoo! Finance.

Open	**107.67**	Market Cap	**580.5B**
Prev Close	**106.73**	P/E Ratio (ttm)	**12.56**
Bid	**107.10 x 100**	Beta	**12.56**
Ask	**107.75 x 100**	Volume	**25,551,265**
Day's Range	**106.82 - 108.00**	Avg Vol (3m)	**32,180,656**
52wk Range	**89.47 - 123.82**	Dividend & Yield	**2.82(2.15%)**
1y Target Est	**124.11**	Earnings Date	**10/25/2016 - 10/31/2016**

Open Price is the price at which a stock first trades upon the opening of an exchange on a given trading day. **Previous Close**, on the other hand, refers to the last price of a stock on a trading day. According to Davila (2016), if the Opening price is significantly different from the Previous Close Price, it may indicate that traders have reacted to either a positive or negative information regarding the current status of the company that issues a certain stock.

The **Bid** is the buyable stock price in the market while the **Ask** is the acceptable price for a seller. The measure (also referred to as "spread") of the demand and supply for a stock is the determined by the difference between the bid and ask.

The **52-Week Range** indicates the highest and lowest price at which a stock has been traded over the previous 52 weeks/1 year (Hayes, 2017). Same concept applies to the **Day's Range** except it only monitors the price movement within

the day. **One-Year Target Estimate** is simply the prediction of stock price over a year.

Market Capitalization, commonly referred to as "Market Cap," is the market's price value on a company's share (Davila, 2016). Its value constantly changes and often used to measure the ranks of publicly-traded companies.

P/E Ratio provides a guide as to how much a trader can expect to invest in a company. Its value can derived by the following formula:

$$P/ERatio = \frac{current\_price\_of\_stock}{earnings\_per\_share} \tag{2.1}$$

The **Beta** refers to the measure of volatility (the rapid and unpredictable change) of a stock price compared to that of the market, which has a beta of 1.0. Davila (2016) explains that stocks having a beta less than the benchmark would move less than the market. Likewise, a beta more than the benchmark would move faster.

Volume shows the total number of shares made per day whereas **Average Volume** tracks the average number of trades per day within the specified time period (Hayes, 2017).

The **Dividend** represents the annual payment per share while the **Yield** indicates the return on the dividend percentage derived from the following formula:

$$Dividend = \frac{annual\_dividends\_per\_share}{price\_per\_share} \tag{2.2}$$

It is important to note that if a stock table provides no dividend and yield figures, the company does not currently pay out dividends.

Lastly, **Earning Date** is when one can expect to release an official public statement of a company's profitability for a specific time period.

2.2 Algorithmic Trading

Algorithmic trading (automated trading, black-box trading or simply algo-trading) can be defined as the process of placing trade and getting profit at a high speed and frequency by using computers that were given a set of rules to follow. According to Seth (2014), these set of instructions are based on timing, price, amount, and more mathematical model that the human-being alone cannot handle. Moreover,

these systems wants to find the expected market prices, profits from statistical patterns of financial markets, optimal execution of orders, disguise and detect strategies of traders (H. Li, n.d.).

H. Li stated that algorithmic trading can be used at any stage of trading process and other purposes such as market making, spread trading, arbitrage, and macro trading. In trade-execution programs, the algorithm decide features like timing, price, and order's quantity splits. While on other systems, the whole trading process is completely automated.

There is an "Algorithmic Trading System Components" that describe that the trading process can be divided into four steps: pre-trade analysis, trading signal generation, trade execution, and post-trade analysis (H. Li, n.d.).

In pre-trade analysis, the system compares and contrasts the performances that were taken before of several index-tracking strategies to help in the selection of strategy that best fits the present market conditions. Pre-trade analysis includes the alpha model which predicts the future behavior of the financial instruments to trade, the risk model which evaluates the stage of exposure/risk associated with the financial instruments, and the transaction cost model which calculates the (potential) costs associated with trading the financial instruments (H. Li, n.d.).

Trading signal generation includes the portfolio construction model. It takes the results of the alpha, risk, and transaction models as inputs and it then decides what portfolio of financial instruments should be owned and how many. Even though trading signal generation overlaps with pre-trade analysis, there is a difference between them. Trading signal generation considers an actual trading signal generated by an algorithm with a specific price and possibly a quantity, and risk management recommendations (H. Li, n.d.).

Trade execution includes the execution model. It performs the execution of trades that makes several decisions with restrictions on transaction costs and trading duration. The system inspects factors such as order size, trading mechanism,and degree of trader's anonymity. It also determines whether to execute the trade immediately by submitting market orders or wait to get better price by submitting limit orders (H. Li, n.d.).

Since the trading platforms are accessed online, traders are naturally dependent on the network connectivity. Any existing trading algorithm would should be at least be able to read and monitor current market prices. It should also have the capability to buy and sell especially when opportunities to place orders occur. On the other hand, Seth (2014) explains that "the more complex an algorithm, the more strict backtesting is needed before it is put into action." There additional

risks and challenges to consider when utilizing trading algorithms in the market such as failure risks, network connectivity errors, time-lags between trade orders and execution etc.

2.2.1 Advantages

The advantages of algorithmic trading include trades that were made are at their best possible prices and at the right time. It can be backtested on available data, historical or real-time, to know its trading strategy. Seth (2014) explains that it also reduces transaction cost, lessens mistakes and time in order placement (having higher chances of execution at desired levels), and avoids significant price changes. Further, algorithmic trading diminishes the risk of manual errors and possibilities of mistake by human traders based on their emotions and psychological factors.

In 2003, Hryshko and Downs attempted to "create a system with a Genetic Algorithm (GA) engine to emulate trader behaviour on the Foreign Exchange Market and to find the most profitable trading strategy." Their findings suggest that GA is capable of understanding trading rules and use them to optimize trading performance. Kissell and Malamut (2005) believes that it is essential for traders to compare and analyze alternative algorithms to determine the most suited for them when developing their own customized algorithms.

2.2.2 Disadvantages

The disadvantages of algorithmic trading include taking out human intellect and gut feeling during the trades. Considering that computers lack the power of decision-making, it leads to a problem when it comes to a major macro-market change such as subprime crisis or the flash crash. "Macguire quotes Richard Brown, who says, 'The subprime crisis was a unique situation in which there was no way for the news algorithms to necessarily be able to know what to do, which is why human intervention was necessary'"(Rao, 2015). On May 2010, the market was unstable causing algorithms to withdraw hence the flash crash, an abrupt decline in the market and a recovery for a secured market(Navone & Putnins, 2016). Different firms use different algorithms that counter each other. When this occur, some firms changes their algorithm bringing them to lose and this will prompt to a huge shift downward. A ripple effect, as stated by Rao (2015), takes place because currently the algorithms are not programmed for that situation resulting to decline of the market.

2.3 Technical Indicators

2.3.1 Exponential Moving Averages (EMA)

There are two most popular types of moving averages which are the Simple Moving Average (SMA) and the Exponential Moving Average (EMA). Moving averages polishes the price action and filters out the noise of the data. It tells the current direction of the price although it is delayed because it is based on past data prices. SMA and EMA can be used to determine direction of trend or define potential support and resistance levels.

SMA is created by adding recent closing prices and then dividing it by the number of time period. It is called moving average, therefore old data will be replaced by new data making the average to move. EMA is more weighted on recent prices while SMA is equally weighted to all values.

2.3.2 Moving Average Convergence Divergence (MACD)

Moving Average Convergence/Divergence oscillator (MACD), developed by Gerald Appel, is both trend following and momentum. It turns moving averages into a momentum oscillator by subtracting the longer moving average from the shorter moving average. It fluctuates above and below the zero line while the moving averages converge, cross and diverge.

MACD main point is about the convergence and divergence of the two moving averages. When the moving averages move towards each other, a convergence happens. Meanwhile, when the moving averages move apart, a divergence happens. The shorter moving average (lower number of days) is faster and accountable for most MACD movements and the longer moving average (higher number of days) is slower and less reactive. The MACD line oscillates above and below the zero line. Positive MACD occurs whenever the shorter moving average is above the longer moving average. It creates an increasing upside momentum. On the other hand, negative MACD occurs whenever the shorter moving average is below the longer moving average. It creates an increasing downside momentum.

MACD has its drawbacks of simply being a moving average which are lagging indicators. It also calculates the absolute difference between two moving averages and not the percentage difference. Since MACD is calculated by subtracting one moving average from the other. As price increases, the difference between the two moving averages is destined to grow. Thus making it difficult to compare

MACD levels over a long period of time, especially for stocks that have grown exponentially. Another drawback is that it does not have good identification of overbought and oversold levels. It does not have any upper or lower limits to bind its movement and may continue to overextend beyond historical extremes.

2.3.3 Relative Strength Index (RSI)

Relative Strength Index (RSI), developed by J. Welles Wilder, is a momentum oscillator that measures the speed and price change movement. RSI moves between zero and 100. Wilder claims that whenever RSI is above 70, it is considered as overbought and oversold whenever it is below 30. It has a benchmark that indicates if it is overbought or oversold.

A disadvantage of RSI indicator can remain oversold or overbought for an extended period of time in a trending market and giving out many false buy or sell signals. In addition, price bottoms and price tops can occur long after oversold and overbought zones reached. To further explain, there is no guarantee that oversold or overbought would immediately start to change its direction.

2.3.4 On Balance Volume (OBV)

On Balance Volume (OBV), developed by Joe Granville, has a cumulative indicator that measures buying and selling pressure. It adds the volume on up days and subtracts on down days. It is one of the first indicators to measure positive and negative volume flow. Divergences between OBV and price are used to predict price movements or are used to confirm price trends. It tracks number of buyers and sellers on a certain stock and then bases its decision on that.

OBV has its limitations due to its forecasting ability of reversals. It sends out a large number of false signals that coincides with the valid ones. Another limitation is when a massive volume spike occurs in a day, this will affect succeeding indicators.

2.4 Reinforcement Learning

Reinforcement Learning (RL) are algorithms that learn to take the best action (R. S. Sutton & Barto, 2018). The objective of RL is to maximize the rewards

it receives over its lifetime. There are mainly four components in RL: agent, action, environment, reward (Arulkumaran, Deisenroth, Brundage, & Bharath, 2017). The **agent** is the component taking actions in the environment. The **actions** that the agent take will affect the environment and the rewards. The whole **environment** can be broken down into states. **Rewards** are numerical signals that enable the agent to learn the best actions. The RL cycle starts with

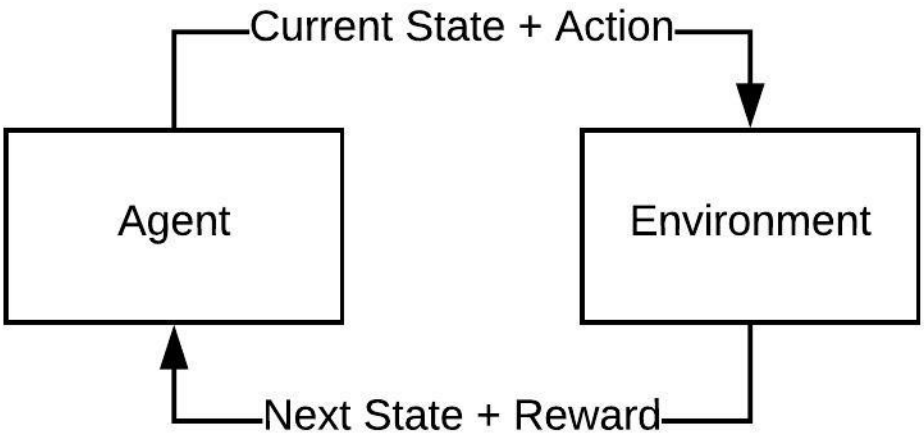


Figure 2.1: RL General Cycle

the agent in the initial state. The agent takes an action in the environment, represented by its current state. The environment then returns a reward based on the action. The agent is then transitioned onto the next state. This is further formalized into a Markov Decision Process (R. S. Sutton & Barto, 2018).

2.4.1 Markov Decision Process

A **Markov Decision Process (MDP)** is composed of four elements: states, actions, rewards, and transition probabilities (Kaelbling, Littman, & Moore, 1996). S is a set of states. **States** represent the current environment of the environment. States can be described in time steps such as S_{t+1} or a transition to a next states denoted as s'. A is a set of actions. $R(s, a)$ is the reward function with the parameters of state and action. $T(s, a, s')$ are the transition probabilities with the parameters of state, action, and next state s'. The agent learns the optimal policy through rewards (Arulkumaran et al., 2017). An **optimal policy**

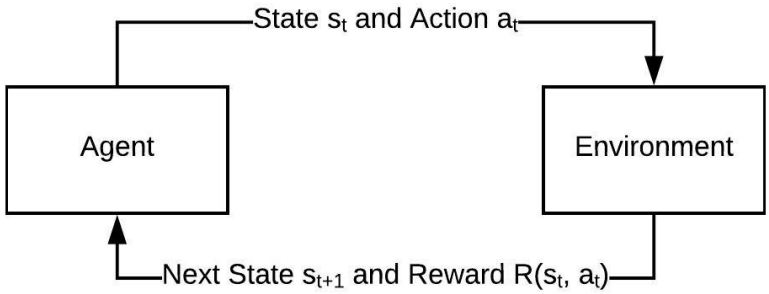


Figure 2.2: Markov Decision Process

is a set of actions that maximize the total reward over time. The agent learns the optimal policy through trial and error over an episode. An **episode** is the lifespan of an agent. An episode ends when the agent reaches a terminal state or when a parameter specifies its end (R. S. Sutton & Barto, 2018). States and actions are feedback that the RL algorithms use to estimate and derive the optimal policy (Arulkumaran et al., 2017). States and actions can be evaluated as value functions.

2.4.2 Optimal Value Functions

If an RL algorithm has a set of optimal value functions for all states and all actions, then deriving the optimal policy would be possible by acting greedily. By always choosing the value functions with the highest estimated reward, the agent will be acting optimally because it is maximizing its total reward (Watkins & Dayan, 1992). When a set of value functions lead to an optimal policy, those value functions are known as **optimal value functions** (R. S. Sutton & Barto, 2018). One of the RL problems then is to learn the optimal value functions. R. S. Sutton and Barto (2018) defines optimal value functions are defined as

$$v_*(s) = \max_\pi v_\pi(s) \tag{2.3}$$

for all $s \in S$ and

$$q_*(s) = \max_\pi q_\pi(s, a) \tag{2.4}$$

for all $s \in S$ and $a \in A$.

2.4.3 Bellman Equation

Estimations of the value functions are improved with each experience of states, actions, and rewards. R. S. Sutton and Barto (2018) defines a Bellman Equation in terms of value functions as

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \Big[r + \gamma v_\pi(s') \Big], \tag{2.5}$$

for all $s \in S$. With each new experience, the value functions will be updated closer to the optimal value functions. The value functions will eventually converge to the optimal value functions due to the bellman equation. The bellman equation assures convergence due to the contraction mapping theorem as proven by Bellman (1954).

In RL, there are different methods of learning the optimal policy but mainly two different types: model-based learning and model-free learning.

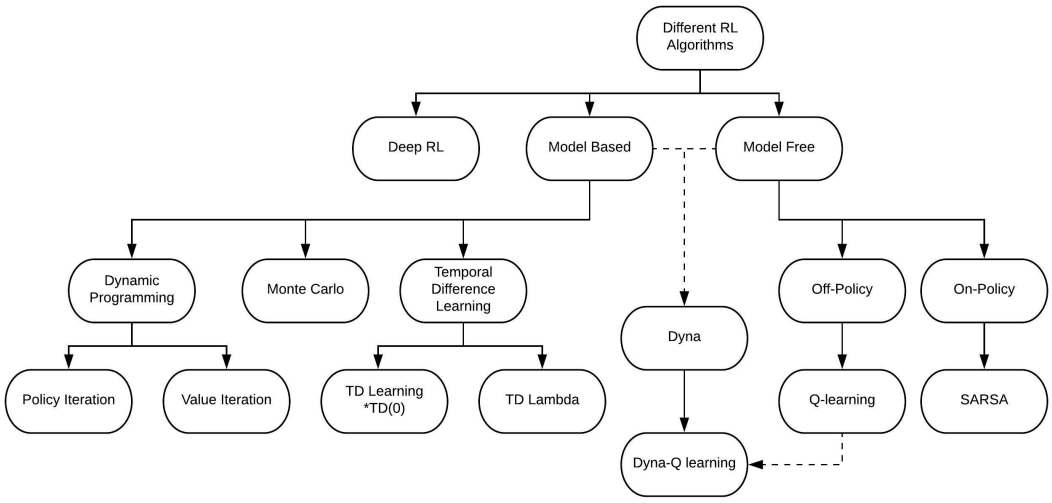


Figure 2.3: RL Algo Tree

2.4.4 Model-Based Learning

Model-based learning learns the optimal policy by learning the model of the environment (Gosavi, 2009). A **model** is composed of states, actions, rewards, and transition probabilities. **Transition probabilities** determines the next state the

agent will land on (R. S. Sutton & Barto, 2018). If the agent is given the transition probabilities of the environment, the agent will be capable of computing all the value functions through dynamic programming (Y. Li, 2017). One of the model-based learning methods is value iteration.

2.4.5 Value Iteration

Value iteration updates its value functions by doing a one-step look-ahead (Kaelbling et al., 1996). A one-step look-ahead simulates all possible actions and computes its corresponding reward. The action with the highest estimated value in the next one-step will be the chosen state-value for the current state the agent is in. The algorithm continually updates its value functions with this one-step look-ahead until the values converge. Kaelbling et al. (1996) constructed a value iteration algorithm as

initialize $V(s)$ arbitrarily
repeat
  for $s \in S$ **do**
    for $a \in A$ **do**
      $Q(s,a) := R(s,a) + \gamma \sum_{s' \in s} T(s,a,s')V(s')$
    end
    $V(s) := \max_a Q(s,a)$
  end
until *policy is good enough*;

Algorithm 1: Value Iteration Algorithm

2.4.6 Model-Free Learning

Model-free learning learns the optimal policy by directly learning the value functions from its interactions with the environment (Gosavi, 2009). Each experience is considered as a sample. The advantage of model-free learning is that it does not need a model of the environment in order to learn the optimal policy. One of the model-free learning methods is Temporal Difference (TD) learning.

2.4.7 TD Learning

TD learning learns from the errors between its predictions and actual outcome (R. Sutton, 1988). TD learning updates its value functions by bootstrapping.

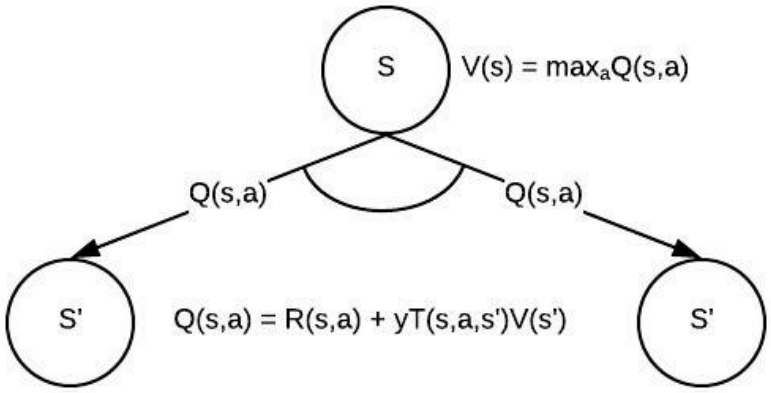


Figure 2.4: Value Iteration

Bootstrapping is estimating the future rewards(Y. Li, 2017). TD learning combines the immediate reward received and the estimates of the future reward. The agent will continue to interact with the environment and receive experiences. Each experience is a sample and is used to update all the values and its estimations. With more samples and updates, the algorithm eventually converges to the optimal value function. R. S. Sutton and Barto (2018) defines the TD learning algorithm as

Input: the policy π to be evaluated
Data: Algorithm Parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in S^+$, arbitrarily except that $V(terminal) = 0$
for *each episode:* **do**
  Initialize S
  for *each step of episode* **do**
    $A \leftarrow$ action give by π for S
    Take action A, ovbserve R, S'
    $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$
    $S \leftarrow S'$
  end until S is terminal
end

Algorithm 2: TD Learning Algorithm

2.5 Reinforcement Learning in Trading

RL has been used for different applications including portfolio management and stock trading.

Moody and Saffell (2001) implements Recurrent Reinforcement Learning (RRL) for discovering investment policies. The RRL optimized for the Sharpe ratio and the differential downside deviation ratio. RRL is unlike Q-learning and TD-learning which estimates value functions. RRL is better because it avoids Bellman's curse of dimensionality, represents the problem simpler, and is more efficient. The results are that RRL strategies are stable and maintains its positions for a long period of time. In contrast, Q-learning switches its positions frequently indicating that it is sensitive to the noise. Lastly, Moody and Saffell (2001)observes that RRL has better trading strategies than Q-learning.

Lee, Park, O, Lee, and Hong (2007) implemented MQ-Trader, a system that suggests buy and sells to traders. It implements Q-learning algorithms in a multiagent framework. The first two agents named as buy and sell agents attempts to buy and sell at the right time. The other two agents determines the best buy and sell price. The purpose for this multiagent approach is to have the Q-learning agents to divide and conquer the problem. It also attempts to model a human trader's behavior.

Pendharkar and Cusatis (2018) implemented on-policy SARSA(λ) and off-policy Q(λ) as trading agents. High-learning frequency agents to outperform single asset stock and bond cumulative returns significantly. But interestingly, they also noted that annual trading frequency to be better than semi-annual and quarterly trading. With a higher frequency in trading would generate more data to learn from. But it was observed that the generation of more data leads to better performance, and in fact performs worse. This goes against the common belief that more data would lead to higher performance. The reason for this lies in the generation of the data. As more data is generated, the RL agents start to believe that stocks will be outperformed by bonds. The reason for this is because of the higher volatility of stocks. The variance of the data is affecting the performance of the agent. This means that more dataset isn't necessarily good and that the dataset might need to be engineered to learn the right actions. Pendharkar and Cusatis (2018) observes that there is a tradeoff between the size of the training set and the learning algorithm. It was also observed that the performance is easily swayed by the selected parameters. In their case, the parameters were optimized for annual trading. In the end, the RL agent beats the S&P 500, AGG, and 10-year T-note portfolio by a large margin.

Jeong and Kim (2019) uses transfer learning for the better performance of the RL agents. The reason is that because the data is highly volatile, they will better train the agents by transfer learning. During uncertain situations, the hold strategy was the most effective in not making mistakes. In uncertain situations, the delay in decision making such as the hold strategy was the most profitable. This hold action stabilizes the network during uncertainty.

Almahdi and Yang (2017) implemented different objective functions; Sharpe ratio, Calmar ratio, and Sterling ratio using RRL. The Sterling ratio performed badly with RRL due to its lacking statistical properties and the Calmar ratio outperforms the Sharpe ratio consistently. Objective functions can be modified for better performance. But the compatibility of the RL algorithms to the objective functions should also be considered. In the end, the RL agent outperforms the benchmark and the hedge fund industry average index (Lee et al., 2007).

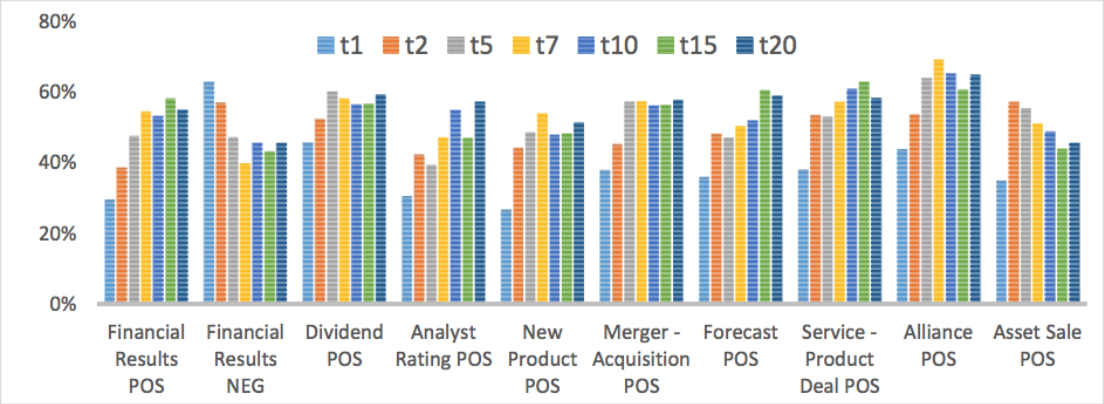
2.6 Quantopian

There has surprisingly been limited available resources about reinforcement learning despite the growing interests. In an attempt to find profitable strategies through technical analysis and mean reversion, Varon and Soroka (2016) applied Q-learning to stock trading with their primal tool for backtesting being a web application called Quantopian. It is a python-based development platform for algorithmic investments which provides historical market data and automated testing execution – all free of cost. It also provides capital, education, data, a research environment, and a development platform to algorithm authors. Quantopian hosts a community where members can probe questions and collaborate ideas and code as well as share data. Members can learn from each other regarding new techniques and strategies that they can use to improve their own algorithms.

