AUTOMATING THE CREATION OF TRADING STRATEGIES USING DEEP REINFORCEMENT LEARNING: ALGORITHMIC TRADING

by

Virendra Kumar Yadav, MSc.Artificial Intelligence and Machine Learning

DISSERTATION

Presented to the Swiss School of Business and Management Geneva

In Partial Fulfillment

Of the Requirements

For the Degree

DOCTOR OF BUSINESS ADMINISTRATION

SWISS SCHOOL OF BUSINESS AND MANAGEMENT GENEVA

FEBRUARY, 2024

AUTOMATING THE CREATION OF TRADING STRATEGIES USING DEEP REINFORCEMENT LEARNING: ALGORITHMIC TRADING

by

Virendra Kumar Yadav

APPROVED BY

____

Dissertation chair

RECEIVED/APPROVED BY:

____

Admissions Director

Dedication

I dedicate my dissertation work to my parents, my wife Pooja Yadav, my son Kinshuk, Kevit, my friends, and India's young generation. I dedicate this work to those unable to pursue their higher education dream because of their social and financial status.

I am also dedicating my work to God, my creator, a source of inspiration, wisdom, knowledge, and understanding. God Almighty has been the source of my strength throughout this program.

Acknowledgements

My deepest gratitude goes to God and my supervisor Dr George E. Iatridis, for providing all needed to complete this project and the program. There was never a lack or want. Throughout this entire study, He took care of everything that would have stopped me in my tracks and strengthened me even through my most difficult times.

I want to express my gratitude and appreciation to those who have helped me in this work, or I need to continue the spirit. I am grateful for the love, patience, and support of my family, wife, and friends during this time. I want to thank my supervisors, Dr George E. Iatridis and the Swiss School of Business and Management Geneva, for their encouragement and support throughout this project. The biggest challenge in my research was finding a useful stock dataset.

The data used in this research was made available by various online platforms for public usage. I thank my supervisor for providing insight and expertise that helped me understand and use the datasets.

My sincere appreciation also goes to Dr Rupal Bhargava for taking a session on paper publishing, Dr Snehansu Sahu on Literature Review, Dr Emilyn Costa for teaching me technical aspects of persuasive thesis writing, and Dr Kyongsik Yun for sharing his experience in the field of research.

My gratitude goes to my colleagues and fellow learners at upGrad and SSBM making my DBA enjoyable. I am grateful for our numerous discussions about various topics related to our dissertation, research area, and industry trends. Finally, I thank upGrad for introducing me to this program and providing all support, guidance, and help needed to complete this program.

ABSTRACT

AUTOMATING THE CREATION OF TRADING STRATEGIES USING DEEP REINFORCEMENT LEARNING: ALGORITHMIC TRADING

Virendra Kumar Yadav

2024

Dissertation Chair: Dr. Iva Buljubasic

Co-Chair: Dr. Ljiljana Kukec

As before, the RL state space's dimensionality is probably too low for the agent to learn a trading algorithm that is highly adaptable and appropriate for a variety of volatility regimes. The VG training data suggests that the best course of action in this case is to apply "more of the same". The main problem will now be persuading the RL agent to make use of the additional information that the 25 multi-asset futures contracts obviously provide. Third, more investigation is required to comprehend why evaluation frequency, with semi-annually being the terrible exception, is so important in preventing catastrophes like the one in 2008. In numerous respects, this appears strange. Fourth, the one strategy that has worked. Since investors are often risk adverse, the objective is to maximise a risk-adjusted performance function, such as the Sharpe ratio. This really results in a concave utility function. After learning the distribution, we may select actions that have the highest predicted Q-value and the lowest standard deviation from it to maximise the Sharpe ratio. The mean-variance portfolio theory and reward functions may be used to get an excellent anticipated algorithmic trade return with minimal volatility.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I:

INTRODUCTION

1. INTRODUCTION

1.1 Financial Trading Strategy System

Due to the complicated and varied investment aims, as the stock market develops, the effectiveness of artificially subjective investing modes constantly declines. The previous subjective investing increasingly supplanted builds investment plans using data and models, thanks to advancements in data science and statistical methodology. By integrating open market data with statistical approaches to choose companies with investment value, a new investing model somewhat mitigates the subjective influence of humans. The multifactor model, which is now the most popular quantitative stock selection methodology, is based on identifying the components that have the best connection with the stock return rate and may, to some extent, forecast the stock return. However, researchers discovered via empirical testing that it was unable to provide investors with consistent profits because of its poor forecast accuracy and unstable prediction outcomes. The financial market is also a dynamic system with high complexity, including long- and short-term fluctuations and linear and nonlinear information. Scholars discovered this through empirical research using financial market data. The formation and change of stock price involve a variety of uncertain factors, and there are complex relationships among them.

The use of machine learning algorithms in the study of financial time series has drawn a lot of attention from academics who want to better understand and analyse financial data as well as make predictions that are more accurate. Machine learning algorithms, as opposed to linear models, take into consideration the nonlinear connection between variables. It excels at handling large data, notably the enormous volume of financial data since it does not need to be based on the assumptions of independence and specific distribution and has more flexibility and efficiency. First, it departs from the conventional research framework of increasing the number of explanatory variables in a multifactor model to increase its explanatory power, and it provides a new trading strategy system by incorporating one of the most recent machine learning algorithms, LightGBM, into the system.

LightGBM can capture the nonlinear relationship between pricing factors without taking into account the precise distribution format of financial data, and the Exclusive Feature Bundling method can address the issue of sparsity in high-dimensional characteristic matrices and demonstrate the accuracy of stock return predictions. Second, the new system may be used to calculate a factor's significance score. It clearly demonstrates the effects that various factors have on stock return, which has a significant practical impact on stock selection.

Thirdly, they create a mean-variance model with a CVaR constraint for the stock position allocation module and evaluate the trading strategy system using real data from the Chinese A-share market.) The outcome of the experiment demonstrates that the system may offer investors a consistent excess return and offers a rationale for and practical implications for China's stock market (Yanjun Chen et al.).

These strategies include multiple repetitions throughout the trading session, are short-lived, and have a clear objective as some of its primary characteristics. Additionally, the majority of the efficacy of algorithmic trading strategies is dependent not only on microstructural elements but also on financial considerations (such as the asset's value). Crucial element effectiveness strategies statistically quantified because they are applied consistently and extensively. The trading business has recently been driven toward automation and a greater emphasis on trading algorithms by a number of causes.

Market fragmentation:

Regulations like MiFID and RegNMS that have liberalised the "exchanges market" have allowed the emergence of new electronic market-places where assets may be exchanged (which before was centred on principal exchanges) and the introduction of new market players (such as high-frequency traders).

• Electronic access: Most users access exchanges electronically (specially in equities, and increasingly more for other asset classes). Nowadays, trading occurs with extremely little delay, and any form of plan must include an automated approach (short term and long term).

• Technology: Advances in processing power and the vast volumes of data that can now be gathered in real-time inspire new methods for handling the optimization process, such as through the use of sophisticated control techniques and data mining and machine learning algorithms. (Joaquin Fernandez-Tapia).

1.2 Strategic Analysis of Markets Method (SAMM)

The SAMM makes the assumption that market inefficiencies exist and can be found and taken advantage of through a six step process. It is an effective approach because it uses strategic thinking and human behaviour as its pillars. Unlike the efficient market hypothesis, it does not make irrational assumptions. Because the market game is only played by skilled gamblers, it necessitates knowledge of gambling theory. To describe how markets change over time, a strategic life cycle model (the POPP) is necessary. Additionally, it necessitates the explicit inclusion of humans in behavioural finance, the study of financial decision making. The SAMM is created by rationally combining these components.

Six steps make up the SAMM:

(1) To create trading ideas, choose probable price distorters (described below).

(2) Compile datasets to examine the effects of price distortion on prices.

(3) Create strategic strategies in response to any substantial pricing implications that Step 2 revealed.

(4) Transform strategic plans into trading algorithms to produce Potentially Profitable Gambling Systems (PPGSs).

(5) Backtest trading algorithms and repeat Steps (3) through (5) as necessary.

(6) After Steps (1) through (5) are finished.

1.3 A strategic plan

Strategies are frequently informal action planning schemes. For instance, a strategy like "In a bull market, buy popular, high-beta stocks; in a bear market, selectively short formerly popular stocks but hold them only for a short period" necessitates determining the type of market—bear or bull—as well as the standards for determining "popularity". Algorithmic traders cannot utilise it since it is not detailed enough. May be made profitable by choosing certain parameters is what they refer to as traded algorithmically, a strategic plan.

1.4 The Pursuit of Profits Paradigm (POPP)

"Chase after riches", the foundation of Pursuit of Profits Paradigm (POPP). Perhaps surprise, chasers lose money since the chase is so predictable (save for Warren Buffett)! According to Steven D. Moffitt's description of the POPP's history in Table 1, there are four fairly random phases.

Table 1.1: Phases in the Pursuit of Profits Paradigm (POPP) (Steven D. Moffitt)

Phases	Names	Description
(A)	Eureka!	Discovery of the Strategic Plan by just few innovators.
(B)	Early Copycat	A sustainable strategic plan spreads.
(C)	Late Copycat	Strategic Plan spreads uncontrollably with supportive feedback.
(D)	Crash	Strategic Plan fails as a result of an unforeseen cause.

1.5 Market Direction Prediction problems

In the past ten years, artificial intelligence has been widely used in a variety of fields, and its use has been shown to significantly enhance outcomes. Financial markets are one of the applications that is quite intriguing. Using artificial intelligence and machine learning to exploit these marketplaces might lead to significant advances. Loan credit scoring, credit evaluation, sovereign credit ratings, mortgage choice decisions, portfolio management, financial performance prediction, and market direction prediction are a few examples of these uses. The conclusions are startling and provocative. First off, consumer acceptance of various cryptocurrencies has truly exploded, with millions of 'active' wallets and a market worth in the billions in 2016.

In addition to borderless exchange operations and regionally concentrated mining operations, the cryptocurrency sector is also localised. Third, the business is getting more fluid as the distinctions between exchanges and wallets become more "blurred" and many cryptocurrencies, other than just bitcoin, are now backed by a developing ecosystem that serve a variety of purposes. (Asgari, M., Khasteh, H., 2021)

1.6 Using Deep Reinforcement Learning To Automate Indian Stock Trading

The stock market is always changing and there is a lack of accurate and adequate data, predicting stock prices is a very difficult assignment. One technique for resolving such difficult decision-making issues is reinforcement learning. When it comes to predicting stock prices and optimising anticipated return, reinforcement learning may show to be a superior alternative strategy. Deep Learning techniques can take features out of highly dimensional data. However, it is incapable of forming decisions. Deep Learning methodology capacity for decisions. To address the algorithmic trading issue, researchers have looked into RL approaches. Without the requirement to create forecasting models, the Recurrent Reinforcement Learning (RRL) algorithm has been utilised to find new investing strategies.

Trading in foreign exchange markets has been done using adaptive reinforcement learning (ARL). People have recently looked into the DRL approach to address the algorithmic trading issue.

1.7 High-Frequency and Quantitative Trading

Prior to the automation of trading platforms in the financial sector, the "cacophony of the marketplace and apparent randomness of trade" was primarily managed through human sociality. Today, however, that is done by "managing the punctuated electronic signals that encode the orders from masses of anonymous investors," which is accomplished by "toying with the nimble algorithms, sophisticated computer processors, hacked routers, and specialised telecommunication systems that are the materia mentis." Even though all orders must be completed using automated systems and algorithmic trading makes up the vast bulk of trading in financial markets, manual trading is still permitted. Algorithmic trading, according to Kirilenko and Lo, is "the automation of the purchase and sale of financial assets via the use of mathematical models, computers, and telecommunications networks."

"A sequence of achievements in the quantitative modelling of financial markets," "an almost simultaneous set of developments in computer technology" have all contributed to its ascent during the past two and a half decades. According to Beverungen (2019), trading is mostly carried out by algorithms, markets have been automated.

1.9 Machine learning

Programming computers to automatically learn and get better over time is called machine learning. While algorithms that are efficient for certain learning tasks have been developed and a theoretical knowledge of learning has emerged, computers still cannot learn as effectively as people can at this time. The development of computer algorithms that demonstrate practical learning, particularly in speech recognition and data mining As a result, new computational intelligence approaches integrating components of learning, evolution, and adaptability were created with the goal of automatically creating lucrative portfolios utilising technical analysis indications.

Neural networks, swarm intelligence, fuzzy systems, and evolutionary computing, in particular, may be used to the financial markets in a number of ways, such as predicting future stock price movement or optimising a collection of investment assets (funds and portfolios). These strategies presuppose the existence of stock return patterns that may be taken advantage of through historical examination of stock prices, returns, and other significant indicators. In order to create apps that can automatically manage a portfolio, new methodologies may be used to financial markets thanks to the quick advancement of computer science technology. As a result, there is a great deal of interest in and potential motivation for creating automated programmes that would trade in the market similarly to a technical trader while being somewhat independent.

Technical analysis was the basis for the development of mechanical trading systems (MTS). It is an algorithm that has been theoretically described and is intended to assist the user in making objective trading decisions based on historically recurring events.

In his presentation, financial markets and discussion of the characteristics of asset returns, Bloch, D. A. (2018) focuses on the long-range dependency (LRD) and the challenges associated with its empirical measurement. Then, they discuss artificial neural networks (ANN) and recurrent neural networks after introducing neural networks (RNN). They describe the learning process quickly and take into account prediction problems for chaotic series prediction (regression issue), autonomous pattern development, and more (classification problem). In order to learn more about neural networks' capacity to replicate the dynamics of financial time series, Bloch, D. A. (2018) tested them on the Mackey-Glass signal and a number of real-time financial time series. Reservoir Computing (RC), a technique created to quickly and effectively train and analyse RNNs, is the method they use to represent transient processes.

Only between the State units and the Output units are learning-free recurrent connections, and adaptation is based on a straightforward associative learning process. Since feeding the output signal back into the reservoir enables the ability to handle prediction tasks, RC models with output feedback were proposed to produce financial time series with intriguing features. They also take into account a running online algorithm that acts as an adaptive filter in order to increase the prediction returns' accuracy.

1.10 Quantitative Trading With Machine Learning

Since the idea that the distribution of price dynamics is wholly random is now under review, the efficient market hypothesis (EMH) and the growing evidence against it have reignited interest in the claims of technical analysis (TA). Across a wide variety of investment horizons, the frequency distribution of returns often features a fat tail and a high peak. Furthermore, compared to the square root of time, the standard deviation of returns rises quicker. Correlations are stochastic as a result, and volatility is extremely unstable. In light of these findings, Bloch, D. A. (2018) described the financial market as a system with fat tails, stable distributions, and persistence. Trading involves studying the technical factors that control short-term market movements as well as the market's behaviour, whereas investing entails studying the market's fundamentals, which may take several years to be reflected in the market.

While trading offers the potential for greater profits, it is riskier than long-term investing. There has been a significant amount of research on the mathematical analysis of the behaviour of stock prices, stock markets, and successful trading strategies in these environments. Assuming that historical market data yields accurate forecasts of the market's future performance, technical analysis (TA) focuses on indications designed to assist traders in determining if current behaviour is indicative of a certain trend and the timing of a prospective future transaction. However, since there isn't a superior indication, the signs should be blended to offer a range of viewpoints. Additionally, a technical indication must always be applied to a time window. The difficulty of figuring out the appropriate time frame may be solved by using advanced search and evaluation techniques.

1.11 The analysis using Associative Reservoir Computing

Given that an associative link between two entities is bidirectional, the reverse relation (one-to-many) is unclear if the forward relation is many-to-one. If there are several causes that lead to the same consequence, then this is always the case. Formally speaking, there is no one-of-a-kind solution to the inverse issue if there is a non-injective relation between the inputs and outputs. This issue arises often in financial time series, and conventional feed-forward models are unable to represent alternative mapping solutions without extra information. When dealing with persistent processes, they take into account how the Associative Reservoir Computing Theory (ARC) may be used to create multistable attractor dynamics that accurately reflect the variety of possible solutions. To do this, they replace the forward representation for the reservoir with a combined representation that includes input and output.

In other words, to create a matrix mapping the reservoir states to the input layer and learn a mapping from output to input, they rely on a feedback matrix to access the latter. The Mackey-Glass signal and FX time series are then used to evaluate the model. Due to their multifractal nature, financial time series prediction is a very difficult problem. Even when the fundamental links are unknowable or difficult to articulate, models are necessary to capture nuanced functional relationships among the empirical data.

Fortunately, these connections are not necessary in order to forecast future trends. This is so because the direction of an underlying asset relates to how the asset price will move or how it will generally fluctuate in the future. As a result, the evolution of a characteristic through time is more significant than its absolute value. In addition, networks are less complex when trends are forecasted rather than price levels. Various forecasting systems for the underlying asset's directions. While accuracy is a useful indicator of a network's overall performance, it must be used in conjunction with return metrics that take into account the size of the trends that a system finds. A framework for predicting future price levels and asset price orientations concurrently is one option.

A formula to predict the direction and magnitude of market returns. In conclusion, we formulate a set of guidelines for our trading algorithm construction based on the findings. It makes sense to concentrate on mastering the conventional forms that stock analysts have identified since stock prices are made up of many patterns that recur periodically. Creating patterns autonomously and predicting them with an eye on predicting price trends, where the patterns are unique configurations of market prices It is now well accepted that temporal difference (TD) approaches are far superior than supervised learning techniques for multi-step prediction-learning tasks. In order to learn to forecast financial patterns, which are multi-step prediction-learning issues, they also employ temporal differences (TD) approaches. Additionally, they demonstrate how to employ TD techniques to deduce details on the size of potential market returns.

1.12 Deep Reinforcement Trading with Predictable Returns

The estimate of an optimum strategy, which necessitates projecting financial quantities like risks and returns for numerous periods in the future, is made even more challenging by the addition of the time dimension. Single-period models are still often used since their dynamic counterparts are impractical, and forecasting might result in systematic mistakes because of uncertainty around the model of choice or the poor signal-to-noise ratio that exists by nature in the financial data. Even when a multi-period model is successful in capturing market impact or alpha decay, traditional optimal control approaches rely on a number of constricting assumptions that are unsuitable for simulating the real financial world. Recent DRL algorithms are frequently uncontrolled homemade concoctions. Because of this, analysing their performance in actual financial trading problems involves a complex interplay of various effects, some of which are related to the quality of the dataset and the signals used to forecast returns, while others are connected to the particular algorithm and problems with its trainability.

To the best of our knowledge, there aren't many research studies that look at how DRL performs in financial trading difficulties other than those caused by market efficiency: the hunt for a reliable signal to forecast returns or the potential absence of any signals in the dataset. Because of this, they take into account a controlled setting where a signal is known to exist and investigate the capacity of DRL agents to spot lucrative business prospects. Our work's key innovation is how they use a data-driven DRL environment in which agents may use their expertise to improve the state-action space and accelerate learning in addition to competing against traditional tactics. They evaluate several DRL techniques on a range of financial data with various characteristics to examine their flexibility when the simulated dynamics is incorrectly stated in relation to the benchmark model's assumptions.

When a model is misspecified, such as when there are severe events and volatility clustering, DRL algorithms can achieve the benchmark strategy's performance or even surpass it. By providing DRL agents with knowledge about the usual size of a decent strategy to start with and change, they also demonstrate how classical tactics may be helpful (Brini, A., & Tantari, D. (2021)).

1.13 Dataset

Weekends and holidays are not trading days, hence they are not included in any of the studies done in this work. All equities on the NASDAQ market have a minimum tick size of $0.0001 ($). Additionally, LOBSTER offers the derived LOB aggregate, which serves as the work's input data. Futhermore, the LOB is dense, allowing for the transmission of volumes to the DRL agent alone without data loss, and each tick has a size that is relevant.

The training dataset is made up of 60 files in total (one per trading day between 04-02-2019 and 30-04-2019). The validation dataset has 22 records (one per trading day between 01-05-2019 and 31-05-2019). There are 20 files total in the test dataset (one per trading day between 03-06-2019 and 28-06-2019). This is carried out to make it easier for the agent to learn in typical market circumstances (Briola, A., Turiel, J., Marcaccioli, & Aste, 2021).

1.14 pairs trading strategy

One of the typical trading strategies used by statistical arbitrageurs is pairs trading. In addition to being a popular trading strategy, pairs trading is also simple to apply. Two equities are identified by a stock arbitrager as having prices that follow one another historically. If the pair prices diverge sufficiently, the approach calls for shorting the increasing-price security while simultaneously buying the declining-price asset. The idea behind the pair trade is to make money from convergence factors that favour long-term historical pricing correlations over short-term price changes. In a pricing system that is relatively efficient, simple solutions based on mean-reversion assumptions shouldn't always result in gains. However, Broussard, J. P., and Vaihekoski found that trading document pairings generates continual arbitrage gains in the U.S. equity market, which is thought to be the most efficient and liquid in the whole world (2012). According to Broussard, J. P., and Vaihekoski (2012), although pairs trading profitability has usually decreased, the approach still produces significant performance during choppy market timeframes, such as the most recent financial crisis.

Surprisingly few research have looked at pair trading in markets other than the US since the release of Broussard, J. P., & Vaihekoski (2012). Who assesses the profitability of pairs trading in Taiwan and Broussard, J. P., & Vaihekoski (2012), who study pairs trading in Brazil, are two outliers. Despite the technique's seeming simplicity, this may be because it is difficult to anticipate gains from pairs trading computationally. They examine the function of the pairs trade start criterion in this study as well as the impact on returns of requiring trading in the pairs' chosen equities on days when the market is less volatile. They also go over certain points that were left out of Broussard, J. P., & Vaihekoski's discussion of the strategy's practical application (2012). They also provide the source code for the implementation. The pairs trading strategy is lucrative in Finland, a stock market with far less liquidity than the one in the United States, they demonstrate in our last section. Because the pairs trading approach involves two transactions rather than one, examining performance in a less liquid market strengthens the case that the strategy is universally applicable and efficient.

There are two more reasons why analysing the pairs trading strategy in Finland is fascinating. First off, the assessment period, 1987–2008, covers the global technology boom and fall as well as the financial institution crises in 1990 and 2008 that had an impact on stock purchasers in Finland. It is crucial to take into account how the pairs trading strategy works in such turbulent market situations when assessing its potential to function as a risk management alternative to traditional strictly directional trading approaches. Second, a unique institutional characteristic of Finnish ordinary and preferred shares enables a more in-depth analysis of potential causes for price fluctuation revealed by assets that are not only comparable but also have access to the same cash flow source.

The results show that the pairs trading strategy consistently works, even in markets with low liquidity. Profiting on price discrepancies between common and preferred shares of the same corporation also seems to produce large profits. The latter result significantly adds to the body of research since one of the arguments used to support lucrative trading opportunities brought on by pair price divergence is trader risk aversion with reference to the unpredictable nature of fundamental value computation. Prices may vary often if the trader's assessment of fundamental value is highly ambiguous since no deals are entered into in order to profit from mispricing. As a result, the risk aversion of the trader may diminish the profit potential of the pairs trading strategy since inaction may lead to non-convergence. The fact that profits may be produced through price differential among assets with claims on the same cash flow source further supports this unsophisticated investment trading strategy.

The estimating approach used as well as the pairs trading strategy are briefly introduced in Broussard, J. P., & Vaihekoski's (2012) article. The information and Finnish institutional traits.

1.15 Informed Trader Strategy

Although knowledge is a great commodity, having more in-depth knowledge than others costs money. In order to process information and make trading judgements, modern trading algorithms make use of unparalleled computer power. In this study, they demonstrate how an informed trader (IT) develops a trading strategy for companies in order maximise projected. IT has a market perspective that is encoded in a previous on price movements. Limit orders are a more passive and less expensive alternative to aggressive trading using market orders, but there is no certainty as to when they will be executed. The selection of order type is influenced by a number of variables, including as the amount of stock on hand, the amount of time left before a position most accurate forecasts approach to build up or reduce inventory gets more aggressive as the IT grows more certain of the pricing direction going forward.

Crucial concern is how to do this while taking into account the uncertainty in her projections, selecting the best combination of market and limit orders.

The majority work on

(i) Optimum liquidation/acquisition and

(ii) Market creating has been done.

Cartea,., Jaimungal, S., & Kinzebulatov (2016) add to the expanding body of literature in various ways and in a fresh way. By utilising stochastic control techniques, it mixes data and dynamic learning into an algorithmic trading issue. The trader employs directed tactics while trading all or a portion of a collection of assets. Additionally, they demonstrate how to trade utilising publications - exceptions - do. Our findings demonstrate how the IT adapts her technique to price fluctuations in order to maximise trading profits.

The volatility of pricing, however, is a significant factor that affects the effectiveness of learning and, thus, the profitability of the approach. For instance, they demonstrate through simulations that if the IT only trades in one asset and only gains knowledge from that asset's dynamics. Compromised in severe situations of excessive volatility, and she is unable to implement forecasts. Stacks up against those of competitors that use inferior information or are unable to take advantage of changing market conditions. The IT's approach outperforms the other traders' tactics second-order stochastically in terms of both (higher) mean and (lower) standard deviation of earnings.

Since inventories are discrete, are unable to simply utilise classical findings to demonstrate the convergence of the scheme; thus, build the demonstration from the ground up. In doing so, make advantage of recent findings that extend to systems PDEs that meet a certain monotonicity requirement and which may be applicable if inventories were continuous. The midprice model is shown in Cartea, Jaimungal, and Kinzebulatov's (2016) study, which also demonstrates how the IT sector may benefit from midprice innovations. The ideal trading strategy for the IT is derived. How does behaviour financial results compare to those of other traders who are less knowledgeable or unable to take advantage of market dynamics?

CHAPTER II:

REVIEW OF LITRATURE

2. LITERATURE REVIEW

2.1 Pairs trading

Pairs trading are a risk-free investment strategy approaches that also known as convergence trading (Kanamura, et al. 2009). Gerry Bamberger pioneered pair trading, since then scholars have offered several strategies for optimising pair trading. The Distance Method (Jacobs & Weber, 2015; Perlin, 2009; Broussard & Vaihekoski, 2012), Machine Learning (Huck, 2009; 2010), the Co-integration Approach (Vidyamurthy, 2004). (Sarmento & Horta 2020) investigated how the incorporation of ML may improve pairs trading. They suggested a novel method of searching for pairings based on the use of PCA followed by the OPTICS algorithm.

Due to the unpredictability of the stock market, that choosing a trading strategy (sell or buy) and being consistently accurate in their predictions can be tough for investors. Pair trading, an economy strategy that avoids following the overall industry's direction, may reduce investment risk and boost the chances for benefiting including in difficult environments (Evan et.al, 2006). Because of stock market volatility, many people who sell without the need for a well-thought-out plan may end up squandering time and money. The method, developed by Quantitative Analysts (quants) in the mid-1980s, entails selecting two strongly correlated shares and pairing a long (buy) investment in one with a short (sell) one in the other. Before going long (buy) on the undervalued and short (sell) on the overpriced, investors in pair trading look for a weakening in the correlation between the two shares. The slots will be closed once the link returns to its historical course, which is also known as statistical arbitrage.

