Winter 2023

Reinforcement Learning for Stock Option Trading

James Garza
CCT College Dublin

CCT College Dublin

Assessment Cover Page

Module Title:	MSc Data Analytics
Assessment Title:	Master's Thesis
Lecturer Name:	
Student Full Name:	James Garza
Student Number:	Sba19053
Assessment Due Date:	24/2/2023
Date of Submission:	24/2/2023

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other sources. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Reinforcement Learning for Stock Option Trading

James Garza SBA19053

The 24th of February 2023

I hereby certify that the material, which l now submit for assessment on the programmes of study leading to the award of Master of Science in Data Analytics, is entirely my own work and has not been taken from the work of others except to the extent that such work has been cited and acknowledged within the text of my own work. No portion of the work contained in this thesis has been submitted in support of an application for another degree or qualification to this or any other institution. I understand that it is my responsibility to ensure that I have adhered to CCT's rules and regulations.

I hereby certify that the material on which I have relied for the purpose of my assessment is not deemed as personal data under the GDPR Regulations. Personal data is any data from living people that can be identified. Any personal data used for the purpose of my assessment has been pseudonymised and the data set and identifiers are not held by CCT. Alternatively, personal data has been anonymised in line with the Data Protection Commissioners Guidelines on Anonymisation.

I consent that my work will be held for the purposes of educational assistance to future students and will be shared on the CCT website (www.cct.ie) and the Research THEA website (https://research.thea.ie/). I understand that documents once uploaded onto the website can be viewed throughout the world and not just in Ireland.

24/2/2023

Signature of Candidate Date

Acknowledgements:

I want to especially thank my wife and kids for their support in my academic and professional journey. Thanks to my parents and in-laws. Additionally, a thank you to all my lecturers and CCT Faculty. Finally, I would like to thank my colleagues Takfarinas Saber and Marina Iantorno for sharing their professional advice.

Table of Contents

Table of Figures

Table of Tables

Abstract

Reinforcement learning has recently seen an increase in popularity due to its ability to learn from past experience and its capability of adapting quickly and effectively to new market conditions. This research will focus on reinforcement learning and its importance in trading stock options.

Option traders can trade options with one of two option expirations: American or European style. This research will base the analysis on the American expiration style, considered more challenging in trading than the European expiration style. This could lead to the possibility of improving the current trading techniques. In addition, this research aims to understand the role that reinforcement learning plays in trading stock options and evaluate its effectiveness in different market environments. Reinforcement learning has the potential to identify optimal trading strategies for stock options, and could assist current traders in their trading strategies.

Trading and markets have existed for millennia, going as far back as Babylon in 2000 BC, with currency exchange and commodities (Kirkpatrick and Dahlquist, 2010). However, markets have evolved and become more complex than in those early trading days. Automation of trading and trading tasks has enabled organisations to act more quickly, consistently, and cost-effectively, all while reducing the risk of human error.

The complexity of the markets undeniably increases the difficulty of option trading in dynamic environments. Two questions that arise are:

Can Reinforcement Learning models use historical option data to develop effective option trading strategies?

Can Reinforcement Learning assist human traders in trading options?

These questions are hard to answer at a glance and require robust research and exploration to understand the behaviour of this market segment.

Additionally, this research will explore the potential benefits of utilising Reinforcement Learning in stock option trading and how it might be used to modify existing techniques (Moody and Saffell, 2001). The Reinforcement Models that will be explored are Actor-Critical (A2C), Deep Deterministic Policy, Proximal Policy Gradient (DDPG), and Proximal Policy Optimization (PPO). These models use Reinforcement Learning algorithms that train an agent to solve tasks by trial and error. This research will attempt to use these trading agents to develop algorithmic trading strategies, which are difficult for human traders. In chapter 5, there is a complete description of how they work.

Ultimately, this research found that Reinforcement Learning can develop trading strategies that could assist human traders. These trading agents are based on machine learning models, which allow them to identify and analyse patterns in the data that human traders may miss. But this research gives evidence to support the results and encourages more work to be done before these can be fully autonomous strategies.

Keywords: Reinforcement Learning, option trading, and trading strategies.

Chapter 1: Introduction

The stock market is not a new term in business. Merchants in Venice exchanged their products for goods and were credited with trading government securities in the 13th century. Bankers oversaw the trading of government securities a few years later. However, the first stock markets are linked to Belgium and the Netherlands between 1400 and 1500. These countries hosted their stock system but were missing the stocks. Despite the existence of institutions dedicated to trading, there were no businesses devoted to trading company shares. Instead of that, the markets dealt with businesses and individual debts. But the first publicly traded company appeared in 1600, named "Governor and Company of Merchants of London Trading with East Indies", better known as the "East India Company." With this organisation, investors changed their trading approach and started purchasing shares in multiple companies instead of "putting all the eggs in the same basket." This method was very successful and was soon adopted by businesses in other countries. The only real issue in place was the absence of regulations, which made it challenging to distinguish between legitimate and illegitimate companies. As a result, it created a bubble in the stock market that exploded in 1825. With the intention to regulate the trade market, two institutions were defined as the main stock market organisations globally: the London Stock Exchange was the main stock market for Europe, while the New York Stock Exchange was the main exchange for America and the world (Kirkpatrick and Dahlquist, 2010) (Hur, 2016).

The existence of two main trade centres led to the existence of two types of stock options: European-style and American-style stock options. Contrary to what could be believed, it has nothing to do with the location of where the stock option is traded. Instead, it relates to the fact that European-style stock options only allow the buyer to exercise the contract on the expiration date. In contrast, American-style stock options allow buyers to exercise contracts anytime. Furthermore, most research has focused on call stock options rather than put stock options. The difference between one and another is that call options give the buyer the right to buy, and put options give the buyer the right to sell.

Following the historical events, it is essential to understand that stock options were traded formally on an exchange starting in 1973. Options were traded privately previously through institutions until the Chicago Board of Options Exchange formalised stock options trading (www.cboe.com, n.d.). Formalising stock options trading removed counterparty risk where another party would not fulfil their stock option obligation —standardising strike prices, expiration dates, and calls and puts.

Stock option contracts allow traders to hedge positions, lock in a buy or sell price, and finally, just trade options for profits (Avellaneda, Levy and ParÁS, 1995) (Amilon, 2003). The Black-Scholes model was created to price stock options. However, stock option pricing is very volatile, especially when overall market volatility increases. As a result, in times of high stock market volatility, Black-Scholes pricing cannot accurately predict stock options pricing. This is a problem for institutions and individual traders looking to trade options to either hedge positions (Lopez De Prado, 2018). In order to combat this problem, some traders are looking for alternate models that may better predict stock option pricing in times of high market volatility.

This thesis is organised as follows: Chapter 1 is the introduction and gives the background and context for this research. Chapter 2 is the research design and describes the research's primary data collection, problem identification, research objectives, validity types, and ethical considerations. Chapter 3 provides an overview of relevant literature, a theoretical framework, key concepts, definitions, and research gaps. The research methods for imputation, statistical tests, Reinforcement Learning decision agent strategy creation, and trading analysis methods are all detailed in Chapter 4. Chapter 5 is about the implementation of the process, including challenges and solutions. Chapter 6 presents the results in tables, figures, and strategy analysis. Chapter 7 is a discussion of the limitations of the study and future research directions. Finally, the overall conclusion is presented in Chapter 8.

The next chapter, "Research Design," will discuss the methodology and procedures to address the research objectives discussed in the first chapter. The chapter will begin with an overview of the primary data collection methods. Then talk about the process of figuring out what the problem is and clarification. The clarification will involve refining and narrowing our research questions to ensure they are specific, doable, and relevant. In the research objectives, which will guide our data collection and analysis processes, this research will also describe the type of validity used and how this research will ensure that the data collected is reliable and valid. Finally, this research will discuss the ethical considerations taken into account throughout the research process, including interviewee consent, strategy assumptions, and data protection.

Chapter 2: Research Design

This research's methodology and procedures are covered in this chapter to address the goals of the research. An overview of the primary data collection techniques will be presented at the beginning of the chapter. Then, go over the procedure for identifying the issue and clarifying it. The clarification ensures that the research questions are precise, realistic, and pertinent. This research will also describe the type of validity used in this research and how this research will ensure that the data collected is reliable and valid in the research objectives, guiding the data collection and analysis processes. Finally, this research will review the ethical issues that were kept in mind while conducting the research, such as data protection, interviewee consent, and strategy assumptions.

Primary Data

Primary data was collected via interviews with industry experts. Most of these experts were known by this research through years of industry experience. One interview was sought via LinkedIn after reviewing their work in this field. The interviews were conducted in a semi-structured way. The interview was structured around twelve open-ended questions to help answer the research objectives. In addition, there was space available for the interviewees to add anything they thought would be particularly helpful or that was missing from this research. Forms of consent are signed and attached to the appendix. Extra precautions were taken to ensure that the interviewees did not influence the direction of the research. The interviews were recorded and transcribed using Microsoft Office 365, and audio files and word transcripts were stored on a secured drive to protect the data of the interviewee. Most interviews lasted about an hour. This research aimed to get five interviews but only obtained four. Nonetheless, the data from the interviews proved to be of immense value.

Problem Identification and Clarification:

In light of how difficult it is to trade options and the inefficiencies of current models, a question comes up that led to this research:

Can Reinforcement Learning help option traders by developing a trading strategy based on actual market prices during normal, typical, and volatile markets?

Reading through the research and analyses, most of the working consensus predicts European-style call stock options. Call options give the buyer the right to buy, and put options give the buyer the right to sell. In this sense, it is possible to choose one of the two following routes:

Is there a gap in the research performed so far?

Is it challenging to trade American-style put stock options?

The fact is that stock options have traded on an exchange for almost 50 years and that the author of this research has been in the industry for over 20 years. This has led the researcher to conclude that there is a gap in the research. This gap is due to the difficulty in trading American-style put stock options. Trading American-style put stock options can be difficult for traders, even with the Black-Scholes model, which seems lacking in this area.

Intra-day stock and option one-minute data have been acquired. One-minute data contains the open, high, low, and close prices. This data can be used to identify market trends and patterns, which can then be used to develop trading strategies for American-style put stock options. Utilising this one-minute data, traders can identify trends and patterns in the market, which they can then use to develop trading strategies for American-style put stock options.

Research Objectives:

In light of the troublesome issues we've already discussed and by following the questions that led to this research, we can sum up the goal as follows:

- Collect one-minute S&P 500 SPY ETF and one-minute put options data. This may involve paying for data or downloading using an API.
- Apply the subsequent techniques to the data:
    - Determine whether the one-minute stock data is stationary by performing a stationarity assessment, utilising the Augmented Dickey-Fuller (ADF).
    - Determine the distribution of one-minute stock data. This could involve a visual inspection using histogram plots, statistical tests like the Shapiro-Wilk test, or both.
    - Evaluate imputation techniques. This can be accomplished using techniques like mean imputation, K-nearest neighbours (KNN) imputation, and linear interpolation. Compare the results and evaluate how various imputation techniques affected the stationarity and distribution of the data.
- Create trading strategies for American-style put stock options using Reinforcement Learning.
- Compare the trading results of the trained Machine Learning model with industry strategy evaluation metrics.

In order to accomplish these goals, this research uses an exploratory and descriptive approach. Notably, a statistical analysis will evaluate the accuracy of the models used during the research. In addition, various trading strategy evaluation metrics will be used to assess the models' effectiveness, such as the Sharpe ratio and maximum drawdown.

This research has been divided into six sections:

1. Literature Review of previous work: Understanding the studies performed before beginning the current research was vital. Through this process, the researcher gained insights and understood what had failed. This process has also generated ideas to improve techniques that have worked. Although there are several publications on the trading industry, this thesis presents the most relevant ones.
2. Methodology: the author of this research detailed the techniques and methodology followed, creating a coherent and comprehensive research process. The theory and logic behind each method have been explained thoroughly.
3. Implementation: in this chapter, the reader will find what was implemented and why—the difficulty of various methods and technology encountered and how they were overcome. At the same time, visualisations will be presented to deepen the understanding of the data and the insights gained during the research in the exploratory data analysis.
4. Results: This chapter presents the findings of the analyses and techniques used. This research's author describes the best approach and whether the results were as expected.
5. Discussion: This chapter discusses the implications and limitations of Reinforcement Learning decision agent trading strategies to provide a balanced and objective perspective and suggest avenues for future work.
6. Conclusions and discussion for further steps: rather than conclude an investigation, researchers should consider adding value to the industry with the possibility of additional work. After carefully considering the results, a discussion is set up, considering related work and possible ways to move forward. The findings indicate that the best technique for this research was the one suggested by the researcher, and the results were as expected.

Validity type:

The two types of validity in this research that are very important are content and statistical conclusions. In the initial research and literature review of academic papers, there seems to be a lot of focus on European-style call stock options and very little on American-style put stock options. A possible way to overcome the content validity of this research was to conduct four interviews with industry experts who have provided insights into why one style of option is researched over another. Naturally, these experts have their own biases, considered in this research. However, with their input, the researcher understood better why one option style is more heavily researched.

Because the data will only be a small sample size during certain market conditions, the statistical conclusion for stock options could be significant. Therefore, the trained model must be deployed in real-world market conditions in a simulated trading account to validate it. The trades could be analysed using trained data to validate the model's performance. Furthermore, the industry has established strategy metrics standards to validate the strategies' results. The recent market activity could also be better for training a model than historical market data irrelevant to current market conditions. Therefore, the model must be regularly retrained with current market data to accurately train on real-time market conditions. As such, this could ensure a model's trading agent's success in real-world trading conditions and must be regularly retrained with the most up-to-date market data. This is the key to establishing a successful trading strategy.

Ethical considerations:

There is a risk to trading in any market because markets are volatile. Traders can and do lose money as well as make money. A concern is that investors could see the machine learning model replace trading strategies, as this could ensure profits or remove risk. Prado suggests that the predictions of Machine Learning compared to actual market results are unethical as they do not reflect reality when deployed in the real world. Additional ethical issues are wasting time and resources and investors losing money. Disclosures must be made that all trading involves risk and that the Reinforcement Learning model's trading strategies may result in portfolio losses. Backtesting the models’ predictions on market data do not guarantee future profits. When the model strategies are backtested in a simulated account, this does not guarantee future performance in real markets.

Primary data must also be protected and fall under GDPR, as an industry expert could want the interview data deleted or removed from the research. Another consideration is that the trading costs could differ substantially from trading platform to platform. Therefore, trading costs could affect the profitability of the model's trading performance. Finally, licencing and the use of the data could be an issue. There is a lot of free data, though not much is accurate. This could be solved by researching how the data has been sourced, checking the provider's credentials, and trialling a few different platforms to ensure their terms and conditions are acceptable.

To conclude this chapter, this research has presented a detailed overview of the research design. By outlining the primary data collection methods, problem identification and clarification processes, research objectives, validity types, and ethical considerations, this research has demonstrated the systematic approach that will be used to address the research objectives and questions. The next chapter will present a comprehensive literature review, which will provide a critical analysis of the existing literature on the topic of our study.

Chapter 3: Literature Review

Chapter three of this thesis presents a comprehensive literature review and critically evaluates the existing research on the topic. The chapter will begin with an overview of the relevant concepts and definitions, clearly understanding the key terms used in the literature. By conducting a rigorous literature review, this research aims to provide a comprehensive and up-to-date analysis of the existing research and identify the areas where this research can make a significant contribution. In addition, the literature review will inform the subsequent chapters of the thesis, including the research design, data collection, and data analysis.

Machine Learning is changing almost virtually everything in our everyday lives, and it is inspiring in the finance industry, where it has the potential to be transformative. Academic papers and financial books about investing or stock markets largely fail in two main areas. They either offer explanations without rigorous academic theory or are written by authors who have never practised what they teach—containing elegant mathematics and theories to describe a world that does not exist (Hariom Tatsat, 2020). The theorem may be true logically, but that does not mean it is true in the real world. In this research's search for academic papers, many are divorced from practical applications to the financial market. Many applications in the trading and investment world are not grounded in sound science. This research will attempt to bridge the gap that separates academia and the industry, and the author has personal experience on both sides of the rift (Lopez De Prado, 2018).