2.6.1 Backtesting

On the other hand, one of the experiments conducted by Ben-Ami and Feldman (2017) suggests that Quantopian can be used to evaluate event polarity as indicators for future positive and negative returns as seen in Figure 2.5. The results show that events have varying effects over different time periods. Moreover in 2016, a study on Fundamental Signals for Algorithmic Trading was conducted by Wong et al. (2016) has proved Quantopian to be helpful because it was their primary source for backtesting on different dataset combinations which resulted to

Figure 2.5: Predicting stock returns using event polarity, by event, by time period.



formulating an algorithm that generates mildly accurate predictions on the future stock prices of companies.

Smolyakov (2017), a Data Scientist at Shopify, testifies to how Quantopian is exhibited to be excellent for researching and implementing trading strategies. Because of its number of in-house tools to support mathematical computations, the tool provides a convenient ipython notebook interface for research and a simple to use IDE for implementation and back-testing of strategies that automatically logs key performance metrics and compares your strategy against a benchmark. In 2015, Cumming claims that although Quantopian is useful as an IDE for writing codes in python and run-test on historical data, it was simply not suitable for his work due to the lack of support for foreign exchange trading, which was his research focus at the time.

Chapter 3

Theoretical Framework

3.1 Technical Indicators

3.1.1 Exponential Moving Averages (EMA)

$$SMA = \frac{\sum_{i=1}^{n} price_i}{n} \tag{3.1}$$

EMA requires a Simple Moving Average (SMA) as the initial calculation of past prices. The formula for SMA, as shown in Equation 3.1, is the average of prices in n-day period, where n is the number of days of past prices.

$$mult = \frac{2}{n+1} \tag{3.2}$$

A weighting multiplier must also be calculated as the algorithm applies weights to recent prices. The formula is simply 2 divided $n + 1$.

$$EMA = (close - EMA(n-1)) * mult + EMA(n-1) \tag{3.3}$$

Finally, Equation 3.3 shows how the the EMA values are computed by subtracting the EMA of the previous day, denoted as *n+1*, from the close price denoted as close, multiplied by the weighting multiplier and adding it to the EMA of the previous day.

This indicator requires two EMA values, particularly a *long_ema* and a *short_ema*, in order to identify the trend movement of a certain stock. If the *short_ema* crosses

above the *long_ema*, it indicates an uptrend shift, which then implies a buying opportunity. Consequently, if the *short_ema* crosses below the *long_ema*, it indicates that the stock experiences a downtrend shift, signalling a selling opportunity.

3.1.2 Moving Average Convergence Divergence (MACD)

$$MACD = EMA(s, 12) - EMA(s, 26) \tag{3.4}$$

The MACD line is calculated by subtracting the 26-day EMA from the 12-day EMA of a certain stock denoted as s. This serves as the momentum line indicator of the stock prices.

$$SIGNAL = EMA(MACD, 9) \tag{3.5}$$

A 9-day EMA of the MACD line is also calculated which acts as a signal line that identifies turns in the trend. The number of days are arbitrarily selected for the algorithm.

Like Exponential Moving Averages, the algorithm focuses on the crossovers between two lines – the MACD line and the Signal line. Specifically, if the MACD line crosses above the Signal line, a bullish trend (uptrend) is being observed which could indicate a buying opportunity. However, if the MACD line crosses below the Signal line, then the trend is bearish (downtrend) and a selling opportunity.

3.1.3 Relative Strength Index (RSI)

$$RSI = 100 - \frac{100}{1 + RS} \tag{3.6}$$

As seen in Equation 3.6, RSI is computed by getting the quotient of 100 and *1 + RS* and subtracting it from 100.

$$RS = \frac{AverageGain}{AverageLoss} \tag{3.7}$$

The relative strength, denoted as *RS*, has the formula of dividing the *Average Loss* from the *Average Gain*.

$$AverageGain = \frac{\sum_{i=1}^{14} gain_i}{14} \tag{3.8}$$

$$AverageLoss = \frac{\sum_{i=1}^{14} loss_i}{14} \tag{3.9}$$

Equations 3.8 & 3.9 have similar equations which is the sum of their respective values in a 14-day period divided by 14.

RSI requires two thresholds for indicating overbought and oversold stocks. The value from the previous computation will be compared to the given thresholds in order to determine opportunities for buying and selling. If the RSI value is greater than the overbought mark, then it must be an indication to sell. And if it is less than the oversold mark, it then implies a buying action.

3.1.4 On Balance Volume (OBV)

$$obv_n \leftarrow obv_n + volume_n \tag{3.10}$$

OBV keeps track of the volume as an assessment for buying and selling pressures in the stock market. The algorithm states that given the current and previous price, if the latter is higher than the former, then the current OBV value is updated according to the formula provided in Equation 3.10.

$$obv_n \leftarrow obv_n - volume_n \tag{3.11}$$

On the other hand, if the current price falls short compared to the previous price, then the current OBV value adjusts based on the formula of Equation 3.11.

$$obv_n <= price_n \tag{3.12}$$

$$obv_n >= price_n \tag{3.13}$$

The OBV indicator measures the competition between the volume and the prices. If the current price exceeds the current OBV value, it signals a potential weak buying pressure and that the trend may be starting to shift (Equation 3.12). At such point, it is an indication of a selling opportunity. A buying opportunity is then identified if the current OBV value is higher than the current price (Equation 3.13). This means that the buying pressure is starting to gain momentum while prices are still affordable.

3.2 Value Functions

Value functions represents how rewarding it is for an agent to be in a certain state or to take a certain action (Y. Li, 2017). Value functions can be evaluate two things: states and actions. **State-value functions** represents how much rewarding it is to be in a certain state. **Action-value functions** represent how rewarding it is to take a certain action in a certain state. Values functions serves as an estimation of future rewards. R. S. Sutton and Barto (2018) defines a state-value function following policy π as

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right] \tag{3.14}$$

for all $s \in S$ and defines an action-value function following policy π as

$$q_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right] \tag{3.15}$$

3.3 Rewards

Rewards are an important feature in RL as it serves as the signal for learning. Rewards are the objective functions the RL is trying to maximize for. **Immediate rewards** are rewards received after taking an action (Gosavi, 2009). While **delayed rewards** are estimated rewards that come from the future states (Kaelbling et al., 1996). RL algorithms must learn from delayed rewards by estimating or incorporating them correctly. Delayed rewards can be incorporated into the algorithm as discounted rewards (Gosavi, 2009). Rewards in the future are valued less compared to immediate rewards. Because of that, rewards in the future are computed by multiplying the future rewards by a discount factor denoted as γ. A discount rate of 0 will make the agent myopic and will only act to maximize the next immediate reward without concern for the succeeding rewards after it. Meanwhile a discount rate of 1 will make the agent far-sighted and act as to maximize future rewards until the end. Aggregating a series of rewards into a single value is known as the expected return (R. S. Sutton & Barto, 2018). The objective of RL is to maximize the expected return. R. S. Sutton and Barto (2018) defines the expected return as

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T \tag{3.16}$$

3.4 Exploration-Exploitation

In TD learning the agent must take an action to sample from the environment. How the agent decides which of its actions to take is based on the ϵ-greedy policy (Arulkumaran et al., 2017). The **ϵ-greedy policy** addresses the issue of exploration-exploitation trade-off (Kaelbling et al., 1996). Given an information, the agent has a choice: exploit and get the highest known reward, or do a random action and explore possibly higher rewards. The ϵ-greedy policy has a parameter ϵ that determines whether or not to take an action. With probability ϵ, the agent will explore and take a random action. With probability $1 - \epsilon$, the agent will exploit and act greedily. An epsilon of 0.1 means that the agent will take random actions to explore other states 10% of the time. A higher value of epsilon will result into the agent exploring more, resulting to an adaptive agent. On the other hand, an adaptive agent is also less able to exploit their knowledge. Once the agent has explored enough, ϵ can be decreased as exploitation becomes a priority towards the end (Arulkumaran et al., 2017).

3.5 Q-learning

Q-learning is a model-free learning method that directly learns its action-values from experiences (Watkins, 1989). Action-values, also known as **Q-values**, represent how rewarding it is to take a certain action from a certain state. Once the Q-values are known, the agent will act greedily with respect to the Q-values to reach the optimal policy (Watkins & Dayan, 1992). The Q-learning value update by R. S. Sutton and Barto (2018) is defined as

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big] \tag{3.17}$$

Q-learning, and most RL algorithms are based on this general update equation. The term $R_{t+1}+\gamma \max_a Q(S_{t+1}, a)$ from equation 3.17 is known as the **TD target**. It represents the the more accurate version of the Q-values the algorithm updates towards to. Specifically, $\max_a Q(S_{t+1}, a)$ takes the maximum value as done in value iteration and bootstraps the future rewards as done in TD learning. The term $R_{t+1}+\gamma \max_a Q(S_{t+1}, a)-Q(S_t, A_t)$ is known as the **TD error**. It represents the difference between the agent's old prediction and the newer more accurate prediction. The α is a parameter of learning rate. By how much the old prediction is adjusted is determined by the learning rate. With a higher learning rate, the predictions will adjust sharply to the newer prediction. The learning rate should

be optimized such that it's small enough to converge to a value but big enough to learn quickly.

Initialize $Q(s, a)$, for all $s \in S^+$, $a \in A(s)$,
arbitrarily except that $Q(terminal,) = 0$
foreach *episode* **do**
  (a) Initialize S
  foreach *step of the episode* **do**
    (b)Choose A from S using policy derived from Q (e.g., ϵ-greedy)
    (c) Take action A; observe R, S'
    (d) $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
    (e) $S \leftarrow S'$
    until S is terminal
  end
end

Algorithm 3: Q-learning

Q-learning will converge to the optimal value function given that there will always be exploration in the algorithm (Watkins & Dayan, 1992). Q-learning updates its value functions by adjusting its old prediction towards the new estimated prediction.

3.6 Dyna-Q learning

Dyna-Q learning uses both model-based methods and model-free methods (R. S. Sutton, 1991). Dyna-Q directly learns the Q-values like in Q-learning but also constructs models that it can run simulations on. As the agent takes more samples from the environment, the algorithm estimates the transition probabilities based from the samples. Dyna-Q learning can then construct a model of the environment and update Q-values by running simulations in it. R. S. Sutton and Barto (2018) defines Dyna-Q as shown in Algorithm 4.

initialize $Q(s, a)$ and $Model(s, a)$ for all $s \in S$ and $a \in A(s)$
repeat
  (a) $S \leftarrow$ current (nonterminal) state
  (b) $A \leftarrow \epsilon$-greedy(S, Q)
  (c) Take action A; observe resultant reward
  (d) $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
  (e) $Model(S, A) \leftarrow R, S'$ (assuming deterministic environment)
  (f) **repeat**
    $S \leftarrow$ random previously observed state
    $A \leftarrow$ random action previously taken in S
    $R, S' \leftarrow Model(S, A)$
    $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
  until n *times*;
until *forever*;

Algorithm 4: Dyna-Q learning

Q-learning was the cycle of acting to get experience and directly learning the values (R. S. Sutton & Barto, 2018). Q-learning is the cycle of acting to get experience and directly learning the values (R. S. Sutton & Barto, 2018). In Algorithm 4, steps (a), (b), (c), and (d) is the part where Q-learning normally cycles through. Dyna-Q adds to this cycle by also using the experience for model learning. By constructing a model, Dyna-Q can use the model to learn Q-values with planning methods such as value iteration (Hwang, Jiang, & Chen, 2013). The update of Q-values happens in two parts of the algorithm: learning from the environment and learning from the model. (d) $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$ part is the same with the Q-learning. Meanwhile, in Algorithm 4 (f) is where the algorithm runs simulations and updates the Q-values.

Dyna-Q learning is capable of learning faster than Q-learning (Hwang et al., 2013). Dyna-Q is also advantageous in sampling because the stock trading environment can be expensive as each action costs money.

3.7 Reinforcement Learning Algorithm Selection

Q-learning and Dyna-Q learning are the RL algorithms selected for analyzing and comparing a model-free learning method and a dyna learning method. Model-based learning methods need to learn the transition models of the environment (Gosavi, 2009). Because of this, value function approximations and modeling will be the main focus of model-based learning methods. These methods that are related to stock trading include stochastic modeling and deep learning. As

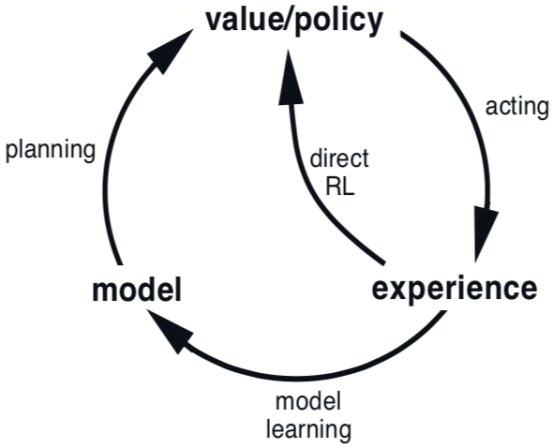


Figure 3.1: Learning Cycle
(R. S. Sutton & Barto, 2018, p.162)

stochastic modeling methods and deep learning methods are out of the scope of this study, model-based learning methods will not be considered. Dyna learning will instead partly represent for model-based learning methods.

3.8 Reinforcement Learning Pipeline

RL algorithms will be implemented as frameworks. This means that given any environment with states and actions, the RL algorithm's value updates must be still be able to function. As long as the RL algorithm is given state inputs and rewards as feedback, it should be able to learn the optimal policy. With a working framework, the main focus will be engineering the environment: state inputs, actions, and rewards. The state inputs are composed of basic stock features: open, high, low, close, and volume. The set of actions will be doing nothing, buying, and selling. The next states will include the stock features and the rewards based on the action. The reward that will depend on what objective function is used. The RL algorithm will be simulated on the Quantopian stock trading environment. The RL will continue its simulations until a certain time. The performance will then be measured and compared to the selected baselines. Based from the analysis, further testing and documentation will be done.

Chapter 4

Research Methodology

This chapter lists and discusses the specific steps and activities that have been performed by the proponents to accomplish the project. The discussion covers the activities from pre-proposal to Final Thesis Writing.

4.1 Research Activities

4.1.1 Review of Related Literature

Literature on RL, stock trading, and algorithmic trading were reviewed to gain the prerequisite knowledge to conduct this research. Specifically, existing works on RL applied to trading or portfolio management was the focus to provide insight into the sound methodologies of how it is implemented and its performance measured. Prerequisite knowledge in implementing RL and algorithmic trading were gained. Terminologies and measurements from stock trading were familiarized during this phase.

4.1.2 Review of Ethical Issues

Ethical concerns were reviewed, particularly of the data: if it had any risk in compromising private information. The General Research Ethics Checklist was also filled up under the supervision of the advisor.

4.1.3 Dataset and Tools Exploration

The proponents searched for datasets that were used in the research. The proponents were able to search for tools that helped such as tools that include the backtesting of trading algorithms. Then, the dataset to be utilized and the tools for backtesting were confirmed for appropriate use. The dataset suited the needs of the research and of the intended implementation of the RL methods and also satisfy ethical concerns. The proponents has familiarized themselves with the dataset and tools for research uses.

4.1.4 Identification of Stock Types

Figure 4.1: Uptrend Stock

Figure 4.2: Downtrend Stock

Various stock types were identified and used as part of the trading simulation. Three types were utilized namely, *Uptrend*, *Downtrend* and *Oscillating*. Such

Figure 4.3: Oscillaing Stock

stocks were classified based on the overall trend from the real market as it is reflected in Quantopian environment as well. Figures 4.1, 4.2 and 4.3 are examples of *Uptrend*, *Downtrend* and *Oscillating* stock trend movements respectively and they were carefully selected and considered in terms of backtesting compatibility and scope requirements.

4.1.5 Identification of RL Methods and Baseline Performance

First, the baseline performance was identified. The baseline served as core component in comparing and assessing the performance of RL methods. The baseline could take the form of trading algorithms or trading strategies. Then RL methods needed justifications for selecting such methods. These justifications were derived from or connected to existing works of applied RL to trading.

4.1.6 Implementation of RL Methods in Quantopian

RL methods were implemented in the form of trading algorithms and complied with the format compatible with the Quantopian platform. There was no form of data cleaning done as the data in Quantopian has already been preprocessed. The proponents were able to adapt their implementations compatible with the data and format of Quantopian.

4.1.7 Comparative Analysis of Performance

The performance of RL methods were measured through backtesting. Backtesting is an assessment of how accurately the algorithm predicted and consequently how well it performed. These results were compared to the baseline to gain a overall understanding of how well RL performed. Then RL methods were compared to each other and derived the causes for their performance.

4.1.8 Documentation

Documentation was of recurring importance and was done all throughout the research. Important concepts and findings were documented in each phase to serve as wayposts to the proponents.

4.2 Calendar of Activities

Table 4.1 shows a Gantt chart of the activities. Each bullet represents approximately one week worth of activity.

4.3 Algorithm Validation

Before Q-learning and Dyna-Q learning can be implemented into the Quantopian platform, the algorithms themselves must first be validated. **OpenAI Gym** is a library filled with toy environments specifically structured for RL (Brockman et al., 2016). The developed Q-learning and Dyna-Q learning algorithms are tested on these environments in order to validate their implementation. The implementation is validated when the performance indicates that the algorithm converges to a more optimal policy. The "learning" or convergence of the algorithm is indicated when the total rewards are getting higher or when the agent stabilizes its total rewards to a certain number. A random policy will be used as benchmark. A random policy has the hyperparameter $\epsilon = 1.0$. An ϵ of 1.0 means that the algorithm does not exploit its knowledge of the best action and is always exploring, i.e. always choosing a random action.

The toy environments used were the following: Taxi, Blackjack, and Cart Pole. For the taxi environment its state space is represented as a single discrete value.

Table 4.1: Timetable of Activities

Activities (2018-2019)	A	S	O	N	D	J	F	M	A	M	J	J
Prerequisite Learning of RL and Trading Concepts	••••	••••	••••	••••	•							
Identification of Reinforcement Learning Methods and Baseline Performance				••••	•							
Review of Ethical Issues					•							
Dataset and Tools Exploration						•						
Implementation of RL and Trading Algorithms in Quantopian						•	••••	••••	••••			
Comparative Analysis of RL and Trading Algorithms										••••	••••	••••
Review of Related Literature	•	•	•	•	•	•	•	•	•	•	•	•
Documentation	•	•	•	•	•	•	•	•	•	•	•	•

For Blackjack, its state space is represented as a list or array of values that are discrete. Finally for the Cart Pole environment, its state space is represented as a list of values that are continuous.

Table 4.2: Toy Environments

	State Space	Action Space	Demonstrated Learning
Taxi	Single Discrete	Single Discrete	Yes
Blackjack	Multiple Discrete	Single Discrete	Yes
Cart Pole	Multiple Continuous	Single Discrete	No

Table 4.3: Taxi: Total Rewards

	Total Rewards
Random Policy	-384,744
Q-Learning	-83,140
Dyna-Q Learning	-23,895

Table 4.4: Blackjack: Total Rewards

	Total Rewards
Random Policy	-219
Q-Learning	-156
Dyna-Q Learning	-126

Table 4.5: Cart Pole: Total Rewards

	Total Rewards
Random Policy	11,370
Q-Learning	11,467
Dyna-Q Learning	11,146

Based from Tables 4.3 and 4.4, Dyna-Q learning is shown to learn faster than Q-learning. Both Q-learning and Dyna-Q learning perform better than the random policy. This indicates that both RL algorithms are learning the optimal policy from the environment. Contrasted to the Cart Pole environment on Table 4.5, Q-learning and Dyna-Q learning perform not much better than the random policy, in some cases, even worse than the random policy. This is because the current implementation of Q-learning and Dyna-Q learning is a tabular approach. Continuous values expands the state space to infinity. This prevents the algorithms to learn as learning requires the trial and error on the same state for a certain number of times.

Based from the results, Q-learning and Dyna-Q learning have been validated to be capable of learning from discrete state spaces. For these algorithms to perform properly on the Quantopian platform, the states must be discretized.