Investors that engage in pair trading look for a under-priced, one calculated to be overpriced. Profits can be gained if a stock's price deviates somewhat from the estimated mean, according to the hypothesis, because the price will eventually revert to the mean ratio. The procedure of selecting pairings of equities to perform the strategy is not well recognized. As a result, there are multiple ways employed in the pairs-trading approach. The Cointegration Method, Distance Method, and Stochastic Spread Method are the three main ones (Vidyamurthy, 2004).

Evan et al. (2006) discovered that pairs trading in the US stock market might provide higher return of up to 11% yearly for individuality pairings between 1962 and 2002. The values of the component portfolios would be co-integrated if the short and long components fluctuated, and the pairs trading technique operate. However, practise, endowments entirely aligned (Vidyamurthy, 2004). According to the authors, the extra returns are related to an undiscovered systematic risk factor that has yet to be discovered. They support this claim by showing out that the returns on non–overlapping pairs' portfolios are highly correlated. Elliot et al. (2005) used a random variable with mean–reverting features to simulate probabilities of additional divergence. Despite the fact that the study is completely theoretical and does not provide any empirical evidence for the technique, Similarly, Lin et al. (2006) describe a derivative of Vidyamurthy's cointegration approach in their paper "Loss protection in pairs trading trough minimum profit bounds: A cointegration approach". According to the authors, this method will meet the requirements for a trade to generate a profit that was at least equal to the trading costs.

Gatev et al. proposed the classic DIM in pairs trading research, which has been thoroughly tested (2006). Their pairs trading technique provides for a fully submerged, and 1.44 % for a portfolio of the top twenty pairings. Their technique generates a 12.5 percent annualised return. Furthermore, they discover that lower thresholds are required to increase profits, even after accounting for trading expenses. They come to the conclusion that pairs trading is a risk-free approach with little impact from market risk. Perlin (2009) examines the success of the pairs trading strategy in developing nations. In comparison to weekly and monthly frequency, daily frequency yields the highest outcomes. However, because their final finding is based on in-sample data, the rate of return is incorrect. (Jacobs et al. 2015) employ to conduct worldwide exchanges, showing the approach consistently lucrative over time, although the returns are not consistent. Despite the fact that Vidyamurthy's technique substantially simplifies the computation of total profit, it is worthless. This study uses GA to address the problem.

2.2 The pairs trading and machine learning method

From 2003 to 2012, (Huang et al. 2015) used GA to optimise pairings created ten equities on the Taiwan stock market. Stock weights and trading thresholds are the optimization goals. Their strategy outperforms the Buy-and-Hold strategy. Their technique, however, necessitates a significant calculation suited limited pair trading. They only demonstrate out-of-equilibrium financial markets. To make trading judgments, they compare the simulation findings with future spread data.

The Sharpe Ratio of both approaches lowers over time, according to the researchers. They then suggest a novel trading signal that improves the performance of ETF-based signals by utilising trading volume. The Sharpe Ratio techniques using trade volume information hits 1.51 during the years 2003-2007, according to their findings.

2.3 MLPs in the stock forecasting field

Stock price prediction is highly difficult due to the financial market's long- and short-term volatility, as well as the complicated interplay of linear and nonlinear information. Without particular distributions, the new method can capture the nonlinear connection between price components. It can also intuitively select factors with significant influence using the factor significance score. In addition, the system's risk assessment, which is based on, is more thorough reliable than standard (Chen et al., 2020).

Price fluctuations are very volatile, and significantly influenced by erratic data from social medias, making it difficult to design effective and practical trading strategies. Existing NLP algorithms treat stock forecasting as largely a regression or classification problem, and they aren't intended to assist consumers in making sound investment decisions. They also don't take into consideration the temporal dynamics of massive amounts of widely influencing content, to which the market responds quickly. They proposed a DRL technique based on these faults that leverages textual data to create time-aware stock trading judgements while maximising profit (Sawhney et al., 2021).

Using the concept of stochastic estimation, (Fernandez-Tapia, 2015) developed a concept is to use a modification of the stochastic gradient-descent algorithm which may learn from its failures and improve its approach by taking use of the system's continuous nature while providing trading platform estimations. The benefit of this technique is that the algorithm explores the system in real time, eliminating the need for explicit price dynamics requirements, which is required in the stochastic-control approach. Furthermore, Avellaneda-Stoikov model with a discrete-time form for price/liquidity modelling, comparable to its evolution in the paper by Lehalle, Laruelle and Pages in the context of optimum liquidation strategies.

The SAMM (Strategic Analysis of Market Method) formalises the process of trading system discovery and development. It entails two processes that are rarely discussed: (1) a systematic way for identifying prospective trading edges without the need of data, and (2) a method for estimating a system's anticipated life duration using "strategic life cycles." Because of semi-predictable FED operations, market segmentation of the yield curve, and likely inefficient forward markets, futures expirations are likely "price distorters" (Moffitt, 2018).

Stock prediction is critical for optimising earnings from stock investments since it forecasts future stock price patterns. Despite the fact that much research has gone into using deep neural networks to enhance stock prediction, recent studies have shown two key flaws. In particular, and turned market price series into graphs while translating time series into complicated networks.

2.4 Financial Trading System based on Machine Learning

Machine learning techniques have lately witnessed an increase in their use in financial trading. Many quantitative trading systems have placed a premium on the ability to extract patterns from historical data price and apply them consistently in the future. Implementing financial trading methods based on machine learning, on the other hand, is difficult due to the necessity for precise hyperparameter fine-tuning, targets/rewards, and other factors. Furthermore, most existing systems are unable to utilise the data offered across a wide range of financial instruments. They presented a deep reinforcement learning-based technique in this research, which assures that the trading agent receives consistent rewards while avoiding the profit-and-loss payouts.

To do this, they also employed an unique price trailing-based incentive structuring strategy that significantly enhances the Sharpe ratio, agent's profit and maximum drawdown. Furthermore, they established a data preparation approach that allows the agent to be trained on a variety of FOREX currency pairings, allowing for the construction of market-wide RL agents as well as the use of increasingly effective repeating deep learning techniques without the risk of overfitting. Speedlab AG has offered a challenging large-scale data set, which encompasses 28 instruments, is used to demonstrate the potential of the suggested solutions to improve multiple performance indicators (Tsantekidis et al., 2021).

Machine learning has garnered considerable attention from academics in recent years as a way to better evaluate and predict financial data. Machine learning offers a distinct edge in dealing with financial data as compared to the traditional methodology. First and foremost, it can automatically detect hidden characteristics in financial data, decreasing the need for human interaction. Second, the fundamental linear stock valuation model is predicated on the premise of a linear financial system. The stock market is a dynamic system comprising both nonlinear and linear material, according to research into nonlinear properties of financial time series such as prolonged memory and non-pairing distribution.

Machine learning models are not confined to the probability distribution of investment income and can cope with high-dimensional and collinear elements. Calculating a high-dimensional covariance matrix is not required for machine learning models (Chen et al. 2020).

Our suggested framework's efficacy is demonstrated using real-world stock data, and our technique outperforms many state-of-the-art benchmarks. Furthermore, our framework achieves the largest cumulative profits in the trading simulations. These findings complement the present usage techniques domain have significant financial market making (Junran Wu et al., 2022).

The way financial markets function has been completely transformed by algorithmic trading robots, which have increased trading activity's efficiency and accuracy to never-before-seen levels. This essay functions as a thorough manual, exploring the nuances of Algorithmic Trading Robots and illuminating the processes that contribute to their effectiveness (Nasir khan, 2023).

2.5 Algorithmic Trading

Algorithmic trading is a prolonged perception and decision-making problem in which the algorithm must simultaneously explore the environment while making right automated data, and the algorithm must learn feature representation from highly nonstationary, picking signal elements, in particular, total data new trends based. Furthermore, taught together frequently to maximise investment returns by utilising indicators in the learning algorithm that have been guided, obtaining more current data, and tightening limitations on the gradient descent. A set of analyses demonstrate the suggested method's strong superiority and broad application (Lei et al., 2020).

In algorithmic trading and financial management, machine learning models are becoming more common. The rise of machine learning in finance has posed a challenge to traditional modelling and model-use procedures, necessitating the creation of practical solutions to deal with the complexities of these approaches. The study analyses how quants manage model complexity in the process of constructing machine learning-powered trading algorithms, based on interviews with quants who utilise machine learning approaches to solve financial problems. Recognizing the human's changing role in today's data and model-driven finance, according to (Hansen, 2020), necessitates an understanding of how quants deal with the complexity of learning models.

The study of finance research on the human–model interaction by exploring it in the context of machine learning model use. A novel algorithmic investment management strategy that results in lucrative automated trading strategies. As a result, a large number of scaling laws have been found. By integrating the trade model design with an agent-based approach inspired by the study of complex systems, a new guiding principle was produced. This revolutionary approach to developing automated trading procedures is a low-cost way to create a new form of investment strategy that earns profits while simultaneously providing liquidity to financial markets, with no a priori asset limits (Golub et al., 2018).

To develop statistical arbitrage methods in electronic marketplaces, authors used reinforcement learning (RL) approaches. The optimum tactics for an agent trading in a foreign exchange (FX) triplet are determined using double deep Q network learning (DDQN) and a new type of reinforced deep Markov models (RDMMs). An FX triplet is made up of three currency pairings where the exchange rate of one pair is redundant since it is decided by the exchange rates of the other two pairs due to no-arbitrage. To apply the methods and demonstrate their financial effectiveness, the simulations of a co-integrated model of exchange rates were employed. Furthermore demonstrated the financial success of the trading strategy based on the RDMM technique using simulations (Cartea et al. 2021).

The current state of DLR in the subdomain of AI in finance, specifically automated low-frequency quantitative stock trading. DRL in stock trading has shown significant application potential in comparison to professional traders under strong assumptions (Pricope, 2021).

An algorithmic trading system that combines a pairs trading approach with environmental, social, and governance (ESG) evaluations is proposed. It responds to the market for socially conscious investing options by creating a special algorithm that combines co-integrated stock identification techniques with ESG data. This makes it possible to choose lucrative pairings that follow ESG guidelines. Technical indicators are also included for the best possible trade execution within this sustainable framework.

The model's efficacy is demonstrated by extensive backtesting, which regularly yields positive returns that outperform traditional pairs trading techniques while preserving ESG principles. This opens the door for a revolutionary method of approaching algorithmic trading, providing information to scholars, policymakers, and investors alike (Eeshaan Dutta, 2024).

In order to address these issues, Moli Qin (2023) presents an innovative three-stage hierarchical reinforcement learning framework for high frequency trading called EarnnHFT, which is an efficient hierarchical reinforcement learning technique. Firstly, we calculate a Q-teacher, which is the best action value determined by dynamic programming, to improve the performance and training effectiveness of second-level reinforcement learning agents. In stage II, hundreds of RL agents are trained with varying return rate preferences, and only a small percentage of them will be chosen for the pool depending on profitability. This creates a diversified pool of RL agents for various market trends, differentiated by return rates.

In the third step, we train a minute-level router that selects a second-level agent from the pool dynamically in order to get consistent performance in various markets. We show through comprehensive trials in a high-fidelity simulation trading environment over a range of market trends on Crypto exchanges that EarnHFT significantly surpasses six state-of-the-art baselines in six key financial metrics, outperforming the runner-up by 30% in profitability.

(Gang Hu, 2023) explores the improvement of the conventional Deep Q-Network (DQN) trader model by including state-of-the-art methods like Regularised Q-Learning, Duelling DQN, Double DQN, Prioritised Experience Replay, and Noisy Networks. Based on extensive empirical testing on several financial instruments such as AAPL and BTC/USD, the study clearly shows improvements in performance compared to the original model. The DQN-vanilla variation of the improved DQN trader shows impressive increases, increasing its arithmetic return from 261% to 287% and improving its Sharpe Ratio, which indicates superior risk-adjusted returns.

The clever application of CNN1D and CNN2D architectures highlights the effectiveness of convolutional layers in capturing market movements and further magnifies profits. When the improved model is applied to the AAPL stock, significant gains are seen, demonstrating its consistency. The CNN-based models exhibit remarkable returns that showcase their architectural strength, whereas the DQN-pattern variant maintains a consistent performance. These results show the possibility of using convolutional neural networks in financial trading systems, surpassing even the performance of the baseline model. The results of the study verify that automated trading strategies may perform much better when these advanced deep learning techniques are applied inside a reinforcement learning framework. The significance of ongoing innovation in this field is supported by the performance consistency across a range of financial instruments.

The study's recommendation for future research emphasises the possibility of novel reinforcement learning techniques to increase the model's efficacy in a greater range of financial contexts. The study's main contributions and consequences are summarised in the abstract, which also lays the groundwork for further breakthroughs in the field of AI-driven financial trading.

“The Rise of Algorithmic Trading" examines how algorithms are becoming more and more important in the financial markets and how this is affecting trading strategies. Trading using computer programmes to carry out deals automatically and without human involvement is known as algorithmic trading. The history of algorithmic trading, its benefits and drawbacks in relation to conventional trading techniques, and the many kinds of algorithms employed in trading are all covered in this study. Along with the effects of algorithmic trading on market efficiency, liquidity, and volatility, the function of AI and machine learning in algorithmic trading is also examined. The study also addresses the dangers and difficulties of algorithmic trading, such as unforeseen losses and technical malfunctions (I.V. Dwaraka Srihith, 2023).

2.6 Machine Learning and Quantitative Trading

The ML technique has been widely utilised in quantitative investing, although there are several issues with factor testing, factor generation, and the loss function's purpose. The QFII, the institutional and the shareholder concentration index are all part of the CSI 300 index's style index. The classic factor testing technique is used to assess the linear or sequence validity. Machine learning is used in quantitative trading to discretize the return on capital and minimise the objective functions. However, it disregards a significant amount of information on the return on capital. Machine learning's primary goal is to maximise the cumulative return on capital. Finally, the problem is solved using the heuristic technique Particle Swaps Optimization (PSO), which yields the relevant parameters (Dai, 2020).

Machine learning techniques are now being used to research and build quantitative trading systems. Although a thorough examination of the forex, stock and futures exchange markets has previously been conducted in this context, machine learning approaches have received less attention in the growing cryptocurrency exchange market. Furthermore, compared to baseline techniques only focused on Bitcoin trading, trading numerous cryptocurrencies at the same time greatly enhances total profits (Attanasio et al., 2019).

Trading price data is used to create technical indicators in technical analysis, which is widely utilised across the world. These indications aid investors to purchase and sell for maximum profit. A quantitative trading machine learning (QTML) was suggested in this study to recommend when a stock should be bought or sold. For all purchasing and selling transactions, the trial uses a 0.2 percent commission charge. The results suggest that the proposed QTML model can help investors make more money, especially in companies with fluctuating or declining tendencies (Vinitnantharat et al., 2019).

Quantitative trading has become increasingly popular as computer science and artificial intelligence have advanced, owing to its efficiency and consistency. The possibilities of deep reinforcement learning in quantitative trading are investigated. The agent receives raw financial data as input and produces trading decisions as output. The agent's aim is to increase the final profit. In addition, authors also employed many technical indicators as an additional input to increase the agent's performance while limiting the effect of market noise. On the stock market, the suggested system has been back-tested.

The findings show that our strategy works effectively in the majority of situations (Jia Wu et al., 2019). Quantitative trading is an important aspect of financial markets that requires fast calculations, although no quantum algorithms have yet been developed in this discipline. Also created are two tool methods for cointegration test and condition number estimation (Zhuang et al., 2021).

Quantitative trading is a type of automated trading system series manages the trading strategies. It uses tools in finance, physics, forecast, profit from enormous data. As a result, a large number of scaling laws have been found. By integrating the trade model design with an agent-based approach inspired by the study of complex systems, a new guiding principle was produced. This revolutionary approach to developing automated trading procedures is a low-cost way to create a new form of investment strategy that earns profits while simultaneously providing liquidity to financial markets, with no a priori asset limits (Ta et al., 2018).

Quantitative trading seeks advantageous returns by using statistical or mathematical methods to identify trends in historical data. In the testing stage, most of the associated frameworks show low generalizability. As a result, researchers used adversarial learning and an unique sampling technique to manage (Reinforcement Learning) RL portfolios. The intention was to develop a portfolio of five assets from Dow Jones Industrial Average members and use our trading approach to produce excellent results.

During the RL phase, adversarial learning to boost the model's resistance was used. Authors also explored, which increased model's processing efficiency. The results of the studies showed that the model created with our sampling method outperformed the random learning methodology. The Sharpe ratio increased by 6%–7%, while profit increased by roughly 45 percent. As a result, the sampling approach used and the learning framework and suggested are both favourable to developing trustworthy trading rules (Huang et al., 2021).

(Wang et al., 2021) used available stock data to examine an intraday trading strategy under T+1 utilising Multilayer Perceptron (MLP) and Markowitz optimization. The empirical findings show that Markowitz portfolio optimization is profitable and that MLP may be used to predict intraday stock prices. To further demonstrate the possibility of this method, the findings integrate the MLP and the Markowitz optimization, with the trading technique.

(Da Costa & Gebbie, 2020) investigated the feasibility of a composable mechanistic online ML approach for learning signs from lower frequencies time series data. The approach has been validated using daily selected ending data information from JSE equities markets. The data processing is split into two steps: unsupervised learning with a stacked autoencoder, followed by batches and reinforcement learning with these learnt features, with the outcome being a point prediction of observed time-series feature fluctuations. Weight initializations are implemented using variance-based initializations and constrained Boltzmann machine pre-training. The process is used to initialise an online feedforward neural network, which is subsequently used to execute historical simulations.

Quantitative investment success is usually built on accurate predictions of future stock prices. ML-based techniques have recently proved their ability to provide higher appropriate stock forecasts and have become essential parts in contemporary quantitative trading systems. The Temporal Routing Adaptor (TRA), a new architecture that allows current stock prediction models to mimic different stock trading patterns. Despite this, the lack of specific pattern identifiers makes training an efficient TRA-based model rather difficult (Lin et al., 2021).

Despite their effectiveness, present approaches have several limitations, including as poor performance when dealing with vast amounts of data that are intrinsically complex, have high dimensionality, and are dynamic in nature. Furthermore, these techniques are ineffective in uncovering hidden data relationships (dependencies). This study provides a review of some of the most important studies in the subject of quantitative finance, offering thorough ML techniques. Finally, the study includes comparison evaluations of the efficacy of several ML-based systems (Rundo et al., 2019).

(De Spiegeleer et al., 2018) demonstrated how machine learning approaches may be applied to conventional quantitative issues. ML approaches based on Gaussian process regression, we can achieve speedups of many orders of magnitude for many classical tasks. We must pay a price for this increased speed: a loss of precision. However, we show that this lowered accuracy is frequently well within normal bounds, making it extremely acceptable in practise. Fitting and estimate are two concrete examples. In the fitting context, designers fit advanced Greek models and sum up market volatility surfaces. In the estimate context, we minimize calculation rates for American option pricing, and exotic option pricing using sophisticated models other than the Black-Scholes setup.

Determining the dynamic trading strategy that optimises projected utility of ultimate wealth in multi-period trading with actual market influence is a difficult task. We show in this study that reinforcement learning techniques (particularly, Q-learning) may successfully handle the risk-averse scenario with the right reward function. We present a proof of concept in the form of a simulated market that allows for statistical arbitrage even when trading fees are included. This arbitrage is discovered and exploited by the Q-learning agent (Ritter, 2017).

This is the first in a series of essays about asset management and machine learning. Asset management is made up of five functions: risk management, portfolio construction, infrastructure, capital management and deployment, and sales and marketing. This article discusses how to create a portfolio using ML. The phrase "algorithmic trading" used to refer to the automation of sales trading platforms, but as more sophisticated programs have been developed, the word has grown to include alpha factor design, idea generation, asset allocation, strategy testing and position size.

ML may assist in all of these areas when used as a decision-making tool (Snow, 2020). Furthermore, the classifiers and sorting algorithms may be used in such a way that the optimum performance can be achieved with a small number of relevant features. Therefore the study paves the way for more advanced features to be developed, as well as more sophisticated feature selection algorithms (Ntakaris et al., 2020).

Despite a growing body of literature, multivariate time series portfolio management is rarely explored. On the other side, most studies concentrate on improving strategy rules rather than determining the best portfolio weight. To exploit the concerns, authors proposed a simple yet successful technique dubbed Volatility & Model Adaption Trade-off (VMAT) in this study. Experiment tests reveal that it outperforms the competition in terms of profit (Yu & Xie, 2021).

Pairs trading is a quantitative trading method that takes advantage of out-of-balance financial markets. The stochastic (residual) spread approach, in particular, is based on more classic estimating techniques such as the Expectation Maximization algorithm. The findings reveal that the recurrent neural network outperforms the other approaches, since it delivers the greatest returns (about 11%) and the highest Sharpe and Sortino ratios (Kole, 2017).

2.7 Reinforcement Learning

Reinforcement Learning (RL) advances include a huge range of applications, motivating more research in the field. This enables a rethinking of the issue arrangement in light of real constraints. By presenting trading as a real-world hierarchical RL issue, TradeR use hierarchical RL to execute trade bids on high frequency actual market events such as sudden price changes during the COVID19 stock market meltdown in the 2019 fiscal year. TradeR displays resistance to rapid price swings and catastrophic losses while retaining profitable results (Suri et al., 2021). Trading price data is used to create technical indicators in technical analysis, which is widely utilised across the world. These indications aid investors in determining the best time to purchase and sell for maximum profit.

In investment firms, stock trading technique is quite important. In the complicated and dynamic stock market, however, determining the best approach is difficult. We look at how deep reinforcement learning may be used to improve stock trading strategies and so increase investment returns. Our trading stocks are 30 stocks, and their daily prices serve as the training and trading market environment. We create an adaptive trading method by training a deep agent. Performance assessed and compared that of the Dow Jones Industrial Average technique and the typical min-variance portfolio allocation and. In terms of both the cumulative returns and the Sharpe ratio, the suggested deep reinforcement learning strategy outperforms the two baselines (Xiong et al., 2018).

Machine learning breakthroughs are finding commercial uses in a variety of areas, including banking. This study focuses on applications in the investing process, which is one of the main activities of finance. Return forecasting, risk modelling, and portfolio creation are all part of this. Through an exhaustive analysis of recent literature, the research assesses the present state of the art. The major use cases are emphasised, and themes and technologies are discovered and categorised (Emerson et al., 2019).

RRL has been discovered to be a successful machine learning approach for developing financial trading systems. A genetic algorithm (GA) is used in this research to enhance performance an equities trading system. (Zhang & Maringer, 2016) assessed the stability and profitability of the proposed GA-RRL trading system. They discovered that when the number of businesses with a considerably favourable Sharpe ratio grows, after feeding the GA's indicators into the RRL trading system.

Reinforced learning approaches have been used to a variety of fields, including algorithmic trading. The stock market is understood as a game with a Markov property consisting of states, actions, and rewards in this study. The asynchronous advantage actor-critic technique is used in conjunction with multiple neural network architectures to propose and evaluate a system for trading the fixed volume of a financial asset. In this method, the use of recurrent layers is examined. The trials were carried out on real data that had been anonymised. The top architecture showed a trading technique for RTS Index futures (MOEX:RTSI) that generated a profit of 66 percent per year after commissions (Ponomarev et al., 2019).

On the one hand, financial time series are multifractal, which means they have non-Gaussian distributions, outliers, and long-range dependent dynamics. Machine learning (ML) models, on the other hand, are processes that largely rely on statistical models and techniques, but are classified as black box models because they lack explicit knowledge of the relationships created between explanatory factors (input) and dependent variables (output). However, knowing the statistical features of the time series produced by the ML model under discussion is critical for anticipating market returns or producing autonomous patterns. The findings of utilising recurrent neural networks to anticipate market returns are first presented. After that, discussed about forecasting stock price directions and present an algorithm for projecting both future price levels and asset price directions at the same time. They also investigated into autonomous pattern formation and the importance of mastering classic stock analyst forms. Then, to anticipate stock price fluctuation patterns, and used complicated network analysis.

Finally, authors explored the statistical properties of financial time series and offer a formula for projecting market returns as well as their orientations. Also suggested reversing the causality and deciding on the framework by describing how the model should be described before starting to analyse the data. We should establish the framework by carefully identifying the number of dependent variables in a regression model, or the number of elements and components in a stochastic model, and then appropriately defining model hypotheses that make financial or economic sense. Hence employed the multifractal formalism (MF) as a framework for examining the ability of an ML model to recreate specific non-overlapping statistical features of the time series based on these data.

First, we establish a collection of theoretical models with different statistical features, as well as a set of machine learning models capable of reproducing the qualities of the former set of models. We next utilise ensemble techniques to merge these now calibrated ML models to construct a meta-model by training each individual ML model to mimic their particular theoretical model. The latter's weights, on the other hand, are calculated statistically as a function of the Hurst exponent. As a result, based on the changing features of the financial series over time, the meta-model is dynamically recombined. We may train each constituent model to learn a wide number of patterns or technical indicators once we've selected an ensemble of ML models with specified non-overlapping statistical features. After that, we create a trading system with methods that account for a certain amount of Hurst exponent (Bloch, 2018).

2.8 Deep Reinforcement Learning

A key advancement is the coupling of improvements in deep learning for learning illustrations with Reinforcement Learning (RL) (Hinton et al., 2012; Krizhevsky et al., 2012). DRL's most amazing use is the ability to perform a series of tasks in various contexts. DRL's success in the games area suggests that it will have a potential use in trades. DRL has recently been used to finance in a tiny quantity of literature. The learning system's findings on both stock-index and commodities show that it is capable of optimum action learning.

The performance of an optimization model can be improved by using return prediction from classical time series models in portfolio creation. They used two machine learning models, support vector regression (SVR) and random forest (RF), as well as predict return in portfolio formation because outperform time series models. To be more explicit, before forming a portfolio, this study uses these prediction models to preselect stocks (Ma et al., 2021).

Multiple portfolio optimization strategies are used to enhance portfolio performance, including simulation modelling Monte Carlo simulation (MCS), equal-weighted modelling (EQ) and optimization modelling mean variant optimization (MVO). The findings demonstrated that our suggested LSTM prediction model is effective in predicting stock prices with high accuracy. Furthermore, optimization approaches significantly improved the generated portfolios' return and Sharpe ratio (Ta et al., 2020).