Stock options are one central area of finance that machine learning could considerably impact. Stock options are contracts that trade on an exchange and give buyers the right to buy or sell the underlying stock (Gaspar, Lopes and Sequeira, 2020). Stock option trading can also hedge positions or reduce risk (Strong, 2005). Some stock option traders trade stock options without any other consideration, such as hedging or reducing risk. Traditionally, options have been priced using the Black Scholes model, developed in 1973 (Culkin and Das, 2017). However, other pricing models have been used to predict the price of stock options, such as the Geometric Brownian Motion, the Heston Model, and Brent's Method for implied volatility. These pricing models fall short in market volatility, which can be detrimental for traders (Liu, Oosterlee and Bohte, 2019) (Culkin and Das, 2017).

The prediction of stock option prices for trading using machine learning models has not yet been able to adequately address accurate market prices, particularly in volatile markets (Mostafa, Dillon and Chang, 2015). This research will address this problem with modern Machine Learning models, particularly Reinforcement Learning with Neural Networks (Liu, Oosterlee and Bohte, 2019). Neural networks, particularly deep neural networks, can predict the option price quickly for trading and be robust in volatile markets (Mostafa, Dillon and Chang, 2015)(Ruf and Wang, 2020). In addition to neural networks, Reinforcement Learning has shown promising results in pricing stock options, though they can be computationally taxing (Giurca and Borovkova, 2021)(Jang and Lee, 2018). Rather than using Reinforcement Learning to predict the price of the option, this research will employ Reinforcement Learning to develop trading strategies for profitability. The Reinforcement Learning decision agent, or trading agent, will be modelled after a human trader, and its trading profits will be evaluated with industry metrics.

This research will employ reinforcement learning on the S&P 500 stock index ETF. The S&P 500 represents 500 companies trading on the US stock market with large, mid-size, and small capitalisation. An Exchange Traded Fund (ETF) tracks an index such as the S&P 500. ETFs trade on an exchange like stocks. The SPY is an ETF that tracks the S&P 500 and has American-style put options (Thomaidis, Tzastoudis and Dounias, 2007).

Options

Options are contracts that grant the holder the right, but not the obligation, to buy or sell the underlying asset at a predetermined price by a predetermined date (Strong, 2005). Stock options are generally traded on an exchange. One of the oldest options exchanges is the Chicago Board of Options Exchange, founded in 1973 (www.cboe.com, n.d.). Options can be traded for hedging a portfolio or just for speculation (Strong, 2005) (AN et al., 2014). This research will focus on the S&P 500 ETF put options. The S&P 500 ETF (exchange-traded fund) is a popular option for investors looking to hedge their portfolios or engage in speculative trading.

Calls and Puts

Option holders have the right to trade the underlying assets. For the right to buy, the trader would buy a call, and for the right to sell, the trader would buy a put. The price at which the asset is traded is the strike price. When the asset prices are close to the stock price, this is called an "at the money" strike. The highest time values and volatility are at the "at the money" strike prices. This is the same for both calls and puts. "In the money" for calls is when the strike price is below the stock price. The right to buy at a lower price than the market makes those strikes "in-the-money." If it is above the stock price, but below the strike prices, those strike prices are said to be "out of the money." Puts are the other way around, as they are the right to sell. If the strike price is higher than the stock price, this is an "in the money" strike, as you can sell your stock for more than the market price. The strike prices below the stock price are "out of the money" (Strong, 2005). This research will focus on the "at the money" strike price, as this is where most traders will trade the option.

Option Expiration

In general, there are two main types of option expiration styles: not only stock options but also index options. The two styles are European and American options, and this is not about where they are located but how you can trade them. The name does not mean the location of where these options are traded but how they can be traded (Liu et al., 2021). The American-style option allows the holder the right to exercise the option at any time prior to expiration. The European-style option requires the holder to exercise the option at expiration. Exercising would be to either buy or sell, depending on the option (Strong, 2005). American-style options provide more flexibility than European-style options, as the American-style allows the holder to choose when to exercise the option, whether before or after expiration.

Option Greeks

The Greeks calculate the option price based on the underlying stock movement, interest rates, market volatility, and time. Delta, Gamma, Theta, Vega, and Rho are the five Greeks. Delta is based on the underlying stock movement, and Gamma is the Delta of the Delta. Market makers use the Gamma to hedge their positions. Theta is the stock option time value, which is how much time decay will occur based on how far out the maturity is. Vega is the underlying volatility, but it is also linked to the overall market volatility. Finally, there is Rho, which is the interest rate aspect of the option price. Since options have time values, what is the cost of money, and Rho quantifies that cost (Hull et al., 2022) (Ahn et al., 2012) (Strong, 2005). Arguably the most important or used options for Greeks are the Delta, Gamma and Vega (Hull et al., 2022). This research will develop trading models that can take into account option Greeks, and as the model is trained, it should be able to detect where the option strike price is in relation to the underlying with days to expiration (Theta) and volatility around the "at the money."

Option Pricing Models

Black Scholes Option Pricing Model

The Black Scholes Option Pricing model was published in 1973 and is used in stock option pricing. It allows traders to calculate the fair value of the option that is being traded (Merton, 1973). The Black Scholes model inputs are the option Greeks, and this makes the model a parametric model (Wei et al., 2020) (Hellmuth and Klingenberg, 2022) (BARONE-ADESI and WHALEY, 1987) (Klibanov, Golubnichiy and Nikitin, 2022) (Berkowitz, 2009). There are assumptions and limitations in the Black Scholes model. One of the limitations is the pricing of implied volatility. The assumptions in the Black Scholes models are that volatility and interest rates are constants (Mostafa, Dillon, and Chang, 2015)(Strong, 2005). The difficulty of pricing options is that pricing models do not consider the multi-dimensional inputs, and the model's deviations have come about to deal with this limitation. One primary model that has been developed to overcome the pricing limitation is the Partial Differential Equation (PDE). PDE can be calibrated using market data rather than historical market data. This is particularly important in high-volume options trades. The Black Scholes Option Pricing Model works well with European-style options (Liu, Oosterlee, and Bohte, 2019). However, the Black Scholes model fails to accurately price more complex, American-style options (Russell Rhoades, Appendix A).

The geometric Brownian motion is an assumption of the Black-Scholes model. The assumption is that the underlying stock price follows the geometric Brownian motion (Merton, 1976) (Trønnes, 2018). Consider the following where C is a call, T for time to expiration, S is the current stock price, K is the option strike price, e is the base of natural logarithms, R is the riskless interest rate, T is the time until expiration, σ is the stand deviation of returns in the underlying asset N(d_1) and N(d_2) are the cumulative standard normal distribution functions; finally, ln is the natural logarithm (Strong, 2005). A European-style call option equation is as follows:

$$C = S\,N\,(d_1) - Ke^{-RT}\,N(d_2)$$

$$\text{where } d_1 = \frac{ln\left(\frac{S}{K}\right) + \left(R + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$\text{and } d_2 = d_1 - \sigma\sqrt{T}$$

(Strong, 2005).

Black Scholes Option Pricing Model Alternatives

There are various alternative approaches to computing option prices than Black Scholes. One interesting approach to overcoming the limitations of the Black Scholes Option Pricing model is to use non-parametric models. Wei claims in their paper that their non-parametric approach is far superior to the traditional parametric approach (Wei et al., 2020) (Bakshi, Cao and Chen, 1997) (Hutchinson, LO and Poggio, 1994) (Bennell and Sutcliffe, 2004). Black Scholes parameters are the option Greeks, and more inputs seem to be required for dynamic and quickly moving market conditions. The Heston Model is one model that allows the volatility/variance diffusion process, which can overcome Black Scholes' constant volatility limitations. Liu et al. argue that using numerical methods allows for more robust numerical models such as Monte Carlo simulations or finite differences (Liu, Oosterlee and Bohte, 2019) (Bayraktar and Young, 2007) (Bouchard and Touzi, 2004) (Macbeth and Merville, 1979) (Crisan, Manolarakis and Touzi, 2010) (Giurca and Borovkova, 2021). Another alternative to Black Scholes could be Brent's pricing method, which is a more efficient and robust algorithm. Suppose the research approach is to predict implied volatility on American-style put options. In that case, efficiency will be key to keeping computational costs down and predicting in an efficient and timely fashion (Ye and Zhang, 2019) (Liu, Oosterlee and Bohte, 2019) (Liu et al., 2021). Brent's pricing method provides an alternative solution to the Black Scholes model by reducing computational costs. The more accurate and timely the price predictions, the better traders can perform in volatile market conditions.

Data

Data is critical in training Machine Learning models. The saying "garbage in, garbage out" will apply to the collected data. There were incidental costs to acquiring the SPY data, and this data was validated for accuracy. Option data was collected with polygon.io, which is free and has historical option data. However, this research could not validate the accuracy of the option data as it is expensive to find accurate data to compare. The data collected was intra-day or one-minute data on a put option close to "at-the-money" with a 120-day expiration. The ETF data collected was for the SPY ETF and one-minute data. This research collected this data to train and test a Reinforcement Learning model that could develop trading strategies. The collected data was stored in either a CSV or an H5 file. The research aimed to create a Reinforcement Learning model to utilise the one-minute data collected on put options and SPY ETFs to generate profitable trading strategies. The invoice receipt and use of data are attached in the appendix.

Statistical Tests

A data set is a collection of data that are not presented individually because the information is not helpful in its individual form. Statistics are used to summarise a set of observations, communicate the most significant amount of information as simply as possible, and draw conclusions. Descriptive statistics are information, while inferential statistics are conclusions subject to random variation, like observation errors and sampling variation. Statistical methods are employed to draw inferences from the data (Gupta et al., 2019). Descriptive and inferential statistics describe the data and test for its normality. In this research, we have discussed the summary measures to describe the data and methods used to test for the normality of the data (Agresti and Kateri, 2021) (Hastie, Friedman, and Tibshirani, 2001) (Bruce, Bruce, and Gedeck, 2020) (Dhillon, Lasser, and Watanabe, 1997).

Continuous data is said to have a normal distribution if it follows the bell curve. The bell-shaped curve is described for the mean and standard deviation, with the mean as the centre and the lower and upper integers as the ends. The mean, mode, and median are all equal in a normal distribution. Several statistical tests, such as the Shapiro-Wilkes test or Levene’s test, will verify normally distributed data (Gastwirth, Gel and Miao, 2009). It is typically assumed that normal distributions follow a bell-shaped curve and are symmetric about the mean, meaning that there is an equal amount of data above and below the mean. This normal distribution could help the model's trading performance.

Time Series

The one-minute data is a time series because there is a sequence of time in the data samples. The Augmented Dickey-Fuller (ADF) test must be applied to determine if the data is stationary. Use the ADF unit root test to determine if the data is stationary. The ADF unit root test is used to find if a series has a unit root or is non-stationary. With the t-statistics and associated p-values of the time series data, the null hypothesis is accepted for the one-minute time series. The ADF test is used to remove the unit root that is seen in the time series with seasonality. A novel test is proposed in the sequence, wherein first differences and returns are compared to each other. If they are close together, the unit root will be removed. The ADF unit root test results indicate that all transformed time series is stationary. The p values are close to zero; therefore, the null hypothesis of no stationary trend is rejected, and all transformed time series is stationary (Zhang, 2003) (Nielsen, 2019). This result is likely significant because both transformed series are "suitable" for machine learning models, which will present no autocorrelation in the errors. A significant improvement in forecasting would be expected by transforming the time series data to stationary data (Idrees, Alam and Agarwal, 2019) (Chatzis et al., 2018) (Livieris et al., 2020) (Combes, Fraiman and Ghattas, 2022) (Shumway and Stoffer, 2017) (Velicer and Fava, 2003). This finding is essential for machine learning models, as data transformation into stationary data significantly improves forecasting accuracy.

Technical Analysis

Technical analysis is based on the principle that markets have trends and is a trade discipline used to calculate investment and trade trends. Traders and investors hope to buy a stock at the beginning of an uptrend at a low price, ride the trend, and sell the stock when the trend ends at a high price. Trends exist in all lengths, from long-term trends over decades to short-term trends that occur minute-to-minute. This research will focus on minute-to-minute trends. The technical indicators used in this research will focus on price, moving averages, and volume movements. The technical analysis calculates stocks by analysing statistics and data such as trading volume and average prices. Technical analysts use tables, graphs, and analytical tools to predict stock price movements, such as support and resistance, the moving average, and volume. Support and resistance determine the trend direction or a shift in price fluctuations and transaction volume of a stock. In contrast, the moving average is a method for determining trend direction. Volume is an important indicator in conducting technical analysis because it is used to measure the value of a stock (Kirkpatrick and Dahlquist, 2010) (Dr Tom Starke, Appendix) (Russell Rhoades, Appendix) (Lawrence Cavanaugh, Appendix) (Edel MacGinty, Appendix).

Technical analysis suggests that it first began in Japan. During the 1800s in Japan, cash-only commodity markets for rice and silver were developing. The first recorded use of technical analysis was of a wealthy trader who used technical analysis and trading discipline to amass a fortune in the markets. His name was Sokyo Honma, and he was born Kosaku Kato in Sakata City, Yamagata Prefecture, during the Tokugawa period. He became wealthy by trading rice and was known throughout Osaka, Kyoto, and Tokyo. Honma's rules are recorded as the "Sakata constitution" and include methods of analysing one day's price record to predict the next day's price. Because Japan is the first place where recorded technical rules have been found, many historians have suggested that technical analysis began in the rice markets in Japan. However, it seems inconceivable that technical analysis was not used in the more sophisticated and earlier markets and exchanges in Medieval Europe. Technical analysis has a poorly recorded history but, by inference, is a very old method of analysing trading markets and prices (Kirkpatrick and Dahlquist, 2010).

Technical analysis can provide statistical smoothing on non-stationary data (Dr Tom Starke, Appendix). Technical analysis cannot make people rich like Sokyo Honma, but it can give a way to predict pricing more accurately. However, it can also be problematic: for every buy technical indicator, there can be a sell, and you end up with paralysis by analysis (Edel MacGinty, Appendix C). This can lead to too much information being taken in, and it can be challenging to make optimal decisions. Therefore, this research will focus on price, volume, and support and resistance technical indicators.

Machine Learning

Reinforcement learning has found its way into finance, mainly because the behaviours of reinforcement learning models are similar to those of traders. However, reinforcement learning-based models go one step further than price prediction-based trading strategies by determining rule-based policies for actions. It creates a decision agent to make trading decisions. As such, Reinforcement Learning-based algorithms do not produce price predictions or learn the market's structure. Instead, they learn the policy of changing trading strategies dynamically in a constantly changing market. They learn through trial and error, figuring out the optimal strategy independently. As a result, Reinforcement Learning algorithms can be easier to use than traditional finance-based hedging strategies. This research uses Reinforcement Learning to develop an algorithmic options trading strategy.

Using an Artificial Neural Network to generate a Q-table of probable rewards, the Markov decision process is the underlying computation for the Q-table. As a result, the Q-table solves a Reinforcement Learning problem, and the Reinforcement Learning model can make decisions. This research applies artificial neural networks and deep Learning to value-based and policy-based reinforcement learning.