Figure 4.4: Taxi: Random Policy

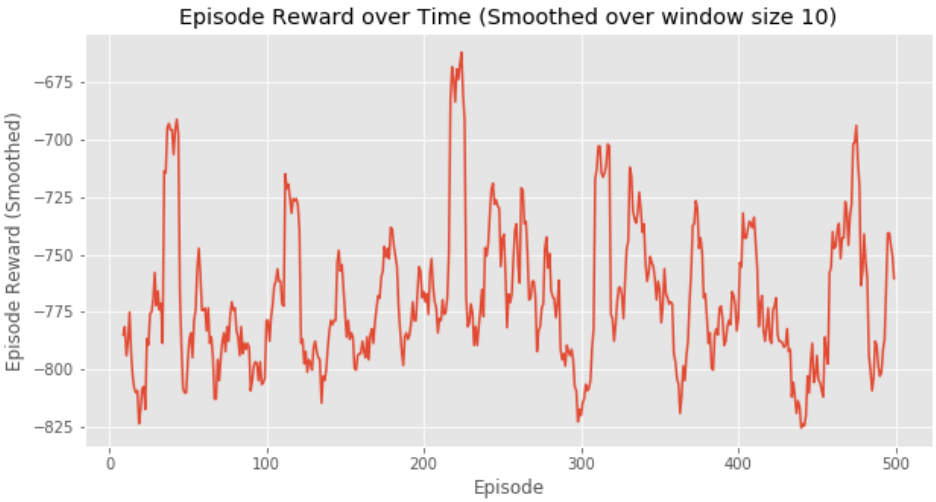


Figure 4.5: Taxi: Q-Learning

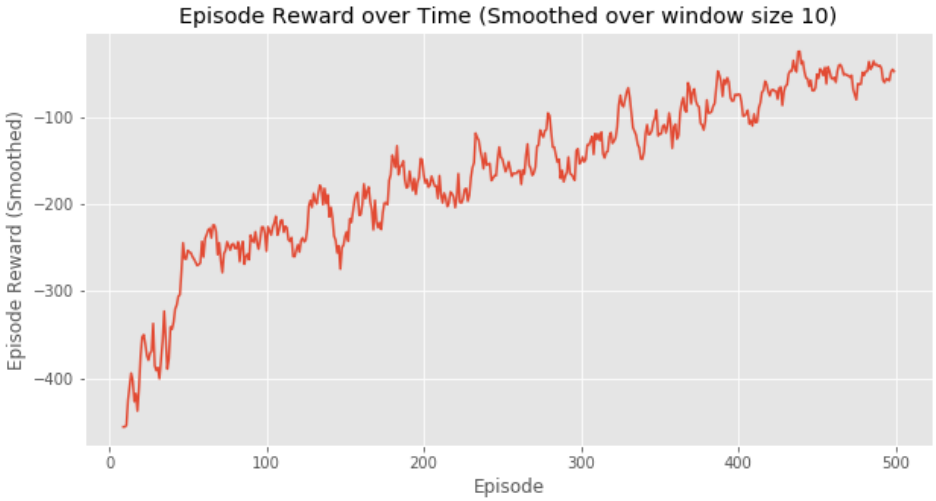


Figure 4.6: Taxi: Dyna-Q Learning

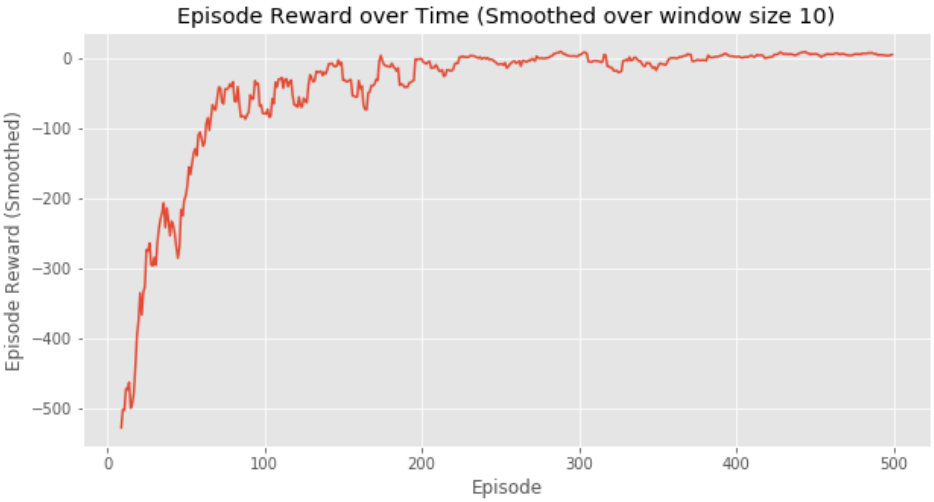


Figure 4.7: Blackjack: Random Policy

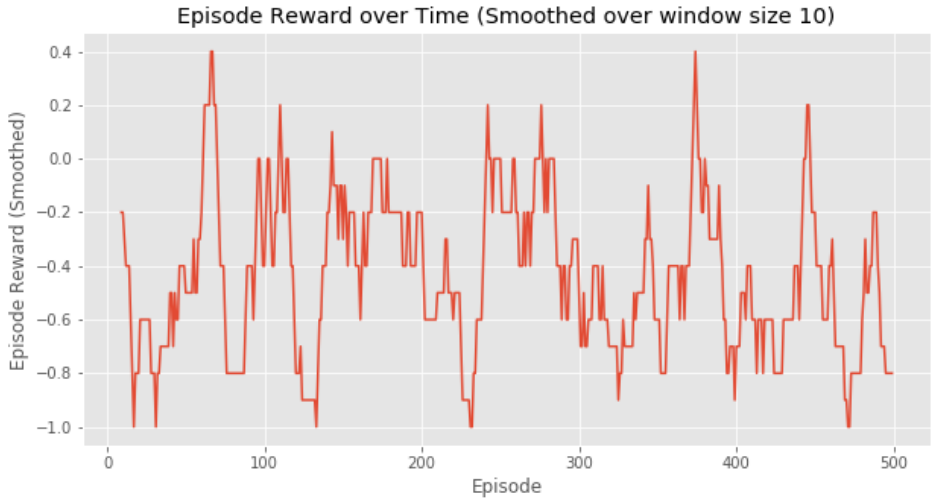


Figure 4.8: Blackjack: Q-Learning Episode Reward over Time

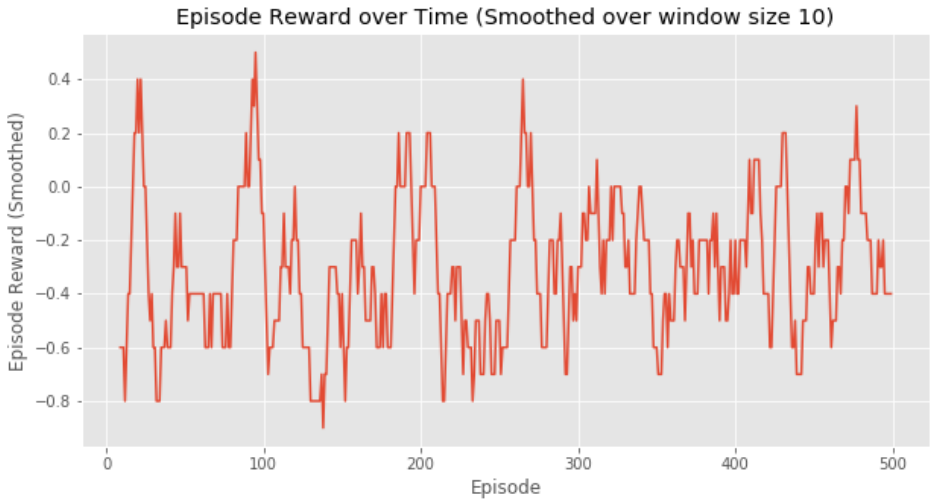


Figure 4.9: Blackjack: Dyna-Q Learning

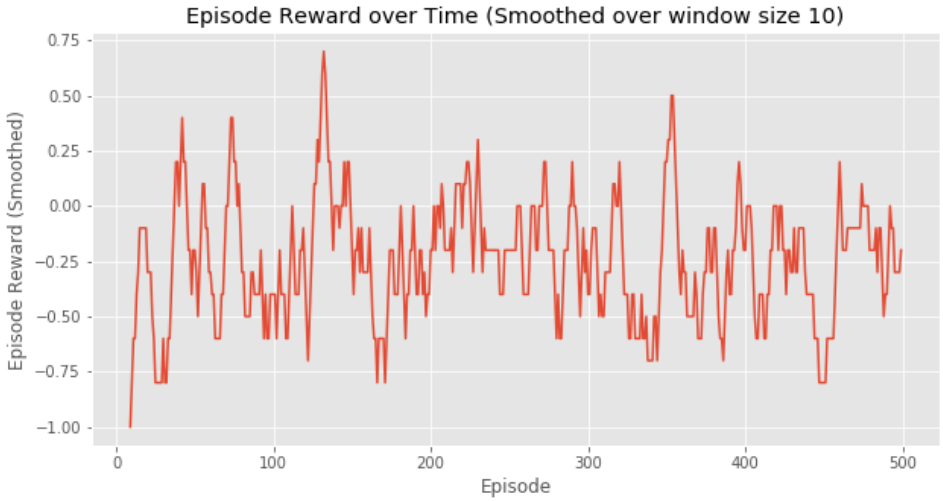


Figure 4.10: Cart Pole: Random Policy

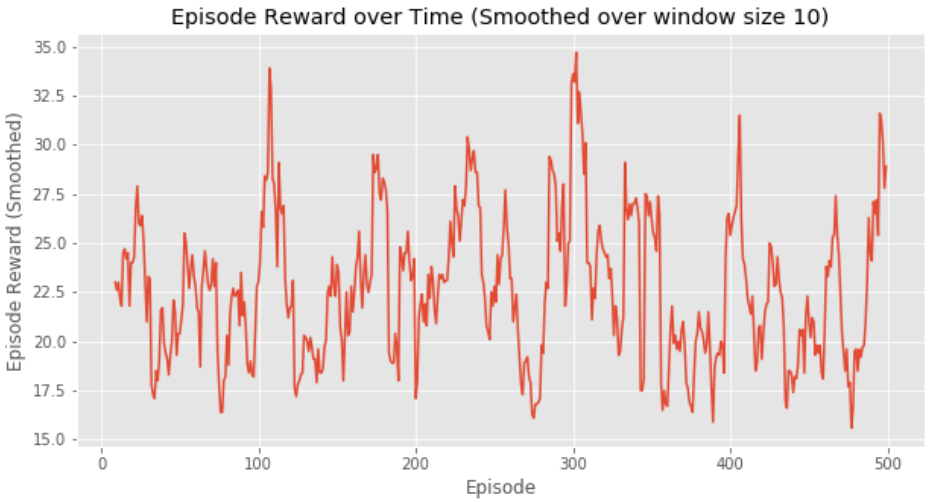


Figure 4.11: Cart Pole: Q-Learning Episode Reward over Time

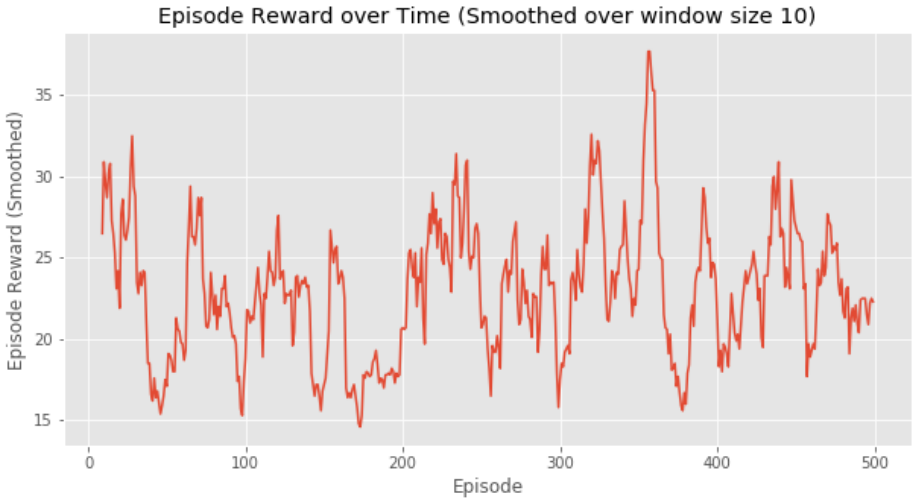
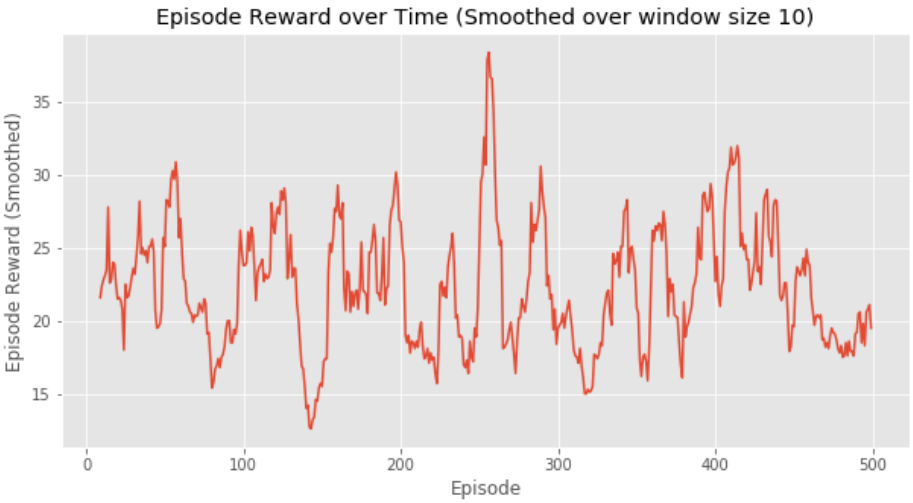


Figure 4.12: Cart Pole: Dyna-Q Learning



Chapter 5

Experimental Framework

5.1 Technical Indicators

5.1.1 Trading Environment Conditions

All stock data and trading simulations are tested in the Quantopian backtesting environment. Such environment simulation allows borrowing of cash and the amount borrowed is normally 2x to 3x based on the cash needed in order to buy a number of shares of a certain stock. The occurrence of less than 100% negative returns is possible due to the feature mentioned above. The experimentation is conducted under the following factors seen in Table 5.1.

Table 5.1: Trading Environment Conditions

Duration	Type of Stock	Frequency
15 years	WMT (Uptrend)	Daily
5 years	CLF (Downtrend)	Hourly
6 months	ACTG (Oscillating)	Minutely

The *Duration* refers to the length of the trading period, having 15 years, 5 years and 6 months as long term, middle term and short term respectively. The *Type of Stock* are the descriptions of the overall behavior of different stocks which is used as datasets for the trading algorithms upon simulation. The primary stocks used are *Walmart Inc. (Uptrend)*, *Cleveland-Cliffs Inc. (Downtrend)* and *Acacia Research Corporation (Oscillating)* as uptrend, downtrend and oscillating respectively. *Frequency*, also called as time frame, determines how often trad-

ing algorithms make specific actions. Daily, hourly and minutely are the time frequencies used in the series of test cases.

5.1.2 Initial Capital

A series of experiments will be conducted under the default starting capital of Quantopian, which is 10 million U.S. Dollars. Likewise, the proponents will also consider experiments on the algorithms under the starting capital of 10 thousand U.S. Dollars, which is believed to be a more realistic setup for trading venture.

5.2 Reinforcement Learning

5.2.1 Testing in Phases

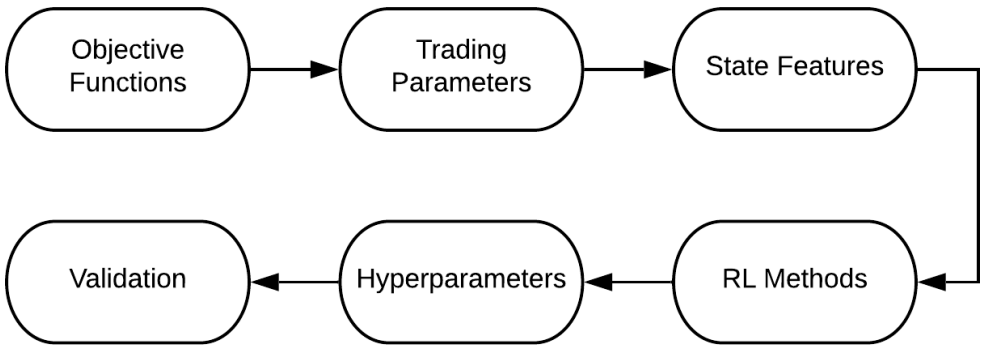
There are many parameters for RL to be tested for. In order to prevent the test cases from exponentially expanding, the different parameters were segmented into test phases. Each test case is composed of 11 independent variables namely: RL method, Trading Duration, Trading Frequency, Stock Type, State Features, Reward or Objective Function, Learning Rate, Discount Rate, Epsilon, Hallucinations, and History. RL method specifies which algorithm was used: Q-learning, Dyna-Q, or random policy. Trading duration specifies how long the agent traded for: 6 months, 5 years, 15 years. Trading frequency specifies how frequently the agent traded: minutely, hourly, or daily. Stock type specifies what kind of stock was traded upon: uptrend, downtrend, and oscillating, also known as sideways. State features specifies what features were used as states: Basic features or Basic features with Indicators. The Reward or Objective function is the metric the agent is maximizing for. Learning rate specifies how much the agent will adjust given new information. Discount rate specifies how far into the future the agent will consider in terms of rewards. Epsilon specifies how often the agent takes random actions to explore the environment. Hallucinations only applies to Dyna-Q and specifies how many times Dyna-Q will simulate on its internal model before taking another action. Finally, History specifies the number of data points towards the past is considered.

RL will be tested in phases, with each phase testing for the best parameter of an independent variable. From each phase, the parameter which yields the best performance will be selected as a fixed parameter for the succeeding test cases. The best performance is the test case with the highest mean of total returns.

Table 5.2: Experimental Framework for RL

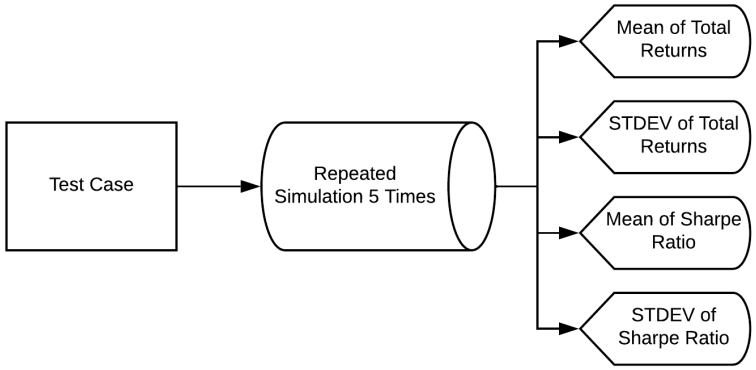
Phases:	Independent Variable	Test Case 1	Test Case 2	Test Case 3	Test Case 4
1. Objective Functions					
	1.1 Objective Function / Reward	Profit	Sharpe Ratio	Net Worth	Total Returns
2. Trading Parameters					
	2.1 Trading Duration	6 Months	5 Years	15 Years	
	2.2 Trading Frequency	Minutely	Hourly	Daily	
	2.3 Stock Type	Oscillating	Uptrend	Downtrend	
3. State Features					
	3.1 State Features	Basics (Only)	Basics + Indicators		
4. RL Methods					
	4.1 RL Methods	Random Policy	Q-Learning	Dyna-Q	
5. Hypeparameters					
	5.1 Learning Rate	0.01	0.1	0.5	
	5.2 Discount Rate	0.5	0.99	1	
	5.3 Epsilon	0.01	0.1	0.5	
	5.4 Hallucinations	10	100	1000	
6. Validation					
	6.1 Nine Different Stocks	3x Oscillating	3x Uptrend	3x Downtrend	

Figure 5.1: Flow Chart for RL Test Phases



Given a test case, this test case will be simulated for 5 times. For each time, the Total Returns and the Sharpe Ratio is recorded at the end of the simulation. This repeated simulations are necessary because the RL algorithms implemented has an element of stochasticity in it. Specifically, during exploration, the agent will randomly choose an action. Because of this, simulations run under the same test case nevertheless yields varying performances. These 5 performances are then

Figure 5.2: Flow Chart for RL Test Phases with Best Parameters



aggregated into their mean and standard deviation resulting into 4 metrics: Mean of Total Returns, Standard Deviation of Total Returns, Mean of Sharpe Ratio, and Standard Deviation of Sharpe Ratio. The parameters that yield the highest mean for total returns is chosen to be used for the succeeding test phases. This method attempts to extract the best possible performance from RL for the comparison between technical indicators and RL methods.

5.2.2 Objective Functions

Objective functions or the reward function is what the algorithm maximizes for. In RL, the objective function serves as the feedback as to determine how good a certain action is. It is the first phase to be tested for as the objective function greatly determines what an agent will learn and how it will behave. Immediate profit is calculated as $Money_t - Money_{t-1}$. It is the difference between the current money held and the previous money held. Net Worth is the total amount of value the agent holds and is calculated as $Money + N_{shares} * Value$. It is the sum of the money held and the value of the shares held. The total value of the shares held is the number of shares multiplied by the current market value of one share.

Table 5.3: Objective Functions

Objective Functions
Profit
Sharpe Ratio
Net Worth
Total Returns

Since the objective functions is the first test phase, the parameters for the other 10 variables were arbitrarily selected. As the test phases are completed, there will be less arbitrary parameters selected. There are 4 test cases, 1 for each objective function. Each test case is ran 5 times resulting to a total of 20 tests for this phase. After the simulations, the performance of the 4 objective functions will be evaluated and the objective function with the highest mean for total returns will selected for the succeeding test phases.

Implementation of Actions

In implementing RL in the environment of stock trading, there were necessary representations that took place in order to be able to trade. The first representation is that the environment needed to be able to do three distinct actions which represent actions states which are, **0** refers to the environment doing nothing or just waiting, **1** refers to the environment buying the stock at this particular time and **2** refers to the environment selling the stock at this particular time. This gave the environment to take an action based on it deems correct or from its rewards. Another implementation that took place was the *Number of Stocks to be Traded*. Taking into account the limitation of money, the number of stocks to be bought or sold was defined through the *sigmoid function* as

$$S(x) = \frac{1}{1 + e^{-x}} \tag{5.1}$$

where x would refer to the duration or how long are you into trading and then was multiplied to the total money the environment has. This is to make sure that the first random actions that the environment takes would not automatically spend a huge amount, and would only spend a little. As the x increase, so does the volume of money to be spent.

5.2.3 Trading Environment Conditions

The next phase to be tested for are the trading environment conditions. These include the trading duration, trading frequency, and the stock type. There are 3 stock types each corresponding to a certain stock. Throughout the test cases, the agent only trades on the one selected stock. This is the second phase in testing in order to establish the environment that the agent will optimize in. The first and second phases lays the foundation for tuning the parameters. With these first 2 phases, what the agent will maximize for and under what conditions

is established. Succeeding test phases will optimize for this objective function under these trading conditions. This will set the experimental conditions for the succeeding phases. The same parameters in technical indicators are used as in Table 5.1. A total of 135 simulations were conducted on the 27 test cases of the trading parameters. The 27 test cases came from the 3 independent variables: trading duration, trading frequency, and stock type that had 3 parameters each. The best parameters for duration, frequency, and stock type will be determined in this phase.

5.2.4 State Features

State features are representations of the trading environment. This determines how well the agent will learn. The state features were separated into two categories: **Basics** and **Indicators**. The Basic features are composed of the 6 stock properties, which are the common representations in stock charts, namely: **open**, **high**, **low**, **close**, **price**, and **volume**. The Indicators are computational features adapted from the technical indicators as well as the available algorithmic libraries in Quantopian. These indicator features are composed of 7 features with our self-engineered features labeled as the following: **ATR**, **Stoch**, **Bollinger bands**, **RSI**, **OBV**, **Net Worth**, and **Purchasing Power**. Average True Range (ATR) is an indicator that measures the volatility of the market by using the *high*, *low* and *close* properties of a certain stock. It then produces *upside signals* and *downside signals* as guides whether to buy of to sell. Stoch makes use of two trend following lines commonly labeled as *slowk* and *slowd*. They are compared against upper and lower benmarks which signals when to buy and to sell. Bollinger Bands features three bands which are the *upper band*, *lower band* and *middle band*. Both the *upper band* and *lower band* are derived from a computation between the *middle band* the stock series, which are used from buying and selling signals afterwards. Net Worth describes whatever is left from the portfolio after selling all assets or stocks and paying off personal debts. Purchasing power is calculated as $Money/Value$. Purchasing power is the money held divided by the current value of one share; It represents how many shares the agent can buy at the moment.

Implementation of State Representation

The testing was done with two variables: **Basics** and **Basics+Indicators**. Basics would only include the basic features while Basics+Indicators would include both the basic features and the indicator features. The purpose of this testing is to determine if adding the indicator features would improve performance. There

were 2 test cases for state features and a total of 10 simulations were conducted. Either Basics or Basics+Indicators will be selected for the succeeding test phases. Besides the State Representation, the a history parameter was also considered, where based on the duration configuration, the number of data points or how far into the past would it consider.

Table 5.4: State Features

Basics	Indicators
open	atr
high	stock
low	bollinger bands
close	rsi
price	obv
volume	net worth
	purchasing power

5.2.5 RL Methods

RL methods are the algorithms used for learning such as Q-learning and Dyna-Q learning. A random policy which decides its actions randomly was added to the parameters of RL methods to serve as the benchmark for determining whether the algorithms did learn something significant from the environment. If there is little difference between the RL methods and a random policy, then that would suggest that the RL methods failed to learn. This is the third phase in the testing that hones in on which method the parameters are tuned for. There were 3 test cases, 1 for each method. A total of 15 simulations were conducted. The best RL method will be determined in this phase. This phase will also determine if RL methods are actually learning in this environment.

Table 5.5: RL Methods

RL Methods
Random Policy
Q-Learning
Dyna-Q Learning

5.2.6 Hyperparameters

The *hyperparameters* of an RL method are the learning rate, discount rate, and epsilon. The *Learning Rate* determines how much the agent will adjust given new feedback. The *Discount Rate* determines how far-sighted the agent will consider in terms of reward. The *Epsilon* or ϵ is the exploration parameter and determines how much the agent will explore. It denotes the probability of exploration. *Hallucinations* is a hyperparameter only available to Dyna-Q. Hallucinations are the simulations run on the internal model of Dyna-Q in order for the agent to converge on the optimal policy faster. The Hallucination hyperparameter indicates the number of simulations done before each action. The fourth phase of the testing is where hyperparameter tuning occurs. This is where we start to really optimize for the unique conditions we have chosen from the objective functions to the trading parameters. Hallucinations were split from the rest of the hyperparameters in testing because this hyperparameter only applied to Dyna-Q. For the first half, there were 27 test cases. For the second half for the hallucinations, there were 3 test cases. Combined, this test phase had 150 simulations. The best hyperparameters for the learning rate, discount rate, and epsilon will be selected in this phase.

Table 5.6: Hyperparameters

Learning Rate	Discount Rate	Epsilon (ϵ)	Hallucinations
0.01	0.5	0.01	10
0.1	0.99	0.1	100
0.5	1.0	0.5	1000

5.2.7 Validation

The last phase for testing is the validation phase. In the validation phase, we choose 9 different stocks and aggregate their performance. There are 3 stock types and for each stock type, 3 stocks have that stock type. The purpose of the last test phase is to validate whether the optimized RL would perform positively on other stocks. With 9 test cases corresponding to a stock, there were a total of 45 simulations conducted.

Table 5.7: Stocks Table for Validation Testing

Company Name	Stock Abbreviation	Stock Type
Acacia Research Corporation	ACTG	Oscillating
Walmart Inc.	WMT	Uptrend
Cleveland-Cliffs Inc.	CLF	Downtrend
Altaba Inc.	AABA	Oscillating
Vertex Pharmaceutical Inc.	VRTX	Uptrend
BRF S.A.	BRFS	Downtrend
China Southern Airlines Co Ltd	ZNH	Oscillating
Illumina, Inc.	ILMN	Uptrend
China Eastern Airlines Corporation Ltd.	CEA	Downtrend

Chapter 6

Results and Analysis

6.1 Technical Indicators

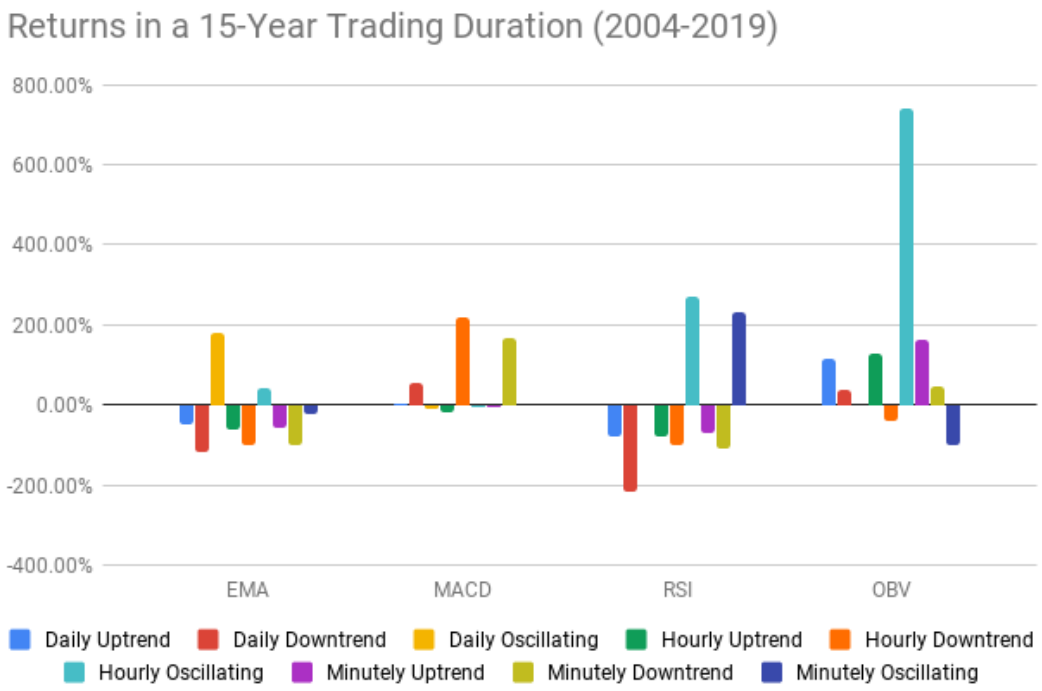
6.1.1 Initial Capital of 10 Million Dollars

Table 6.1: Returns in a 15-Year Trading Duration (2004-2019)

	EMA	MACD	RSI	OBV
Daily Uptrend	-48.58%	3.58%	-79.55%	113.23%
Daily Downtrend	-118.00%	53.88%	-216.60%	36.22%
Daily Oscillating	181.52%	-11.86%	0.33%	-149.48%
Hourly Uptrend	-62.05%	-18.11%	-79.98%	126.35%
Hourly Downtrend	-101.43%	219.94%	-99.79%	-41.18%
Hourly Oscillating	43.00%	-6.49%	268.14%	739.21%
Minutely Uptrend	-56.42%	-7.29%	-68.45%	162.04%
Minutely Downtrend	-101.85%	168.19%	-109.22%	46.05%
Minutely Oscillating	-21.29%	0.62%	230.34%	-98.77%

Results have shown in the experiments under the duration of 15 years date taken from January 1, 2004, to January 1, 2019, EMA performs worst when trading downtrend and uptrend stocks, resulting to around 100% negative returns. This holds true for all tested timeframe frequencies (i.e. daily, hourly, minutely). On the other hand, EMA shows promise when used to trade on oscillating stocks, garnering positive returns as high as 118%.