(Dab et al., 2019) The use of deep reinforcement learning (DRL) to improve execution is referred to as "deep execution." Also illustrated how to use two separate strategies to solve for optimum trade execution and exceed market benchmarks like the time-weighted average price: (1) the deep double Q-network (DDQN), a value-based method, and (2) the proxima policy optimum (PPO), a policy-based method (TWAP).

First, showed that by acting directly on the environment, the DRL may reach the theoretically defined optimum. Second, DRL agents can learn to gain from market prices without ever seeing the price (alpha signals). Finally, being an intelligent trader, the DRL may develop dynamic strategies that outperform static benchmark techniques like the TWAP using the existing data.

Investors and scholars have been increasingly interested in quantitative investment strategies mixed with artificial intelligence in recent years. Existing supervised learning approaches are not well suited to learning problems with long-term goals and delayed rewards in real-world futures trading. Instead of price, volume, and numerous technical criteria, used more than 100 short-term alpha components in the proposed technique to define the states of MDP.

Moreover, unlike DQN (deep Q-learning) and BC (behaviour cloning) in related approaches, they included expert experience in the training stage and construct the temporal difference error by considering both the expert and the agent-environment interaction to enable the representatives more adaptable to unavoidable disturbance in financial data. Experiments with Chinese share price index futures, such as IF (CSI 300) and IC (CSI 500), show that the proposed strategy outperforms three classic technical analysis methods and two deep leaning-based methods (Chen et al., 2021).

We propose that employing a decision ladder (DL) to represent phases and degrees of automation could help in determining information required for automation interface creation. We examine, which are field different levels. This study adds to our understanding of automation and automated trading in general, as well as a more complete description of human-automation interactions and, as a result, a deeper understanding of design requirements (Li et al., 2016).

Portfolios, securities, stock market forecasts, risk management, and debt management are all key financial foundations. As can be seen, these strategies are classified based on their use of Neural Networks, Support Vector Machines, and other quantitative financial factors. There are further classifications based on supervised and unsupervised algorithms, as well as K-Mean clustering. Furthermore, this research digs into the hitherto uncontrollable and unmovable occurrences in market and public psychology, with the goal of proposing a realistic remedy (Vats & Samdani, 2019).

Latent factors are frequently used to provide alpha signals for statistical arbitrage techniques. Furthermore, we take into account the impact of the trader's activities on stated prices as well as the prices they get as a result of trading. By developing a variant of the expectation-maximization algorithm, to demonstrate the usefulness of the optimal method and compare it to techniques that neglect latent factor learning (Casgrain & Jaimungal, 2016).

Using the Proximal Policy Optimization technique. To maximise signal-to-noise ratio in the training data, we choose those training examples with the biggest price fluctuations. The test is then run on the data from the next month. The Sequential Model Based Optimization approach is used to optimise hyperparameters (Briola et al. 2021).

Despite the low signal-to-noise ratio supplied by financial markets, traditional portfolio optimization frequently necessitates anticipating asset returns and their related variations. DRL provides a framework for optimising sequential trader choices via an objective that reflects its reward function as a function of risk and transaction costs. Authors explored the performance of model-free DRL traders in a market context with frictions and various mean-reverting variables controlling the return dynamics.

Because this framework allows for an accurate dynamic programming solution, we can evaluate the limits and capabilities of various value-based algorithms to recover relevant trading signals in a data-driven way and achieve benchmark performance. Furthermore, thorough simulations indicate that our technique ensures flexibility, beating the benchmark when the price dynamics are misspecified and some basic assumptions about the market environment are broken due to the presence of severe events and volatility clustering (Brini et al. 2021).

To achieve long-term returns in algorithmic trading, two major obstacles are feature extraction and trading strategy creation. The previously described approaches, on the other hand, rely largely on domain expertise to extract handmade characteristics and lack an effective mechanism to dynamically change the trading strategy. With recent advances in DRL, sequential real-world issues may now be represented and addressed in a more human-like manner. To better react to the trading market, we enhanced the value-based deep Q-network (DQN) and the asynchronous advantage actor-critic (A3C).

Our trading agent beats the baselines and delivers steady risk-adjusted returns in both the stock and futures markets, according to the experimental data (Li et al. 2019). Developing reliable financial analysis tools may be important for both speculative trading and evaluating market activity and responding quickly to unstable situations to ensure the smooth operation of financial markets. The suggested strategy leads to the development of robust agents that can survive high levels of noise while still collecting pricing patterns and making lucrative judgments (Zarkias et al. 2019).

2.8.1 Financial Trading as a Game: A Deep Reinforcement Learning Approach

Every market practitioner would benefit from an automated method that creates consistent profit from the financial market. Recent advances in deep reinforcement learning give a framework for training such trading agents from start to finish. In this study, we present a Markov Decision Process (MDP) model appropriate for financial trading tasks and solve it using the cutting-edge deep recurrent Q-network (DRQN) technique. Huang suggested many changes to the existing learning algorithm to make it more suited for application in financial trading, that utilised significantly a smaller replay memory (just a few hundreds of bytes in size) compared to those employed in recent deep reinforcement learning algorithms (often millions in size.) and also he created an action augmentation strategy to reduce the requirement for random exploration by delivering additional feedback signals to the agent for all activities.

This enables us to utilise greedy strategy during learning and demonstrates great empirical performance when compared to more typically used -greedy exploration. However, under a certain market assumptions, this strategy is specialised to financial trading. For recurrent neural network training, we sample a longer sequence. As a result of this process, we can now train the agent for each T step. Because the whole calculation is reduced by a factor of T, training time is considerably reduced. We put all of the foregoing into a comprehensive online learning algorithm and test it on the spot foreign exchange market (Huang, 2018).

2.8.2 Idealized trading games in DLR

Deep Q-learning is studied as an end-to-end solution for determining the best methods for acting on time series input. Experiments are carried out using two idealised trade games such as the Univariate game assesses the agent's ability to capture the underlying dynamics, whereas the Bivariate game assesses the agent's ability to exploit the hidden relationship between the inputs. Q values are modelled using stacked gated recurrent units (GRU), convolutional neural networks (CNN), long short-term memory (LSTM) units and multi-layer perceptrons (MLP). All agents successfully identify a profitable strategy in both games. In the Univariate game, GRU-based agents outperform others, but MLP-based agents outperform others in the Bivariate game (Gao, 2018).

2.9 Data mining and Risk management

Choice value is a critical factor in risk management and trading. Many sources were displayed in 2022 (Ge et al.). Three channels are used to display the data that was analysed. Data-handling methods were proposed. The proposed techniques surpass commonly utilised methods such as modelling, according to the chosen database. (Charupat & Miu, 2016) (and trading styles). Bond funds, future prospects, ETFs, indexes, commodities, currency fluctuations, convertibles, organised investments, property investment, miscellaneous, infrastructural facilities were also included. Include with explanatory comments for showing out-of-sample back testing, as well as over 900 dictionary, acronym, and math definitions.

Data mining the practise of identifying recurrent patterns in large data sets in order to extract value from them. In empirical finance, data mining has a bad meaning. Diego (2019) believe that data mining is an unavoidable aspect of financial research. We are all data miners, even if we simply live through a unique past that influences our ideas. Data collection was once costly, and processing resources were scarce. As a result, scientists were forced to focus their efforts on the most feasible scenarios. Data and computational resources are increasingly cheap, and in the age of machine learning, researchers don't even need to submit a hypothesis because the algorithm will most likely find it out.

Researchers today are fortunate to have, one of machine learning, which spans a wide range of approaches. Machine learning has advanced our knowledge in the physical and biological sciences, including being effectively used to consumer behaviour analysis. In all of these applications, a considerable amount of data is utilised. When working with massive amounts of data, patterns will emerge totally by chance.

Machine learning is intended to prevent overfitting by regularly cross validating new patterns, which is one of the most natural benefits. This advantage is helpful once again in the presence of a large volume of data. The range of data in investment finance is substantially more limited, with the exception of tick data. In fact, most capital strategies that promise to outperform a passively benchmark use on monthly and quarterly data. In this case, cross-validation does not alleviate the dimensionality curse. Machine learning and other statistical technologies, which were previously difficult to apply, have a lot of promise for building successful trading strategies, especially in higher-frequency trading.

They could also be very useful in other fields, such as risk management. When employing this equipment, however, we must be cautious. Given the limited nature of the standard data we use in finance, we believe that many of the challenges we face in the era of machine learning are comparable to those we've faced in quantitative finance in general. In back tests, we want to avoid overfitting investment techniques, but we want a stable environment in which to discover new (real) ways. We think that now is an excellent opportunity to take a step back and rethink how we conduct research.

Many people have previously warned about the dangers of data mining, but the issue is much worse now. The level of competition has levelled off in terms of processing capability, data, and statistical ability. As a result, novel ideas run the risk of being overcrowded. Indeed, simply revealing a flaw could set in motion the process of manipulating the market. A methodology for doing empirical financial research is proposed in our work. Research protocols are popular in various fields, and they are designed to avoid obvious errors that could lead to misleading findings. Their approach covers both traditional statistical methods and newer machine learning technologies. (Diego, 2019).

Given that the issue is to make a choice based on huge amounts of data, there are several obvious options. This study begins by reconstructing corresponding, which illustrate different approaches to the problem and are based on distinct components of the several theories that have developed. In terms of tool and equipment, he uses trading simulations. In three scenarios, the baselines are long. Findings show that extreme situations can be avoided. Gives outcomes of three when reference continues to lose. (Zhang, 2021).

2.10 Machine learning anticipating stock price changes

Since the recognised strategy, the experience has been appealing. Developing and debugging a trading agent that selects in, on the other hand, takes time. FinRL, a DRL library that allows novices to learn about design techniques, was released by (X.-Y. Liu et al., 2021). FinRL library provides users with simple tutorials as well as the ability to build neural networks trading agents and undertake extensive back testing to analyse trading performance. It also considers important trading restrictions. FinRL stands out for its comprehensiveness, hands-on instruction, and reproducibility, all of which are beneficial to newcomers: We also provided three application demonstrations.

The capital sector is a dynamic and complicated structure that is affected by a variety of unpredictable elements. As a result, it's impossible to predict stock price swings. Machine learning aims to understand and identify patterns in massive data sets automatically. They could be beneficial for anticipating stock price changes and building automated trading strategies based on these predictions.

Despite the fact that artificial intelligence has been used to predict market movements, published methodologies rarely. (Lu, 2016) examined how different, approaches, can be used to forecast the performance of S&P 500 companies. After that, the best-performing models are employed to develop automated trading strategies.

It's difficult to make continuous earnings in the stock market. As a result of breakthroughs in computer science and reinforcement learning, several artificial intelligence trading algorithms had surpassed Buy&Hold (B&H). On the other hand, the information and procedure are insufficient, limiting the effectiveness of reinforcement learning algorithms. Experiments on five equities that delivers than basic 2021 (Wang et al.). What effect does it have when networks have a large amount of stock? This topic has been empirically researched (Chen et al., 2019). Assess the Chinese people who have been questioned.

When loss, efficiency, and accuracy are evaluated, the findings indicate a forecast. a percent, training features were used to make this. According to the back test, this approach can get a SHARP ratio of 2.204. Artificial intelligence (AI) techniques for financial research and applications have received example, artificial intelligence (AI) technologies scenario identification strategy implementation.

2.11 QT strategies

QT, on the other hand, have modelling frequency utilising AI techniques to balance concerns, introduce iRDPG, that enables an intelligent trading agent to autonomously construct QT strategies. Our model benefits from deep reinforcement learning (DRL) and imitation learning methodologies. In order to account for inputs, construct. In addition, (Y. Liu et al., 2020) proposed imitation learning as a means to balance exploration and exploitation using classic trading approaches. Experiments show strong adapt to a variety of markets.

As a result of the increasing nature of, rigid dealing approaches have been formed, which are huge issues for the banking world. (Wu et al., 2020) offer flexible share trading algorithms deep residual reinforcement learning approaches to overcome this issue. Because stock market data is time-series, the Gated Recurrent Unit (GRU) is utilised to recover relevant economic features that properly reflect the equity market's essential properties for adaptive trading choices. GDQN and GDPG are verified in both trending and turbulent stock markets from various countries to ensure that they are both robust and effective.

AI-driven models have demonstrated remarkable proficiency in anticipating market patterns and executing trades at a speed and precision that far exceed human skills. This is especially true of those models that use machine learning techniques like deep learning and reinforcement learning. Its ability to automate vital functions, such analysing market conditions and carrying out trading plans, has proven crucial. Current QT techniques, however, continue to face significant difficulties, particularly when it comes to efficiently managing noisy and high frequency financial data.

For AI-driven trading agents, finding a balance between exploration and exploitation is another difficulty. Our suggested remedy, QT Net, overcomes these obstacles by introducing an adaptive trading model that uses an intelligent trading agent to create QT strategies on its own. By combining imitative learning techniques with deep reinforcement learning (DRL). We conceptualise the QT mechanism within the context of a partially observable Markov decision process (POMDP) in order to address the difficulties presented by volatile financial datasets. Additionally, the model may leverage conventional trading strategies by integrating imitative learning, fostering a harmonious synergy between discovery and utilisation.

Our trading agent is trained with minute-frequency data from the real-time financial market to provide a more realistic simulation. The results of the experiments highlight the model's capacity to extract strong market characteristics and its flexibility in dealing with a range of market circumstances (Maochun Xu, 2023).

2.12 Trading On-Line

Zhang et al. (2020) used to compare their DLR algorithms to traditional time-series momentum techniques and demonstrate that their method outperforms conventional baseline models, generating positive returns despite high transaction costs. Experiments indicate that the suggested algorithms can follow significant market moves without changing positions, as well as scale down or hold during consolidation periods.

In a volatile stock market, the trading methods they produce similar results like DRL trading strategy, a cutting-edge direct reinforcement learning approach. Stock market prediction and trading has captivated the curiosity of innumerable academics from various scientific subjects due to the market's immense complexity. Among investors, the "Trading System (TS)," a systematic technique for predicting stock prices and trends, is gaining popularity. By drastically increasing the volume of stock market index transactions, Trading On-Line (TOL) has increased market complexity and liquidity. One of the TOL's most important outputs is "automated trading," which is an ad-hoc algorithmic robotic process of automatically examining a large amount of financial data with the goal of launching several trading actions in order to maximise the trading system's profitability.

The trading strategy is known as High Frequency Trading when the number of these automated activities climbs to a significant level (HFT). In this regard, machine learning has lately improved the robustness of trading systems, notably in the HFT industry. The authors (Rundo et al., 2019) provide an unique ad-hoc machine learning-based approach for accurately forecasting stock prices using historical data.

For decades, financial trading has been extensively studied, with market players and researchers always seeking better approaches to increase trading success. This trading technique is evaluated on both actual and simulated price series, and the outcomes are compared to an index benchmark. Traded using trained reinforcement learning agent (Hirsa et al. 2021).

2.13 Algorithmic Quotation

Adaptive correction, which is based on the idea that stock price growth is driven by Markov stochastic propriety, improves the accuracy of stock price estimates. The validation findings for such shares and financial instruments confirm the robustness and efficacy of the suggested automatic trading approach (Rundo et al., 2019). (Hu and colleagues, 2020) investigate the quality of corn, soybeans, and live cattle using quotation. We find that more intensive algorithmic quotation (AQ) is favourable to a number of market quality indicators using data from the CME's limit-order-book and a heteroscedasticity-based identification technique.

Although enhances price reduces impact. In the maize and soybean markets, increased AQ narrows effective spreads greatly, liquid. Moral hazard costs fall when, resulting in a shrinking of effective spreads, there additional evidence that increased AQ activity enhances liquidity provider profitability, but represents compromise qualities necessity algorithmic trading activity to be continuously evaluated and monitored (Hu et al., 2020).

Increasing the price lowers the impact. Increased AQ significantly narrows effective spreads in the maize and soybean markets, making them liquid. When there is additional evidence that increased AQ activity enhances liquidity provider profitability, moral hazard costs fall, resulting in a shrinking of effective spreads. However, enhanced AQ action produces a middle ground quality that requires algorithmic speculative trading to also be constantly assessed and supervised.

The project proposes a LSTM in ANN to anticipate the values of a portfolio of top stocks from five different industries since the election. To anticipate the following day's price, the current price, technical and fundamental indications will be learned. Based on specified rules, the LSTM model will modulate and internally optimise the pricing. The network generates trading signals by comparing real and expected prices, which are then utilised to back-test the data on a trusted platform. Following that, the back-testing findings are analysed to determine the performance and risk measures for two sorts of investors: risk-averse and risk-seeking (Joshi, 2018).

2.14 Market-trading

A mathematical approach for forecasting stock prices that uses daily mean reversion and swing trading to trade various market patterns. The framework is built around three main principles, which are listed below: Short-term mean reversion is used to utilise various characteristics of business sectors in this market-changing method. Market - trading on a regular basis exposes market behaviour in a range of situations. Each framework part is built on the concept of unpredictability and flexible measurements, allowing it to cope with changes in instability throughout time.

In order to properly cope with changes in market unpredictability, each component of the framework must be capable of doing so. Finally, because no regular exchanging model is likely to erase every single false sign, each central framework segment displays vigour when confronted with a routine false flag. False alarms arise regardless of the method used to define the features of the current market. This forecast methodology (Computing, 2019) is meant to handle these scenarios and ensure robustness to market changes and false alerts. (Cartea et al., 2016) provide a strategy whereby an algorithmic trader forecasts.

Price movements over time and changing their estimates, the trader improves their strategy. Also from, three basic concepts underpin the framework, which is mentioned below: In this market-changing strategy, order to simply reversion is employed to take advantage of diverse features of company sectors. Market - frequent trading reveals economic activity in a variety of scenarios. Each component of the system is based on the notion of unpredictable and dynamic assessments, allowing it to adapt to changes in instability over time. Each component of the system must be capable of dealing with variations in market unpredictability in order to function successfully. Furthermore, since no normal swapping system is sufficient to remove every single false signal, each core framework section reacts with vigour when presented with a common false flag. Regardless of the approach used to describe the characteristics, false alarms occur.

The gains of an algorithmic trader are higher and more predictable than those of a trader who is unable to learn from market dynamics or forecasts. Despite the fact that the trader is employing a directed approach, profits are generated from both the spread and the increase in inventory capital. A capacity changes is limited by higher price volatility, although this disadvantage mitigated move in lockstep. Every set of various is founded on the idea of unforeseen and vibrant evaluations, which allows it to respond to different in destabilisation over time. Evey set of various is founded on the idea of unforeseen and vibrant evaluations, which allows it to respond to different in uncertainty over time. Finally, we show that the numerical approach converges to the equations. (Beverungen, 2019) investigates the impact of advances in cognitive computing on financial market algorithmic trading (AI).

Highly complicated data processing and computer systems have long fostered the predominance of "quants" in financial trading. Nonetheless, type automated, that focuses intelligence, over the past two decades. Thus according Armin Beverungen, artificial intelligence already, blurring lines "dumb" HFT and "smart" algorithms. Unlike, which focuses individual sector patterns, new types ecosystem by allowing automated trading to integrate a larger data collection, media? (2019, Beverungen) feel that focusing battlefield would help them better adding unique cognition chronologies more redistribution. Financial democratisation and social governance are thwarted by temporalities algorithm must be addressed through a politics of cognition.

2.15 Automated trading

Algorithmic trading refers to mathematical models that are programmed to generate computerised trading orders. There are two forms of trading: high frequency trading and low frequency trading. In a presentation by the founder of the investor's exchange (IEX), who highlights how closeness and fast connections are critical for optimising high frequency trading algorithms, the subcategory high frequency trading is examined in length. After that, the author (Chye & Han, 2018) discusses how algorithmic trading can aid with liquidity in a constrained setting. Following that, the various strategies for developing an algorithm for lower frequencies trading are discussed.

Automated trading relates to the use of electronic trading study of organizational behavior by statistical equations. Furthermore, four equities from the US stock market are chosen to evaluate how sentiment scores and the difference between short- and long-term moving averages affect the stock's price movement. The findings show that public opinion is split.

An ensemble technique for learning maximising using schemes are explored. The three actor-critic-based methods are used to train a DLR agent and achieve an ensemble trading strategy: Advantage Actor Critic (A2C), Deep Deterministic Policy Gradient (DDPG) and Proximal Policy Optimization (PPO). The ensemble approach inherits and incorporates the best aspects of the three algorithms, allowing it to adapt to changing market conditions. To prevent significant memory usage in training networks with continuous action space. (Yang et al. 2020).

Financial markets are a source of non-stationary multidimensional time series that have piqued the interest of researchers for decades. Each financial instrument has unique qualities that change over time, making analysis difficult. Understanding and developing methodologies for financial time series analysis are critical for effective trading on financial markets. In this paper, we offer a volume-based data pre-processing strategy for improving the suitability of financial time series for machine learning pipelines. For evaluating the method's performance, we employ a statistical methodology. To be specific, they proposed a set of research questions, formalise the hypotheses, compute effect sizes with confidence ranges, and reject the null hypotheses.

We also evaluate the suggested method's trading performance using historical data and compare it to a previously published methodology. Our investigation demonstrates that the suggested volume-based technique successfully classifies financial time series patterns and outperforms a price action-based method, excelling particularly on more liquid financial instruments. Finally, offered a method for directly deriving feature interactions from tree-based models and compare it to the SHAP method (Sokolovsky et al. 2021).

The main purpose of the author Takavingofa Manzini is to use the Python programming language to create an effective trading algorithm on the Quantopian platform. This research piece also examines the most prevalent trading tactics in order to generate strategies for the LeoTrade algorithm. The researcher tweaked the Quantopian risk model and a long-short strategy combination to produce a comprehensive LeoTrade algorithm that uses both strategies. Based on the findings, the researcher created tables and visualisations using Jupyter notebook, a python tool in Quantopian.

The LeoTrade algorithm achieved approximately 115 percent returns with a balanced long/short investment and met all of the Quantopian standards. (*Department Of Accounting And Information Systems Algorithmic Trading : Efficiency Of Leotrade Algorithm*, 2018). Adriano Koshiyama, Nick Firoozye, and Philip Treleaven examine Artificial Intelligence (AI), Machine Learning (ML), and underlying architecture in future Capital Markets. New AI algorithms are created on a regular basis, each mimicking a different aspect cognition, comprehension. Deep Learning, Adversarial Learning, Transmission, and Meta Learning are the four forms of learning that are now impacting society.

Regardless of the fact that all these training modes have been around for over a decade in AI/ML, their importance has expanded as a result of greater data, processing resulted in the development of novel important application models (e.g., Long Short-Term Memory, Generative Adversarial Networks) (Natural Language Processing). Because these new models and applications will define future Capital Markets, it's critical to grasp their computational powers and limits. Because machine learning algorithms virtually self-program and change continually, financial institutions and regulators are increasingly worried about preserving some kind of human control, concentrating on Algorithmic Interpretability/Explainability, Robustness, and Legality.

For example, there is concern that in the future, the ecology of trading algorithms across diverse institutions would 'conspire' and become inadvertently dishonest (cf. LIBOR) or vulnerable to manipulation via polluted datasets (e.g. Microsoft Tay). Excessive algorithmic complexity could lead to the formation of new and distinct types of systemic threats. The goal look into, machine learning, other similar, as well as their computational strengths and weaknesses (Soares Koshiyama et al., 2020).

In latest days, the rapid advancement of innovation has risen. When integrating, the first practical issues to overcome are. As a result, knowing how to determine the market's learning features is critical. Strategy, known as generalisation changes, is used to handle real-time financial data and can de-noise the data. After that, the candlesticks are broken into various subparts based on a present spatial-temporal relationship, and the subparts are clustered to produce learning characteristics. The model also incorporates the learning features organized by the aforementioned K-lines, as well as the deep reinforcement learning approach, which provides adjustment the resulting in a transaction strategy. The model's performance is evaluated using a variety of commodities. The proposal approach is contrasted to various methods for feature learning, such as price, fuzzified price, and K-lines. To compare the accuracy of the proposal strategy, use networks (Fengqian & Chao, 2020).

Quantitative investing tries to maximise return while minimising risk over a group of financial instruments in a sequential trading period. The quick rise promise technology increased in recent years. In the meantime, artificial intelligence (AI) has presented new challenges to the quantitative investing system, boosting the quantitative investment method. In particular, they must be updated; additionally, they must be more powerful; and, finally, there are some unique problems in applying AI technology to diverse financial duties. (Yang et al., 2020) created and built Glib difficulties investing. Glib aims to maximise the potential of AI technologies in quantitative investment, empower research, and create value.

Investors can forecast price fluctuations. Several scholars have worked on using (ML) market patterns in recent years. However, their study was limited to a time, accounting. Furthermore, findings were determined to be vanishingly small. (Lv et al., 2019) simulated different machine learning methods on large-scale stock datasets and track daily exchange performance of equities with and without fee income and also tested six basic machine learning sophisticated constituent companies (CSICS) from 2010 to 2017.

According to the findings of the study, outperformed for majority in directional assessment indicators. Surprisingly, when transaction costs are factored in, the performance of certain traditional ML methods isn't all that different from the best DNN models. Furthermore, all ML algorithms' trading performance is affected by changes in transaction cost. When compared to traditional ML algorithms, DNN models outperform them in terms of transaction cost. Our findings are crucial in choosing the best stock trading algorithm in various markets.

2.16 Digital stock market

In the digital stock market, market making (MM) tactics has played a critical role. Trading with MM strategies that aren't predictive, on the other hand, is risky. (2014, X. Li and colleagues) designed and implemented a two-tier system that comprises a supervised training trading signal generator and a multi-managerial method based on events. Our findings critical in determining which stock trading algorithm is optimal in various markets.

Trading with MM strategies that lack the ability to predict outcomes, on the other hand, is risky. Developed and implemented a two-tier system consisting of a supervised learning trading signal generator and an event-driven approach. The two most crucial jobs in stock trading when employing machine learning algorithms to obtain long-term benefits are feature extraction and trading strategy building. Several ways for creating trading strategies based on trade signals have been developed in order to maximise the rewards. The studies are done out utilising three conventional Deep Reinforcement Learning models on ten Indian stock datasets: Q-Network in Depth. A two-tier system comprising of a training data able to trade signal source as well as an occurrence multi-manager method was successfully deployed.

In the broader financial industry, the stock market is extremely important. The topic about how to obtain in order reap effectiveness has been researched for quite some time. (Y. Li et al., 2020) Experimental data is used to demonstrate the model's reliability and availability, and the model is contrasted to a standard model to illustrate its advantages. This study demonstrates the applicability, as well as benefits, from standpoint processes. Study (Asgari & Khasteh, 2021) uses solve the challenge of detecting the direction of three cryptocurrency markets.