The methodology employed to predict is of great importance, as computational costs and training time will affect the success of this research. Most of the work on pricing stock options uses Neural Networks, specifically Artificial Neural Networks. Neural networks are robust in picking up the data's underlying patterns and are non-parametric. This will allow the model to be robust in learning and predicting (Gaspar, Lopes and Sequeira, 2020) (Almgren, 2003) (Liu et al., 2021). Compute Unified Device Architecture (CUDA) will allow faster training times using the GPU rather than the CPU. On average, Deep Neural Network training time is reduced by over 30%. Deep Neural Networks that will be employed to predict probability may not be needed in this research, but improvements should be better in a simple Artificial Neural Network (Awan, Subramoni and Panda, 2017). Despite these potential benefits, deep Learning is also susceptible to overfitting and can take a long time to train (Liu et al., 2021). To counter these challenges, other methods, such as regularisation and careful hyperparameter optimisation, can be employed to reduce the chances of overfitting (Krishna et al., 2020).

Reinforcement Learning

Reinforcement learning is a process of trial and error, where a reward is given to a system upon achieving an action and satisfying a policy. It incorporates various fields, including computer science, mathematics, and philosophy. There are multiple components to Reinforcement Learning the agent, the actions, the environment, the state, and the Policy (Hirchoua, Ouhbi and Frikh, 2021) (Phil Winder Ph.D, 2020). The agent performs the actions which can be done in an environment; the state is the current situation, and the reward is the consequence of the action. The agent’s last action is evaluated when the environment sends immediate feedback. Reinforcement learning aims to maximise rewards by adapting to the environment. The agent can actively adjust by trying different strategies and deciding on an optimal one. The model doesn't know when it gets rewarded; it only knows when the next reward comes (Enes Bilgin, 2020) (Sutton and Barto, 2018).

Reinforcement Learning attempts to learn the optimal policy. An optimal policy will tell the model how to act to maximise the return in every state. The goal of a reinforcement learning agent is to learn to perform a task in an environment well. The task of this Reinforcement Learning will be to perform option trading. One author explained their methodology and used their trained model for portfolio management using the trained models (Giurca and Borovkova, 2021). These real-world applications are not just for portfolio management but also options trading (Giurca and Borovkova, 2021) (Ritter, 2017) (Ritter and Kolm, 2018) (Buehler et al., 2019) (Kolm and Ritter, 2019). The Reinforcement Decision Agent could bring the financial services industry to some level of automation. Reinforcement Learning, capable of simulating dynamic and uncertain environments, could provide a comprehensive solution for options trading.

The Bellman Equation

The Bellman equations decompose a prospect's immediate reward, plus discounted future rewards, into an immediate and a discounted reward. An agent's aim in reinforcement learning is to get the optimal Q-value and value function. The Bellman equation is a way to get the values. In addition, the Bellman equation can derive the value function and refine the value function and Q-value. Unfortunately, these equations cannot be used directly in many scenarios, but they lay the theoretical foundation of many Reinforcement Learning algorithms (Hariom Tatsat, 2020) (Pavel, Muhtasim and Faruk, 2021).

Markov Decision Process

Numerous Reinforcement Learning issues can be modelled as Markov decision processes. The agent and the environment interact continuously; the agent chooses actions, and the environment reacts to those actions by presenting the agent with new situations, with the goal of helping the agent develop the optimal strategy. The overall algorithm is founded on Bellman equations. States of the Markov Decision Process have the Markov property because the future depends only on the current state and not on the past. This can be formulated as a Reinforcement Learning problem with transition probabilities and rewards for various actions. Variables will be assigned to the three market conditions bull, bear, and stagnant. The presented Markov Decision

Process is a possible scenario in which the transition probabilities are known, and the trader's action is assumed to alter the market's state (Gupta and Dhingra, 2012). If the initial condition is "buy signal," the model can choose between "sell," "buy," and "hold." If the model decides to "buy," the state will continue to be a "buy signal" with certainty but no reward. The model can then decide to stay there forever if it so chooses. However, if the model selects the action "hold," it has a 70% chance of receiving a reward of +50 and remaining in the "hold" state. The model can then attempt to earn the maximum reward possible. Using the Markov decision process, it is possible to devise an optimal policy or strategy to achieve the greatest possible reward over time. To determine the optimal policy, the Bellman equation is used. It generates a table of actions, their respective probabilities, and rewards. The model can then determine the optimal course of action and revise the policy based on the implemented measures. If there is no table containing actions, the model could remain in the "hold" state forever. Using Artificial Neural Networks, the Bellman table is computed (Amellas et al., 2020) (Hariom Tatsat, 2020). The Bellman table is a type of reinforcement learning algorithm that permits an artificial agent to interact with and learn from its surroundings.

Temporal difference

As stated above, in most cases, a Reinforcement Learning problem with discrete actions can be modelled as a Markov decision process, but the agent may not understand the transition probabilities. Furthermore, they don't know what the rewards are going to be. This is where temporal difference learning could be useful. A temporal differencing learning algorithm is very similar to the value iteration algorithm but is tweaked to consider that the agent has only partial knowledge of the Markov Decision Process. They assume that the agent initially only knows the possible states and actions and updates their state estimates based on observed transitions and rewards. The key idea of temporal differencing Learning is updating the value function towards an estimated return, with the learning rate hyperparameter controlling how aggressive the updates are. When the learning rate is close to zero, there is little update. Instead, they may replace the old value with the updated value when they're close to zero (Alexander Alexander Zai, 2020) (Hariom Tatsat, 2020) (Heinrich, 2006).

Artificial Neural networks

A lot of complex things are happening in the brain. It can learn new things, adapt to environmental changes, and analyse unclear information. Although these are all qualities of a thinking machine, the brain is not yet advanced enough for one. Therefore, the analogy of ANNs to human biological brains has been questioned. Some argue that they should abandon the metaphor and focus on mathematics. An ANN is composed of processing units called neurons. These units try to replicate the structure and behaviour of biological neurons. They have inputs and outputs classified as dendrites and synapses, respectively. The realisation of the neuron only happens when certain functions are activated. This realisation is determined by which function activates the neuron. The other functions used in ANNs are the step function, linear function, ramp function, and hyperbolic tangent function. ANNs consist of a large number of simple processors called nodes, and weighted connections link the nodes. Each node only receives input from a few other nodes and produces a single numerical output based on those inputs. For multivariate inputs, each unit of an ANN is not very powerful on its own, generating a scalar output with a single numerical value (Dongare, Kharde and Kachare, 2012; Kukreja et al., 2016) (Aurélien Géron, 2019).

The Perceptron

The Perceptron is a simple ANN architecture invented in 1957 by Frank Rosenblatt. It is based on a slightly different artificial neuron called a threshold logic unit (TLU). It is customary to add a bias feature to each input. There are five types of neurons in a Perceptron; input, bias, passthrough, output, and bias. The Perceptron can use its activation function to classify instances into three classes. It is trained on Hebb's rule, considering its error when making a prediction. Scikit-Learn's Perceptron class uses the Stochastic Gradient Descent algorithm for training. It has one input layer, one or more layers of hidden units, and an output layer that is full and connected to the next layer. The architecture of a Multilayer Perceptron has two input neurons, a hidden layer of four neurons, and three output neurons. The signal flows in only one direction and is an example of a feed-forward neural network. The neural network backpropagates error gradients to calculate how much an error gradient contributed to an error. It also calculates how much each connection contributes to an error. GPU-based Neural Networks can process data more efficiently than traditional Neural Networks. (Aurélien Géron, 2019) (Hornik, 1991).

The Backpropagation, as discussed above, can compute the gradient of the network's error with regard to every single model parameter. Once it has these gradients, it performs a regular Gradient Descent step, and the whole process is repeated until the network converges on the solution (E, Han and Jentzen, 2017) (Sirignano and Spiliopoulos, 2017). The neural network can process a mini-batch of data if requested (for example, one per instance) and go through the whole training set many times. The final output layer is passed on to the next layer until the output layer is reached. This process uses an algorithm to measure the network's output error and is called a loss function (Christoffersen and Jacobs, 2003). Then it computes the contribution of each output connected to the error. This is done by applying the chain rule. The algorithm measures how much error contribution comes from each connection in the layer below. This reverse pass efficiently measures the error gradient across all the connection weights in the network by propagating the error gradient backward through the network. Finally, they use the error gradients computed to tweak all of the neural network's connection weights (Aurélien Géron, 2019) (Gencay and Min Qi, 2001).

This research approach is fascinating as it comes close to actual artificial intelligence. Artificial intelligence is based on the Reinforcement Learning model of updating the policy in each state after each action and learning from each action for future actions (Sagiraju and Mogalla, 2021) (Ritter and Kolm, 2018) (Chiang et al., 2016) (Mnih et al., 2015). The main issue is not knowing how the model came to its decisions or the training time. Combining Reinforcement Learning with Artificial Neural networks creates an interesting prospect for improving training time and policy decisions. Combining the technologies could also reduce the training time needed and allow the trading policy to trade in a timely manner. Substantial losses could occur if the hardware is unable to do so in a timely manner (Anders, Korn and Schmitt, 1998) (Hornik, Stinchcombe and White, 1990) (Liang et al., 2009).

Feed Forward

Feed-forward networks comprise a series of layers of nodes, with connections from later-layer nodes to earlier-layer nodes. These networks are one of the most common learning neural networks, and are often referred to as "neural networks" since they are of a type that is known as a feed-forward network. The power of this type of network emerges from the combination of multiple nodes in an appropriate way. A feed-forward network is made up of three types of elements: (a) a set of synapses, (b) a linear combiner for adding the input signals, and (c) an activation function for limiting the output of a neuron to some finite value. In this ANN architecture, inputting the activation function's input using a bias term is possible. Activating its output occurs when more than a certain number of inputs are activated. Feed-forward networks are composed of layers of nodes that have connections from nodes in a later layer to nodes in a previous layer (Hornik, Stinchcombe and White, 1989) (Dongare, Kharde and Kachare, 2012; Aurélien Géron, 2019). Feed-forward networks can be thought of as directed graphs, with each layer representing the set of nodes and each connection the set of weights between them (Aurélien Géron, 2019).

Back Propagation

Backpropagation is a supervised learning algorithm used in layered feed-forward ANNs. The neurons in the network are layered and send their signals "forward." There may be one or more hidden intermediate layers. The backpropagation algorithm in supervised Learning means that they use examples of the inputs and outputs they want the network to compute. The training begins with random weights, and the goal is to reduce the error until the ANN learns the training data. The number of hidden layers and the number of neurons in each hidden layer are decided by trial and error using the experimental data. The most traditional number of layers and neurons is determined according to the application, the complexity of the problem, and the number of inputs and outputs. Most artificial neurons are composed of a single layer of TLUs, with input neurons connected to input neurons of the previous layer (Kukreja et al., 2016) (Aurélien Géron, 2019) (Dongare, Kharde and Kachare, 2012).

To conclude this chapter, this research has provided a comprehensive and critical analysis of the existing literature research topic. In addition, the literature review has provided a theoretical framework that will guide this research in designing a Reinforcement Learning trading agent, data collection and analysis. In the next chapter, the methodology used to collect and analyse the data will describe the procedures used to address the research questions and objectives.

Chapter 4: Methodology

The methodology that will be used to gather and analyse the data is described in this chapter. This chapter's goal is to thoroughly explain the methods used in this research to address the research questions. The data that was used, the imputation techniques used, the statistical tests conducted, the reinforcement learning models created, the backtesting techniques employed, and the performance metrics used will all be covered in this chapter. Start by describing the data used, its sources, and procedures to gather and clean it. Then, go over the imputation techniques used, such as mean imputation, K-nearest neighbours (KNN) imputation, and linear interpolation, to fill in any missing data. Statistical tests, such as the Shapiro-Wilk and Augmented Dickey-Fuller (ADF) tests, were used to evaluate the data's stationarity and distribution. Then go over the machine learning algorithms created, Reinforcement Learning decision agents used to develop trading strategies. Next, review the steps taken to optimise the models and the algorithms we employed when creating these models. The backtesting techniques, which included simulating trades using historical data, that were used to assess the trading strategies' efficacy will then be discussed. Finally, review the performance metrics used to evaluate the trading strategies, such as the Sharpe ratio and the Maximum Drawdown. By thoroughly explaining the methodology, the hope is to ensure the objectivity and reproducibility of this research's findings and to shed light on the difficulties that arise when using the methodology in real-world situations.

Data

The collected data must be explored before it can be analysed. The approach this research took was to analyse the close of the SPY put option to determine profits or losses as traders would do in actual trading. There could be patterns or trends identified in the data. Reinforcement Learning could identify the pattern or trend in the data to trade successfully. There are two datasets to be collected. These two datasets must be verified for accuracy and stored securely. Acquiring data from various historical market data providers could be difficult, balancing cost and data validity. The data could have noise, which will have to be evaluated as noise in the data could negatively influence the Reinforcement Learning model. Noise is data that is not relevant to making trading decisions. Technical indicators must be added and calculated from the SPY ETF data, as this is what traders look at when making trading decisions. The industry expert interviews will determine which technical indicator and what settings. Finally, the data may need to be scaled. Scaling takes out the extremes of the data values and brings the data within a range that makes it easier for the model to make trading decisions. Scaling does not deal with outliers, and outliers could negatively affect trading decisions. Outliers are data points that fall outside of the 1.5 interquartile range. Outliers could be part of the dataset itself, which will be explored. Outliers can disproportionately impact trading decisions due to their extreme nature. Exploring the data is vital to knowing which methods to implement and evidencing those methods' results.

Imputation

This research will explore the imputation methods of Forward Fill, Back Fill, KNN, linear interpolation, and Mean imputation. The Forward Fill method propagates the last valid observation in the column using interpolation. The Back Fill method does the opposite and propagates the first valid observation after identifying the missing value. The Known Nearest Neighbour is a non-parametric supervised machine learning model that fills objects based on their proximate neighbours. The aim is to keep the integrity of the original data after imputation.

Statistical Tests

Once the data has been imputed, the next step in this research will be to apply statistical methods to the means and variance of the dataset. First, the Augmented Dicky Fuller (ADF) test is used to test if the target variable is stationary. Secondly, the Shapiro-Wilkes test for normal distribution of the percent changes of the dependent variable. Finally, this research will run the seasonal decomposition test to demonstrate if there is any seasonality in the data.

Some insights in the data could help understand how the Reinforcement model arrives at trading decisions. For example, there could be a correlation between the time of day, volume, and other factors (Lawrence Cavanaugh, Appendix).

Reinforcement Learning

Reinforcement Learning is being used more in financial services. This machine learning model goes beyond predicting a label or target variable and is an actual decision agent. The decisions could be evaluated and measured against metrics used in financial services. Seeing how all the inputs affect the model and its decisions is exciting. This research will attempt to implement code from scratch acquired from several GitHub libraries that quants in the financial services field have shared. If this is not feasible, this research will look for Python libraries for Reinforcement Learning. Reinforcement Learning is the process of using feedback from the environment to guide decision-making, making it a great candidate for this research. By implementing the code from scratch and looking for Python libraries, this research will explore how Reinforcement Learning can optimise decision-making in the financial services industry, specifically stock options (Zhang, Zohren and Roberts, 2019).

Reinforcement Learning and Artificial Neural Networks can be computationally costly in their implementation. NVIDIA's CUDA allows Reinforcement Learning models and Artificial Neural Networks to use the Graphics Processing Unit (GPU) rather than the Central Processing Unit (CPU), reducing training time by up to 30%. In addition, the GPU has hundreds of cores, allowing for parallel processing. This research will install CUDA that works with TensorFlow for Neural Networks and Torch (Choquette et al., 2021).

Backtest

When a strategy is conceived, there is a way to test if the strategy is viable or not in the form of a backtest. This takes historical data and applies the strategy to this data to see if the strategy is feasible or not. There are two approaches to backtesting: one is event-driven, and the other is vectorised. Event-driven backtesting is similar to actual trading, with trading costs, slippage, portfolio values, leverage, and stocks. This will give a detailed insight into the strategy closely tied to actual trading. The vectorised backtesting is faster and can approximate actual trading. The possibility of running a vectorised backtest in such a short amount of time means there could be a Monte Carlo analysis of the backtests (Dr Tom Starke, Appendix C). Backtests have the disadvantage of having too many variables to account for and biases. What if the backtest and strategy picked the right time frame, market conditions, or stock selection? Survivorship bias in stock selection is a big issue. The backtest should only be used for proof of concept, and simulated results in real market environments could better validate trading strategies. This is critical to avoid wasting time or money, as discussed in the ethical considerations. When developing trading strategies, it is crucial to be aware of the limitations of backtesting and not rely on past results to indicate future performance.