Figure 6.1: Returns in a 15-Year Trading Duration (2004-2019)



MACD performs well when traded on downtrend stocks which generates returns as high as 200%. This is evidently observed in hourly trading. Almost -10% returns, however, are generated by algorithm when traded on uptrend and oscillating stocks.

RSI resulted negative total return percentage with stocks that were uptrend and downtrend while it resulted a positive return percentage when the stocks was oscillating. For OBV, noticeable results were the longer the trading time the smaller negative total return percentage it has.

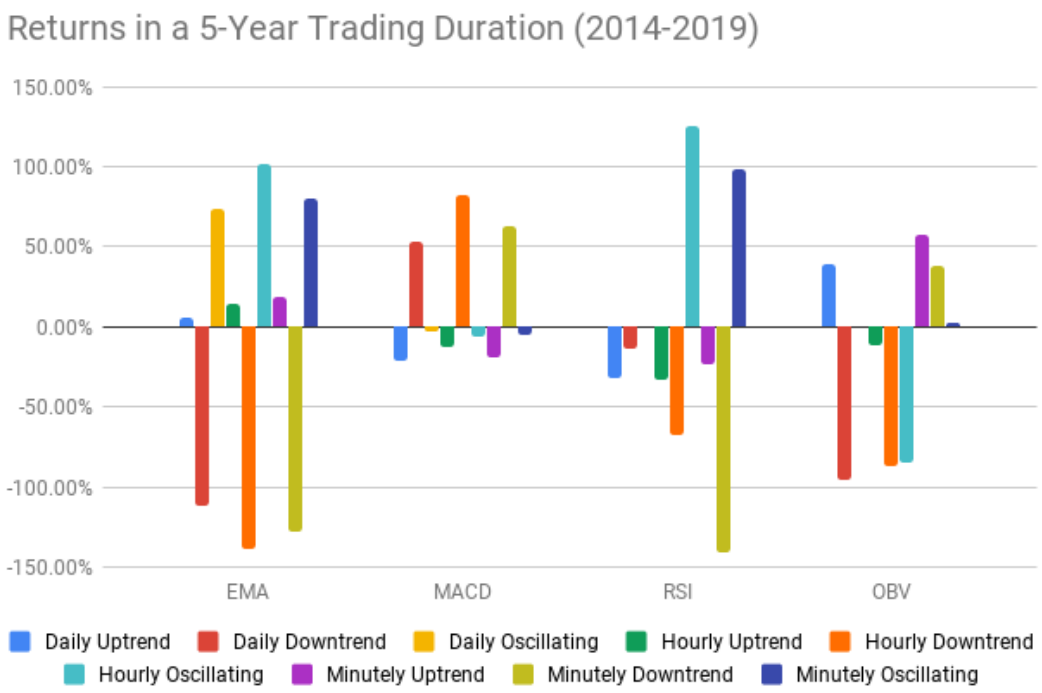
In the experiments of having a 5-year trading duration date taken from January 1, 2014 to January 1, 2019, EMA performs worst for all time frame frequencies when trading on downtrend stocks, generating negative returns of around -100%. The algorithm generates positive returns trading on uptrend stocks yet relatively lower compared to oscillating stocks, which generates 70-100% positive returns.

MACD still performs well downtrend stocks merely generating 100% returns. Its performance becomes worse when trading on uptrend stocks compared to the 15-year duration. Negative returns plummets to around -20%.

Table 6.2: Returns in a 5-Year Trading Duration (2014-2019)

	EMA	MACD	RSI	OBV
Daily Uptrend	5.65%	-21.08%	-31.92%	39.11%
Daily Downtrend	-112.12%	52.69%	-13.32%	-95.70%
Daily Oscillating	73.17%	-2.57%	53.06%	51.86%
Hourly Uptrend	14.49%	-12.64%	-32.62%	-11.89%
Hourly Downtrend	-138.49%	82.63%	-68.05%	-86.87%
Hourly Oscillating	101.44%	-5.87%	125.23%	-85.28%
Minutely Uptrend	18.41%	-18.79%	-23.46%	57.36%
Minutely Downtrend	-128.35%	62.61%	-141.21%	37.82%
Minutely Oscillating	79.57%	-5.16%	98.54%	1.89%

Figure 6.2: Returns in a 5-Year Trading Duration (2014-2019)

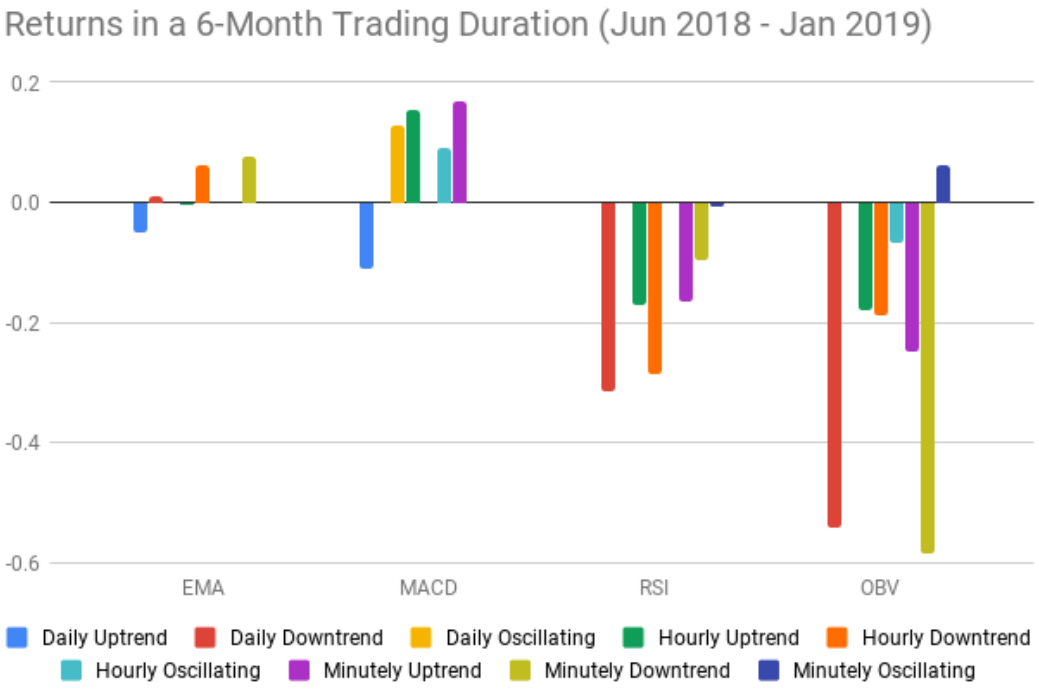


RSI derived a negative Sharpe ratio with a stock that was uptrend. Stocks that were oscillating gave a positive total return percentage, and high percentage of 125.23% when trade was hourly. OBV, on the other hand, the result of hourly trades returned a total negative percentage.

Table 6.3: Returns in a 6-Month Trading Duration (Jun 2018 - Jan 2019)

	EMA	MACD	RSI	OBV
Daily Uptrend	-4.92%	-10.99%	-19.47%	-2.25%
Daily Downtrend	1.06%	-0.06%	-31.53%	-54.10%
Daily Oscillating	0%	12.88%	-0.15%	1.05%
Hourly Uptrend	-0.38%	15.42%	-17.17%	-17.92%
Hourly Downtrend	6.24%	-0.08%	-28.46%	-18.67%
Hourly Oscillating	0%	8.99%	-0.12%	-6.75%
Minutely Uptrend	-0.01%	16.83%	-16.45%	-24.67%
Minutely Downtrend	7.78%	-0.09%	-9.65%	-58.35%
Minutely Oscillating	0%	0%	-0.72%	6.08%

Figure 6.3: Returns in a 6-Month Trading Duration (Jun 2018 - Jan 2019)



EMA does not perform well under a very short trading duration, that is, in a 6 month period. This is observed among uptrend, downtrend and oscillating stocks at any given time frame frequencies. Although it shows positive returns when traded on downtrend stocks, the result is barely 10%.

For MACD, trading on downtrend stocks generally performs well except when the time frame frequency is set to Daily, which generates -10% return. Trading on uptrend stocks receives roughly about 10% positive returns while on oscillating stocks results to around -0.10% returns. Nonetheless, it is a relatively bearable setback.

RSI had an outcome of negative total return percentage with the factors of stocks being uptrend, downtrend, and oscillating and trading style of daily, hourly, and minutely. For OBV, it resulted less positive total return percentage. It is evident that the shorter duration, the outcome would be more on the negative side.

6.1.2 Initial Capital of 10 Thousand Dollars

Table 6.4: Returns in a 15-Year Trading Duration (2004-2019)

	EMA	MACD	RSI	OBV
Daily Uptrend	-54.47%	-7.29%	-70.50%	161.45%
Daily Downtrend	-103.57%	168.19%	-65.24%	43.03%
Daily Oscillating	-151.96%	0.62%	-97.43%	-98.08%
Hourly Uptrend	-59.14%	-18.55%	-57.49%	-34.06%
Hourly Downtrend	-103.24%	276.49%	-101.72%	-99.76%
Hourly Oscillating	-168.02%	20.14%	-95.40%	-52.90%
Minutely Uptrend	-66.51%	-86.82%	-52.16%	0.60%
Minutely Downtrend	-102.24%	-41.56%	-100.93%	-99.30%
Minutely Oscillating	-175.52%	164.61%	-94.30%	-75.97%

For 15 year trading duration with $10000 as starting capital, technical indicator MACD performed best among EMA, RSI, and OBV. It shows that three of the experiments resulted positive returns and least negative returns. MACDs Hourly Downtrend performed the best with 276.49% as total returns percentage. At the same time EMA performed the worst because of high negative returns. EMAs Minutely Oscillating performed the worst with -175.52% as total returns percentage.

For 5 year trading duration with $10000 as starting capital, MACD conducted two positive returns and with MACD minutely oscillating performing the best returning a positive return percentage of 4898.42%. While RSI conducting the worst because of all negative returns and with RSI daily downtrend performing the worst with a negative return percentage of -280.43%.

Table 6.5: Returns in a 5-Year Trading Duration (2014-2019)

	EMA	MACD	RSI	OBV
Daily Uptrend	22.58%	-17.56%	-24.24%	53.09%
Daily Downtrend	-121.98%	118.75%	-280.43%	75.95%
Daily Oscillating	26.26%	-58.98%	-66.88%	-58.44%
Hourly Uptrend	23.87%	-11.89%	-7.97%	-17.77%
Hourly Downtrend	-119.67%	16.55%	-92.12%	-85.46%
Hourly Oscillating	24.39%	-57.72%	-31.72%	-95.89%
Minutely Uptrend	23.63%	-27.60%	-5.05%	-16.08%
Minutely Downtrend	-114.09%	-1.17%	-97.89%	-94.09%
Minutely Oscillating	19.08%	4898.42%	-34.57%	-96.78%

Table 6.6: Returns in a 6-Month Trading Duration (Jun 2018 - Jan 2019)

	EMA	MACD	RSI	OBV
Daily Uptrend	0.60%	11.80%	-15.51%	-32.72%
Daily Downtrend	14.24%	7.68%	14.47%	-55.08%
Daily Oscillating	0.00%	-23.10%	-29.60%	22.01%
Hourly Uptrend	0.72%	13.26%	-15.47%	4.47%
Hourly Downtrend	13.47%	9.30%	22.66%	-34.25%
Hourly Oscillating	0.00%	20.75%	-7.67%	-18.71%
Minutely Uptrend	0.72%	7.78%	-15.53%	5.57%
Minutely Downtrend	11.91%	4.87%	26.69%	-48.51%
Minutely Oscillating	0.00%	247.35%	-3.51%	-21.80%

For 6 Month trading duration date taken from June 1,2018 to January 1, 2019 and with $10000 as starting capital, MACD performed best and positive however it gave one negative result. MACD minutely oscillating performs the best with a total return percentage of 247.35%. OBV performed worst because of high negative returns. OBVs daily downtrend executed with a total return of -55.08%.

6.1.3 General Analysis of Technical Indicators

While EMA points traders to the direction of the trend, it simply lacks the appropriate signals since the stock market undergoes great deals of price ranges, making the algorithm ineffective. Such technical indicator is only concerned with the line

Figure 6.4: Returns of Various Trading Durations with Initial Capital of 10 Thousand

crossovers which does not consider how wide the divergence prior to making trade descisions.

MACD does not seem to have an identifier for overbought and oversold limits which may be the cause of the algorithm to overextend beyond historical extremes. However, it seems to perform optimally with almost 5000% positive returns under the initial capital of 10 thousand dollars.

The oscillating stock had an initial unit price of $14.32. Algorithm can only buy at most 2756 shares at the time given the initial capital. Unit price started to drop at the first quarter of 2015 to roughly $10 which enabled the algorithm to buy more shares. At this point, the unit price oscillated between $10 and $11, which provided MACD opportunities to slowly gain positive returns by buying and selling at the right times.

From 2015 to the first quarter of 2016, the algorithm was able to transact 48304 shares in a day. This was the time when the unit price significantly dropped from $10 to $5 or below. From 2017 to early 2018, unit price started to increase until $7 which made the algorithm slightly lose returns. But from 2018 to 2019, the unit price once again dropped in which MACD regained its loss from the previous

year.

RSI may have the tendency to extend above and below the thresholds, thereby, creating false buy and false sell signals. Because of such limitation, the overbought and oversold markers can be a unreliable factor to identify momentum shifts.

Since OBV depends on the volume of a certain stock, its limitation lies on the prediction of prices with the buying/selling pressure as basis. This results to a large number of false signals along with valid ones, making it difficult to understand the true behavior. Another limitation is when a massive volume spike occurs in a day, this will affect succeeding indicators.

6.2 Reinforcement Learning

6.2.1 Objective Functions

The parameters selected for this test phase are:

- RL method: Q-learning
- Duration: 15 years
- Frequency: Minutely
- Stock: Oscillating
- States: Basics
- Objective Function: **Independent Variable**
- Learning Rate: 0.1
- Discount Rate: 0.99
- Epsilon: 0.1
- Hallucinations: N/A
- History: 10,000 datapoints

For the objective functions, aside from Sharpe Ratio, the rest of the objective functions had a negative total return. Having the Sharpe ratio as the objective

Table 6.7: Objective Functions Results

Objective Functions				
	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
Profit	-56299.78%	671.80	0.29	0.15
Sharpe Ratio	379.66%	385.84	0.50	0.43
Net Worth	-412.96%	58.92	0.14	0.45
Total Returns	-381.42%	134.95	0.21	0.52

function yielded a 379.66% total returns and a 0.5 Sharpe ratio, both of which are the highest. When the objective function is the Sharpe ratio, the agent chooses its actions to maximize its Sharpe ratio which results into having the highest Sharpe ratio of 0.5 among the objective functions. Although, the Sharpe ratio objective function yielded the highest mean of total returns, it's standard deviation has a value of 385.84. This indicates that its performance varies in a wide range. Still, based from these results, Sharpe ratio will be fixed for the succeeding test phases as the objective function.

6.2.2 Trading Parameters

The parameters selected for this test phase are:

- RL method: Q-learning
- Duration: **Independent Variable**
- Frequency: **Independent Variable**
- Stock: **Independent Variable**
- States: Basics
- Objective Function: Sharpe Ratio
- Learning Rate: 0.1
- Discount Rate: 0.99
- Epsilon: 0.1

- Hallucinations: N/A
- History: Dependent Variable on Trading Frequency

Our implementation of the RL methods requires a variable we labeled as History. History indicates the number of datapoints into the past is considered. We selected that if the trading frequency is minutely, 10,000 datapoints to the past will be considered. For hourly trading, 1,000 datapoints. Lastly for daily trading, 100 datapoints.

Table 6.8: Trading Parameters Results

Trading Parameters						
Duration	Frequency	Stock	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
6 months	Minutely	Oscillating	-99.06%	3.39	0.13	1.61
6 months	Minutely	Uptrend	-50.36%	2.24	1.10	0.54
6 months	Minutely	Downtrend	-3.22%	0.11	0.34	0.32
6 months	Hourly	Oscillating	122.86%	2.82	0.96	0.74
6 months	Hourly	Uptrend	-0.46%	0.40	0.27	0.32
6 months	Hourly	Downtrend	10.58%	0.30	0.32	0.39
6 months	Daily	Oscillating	140.30%	2.33	-0.02	1.09
6 months	Daily	Uptrend	-8.34%	0.35	0.74	0.83
6 months	Daily	Downtrend	-25.86%	0.58	0.19	1.19
5 years	Minutely	Oscillating	-3126.68%	68.57	0.27	0.66
5 years	Minutely	Uptrend	745.52%	44.41	0.35	0.34
5 years	Minutely	Downtrend	358.98%	4.00	0.57	0.48
5 years	Hourly	Oscillating	-2490.92%	66.59	0.46	0.28
5 years	Hourly	Uptrend	-974.64%	24.47	0.48	0.20
5 years	Hourly	Downtrend	440.54%	11.22	-0.07	0.44
5 years	Daily	Oscillating	-1763.94%	42.91	0.60	0.19
5 years	Daily	Uptrend	113.56%	0.16	0.56	0.04
5 years	Daily	Downtrend	-67.24%	0.05	-0.46	0.10
15 years	Minutely	Oscillating	379.66%	385.84	0.50	0.43
15 years	Minutely	Uptrend	-436.26%	179.95	0.27	0.35
15 years	Minutely	Downtrend	-15.12%	0.42	0.22	0.06
15 years	Hourly	Oscillating	-7639.36%	75.21	0.12	0.39
15 years	Hourly	Uptrend	-512.42%	44.37	0.49	0.25
15 years	Hourly	Downtrend	-12.62%	0.50	0.13	0.10
15 years	Daily	Oscillating	1059.60%	13.57	0.57	0.21
15 years	Daily	Uptrend	1079.24%	4.23	0.56	0.05
15 years	Daily	Downtrend	1.98%	2.19	0.05	0.08

Table 6.2.2 shows the top three highest total returns based from different trading parameters. The results show that the best trading condition for RL is 15 years of daily trading on an uptrend stock . Because the agent is incapable of short trading, the agent will perform better in an uptrend stock compared to

Table 6.9: Top 3 Trading Parameters Results

Trading Parameters [Top 3 Mean of Total Returns]						
Duration	Frequency	Stock	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
15 years	Daily	Uptrend	1079.24%	4.23	0.56	0.05
15 years	Daily	Oscillating	1059.60%	13.57	0.57	0.21
5 years	Minutely	Uptrend	745.52%	44.41	0.35	0.34

oscillating and downtrend stocks. It was expected that the trading by minutely for 15 years would yield the highest return because it would have the most data points. With minutely trading the RL would learn more from a larger dataset of experience and with 15 years, the RL would be able to exploit what it learned for a longer duration. Though the duration of 15 years was confirmed to yield the highest return, the trading frequency showed otherwise. From these results, it is inferred that although minutely trading gives more data points, daily trading is better because its datapoints are more smoothed than minutely trading. Minutely trading is perhaps much more susceptible to noise such that RL is incapable of learning quickly from this dataset. Based from these results, the trading duration of 15 years, the frequency of daily trading, and an uptrend stock will be fixed as the trading parameters for the succeeding test phases.

6.2.3 State Features

The parameters selected for this test phase are:

- RL method: Q-learning
- Duration: 15 years
- Frequency: Daily
- Stock: Uptrend
- States: **Independent Variable**
- Objective Function: Sharpe Ratio
- Learning Rate: 0.1
- Discount Rate: 0.99
- Epsilon: 0.1

- Hallucinations: N/A
- History: 100 datapoints

Table 6.10: State Features Results

State Features				
	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
Basics	1079.24%	4.23	0.56	0.05
Basics + Indicators	1001.02%	7.01	0.54	0.10

Between the two state features tested, the features only including the basic features performed better than features that included other technical indicators. It was expected that the basic features with indicator features would perform better as it had more features for the RL to learn from but given the results, it is inferred that adding more features just slowed down the agent from learning the optimal policy. The Basics+Indicators features had a standard deviation of 7.01 compared to the Basics only features of 4.23. This indicates that the agent took a longer time to learn and to stabilize. Based from these results, Basics only features will be fixed for the succeeding test phases as the state features.

6.2.4 RL Methods

The parameters selected for this test phase are:

- RL method: **Independent Variable**
- Duration: 15 years
- Frequency: Daily
- Stock: Uptrend
- States: Basic
- Objective Function: Sharpe Ratio
- Learning Rate: 0.1
- Discount Rate: 0.99

- Epsilon: 0.1
- Hallucinations: N/A
- History: 100 datapoints

Table 6.11: RL methods Results

RL Methods				
	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
Random Policy	-241.72%	10.19	0.12	0.33
Q-learning	1079.24%	4.23	0.56	0.05
Dyna-Q	1193.12%	5.43	0.56	0.09

The RL policy had a total returns of -241.72% compared to the RL methods which generally reached 1000% total returns. This validates that the RL methods are learning and is not simply acting randomly. Dyna-Q had the total returns of 1193.12%, performing better than Q-learning by 114%. Dyna-Q and Q-learning have the same algorithm up to the point that Dyna-Q builds a model based from its experiences and simulates on that model to converge faster. This simulation, or rather hallucinations is what gives Dyna-Q an edge over Q-learning. While Q-learning only relies on its experiences gained from the environment, Dyna-Q is able to supplement its learning with experiences gained from hallucinations. Based from these results, Dyna-Q will be fixed for the succeeding test phases as the RL method.