Price data and technical indicators are among the data sources we use. These classifiers are used to create a trading strategy for certain markets. Our test findings on unseen data demonstrate that this strategy has a lot of potential for assisting profit from. Our largest gain over a 66-day period is $860 per 1800-dollar investment. The system's dependability and accessibility are demonstrated using observational evidence, and the system is compared to a conventional system to show its benefits. The drawbacks of these approaches are also explored, as well as their possible consequences for the Efficient Market Hypothesis. We show that with only training data acquired through a well-designed grow. In other words, teach to deal least as well as trader T. DeepTrader is a trading system that uses Level-2 market data.

Surprisingly, clear projections about pricing. Solely with the desired result being DeepTrader creating Q when it is given S with obtained quotation. To put it another way, we educate our DLNN to function like T by feeding. Our methods, believe, can hypothetically build an explainable clone of any trader T (Wray et al., 2020) utilising "black-box" deep learning methodologies. Deep neural networks are artificial neural networks (ANNs) with several hidden layers (DNNs). Because of their outstanding predictive features, such as their resistance to overfitting, they've recently sparked a lot of.

However, its usefulness examined due to its computational complexity. This research examines how deep neural networks (DNNs) can be used to forecast future financial market activity. We demonstrate how and where to back test a basic trading strategy over 43 different Commodity and FX future mid-prices at 5-minute intervals by covering the configuration and training approach in depth.

The authors employed an open source and available as open source code (Dixon et al., 2017). There are numerous typical trading tactics used by individuals and investment firms. The Warren Buffett-endorsed purchase-and-hold technique, in which investigators buy one or a set of companies and let time make money, is a simple but effective one. Individual investors frequently used the stock chart to make decisions based on their own judgement, which was often not dissimilar to random guesses due to a lack of information.

Another current idea is (Zhao, 2019) wanted to know which of the above tactics was the most effective in today's stock market. In order to get a current picture, we used these strategies on four different prominent stocks in 2017 and 2018 and tracked their performance over 100 different time frames. Our findings provide quantitative trading advice for regular investors by measuring the return and risk of each approach. The best technique is to employ the buy-and-hold strategy if the stock market is steadily expanding. Using a decent machine learning method should help if the stock market is relatively stable. Individual investors should invest more time and effort in finding a promising stock.

2.17 Portfolio management

To make money and diversify their risks, people acquire and sell pieces of various assets over a specified period of time. In order to do this, portfolio managers that use quantitative trading techniques look for trends in historical data using statistical or mathematical techniques.

However, for a single asset, both systems produced discrete trading signals, which may not have represented the best trading choice under actual trading conditions. Some research have taken the portfolio problem into consideration to overcome this flaw—concentrating on a single item. Market price movements are extremely unpredictable and difficult to simulate. As a result, the methodologies outlined above suffer from a lack of generalizability. In other words, a particular model could perform well on the training set but badly on the testing set. However, rather than concentrating on a specific trading choice, these research merely sought to forecast future market patterns. Consequently, to increase how broadly our portfolio management methodology may be applied.

A thorough RL framework that has the following enhancements:

• To improve computation efficiency, offered an adaptive sampling method.

• Improved the generalizability of the model

• In order to pretrain our model and take into consideration potential risks throughout the training process, employed the market representation network. Using the market representation network model, rose while maximum drawdown (MDD) was reduced by 40%.

2.18 Stock market prediction

Wang, W., Ma, Y., and Han (2021). In actuality, a wide range of variables, including political developments, corporate news and policies, prevailing economic conditions, interest rates, and investor mood, affect stock prices. Support vector regression (SVR) and random forecasting are two examples of recent machine learning models for stock market prediction that have shown pleasing results (RF). The basis of deep learning technology, artificial neural networks (ANNs), has also been extensively employed for stock market forecasting. There are three of the deep learning technologies that are often utilised in financial time series forecasting.

Given the effectiveness for forecasting, several have used to yield gratifying outcomes when selecting stocks for a portfolio. In actuality, successful stock selection is essential to portfolio management. Individual typically predict investment before determining the best weighting for each stock to form a portfolio. Investors must thus determine the correct investment weight for each chosen stock after the stock preselection procedure before executing trading investments. Modern portfolio theory serves as the basic foundation for this process. Various models are used in contemporary portfolio theory to determine the ideal weight of each item in the portfolio. Maximise for the provided assets under various constraint scenarios. It is possible to determine the ideal.

An optimization model for the portfolio is created via the forerunner contemporary, by simultaneously maximising expected return and minimising investment risk. This approach results in an efficient frontier that displays the asset allocation that reduces overall risk while maintaining a given projected return. The efficiency frontier determines the ideal investment plan for each level of expected return. The MV model is difficult to use in practise due to issues including processing hypotheses that limited. As a result, several models are suggested to address these problems.

For instance, the mean-absolute deviation (MAD) model developed by Ma, Y., Han, and Wang (W.) in 2021 substitutes absolute deviation for variance as a risk indicator. The CVaR model was also put out as a way to overcome these shortcomings. Use the omega model to optimise your portfolio. The truth is that effective stock selection is crucial to managing a portfolio. Usually, people make investing predictions before figuring out the ideal weighting for any asset in a portfolio. After the stock preselection process, investors must thus ascertain the appropriate investment weight for each selected stock before carrying out trading transactions. This technique is built upon the fundamentals of modern portfolio theory. In modern portfolio theory, many models are employed to ascertain the optimal weight of every item inside the portfolio. By maximising the Omega ratio, this strategy aims to maximise likelihood a beyond certain threshold amount.

The mean historical return is widely used as the projected return in conventional portfolio optimization strategies, however this strategy has impact behaviour and yields incorrect predictions of short-term returns. Additionally, as market sentiment has a large influence on stock price in the near term, it is improper to take the particular stock. Portfolio optimization models financial investing. Many academics choose to use projected return instead than anticipated return when developing portfolio optimization models. Some research have also tried to incorporate new predicted outcomes into the original an effort improve performance.

The development of a portfolio optimization model specifically provides a comprehensive foundation for combining stock projections. They develop a portfolio optimization model based on the mean-variance-skewness model with eleven target functions. The linearity and normal distribution hypotheses, on which the ARIMA model is based, could not be consistent with stock return series. Machine learning models outperformed the ARIMA model without these restricted assumptions.

One of the most sophisticated machine learning techniques, deep learning models, had shown promising results in stock market forecasting. Therefore, it is intriguing and crucial to investigate how standard portfolio optimization methods may be enhanced by merging machine learning and deep learning models to predict returns. They are not aware of any study done explicitly on this subject. This study also focuses on creating portfolio optimization models that implement the outcomes of these models according to the paradigm proposed by Ma, Y., Han, and Wang in order to fully exploit their advantages in return prediction (2021).

To fill up the gaps left by other investigations, this study offers two new contributions in this field. Algorithms significantly outperform conventional time series models, guaranteeing that only are picked creation. Additionally, because fewer assumptions, they are better suited for use in real-world situations. Second, by merging the anticipated results of the two models, our technique for the first time enhances both the conventional procedures. Key elements of the conventional are combined with deep machine learning for return prediction these complex optimization models.

Study examines most recent four years and focuses on data from 2007 to 2015 in order to analyse the effects of the proposed investing strategies.

2.19 Techniques for intraday trading using time series

Numerous models have been used to explore stock price forecasting. For the purpose of predicting financial exchange rates, autoregressive moving average (ARMA), and autoregressive generalised autoregressive conditional heteroscedasticity (AR-GARCH). The experimental simulation showed that the ARMA-GARCH model combination results in a GEM algorithm that performs better than others. To generate projections, four MLA models were created. The results proved that deep learning worked better than previous methods. The second-best method was supports vector regression, which had a lower error rate than neural network and random forest techniques. Because of the T+1 limitation, which makes intraday trading in China's stock market difficult, most research only address price prediction. This is the first study of China's intraday trading strategy.

Furthermore, a comprehensive review of twenty-five papers on neural network applications in finance was provided. To choose the most helpful inputs for a prediction model, Wang, Q., Zhou, Y., & Shen, J. (2021) They applied various inputs to support vector regression and multilayer perceptron (MLP) networks needed the various that were the subject of comparison research. Additionally, presented heuristic models outperformed the data mining approach in terms of outcomes. Additionally, the mean square error levels of the MLP and SVR approaches were identical.

Technique using logistic regression to predict the trend in stock prices for the following month in 2009 based on the current month. As a research case, they selected Shenzhen Development stock A (SDSA) from the RESSET Financial Research Database. Their forecasting model was less complex and at least 83% accurate when compared to other models, such as the RBF-ANN forecasting model. The Artificial Neural Network (ANN) and ARIMA approach were used by Wang, Q., Zhou, Y., and Shen (2021) to forecast. According to technique produced fewer errors than the ARIMA method.

In order to assist investors, managers, decision-makers, and users in making an informed choice projected businesses Jordanian stock market using the K-nearest neighbour algorithm and nonlinear regression approach. To forecast the stock price movement, they also took into account the historical time series of the S&P 500 using MLP, CNN, and LSTM recurrent neural networks approaches. As a consequence, they suggested that neural networks might anticipate financial time series for upcoming movements and suggest further strategies to enhance the outcomes. Recent years have seen some study towards developing algorithms for deep learning to forecast stock price. For instance, to evaluate the accuracy of stock market forecasts using machine learning models.

Four MLA models were produced in order to provide forecasts. The outcomes demonstrated that deep learning was superior than other approaches. Supports vector regression, which had lower error than neural network and random forest approaches, was the second-best approach. Since intraday trading in China's stock market is challenging due to the T+1 restriction, the majority of studies exclusively focus on price prediction. The investigation of China's intraday trading technique is novel. In addition, a thorough analysis of 25 publications on the use of neural networks in finance was offered.

Three equities from each of ten categories to prevent unanticipated volatility of certain industry or firms. They open positions for each stock in the portfolio on the first trading day. The following days, trade twice daily in an effort to capitalise. There is a 1.5-hour pause at midday and just 4 hours of trade time each day. In order to train the multilayer perceptron. Forecast stock price between 13:00 and 14:50, this model is employed.

2.20. FinGPT language:

FinGPT is a major language model available as open source for the finance sector. FinGPT adopts a data-centric methodology as opposed to proprietary models, giving practitioners and researchers transparent and easily accessible resources to build FinLLMs. We emphasise that in order to create FinGPT, an autonomous data curation pipeline and a lightweight low-rank adaptation approach are essential.

As stepping stones for users, we also propose some potential applications, including low-code development, algorithmic trading, and robot advising. By working together with the open-source AI4Finance community, FinGPT aims to democratise FinLLMs, promote innovation, and provide new opportunities for open finance (Hongyang (Bruce) Yang, 2023).

• *Democratisation:*

By revealing unrealized potential in open finance, FinGPT, an open-source platform, seeks to democratise financial data and FinLLMs. • Data-centric approach: FinGPT takes a data-centric approach, recognising the importance of data curation. To ensure high-quality data, it applies strict cleaning and preprocessing techniques for a variety of data formats and kinds.

• *End-to-end framework:*

FinGPT uses a four-layered, full-stack framework for FinLLMs.
- Data source layer: With real-time information capture, this layer addresses the temporal sensitivity of financial data and ensures thorough market coverage.

- Data engineering layer: This layer addresses the inherent challenges of high temporal sensitivity and poor signal-to-noise ratio in financial data and is prepared for real-time NLP data processing.

- LLMs layer: This layer ensures the relevance and accuracy of the model by mitigating the extremely dynamic nature of financial data using a range of fine-tuning approaches.

- Application layer: This layer illustrates the potential applications and demonstrations of FinGPT in the financial industry.

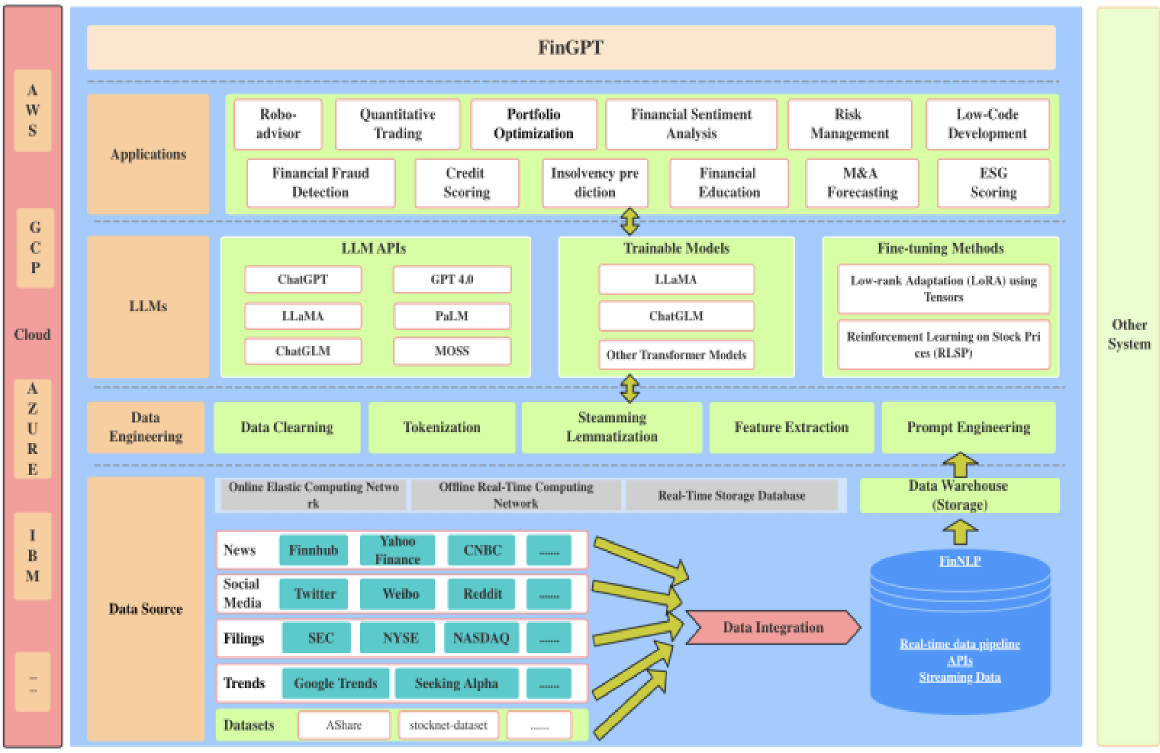


Figure 2.1: FinGPT Framework

2.21. Quantum computing with algorithmic trading

(Khalil mostafa, 2023) By examining the relationship between algorithmic trading and quantum computing, Khalil Mostafa was able to clarify the challenges and potential that come with integrating quantum technology into the financial industry. This investigation offers insights for traders and developers navigating the changing financial environment within the domain of quantum technologies, from investments in quantum computing businesses to quantum algorithms for portfolio optimisation.

(Nisar Ahmad, 2023) Nisar Ahmad explored the possible effects of quantum technologies on the financial landscape by examining the relationship between algorithmic trading and quantum computing. The quantum frontier brings possibilities and difficulties, from improved optimisation to new algorithmic techniques. For algorithmic trading to evolve, traders and engineers must traverse this unexplored region in order to fully utilise the potential of quantum computing.

CHAPTER III:

METHODOLOGY

3.1. Introduction

The process by which findings are achieved, or what was done to arrive at certain conclusions, is described by the research methodology. The chapter provides information on the methodology the researcher employed to gather data and acquire the algorithm's findings. The systematic, organised, and concentrated collecting of data with the goal of learning knowledge to answer research questions is referred to as research methodology. In addition to employing a variety of research tools for data analysis, the researcher tested (backtested) the effectiveness of the LeoTrade algorithm using the dual momentum approach using the Quantopian API and Zipline API. The APIs are essentially identical, however Zipline gets daily data in CSV (comma separated value) files from algorithmic trading servers while Quantopian API operates online. Both are used for backtesting.

Python programming was employed in this study to carry out the method. The Quantopian API's primary language is one of the causes. It would be simple to implement Python on the algorithmic trading system.

3.2. Positivist

According to positivists, reality is stable and can be viewed and described objectively, that is, without interfering with the phenomena being researched. They argue that observations should be reproducible and that phenomena should be separated. This frequently entails altering reality by changing only one independent variable in order to find patterns in and establish connections between some of the components of the social world. In order to comprehend the consequences of each limitation and parameter, the researcher regularly examined as many algorithmic trading algorithm outputs as feasible. The researcher will employ simulations and experiments in accordance with the positivist research philosophy.

3.3. Research approach

A quantitative research methodology was utilized by the researcher. The researcher picked quantitative research because it makes an effort to use certain techniques to make data processing easier. The quantitative research technique was chosen by the researcher because it enables analysis of the effectiveness of algorithms utilising mathematical Python tools integrated within the Quantopian API.

In order to overcome the limitations of using algorithms alone, a mix of research techniques may be necessary to examine algorithms. The algorithm was developed by the researcher using six approaches.

3.4. Examining pseudo-code/source code

The simplest method to attempt to comprehend an algorithm is to look at its pseudo-code and create flowcharts from it. The first step is to carefully disassemble the source code and/or pseudo-code, picking apart the set of rules to ascertain how the algorithm functions to convert input into a result. In actuality, the researcher meticulously went through the documentation, code, and programmer comments, figuring out how the algorithm functions to process data and generate results, and deciphering the translation process used to create the algorithm from Quantopian blogs.

3.4.1. Deconstruction:

Where you simply read through the code and associated documentation to figure out how the algorithm works. The researcher has used two algorithms on two strategies i.e. momentum and mean reversion in order to select the most suitable for this research chapter.

3.4.2. Genealogical Mapping:

Where you ‘map out a genealogy of how an algorithm mutates and evolves over time as it is tweaked and rewritten across different versions of code’. The researcher has chosen the momentum strategies since they are the most researched because of they are more variations of momentum strategies than mean reversion strategies.

3.4.3. Comparative analysis:

Comparative analysis shows how the same fundamental job may be done across a variety of programming languages and operating systems. The Quantopian API online and Zipline API offline were utilised by the researcher to compare the outcomes of backtesting, and Python was used as the programming language to develop the trading algorithms.

Disadvantages:

Some approaches have drawbacks, including the following:

Code is frequently disorganised and difficult to understand;

3.5. Engineering in reverse

A researcher interested in the algorithm at the core of the code's operation is left with the option of attempting to reverse engineer the built software in circumstances where the code is left undocumented. In order to create a model and other implemented codes with their findings, the researcher employed benchmarks on which an efficient algorithm would run.

Disadvantages

As a result, they often only give hazy views of how an algorithm operates in practise rather than revealing its true makeup. As a result, algorithms are typically encrypted and secured.

3.6. Examining how algorithms function in practise

You should also research the algorithm's consequences in the actual world rather than concentrating just on how the algorithm is created and the factors that influence it. To examine the effects of algorithms in quantitative trading, the researcher tested with several algorithmic trading APIs.

This would be expensive because most algorithmic trading systems need a cash investment to get started, even if all you do is watch how the algorithms effect the world around you.

3.7. Automatically Creating Code

This approach requires you to sit down and consider how you may translate a task into code. It makes sense that this is what refer to as "auto-ethnography." The actions that the algorithm would perform were generated by the researcher using flowchart diagrams.

3.8. Drawbacks

The procedure is fundamentally subjective and vulnerable to personal prejudices and flaws. However it can be a useful addition to other research techniques.

3.9. Interviews and Ethnographies of Coding Teams

According to this strategy, some of the prior methods' fundamental lack of contextualization can be corrected. It entails conducting in-depth interviews with or closely watching coding teams as they create algorithms (in the manner of a cultural anthropologist). In order to comprehend how trading algorithms are created on BlogSpot websites and Forums, the researcher has observed other developers as they code trading algorithms.

Drawback:

As these coding teams are rare and largely located in American nations, finding them might be difficult.

3.10. Unpacking the full socio-technical assemblage of algorithms

The whole socio-technical assemblage of algorithms must then be analysed in order to get a deeper knowledge of them. This examination must include a discussion of the justifications for subjecting the system to computing logic in the first place.

Researcher has seen other coders create trading algorithms and explain how and when they would be useful or ineffective as well as the significance of picking a strategy. According to the study, Quantopian can offer the platform on which an algorithm may be deployed and made lucrative. Quantopian.com also provides detailed information on its website about how algorithms are integrated into its platform and how it accesses data sources.

3.11. Research design

3.11.1. Research Experiments

To get the intended outcomes, the researcher employed experimental settings. The scientist used quasi-experiments. "In that an independent variable is manipulated, quasi-experimental research is analogous to experimental research. Because there is either no control group, no random selection, no random assignment, or no active manipulation, it varies from experimental research."

The researcher experimented with various parameter settings, additional method combinations, and the selective application of efficient algorithmic trading strategies like the dual momentum approach. Also, the researcher combined long and short values, yielding a variety of findings until the desired outcomes were obtained. The researcher additionally filled NaN values—values returned if the securities did not trade or did not exist at the given minute—into the algorithmic trading process of the programme using the NumPy Python library.

Trading algorithms usually follow a simple flowchart as depicted below:

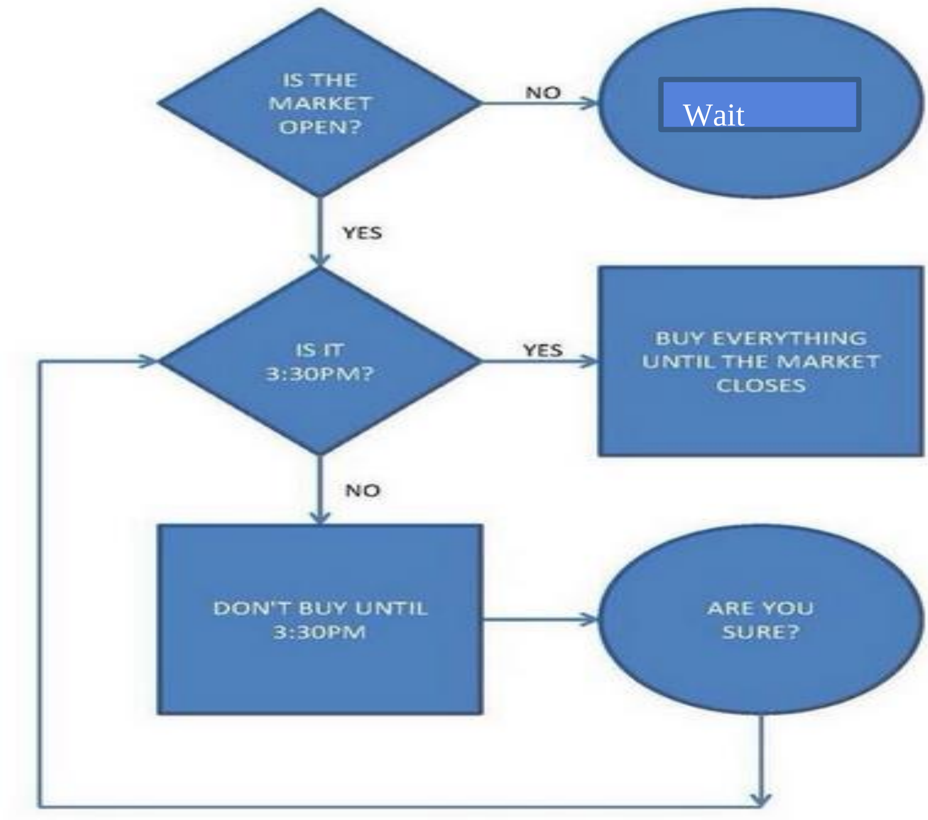


Figure 3.1: LeoTrade algorithm design

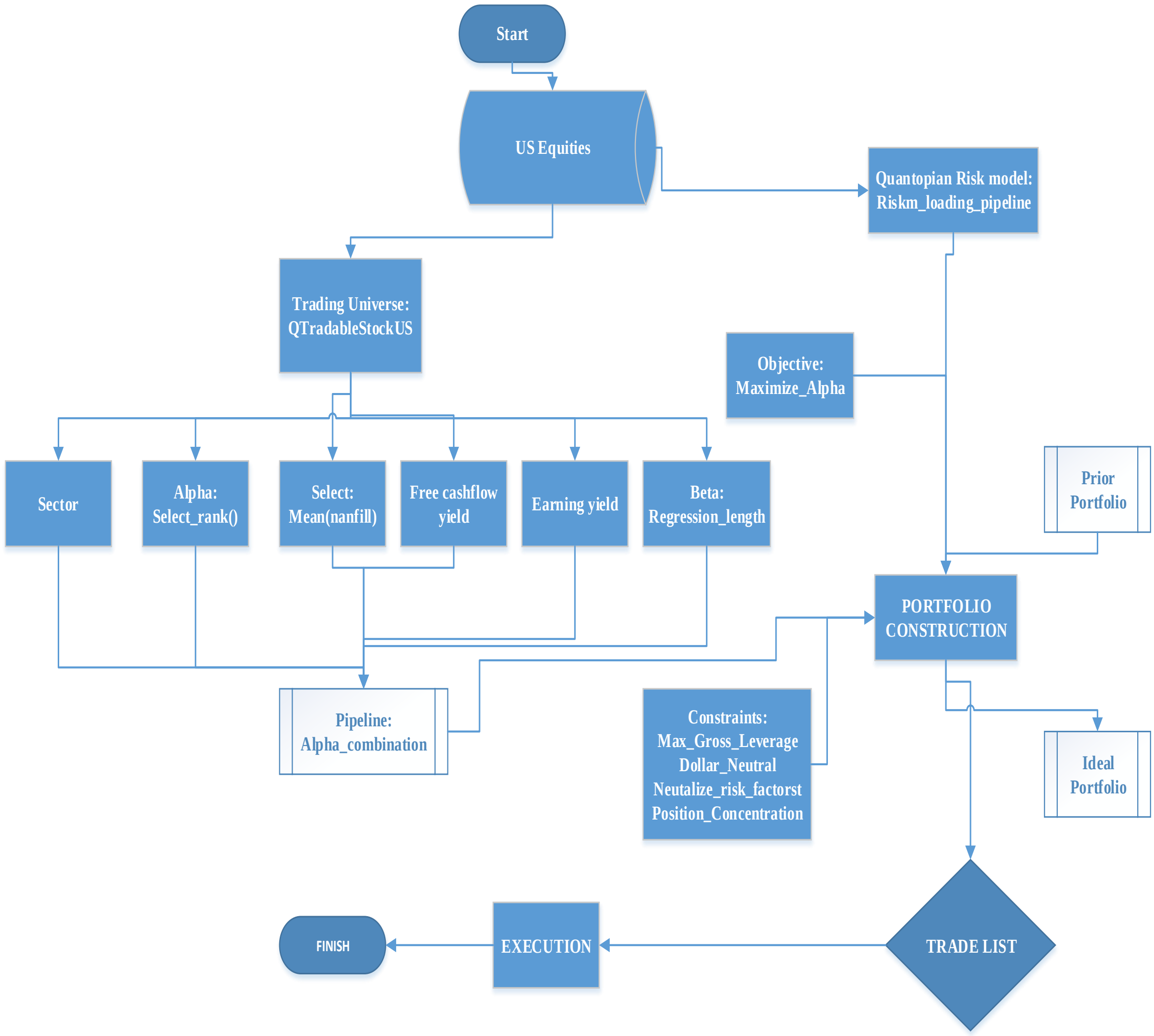


Figure 3.2. LeoTrade algorithm uses US Equity data input then uses the workflow above.