Performance Metrics

Performance metrics will be those used in the financial services industry to evaluate trading performance. These metrics consist of the trading strategy's performance and the risk factors associated with the trading industry (Edel MacGinty Appendix C) (Dr Tom Starke, Appendix C) (Liu et al., 2021b).

Sharpe Ratio

Although it is generally true that "more risk, higher return," the Sharpe ratio can be used to determine whether there is, in fact, a greater reward for taking on more risk. The Sharpe ratio compares the performance of a trading strategy to that of a risk-free asset after risk is taken into account. For example, US Treasury bonds are usually considered risk-free investment assets. The Sharpe ratio is determined by dividing the difference between the investment's returns and the risk-free return by the investment's standard deviation. This ratio can easily show the additional return an investor receives for each unit of heightened risk. William F. Sharpe, who created it in 1966, received credit for its name (Sharpe, 1994) (Gatfaoui, 2015).

Max Draw Down

The maximum drawdown is the most significant decline from a peak to a trough in the trading strategy. This is regarded as an indicator of downside risk. Maximum drawdown is therefore used to measure the risk-reward ratio of a trading strategy since it gives investors an idea of how much money they may lose when investing in a particular trading strategy.

Overall Return

The overall return is calculated by subtracting the portfolio’s final value from its initial value and dividing it by the initial value. This is used to validate a strategy's performance and the Sharpe ratio. This value is what most investment performance is measured by, and it is vital to show proof of concept in the strategy. This value cannot be taken alone, as the above-listed state risk measures are just as important. Thus, investors must consider other factors affecting the overall return, such as volatility, diversification, and the Sharpe ratio.

Chapter four of this thesis presents the methodology used to collect and analyze the data. The chapter began with a description of the data, including the sources of the data. And how the data cleaning and preparation process and any missing data will need to be imputed. The chapter also discussed the statistical tests we will use to assess the stationarity and distribution of the data, such as the Augmented Dickey-Fuller (ADF) and the Shapiro-Wilk tests. In addition, it describes the use of Reinforcement Learning trading agents that will be used to build trading strategies and evaluation of those strategies' metrics. In the next chapter, this research will discuss the implementation of data imputation, statistical tests, and Reinforcement Learning trading agents and evaluate the backtest of the trading strategies using performance metrics such as the Sharpe ratio and Maximum Drawdown.

Chapter 5: Implementation

Chapter five of this thesis describes the implementation of the research methodology, which involves collecting the data and applying the methods described in chapter four. The chapter will begin with a discussion of the data collection process, including the time period, frequency, and sources of the data. Next, this research will describe how the data was obtained and any issues or challenges encountered during the data collection. Next, this research presents the steps it took to prepare the data for analysis, which includes data cleaning, merging and imputation, and statistical tests described in chapter four, and how these methods ensure that the data is suitable for analysis. Finally, this chapter provides a detailed overview of the data analysis process, including how to use Reinforcement Learning decision agents and backtesting to develop and evaluate trading strategies. A detailed account of the data collection and analysis process aims to ensure this research's transparency and reproducibility and provide insights into the practical challenges of applying the methodology in practice.

Data

Several market data providers were used to collect the historical data for the SPY ETF and put option for this research. The SPY ETF data had a cost associated with it, and put option data was free with an API, but verifying the accuracy was impossible without incurring additional fees. The SPY ETF's historical data goes back 20 years, and the data has been confirmed. Once this data is downloaded, technical indicators can be added to the dataset. Based on an interview about how day traders use technical indicators, pivot points and moving averages were added (Edel MacGinty, Appendix C). Acquiring the option data was more tricky as this was a free API.

This research iterated through six strike prices within the same time frame and found the option historical data with the most historical data and the most observations, the 380 strike with 20,990 observations. The SPY ETF observations during this time frame were 255,143. More trading is closer to expiration, and this research then filtered the date from the 1st of October 2022 to the 1st of February 2023. This filter reduced the percentage of missing data values from 74.49% to 48.63% shown in Figures 1 and 2.

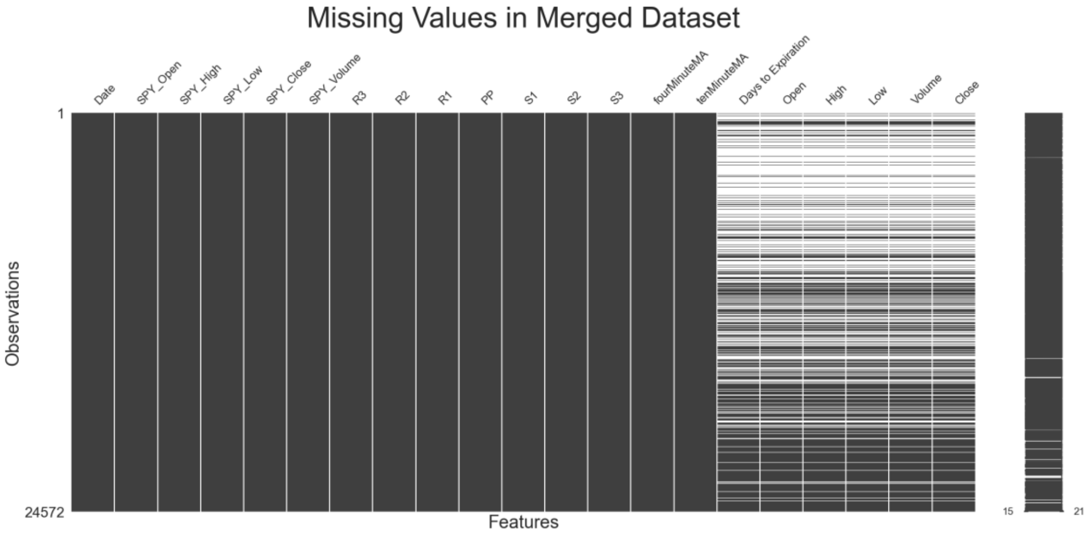


Figure 1

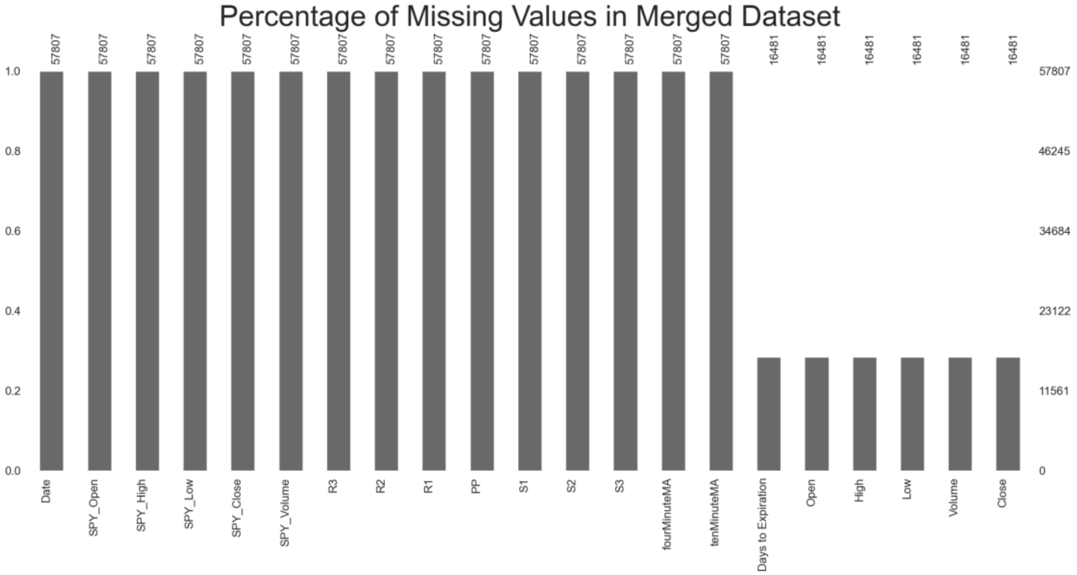


Figure 2

The question is whether patterns and trends can be maintained when nearly half of the data is missing. Despite many missing data values, trends and patterns can still be seen over this period. With filtering out the time frame from the 1st of October, the missing values are 51.37% of the dataset, as demonstrated in Figure 3.

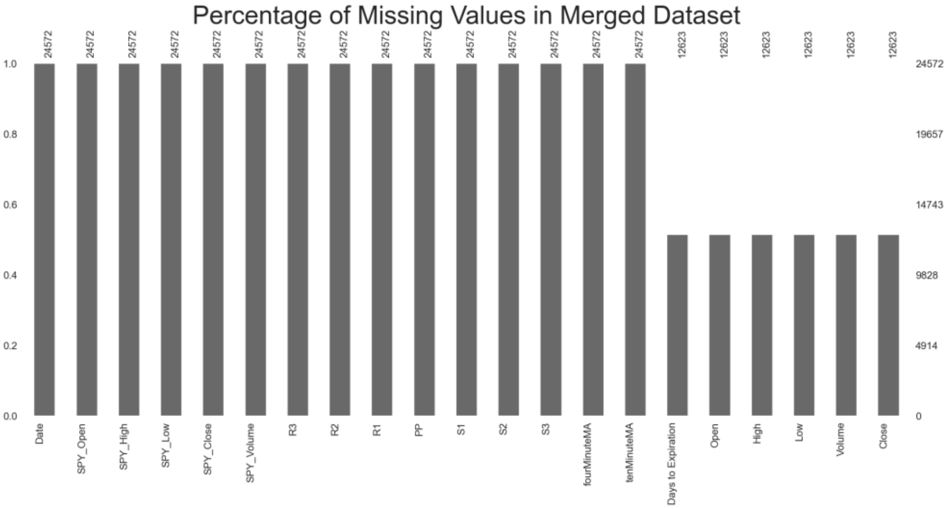


Figure 3

Imputation

The selection of the imputation methodology is important to maintain the pattern and trend of the dataset. The two selected approaches were KNN and Forward Fill interpolation, which were evaluated by running a few statistical tests on the two imputed datasets. Figures 4 and 5 below show both the mean and standard deviation of the methods of imputation:

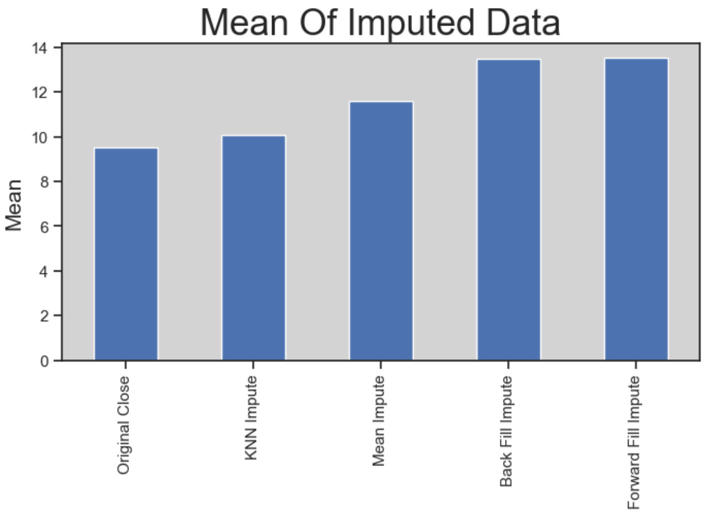


Figure 4

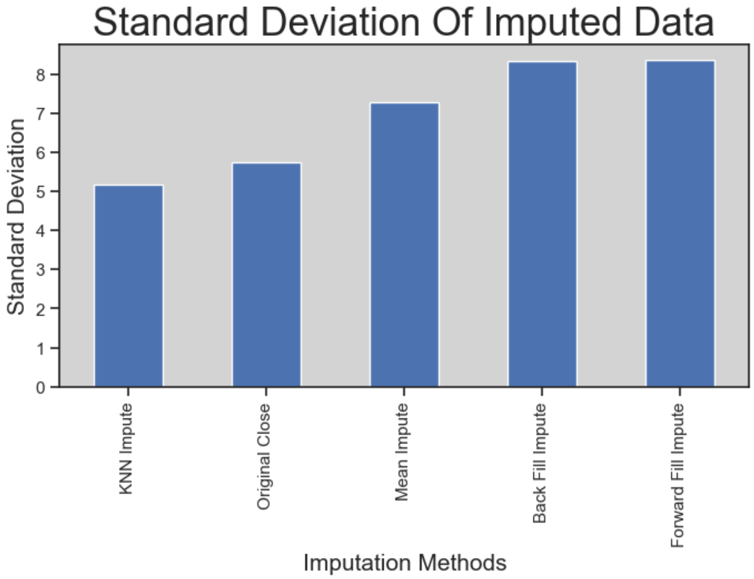


Figure 5

The KNN imputer keeps the mean and standard deviation closer to the original data than the other imputation methods. On the other hand, the forward-fill interpolation method is the furthest away from the original data structure (Heshan Guan and Qingshan Jiang, 2008) (Kalpakis, Gada and Puttagunta, n.d.). This impacts how the reinforcement model trades based on the provided data and the trading profits or losses. However, taking a closer look by plotting with a line graph, the data over time shows a different story. Figure 6 is the KNN imputation line plot which indicates far more volatility over time than the Figure 7 plot of forward-fill interpolation.

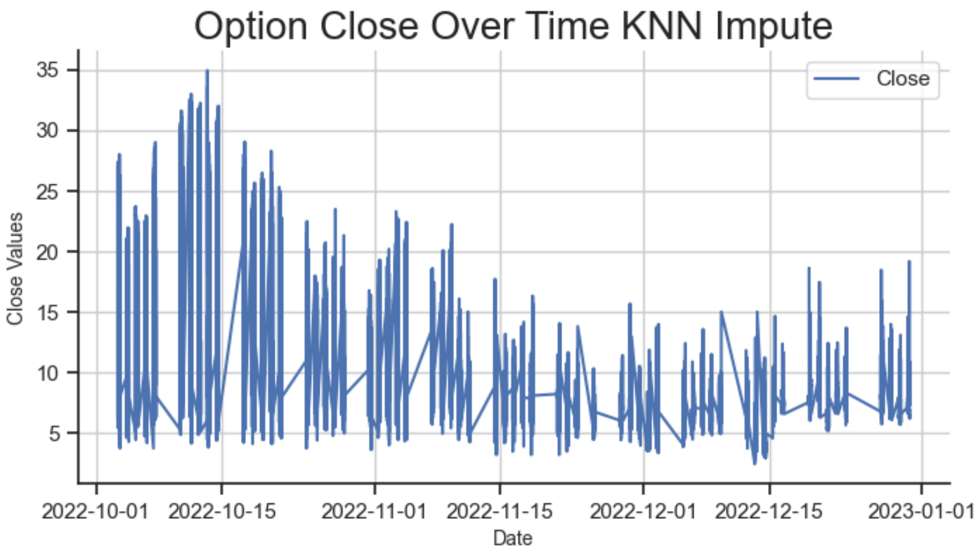


Figure 6

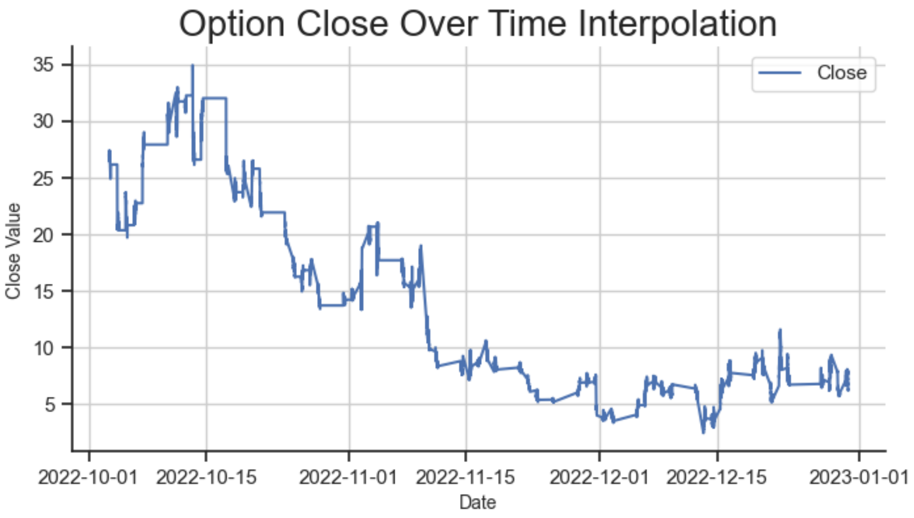


Figure 7

The tables below show insights into the option's volatility throughout the day, week, and month. This could be difficult for the Reinforcement Learning decision agent to accomplish consistently.