6.2.5 Hyperparameters

The parameters selected for this test phase are:

- RL method: Dyna-Q
- Duration: 15 years
- Frequency: Daily
- Stock: Uptrend
- States: Basic

- Reward/Objective Function: Sharpe Ratio
- Learning Rate: **Independent Variable**
- Discount Rate: **Independent Variable**
- Epsilon: **Independent Variable**
- Hallucinations: **Independent Variable**
- History: 100 datapoints

Table 6.12: Top 3 Hyperparameters Results

Hyperparameters [Top 3 Mean of Total Returns]						
Learning Rate	Discount Rate	Epsilon	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
0.1	0.5	0.1	1549.70%	1.69	0.61	0.02
0.01	0.99	0.1	1444.04%	0.71	0.60	0.01
0.5	0.99	0.5	1310.38%	4.05	0.58	0.05

Although the top performing parameter has a learning rate of 0.1, its other top contenders have different learning rates. This shows that hyperparameters our dependent on each other and that configurations of hyperparameters is just as critical as the parametric value. A 0.5 discount rate was unexpected as the agent is myopic or short-sighted compared to other discount rates like 0.99. This indicates that the agent acted greedily on the short term. A possible explanation for this is due to the state representation implemented. The states are represented in a relative manner and focuses on capturing the trend of different features. With the discount rate of 0.5, the agent maximizes its rewards based on trends. The optimal epsilon value is observed to be 0.1. This value is reasonable because the agent is exploiting as much as it can but retains enough exploratory action to adapt to its environment. Because Dyna-Q was used, an additional hyperparameter was tested for: Hallucinations. Given a daily trading frequency, the results show that 100 and 1000 hallucinations perform worse in terms of total returns but not by a big margin. This indicates that too much hallucinations leads to overfitting to the experiences the agent currently has and consequently to the internal model. Dyna-Q learns faster by doing these hallucinations but the effectiveness of the hallucinations is determined by the accuracy of the internal model and the experience the agent gained. If the experience gained no longer reflects the environment, then it would result to a faulty internal model. There is also the case that the environment is changing, and too much hallucinations is preventing the agent to adapt to its newfound experience. The best performance an RL agent was able to achieve is a total returns of 1549.70% with its standard deviation of

Table 6.13: Hyperparameters Results

Hyperparameters						
Learning Rate	Discount Rate	Epsilon	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
0.01	0.50	0.01	159.44%	16.45	0.44	0.19
0.01	0.50	0.10	771.46%	14.93	0.54	0.13
0.01	0.50	0.50	-3774.02%	88.77	0.38	0.25
0.01	0.99	0.01	765.40%	22.09	0.47	0.32
0.01	0.99	0.10	1444.04%	0.71	0.60	0.01
0.01	0.99	0.50	-45.64%	26.55	0.46	0.23
0.01	1.00	0.01	-3080.82%	82.44	0.26	0.47
0.01	1.00	0.10	415.46%	14.04	0.50	0.13
0.01	1.00	0.50	131.66%	28.93	0.29	0.38
0.1	0.50	0.01	1212.26%	3.83	0.57	0.04
0.1	0.50	0.10	1549.70%	1.69	0.61	0.02
0.1	0.50	0.50	32.10%	22.36	0.37	0.37
0.1	0.99	0.01	-6621.88%	181.07	0.48	0.26
0.1	0.99	0.10	-6105.72%	136.62	0.37	0.37
0.1	0.99	0.50	916.94%	11.00	0.51	0.18
0.1	1.00	0.01	1232.84%	5.45	0.57	0.08
0.1	1.00	0.10	963.94%	7.11	0.41	0.36
0.1	1.00	0.50	131.66%	28.93	0.42	0.38
0.5	0.50	0.01	1178.46%	6.77	0.53	0.15
0.5	0.50	0.10	947.72%	5.52	0.54	0.07
0.5	0.50	0.50	1023.08%	7.54	0.43	0.35
0.5	0.99	0.01	1258.62%	2.18	0.58	0.02
0.5	0.99	0.10	1132.30%	6.00	0.59	0.01
0.5	0.99	0.50	1310.38%	4.05	0.58	0.05
0.5	1.00	0.01	1212.94%	34.46	0.56	0.19
0.5	1.00	0.10	1033.94%	7.58	0.47	0.27
0.5	1.00	0.50	-4645.62%	83.68	0.14	0.44

1.69. It also achieved a Sharpe ratio of 0.61 and its standard deviation is 0.02. The RL methods demonstrated that it is capable of learning from the stock trading environment.

6.2.6 Validation

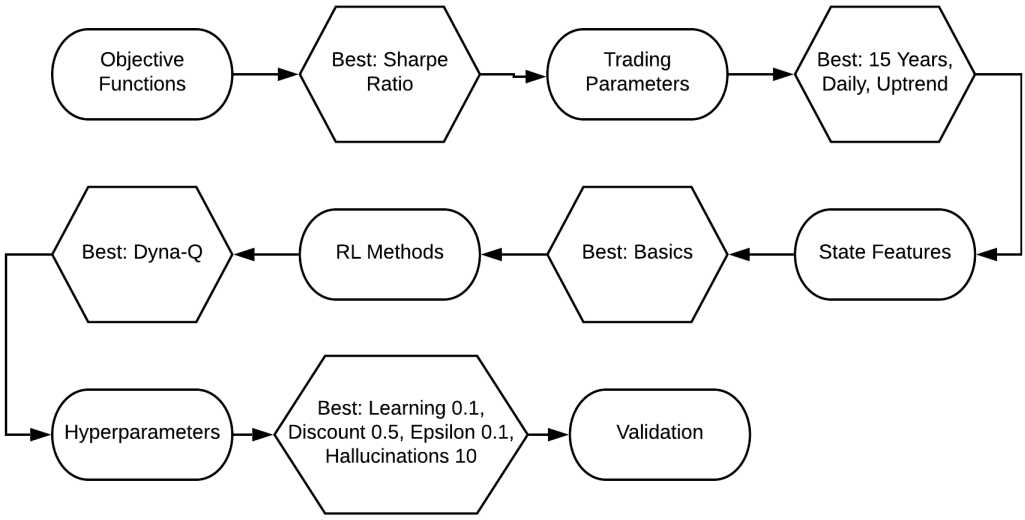
The parameters selected for this test phase are:

- RL method: Dyna-Q
- Duration: 15 years

Table 6.14: Hallucination Results

Hallucinations				
Number of Hallucinations	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
10.00	1549.70%	1.69	0.61	0.02
100.00	1460.54%	1.36	0.60	0.01
1,000.00	1287.92%	1.86	0.58	0.02

Figure 6.5: Flow Chart for RL Test Phases with Best Parameters



- Frequency: Daily
- Stock: Uptrend
- States: Basic
- Reward/Objective Function: Sharpe Ratio
- Learning Rate: 0.1
- Discount Rate: 0.5
- Epsilon: 0.1
- Hallucinations: 10

- History: 100 datapoints

Table 6.15: Validation Results

Validation					
Stock	Stock Type	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
VRTX	Uptrend	1549.70%	1.69	0.61	0.02
ILMN	Uptrend	3586.50%	49.70	0.73	0.23
WMT	Uptrend	38.74%	2.40	0.38	0.06
ACTG	Oscillating	177.08%	4.24	0.00	0.27
AABA	Oscillating	146.04%	0.07	0.35	0.00
ZNH	Oscillating	39.96%	0.84	0.21	0.18
BRF	Downtrend	-68.38%	0.64	-0.02	0.18
CEA	Downtrend	6.44%	0.35	0.10	0.12
CLF	Downtrend	6529.54%	97.39	0.22	0.29

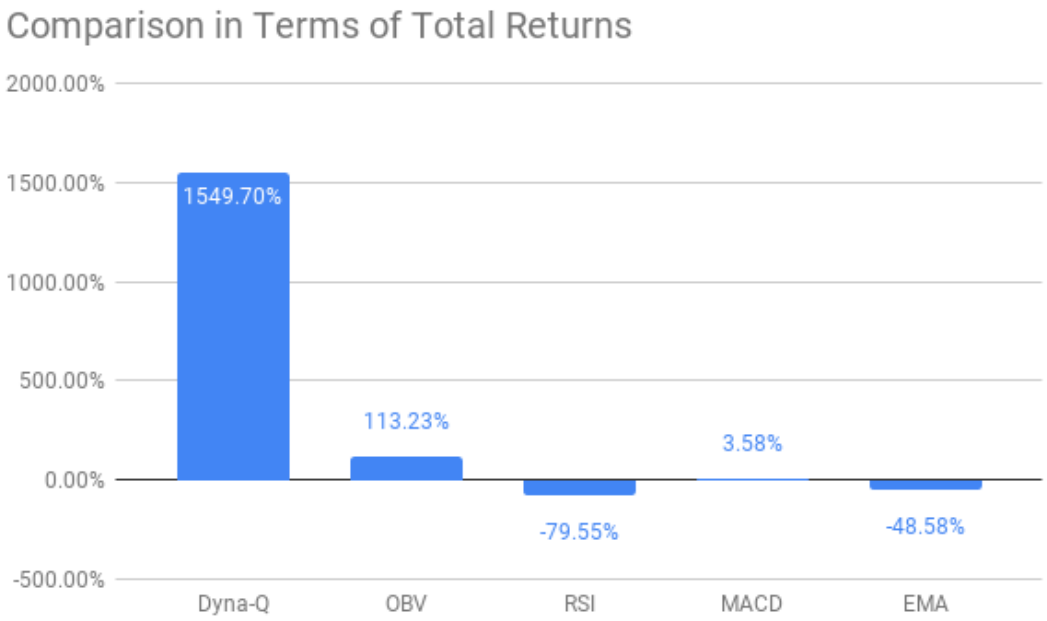
Table 6.16: Validation by Stock Type Results

Validation [Aggregated by Stock Type]				
Stock Type	Mean of Total Returns	Standard Deviation of Total Returns	Mean of Sharpe Ratio	Standard Deviation of Sharpe Ratio
Uptrend	1724.98%	17.93	0.57	0.10
Oscillating	121.03%	1.71	0.19	0.15
Downtrend	2155.87%	32.79	0.10	0.20

RL on average, performed positively except on the downtrend stock BRFS. The most stable performance for RL, indicated by the standard deviations of the total returns, are the oscillating stocks. With the means are further aggregated by stock type, we observe that the oscillating stocks only had a 1.71 mean for standard deviations. The downtrend stocks have the highest mean for total returns and standard deviations. This suggests that it's positive performance is greatly influenced by randomness and can land on a wide range of values. Though limited by only 5 simulations, it has been observed that the value for the total returns tends to lie on far ends such as the instances 12.30% or 21845.20%. As for the Sharpe ratio, the uptrend stocks had the highest mean with an aggregated value of 0.57 when contrasted to 0.19 and 0.10. The validation phase shows that although the parameters were selected according to their past performances on an uptrend stock, the parameters does not necessarily lead to always having the highest total returns with uptrend stocks.

6.3 Comparison Between Technical Indicators and RL Methods

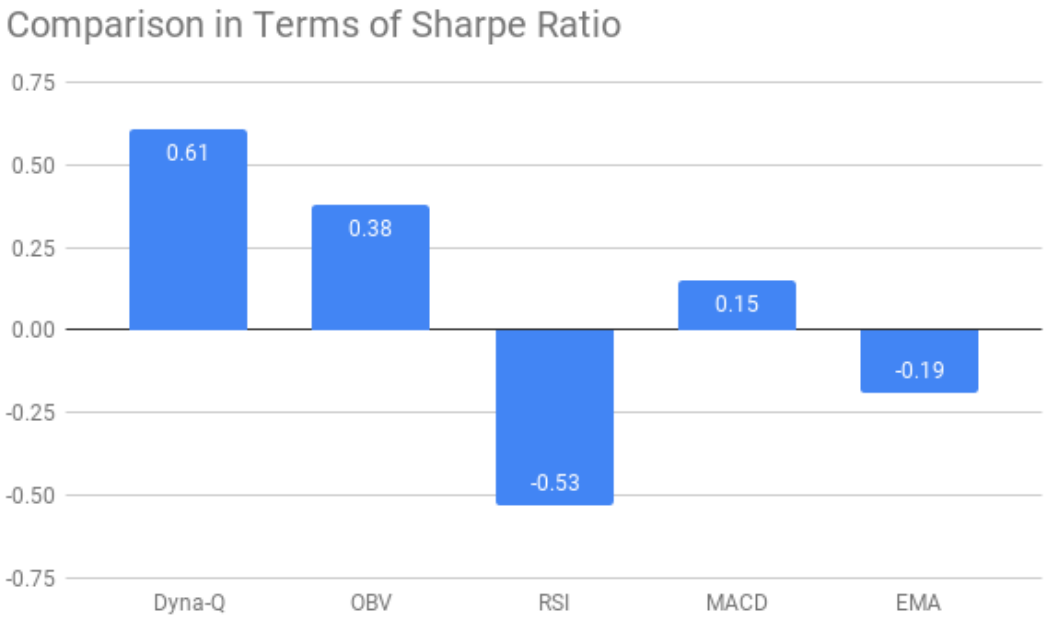
Figure 6.6: Total Returns of RL methods and Technical Indicators



Figures 6.6 and 6.7 have the parameters:

- RL method: Dyna-Q
- **Duration: 15 years**
- **Frequency: Daily**
- **Stock: Uptrend**
- States: Basic
- Reward/Objective Function: Sharpe Ratio
- Learning Rate: 0.1
- Discount Rate: 0.5
- Epsilon: 0.1

Figure 6.7: Sharpe Ratio of RL methods and Technical Indicators



- Hallucinations: 10
- History: 100 datapoints

In terms of total returns, the RL method Dyna-Q outperforms the technical indicators with the trading parameters of 15 years duration, daily trading, and an uptrend stock. There is a big gap between the performance of Dyna-Q and technical indicators in terms of total returns. Dyna-Q reached a 1549.70% total returns and the highest total returns from among the technical indicators is from OBV which had a performance of 113.23%. As for the Sharpe ratios, Dyna-Q reached a 0.61 and OBV had a 0.38. OBV also had the highest Sharpe ratio among the technical indicators. The results show that Dyna-Q performs better than any of the technical indicators in both in terms of total returns and Sharpe ratios. The Sharpe ratio indicates that Dyna-Q is able to trade considering the risks. It not only maximizes the profit but also maximizes the risk-adjusted reward of trades.

Table 6.17: RL and TI in Terms of Total Returns

RL and TI in Terms of Total Returns					
	Dyna-Q	EMA	MACD	RSI	OBV
WMT	38.74%	-48.58%	3.58%	-79.55%	113.23%
CLF	6529.54%	-118.00%	53.88%	-216.60%	36.22%
ACTG	177.08%	181.52%	-11.86%	0.33 %	-149.48 %
ILMN	3586.50%	50.05%	36.35%	1012.07%	792.28%
CEA	6.44%	-97.34%	-0.57%	-58.45%	-93.39 %
ZNH	39.96%	-240.32 %	0.44%	-35.59 %	-99.27 %
VRTX	1549.70%	-57.45%	-21.05%	-75.62%	1527.41%
BRFS	-68.38%	46.58%	-19.43%	-58.45%	-58.35%
AABA	146.04%	-41.53%	-79.25%	-22.07%	23.67%

Table 6.18: RL and TI in Terms of Sharpe Ratio

RL and TI in Terms of Sharpe Ratio					
	Dyna-Q	EMA	MACD	RSI	OBV
WMT	0.38	-0.19	0.15	-0.53	0.38
CLF	0.22	0.2	0.4	0.54	0.6
ACTG	0	0.14	-0.27	-0.17	0.23
ILMN	0.73	0.34	0.43	0.74	0.84
CEA	0.1	-0.27	-0.12	-0.33	-0.05
ZNH	0.21	0.36	-0.01	-0.27	-0.52
VRTX	0.61	0.09	0.01	0.25	0.59
BRFS	-0.02	0.2	-0.01	-0.33	-0.12
AABA	0.35	0.39	-0.33	0.3	0.41

6.3.1 Comparison Using the Validation Stocks

The performance of RL methods were compared to the technical indicators by getting the mean of the total returns and Sharpe ratio from the 5 simulations. In terms of total returns, Dyna-Q performs either on par with technical indicators or significantly better than technical indicators. The only instance in which the mean of the RL method performed worse than all technical indicators is when the agent traded on the downtrend stock BRFS. The gap between the worst of RL and worst of technical indicators in that instance is only 10%. The performance of RL on the stock CLF is outstanding. Dyna-Q's best performance with CLF reached up to 21,845.20% with it's second best following with a 10,739.30% total returns. It's lowest performance was 12.30%. RL's performance is observed to vary in great range. The main drawback in RL methods compared to technical indicators is that it lacks consistency. In terms of total returns, Dyna-Q performed

better than technical indicators for these 9 stocks. As for the Sharpe ratio, Dyna-Q does not generally outperform the technical indicators. Dyna-Q ranged from having the highest Sharpe ratio up to being the second worst. What is notable is that Dyna-Q had no instance of having the worst Sharpe ratio. Having the Sharpe ratio as the objective function for RL have contributed to this performance. When it was traded on the uptrend stocks such as WMT, ILMN, and VRTX, Dyna-Q consistently competed in the top 2 spots for highest Sharpe ratio. CLF was the stock that Dyna-Q had the highest performance on and yet it's Sharpe ratio was second from the worst. The interesting part of this study is that the RL methods while having Sharpe ratio as the objective function, had it's parameters tuned for total returns instead of Sharpe ratio.

Figure 6.8: RL and TI in Terms of Total Returns

Figure 6.9: RL and TI in Terms of Total Returns [Set 1]

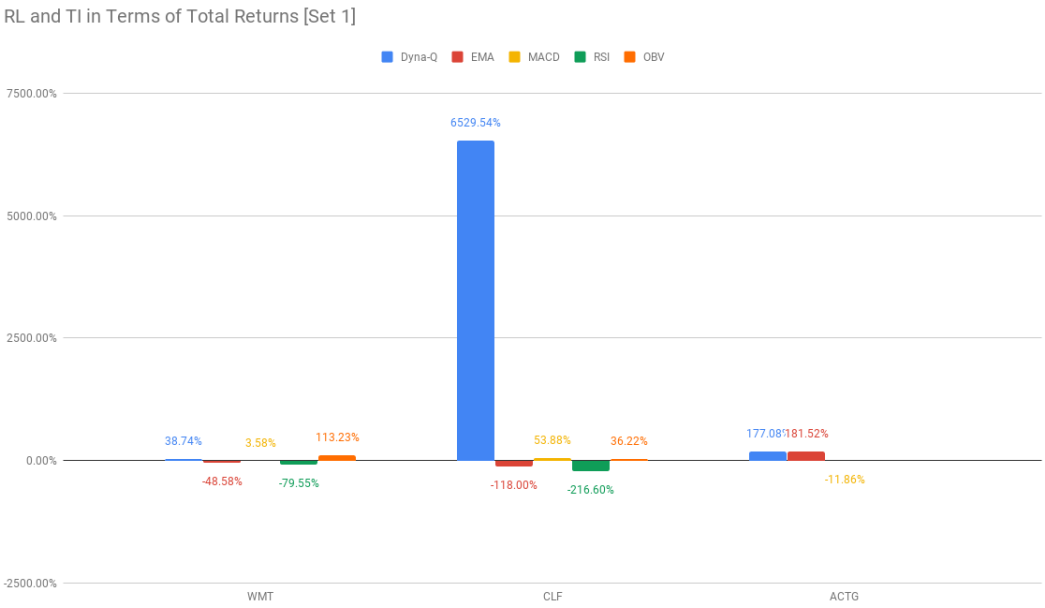


Figure 6.10: RL and TI in Terms of Total Returns [Set 3]

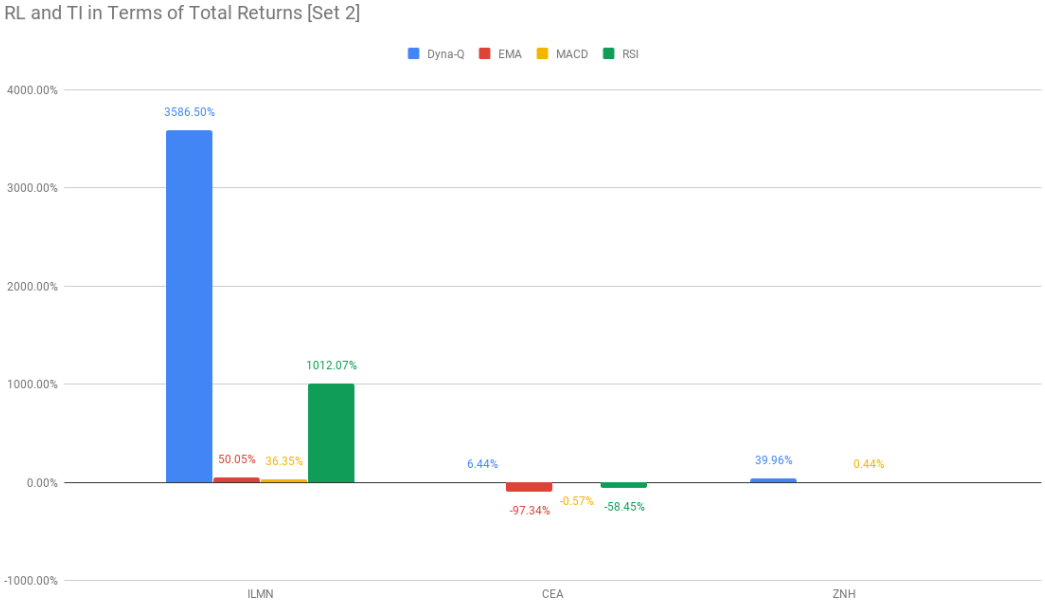


Figure 6.11: RL and TI in Terms of Total Returns [Set 3]

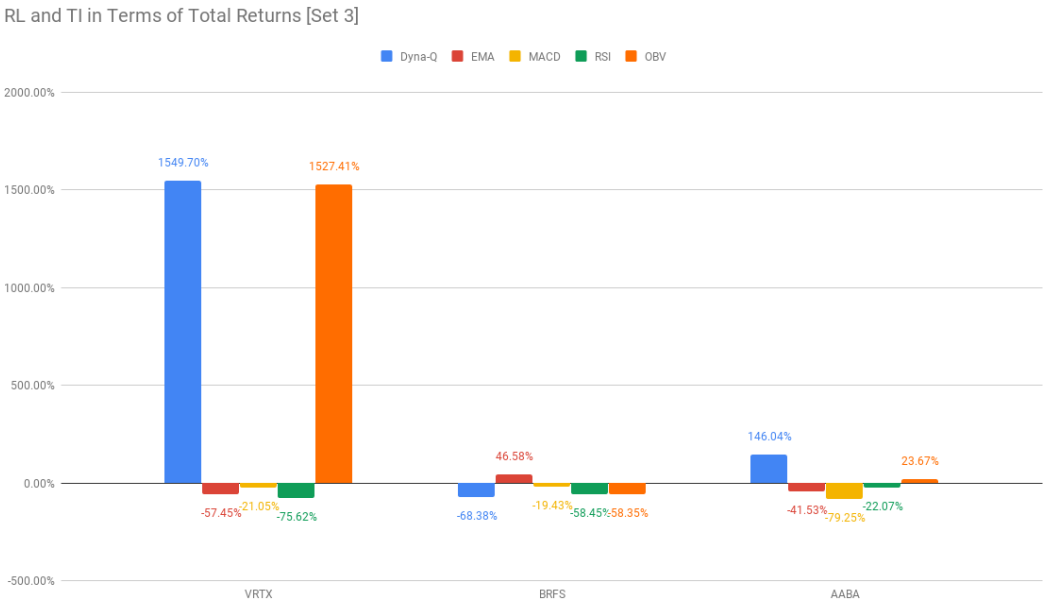


Figure 6.12: RL and TI in Terms of Sharpe Ratio

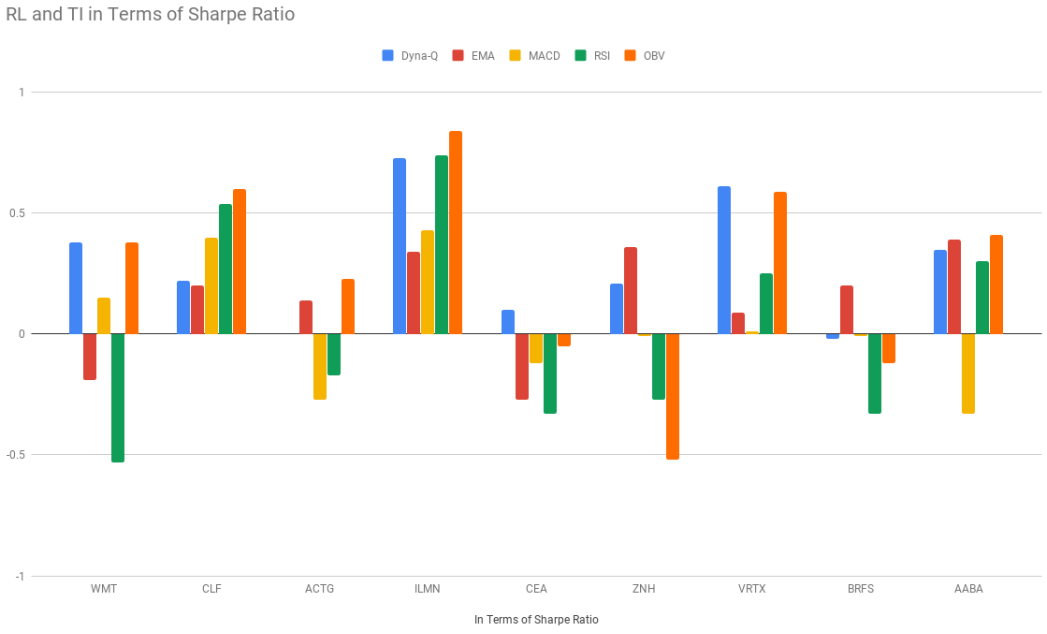


Figure 6.13: RL and TI in Terms of Sharpe Ratio [Set 1]