3.11.2. Steps:

1. **Universe Selection:**

   Establish the range of tradable elements; the range should be wide yet contain some self-similarity to allow for the extraction of relative value. Also, it must to get rid of any illegal or difficult to trade items.

2. **Single Alpha Factor Modeling:**

   Identify and assess distinct expressions that rank the assortment of stocks in your universe.

3. **Alpha combination:**

   Create a final alpha by combining a number of separate alphas into one, which has more predictive value than the best single alpha. This frequently occurs as a result of noise in one alpha cancelling out noise in other alphas, enabling signal to get through.

4. **Risk Model:**

   Identify and quantify the group of risk variables you want to utilise to limit your portfolio.

5. **Portfolio Construction:**

   Use a procedure to create a target portfolio that reduces risk according to your model using your final combined alpha and risk model.

6. **Execution:**

Put in place an algorithmic trading procedure to switch the existing portfolio, if any, to the desired portfolio.

3.12. Data Analysis

Inspection, cleansing, transformation, and modelling of data are all steps in the analysis process, which has the objectives of identifying relevant information, advancing hypotheses, and assisting in decision-making. It entails evaluating and presenting data that was gathered from the topic.

To evaluate the effectiveness of the method, the researcher has relied on graphical representations of its performance. The Quantopian platform already has data analysis programmes like Pandas, Matplotlib, etc. To make the algorithm more effective, all of the data that the researcher has discovered has been represented in the algorithm.

3.13. Research ethics

The researcher has discovered moral concerns with trading algorithms. One of the problems the researcher discovered is accuracy. The platform's data is bias-free, and the LeoTrade algorithm has chosen a backtesting period from 2012 to prevent out-of-sample data under 2008 as much as possible. This has resulted in reliable data.

Quantopian has guaranteed the researcher that the algorithmic trading data utilised is reliable and free from bias. Also, Quantopian's method is licenced to the researcher, and all of the user's used scripts are open source and free to copy and modify.

3.14. LFT Strategies

LTF, which can be modelled by quant traders, skilled programmers, and financial institutions with sufficient technological skills, is a more practical approach to apply AT by people with limited technology and in close proximity. Whereas LFT is enhanced based on a design philosophy, HFT is measured in milliseconds to microseconds and is optimised by being close to stock exchange data centres.

The LFT algorithm can be based on selecting certain preferences from the whole stock market, such as the highest market capitalization equities or the most liquid securities in the top 99 percentile. Back testing software uses an algorithm to provide a return/risk metric that is tailored to a trader's approach. When the strategy has been tested and is ready for live algorithmic trading, the software may be connected to an interface on a broker's platform via an artificial programming interface (API).

LFT-implementing algorithms may be tested using specialised back testing platforms. Quantopian is the most pertinent back testing engine because the chapter's emphasis is on the US equities market. Following platform selection, data such as daily closing prices, market cap, volume, and other factors that influence stock trading should be accessible with ease. Free back testing services and access to financial market data are provided by the free open source firm Quantopian. The design theory makes up the third component of the LFT.

The design theory may be divided into three categories that include sentiment analysis, machine learning approaches, and technical indicators like moving averages. The most attention has been paid to technical indications in recent years when predicting price changes. The most popular indicators for detecting changes in price are moving averages (MA). According to a recent study, only the short-term 21-day MA and the long-term 200-day MA are employed as moving averages, and the difference between them indicates whether to short or long a stock. According to the research, investors consistently underreact to fundamentals and projected price patterns. Use several return models, investment time horizons, and factor models to develop their hypothesis on investors' under reaction to price movements in stock.

The strategy's gain comes from investors' underreaction to the moving average distance (MAD). They contend that this is the case because the majority of investors base their judgements on readily available information, which can occasionally be useless.

The anchor bias occurs when investors begin to stray from the 200 Day MA as a result of news on their underlying asset. Additionally, the study comes to the conclusion that positive sentiment impacts price increases when the spread between the short-term MA and long-term MA is large and positive (21-day MA above 200-day MA), while negative sentiment has little effect on price declines because investors anchor to the lower long-term MA. Positive announcements have less of an impact on the price because under reaction is minimal in a negative spread (200-day MA is above 21-day MA), where bad sentiment (i.e., negative earnings surprises, sell recommendation announcements, and seasoned equity issues) has a greater impact on price drifting down. Using this method, you may build an LFT algorithm for a single stock or a portfolio of stocks by looking at how rewarding the returns are in such a strategy.

To demonstrate the validity of incorporating a technical indicators method into the overall plan, an algorithm concept will be provided. Machine learning is another method that may be utilised to create LFT algorithms. This method uses a large sample of data, in this instance historical market prices (time series), and after sufficiently enough training, the algorithm learns how to handle the data's volatility (price variations). Before making an initial investment, machine learning approaches need time to figure out how to replicate and outperform the market as represented by an index. The Spyder (SPR) ETF (Exchange Traded Fund) index, which mimics the S&P 500 market index, serves as the index on the Quantopian platform.

The ability to ride market volatility after being educated is one benefit that various machine learning approaches provide. In a study, using three approaches to forecasting security prices, the quantopian platform, that MA have rewarding profits at first but lag in the end and vice versa for machine learning. However, in research chapter, conclude a combination of sentiment analysis and machine learning have the most rewarding return/risk framework.

Sentiment analysis evaluates strong, viable texts from the Sentdex database retrieved through the Quantopian platform and generates a sell signal for scores below -1 and an acceptable purchase strategy for scores above 3. While the use of machine learning will reduce risk exposure. Many AT techniques have demonstrated risk-reward ratios that are quite favourable. Based on the anchoring bias theory and MAD factor in the study, this chapter suggests an LFT.

An algorithm can read millions of articles and determine a precise sentiment score or attitude for each security in less time than it would take a person to read one item and determine sentiment. Also, the suggested algorithm model is examined using the Sentdex free sentiment database version. Sentdex is significant because it generates, via a score ranging from -3 to 6, rich yet pertinent information from articles published by reliable sites like the Wall Street Journal, CNBC, and Forbes. Four active equities are picked to represent both strategies in the suggested strategy.

Figure 3.3: Nasdaq (Time series)

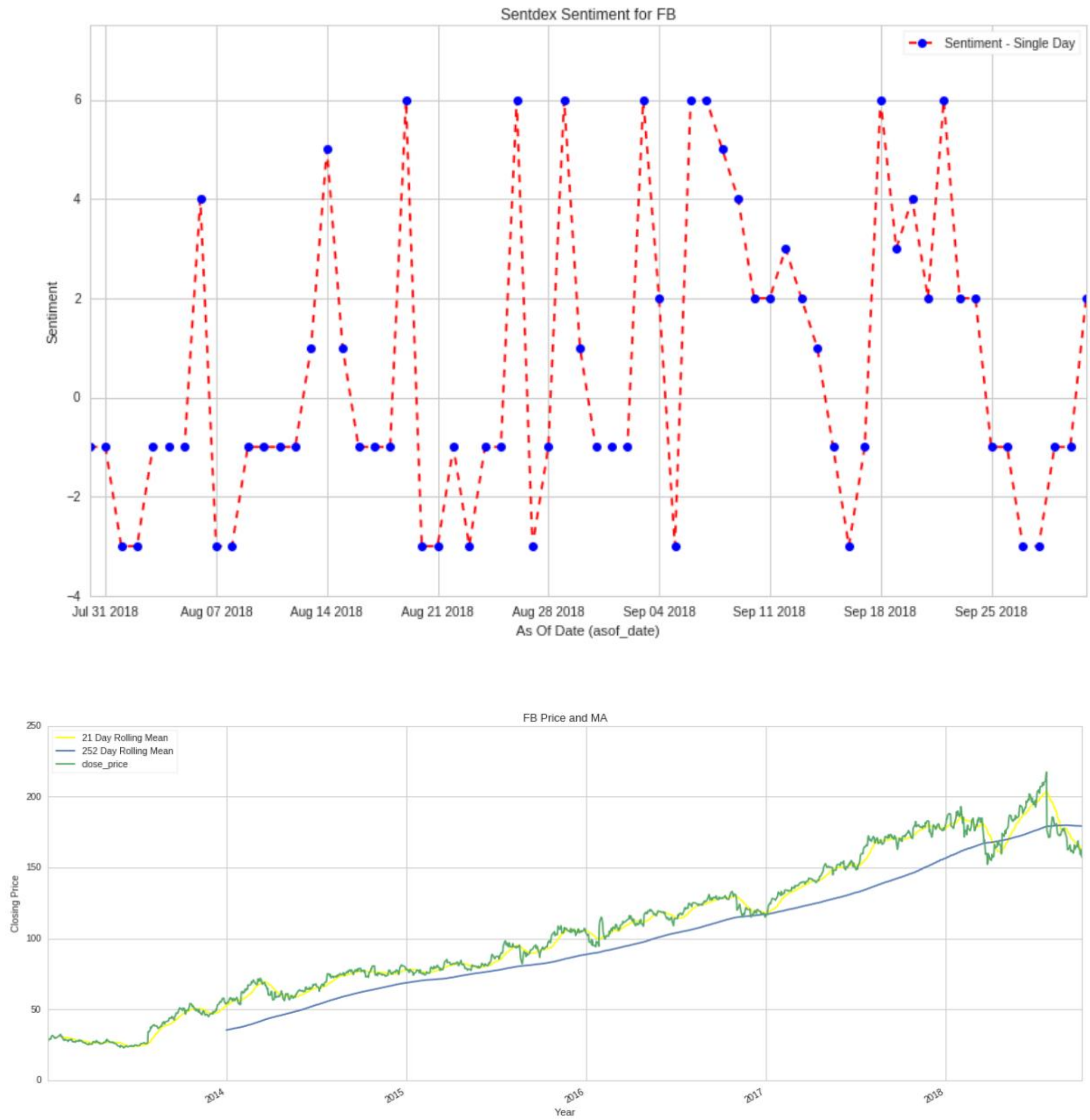


Figure 3.4: Facebook (Time series)

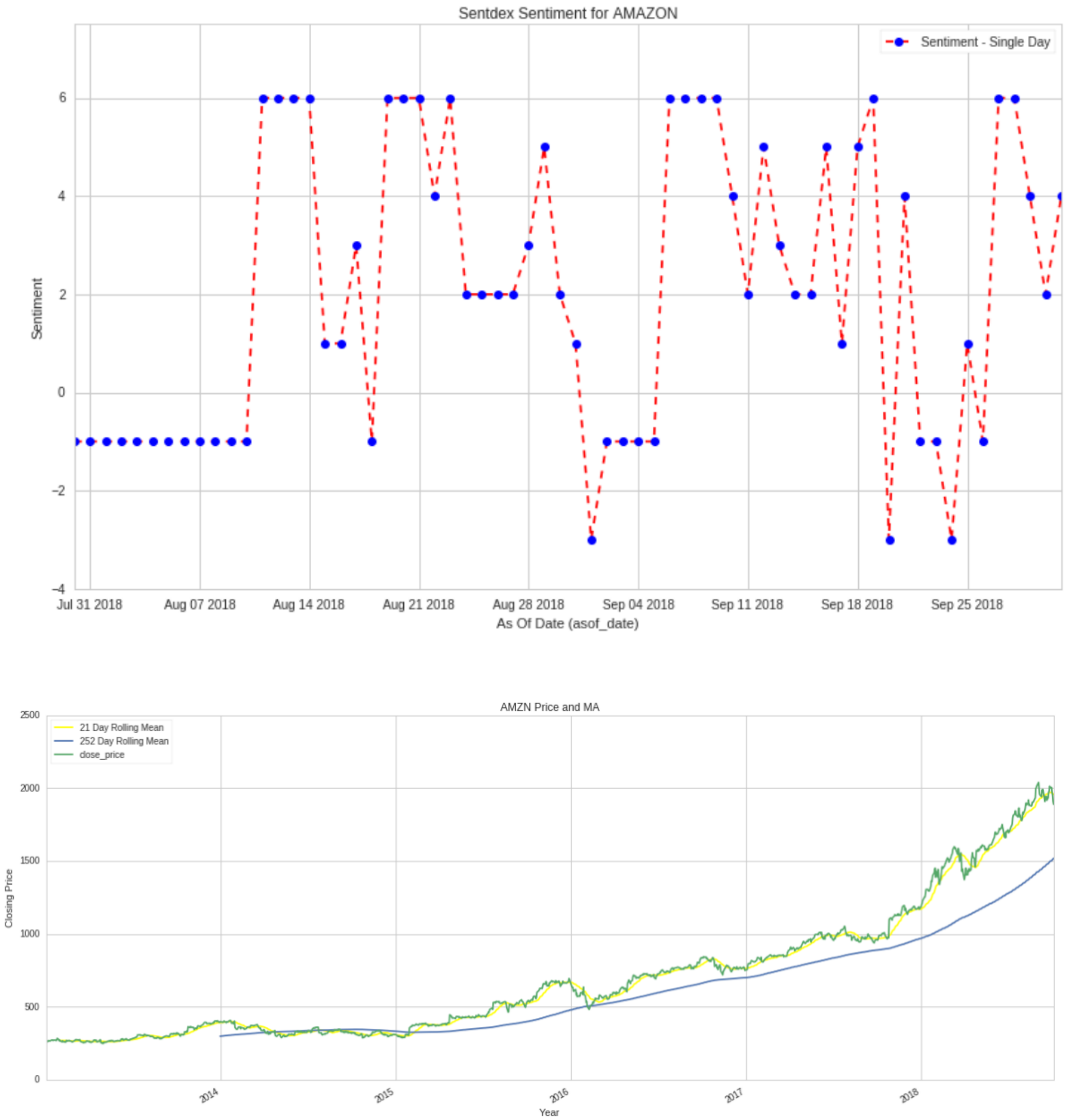


Figure 3.5: Amazon (Time series)

Nasdaq, Facebook, and Amazon are the stocks. Market capitalization, volatility, and to some extent liquidity of these companies vary, but the goal is to address an observable approach through an empirical research that may be elaborated upon in the future. The observation period lasted for two months, from July 30 to October 1 of that year. The workspace for research is a Quantopian research notebook since it offers direct access to historical prices for securities and sentiment scores for each day of the underlying security. After obtaining historical closing prices for each stock, the 21-day MA and the 252-day MA are graphed to show the short-term and long-term MA for that stock, respectively. When the 21-day MA is positive (negative), the gap between the MA Subsequently, for the months of August and September 2018, the sentiment scores and closing prices for the equities are graphed, accordingly.

The goal is to determine when a feeling affects the price of the stock. The four equities demonstrate that a certain sentiment score causes a security's price to change about seven days later. As an illustration, the sentiment scores graph between November 10, 2018, and November 14, 2018, has a discernible link with the trajectory of the closing stock prices between November 17, 2018, and November 21, 2018, for any of the selected stocks. Not all stocks, however, respond the same way to a mood score. For instance, the Facebook stock reversed the positive spread of the MA in September, when the Long-term 252 MA was higher than the 21 MA. As a result, in the two aforementioned equities, negative sentiment has a greater influence on price declines than good sentiment does on price increases. Contrarily, the Amazon stocks have a significant positive spread, indicating that good sentiment will have a greater influence on price increases than negative sentiment would have on price decreases.

This supports the idea put out by that positive MA spreads often have lower prices than negative MA spreads. The results are shown in the figures above, which also show the roughly 7-day adjustment period for sentiment and the effects of each sentiment score on stock prices.

The data demonstrate that an LFT algorithm may be developed based on the MA spread and the delay in the influence of the sentiment signal score, which may indicate a stock's price change and trend. Hence, a list of requirements that must be met in order for the algorithm to execute a buy or sell order must be established. Using a filter in the back-testing engine (such as Pipline in Quantopian) to search through all US listed and actively traded companies and return the equities with the biggest spreads in their short- or long-term MA is one method of putting this technique into practise. The stocks with the highest positive and negative spreads are chosen and added to the portfolio, however depending on the restriction the algorithm filter produces, the portfolio may only contain one stock or as many stocks as desired.

A further filter can be used based on the capital asset pricing model once the equities have been divided into positive or negative MA spread categories (CAPM). This phase is necessary to distribute the portfolio weights among the securities with the highest alpha, or, to put it another way, the securities that have performed the best relative to the market index, in our example, the SPR ETF. Additionally, the buy or sell strategy will be implemented after reading the Sentdex sentiment score 7 once the most liquid and widely dispersed stocks have been gathered. When the market opens, a purchase order will be executed if the computer detected a sentiment score above 3 and the stock is in the significant positive spread category. The algorithm will execute a sell signal after 7 days have passed if it found with a sentiment score below -1. Without human assistance, the algorithm has to tie up two loose ends in order to maximise efficiency. The first is the exit plan for each trade placed in order to reduce the chance of a downside.

Based on the 7-day delay results of this study and the anchor bias theory, profits will be decreased yet assured. Stock removed from its category and no longer traded by the algorithm begins fall certain supplied. A new with a wider spread over the specified threshold can then be substituted determined multiplying dividing the result. The second problem that has to be resolved, even if it is outside the scope of the research. An open-ended limitation, this one can be predicated on weights that sum up to 1, as long as there are twice as many long stocks as short ones. The algorithm is even capable of combining zero, which executes the same weight for both. Yet the more alpha a stock has in its CAPM, the more weight that it will receive relative to other established. A comprehensive empirical and statistical methodology may be used to further explore this LFT technique, for instance.

3.15. The Proposed FinRL Library

Environments, agents, and applications are the three layers that make up the FinRL library. We initially provide an overview of the architecture before presenting.

3.15.1. FinRL Library

Its characteristics are outlined as follows:

• A three-layer structure Stock market environment, DRL trading agent, and stock trading applications make up the three levels of the FinRL library. The agent layer engages in exploration-exploitation with the environment layer, whether to repeat previously successful decisions or to do new actions in the hopes of reaping bigger benefits. The bottom layer is transparent to the top layer since it supplies APIs to the latter.

Layer 1:

An overview of application layer

Benchmark Single Stock Multiple Stock Portfolio User-defined

Test	Algorithmic Trading	Algorithmic Trading	Allocation	Trading Tasks

Layer 3:

An overview of finance market environment layer

Benchmark Environment	NASDAQ-98 constituents, DJA constituents,	SSE 46 constituents, CSI 270 constituents,	User-import Datasets

• *Modularity:*

Each layer consists of a number of modules, each of which defines a different function. To carry out their stock trading duty, one can implement certain modules from any layer. Also, it is possible to update already-existing modules.

• *Ease of Use, Applicability, and Extensibility*:

FinRL exposes DRL algorithms as modules and was created specifically for automated stock trading. FinRL is made available but not burdensome in this way. FinRL offers three readily reproducible trading jobs as use cases. There are reserved APIs for each layer that enable users to create new modules.

3.16. Time-driven Algorithmic Trading Simulator

3.16.1. Environment

Our trading environments, built on the Open AI Gym platform, imitate real stock markets using time-driven simulation techniques. Our trading agent takes note of a variety of aspects to improve its learning several.

We offer users a range of benefits, including:

- Balance $b_t \in \mathbb{R}_+$
- Shares own $h_t \in \mathbb{Z}_+^n$Closing price $\boldsymbol{p}_t \in \mathbb{R}_+^n$
- Trading volume $\boldsymbol{v}_t \in \mathbb{R}_+^n$.

Various levels let the following characteristics' data to be collected daily, hourly, or even minutely.

3.17. Backtesting with Algorithmic Trading Restrictions

We include algorithmic trading limitations, risk-aversion, and automated backtesting tools to more accurately imitate real trading.

3.18. Including restrictions on algorithmic trading.

A trade's execution results in transaction expenses. Transaction expenses come in many different forms, including broker commissions and SEC fees.

In our environments, we let users

• Set sum of money each transaction, regardless of the number of shares traded.

• Per share percentage: The most popular transaction cost rates are expressed

For stock trading, we also need to take market liquidity into account. To imitate real-world trading in our scenarios. Also, it affects one's trading approach while dealing with varying levels of market volatility.

$$\text{turbulence}_{t} = (\boldsymbol{y_t} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{y_t} - \boldsymbol{\mu})' \in \mathbb{R},$$

Table 3.1. Results of PPO-based single-stock algorithmic trading in the FinRL library.

2019/01/01-2022/12/31	SPY	QQQ	GOOGL	AMZN	AAPL	MSFT	S&P
Initial value	100,00	100,00	100,00	100,00	100,00	100,00	100,00
Final value	114,13	157,55	162,75	187,26	165,67	168,64	128,34
Annualized return	15.04%	31.75%	36.97%	45.07%	37.12%	35.87%	18.07%
Annualized Std	10.05%	28.33%	32.79%	30.12%	26.16%	32.84%	26.78%
Sharpe ratio	1.32	1.04	1.07	1.35	1.27	1.06	0.68
Max drawdown	19.06%	27.94%	28.06%	20.71%	21.67%	27.88%	34.07%

3.19. Presentation

Three use examples are used to illustrate our point: portfolio allocation, multiple stock algorithmic trading, and single stock algorithmic trading. For each use case, the FinRL library offers workable and replicable solutions. In order to provide a benchmark for the quantitative finance community, we choose three use cases and replicate the outcomes using FinRL.

Train a trading agent in FinRL using the PPO algorithm. The Covid-19 market crisis has resulted in a significant maximum drop in figure.

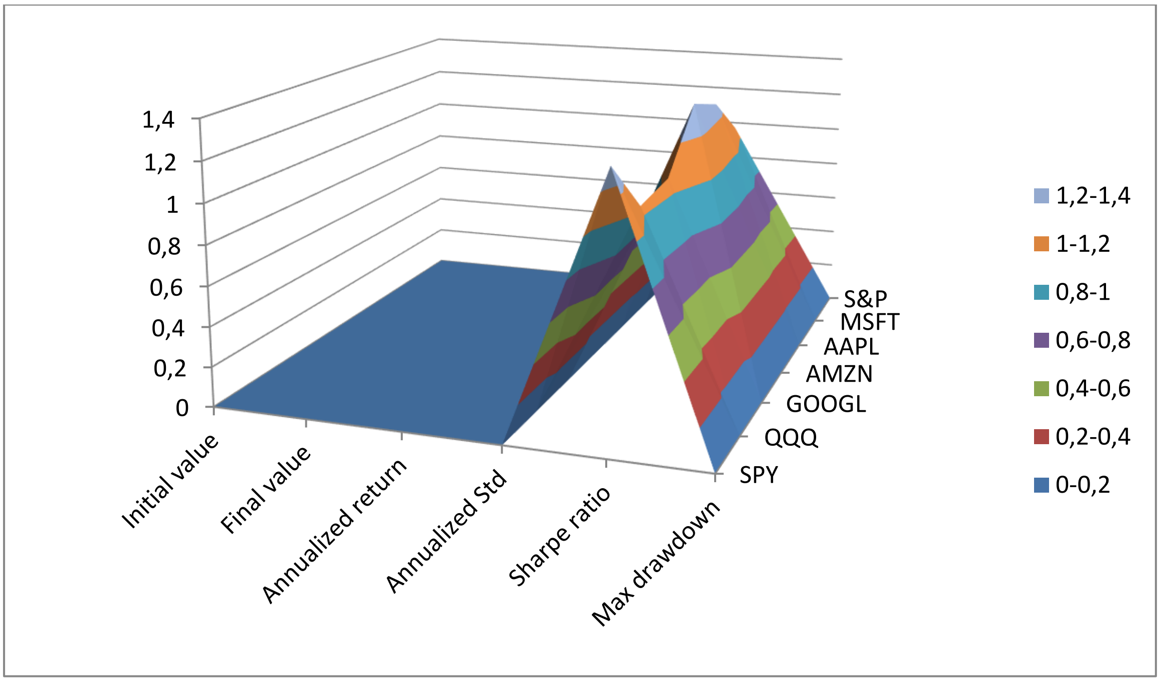


Figure 3.6. The Covid-19 market crisis has resulted in a significant maximum drop.

Technique for creating referred to as meaning makes predictions about a stock's future price movements based on its past performance. According to the theory, there are patterns that may be used to forecast by observing how the stock has historically moved in those time frames. Although our method excludes crucial elements like general market data and the success of other firms, we think this basic model might serve as a useful. This model's implementation requires several specifics handled. Control the "confidence level" of the predictions using probability models like logistic regression, and because crucial exact correct. While the second one has a bigger inaccuracy, nonetheless, the returns are often dispersed at random with a mean of zero.

In order to avoid overfitting, we implemented a ridge penalty on top of logistic regression. We can prevent any one of the calculated coefficients from being excessively big by adjusting lambda. As yesterday is the return that has the most bearing on today, its coefficient would be much higher than the others.

In this thesis, we evaluate a prediction model's performance in two different methods. Given that we may choose a confidence level threshold models be able to predict for every observation. For instance, only predictions of Y = 1 with a probability greater than 0.6 may be counted. Consequently, we are solely concerned with the "true positive rate," which determines how many times the model correctly predicts "up" given a threshold. We will have proof that, at a 5% significance level, our model's prediction outperforms random guessing if we are able to correctly reject the null hypothesis.

3.20. Conclusion

The chapter's discussion of the methodological choices used for this investigation was its main objective. It became clear that the research was quantitative in nature. The chapter also provided an evaluation for data collection procedures and instruments used in researching the algorithm. Research philosophy, design and ethics were discussed in this chapter. Also, the appropriate data analysis process was covered. The representation, interpretation, and analysis of the research data used to assess the algorithm's effectiveness are shown in the upcoming two chapter.

CHAPTER IV:

RESULTS

4.1. Introduction

On algorithmic trading's effectiveness in automating online algorithmic trading from the study approaches employed are summarised and analysed in this chapter. This chapter uses line graphs, tables, column charts, bar charts, and other types of visualisation to demonstrate the effectiveness of algorithmic trading using Quantopian benchmarks. The effectiveness of algorithmic trading is evaluated using the returns, trading universe, portfolio holdings, and other factors detailed in this study thesis' chapter.

4.2. MDP model for stock algorithmic Trading

We use a Markov Decision Process (MDP) to simulate the stochastic character of the dynamic stock market as follows:

- Each stock may be sold, purchased, or held, with the resulting decrease, increase, or stay the same of the stock shares z, accordingly.
- Policy (s): probabilistic distribution of actions at state s, which represents the algorithmic trading strategy at state s.
- A state's expected reward for acting in accordance with policy is represented by the Q-value $Q_{\pi}(s, a)$.

Including Algorithmic Trading limits: issues for practise include assumption and limits listed below represent these issues.