Table 1

Imputation Method	Minute Volatility %	Daily Volatility %	Monthly Volatility %
SPY 380 Put KNN	0.44%	9.33%	42.75%
SPY 380 Put Interpolate	0.01%	0.23%	1.06%

The daily volatility with the KNN imputes is significantly higher than Fill-forward interpolation impute. Less volatility will tend to train a better Reinforcement Learning trading decision agent. Thus, if the goal is to train a Reinforcement Learning trading decision agent that will be as effective as possible, the feed-forward interpolate impute should be preferred over the KNN impute.

A final point regarding the imputation methods: when a box plot is done, the KNN impute method has many outliers, whereas the forward-fill interpolate impute method has no outliers on the target variable. This means that when using the forward-fill method, we are more likely to have reliable data points with little or no distortion of the actual value of our target variable, as demonstrated in Figure 8 and Figure 9, respectively.

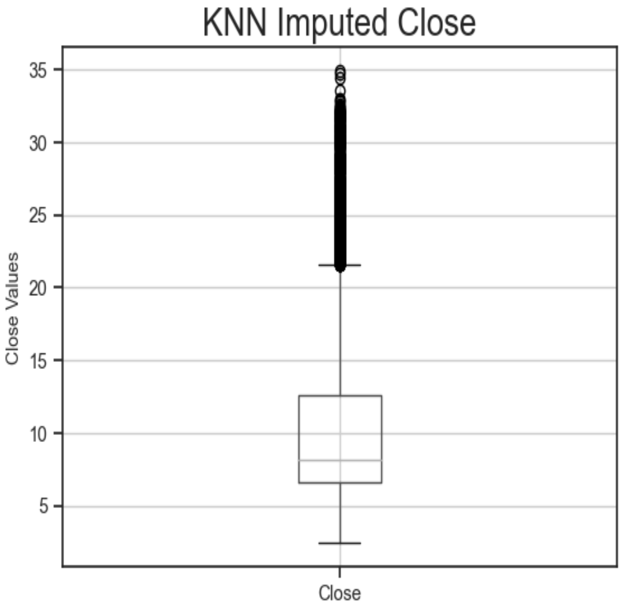


Figure 8

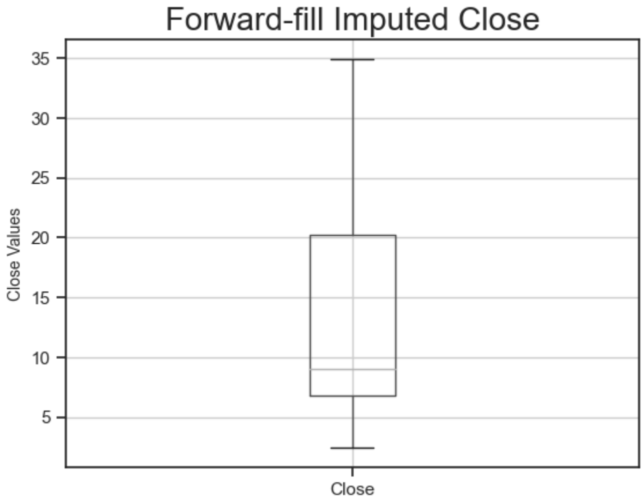


Figure 9

The line plot demonstrated that this dataset looks more like a time series dataset in its movements. The minute, daily, and monthly volatility is additionally lower, and finally, the lack of outliers would indicate a better dataset to train the model on. Both datasets will be used to train and test the Reinforcement Learning models to confirm the above conclusions.

Statistical Tests

The first statistical test is to determine if the label is stationary or non-stationary. The statistical test to do this is the Dicky Fuller test (Zhang, 2003) (Nielsen, 2019). If the p-value is greater than 0.05, the null hypothesis (H0) is not rejected, and the label data is non-stationary. Otherwise, reject the null hypothesis (H0) and conclude that the label data is stationary if the p-value is less than or equal to 0.05 (Avishek Pal, 2017). The forward-fill interpolated imputed data has a p-value of 0.479632, which is greater than the 0.05 threshold, indicating that the data is not stationary and must be rejected. The results are in Table 2.

Table 2

	KNN Impute	**Forward Fill Impute**
Interpolate ADF Statistic	-4.90	-1.61
Interpolate p-value	0.000035	0.479632
Interpolate Critical Value 1%	-3.43%	-3.43%
Interpolate Critical Value 5%	-2.86%	-2.86%
Interpolate Critical Value 10%	-2.57%	-2.57%

The distribution of the label data showed a non-normal distribution, which was expected from time series data. Therefore, to deal with the non-normal distribution of the dataset, technical indicators were added as a form of smoothing the data (Dr Tom Starke, Appendix C).

The final statistical test was to seasonally decompose the label to validate that there is no seasonality in the data. Then, employing the stats model library in Python, there is a plot that shows if there is seasonality in the dataset itself. This test showed no seasonality in the dataset, confirming our hypothesis and reinforcing that the data validly represents current trends. Plotted results are in Figures 10 and 11, respectively.

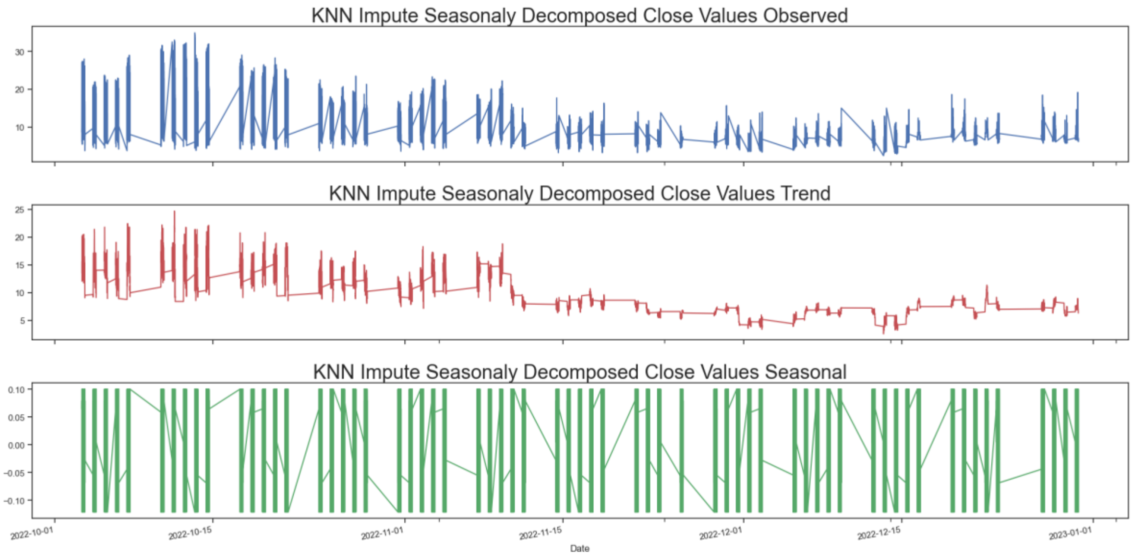


Figure 10

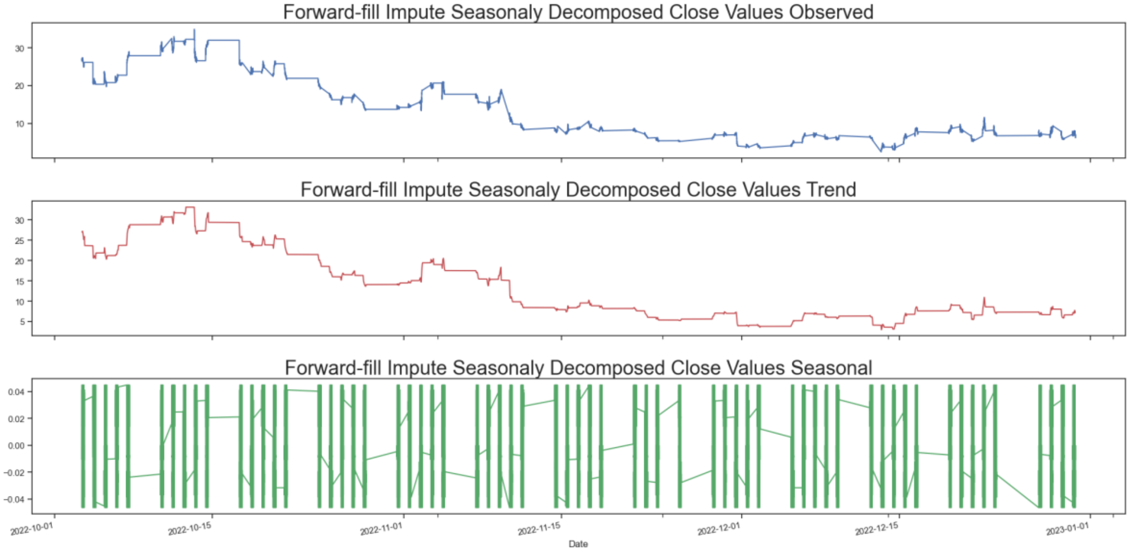


Figure 11

Reinforcement Learning

Reinforcement Learning decision agents need to evaluate each state and make a decision. There is a reward table that can show the best decision. The problem is that the decision agent can wait and do nothing because the best reward holds until the historical data ends. The reward function helps the decision agent make decisions closer to what a human would do. Finding the right balance in the reward function is difficult (Deng et al., 2017) (Zhang, Zohren and Roberts, 2019). The Bellman Equation will be used to mitigate that lack of action by a decision agent. It uses the Bellman Equation to produce a Q table to find the probabilities and rewards for each agent's decision. The Q table must be produced quickly, and this is done using Artificial Neural Networks. This Q table is based on previous states and actions. A discount factor will reduce the Q table values over time; this is called the Epsilon, and once the Epsilon reaches zero, the Q table and reward table are identical (Dr Tom Starke, Appendix C). Implementing the Reinforcement model is the next step.

Initially, this research used Python code for Reinforcement Learning, not a Python library but code written by quants. Upon finding FinRL, a Reinforcement Learning Python library for quants, this research replaced earlier code with the FinRL code. FinRL has lots of Reinforcement Learning model code that can be easily implemented once you understand how the data needs to be formatted for input. FinRL has a preprocessing function that prepares the data for the model and adds a new column of daily returns. Because it is used in all models, the results will be consistent.

The splitting of the data function from FinRL splits it based on the date, and this research used two months to train and one month to test. FinRL libraries used the A2C, DDPG, and PPO to get baseline results with the default settings (Wang et al., 2015) (Liu, 2022) (Liu et al., 2021b) (Yang et al., 2020). Descriptions of these functions are below.

Advantage Actor-Critic (A2C)

Advantage Actor-Critic (A2C) is a standard actor-critic Reinforcement Learning algorithm used to improve gradient policy updates, using an advantage function to lower the variance and increase the model's robustness. It is more economical, quicker, and more effective with large batch sizes, making it ideal for stock option trading (Bekiros, 2010) (Zhang, Zohren and Roberts, 2019).

Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) is a Reinforcement model that combines Q-learning and policy gradient frameworks using Artificial Neural Networks as function approximators. It is proposed to deterministically map states to actions to better fit the continuous action space environment (Liang et al., 2018) (Zhang, Zohren and Roberts, 2019).

Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) is a Reinforcement Learning model; its primary function is to control the policy gradient update and ensure that the new policy will not be too different from the previous one. PPO improves the stability of the policy network training by restricting the policy update at each training step. It is stable, fast, and simpler to implement and tune (Schulman et al., 2017) (Zhang, Zohren and Roberts, 2019).

Optuna Optimization

A hyperparameter Python tuning library called Optuna is compatible with FinRL and other frameworks. It uses fundamental elements like objectives, trail suggestions, and optimisation. It begins by offering a selection of hyperparameters. Then, the hyperparameters are tuned to minimise or maximise the objective (Akiba et al., 2019).

This research then optimised the settings using the Optuna function to evaluate the baseline results and see if there was any improvement by hyper tuning the parameters. The Optuna function searched through the parameters such as time steps, learning rate, batch size, discount factor, etc. (Liu, 2020). The results can then be evaluated using the financial metrics of the backtest.

Implementing CUDA GPU processing would have been ideal for improving the computational training and testing time, but these efforts did not yield the desired outcome. With Tensorflow, CUDA worked with GPU but not with Torch. This made training times 30% longer. Despite these setbacks, the results of the backtest were still quite promising.

Backtest

Implementing the backtesting was simple with FinRL, and the actions and values were saved into a data frame. This data frame can then be analysed using the Pyfolio Python library, producing the financial metrics and plots of the drawdown and rolling Sharpe ratio. Backtests can provide an inaccurate prediction of how the Reinforcement Learning decision agent will perform in real-world markets. Backtests are subject to bias as the results can be determined by chance by selecting the correct time frame. Additional analysis and testing must be done before these decision agents trade actual funds. The backtest gives you a proof of concept but not actual real-world results until it is tested. Backtests are a useful tool for determining the potential performance of a Reinforcement Learning decision agent. Still, they should not be interpreted as proof that it is reliable enough to trade real money.

In this chapter, this research has presented the implementation of the research methodology, which involved collecting and analyzing the data using the methods described in chapter four. Describing the data collection process, data cleaning, applying imputation methods, and statistical tests steps taken are detailed to ensure the data is suitable for analysis. The next chapter will present the implementation results, including a detailed analysis of the trading strategies developed and evaluated and an interpretation of the findings in the context of this research questions and objectives.

Chapter 6: Results

Chapter six of this thesis presents the results of this research, which aim to evaluate the effectiveness of using one-minute data to develop and evaluate option trading strategies. In addition, the chapter will present a detailed analysis of the trading strategies that were developed and evaluated. The results chapter will provide this thesis's main contribution and demonstrate the methodology's practical implications for developing and evaluating trading strategies using one-minute data.