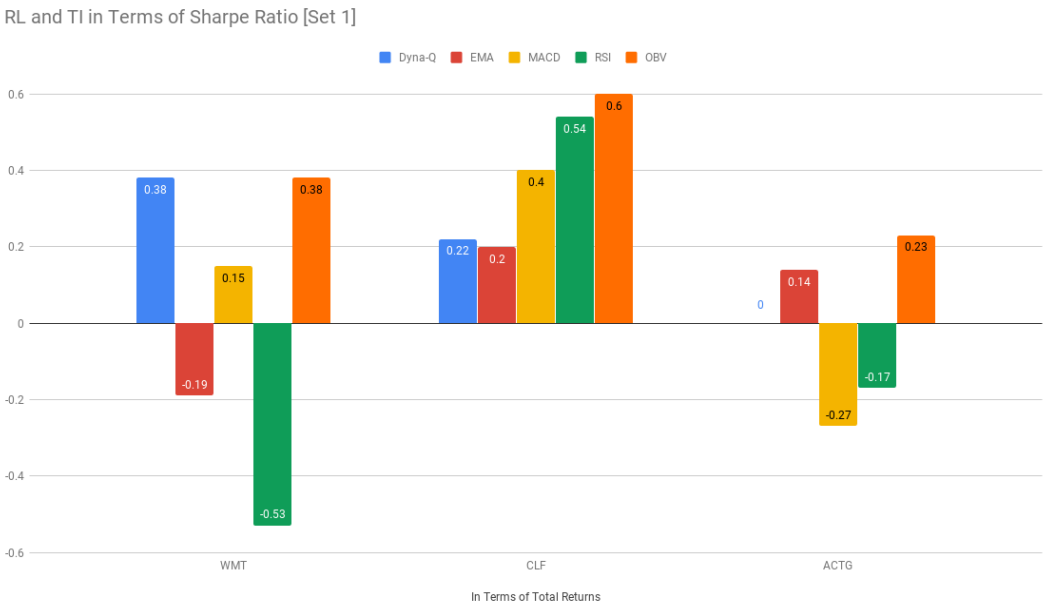


Figure 6.14: RL and TI in Terms of Sharpe Ratio [Set 2]

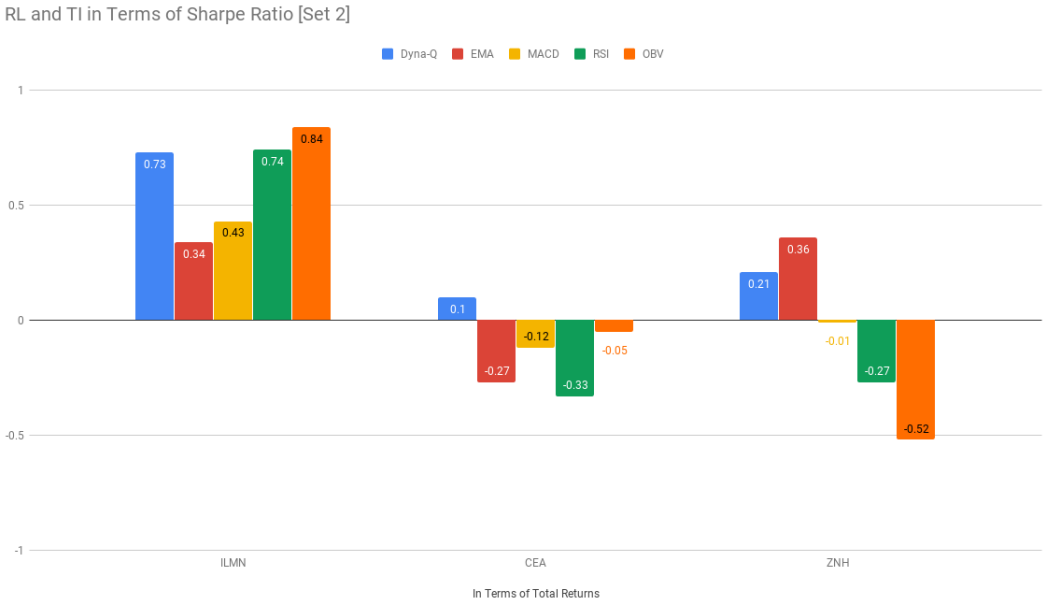
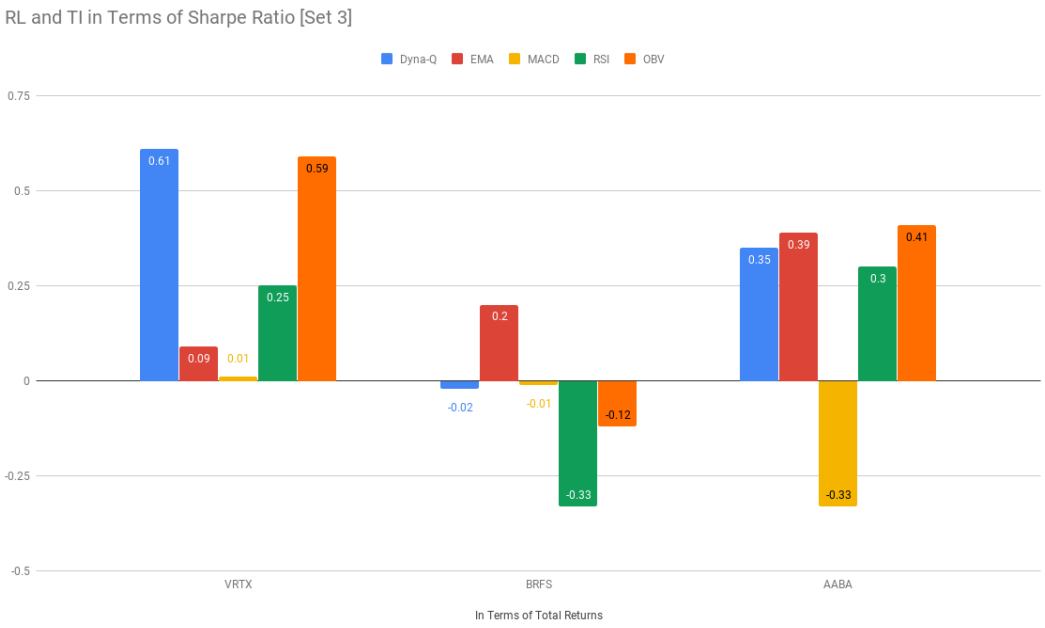


Figure 6.15: RL and TI in Terms of Sharpe Ratio [Set 3]



6.3.2 Total Returns and Position

Table 6.19: RL and TI Total Returns and Position Table

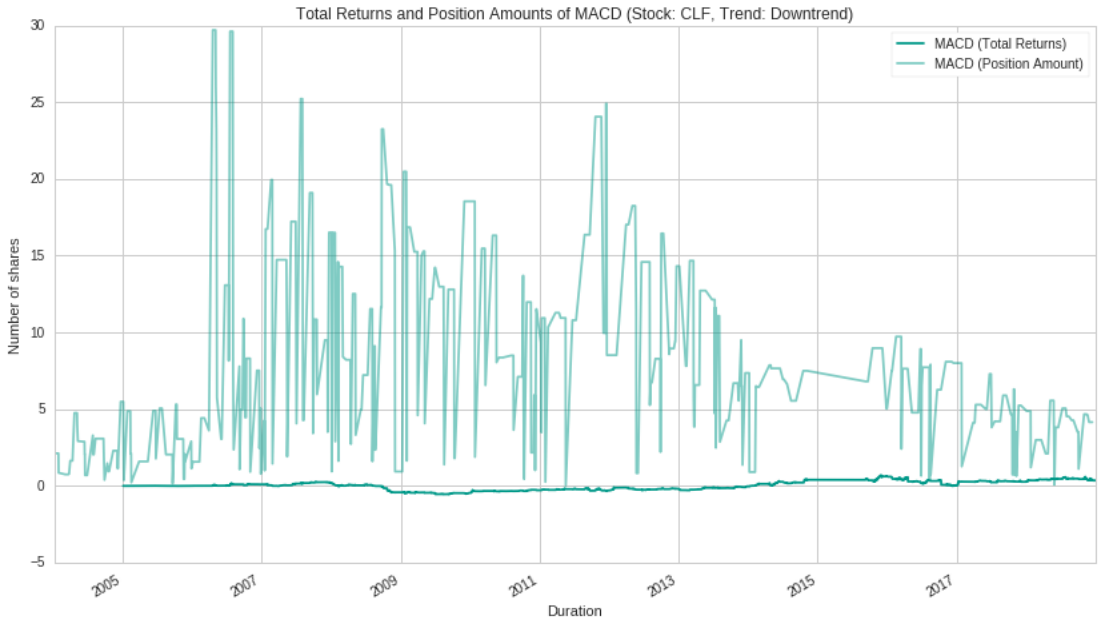
	Algorithm	Stock	Total Returns
RL Highest Total Returns	Dyna-Q	CLF	21845.20%
TI Highest Total Returns on CLF	MACD	CLF	53.88%
RL Highest Total Returns on VRTX	Dyna-Q	VRTX	1542.50%
TI Highest Total Returns	OBV	VRTX	1527.41%

The following results used for analysis were retrieved from the previous results. Their trading conditions have the parameters of trading duration of 15 years and a trading frequency of daily trading.

Position is defined as the number of stocks being held by the agent. This feature is used as a proxy for the trading actions taken by RL methods and technical indicators. The highest performance achieved by RL is Dyna-Q on the downtrend stock CLF with a total returns of 21845.20%. To compare to this, we selected the highest performing technical indicator on CLF which is MACD with a total returns of 53.88%. The reverse was also done for the technical indicators side. The highest performance achieved by the technical indicators is OBV on the uptrend stock VRTX with a total returns of 1527.41%. To this we compared a Dyna-Q performance that was similar in total returns to see if Dyna-Q took the same actions as OBV, to explain the similarity in results. The selected performance from Dyna-Q had a total returns of 1542.50%.

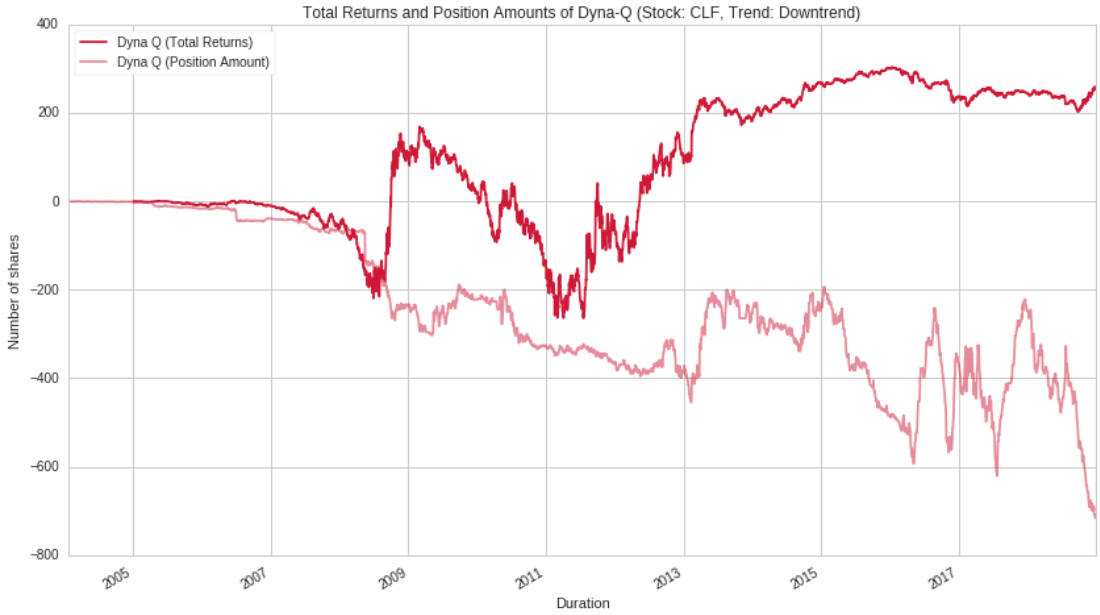
Total Returns and Position on the CLF Stock

Figure 6.16: Total Returns and Position Amounts of MACD (Stock-CLF)



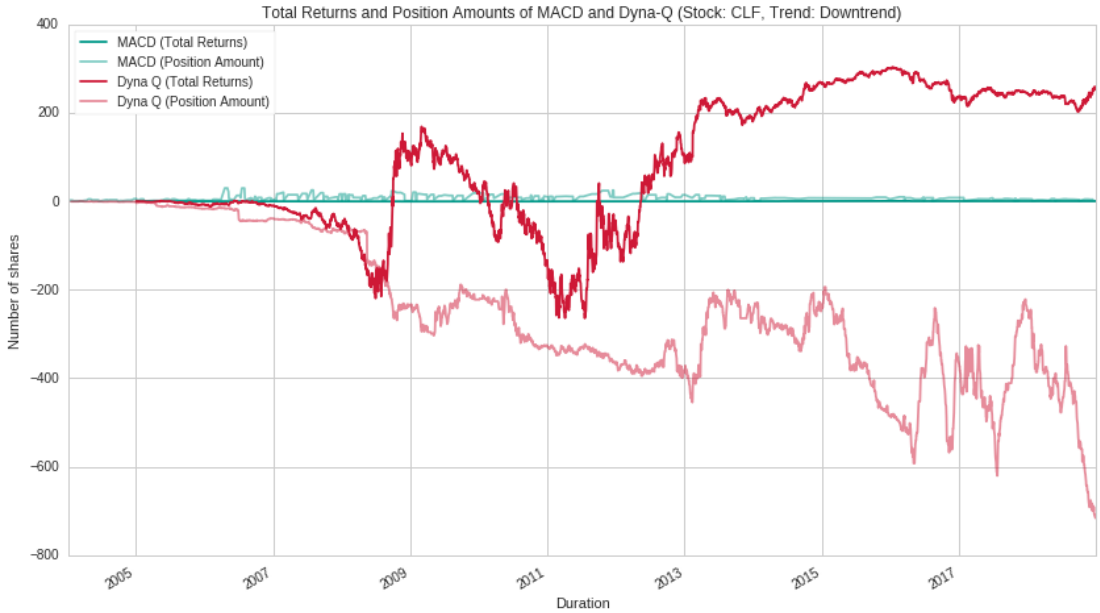
In figure 6.16 shows the total returns and position amount of MACD. Around 2006 until around 2011, the algorithm had a consistent buy and sell decisions. After 2011, the algorithm decisions are not as abrupt. With its actions, it resulted a minimum movement of positive total returns.

Figure 6.17: Total Returns and Position Amounts of Dyna-Q (Stock-CLF)



In figure 6.17 shows the total returns and position amounts of Dyna Q. The algorithm started selling shares with a minimum amount then gradually increasing its amount of sales and from time to time buys shares. With its actions, the algorithm resulted with positive total returns. In Quantopian, short trades are not allowed. When the algorithm runs out of money, it can still buy by borrowing money. The negative number of shares means that the algorithm bought stocks with borrowed money.

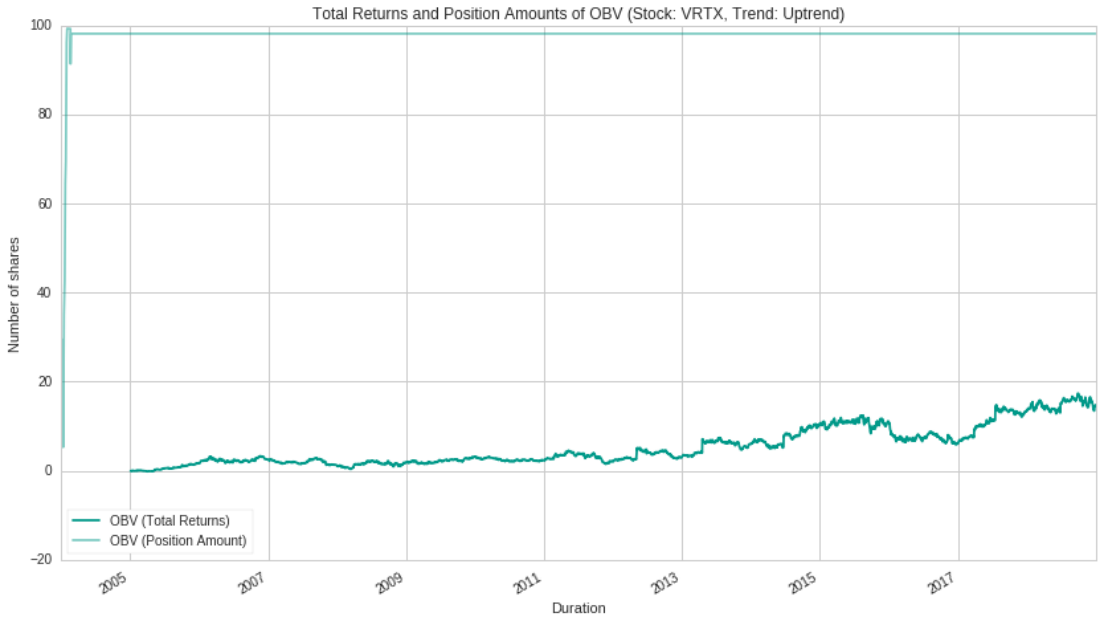
Figure 6.18: Total Returns and Position Amounts of MACD and Dyna-Q (Stock-CLF)



In figure 6.18 shows the combined results of figure 6.17 and figure 6.16. In comparison, Dyna Q resulted with a greater scale of total returns compared to MACD.

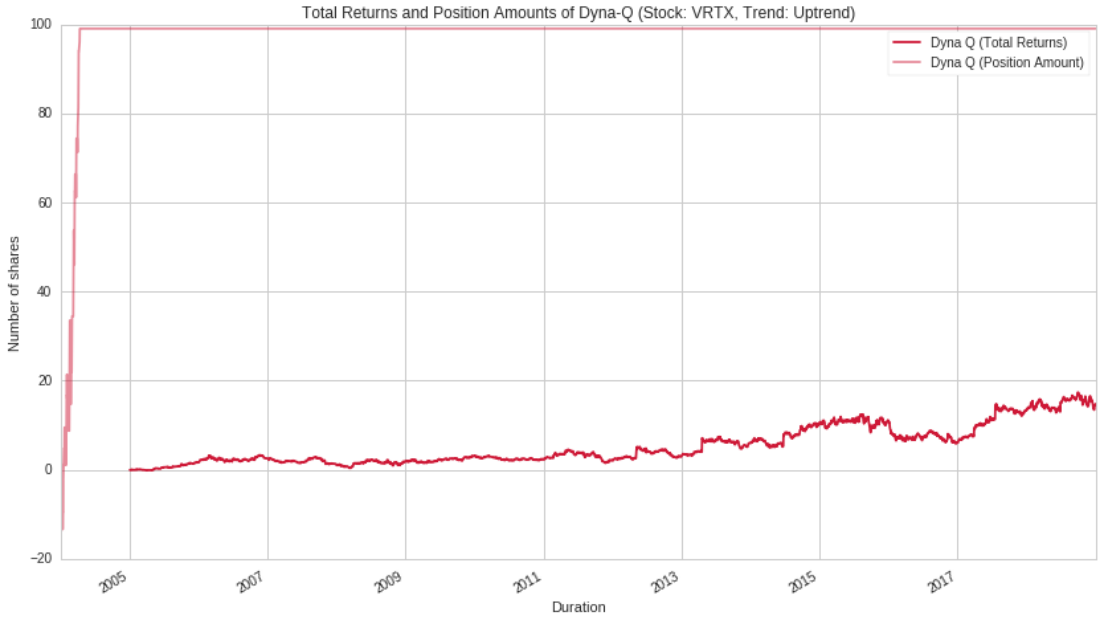
Total Returns and Position on the VRTX Stock

Figure 6.19: Total Returns and Position Amounts of OBV (Stock-VRTX)



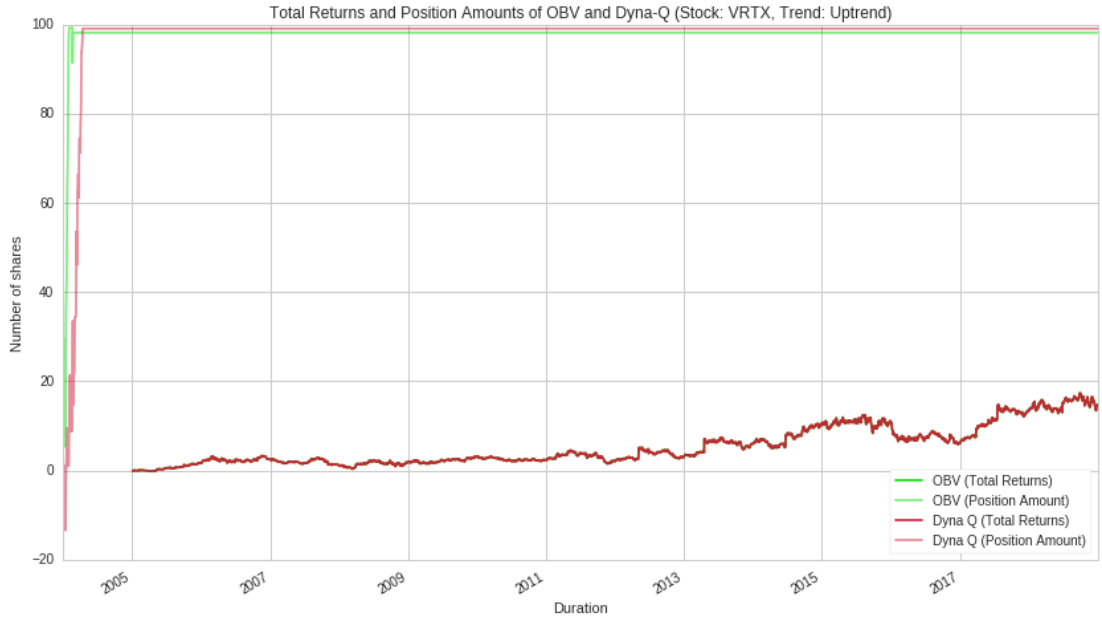
In figure 6.19 shows the total returns and position amount of OBV, while using the stock VRTX as basis. The algorithm started with buying shares. It then sold shares from some time and held it shares throughout the period of time. It resulted a positive total returns.

Figure 6.20: Total Returns and Position Amounts of Dyna-Q (Stock-VRTX)



In figure 6.20 shows the total returns and position amount of Dyna Q while using the stock VRTX. The position amount started with a negative movement which means the algorithm sold shares. It is then followed by a positive movement which means the algorithm bought shares. It then had a constant movement from a period of time of 2004 which means the algorithm held its shares. With those actions, the algorithm resulted in a positive total returns.

Figure 6.21: Total Returns and Position Amounts of OBV and Dyna-Q (Stock-VRTX)



In figure 6.21 shows the combined results of figure 6.17 and figure 6.19. OBV and Dyna-Q have very similar actions and consequently, total returns. We can see that at the start, Dyna-Q were doing some exploration. But as time progresses Dyna-Q seems to have converged to the same strategy as OBV: holding onto the shares and doing nothing for a long time period. This raises the question of: if Dyna-Q converges to the OBV's strategy, does that indicate that OBV's strategy is the optimal strategy for this stock? Such new questions gained from these comparisons are reserved for future works.

Chapter 7

Conclusion and Recommendations

A comparative analysis between technical indicators and RL methods were conducted under a stock trading environment. The technical indicators served as benchmarks for the performance of RL methods. The results of the study affirm that RL methods are capable of learning the optimal policy in a stock trading environment. For the RL methods, the testing was done in phases. With each phase, the best performing parameters are identified and used for the succeeding test phases. The objective function that performed the best is the Sharpe ratio. It performed the best both in terms of total returns and Sharpe ratio. Adding more state features derived from technical indicators worsens performance because it increases the features it needs to learn and slows down the learning of the agent. Dyna-Q learning outperforms Q-learning because of the added hallucinations. The best setup of hyperparameters was a learning rate of 0.1, a discount rate of 0.5, and an epsilon of 0.1. For Dyna-Q, 10 hallucinations were found to be the best. When Dyna-Q was compared to the technical indicators under the parameters Dyna-Q performed best on, Dyna-Q significantly outperformed the technical indicators. When the Dyna-Q was also tested on other stocks, performances were varying. Dyna-Q did not always achieve the highest total returns or Sharpe ratio when compared to the technical indicators. The main drawback of Dyna-Q is that it does not perform consistently. Dyna-Q's total returns and Sharpe ratio often varies greatly with each run of the simulation. The results of this study show that RL methods have the potential to outperform technical indicators in terms of total returns and Sharpe ratio.

The main limitations of this study is in the RL methods and implementation. Performance could be further improved with value function approximation and

the use of deep learning methods. This would lead to the RL methods to learn faster and learn better policies than the current performance. The RL methods used, Q-learning and Dyna-Q learning, were used in a tabular way and is limited to discrete states. By utilizing other RL methods, the states and actions can be expanded to be continuous. Finally, all of the results in RL were achieved by having the RL methods start from scratch. In other words, no training was done beforehand. Given the limited time window for RL to learn the optimal policy, this is a grave disadvantage to RL methods. This disadvantage might be solved by applying transfer learning. Through transfer learning, the RL methods can significantly decrease the learning time needed to converge to the optimal policy. Observing the positions taken by RL methods may serve as indicators to understand what the agent has learned. Positions in trading can be used for future work in the transparency and explainability of RL methods. Lastly, RL methods have an element of stochasticity and the accuracy of the results can be further improved with more testing.

Appendix A

Research Ethics Documents

DE LA SALLE UNIVERSITY
General Research Ethics Checklist

This checklist is to ensure that the research conducted by the faculty members and students of De La Salle University is carried out according to the guiding principles outlined in the Code of Research Ethics of the University. The investigator is advised to refer to the De La Salle University Code of Research Ethics and Guide to Responsible Conduct of Research before completing this checklist. Statements pertinent to ethical issues in research should be addressed below. The checklist will help the researchers and evaluators determine whether procedures should be undertaken during the course of the research to maintain ethical standards. The University's Guide to the Responsible Conduct of Research provides details on these appropriate procedures.