- Let the price and share count vectors for the stocks in the purchasing set be

$$x_t^B = \left[x_t^i : i \in \mathcal{B}\right] \tag{1}$$

$k_t^B = \left[k_t^i : i \in \mathcal{B}\right]$. For the selling stocks, we may define $\boldsymbol{x}_{\mathrm{t}}^S$ and $\boldsymbol{k}_t^S$, and for the holding stocks, $\boldsymbol{x}_t^H$ and $\boldsymbol{k}_t^H$. As a result, the restriction for non-negative balance may be written as

$$b_{t+1} = b_t + (\boldsymbol{x}_t^S)^T \boldsymbol{k}_t^S - (\boldsymbol{x}_t^B)^T \boldsymbol{k}_t^B \geq 0. \tag{2}$$

- Transaction costs: There are transaction costs associated with every deal. Transaction expenses come in many different forms, including exchange fees, execution fees, and SEC fees. Different brokers have different commission charges. Variances, project each trade's (whether it be a purchase or sale), as,

$$c_t = \boldsymbol{x}^T \boldsymbol{k}_{\mathrm{t}} \times 0.1\% \tag{3}$$

- Market collapse risk aversion: Market crashes can occur quickly as a result of occurrences like wars, the explosion of a stock market bubble, the default of a sovereign government, and financial crises. Which monitors sudden fluctuations in asset values, to lower the risk in the worst-case scenario, such as a worldwide financial crisis:

$$turbulence\ _t = (y_t - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(y_t - \boldsymbol{\mu})' \in \mathbb{R} \tag{4}$$

$y_t \in \mathbb{R}^D$ stands for the stock returns for the current period t, $\mu \in \mathbb{R}^D$ for the average historical return, and $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ for the covariance of historical returns.

We simply stop purchasing, and the algorithmic trading agency immediately sells all shares, when market circumstances are extremely choppy and turbulence t is over a threshold. Once the turbulence index drops below the threshold, we start trading again.

4.3. Discussion

4.3.1. Return maximising as a trading objective

Objective create algorithmic trading plan maximizes value

$$r(s_t, a_t, s_t) = (b_t + \boldsymbol{x}_{t+1}^T \boldsymbol{z}_t) - (b_t + \boldsymbol{x}_t^T \boldsymbol{z}_t) - c_t \qquad (5)$$

Signify the portfolio value at time t. h_t follows in order to further breakdown the return:

- $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \boldsymbol{k}_t^S + \boldsymbol{k}_t^B$ (6)

The balance's transition, b_t, is specified.

Now, $r(s_t, a_t, s_{t+1}) = (b_{t+1} + \boldsymbol{x}_{t+1}^T \boldsymbol{z}_{t+1}) - (b_t + \boldsymbol{x}_t^T \boldsymbol{z}_t) - c_t$ also written as

- $r(s_t, a_t, s_{t+1}) = r_H - r_s + r_B - c_t$ (7)

  where

- $r_H = (\boldsymbol{x}_{t+1}^H - \boldsymbol{x}_t^H)^T \boldsymbol{z}_t^H,$ (8)

  $r_S = (\boldsymbol{x}_{t+1}^S - \boldsymbol{x}_t^S)^T \boldsymbol{z}_t^S,$ (9)

  $r_B = (\boldsymbol{x}_{t+1}^B - \boldsymbol{x}_t^B)^T \boldsymbol{z}_t^B,$ (10)

  $r(s_t, a_t, s_{t+1}) = r_H - r_s + r_B - c_t$ shows that we should purchase.

  In order to maximize the positive change in sell the stocks whose prices will decline at that moment in order to reduce the negative change.

$r_S = (\boldsymbol{x}_{t+1}^S - \boldsymbol{x}_t^S)^T \boldsymbol{z}_t^S$ becomes

- $r_{\text{sell}} = (\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)^T \boldsymbol{k}_t$ (11)
- It means that since stock prices will all decline, we want to sell all of the stocks we own in order to minimise the negative impact to the portfolio's worth.

As seen below, the model is initialised. The stock values at time 0 are specified as p_0, and the beginning fund amount is b_0. For any state, all actions have a uniform distribution of (s), and the x and $Q_\pi(s, a)$ are both 0.

According to course of action is one in which the expected reward of acting in state s_t equals sum immediate reward $r(s_t, a_t, s_{t+1})$ in state s_{t+1}.

For sake of convergence, let's discount the future benefits by a factor of $0 < \gamma < 1$.

4.3.2. To determine the efficiency of algorithmic trading

Algorithmic trading certainly must have performance measurement. Without performance evaluation and meticulous record keeping, it is difficult, if not impossible, to tell if the returns from our approach have been the result of chance or of any real competitive advantage. As a result, the total backtesting performance of algorithmic trading utilising the long-short strategy and the Quantopian risk model approach is as follows.

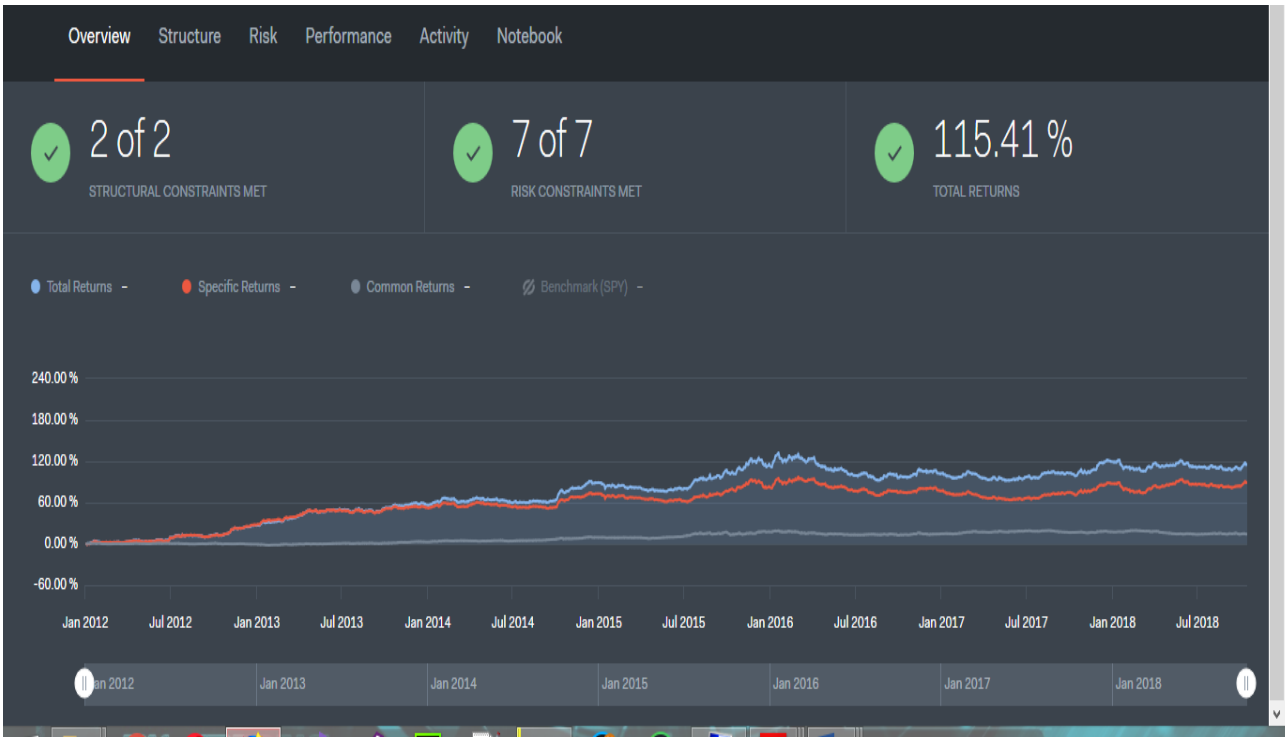


Figure 4.1. Algorithmic trading performance overview

The algorithm is the most effective when compared to the other two algorithmic trading from which algorithmic trading is derived in terms of overall performance utilising Quantopian benchmarks including structural constraints, risk restrictions, and total returns. Figure 1 depicts the overview of the performance of algorithmic trading.

The entire backtesting results shown in figure 4.2 of the algorithm based on the long-short and Quantopian risk model are shown in the table below.

Table 4.1. Algorithmic trading backtesting overview.

	Backtest
Annual return	11.9%
Cumulative returns	116.9%
Annual volatility	9.9%
Sharpe ratio	0.9
Calmar ratio	0.69
Stability	1.0
Max drawdown	-17.3%
Omega ratio	0.95
Sortino ratio	1.91
Skew	0.33
Kurtosis	3.66
Tail ratio	1.12
Daily value at risk	-1.2%
Gross leverage	1.00
Daily turnover	11.9%
Alpha	0.13
Beta	-0.08

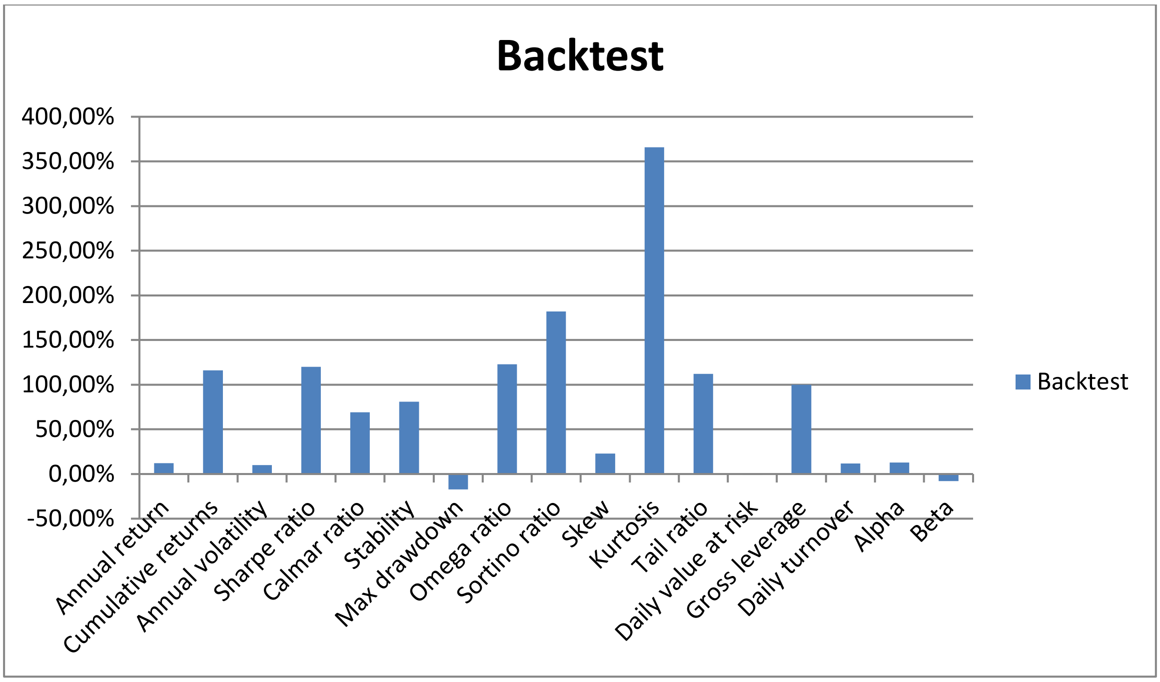


Figure 4.2. Algorithmic trading backtesting overview.

Below is a table showing the top 5 drawdown periods of algorithmic trading.

Table 4.2. The top 5 drawdown periods of the algorithmic trading

Worst drawdown periods	Net drawdown in %	Duration
0	17.44	NaN
1	7.94	179
2	6.16	21
3	5.08	107
4	4.21	48

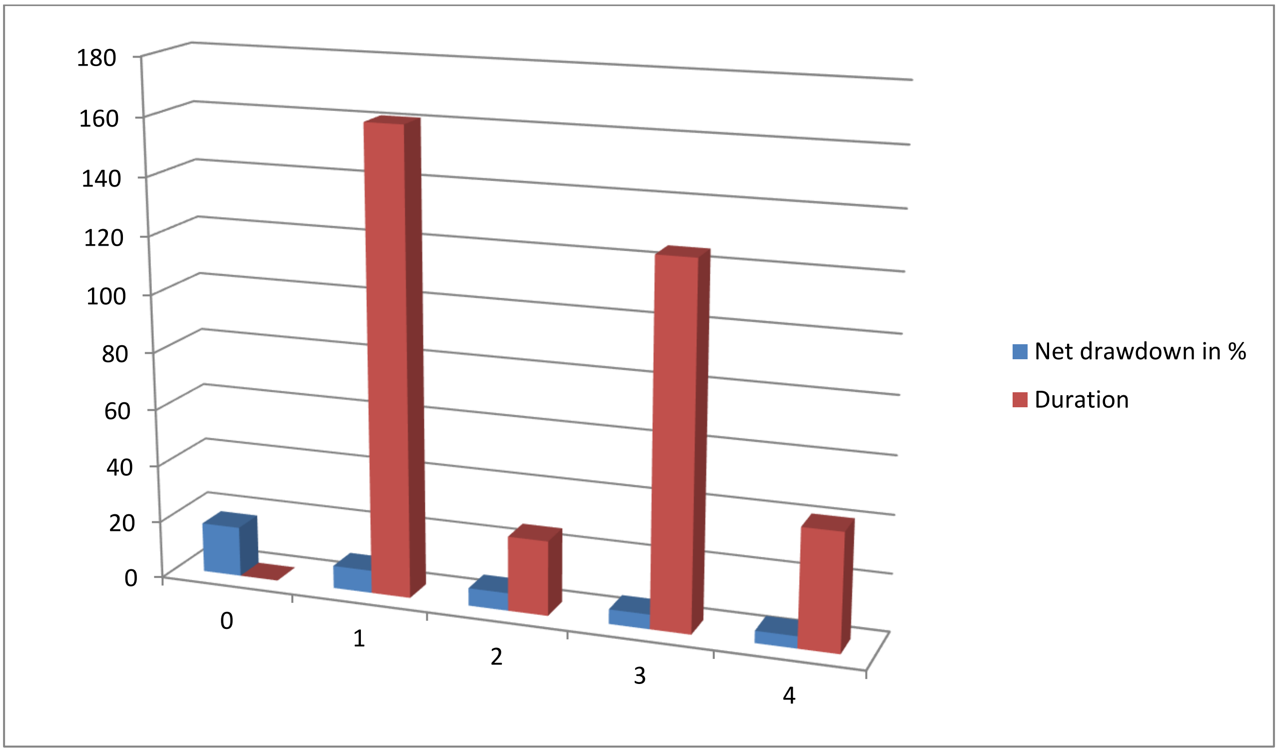


Figure 4.3. The top 5 drawdown periods of the algorithmic trading

Figure 4.4. Quantopian risk model algorithmic trading's overall performance

Although this algorithmic trading met two structural limitations and produced total profits of 63.82%, it was unable to satisfy all risk constraints. The topic may be condensed if we make the supposition that limit orders are always at the front of the line, which is not too difficult to do.

Figure 4.5. Long-short strategy algorithmic trading's overall performance

Although the long-short algorithmic trading (shown in figure 4.5) satisfied all seven risk restrictions and two structural constraints, it had returns of -31.63%.

Relevant to my discussion are three major recent advancements in algorithmic trading including AI. First, there is a push to automate quantitative trading, which involves employing computers for both placing orders and formulating strategies, similar to how HFT is currently automated.

Summary data are typically used to assess an efficient algorithmic trading. These particular outcomes of algorithmic trading are displayed in the table below.

Table 4.3 Summary Statistics

Summary Statistics	
Specific Annualised Return	10.0%
Common Annual Return	2.01%
Total Annualised Return	12.04%
Particular Sharpe Ratio	1.10

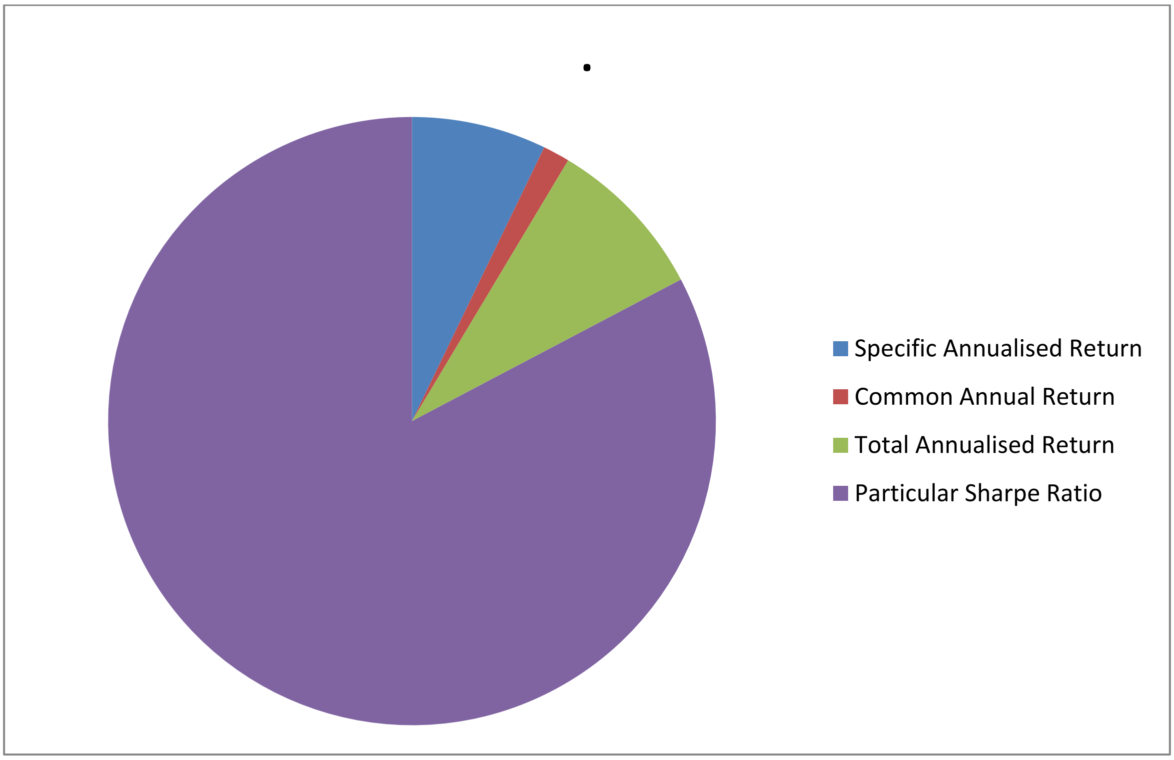


Figure 4.6. Summary Statistics

All of the backtesting period's monthly returns are represented using a heat map, along with the yearly returns on a bar chart and the distribution of monthly returns on a column chart.

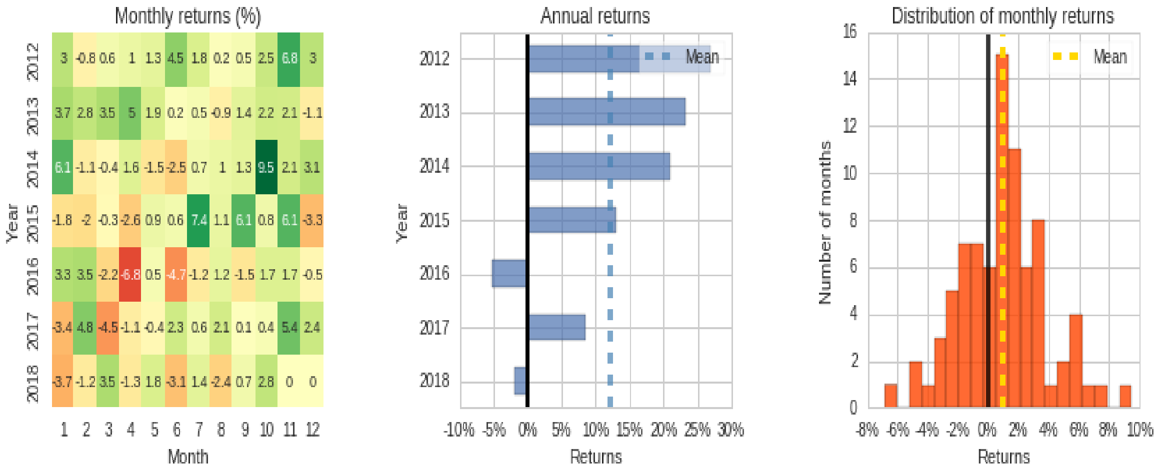


Figure 4.7. Monthly and annual returns

4.4. To create a successful algorithmic trading employing a combination of tactics

4.4.1. Analysis of strategy and portfolio

4.4.1.1. Return Quantile

The easiest method to illustrate the statistical word "quantile" is to provide a specific example of a quantile, such as the median value of a price series. In a straightforward example, if a data series contains 100 closing prices and we order them from lowest to highest, the precise midpoint of the ranking (i.e., the 50th closing price) will be the median value. When using a quantile technique, as opposed to a moving average, pricing data must be continuously ranked, and the value that is on the edge of the percentile or quantile that one wants to focus on must then be chosen.

A daily, monthly, and yearly return quantile for algorithmic trading displayed in figure 4.8 below.

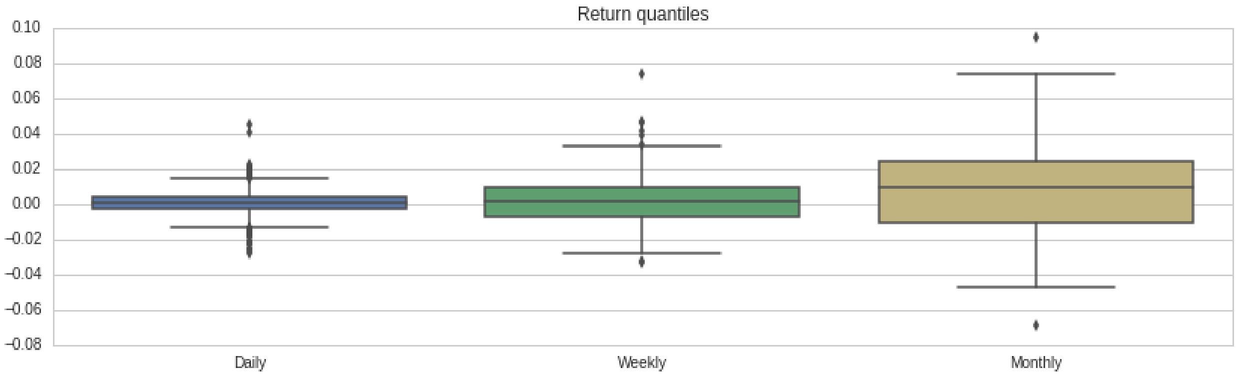


Figure 4.8. Return Quantile

The components of a person's or an organization's investment portfolio, are referred to as holdings. As well as more specialised instruments like private equity and hedge funds, can all be included in a portfolio's holdings. The trading holdings are shown below.

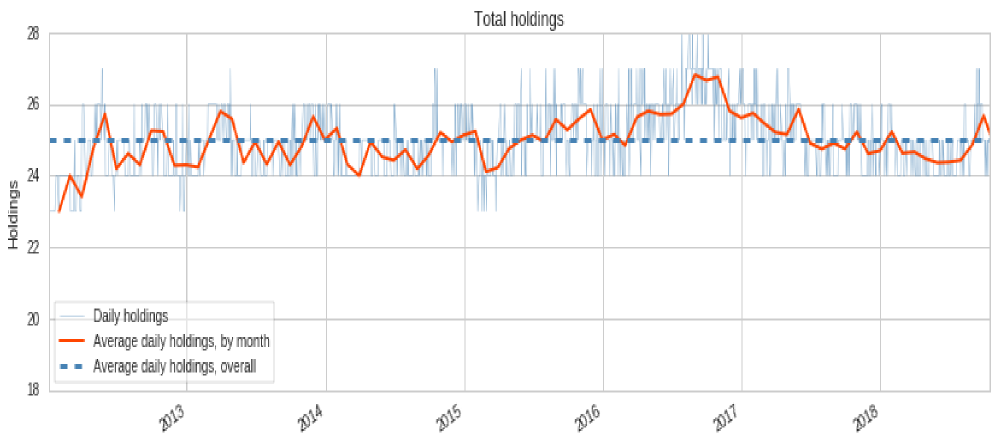


Figure 4.9. Total Holdings

Exposure of show that the algorithmic trading has a balanced long-short strategy which is in the algorithmic trading enabling it to effectively invest in the long investments and short investments at a given maximum percentage.

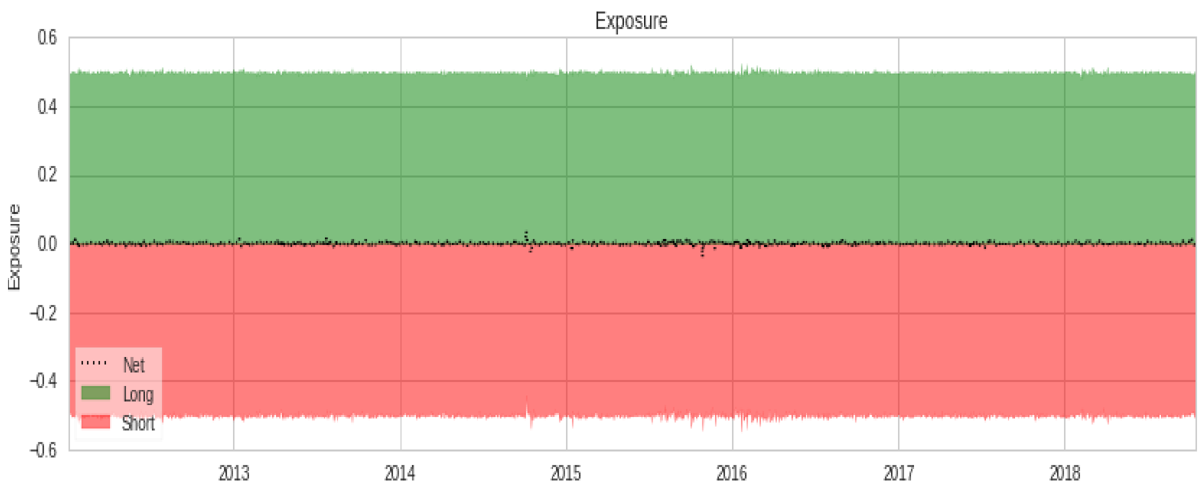


Figure 4.10. Exposure

In order to reduce risks and increase earnings, Quantopian risk model design is used in algorithmic trading. The graph below displays the capital allocations throughout a portfolio.