Performance Evaluation

The Reinforcement Learning models took over 15 hours to train and test the decision agent. First, there is an evaluation of the overall return results. In both datasets, the A2C and DDPG models have the same results, which are higher than the PPO results. One interesting observation is that the KNN imputed data appears to have a higher overall return and is more stable daily. In contrast, the forward-fill imputed data is much more volatile. This is interesting, as the dataset and backtest result plots seem opposite. Although the KNN imputed data produces higher overall returns and is more stable than the forward-fill imputed data, the visual plots suggest that there are more anomalies in the KNN imputed data when compared to those in the forward-fill impute. See the plots below:

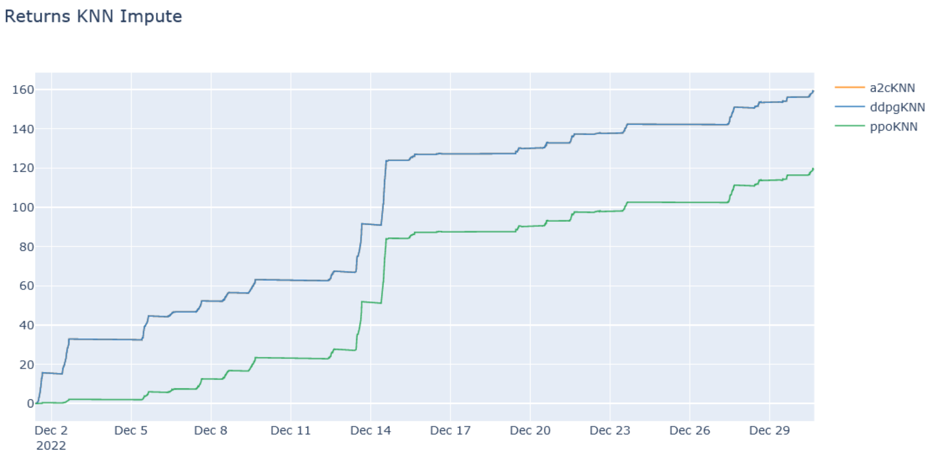


Figure 12

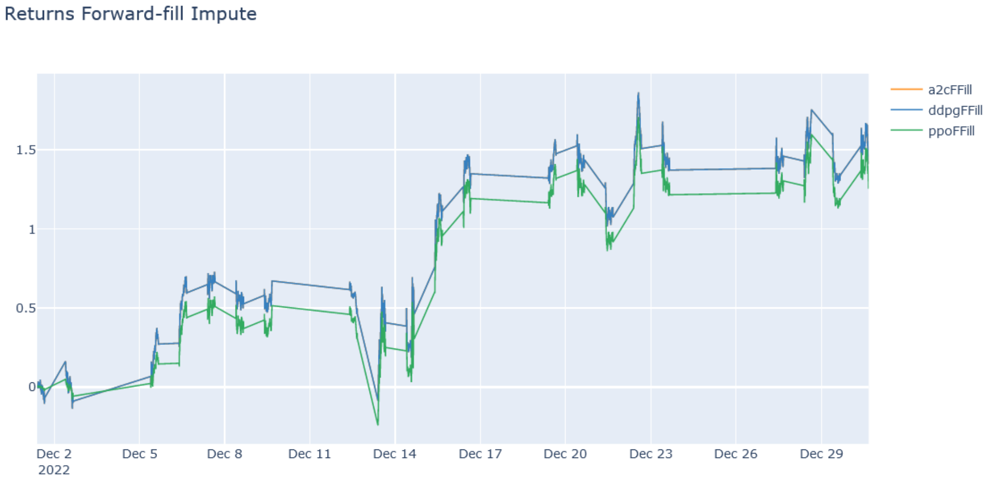


Figure 13

Now to look at the metrics, though the overall annual return was higher for the Forward-fill impute dataset, the Sharpe ratio is significantly lower. The volatility is higher in the KNN impute and the drawdown. This gives conflicting results, as they are not as clear as they should be. From the visual plots, it is pretty evident that the KNN impute dataset had far better performance than the Forward-fill impute dataset. Table 3 compares the imputation of non-hyper-tuned parameters.

Table 3

	KNN Imputation Deep Deterministic Policy Gradient Results	**Forward-fill Imputation Deep Deterministic Policy Gradient Results**
Annual return	1.32%	1.16%
Cumulative returns	53.54%	45.63%
Annual volatility	23.52%	347.28%
Sharpe ratio	0.18	1.41
Calmar ratio	0.02	0.01
Stability	0.30	0.22
Max drawdown	-67.33%	-83.68%
Omega ratio	1.05	1.56
Sortino ratio	0.26	2.97
Tail ratio	1.07	1.41
Daily value at risk	-2.95%	-41.81%

Finally, the parameters should be fine-tuned to see if the above results can be improved in terms of return and the Sharpe ratio, drawdown, and volatility. See the results below:

Table 4

	A2C KNN Imputation Results	DDPG KNN Imputation Results	PPO KNN Imputation Results	A2C Forward-fill Imputation Results	DDPG Forward-fill Imputation Results	PPO Forward-fill Imputation Results	Optimize DDPG KNN Imputation Results	Optimize DDPG Forward-fill Imputation Results
Annual return	1.16%	1.16%	1.16%	1.32%	1.32%	0.27%	1.16%	1.32%
Cumulative returns	45.63%	45.63%	45.81%	53.54%	53.54%	9.29%	45.63%	53.54%
Annual volatility	347.28%	347.28%	346.96%	23.52%	23.52%	7.37%	347.28%	23.52%
Sharpe ratio	1.41	1.41	1.40	0.18	0.18	0.07	1.41	0.18
Calmar ratio	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.02
Stability	0.22	0.22	0.22	0.30	0.30	0.00	0.22	0.30
Max drawdown	-83.68%	-83.68%	-83.68%	-67.33%	-67.33%	-30.40%	-83.68%	-67.33%
Omega ratio	1.56	1.56	1.56	1.05	1.05	1.02	1.56	1.05
Sortino ratio	2.97	2.97	2.97	0.26	0.26	0.10	2.97	0.26
Tail ratio	1.41	1.41	1.41	1.07	1.07	1.03	1.41	1.07
Daily value at risk	-41.81%	-41.81%	-41.78%	-2.95%	-2.95%	-0.93%	-41.81%	-2.95%

After looking at the results, it doesn't seem like the optimization of the hyperparameters made any difference. This is because there weren't enough parameters for optimization, and it took more than 10 hours to train and test the results, which means that computational costs must be considered. Even with the lack of optimization and the time taken to obtain the results, it is still valuable to review the results to understand why there was no improvement. Looking more closely at the results, it became apparent that the parameters chosen for optimization may have been too general to truly make an impact.

In this chapter, the research results have been presented, which aimed to evaluate the effectiveness of using one-minute stock data to develop a Reinforcement Learning decision agent and evaluate the decisions in the trading strategies. The analysis of the trading strategies that were developed and evaluated demonstrated the practical implications of the methodology and provided a comprehensive understanding of the dynamics of the financial markets. The findings have shown that the Reinforcement Learning approach can be used to build trading strategies with excellent performance metrics. However, certain limitations and constraints will be discussed in the next chapter.

Chapter 7: Discussion

Chapter seven of this thesis discusses the implications and limitations of this research, which aims to evaluate the effectiveness of using one-minute stock data to develop and evaluate Reinforcement Learning decision agent trading strategies. This chapter will interpret the findings of this research in the context of the research questions introduced in the first chapter. Furthermore, it identifies this research's possible financial industry implications and acknowledges the limitations and constraints. This includes the assumptions and constraints placed on the data and methodology. By discussing the implications and limitations of this research, the aim is to provide a balanced and objective perspective on the practical implications of the findings and to suggest avenues for future work.

Implications

This research has demonstrated that Reinforcement Learning decision agents can assist traders in their strategies and trading decisions. The number of stock options that traders can trade at any given time is limited, whereas reinforcement learning decision agents do not have a limit but are constrained by training time. Some Reinforcement Learning decision agents had higher volatility or drawdown levels in the strategy evaluation of the performance metrics. Still, there was a lower overall return on capital employed. Instead of discarding the other strategies, they could be used for different types of trading with various risk profiles for investment clients or portfolios. Some investors can take on more risk than others. There is plenty of room to employ all models in stock option trading. By diversifying the risk and combining strategies, there are ways to maximise the potential return on capital employed without exceeding acceptable risk levels.

Limitations

Several constraints and limitations must be considered when interpreting this research's results. First, this research may not represent the overall market because it was based on a small number of SPY ETF option puts data and a particular time frame. Other put strike prices and expiration dates could have been considered. Further study may be required to confirm these findings in different markets or contexts because the results of this study may not generalise to other put options, markets, or periods. Second, due to the one-minute SPY ETF and 380 strike put option data being used in this study, there may have been market noise and volatility in the data. The accuracy and dependability of the findings of this study may be impacted by the computational and technological difficulties associated with using this data. Thirdly, the use of particular imputation techniques, statistical analyses, performance metrics, and a set of presumptions and restrictions served as the foundation for this research. Additional research may be required to explore the sensitivity of these research findings to these decisions. Alternative methods and metrics may have produced different results. Finally, this research did not consider outside variables that could significantly impact the conclusions, like trading costs, execution mistakes, geopolitical events, economic trends, and regulatory changes. It might be necessary to conduct additional research to determine how these outside factors affect the efficacy of the methodology employed in this research. Overall, even though this research sheds light on how one-minute data can be used practically and how machine learning can be applied in finance, it is essential to recognise the constraints and limitations of this study and proceed with caution when interpreting the results.

This research acknowledges the restrictions and limitations of this research, such as the presumptions made—additionally, the constraints imposed on the data and methodology. The conclusion, which will summarise the key findings of this research, restate the research questions, and provide a discussion of the contributions and limitations of this research, will be presented in the final chapter of this thesis. Finally, discuss the ramifications of these research findings for the finance and investment industries and directions for future study. The hope is to demonstrate the value and applicability of this research by offering a thorough and fair conclusion. The hope, as well, is to add to the growing body of knowledge on applying machine learning techniques in finance.

Chapter 8: Conclusion

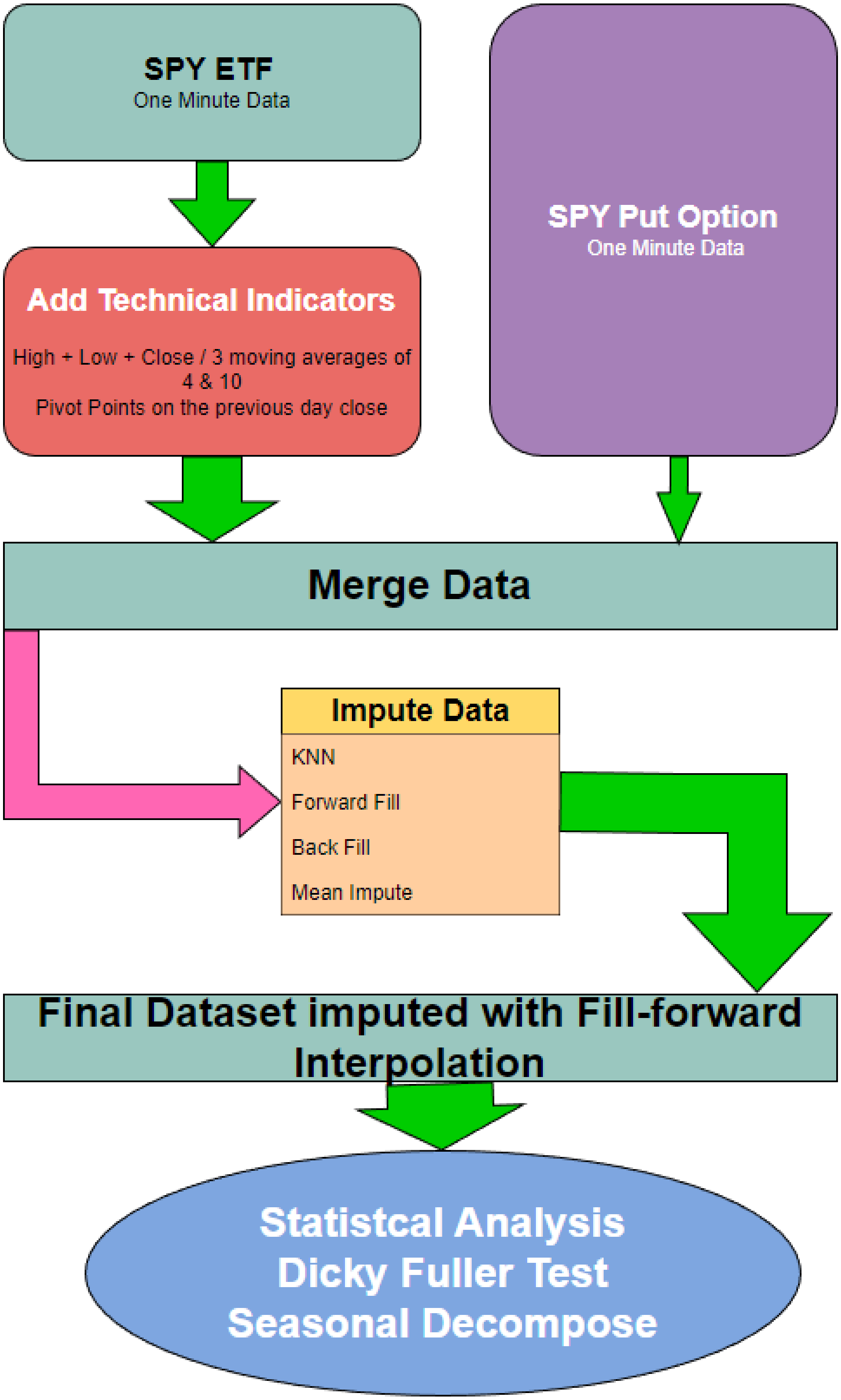
The results of this thesis indicate that a promising area of research is the application of reinforcement learning to finance. Reinforcement learning models offer traders and investors an intriguing possibility because they can draw lessons from the past and adapt to shifting market conditions. The outcomes of our backtesting show that the decision-making capabilities of reinforcement learning have the potential to support human traders and more effectively optimise trading strategies. However, more study is needed to confirm these encouraging findings, particularly in light of the missing data and imputation techniques that significantly impact the outcomes. The power of Reinforcement Learning combined with human experience and intuition may allow traders to use a much more effective tool. While human expertise and intuition could be used to create a reward function to train the Reinforcement Learning decision agent, technology could decrease human error and respond more quickly to shifting market conditions. Though successful automated trading strategies may result from reinforcement learning in finance, it's critical to remember that human expertise and intuition cannot be mechanically duplicated or reduced to a set of rules.

Nevertheless, the use of Reinforcement Learning in finance has the potential to introduce a novel trading strategy and give investors access to previously untapped opportunities, as this thesis' conclusion demonstrates. Therefore, it is essential to recognise this research's constraints and limitations mentioned in chapter 7 and conduct additional research to confirm the findings. The findings of this research are nonetheless intriguing, and the hope is that it will spur further research into and advancements in Reinforcement Learning in finance.

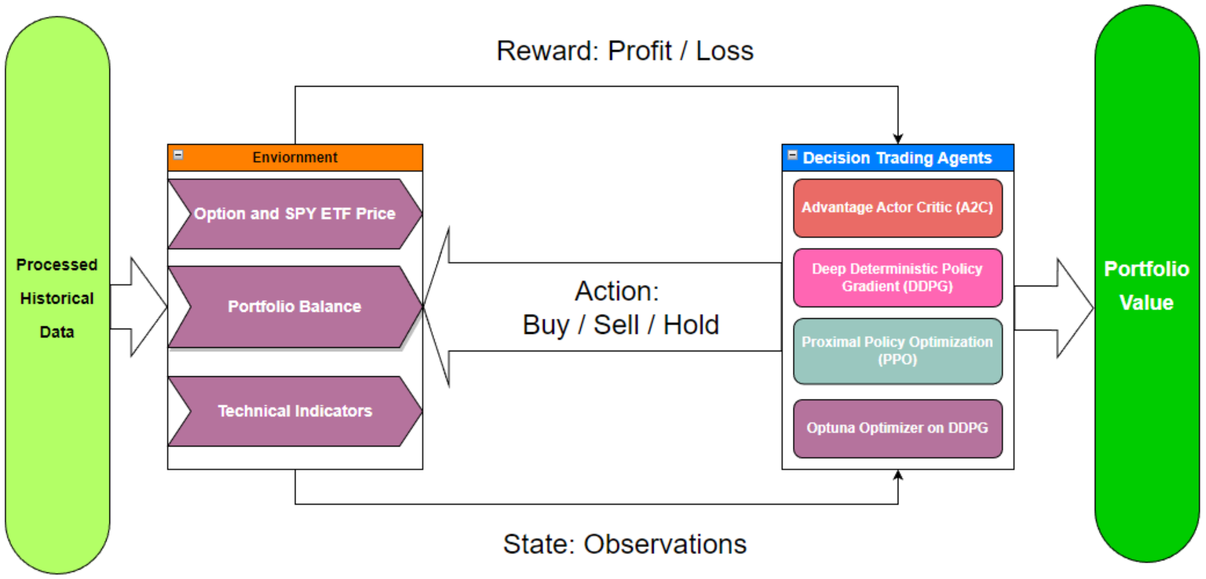
Additionally, the results of this thesis have significant applications in the area of finance and investment. Machine learning algorithms may change how traders and investors make trading decisions. Traders and investors could improve their understanding of market dynamics and create trading strategies that are more effective and efficient by using Reinforcement Learning models. Machine learning algorithms may also find trading opportunities that have gone unnoticed or unexploited. In this way, applying machine learning to finance could enhance the effectiveness and efficiency of financial markets while generating benefits for various stakeholders. This research's conclusions might not apply to other markets or periods. Additional research might be required to confirm them, mainly if the models are to be used in different segments. The Reinforcement Learning decision agents' cumulative returns range from 9% to 54%, a crucial indication that the models work successfully.