Details of the Research	
Students	Willian Dominique Aboy Mariel Luis Jose Mikhael Uriel Promentilla Mike Jaren Yap
Thesis Adviser	Duke Danielle Delos Santos
Department	Software Technology Department
Title of the Research	Comparative Analysis Between Technical Indicators and Reinforcement Learning Methods for Algorithmic Trading
Term(s) and Academic year in which research is to be conducted	Term 1-3 in the Academic Year 2018-2019

This checklist must be completed AFTER the* De La Salle University Code of Ethics *has been read and BEFORE gathering data.

Questions	**Yes**	**No**
1. Does your research involve human participants (this includes new data gathered or using pre-existing data)? If your answer is **yes**, please answer **Checklist A (Human Participants).**		✔
2. Does your research involve animals (non-human subjects)? If your answer is **yes**, please answer **Checklist B (Animal Subjects).**		✔
3. Does your research involve Wildlife? If your answer is **yes**, please answer **Checklist C (Wildlife)**.		✔
4. Does your research involve microorganisms that are infectious, disease causing or harmful to health? If your answer is **yes**, please answer **Checklist D (Infectious Agents).**		✔

5. Does your research involve toxic/chemicals/ substances/materials? If your answer is **yes**, please answer **Checklist E (Toxic Agents).**		✔

Research with Ethical Issues to address:

If you have a YES answer to any of the above categories, you will be required to complete a detailed checklist for that particular category. A YES answer does not mean the disapproval of your research proposal. By providing you with a more detailed checklist, we ensure that the ethical concerns are identified so these can be addressed in adherence to the University Code of Ethics.

Declaration of Conflict of Interest

[✔] I do not have a conflict of interest in any form (personal, financial, proprietary, or professional) with the sponsor/grant-giving organization, the study, the co-investigators/personnel, or the site.

[ ] I have a personal/family or professional interest in the results of the study (family members who are co-proponents or personnel in the study, membership in relevant professional associations/organizations).

Please describe the personal/family or professional interest:

[ ] I have propriety interest vested in this proposal (with the intent to apply for a patent, trademark, copyright, or license)

Please describe propriety interest:

[ ] I have significant financial interest vested in this proposal (remuneration that exceeds P250,000.00 each year or equity interest in the form of stock, stock options or other ownership interests).

Please describe financial interest:

RESEARCH ETHICS CLEARANCE FORM For Thesis	
Names of student researcher/s :	*William Dominique Aboy* *Mariel Luis* *Jose Mikhael Uriel Promentilla* *Mike Jaren Yap*
College:	College of Computer Studies
Department:	Software Technology Department
Research Title:	Comparative Analysis Between Technical Indicators and Reinforcement Learning Methods for Algorithmic Trading
Course:	BS Computer Science with specialization in Software Technology
Expected duration of project:	from: 2018 to: 2019
Ethical considerations	

To the best of our knowledge, the ethical issues listed above have been addressed in the research.

Duke Danielle Delos Santos

Name and signature of adviser/mentor

Date:

Name and signature of panelist

Date:

Name and signature of panelist

Date:

[1]The same form can be used for the reports of completed projects. The appropriate heading need only be used.

Appendix B

Turnitin Certificate

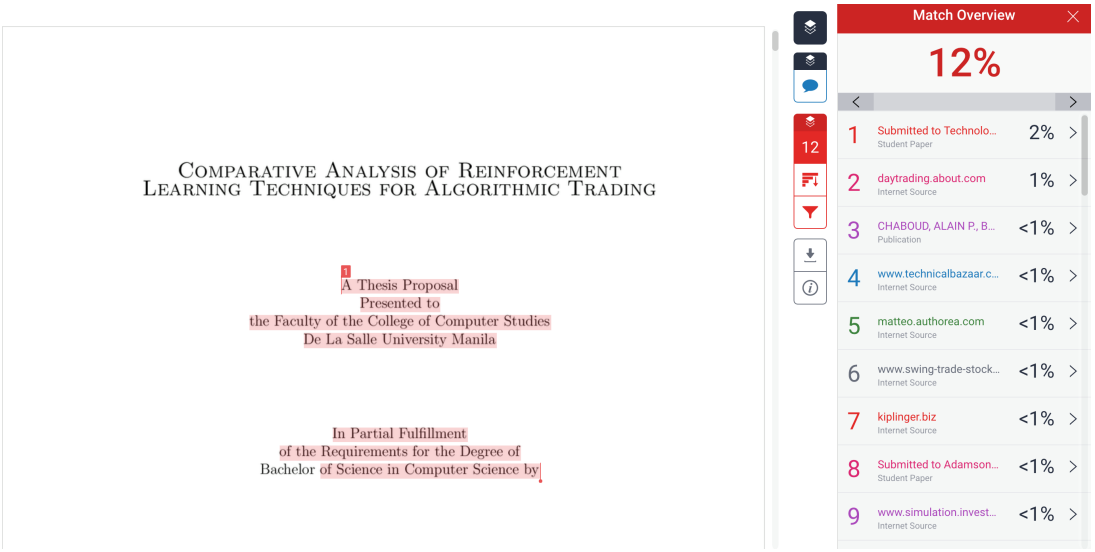


Figure B.1: Turnitin Certificate

Appendix C

Thesis RL and Trading Algorithm Test Cases

Algorithm	Duration	Frequency	Stock	States	Reward / Objective Function	Learning Rate	Discount Rate	Epsilon
Objective Functio								
Q-learning	15 years	Minutely	Oscillating	Basics	Profit	0.1	0.99	0.1
Q-learning	15 years	Minutely	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Minutely	Oscillating	Basics	Net Worth	0.1	0.99	0.1
Q-learning	15 years	Minutely	Oscillating	Basics	Total Returns	0.1	0.99	0.1
Aggregation								
Trading Paramet								
Q-learning	6 months	Minutely	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	6 months	Minutely	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	6 months	Minutely	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	6 months	Hourly	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	6 months	Hourly	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	6 months	Hourly	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	6 months	Daily	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	6 months	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	6 months	Daily	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Minutely	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Minutely	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Minutely	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Hourly	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Hourly	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Hourly	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Daily	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	5 years	Daily	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1

Hallucinations	History (Datapoints)	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)
N/A	10000 (Minutely)	-48940.00%	0.32	-2911.10%	0.25	-170270.00%	0.3	-48955.90%
N/A	10000 (Minutely)	20897.40%	0.81	4140.30%	0.8	36327.90%	0.14	5007.10%
N/A	10000 (Minutely)	-10301.50%	-0.29	374.50%	0.46	3134.00%	0.75	-129.60%
N/A	10000 (Minutely)	2334.90%	0.71	-6712.70%	-0.23	-16494.00%	-0.22	20130.60%
N/A	10000 (Minutely)	247.80%	1.86	-644.50%	-0.97	-62.70%	-0.66	-132.60%
N/A	10000 (Minutely)	109.00%	1.13	-36.40%	1.41	116.00%	1.62	-9.20%
N/A	10000 (Minutely)	4.00%	0.42	5.20%	0.42	-12.80%	-12.80%	-18.30%
N/A	1000 (Hourly)	8.10%	0.53	2.80%	0.32	626.50%	2.09	-30.80%
N/A	1000 (Hourly)	9.00%	0.59	-68.80%	-0.12	8.00%	0.54	10.10%
N/A	1000 (Hourly)	5.50%	0.43	-9.10%	-0.13	11.20%	0.62	11.90%
N/A	100 (Daily)	-52.80%	-1.06	240.40%	-1.21	504.00%	1.36	2.50%
N/A	100 (Daily)	11.20%	0.68	-66.80%	1.79	15.60%	0.87	15.60%
N/A	100 (Daily)	6.30%	0.66	-0.50%	0.04	15.20%	1.91	-23.30%
N/A	10000 (Minutely)	936.60%	1.17	205.50%	0.77	-15193.30%	-0.39	-2198.10%
N/A	10000 (Minutely)	-2965.20%	-0.19	125.70%	0.58	-1906.00%	0.48	153.30%
N/A	10000 (Minutely)	-65.50%	-0.2	948.20%	0.94	52.20%	0.44	495.40%
N/A	1000 (Hourly)	138.10%	0.65	176.60%	0.72	-14358.60%	0.05	114.20%
N/A	1000 (Hourly)	-5351.70%	0.12	96.00%	0.52	144.90%	0.62	112.20%
N/A	1000 (Hourly)	2446.50%	-0.26	-102.30%	0.55	-69.10%	-0.43	-2.60%
N/A	100 (Daily)	168.90%	0.71	141.30%	0.65	176.20%	0.72	-9439.50%
N/A	100 (Daily)	123.90%	0.58	117.00%	0.57	84.80%	0.49	121.10%
N/A	100 (Daily)	-65.20%	-0.37	-65.90%	-0.38	-69.00%	-0.43	-74.40%

Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Mean (Total Returns)	Standard Devation (Total Returns)	Mean [Sharpe Ratio]	Standard Deviation [Sharpe Ratio]
0.09	-10421.90%	0.51	-56299.78%	671.796863	0.29	0.1507647174
0.82	-64474.40%	-0.07	379.66%	385.8386426	0.50	0.4309872388
-0.04	4857.80%	-0.2	-412.96%	58.91634868	0.14	0.4492549388
0.84	-1165.90%	-0.04	-381.42%	134.9478703	0.21	0.5215074304
			379.66%	58.91634868	0.50	0.1507647174
			379.66%	58.91634868	0.50	0.1507647174
			-381.42%	134.9478703	0.29	0.4309872388
			-412.96%	385.8386426	0.21	0.4492549388
1.87	96.70%	-1.47	-99.06%	3.385947474	0.13	1.613576772
0.2	-431.20%	1.15	-50.36%	2.236064579	1.10	0.5431114066
0.23	5.80%	0.74	-3.22%	0.1144080417	0.34	0.3177495869
1.32	7.70%	0.52	122.86%	2.820065301	0.96	0.740695619
0.63	39.40%	0.22	-0.46%	0.4040962757	0.27	0.3191708007
0.72	-20.10%	-0.04	10.58%	0.3031166442	0.32	0.3854218468
0.3	7.40%	0.5	140.30%	2.32506215	-0.02	1.09257494
0.87	-17.30%	-0.52	-8.34%	0.3544683343	0.74	0.8257542007
-1.27	-5.10%	-0.37	-25.86%	0.5830774391	0.19	1.188162447
-0.1	615.90%	-0.09	-3126.68%	68.57225241	0.27	0.6637168071
0.63	8320%	0.24	745.52%	44.41413629	0.35	0.336110101
0.9	364.60%	0.79	358.98%	4.000373258	0.57	0.4753735373
0.59	1475.10%	0.27	-2490.92%	66.59301813	0.46	0.2849210417
0.56	125.40%	0.58	-974.64%	24.46916593	0.48	0.204450483
0.22	-69.80%	-0.44	440.54%	11.21951836	-0.07	0.4390558051
0.27	133.40%	0.64	-1763.94%	42.90806263	0.60	0.1867351065
0.58	121.00%	0.57	113.56%	0.1626416306	0.56	0.03834057903
-0.53	-61.70%	-0.6	-67.24%	0.04770010482	-0.46	0.09984988733

Algorithm	Duration	Frequency	Stock	States	Reward / Objective Function	Learning Rate	Discount Rate	Epsilon
Q-learning	15 years	Minutely	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Minutely	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Minutely	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Hourly	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Hourly	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Hourly	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Daily	Oscillating	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Daily	Downtrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Aggregation								
State Features								
Q-learning	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Daily	Uptrend	Basics + Indicato	Sharpe Ratio	0.1	0.99	0.1
Aggregation								
RL Method								
Random Policy	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Q-learning	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Aggregation								
Hyperparameters								
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	0.5	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	0.5	0.1

Hallucinations	History (Datapoints)	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)
N/A	10000 (Minutely)	20897.40%	0.81	4140.30%	0.8	36327.90%	0.14	5007.10%
N/A	10000 (Minutely)	1273.70%	0.59	-24783.70%	0.12	25524.50%	-0.25	1071.00%
N/A	10000 (Minutely)	14.20%	0.22	-82.70%	0.31	22.00%	0.23	-7.50%
N/A	1000 (Hourly)	4573.50%	0.81	-13309.50%	-0.08	-7768.10%	-0.12	-14335.90%
N/A	1000 (Hourly)	1439.20%	0.6	1383.40%	0.59	-8443.30%	0.05	1748.40%
N/A	1000 (Hourly)	-66.60%	-0.02	-46.40%	0.07	-18.80%	0.15	-131.60%
N/A	100 (Daily)	1701.20%	0.67	579.60%	0.52	3119.10%	0.75	-123.90%
N/A	100 (Daily)	574.80%	0.49	661.60%	0.51	1349.50%	0.59	1368.30%
N/A	100 (Daily)	-99.00%	-0.05	-48.60%	0.06	-48.80%	-0.01	383.20%
N/A	100 (Daily)	574.80%	0.49	661.60%	0.51	1349.50%	0.59	1368.30%
N/A	100 (Daily)	1335.60%	0.59	1350.40%	0.59	-72.70%	0.37	1366.10%
N/A	100 (Daily)	-1284.10%	-0.12	1342.50%	0.59	-105.60%	0.23	-945.70%
N/A	100 (Daily)	574.80%	0.49	661.60%	0.51	1349.50%	0.59	1368.30%
10	100 (Daily)	1526%	0.61	1478.60%	0.6	230.90%	0.4	1326.40%
10	100 (Daily)	-462.40%	0.21	1559.30%	0.61	801.00%	0.53	1328.00%
10	100 (Daily)	1188.10%	0.58	-1877.60%	0.31	1396.70%	0.6	1712.70%

Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Mean (Total Returns)	Standard Devation (Total Returns)	Mean [Sharpe Ratio]	Standard Deviation [Sharpe Ratio]
0.82	-64474.40%	-0.07	379.66%	385.8386426	0.50	0.4309872388
0.55	-5266.80%	0.32	-436.26%	179.9506048	0.27	0.3450072463
0.18	-21.60%	0.16	-15.12%	0.4154716597	0.22	0.05787918451
-0.08	-7356.80%	0.07	-7639.36%	75.2071305	0.12	0.3924920381
0.62	1310.20%	0.59	-512.42%	44.36638998	0.49	0.2462722071
0.24	6.70%	0.2	-12.62%	0.5008624562	0.13	0.1042592922
0.22	2010.80%	0.7	1059.60%	13.56546794	0.57	0.2146392322
0.59	1442.00%	0.6	1079.24%	4.234010664	0.56	0.05176871642
0.1	-176.90%	0.13	1.98%	2.194656283	0.05	0.07503332593
			1079.24%	0.04770010482	1.10	0.03834057903
	-76.3936		1079.24%	0.04770010482	1.10	0.03834057903
	-31.2668		1059.60%	0.1144080417	0.96	0.05176871642
	-24.9092		745.52%	0.1626416306	0.74	0.05787918451
0.59	1442.00%	0.6	1079.24%	4.234010664	0.56	0.05176871642
0.59	1025.70%	0.56	1001.02%	7.006295562	0.54	0.09591663047
			1079.24%	4.234010664	0.56	0.05176871642
-0.26	-215.70%	0.17	-241.72%	10.18548412	0.12	0.3307113545
0.59	1442.00%	0.6	1079.24%	4.234010664	0.56	0.05176871642
0.59	1403.40%	0.59	1193.12%	5.432210204	0.56	0.08871302046
			1193.12%	4.234010664	0.56	0.05176871642
0.59	-2428.70%	0.26	159.44%	16.44832315	0.44	0.1902629759
0.62	1437.40%	0.6	771.46%	14.92601894	0.54	0.1304607221

Algorithm	Duration	Frequency	Stock	States	Reward / Objective Function	Learning Rate	Discount Rate	Epsilon
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	0.5	0.5
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	0.99	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	0.99	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	0.99	0.5
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	1	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	1	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.01	1	0.5
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.5	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.5	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.5	0.5
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.99	0.5
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	1	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	1	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	1	0.5
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	0.5	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	0.5	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	0.5	0.5
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	0.99	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	0.99	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	0.99	0.5
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	1	0.01
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	1	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.5	1	0.5
Aggregation								

Hallucinations	History (Datapoints)	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)
10	100 (Daily)	709.70%	0.51	-2834.00%	0.11	1031.80%	0.56	-19347.50%
10	100 (Daily)	-3076.00%	-0.09	2605.40%	0.66	1585.50%	0.61	1347.10%
10	100 (Daily)	1521.20%	0.61	1385.20%	0.59	1521.60%	0.61	1402.50%
10	100 (Daily)	-4770.60%	0.05	690.70%	0.51	1154.10%	0.57	1394.30%
10	100 (Daily)	-17574.90%	-0.4	955.60%	0.55	1652.60%	0.62	-2046.70%
10	100 (Daily)	1370.20%	0.59	918.60%	0.54	467.50%	0.47	-2009.90%
10	100 (Daily)	-399.60%	-0.17	1376.80%	0.59	2654.30%	0.48	-4652.80%
10	100 (Daily)	935.10%	0.55	676.30%	0.51	1518.80%	0.6	1477.90%
10	100 (Daily)	1396.30%	0.59	1542.50%	0.61	1771.30%	0.63	1377.80%
10	100 (Daily)	-3667.60%	-0.24	1652.40%	0.62	1370.60%	0.59	-499.00%
10	100 (Daily)	1529.60%	0.61	1369.30%	0.59	1361.90%	0.59	1641.90%
10	100 (Daily)	727.80%	0.52	-3876.30%	-0.28	1451.80%	0.6	-30221.90%
10	100 (Daily)	1429.30%	0.6	1436.40%	0.6	1363.70%	0.59	1405.10%
10	100 (Daily)	1739.00%	0.62	1345.90%	0.59	1392.10%	0.6	1387.00%
10	100 (Daily)	665.40%	0.51	-157.80%	-0.23	1410.00%	0.6	1409.30%
10	100 (Daily)	1701.00%	0.62	-1474.80%	-0.26	1146.00%	0.57	1302.10%
10	100 (Daily)	1705.30%	0.62	1353.50%	0.59	1360.20%	0.59	-6.20%
10	100 (Daily)	708.30%	0.52	1428.40%	0.6	189.80%	0.44	871.40%
10	100 (Daily)	784.70%	0.53	1479.60%	0.6	-198.90%	-0.2	1607.10%
10	100 (Daily)	1382.50%	0.59	1437.90%	0.6	936.10%	0.55	1130.30%
10	100 (Daily)	1404.70%	0.59	1386.60%	0.6	1380.60%	0.6	1098.90%
10	100 (Daily)	650.40%	0.5	1764.40%	0.62	1382.80%	0.59	1380.80%
10	100 (Daily)	-2427.50%	0.24	1433.50%	0.6	6816.30%	0.74	1347.60%
10	100 (Daily)	-260.80%	-0.02	979.90%	0.55	1562.60%	0.61	1397.50%
10	100 (Daily)	-4212.90%	-0.33	-1858.40%	-0.31	569.50%	0.49	-19094.90%

Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Mean (Total Returns)	Standard Devation (Total Returns)	Mean [Sharpe Ratio]	Standard Deviation [Sharpe Ratio]
0.12	1569.90%	0.61	-3774.02%	88.76570971	0.38	0.2463128092
0.59	1365.00%	0.59	765.40%	22.08642854	0.47	0.3154679065
0.59	1389.70%	0.59	1444.04%	0.7090446389	0.60	0.01095445115
0.6	1303.30%	0.59	-45.64%	26.55178242	0.46	0.2340512764
-0.07	1609.30%	0.61	-3080.82%	82.43830059	0.26	0.4692227616
0.29	1330.90%	0.59	415.46%	14.04260305	0.50	0.1252198067
-0.08	1679.60%	0.62	131.66%	28.93246408	0.29	0.3819293128
0.6	1453.20%	0.6	1212.26%	3.82966791	0.57	0.04086563348
0.59	1660.60%	0.62	1549.70%	1.692175966	0.61	0.01788854382
0.3	1304.10%	0.59	32.10%	22.36335487	0.37	0.3661557046
0.61	-39012.10%	0.02	-6621.88%	181.0706173	0.48	0.2595765783
0.4	1390.00%	0.59	-6105.72%	136.6161936	0.37	0.3698378023
0.59	-1049.80%	0.19	916.94%	10.99808207	0.51	0.1811905075
0.59	300.20%	0.43	1232.84%	5.449603041	0.57	0.07700649323
0.59	1492.80%	0.6	963.94%	7.114131556	0.41	0.3619806625
0.59	1368.40%	0.59	131.66%	28.93246408	0.42	0.3816673945
0.26	1479.50%	0.6	1178.46%	6.773612795	0.53	0.1525450753
0.54	1540.70%	0.61	947.72%	5.523286947	0.54	0.06870225615
0.61	1442.90%	0.6	1023.08%	7.542875526	0.43	0.352519503
0.57	1406.30%	0.59	1258.62%	2.178199532	0.58	0.02
0.57	1429.60%	0.6	1132.30%	5.997369673	0.59	0.01303840481
0.59	1373.50%	0.59	1310.38%	4.049368864	0.58	0.04549725266
0.59	1577.90%	0.61	1212.94%	34.45809235	0.56	0.1868956928
0.6	1490.50%	0.6	1033.94%	7.581847222	0.47	0.273806501
0.26	1368.60%	0.59	-4645.62%	83.68181602	0.14	0.4366921112
			1549.70%	0.7090446389	0.61	0.01095445115
			1549.70%	0.7090446389	0.61	0.01095445115
			1444.04%	1.692175966	0.60	0.01303840481
			1310.38%	2.178199532	0.59	0.01788854382

Algorithm	Duration	Frequency	Stock	States	Reward / Objective Function	Learning Rate	Discount Rate	Epsilon
Hallucinations								
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.5	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.5	0.1
Dyna-Q	15 years	Daily	Uptrend	Basics	Sharpe Ratio	0.1	0.5	0.1
Aggregation								

Hallucinations	History (Datapoints)	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)
10	100 (Daily)	1396.30%	0.59	1542.50%	0.61	1771.30%	0.63	1377.80%
100	100 (Daily)	1611.90%	0.61	1348.50%	0.59	1338.50%	0.59	1601.40%
1000	100 (Daily)	1416.50%	0.6	1465.40%	0.6	1031.40%	0.56	1156.60%

Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Mean (Total Returns)	Standard Devation (Total Returns)	Mean [Sharpe Ratio]	Standard Deviation [Sharpe Ratio]
0.59	1660.60%	0.62	1549.70%	1.692175966	0.61	0.01788854382
0.61	1402.40%	0.59	1460.54%	1.356266677	0.60	0.01095445115
0.57	1369.70%	0.59	1287.92%	1.855851476	0.58	0.01816590212
			1549.70%	1.356266677	60.80%	0.01095445115

Algorithm	Duration	Frequency	Stock	States	Reward / Objective Function	Learning Rate	Discount Rate	Epsilon
Validation for Oth								
Dyna-Q	15 years	Daily	ACTG	Oscillating	Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	VRTX	Uptrend	Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	ILMN	Uptrend	Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	WMT	Uptrend	Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	AABA	Oscillating	Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	ZNH	Oscillating	Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	BRFS	Downtrend	Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	CEA	Downtrend	Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	CLF	Downtrend	Basics	Sharpe Ratio	0.1	0.5
Random Stocks								
Dyna-Q	15 years	Daily	ATML*		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	BAC*		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	DHLI		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	BIIB		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	ES		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	SYY		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	ALB		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	ETFC		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	BLK		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	EZU		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	GPN		Basics	Sharpe Ratio	0.1	0.5
Dyna-Q	15 years	Daily	PALM		Basics	Sharpe Ratio	0.1	0.5

Hallucinations	History (Datapoints)	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Total Returns (%)
0.1	10	100 (Daily)	900.80%	-0.12	213.30%	0.4	-65.40%	0.05
0.1	10	100 (Daily)	1396.30%	0.59	1542.50%	0.61	1771.30%	0.63
0.1	10	100 (Daily)	3660.60%	0.78	7372.90%	0.87	5904.10%	0.85
0.1	10	100 (Daily)	133.90%	0.39	-388.20%	0.28	143.10%	0.4
0.1	10	100 (Daily)	154.10%	0.35	146.80%	0.35	146.80%	0.35
0.1	10	100 (Daily)	45.80%	0.22	117.40%	0.33	126.70%	0.38
0.1	10	100 (Daily)	-179.40%	-0.32	-61.30%	0	-23.00%	0.13
0.1	10	100 (Daily)	-38.90%	-0.02	24.30%	0.22	-1.90%	0.04
0.1	10	100 (Daily)	28.20%	0.38	10739.30%	0.31	21845.20%	-0.3
							14981.00%	
0.1	10	100 (Daily)	445.50%	0.5	461.50%	0.48	26.80%	0.28
0.1	10	100 (Daily)	-8861.50%	0.02	-56.60%	0.13	133.20%	0.42
0.1	10	100 (Daily)	43.50%	0.28	-71.10%	0.28	104.90%	0.33
0.1	10	100 (Daily)	691.10%	0.57	651.40%	0.56	654.60%	0.56
0.1	10	100 (Daily)	340.70%	0.62	-122.80%	0.39	435.60%	0.69
0.1	10	100 (Daily)	1837.30%	0.17	127.40%	0.37	174.50%	0.43
0.1	10	100 (Daily)	689.70%	0.58	1466.80%	0.66	-1745.10%	0.28
0.1	10	100 (Daily)	-69.20%	0.16	-66.30%	0.17	-66.30%	0.17
0.1	10	100 (Daily)	-1417.60%	0.38	612.10%	0.56	-1597.90%	0.31
0.1	10	100 (Daily)	328.20%	0.44	-1266.30%	-0.24	-52.30%	-0.07
0.1	10	100 (Daily)	-100.10%	-0.35	415.10%	0.54	-350.80%	0.28
0.1	10	100 (Daily)	100.90%	0.36	-175.00%	0.11	-34.30%	0.24