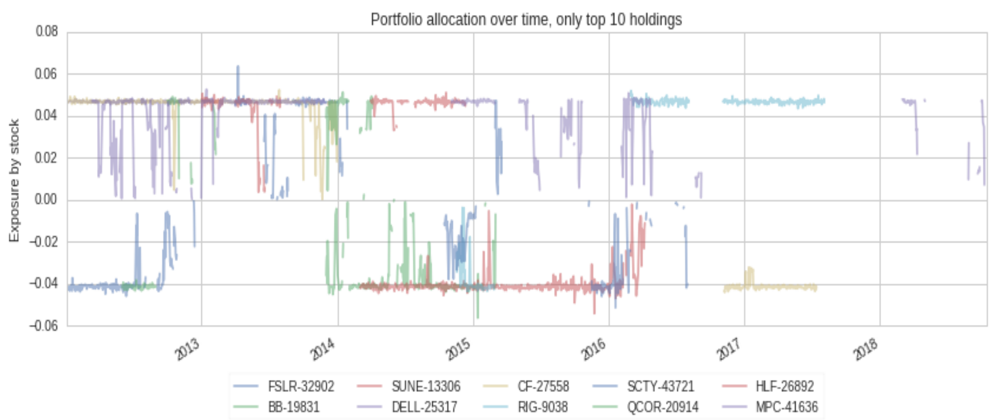


Figure 4.11. Portfolio allocation over time

The graph below (figure 4.12) illustrates all leverage across the backtesting period and the lowest and maximum capital investments made in each transaction in algorithmic trading.

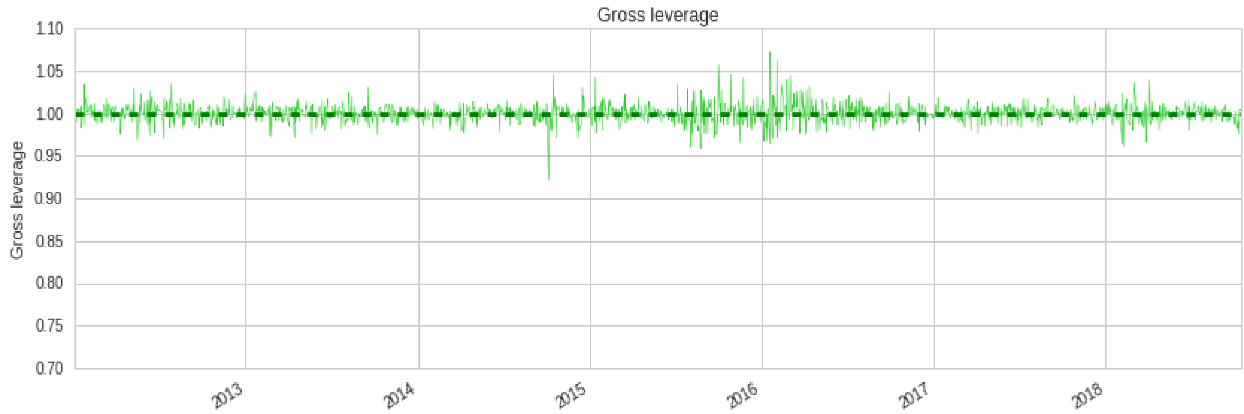


Figure 4.12. Gross leverage

4.5. To determine the performance of algorithmic trading

4.5.1. Risk/Reward analysis

In order to evaluate the risk/reward of algorithmic trading, the researcher employed the Sharpe ratio. Additionally, common risk characteristics have been provided by algorithmic trading. The tables below illustrate all of the risk factor exposures:

Table 4.4 Exposure summary

Exposures Summary	Average Risk Factor Exposure	Annualized Return	Cumulative Return
basic_materials	-0.02	-1.31%	-7.97%
consumer_cyclical	0.03	0.75%	5.27%
financial_services	0.07	1.31%	7.21%
real_estate	-0.02	-0.17%	-1.71%
consumer_defensive	0.02	0.20%	1.64%
health_care	0.01	0.21%	1.37%
utilities	-0.01	0.02%	0.10%
communication_services	-0.01	0.03%	0.17%
energy	-0.11	-1.38%	-9.2%
industrials	0.01	-0.06%	-0.36%
technology	0.02	-0.04%	-0.09%
momentum	-0.00	0.08%	0.22%
size	0.17	-0.09%	-0.43%
value	0.03	0.04%	0.52%
short_term_reversal	0.01	0.02%	0.58%

volatility	-0.27	2.14%	16.77%

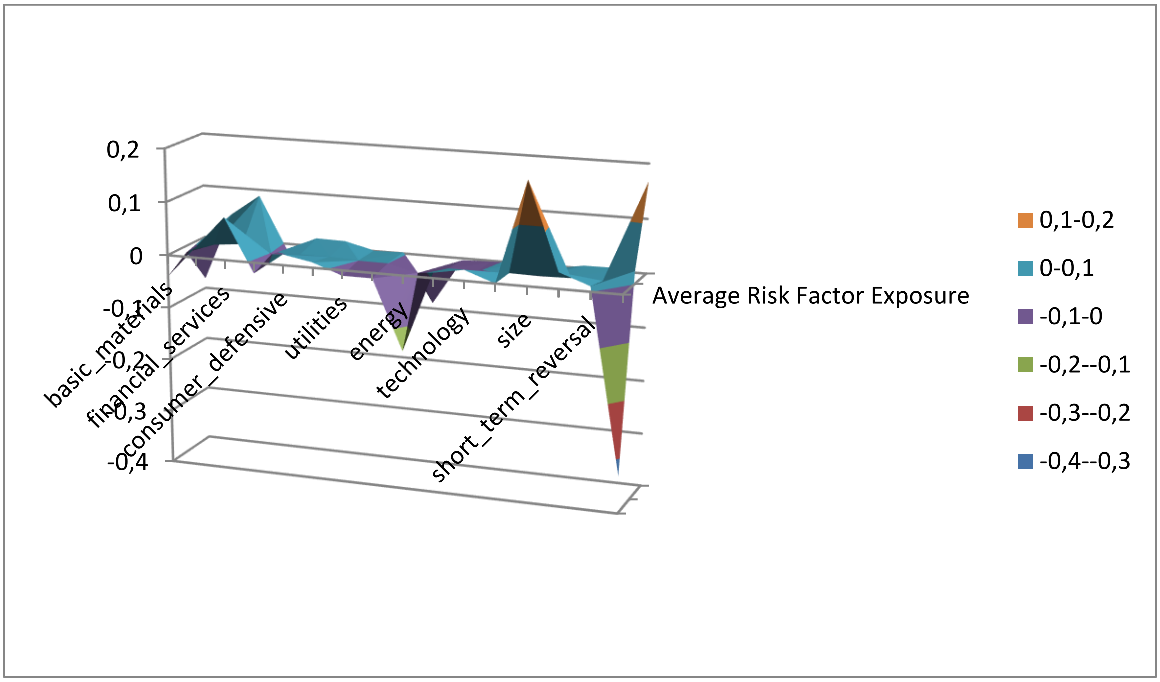


Figure 4.13. Exposure summary

Below is a time series graph of the cumulative returns from the previous tables. The trading activity for algorithmic trading throughout the backtesting period is displayed.

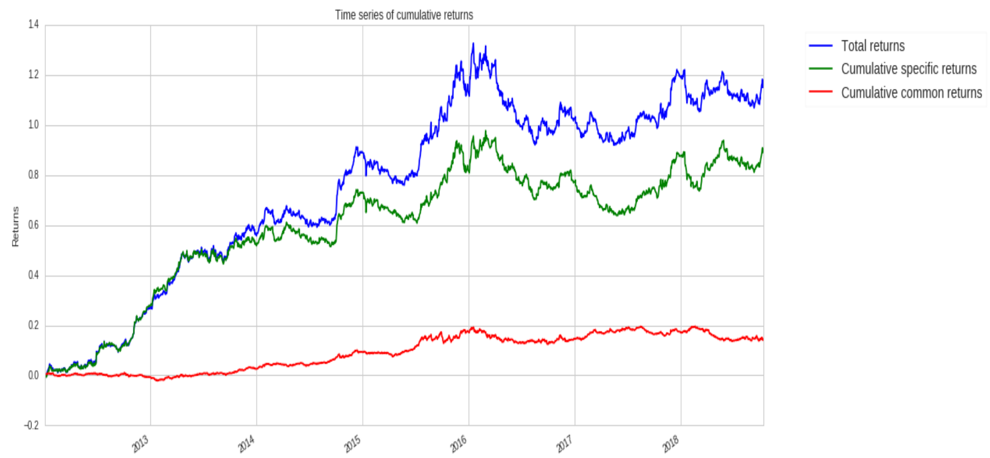


Figure 4.14. Time series

The cumulative common sector returns are as follows showing assets traded of companies in each and every industry.

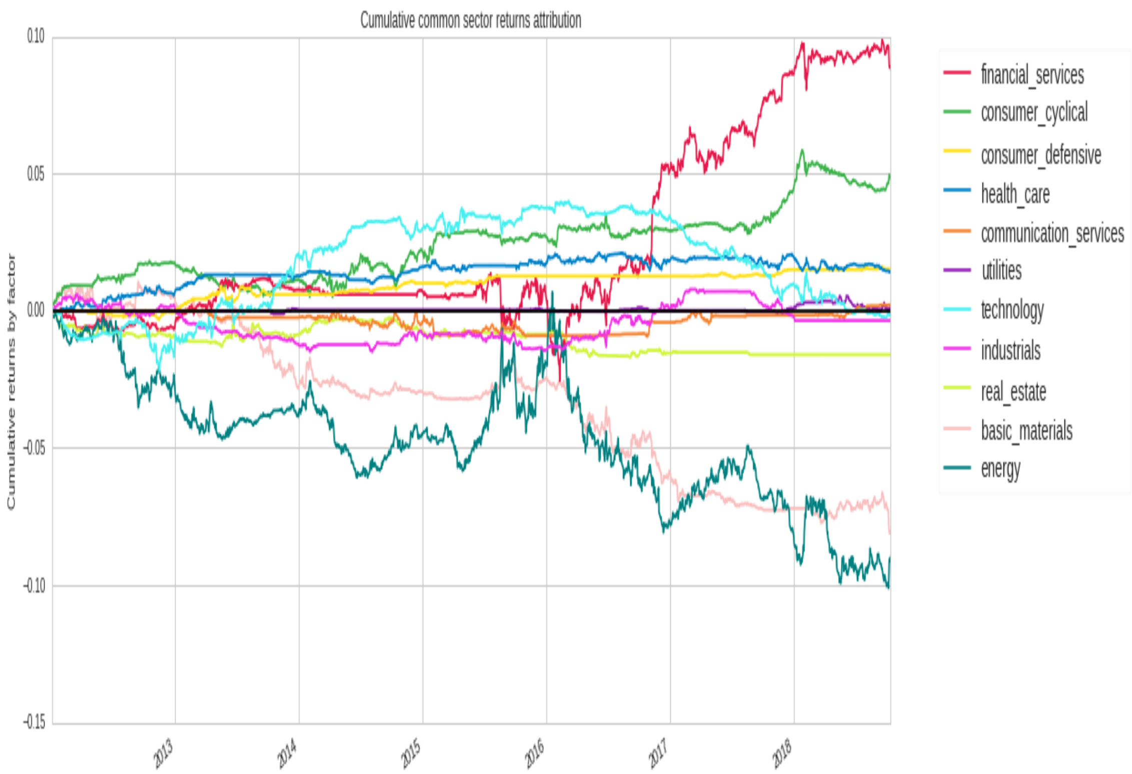


Figure 4.15. Cumulative common sector returns

4.6. Returns analysis

As a result, the algorithmic trading has given positive total returns during the backtesting period. Below is a cumulative returns line graph of algorithmic trading:

Figure 4.16. Cumulative returns

4.7. Drawdown analysis

Figure 4.17 below is a line graph showing top 5 drawdown periods in all shaded areas which shows the performance of the algorithmic trading over the backtesting period.

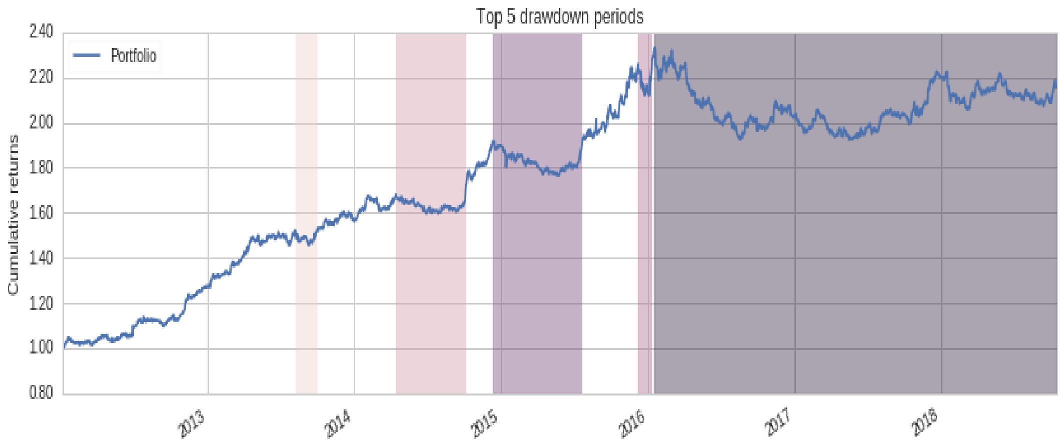


Figure 4.17. Drawdowns

An underwater plot showing all values below zero of drawdown percentages within the backtesting period:

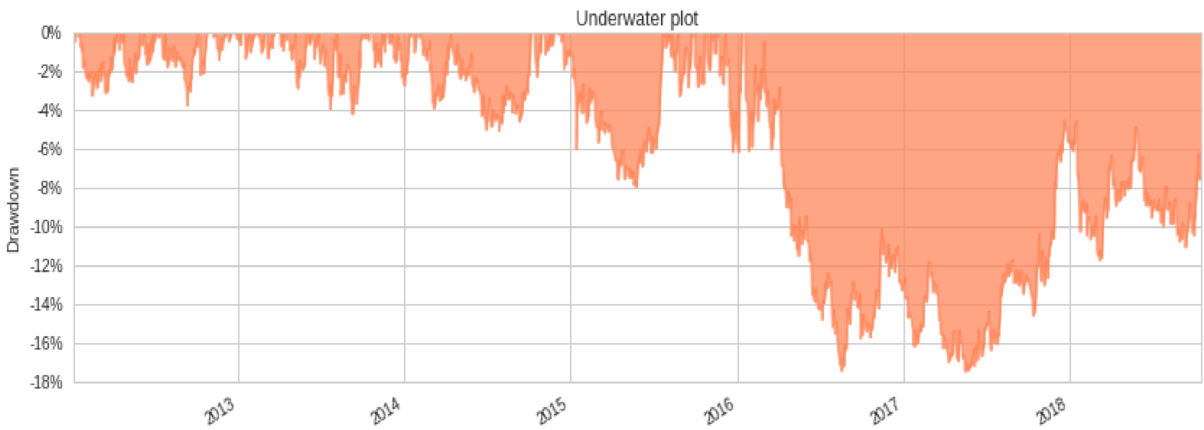


Figure 4.18. Underwater plots for drawdowns

When analyzing drawdowns from a backtesting result, there are events which are checked and these events are historical event dates that may have had significant impact on markets.

4.8. Research Questions

The research questions for this study:

1. How determine the efficiency of the algorithm by applying known algorithmic trading strategies?
2. How to design an effective trading algorithm using known model?
3. How test and implement an algorithm on a algorithmic trading platform to measure its overall performance?

4.9. Conclusion

A commodity is moved between people via algorithmic trading, which dates back to ancient times, based on the supply and demand principle. A demand-driven fungible asset is traded for a supply of a commodity. The algorithmic trading of stock shares for money on the stock market foreshadows that. After the introduction of online algorithmic trading, AT has expanded due to the rise of new technology initiatives. Perhaps it comes as expected that algorithmic trading has long been a key component of AI breakthroughs, including different types applied there. It should come as expected that further sidesteps human cognition if markets have been designed to not value human cognition and to assume that the truth lies in the information processor that is the market itself.

This chapter summaries and analyses the data on algorithmic trading's efficiency in automating online algorithmic trading from the study methodologies used. The success of algorithmic trading using Quantopian benchmarks is demonstrated in this chapter utilizing line graphs, tables, column charts, bar charts, and other methods of visualization. Using the returns, trading universe, portfolio holdings, and other criteria described in this research thesis' chapter, the efficacy of algorithmic trading is assessed.

CHAPTER V:

DISCUSSION

5.1. Introduction

In algorithmic trading, creating a lucrative trading strategy is essential since the algorithm may control and carry out automatic trading choices. A key research difficulty in algorithmic trading on the financial markets is determining a precise algorithmic trading rule at a certain moment. But a sophisticated, dynamic algorithmic trading system based on the most recent trends in a certain price time-series could be able to resolve this problem. By using the price time-series as its environment, Reinforcement Learning (RL) achieves the greatest dynamic algorithmic trading. To suggest a dynamic algorithmic trading strategy that makes use of RL, a thorough description of the environmental circumstances is necessary.

For the purpose of creating algorithmic trading strategies for continuous futures contracts, we use Deep Reinforcement Learning algorithms. When constructing reward functions that scale algorithmic trading positions in keeping with market volatility, both discrete and continuous action spaces are taken into account. We evaluate the performance of our algorithms and compare it to those of other asset classes such as commodities, stock, demonstrate that despite having large transaction costs, our algorithms continue to generate profits and beat traditional time series momentum techniques. The results of the studies demonstrate that the suggested algorithms may scale down or hold during times of consolidation while still tracking broad market trends and remaining steady.

5.2.Summary of the Dataset

Asset categories in our dataset, which covers the years 2005 through 2019, also contain foreign currencies. Every five years, we update our model and optimise the parameters based on the most recent information. The following five years saw the establishment of model parameters in order to get outcomes not included in the sample.

5.2.1. Standard Algorithms

We compare our methods to the following core models, which make use of conventional methods for calculating time-series momentum:

Only Long and Sign(R)

$A_t = \text{sign}\ (r_{t-252:t})$

When MACD_t is used, as it is in Equation 3, the signal is MACD. Additionally, we may average a number of signals with various time periods to provide a final indicator:

$$\widehat{\text{MACD}}_t = \sum_k \text{MACD}_t(S_k, L_k)$$

where S_k and L_k define short and long time scales, namely $S_k \in \{8,16,32\}$ and $L_k \in \{24,48,96\}$.

With our RL algorithms, DQN, PG, and A2C, we contrast the aforementioned baseline models with each other. A2C has a continuous action space [-1,1], whereas DQN and PG have discrete action spaces [-1,0,1]. Table 1 lists the hyperparameter values for several RL algorithms.

Table 5.1. Hyperparameter values for several RL algorithms

γ	Bp	Memory Size	τ
0.3	0.0020	5,000	1,000
0.3	0.0020	-	-
0.3	0.0020	-	-

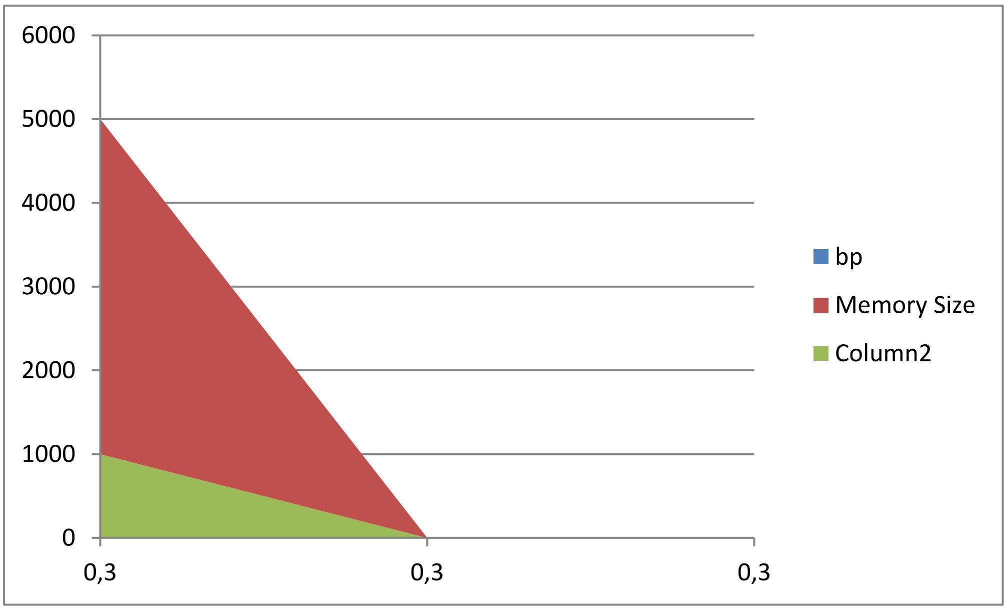


Figure 5.1. Hyperparameter values for several RL algorithms

5.3.Controlling Overfitting Procedures

The following techniques are used to prevent overfitting from occurring while backtesting data since it can be problematic in a variety of application areas, notably finance. The most frequent cause of overfitting is a poor ratio of training samples to model parameters. Even though there is no guarantee that extra samples would solve the overfitting problem, in our case, we find that expanding the dataset to allow training on a variety of futures contracts rather than just one contract significantly helps to minimise the problem. Additionally, if feasible, purposefully reduce amount of free parameters in our networks. Each of our models contains two hidden layers with a modest number of neurons in each layer. As a result, the network is less prone to overfit since it contains fewer parameters.

A regularisation technique that is frequently used in deep learning. Because of the size of our networks, we did not observe any appreciable gains, but we nevertheless advise using dropout in accordance with the usual recommendation to use the least flexible model that produces equivalent cross-validation results. In order to tune the hyperparameters, we lastly employ 10% of each training set in a different cross-validation set. We utilise early stopping with 20 epochs and track validation performance to make it easier for users to select the best model. The cross-validation set is used for any hyperparameter optimisation, leaving the test set alone for the analysis of the test results. Test data leakage, another aspect of overfitting, is thereby diminished.

5.4.ExperimentalResults

We test both our methods and the baseline models for each contract. The next step is to give each contract an equal weight in order to determine the algorithmic trading return of the portfolio.

$$R_t^{\text{port}} = \frac{1}{p}\sum_{i=1}^{N} R_t^i$$

R_t^i is the algorithmic trade return for contract i at time t, and p is number contracts that were taken into consideration. We assess this portfolio's performance using the following parameters.Annualised estimated algorithmic trade returns are

E(R).Standard deviation of algorithmic trading returns, The annualised standard deviation of algorithmic trade returns that are negative is referred to as annualised downside risk (ADR), also known as downside deviation (DD).

Sharpe: (E(R)/std(R)) annualised Sharpe ratioSortino: a Sharpe ratio version that evaluates risk through downside deviation (E(R)/DD) largest Drawdown (MDD) displays the largest loss that may be seen from any portfolio peak to trough.

Calmar: The predicted annual rate of return is compared to MDD using the Calmar ratio; generally speaking, the larger the ratio, the better the performance of the portfolio.

%+ve Returns: percentage of positive algorithmic trade returns and

$\frac{Ave.P}{Ave.N}$: the ratio between positive and negative algorithmic trade returns.

In table 5.2, where a further implemented, show our findings. This equalises the volatility of various approaches so that we can compare measures like predicted and cumulative algorithmic trade returns directly. Using various asset classifications as a basis, Table 2 is divided into five sections. The outcomes display a portfolio's performance utilising solely contracts from that particular asset type. The last section of the show, when we create a portfolio utilising every contract in our information, stands out as an outlier. Table 3 displays the cumulative algorithmic trading results for various models and asset types. Across the majority of asset classes, with the exception of the equities index, which benefits more from a long-only approach? This can be explained by the fact that significant rising trends dominated the majority of equity indices during the testing period. In a similar vein, fixed incomes saw an increasing tendency up to 2016 before entering the present consolidation phases.