Moreover, the Sharpe ratios of 0.26 and 2.97 have lots of scope for use in the finance industry. Finally, machine learning algorithms' computational and technological challenges may also impact the findings. Despite these drawbacks, this research is a starting point for further investigation into innovative and effective trading methods. This research adds to the growing body of literature on developing and evaluating trading strategies using machine learning techniques. It emphasises the significance of considering the real-world applications of machine learning in finance.

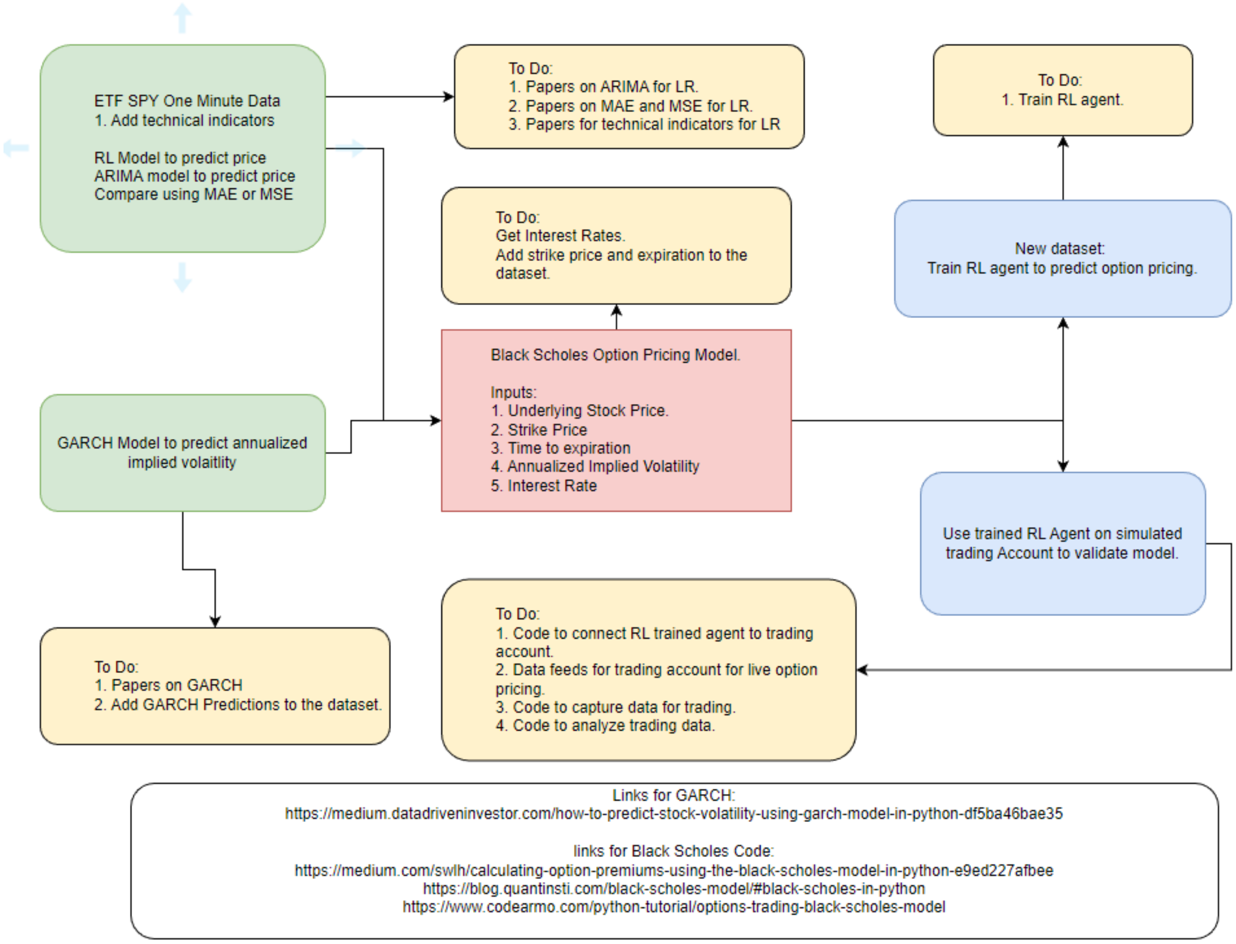
Appendix A: Workflow



Reinforcement Learng - Stock Option Trading Strategy



Appendix B: Previous Workflow



ETF SPY One Minute Data
1. Add technical indicators

RL Model to predict price
ARIMA model to predict price
Compare using MAE or MSE

To Do:
1. Papers on ARIMA for LR.
2. Papers on MAE and MSE for LR.
3. Papers for technical indicators for LR

To Do:
1. Train RL agent.

To Do:
Get Interest Rates.
Add strike price and expiration to the dataset.

New dataset:
Train RL agent to predict option pricing.

Black Scholes Option Pricing Model.

Inputs:
1. Underlying Stock Price.
2. Strike Price
3. Time to expiration
4. Annualized Implied Volatility
5. Interest Rate

GARCH Model to predict annualized implied volaitlity

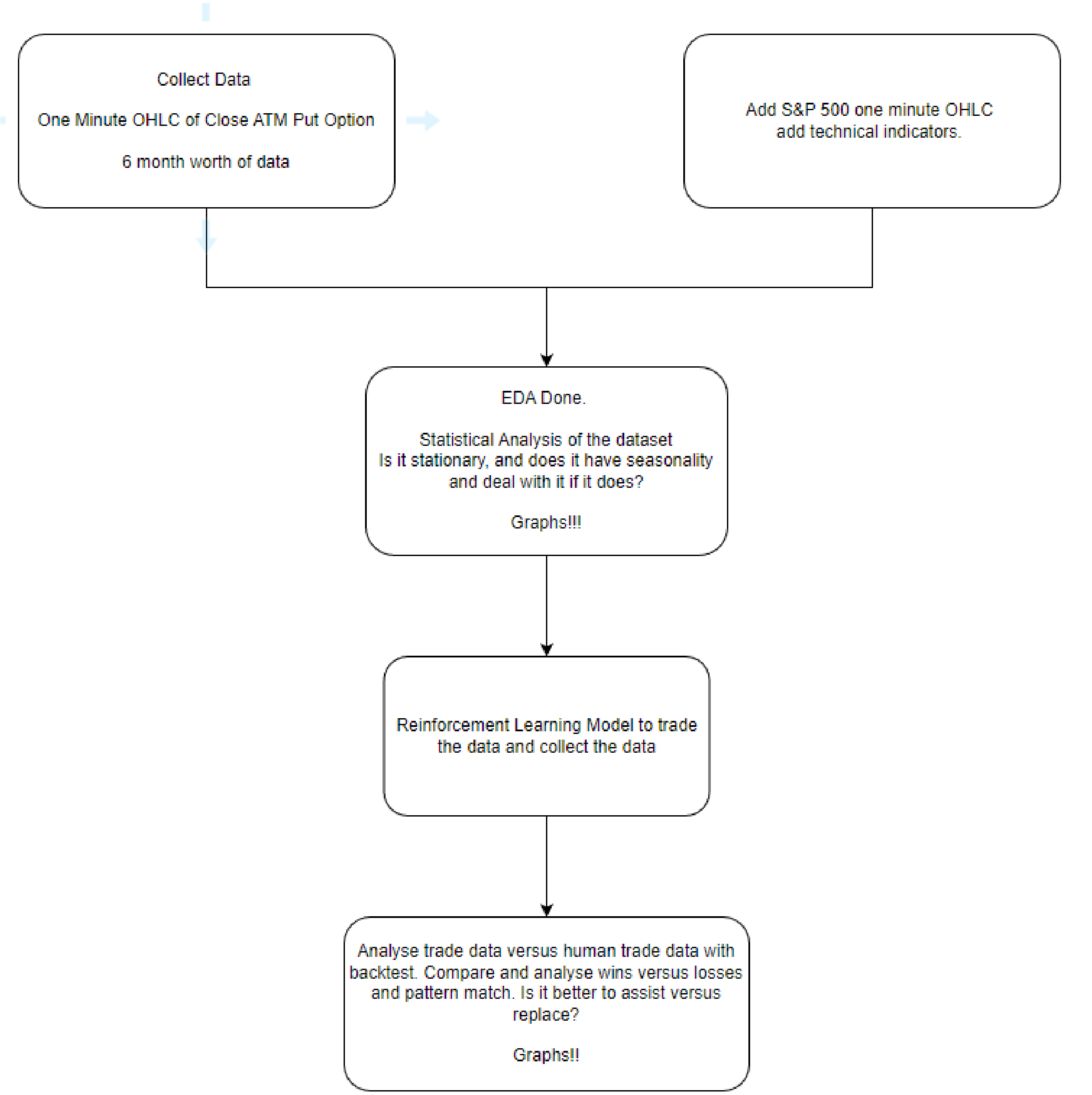
Use trained RL Agent on simulated trading Account to validate model.

To Do:
1. Code to connect RL trained agent to trading account.
2. Data feeds for trading account for live option pricing.
3. Code to capture data for trading.
4. Code to analyze trading data.

To Do:
1. Papers on GARCH
2. Add GARCH Predictions to the dataset.

Links for GARCH:
https://medium.datadriveninvestor.com/how-to-predict-stock-volatility-using-garch-model-in-python-df5ba46bae35

links for Black Scholes Code:
https://medium.com/swlh/calculating-option-premiums-using-the-black-scholes-model-in-python-e9ed227afbee
https://blog.quantinsti.com/black-scholes-model/#black-scholes-in-python
https://www.codearmo.com/python-tutorial/options-trading-black-scholes-model


Collect Data
One Minute OHLC of Close ATM Put Option
6 month worth of data
Add S&P 500 one minute OHLC
add technical indicators.
EDA Done.
Statistical Analysis of the dataset
Is it stationary, and does it have seasonality and deal with it if it does?
Graphs!!!
Reinforcement Learning Model to trade the data and collect the data
Analyse trade data versus human trade data with backtest. Compare and analyse wins versus losses and pattern match. Is it better to assist versus replace?
Graphs!!

Appendix C: Interview Transcripts

Dropbox Link:

https://www.dropbox.com/sh/8ncwskvwuvqperf/AAAxi9R2H1x_ryAtzGI_khdUa?dl=0

Appendix D: Data Permissions

Dropbox link:

https://www.dropbox.com/sh/hyq8g54kkitnozn/AAAteALULXY6RwQtbHv71n5Va?dl=0

Appendix E: Consent Forms

Dropbox link:

https://www.dropbox.com/sh/7twq744aaocvx1o/AABns43ajHJjbDOM6STEv5Tja?dl=0

Reference list

Agresti, A. and Kateri, M. (2021). *Foundations of statistics for data scientists : with R and Python*. Boca Raton: Crc Press.

Ahn, J.J., Kim, D.H., Oh, K.J. and Kim, T.Y. (2012). Applying option Greeks to directional forecasting of implied volatility in the options market: An intelligent approach. *Expert Systems with Applications*, 39(10), pp.9315–9322. doi:https://doi.org/10.1016/j.eswa.2012.02.070.

Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *arXiv:1907.10902 [cs, stat]*. [online] Available at: https://arxiv.org/abs/1907.10902.

Alexander Alexander Zai (2020). *Deep Reinforcement Learning in Action.* Manning Publications Company.

Almgren, R.F. (2003). Optimal execution with nonlinear impact functions and trading-enhanced risk. *Applied Mathematical Finance*, 10(1), pp.1–18. doi:https://doi.org/10.1080/135048602100056.

Amellas, Y., Bakkali, O.E., Djebli, A. and Echchelh, A. (2020). Short-term wind speed prediction based on MLP and NARX network models. *Indonesian Journal of Electrical Engineering and Computer Science*, 18(1), p.150. doi:https://doi.org/10.11591/ijeecs.v18.i1.pp150-157.

Amilon, H. (2003). A neural network versus Black-Scholes: a comparison of pricing and hedging performances. *Journal of Forecasting*, 22(4), pp.317–335. doi:https://doi.org/10.1002/for.867.

AN, B.-J., ANG, A., BALI, T.G. and CAKICI, N. (2014). The Joint Cross Section of Stocks and Options. *The Journal of Finance*, 69(5), pp.2279–2337. doi:https://doi.org/10.1111/jofi.12181.

Anders, U., Korn, O. and Schmitt, C. (1998). Improving the pricing of options: a neural network approach. *Journal of Forecasting*, 17(5-6), pp.369–388. doi:https://doi.org/10.1002/(sici)1099-131x(1998090)17:5/6%3C369::aid-for702%3E3.0.co;2-s.

Andrew, A.M. (1999). REINFORCEMENT LEARNING: AN INTRODUCTION by Richard S. Sutton and Andrew G. Barto, Adaptive Computation and Machine Learning series, MIT Press (BraADFord Book), Cambridge, Mass., 1998, xviii + 322 pp, ISBN 0-262-19398-1, (hardback, £31.95). *Robotica*, 17(2), pp.229–235. doi:https://doi.org/10.1017/s0263574799211174.

Avellaneda, M., Levy, A. and ParÁS, A. (1995). Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance*, 2(2), pp.73–88. doi:https://doi.org/10.1080/13504869500000005.

Avishek Pal (2017). *Practical time series analysis : master time series data processing, visualisation, and modeling using Python*. Birmingham: Packt.

Awan, A.A., Subramoni, H. and Panda, D.K. (2017). An In-depth Performance Characterisation of CPU- and GPU-based DNN Training on Modern Architectures. *Proceedings of the Machine Learning on HPC Environments*. doi:https://doi.org/10.1145/3146347.3146356.

Bakshi, G., Cao, C. and Chen, Z. (1997). Empirical Performance of Alternative Option Pricing Models. *The Journal of Finance*, 52(5), pp.2003–2049. doi:https://doi.org/10.1111/j.1540-6261.1997.tb02749.x.

BARONE-ADESI, G. and WHALEY, RE (1987). Efficient Analytic Approximation of American Option Values. *The Journal of Finance*, 42(2), pp.301–320. doi:https://doi.org/10.1111/j.1540-6261.1987.tb02569.x.

Bayraktar, E. and Young, V. (2007). Pricing options in incomplete equity markets via the instantaneous Sharpe ratio. *Annals of Finance*, 4(4), pp.399–429. doi:https://doi.org/10.1007/s10436-007-0084-0.

Bekiros, S.D. (2010). Heterogeneous trading strategies with adaptive fuzzy Actor–Critic reinforcement learning: A behavioral approach. *Journal of Economic Dynamics and Control*, 34(6), pp.1153–1170. doi:https://doi.org/10.1016/j.jedc.2010.01.015.