Sharpe Ratio	Total Returns (%)	Sharpe Ratio	Mean (Total Returns)	Standard Devation (Total Returns)	Mean [Sharpe Ratio]	Standard Deviation [Sharpe Ratio]	
-66.60%	0.04	-96.70%	-0.35	177.08%	4.237376865	0.00	0.2742808779
1377.80%	0.59	1660.60%	0.62	1549.70%	1.692175966	0.61	0.01788854382
-4980.80%	0.32	5975.70%	0.84	3481.30%	51.97466495	0.73	0.2327444951
-5506.80%							
180.30%	0.43	124.60%	0.38	38.74%	2.396014253	0.38	0.05683308895
135.70%	0.34	146.80%	0.35	146.04%	0.06588095324	0.35	0.004472135955
-38.30%	0.17	-51.80%	-0.07	39.96%	0.8381439614	0.21	0.1755847374
-31.10%	0.11	-47.10%	-0.01	-68.38%	0.6379174712	-0.02	0.1801943395
55.20%	0.24	-6.50%	0.03	6.44%	0.3531724791	0.10	0.1192476415
12.30%	0.36	22.70%	0.36	6529.54%	97.38741273	0.22	0.2929505078
			Uptrend Summar	1689.91%	18.68761839	0.57	0.1024887093
			Oscillating Summ	121.03%	1.713800593	0.19	0.1514459171
			Downtrend Sumr	2155.87%	32.79283423	0.10	0.1974641629
123.20%	0.31	27.90%	0.28	216.98%	2.195027266	0.37	0.1104536102
-288.30%	0.06	6126.80%	-0.36	-589.28%	53.48636943	0.05	0.2794279871
-147.70%	0.28	-419.60%	0.3	-98.00%	2.048236559	0.29	0.0219089023
634.20%	0.56	685.70%	0.57	663.40%	0.2417984698	0.56	0.005477225575
414.50%	0.68	-85.70%	-0.08	196.46%	2.770720899	0.46	0.3253459697
174.50%	0.43	178.30%	0.43	498.40%	7.487624189	0.37	0.1126055061
-4324.20%	0.21	-1907.30%	-0.25	-1164.02%	23.04037134	0.30	0.3601805103
-70.70%	0.15	-70.80%	0.15	-68.66%	0.02245662486	0.16	0.01
1795.30%	-0.42	-10039.10%	0.19	-2129.44%	46.44125265	0.20	0.3736709783
14.10%	0.16	-83.60%	0.29	-211.98%	6.116527585	0.12	0.2731849191
-148.30%	-0.22	-321.10%	-0.03	-101.04%	2.83014694	-0.59	1.508137659
-280.70%	-0.36	162.40%	0.37	-45.34%	1.849797097	0.14	0.3008820367

				10M Starting Capital		10,000 Starting Capital	
Algorithm	**Timeframe / Frequency**	**Type of Stock**	**Duration**	**Total Returns (%)**	**Sharpe Ratio**	**Total Returns (%)**	**Sharpe Ratio**
EMA	Daily	Uptrend	15 years	-48.58%	-0.19	-54.47%	-0.26
EMA	Daily	Downtrend	15 years	-118.00%	0.2	-103.57%	0.16
EMA	Daily	Oscillating	15 years	181.52%	0.14	-151.96%	0.16
EMA	Hourly	Uptrend	15 years	-62.05%	-0.33	-59.14%	-0.3
EMA	Hourly	Downtrend	15 years	-101.43%	0.09	-103.24%	0.17
EMA	Hourly	Oscillating	15 years	43.00%	0.08	-168.02%	0.21
EMA	Minutely	Uptrend	15 years	-56.42%	-0.28	-66.51%	-0.34
EMA	Minutely	Downtrend	15 years	-101.85%	0.09	-102.24%	0.19
EMA	Minutely	Oscillating	15 years	-21.29%	0	-175.52%	0.16
EMA	Daily	Uptrend	5 years	5.65%	0.13	22.58%	0.18
EMA	Daily	Downtrend	5 years	-112.12%	0.2	-121.98%	0.11
EMA	Daily	Oscillating	5 years	73.17%	0.51	26.26%	0.48
EMA	Hourly	Uptrend	5 years	14.49%	0.11	23.87%	0.2
EMA	Hourly	Downtrend	5 years	-138.49%	0.45	-119.67%	0.1
EMA	Hourly	Oscillating	5 years	101.44%	0.51	24.39%	0.49
EMA	Minutely	Uptrend	5 years	18.41%	0.16	23.63%	0.19
EMA	Minutely	Downtrend	5 years	-128.35%	0.12	-114.09%	0.16
EMA	Minutely	Oscillating	5 years	79.57%	0.42	19.08%	0.07
EMA	Daily	Uptrend	6 months	-4.92%	0.22	0.60%	0.58
EMA	Daily	Downtrend	6 months	1.06%	-0.25	14.24%	0.44
EMA	Daily	Oscillating	6 months	0%	NaN	0.00%	NaN
EMA	Hourly	Uptrend	6 months	-0.38%	0.5	0.72%	0.59
EMA	Hourly	Downtrend	6 months	6.24%	0.07	13.47%	0.43
EMA	Hourly	Oscillating	6 months	0%	NaN	0.00%	NaN
EMA	Minutely	Uptrend	6 months	-0.01%	0.51	0.72%	0.59
EMA	Minutely	Downtrend	6 months	7.78%	0.15	11.91%	0.39
EMA	Minutely	Oscillating	6 months	0%	NaN	0.00%	NaN
MACD	Daily	Uptrend	15 years	3.58%	0.15	-7.29%	0.04
MACD	Daily	Downtrend	15 years	53.88%	0.4	168.19%	0.54

				10M Starting Capital		10,000 Starting Capital	
Algorithm	**Timeframe / Frequency**	**Type of Stock**	**Duration**	**Total Returns (%)**	**Sharpe Ratio**	**Total Returns (%)**	**Sharpe Ratio**
MACD	Daily	Oscillating	15 years	-11.86%	-0.27	0.62%	-0.37
MACD	Hourly	Uptrend	15 years	-18.11%	-0.02	-18.55%	-0.01
MACD	Hourly	Downtrend	15 years	219.94%	0.71	276.49%	0.84
MACD	Hourly	Oscillating	15 years	-6.49%	-0.23	20.14%	0.07
MACD	Minutely	Uptrend	15 years	-7.29%	0.04	-86.82%	-0.25
MACD	Minutely	Downtrend	15 years	168.19%	0.54	-41.56%	0.5
MACD	Minutely	Oscillating	15 years	0.62%	-0.37	164.61%	0.11
MACD	Daily	Uptrend	5 years	-21.08%	-0.15	-17.56%	-0.33
MACD	Daily	Downtrend	5 years	52.69%	0.34	118.75%	0.68
MACD	Daily	Oscillating	5 years	-2.57%	-0.46	-58.98%	-0.32
MACD	Hourly	Uptrend	5 years	-12.64%	-0.26	-11.89%	-0.25
MACD	Hourly	Downtrend	5 years	82.63%	0.63	16.55%	0.51
MACD	Hourly	Oscillating	5 years	-5.87%	-0.87	-57.72%	-0.39
MACD	Minutely	Uptrend	5 years	-18.79%	-0.37	-27.60%	-0.42
MACD	Minutely	Downtrend	5 years	62.61%	0.53	-1.17%	0.13
MACD	Minutely	Oscillating	5 years	-5.16%	-1.04	4898.42%	0.99
MACD	Daily	Uptrend	6 months	6.08%	0.89	11.80%	1.22
MACD	Daily	Downtrend	6 months	-10.99%	-0.97	7.68%	1.05
MACD	Daily	Oscillating	6 months	-0.06%	-1.12	-23.10%	-1.83
MACD	Hourly	Uptrend	6 months	12.88%	1.15	13.26%	1.39
MACD	Hourly	Downtrend	6 months	15.42%	1.45	9.30%	1.25
MACD	Hourly	Oscillating	6 months	-0.08%	-1.41	20.75%	1.68
MACD	Minutely	Uptrend	6 months	8.99%	0.99	7.78%	0.91
MACD	Minutely	Downtrend	6 months	16.83%	1.5	4.87%	0.2
MACD	Minutely	Oscillating	6 months	-0.09%	-1.9	247.35%	2.32
RSI	Daily	Uptrend	15 years	-79.55%	-0.53	-70.50%	-0.42
RSI	Daily	Downtrend	15 years	-216.60%	0.54	-65.24%	0.43
RSI	Daily	Oscillating	15 years	0.33 %	-0.17	-97.43%	0.12
RSI	Hourly	Uptrend	15 years	-79.98%	-0.51	-57.49%	-0.26

				10M Starting Capital		10,000 Starting Capital	
Algorithm	**Timeframe / Frequency**	**Type of Stock**	**Duration**	**Total Returns (%)**	**Sharpe Ratio**	**Total Returns (%)**	**Sharpe Ratio**
RSI	Hourly	Downtrend	15 years	-99.79%	0.43	-101.72%	0.51
RSI	Hourly	Oscillating	15 years	268.14%	0.2	-95.40%	0.1
RSI	Minutely	Uptrend	15 years	-68.45%	-0.41	-52.16%	-0.21
RSI	Minutely	Downtrend	15 years	-109.22%	0.32	-100.93%	0.44
RSI	Minutely	Oscillating	15 years	230.34%	0.2	-94.30%	0.15
RSI	Daily	Uptrend	5 years	-31.92%	-0.19	-24.24%	-0.07
RSI	Daily	Downtrend	5 years	-13.32%	0.31	-280.43%	0.36
RSI	Daily	Oscillating	5 years	53.06 %	0.45	-66.88%	-0.23
RSI	Hourly	Uptrend	5 years	-32.62%	-0.22	-7.97%	0.15
RSI	Hourly	Downtrend	5 years	-68.05%	0.42	-92.12%	0.3
RSI	Hourly	Oscillating	5 years	125.23%	0.57	-31.72%	0
RSI	Minutely	Uptrend	5 years	-23.46%	-0.06	-5.05%	0.17
RSI	Minutely	Downtrend	5 years	-141.21%	0.34	-97.89%	-0.1
RSI	Minutely	Oscillating	5 years	98.54%	0.51	-34.57%	0.02
RSI	Daily	Uptrend	6 months	-19.47 %	-1.85	-15.51%	-0.97
RSI	Daily	Downtrend	6 months	-31.53%	-1.12	14.47%	0.52
RSI	Daily	Oscillating	6 months	-0.15 %	-1.9	-29.60%	-1.85
RSI	Hourly	Uptrend	6 months	-17.17%	-0.99	-15.47%	-0.97
RSI	Hourly	Downtrend	6 months	-28.46%	-0.94	22.66%	0.83
RSI	Hourly	Oscillating	6 months	-0.12%	-0.74	-7.67%	-0.42
RSI	Minutely	Uptrend	6 months	-16.45%	-1	-15.53%	-0.97
RSI	Minutely	Downtrend	6 months	-9.65%	-0.34	26.69%	0.86
RSI	Minutely	Oscillating	6 months	-0.72%	-2.05	-3.51%	-0.22
OBV	Daily	Uptrend	15 years	113.23%	0.38	161.45%	0.46
OBV	Daily	Downtrend	15 years	36.22%	0.6	43.03 %	0.59
OBV	Daily	Oscillating	15 years	-149.48 %	0.23	-98.08 %	-0.27
OBV	Hourly	Uptrend	15 years	126.35%	0.43	-34.06%	-0.01
OBV	Hourly	Downtrend	15 years	-41.18%	0.46	-99.76%	-0.05
OBV	Hourly	Oscillating	15 years	739.21%	0.63	-52.90%	0.26

				10M Starting Capital		10,000 Starting Capital	
Algorithm	**Timeframe / Frequency**	**Type of Stock**	**Duration**	**Total Returns (%)**	**Sharpe Ratio**	**Total Returns (%)**	**Sharpe Ratio**
OBV	Minutely	Uptrend	15 years	162.04%	0.46	0.60%	0.16
OBV	Minutely	Downtrend	15 years	46.05%	0.59	-99.30 %	0.05
OBV	Minutely	Oscillating	15 years	-98.77%	-0.27	-75.97%	0.16
OBV	Daily	Uptrend	5 years	39.11%	0.45	53.09%	0.69
OBV	Daily	Downtrend	5 years	-95.70%	0.05	75.95%	0.55
OBV	Daily	Oscillating	5 years	51.86 %	0.4	-58.44%	-0.02
OBV	Hourly	Uptrend	5 years	-11.89%	-0.06	-17.77%	-0.13
OBV	Hourly	Downtrend	5 years	-86.87%	0.14	-85.46%	-0.2
OBV	Hourly	Oscillating	5 years	-85.28%	-0.68	-95.89%	-0.99
OBV	Minutely	Uptrend	5 years	57.36%	0.72	-16.08%	-0.12
OBV	Minutely	Downtrend	5 years	37.82%	0.5	-94.09%	-0.41
OBV	Minutely	Oscillating	5 years	1.89%	0.47	-96.78%	-1.07
OBV	Daily	Uptrend	6 months	-2.25 %	-0.15	-32.72%	-3.16
OBV	Daily	Downtrend	6 months	-54.10%	-2.25	-55.08%	-2.7
OBV	Daily	Oscillating	6 months	1.05 %	1.29	22.01%	1.48
OBV	Hourly	Uptrend	6 months	-17.92%	-1.72	4.47%	0.52
OBV	Hourly	Downtrend	6 months	-18.67%	-0.84	-34.25%	-0.43
OBV	Hourly	Oscillating	6 months	-6.75%	-1.8	-18.71%	-1.51
OBV	Minutely	Uptrend	6 months	-24.67%	-2.36	5.57%	0.57
OBV	Minutely	Downtrend	6 months	-58.35%	-2.69	-48.51%	-1.07
OBV	Minutely	Oscillating	6 months	6.08%	1.95	-21.80%	-1.61
				739.21%	1.95	4898.42%	2.32

					10M Starting Capital	
Algorithm	**Duration**	**Timeframe / Frequency**	**Stock Name**	**Type of Stock**	**Total Returns (%)**	**Sharpe Ratio**
EMA	15 years	Daily	ILMN	Uptrend	50.05%	0.34
EMA	15 years	Daily	CEA	Downtrend	-97.34%	-0.27
EMA	15 years	Daily	ZNH	Oscillating	-240.32 %	0.36
MACD	15 years	Daily	ILMN	Uptrend	36.35%	0.43
MACD	15 years	Daily	CEA	Downtrend	-0.57%	-0.12
MACD	15 years	Daily	ZNH	Oscillating	0.44%	-0.01
RSI	15 years	Daily	ILMN	Uptrend	1012.07%	0.74
RSI	15 years	Daily	CEA	Downtrend	-58.45%	-0.33
RSI	15 years	Daily	ZNH	Oscillating	-35.59 %	-0.27
OBV	15 years	Daily	ILMN	Uptrend	792.28%	0.84
OBV	15 years	Daily	CEA	Downtrend	-93.39 %	-0.05
OBV	15 years	Daily	ZNH	Oscillating	-99.27 %	-0.52
EMA	15 years	Daily	WMT	Uptrend	-48.58%	-0.19
EMA	15 years	Daily	CLF	Downtrend	-118.00%	0.2
EMA	15 years	Daily	ACTG	Oscillating	181.52%	0.14
MACD	15 years	Daily	WMT	Uptrend	3.58%	0.15
MACD	15 years	Daily	CLF	Downtrend	53.88%	0.4
MACD	15 years	Daily	ACTG	Oscillating	-11.86%	-0.27
RSI	15 years	Daily	WMT	Uptrend	-79.55%	-0.53
RSI	15 years	Daily	CLF	Downtrend	-216.60%	0.54
RSI	15 years	Daily	ACTG	Oscillating	0.33 %	-0.17
OBV	15 years	Daily	WMT	Uptrend	113.23%	0.38
OBV	15 years	Daily	CLF	Downtrend	36.22%	0.6
OBV	15 years	Daily	ACTG	Oscillating	-149.48 %	0.23
EMA	15 years	Daily	VRTX	Uptrend	-57.45%	0.09
EMA	15 years	Daily	BRFS	Downtrend	46.58%	0.2
EMA	15 years	Daily	AABA	Oscillating	-41.53%	0.39

					10M Starting Capital	
Algorithm	**Duration**	**Timeframe / Frequency**	**Stock Name**	**Type of Stock**	**Total Returns (%)**	**Sharpe Ratio**
MACD	15 years	Daily	VRTX	Uptrend	-21.05%	0.01
MACD	15 years	Daily	BRFS	Downtrend	-19.43%	-0.01
MACD	15 years	Daily	AABA	Oscillating	-79.25%	-0.33
RSI	15 years	Daily	VRTX	Uptrend	-75.62%	0.25
RSI	15 years	Daily	BRFS	Downtrend	-58.45%	-0.33
RSI	15 years	Daily	AABA	Oscillating	-22.07%	0.3
OBV	15 years	Daily	VRTX	Uptrend	1527.41%	0.59
OBV	15 years	Daily	BRFS	Downtrend	-58.35%	-0.12
OBV	15 years	Daily	AABA	Oscillating	23.67%	0.41
					1527.41%	

References

Almahdi, S., & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, *87*, 267 - 279. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0957417417304402` doi: https://doi.org/10.1016/j.eswa.2017.06.023

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). A brief survey of deep reinforcement learning. *CoRR*, *abs/1708.05866*. Retrieved from `http://arxiv.org/abs/1708.05866`

Bellman, R. (1954, 11). The theory of dynamic programming. *Bull. Amer. Math. Soc.*, *60*(6), 503–515. Retrieved from `https://projecteuclid.org:443/euclid.bams/1183519147`

Ben-Ami, Z., & Feldman, R. (2017). *Event-based trading: Building superior trading strategies with state-of-the-art information extraction tools* (Unpublished master's thesis). The Hebrew University.

Biais, B., Foucault, T., & Moinas, S. (2014). Equilibrium fast trading. *Journal of Financial economics*, *116*(2), 292-313.

Boehmer, E., Fong, K., & Wu, J. (2015). *International evidence in algorithmic trading* (Unpublished master's thesis). AFA 2013 San Diego Meetings Paper.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *Openai gym.*

Chaboud, A., Chiquoine, B., Hjalmarssonh, E., & Vega, C. (2014). Rise of the machines: Algorithmic trading in the foreign exchange market. *The Journal of Finance*, *69*(5).

Chen, N.-F., Roll, R., & Ross, S. A. (1986). Economic forces and the stock market. *The University of Chicago Press Journals*, *59*(3), 383-403.

Chu, B. (2018). Stock markets: What is algorithmic trading? and is it really responsible for the current financial mayhem? *Independent*.

Cumming, J. (2015). *An investigation into the use of reinforcement learning techniques within the algorithmic trading domain* (Unpublished master's thesis). Imperial College London.

Davila, D. (2016). Beginner's guide to reading a stock table. *The Conversation*.

Du, X., Zhai, J., & Lv, K. (2009). *Algorithm trading using q-learning and recurrent reinforcement learning* (Unpublished master's thesis). Stanford University.

Fama, E., & French, K. (1988). Permanent and temporary components of stock prices. *Journal of Political Economy*, *96*(2), 246-273.

Glassman, J. (2013, March). The art of selling stocks. *Kiplinger*. Retrieved from `https://www.kiplinger.com/article/investing/T052-C000-S002-the-art-of-selling-stocks.html`

Gosavi, A. (2009, 05). Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing*, *21*, 178-192. doi: 10.1287/ijoc.1080.0305

Hayes, A. (2017). Stocks basics: What are stocks? *Investopedia*.

Hryshko, A., & Downs, T. (2003). An implementation of genetic algorithms as a basis for a trading system on the foreign exchange market. In *Evolutionary computation, 2003. cec'03. the 2003 congress on* (Vol. 3, pp. 1695–1701).

Hur, J. (2016). History of the stock market. *Be Business Ed*.

Hwang, K.-S., Jiang, W.-C., & Chen, Y.-J. (2013). Adaptive model learning based on dyna-q learning. *Cybernetics and Systems*, *44*(8), 641-662. Retrieved from `https://doi.org/10.1080/01969722.2013.803387` doi: 10.1080/01969722.2013.803387

Investopedia. (2018). *Stock risk.* Retrieved from `https://www.investopedia.com/exam-guide/series-65/portfolio-risks-returns/stock-risks.asp`

Investor.gov. (2018). *Stocks.* Retrieved from `https://www.investor.gov/introduction-investing/basics/investment-products/stocks`

Jeong, G., & Kim, H. Y. (2019). Improving financial trading decisions using deep q-learning: Predicting the number of shares, action strategies, and transfer learning. *Expert Systems with Applications*, *117*, 125 - 138. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0957417418306134` doi: https://doi.org/10.1016/j.eswa.2018.09.036

Johnston, K. (2011). *The impact of technology on the stock market.* Retrieved from `https://finance.zacks.com/impact-technology-stock-market-5807.html`

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *CoRR*, *cs.AI/9605103*. Retrieved from `http://arxiv.org/abs/cs.AI/9605103`

Kissell, R., & Malamut, R. (2005). Understanding the profit and loss distribution of trading algorithms. *Algorithmic Trading*, 41–49.

Lee, J. W., Park, J., O, J., Lee, J., & Hong, E. (2007, Nov). A multiagent approach to$q$-learning for daily stock trading. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *37*(6), 864-877. doi: 10.1109/TSMCA.2007.904825

Li, H. (n.d.). *Introduction to algorithmic trading strategies lecture 1.*

Li, Y. (2017). Deep reinforcement learning: An overview. *CoRR*, *abs/1701.07274*. Retrieved from http://arxiv.org/abs/1701.07274

Little, K. (2018, June). How stock trading works. *The Balance*.

Lo, A., & MacKinlay, C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *Oxford Journals*, *1*(1), 41-66.

Mitchell, C. (2017, April). How the stock market works. *Investopedia*.

Mitchell, C. (2018, November). How does the stock market work? *Investopedia*.

Moody, J., & Saffell, M. (2001, July). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, *12*(4), 875-889. doi: 10.1109/72.935097

Navone, M., & Putnins, T. (2016). Explainer: the good, the bad, and the ugly of algorithmic trading. *The Conversation*.

Necchi, P. (2012). *Reinforcement learning for automated trading* (Unpublished master's thesis). Politecnico di Milano.

Pendharkar, P. C., & Cusatis, P. (2018). Trading financial indices with reinforcement learning agents. *Expert Systems with Applications*, *103*, 1 - 13. Retrieved from http://www.sciencedirect.com/science/article/pii/S0957417418301209 doi: https://doi.org/10.1016/j.eswa.2018.02.032

Rao, G. (2015). *The dark side of algorithmic high-frequency trading.* Retrieved from https://www.linkedin.com/pulse/dark-side-algorithmic-high-frequency-trading-gautam-rao/

Ratnaparkhi, S. (2017). *Why you should be doing algorithmic trading?* Retrieved from https://www.quantinsti.com/blog/why-you-should-be-doing-algorithmic-trading

Seth, S. (2014). Basics of algorithmic trading: Concepts and examples. *Investopedia*.

Smith, B. (2003). *History of the global stock market: from ancient rome to the silicon valley*. University of Chicago Pres.

Smolyakov, V. (2017, Jun). What is your review of quantopian? *Quora*.

Sutton, R. (1988, 08). Learning to predict by the method of temporal differences. *Machine Learning*, *3*, 9-44. doi: 10.1007/BF00115009

Sutton, R. S. (1991, July). Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, *2*(4), 160–163. Retrieved from http://doi.acm.org/10.1145/122344.122377 doi: 10.1145/122344.122377

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. The MIT Press.

Tillier, M. (2017, December). *What is the stock market, and how does it work?* Retrieved from https://www.nasdaq.com/article/what-is-the-stock-market-and-how-does-it-work-cm895748

Varon, J., & Soroka, A. (2016). Stock trading with reinforcement learning.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards* (Unpublished doctoral dissertation). King's College.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. In *Machine learning* (pp. 279–292).

Wong, J., Souroutzidis, Y., Lai, M., Mei, E., Sagwal, A., & Borland, D. L. (2016). *Fundamental signals for algorithmic trading* (Unpublished master's thesis). Stanford University.