Bennell, J. and Sutcliffe, C. (2004). Black-Scholes versus artificial neural networks in pricing FTSE 100 options. *Intelligent Systems in Accounting, Finance & Management*, 12(4), pp.243–260. doi:https://doi.org/10.1002/isaf.254.

Berkowitz, J. (2009). On Justifications for the ad hoc Black-Scholes Method of Option Pricing. *Studies in Nonlinear Dynamics & Econometrics*, 14(1). doi:https://doi.org/10.2202/1558-3708.1683.

Bouchard, B. and Touzi, N. (2004). Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications*, 111(2), pp.175–206. doi:https://doi.org/10.1016/j.spa.2004.01.001.

Bruce, P.C., Bruce, A. and Gedeck, P. (2020). *Practical statistics for data scientists : 50+ essential concepts using R and Python*. Sebastopol, Ca: O'reilly Media, Inc.

Buehler, H., Gonon, L., Teichmann, J., Wood, B., Mohan, B. and Kochems, J. (2019). Deep Hedging: Hedging Derivatives Under Generic Market Frictions Using Reinforcement Learning. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3355706.

Chatzis, S.P., Siakoulis, V., Petropoulos, A., Stavroulakis, E. and Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications*, 112, pp.353–371. doi:https://doi.org/10.1016/j.eswa.2018.06.032.

Chiang, W.-C., Enke, D., Wu, T. and Wang, R. (2016). An adaptive stock index trading decision support system. *Expert Systems with Applications*, 59, pp.195–207. doi:https://doi.org/10.1016/j.eswa.2016.04.025.

Choquette, J., Gandhi, W., Giroux, O., Stam, N. and Krashinsky, R. (2021). NVIDIA A100 Tensor Core GPU: Performance and Innovation. *IEEE Micro*, 41(2), pp.29–35. doi:https://doi.org/10.1109/mm.2021.3061394.

Christoffersen, P.F. and Jacobs, K. (2003). The Importance of the Loss Function in Option Valuation. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.424461.

Combes, F., Fraiman, R. and Ghattas, B. (2022). Time Series Sampling. *ITISE 2022*. doi:https://doi.org/10.3390/engproc2022018032.

Crisan, D., Manolarakis, K. and Touzi, N. (2010). On the Monte Carlo simulation of BSDEs: An improvement on the Malliavin weights. *Stochastic Processes and their Applications*, 120(7), pp.1133–1158. doi:https://doi.org/10.1016/j.spa.2010.03.015.

Culkin, R. and Das, S. (2017). Machine Learning in Finance: The Case of Deep Learning for Option Pricing. *Journal of Investment Management*, 15(4)(92-100).

Deng, Y., Bao, F., Kong, Y., Ren, Z. and Dai, Q. (2017). Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3), pp.653–664. doi:https://doi.org/10.1109/tnnls.2016.2522401.

Dhillon, U.S., Lasser, D.J. and Watanabe, T. (1997). Volatility, information, and double versus walrasian auction pricing in US and Japanese futures markets. *Journal of Banking & Finance*, 21(7), pp.1045–1061. doi:https://doi.org/10.1016/s0378-4266(97)00012-5.

E, W., Han, J. and Jentzen, A. (2017). Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations. *Communications in Mathematics and Statistics*, 5(4), pp.349–380. doi:https://doi.org/10.1007/s40304-017-0117-6.

Enes Bilgin (2020). *Mastering Reinforcement Learning with Python*. Packt Publishing Ltd.

Gaspar, R.M., Lopes, S.D. and Sequeira, B. (2020). Neural Network Pricing of American Put Options. *Risks*, 8(3), p.73. doi:https://doi.org/10.3390/risks8030073.

Gastwirth, J.L., Gel, Y.R. and Miao, W. (2009). The Impact of Levene’s Test of Equality of Variances on Statistical Theory and Practice. *Statistical Science*, [online] 24(3), pp.343–360. doi:https://doi.org/10.1214/09-sts301.

Gatfaoui, H. (2015). Estimating Fundamental Sharpe Ratios: A Kalman Filter Approach. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.2838935.

Gencay, R. and Min Qi (2001). Pricing and hedging derivative securities with neural networks: Bayesian regularisation, early stopping, and bagging. *IEEE Transactions on Neural Networks*, 12(4), pp.726–734. doi:https://doi.org/10.1109/72.935086.

Giurca, A. and Borovkova, S. (2021). Delta Hedging of Derivatives using Deep Reinforcement Learning. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3847272.

Grus, J. (2019). *DATA SCIENCE FROM SCRATCH : first principles with Python.*.

Gupta, A. and Dhingra, B. (2012). Stock market prediction using Hidden Markov Models. *2012 Students Conference on Engineering and Systems*. [online] doi:https://doi.org/10.1109/sces.2012.6199099.

Gupta, A., Mishra, P., Pandey, C., Singh, U., Sahu, C. and Keshri, A. (2019). Descriptive Statistics and Normality Tests for Statistical Data. *Annals of Cardiac Anaesthesia*, [online] 22(1), p.67. doi:https://doi.org/10.4103/aca.aca_157_18.

Hariom Tatsat (2020). *MACHINE LEARNING AND DATA SCIENCE BLUEPRINTS FOR FINANCE : from building trading strategies to... robo-advisors using Python.* SL: O'reilly Media.

Hastie, T., Friedman, J. and Tibshirani, R. (2001). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. New York, Ny Springer New York Imprint: Springer.

Heinrich, S. (2006). The randomised information complexity of elliptic PDE. *Journal of Complexity*, 22(2), pp.220–249. doi:https://doi.org/10.1016/j.jco.2005.11.003.

Hellmuth, K. and Klingenberg, C. (2022). Computing Black Scholes with Uncertain Volatility—A Machine Learning Approach. *Mathematics*, 10(3), p.489. doi:https://doi.org/10.3390/math10030489.

Heshan Guan and Qingshan Jiang (2008). Pattern matching of time series and its application to trend prediction. *2008 2nd International Conference on Anti-counterfeiting, Security and Identification*. doi:https://doi.org/10.1109/iwasid.2008.4688342.

Hirchoua, B., Ouhbi, B. and Frikh, B. (2021). Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy. *Expert Systems with Applications*, 170, p.114553. doi:https://doi.org/10.1016/j.eswa.2020.114553.

Hornik, K. (1991). Approximation capabilities of multilayer feeADForward networks. *Neural Networks*, 4(2), pp.251–257. doi:https://doi.org/10.1016/0893-6080(91)90009-t.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feeADForward networks are universal approximators. *Neural Networks*, [online] 2(5), pp.359–366. doi:https://doi.org/10.1016/0893-6080(89)90020-8.

Hornik, K., Stinchcombe, M. and White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feeADForward networks. *Neural Networks*, 3(5), pp.551–560. doi:https://doi.org/10.1016/0893-6080(90)90005-6.

Hull, J.C., Cao, J., Chen, J., Farghadani, S., Poulos, Z., Wang, Z. and yuan, J. (2022). Gamma and Vega Hedging Using Deep Distributional Reinforcement Learning. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.4106814.

Hutchinson, JM, LO, AW and Poggio, T. (1994). A Non-parametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks. *The Journal of Finance*, 49(3), pp.851–889. doi:https://doi.org/10.1111/j.1540-6261.1994.tb00081.x.

Idrees, S.M., Alam, M.A. and Agarwal, P. (2019). A Prediction Approach for Stock Market Volatility Based on Time Series Data. *IEEE Access*, 7, pp.17287–17298. doi:https://doi.org/10.1109/access.2019.2895252.

Ilmanen, A. (2011). Expected Returns. doi:https://doi.org/10.1002/9781118467190.

Jang, H. and Lee, J. (2018). Generative Bayesian neural network model for risk-neutral pricing of American index options. *Quantitative finance*, 19(4), pp.587–603. doi:https://doi.org/10.1080/14697688.2018.1490807.

Kalpakis, K., Gada, D. and Puttagunta, V. (n.d.). Distance measures for effective clustering of ARIMA time-series. *Proceedings 2001 IEEE International Conference on Data Mining*. doi:https://doi.org/10.1109/icdm.2001.989529.

Kirkpatrick, C.D. and Dahlquist, J.A. (2010). *Technical Analysis*. FT Press.

Klibanov, M., Golubnichiy, K. and Nikitin, A. (2022). Application of Neural Network Machine Learning to Solution of Black-Scholes Equations. doi:https://doi.org/10.48550/ARXIV.2111.06642.

Kolm, P.N. and Ritter, G. (2019). Modern Perspectives on Reinforcement Learning in Finance. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3449401.

Liang, X., Zhang, H., Xiao, J. and Chen, Y. (2009). Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing*, 72(13-15), pp.3055–3065. doi:https://doi.org/10.1016/j.neucom.2009.03.015.

Liang, Z., Chen, H., Zhu, J., Jiang, K. and Li, Y. (2018). Adversarial Deep Reinforcement Learning in Portfolio Management. *arXiv:1808.09940 [cs, q-fin, stat]*. [online] Available at: https://arxiv.org/abs/1808.09940 [Accessed the 23rd of December 2022].

Liu, S., Leitao, Á., Borovykh, A. and Oosterlee, C.W. (2021a). On a Neural Network to Extract Implied Information from American Options. *Applied Mathematical Finance*, 28(5), pp.449–475. doi:https://doi.org/10.1080/1350486x.2022.2097099.

Liu, S., Oosterlee, C. and Bohte, S. (2019). Pricing Options and Computing Implied Volatilities using Neural Networks. *Risks*, 7(1), p.16. doi:https://doi.org/10.3390/risks7010016.

Liu, X.-Y. (2020). FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3737257.

Liu, X.-Y. (2022). FinRL-Meta: Market Environments and Benchmarks for Data-Driven Financial Reinforcement Learning. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.4253139.

Liu, X.-Y., Xiong, Z., Zhong, S., Yang, H. and Walid, A. (2022). Practical Deep Reinforcement Learning Approach for Stock Trading. *arXiv:1811.07522 [cs, q-fin, stat]*. [online] Available at: https://arxiv.org/abs/1811.07522.

Liu, X.-Y., Yang, H., Gao, J. and Wang, C. (2021b). FinRL: Deep Reinforcement Learning Framework to Automate Trading in Quantitative Finance. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3955949.

Livieris, I.E., Stavroyiannis, S., Pintelas, E. and Pintelas, P. (2020). A novel validation framework to enhance deep learning models in time-series forecasting. *Neural Computing and Applications*, 32(23), pp.17149–17167. doi:https://doi.org/10.1007/s00521-020-05169-y.

Lopez De Prado, M. (2018). *Advances in financial machine learning*. New Jersey: Wiley.

MACBETH, JD and MERVILLE, LJ (1979). An Empirical Examination of the Black-Scholes Call Option Pricing Model. *The Journal of Finance*, 34(5), pp.1173–1186. doi:https://doi.org/10.1111/j.1540-6261.1979.tb00063.x.

Merton, R.C. (1973). Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science*, 4(1), p.141. doi:https://doi.org/10.2307/3003143.

Merton, R.C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1-2), pp.125–144. doi:https://doi.org/10.1016/0304-405x(76)90022-2.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, [online] 518(7540), pp.529–533. doi:https://doi.org/10.1038/nature14236.

Moody, J. and Saffell, M. (2001). Learning to trade via direct Reinforcement. *IEEE Transactions on Neural Networks*, 12(4), pp.875–889. doi:https://doi.org/10.1109/72.935097.

Mostafa, F., Dillon, T. and Chang, E. (2015). Computational Intelligence Approach to Capturing the Implied Volatility. *IFIP Advances in Information and Communication Technology*, pp.85–97. doi:https://doi.org/10.1007/978-3-319-25261-2_8.

Müller, A.C. and Guido, S. (2017). *Introduction to machine learning with Python : a guide for data scientists*. Beijing: O'reilly.

Nielsen, A. (2019). *Practical time series analysis : prediction with statistics and machine learning*. Sebastopol, Ca: O'reilly Media, Inc.

Pavel, M.I., Muhtasim, D.A. and Faruk, O. (2021). Decision Making Process of Stock Trading Implementing DRQN And ARIMA. *2021 IEEE Madras Section Conference (MASCON)*. doi:https://doi.org/10.1109/mascon51689.2021.9563476.

Phil Winder Ph.D (2020). *Reinforcement Learning*. O'Reilly Media.

Ritter, G. (2017). Machine Learning for Trading. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3015609.

Ritter, G. and Kolm, P.N. (2018). Dynamic Replication and Hedging: A Reinforcement Learning Approach. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3281235.

Ruf, J. and Wang, W. (2020). Neural networks for option pricing and hedging: a literature review. *The Journal of Computational Finance*. doi:https://doi.org/10.21314/jcf.2020.390.

Sagiraju, H.K. and Mogalla, S. (2021). Application of multilayer perceptron to deep reinforcement learning for stock market trading and analysis. *Indonesian Journal of Electrical Engineering and Computer Science*, 24(3), p.1759. doi:https://doi.org/10.11591/ijeecs.v24.i3.pp1759-1771.

Schulman, J., Wolski, F., Dhariwal, P., RaADFord, A. and Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. [online] arXiv.org. Available at: https://arxiv.org/abs/1707.06347.

Sharpe, W.F. (1994). The Sharpe Ratio. *The Journal of Portfolio Management*, 21(1), pp.49–58.

Shumway, R.H. and Stoffer, D.S. (2017). *Time series analysis and its applications : with R examples*. Cham, Switzerland: Springer.

Sirignano, J. and Spiliopoulos, K. (2017). Stochastic Gradient Descent in Continuous Time. *SIAM Journal on Financial Mathematics*, 8(1), pp.933–961. doi:https://doi.org/10.1137/17m1126825.

Strong, RA (2005). *Derivatives : an introduction*. Mason, Ohio Thomson/South-Western.

Sutton, R.S. and Barto, A. (2018). *Reinforcement learning : an introduction*. Cambridge, Ma ; Lodon: The Mit Press.

Théate, T. and Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, 173, p.114632. doi:https://doi.org/10.1016/j.eswa.2021.114632.

Thomaidis, N.S., Tzastoudis, V.S. and Dounias, G.D. (2007). A Comparison Of Neural Network Model Selection Strategies For The Pricing Of S&P 500 Stock Index Options. *International*

Journal on Artificial Intelligence Tools, 16(06), pp.1093–1113. doi:https://doi.org/10.1142/s0218213007003709.

Trønnes, H. (2018). Pricing Options with an Artificial Neural Network: A Reinforcement Learning Approach. *NTNU*, 52.

Velicer, W.F. and Fava, J.L. (2003). Time Series Analysis. *Handbook of Psychology*. doi:https://doi.org/10.1002/0471264385.wei0223.

Wang, Z., Schaul, T., Hessel, M., Hasselt, van, Lanctot, M. and de Freitas, Nando (2015). *Dueling Network Architectures for Deep Reinforcement Learning*. [online] arXiv.org. Available at: https://arxiv.org/abs/1511.06581.

Wei, X., Xie, Z., Cheng, R., Zhang, D. and Li, Q. (2020). An Intelligent Learning and Ensembling Framework for Predicting Option Prices. *Emerging Markets Finance and Trade*, 57(15), pp.4237–4260. doi:https://doi.org/10.1080/1540496x.2019.1695598.

www.cboe.com. (n.d.). *About Us*. [online] Available at: https://www.cboe.com/about/ [Accessed the 22nd of September 2022].

Yang, H., Liu, X.-Y., Zhong, S. and Walid, A. (2020). Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3690996.

Ye, T. and Zhang, L. (2019). Derivatives Pricing via Machine Learning. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3352688.

Zhang, G.Peter. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, [online] 50, pp.159–175. doi:https://doi.org/10.1016/s0925-2312(01)00702-0.

Zhang, Z., Zohren, S. and Roberts, S. (2019). *Deep Reinforcement Learning for Trading*. [online] arXiv.org. Available at: https://arxiv.org/abs/1911.10107 [Accessed the 30th of January 2022].