Table 5.2. Portfolio-Level Volatility Targeting Experiment Results

	$E(R)$	Std (**R**)	DD	Sharpe	Sortino	MDD	Calmar	% of + Ret	Ave. Ae.
Commodity									
Long	-0.710	0.979	0.604	-0.726	-1.177	0.350	-0.140	0.473	0.989
Sign (R)	0.347	0.980	0.572	0.354	0.606	0.116	0.119	0.494	1.084

MACD	-0.171	0.978	0.584	-0.175	-0.293	0.190	-0.060	0.486	1.026
DQN	0.703	0.973	0.552	0.723	1.275	0.066	0.501	0.498	1.135
PG	0.062	0.982	0.585	0.063	0.106	0.039	0.023	0.495	1.029
A2C	0.223	0.955	0.559	0.234	0.399	0.141	0.091	0.487	1.093

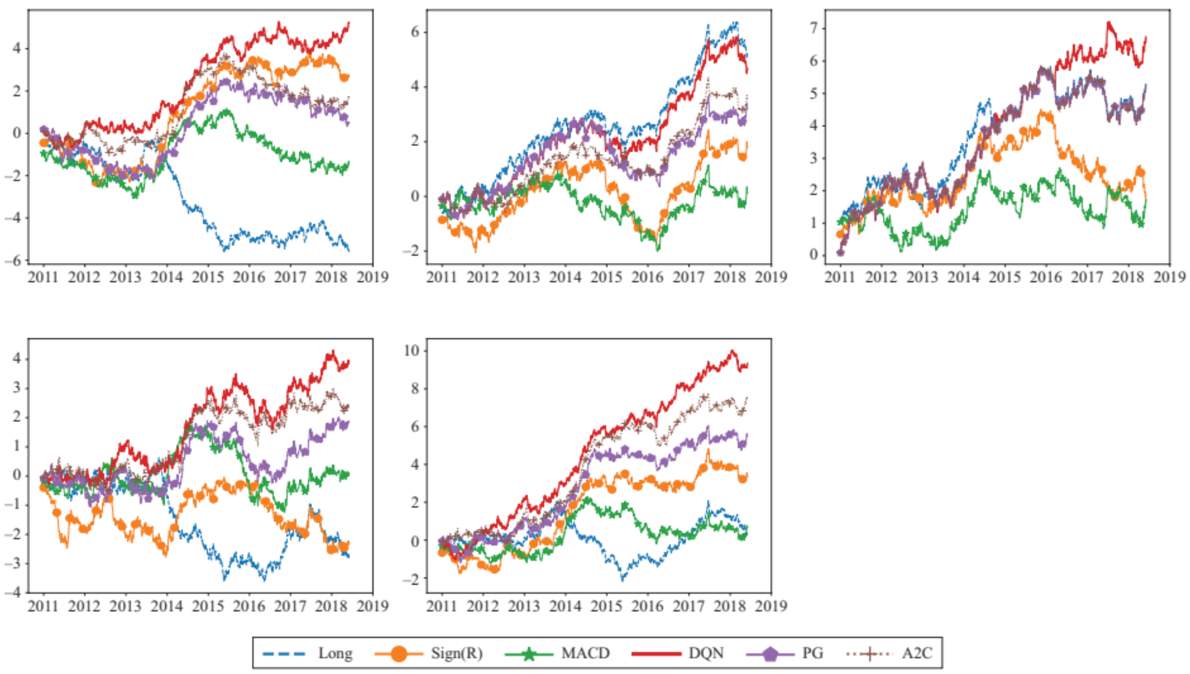


Figure. 5.2. Cumulative Algorithmic trade Return

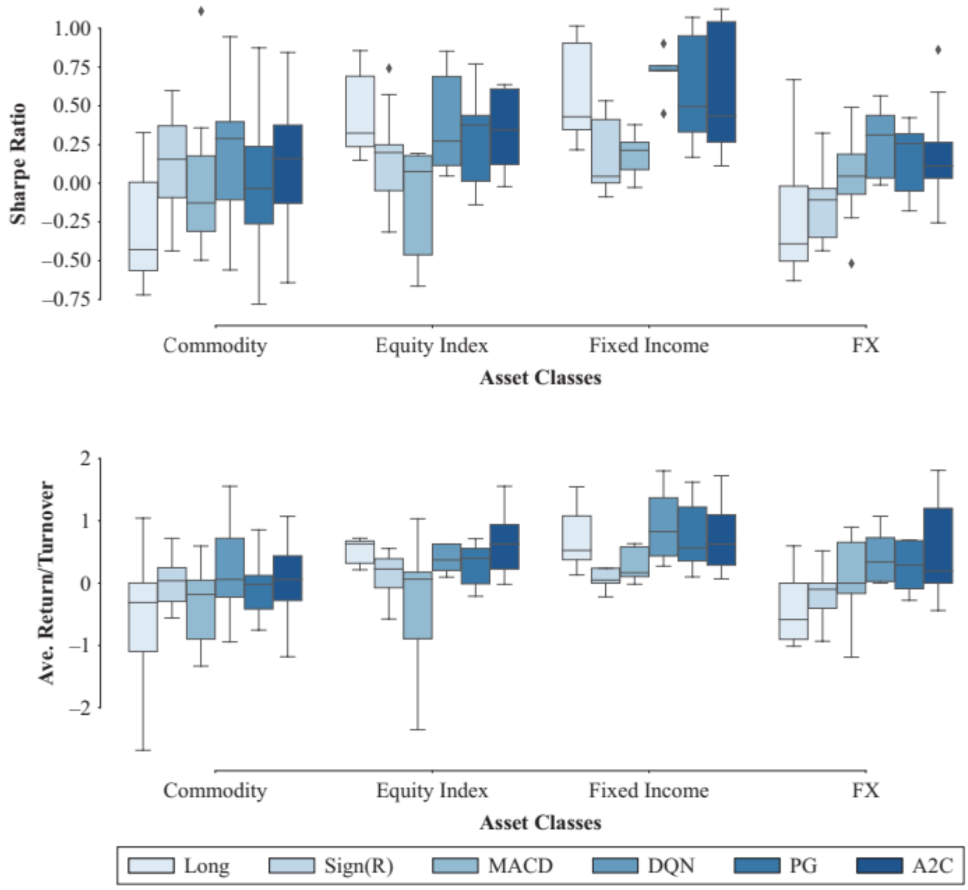


Figure 5.3. Average Algorithmic trade Return per Turnover (bottom) and Sharpe Ratio (top) for Individual Contracts

If significant trends continue, we could choose to keep our holdings in certain situations. However, the more volatile commodities. RL algorithms outperform other algorithms in some markets because they can algorithmic trade at the right times. Outperforms all other models, with the A2C strategy coming in second. Table 4 illustrates this point. When we look into the facts around this conclusion, we find that A2C produces more turnovers, which lower average returns per turnover. We also evaluate how effectively our solutions function in the presence of various transaction costs. In Panel A of Table 5, we show the annualised Sharpe ratio for the portfolio using all contracts at different cost rates.

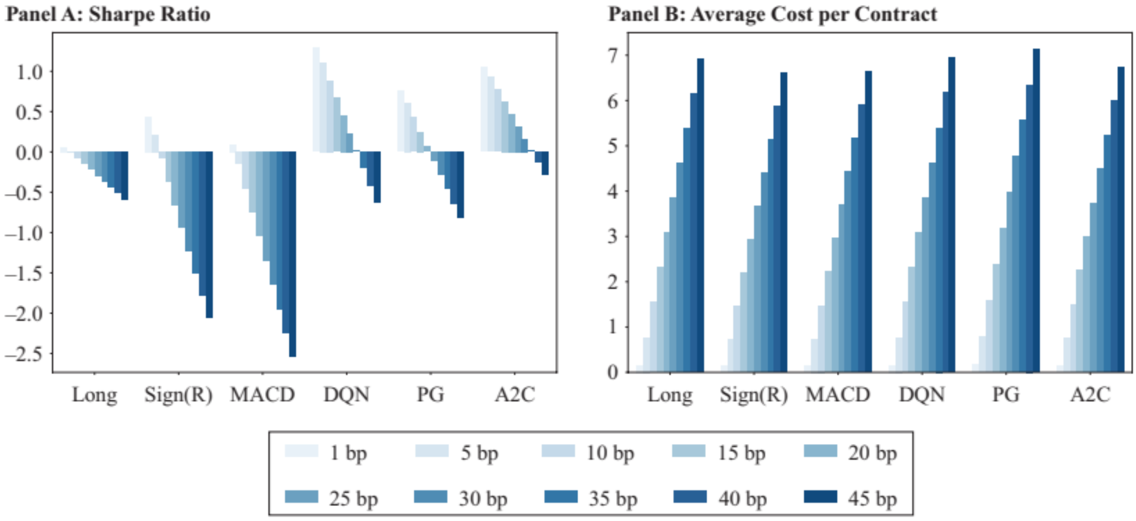


Figure 5.4. Sharpe Ratio and the Average Cost per Contract at Various Cost Rates.

We assess each individual deal's performance. We display algorithmic trading using boxplots in Table 4. We are confident in the consistency of our model since the success of our method is not dependent on any one contract that performs better than others. Overall, these findings confirm our past findings that RL algorithms often outperform other types of algorithms.

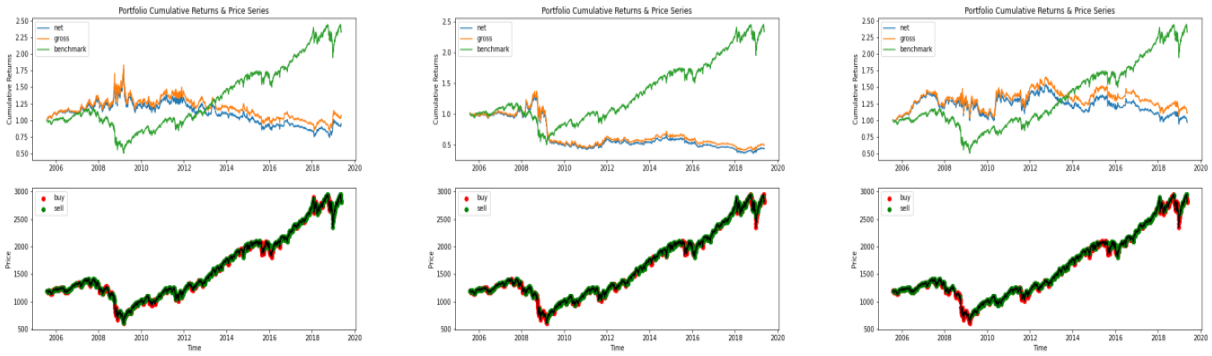
5.5.Naive tactics

As training data, we first construct the models utilising 1 out of every 90 routes from a simulated process. Lacks trustworthy directional information, so both the model's and the benchmark's expected returns should be close to zero because there is no trend signal for algorithmic trading in this regime. Third, the model appears to have significantly outperformed the benchmark for the down-trend regime. This statistical pattern might be explained by the fact that whereas the benchmark prohibits short positioning, our model does.

Figure 5.5 shows the naive strategy's out-of-sample cumulative returns over the first five years.

5.6.Advanced techniques

We train our agent using 25 liquid, multi-asset futures, and we label the resultant strategy as advanced. Table 5 displays the pertinent performance data. None of the complicated strategies can outperform long-only strategies important because, recession, demonstrating that they provide consistent downside protection. In this case, holding a long position could be the wisest course of action because the true data shows a significant upward trend since 2008 [2]. But it remains a mystery as to why the RL agent in the expanding market was unable to adjust. Would have prevented 2008 recession while also resulting in a very alluring overall out-performance. Last but not least, we notice that for all naïve approaches, the ratio was unable in coming up with a downside-protection plan that retained any upside.



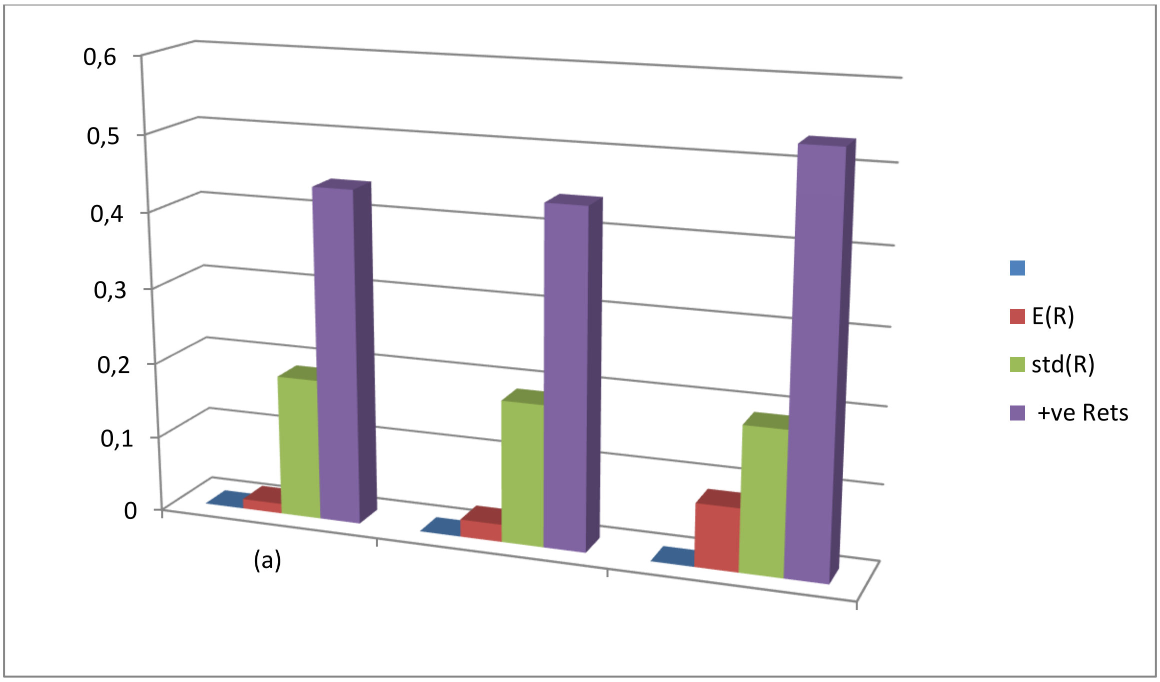
(a) every quarter (b) every half-year (c) every year

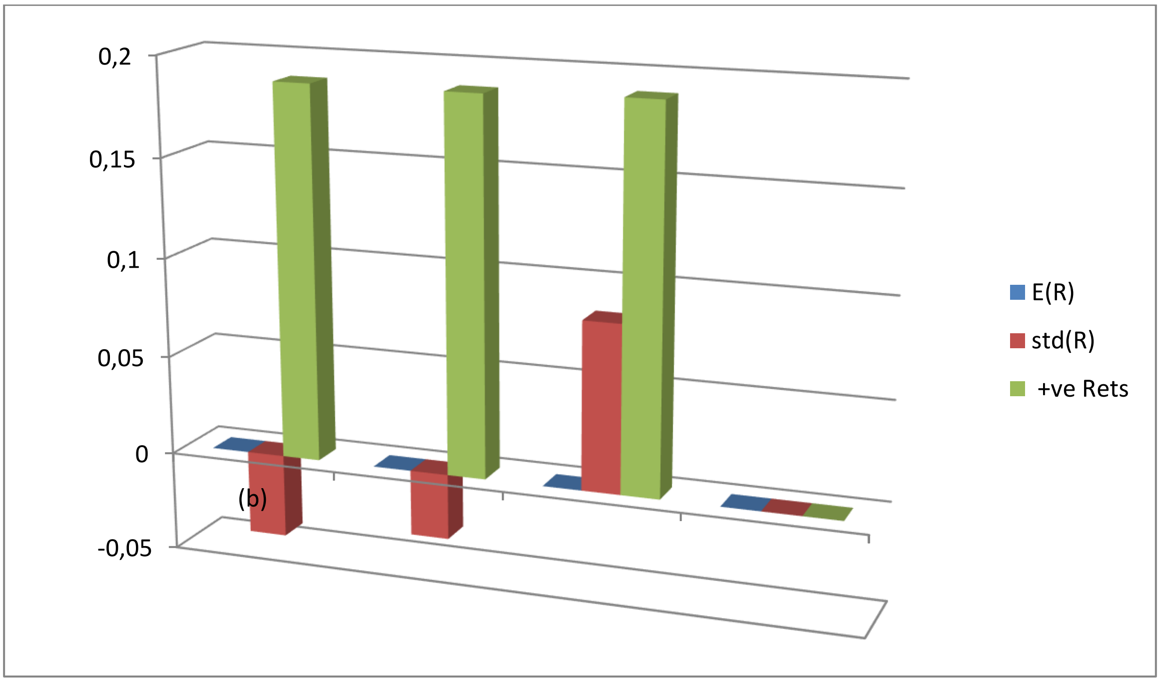
Figure 5.6. Cumulative returns for advanced approaches using 25 futures as training data, updated rollingly on a quarterly (1st), semi-annual (2nd), and annual (3rd) basis.

Table 5.3. The outcomes of the advanced strategy experiments.

MODEL		E(R)	std(R)	% +ve Rets	Ave. Av / Ave
(a)	net	0.013	0.187	0.440	0.787
	gross	0.023	0.187	0.442	0.792
	benchmark	0.083	0.189	0.533	1.142

(b)	net	-0.042	0.188	0.435	0.770
	gross	-0.033	0.188	0.436	0.772
	benchmark	0.084	0.190	0.537	1.161
(c)	net	0.014	0.178	0.401	0.669
	gross	0.024	0.178	0.402	0.674
	benchmark	0.084	0.191	0.540	1.172


0,6
0,5
0,4
0,3
0,2
0,1
0
(a)
E(R)
std(R)
+ve Rets


0,2
0,15
0,1
0,05
0
-0,05
(b)
E(R)
std(R)
+ve Rets

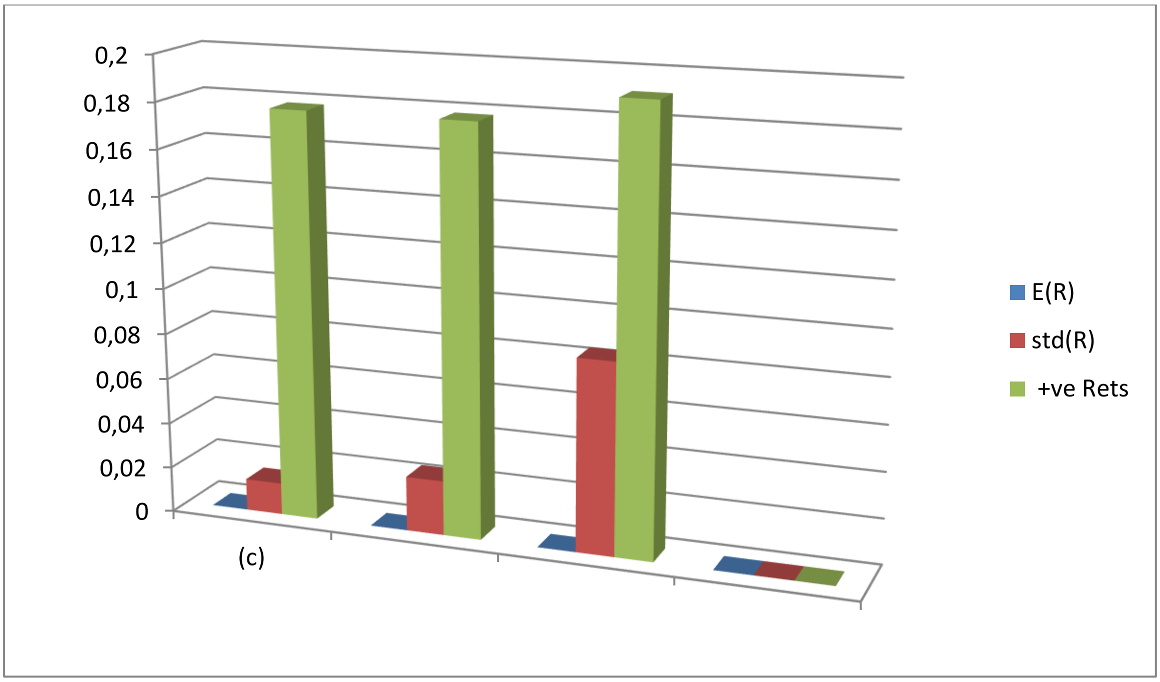


Figure 5.7. Outcomes of the advanced strategy experiments

We believe that a variety of variables, including the following, contribute to the RL agent's subpar performance for advanced techniques: First off, the training data amount is still somewhat little in proportion to the model's parameter number, even with 25 more futures contracts. The problem is theoretically greater here because there is less data than there was with the intentionally faked data. The collection of 25 multi-asset futures contracts doesn't seem to have kept the fundamental return characteristics of the underlying market, but it's possible that they were preserved in the learning data created by the VG process. With regard to the other markets, asset classes, and their inherent correlation dependencies, the RL agent is noticeably unable to make any more significant conclusions.

Again, the dimensionality of the RL state space is probably too low for the agent to learn a algorithmic trading strategy that is very flexible and suitable for a wide range of volatility regimes. Applying "more of the same" in this situation, as demonstrated by the VG training data, seems to be the best course of action. Getting the RL agent to utilise the extra data that the 25 multi-asset futures contracts clearly supply will now be the primary challenge. Third, further research is needed to understand why assessment frequency plays such a significant role in averting the 2008 catastrophe, with semi-annually being the horrible exception. This seems weird in several ways. Fourth, the one strategy that has worked

5.7. Conclusion

Again, the dimensionality of the RL state space is probably too low for the agent to learn a algorithmic trading strategy that is very flexible and suitable for a wide range of volatility regimes. Applying "more of the same" in this situation, as demonstrated by the VG training data, seems to be the best course of action. Getting the RL agent to utilise the extra data that the 25 multi-asset futures contracts clearly supply will now be the primary challenge. Third, further research is needed to understand why assessment frequency plays such a significant role in averting the 2008 catastrophe, with semi-annually being the horrible exception. This seems weird in several ways. Fourth, the one approach that has succeeded. The goal is to maximise a risk-adjusted performance function, such as the Sharpe ratio, because investors are usually risk averse. In reality, this produces a concave utility function. To optimise the Sharpe ratio after learning the distribution, we can choose actions that have the greatest expected Q-value and the smallest standard deviation from it. We extend our approaches for portfolio optimisation by changing the action spaces to incorporate weights for certain contracts in a portfolio. To produce a good predicted algorithmic trade return with low volatility, the mean-variance portfolio theory and reward functions may be applied.

CHAPTER VI:

SUMMARY, IMPLICATIONS AND RECOMMENDATIONS

6.1 Introduction

This chapter summaries the whole project and major findings of the research, highlights the conclusions arrived at based on the findings and gives recommendations on the way forward. This chapter shows all chapters' summary giving a comprehensive meaning of the research findings and their significance.

6.2 Summary of chapters

6.2.1. The aim of this research paper

1. To determine the efficiency of the LeoTrade algorithm using trading strategies.
2. To design an effective trading algorithm using combined strategies.
3. To determine the performance of the algorithm and implementing it on a Trading platform.

6.3. Summary

The research paper has used literature review in three parts, that is, conceptual framework, empirical framework and theoretical framework in chapter two. The researcher has chosen different research methodologies based on the framework which are

- ✓ Examining Pseudo-Code and Source Code,
- ✓ Reflexively Producing Code,
- ✓ Reverse-engineering,

Researcher also used quantitative research approach and proprietary research philosophy like positivist and Interpretivist. The researcher used quasi experiments in order to come with required results. The researcher has used tables and charts to represent results of the algorithm and discussions.

6.4. Research findings

6.4.1. Objective

1. To determine the efficiency of the LeoTrade algorithm using trading strategies.

- From overall results of LeoTrade algorithm, the researcher has found that the efficiency of a trading algorithm is determined by the strategy selected, risk modelling and alpha combinations. The efficiency of a trading algorithm is usually determined by the total returns whereby they must be positive.

6.4.2. Objective

2. To design an effective trading algorithm using combined strategies.

- The researcher has found that a trading algorithm can trade with more than 10 companies and tracking all the activities on the stock market using a combination of strategies. The algorithm can automatically trade each and every day as long as the market is open.

6.4.3. Objective

3. To determine the performance of the algorithm on Quantopian platform.

- The researcher has also found that a good risk model can create user confidence in the trading algorithm since it has effective risk management model provide through detailed visualization on the Jupyter notebook in the Quantopian API. The Quantopian API can also show portfolio positions of the algorithm and all transactions occurred during trading and to which companies.

6.5. Significance of findings

The LeoTrade algorithm performance has proven its efficiency using Quantopian benchmarks. This is important to retail traders to first of all obtain positive returns from a trading algorithm. The better an algorithm is in risk modelling, the better the returns the algorithm retains.

6.6. Recommendations

This research paper is recommended to all retail trader who want to use algorithms in trading. Algorithmic trading can be of use in buying and selling shares in the stock market.

6.7. An algorithmic trading method using deep reinforcement learning (DRL)

This scientific research chapter introduces the Trading Deep Q-Network algorithm (TDQN), a deep reinforcement learning (DRL) solution to the algorithmic trading problem of determining the optimum trading position at any given time during a trading activity in stock markets. This unique trading approach exhibits potential after a careful analysis of its performance, outperforming the benchmark trading methods on average. The TDQN algorithm also provides a number of benefits over more conventional approaches, including a remarkable level of flexibility and a surprising resilience to a range of trading costs. Such a data-driven approach has the major advantage of avoiding the challenging task of creating explicit rules suitable for the particular financial markets under consideration.

Chapter 1 shows that SAMM operates under presumption that there are market inefficiencies, which may be identified and exploited using a six-step procedure. It is a successful strategy because its main foundations are strategic thought and human behaviour. It does not make illogical assumptions, in contrast to the efficient market hypothesis. The market game requires understanding of gambling theory because it is only played by experienced gamblers. It takes a strategic life cycle model (the POPP) to explain how markets evolve over time. Additionally, it demands that behavioural finance, the study of financial decision making, explicitly include humans. These elements are intelligently combined to produce the SAMM.

The SAMM has six steps:

(1) Pick potential price distorters (detailed below) to generate trading ideas.

(2) Put together datasets to look at how pricing distortion affects prices.

(3) Develop tactical plans to address any significant pricing issues that Step 2 indicated.

(4) Create Potentially Profitable Gambling Systems (PPGSs) by converting strategic strategies into trading algorithms.

Test trade algorithms in the past, and then repeat steps three through five as required.

(6) Following the completion of Steps (1) through (5).

6.8. Problems with Market Direction Prediction

Artificial intelligence has been employed extensively during the past ten years in a wide range of industries, and it has been demonstrated that its use considerably improves results. One of the applications that is particularly fascinating is financial markets. Exploiting these markets with artificial intelligence and machine learning might result in important developments. Examples of these usage include loan credit scoring, credit evaluation, sovereign credit ratings, mortgage choice decisions, portfolio management, financial performance forecasting, and market direction forecasting. The conclusions are shocking and thought-provoking. With millions of "active" wallets and a market worth in the billions in 2016, consumer adoption of various cryptocurrencies has actually surged.

6.9. Deep Reinforcement Trading with Predictable Returns

The cryptocurrency industry is localised in addition to cross-border exchange operations and geographically concentrated mining operations. Thirdly, the business is getting more fluid as the distinctions between exchanges and wallets are being "blurred" and more cryptocurrencies than only bitcoin are now backed by a thriving ecosystem that performs a variety of activities. (M. Asgari and H. Khasteh, 2021)

6.10. Deep Reinforcement Trading with Predictable Returns

The addition of the time dimension makes it considerably more difficult to predict an optimal strategy, which requires projecting financial parameters like risks and returns over many future years. Since dynamic analogues are unrealistic, single-period models are nevertheless often utilised. Forecasting may lead to systematic errors due to uncertainty around the model of choice or the low signal-to-noise ratio that occurs by nature in the financial data. Traditional optimum control techniques rely on a variety of limiting assumptions that are inappropriate for reproducing the real financial world, even when a multi-period model is successful in capturing market impact or alpha decay. Recent DRL algorithms are typically haphazard homebrew experiments.

The analysis of their performance in real-world financial trading problems therefore involves a complex interplay of different effects, some of which are connected to the quality of the dataset and the signals used to forecast returns, while others are related to the specific algorithm and issues with its trainability.

To the best of our knowledge, there aren't many studies that examine how DRL performs in trading challenges other those brought on by market efficiency, such as the search for a solid signal to estimate returns or the probable absence of any signals in the dataset. As a result, they consider a controlled environment where a signal is known to exist and look into the capacity. The main contribution of our study is the use of a data-driven DRL environment where agents may compete against conventional strategies while also using their experience to enhance the state-action space and speed up learning. In order to test the adaptability of different DRL algorithms when the simulated dynamics are improperly described in regard to the benchmark model's assumptions, they analyse them on a variety of financial data with different features.

DRL algorithms are capable of matching or even outperforming the performance of the benchmark strategy when a model is misspecified, such as when there are catastrophic events and volatility clustering. They also show how traditional techniques may be useful by educating DRL agents on the typical size of a good approach to begin with and modify (Brini, A., & Tantari, D. (2021)).

The findings demonstrate the consistency of the pairs trading approach, even in markets with little liquidity. It also appears to yield significant profits to profit on price differences between common and preferred shares of the same business. The latter finding considerably expands the body of knowledge since trader risk aversion in light of the illiquidity of fundamental value computation is one of the justifications offered for the lucrative trading opportunities induced by pair price divergence. Given that no transactions are made with the intention of making money via mispricing, prices may change frequently if a trader's evaluation of fundamental worth is very uncertain. As a result, the risk aversion of the trader may reduce the pairs trading strategy's potential for profit as inactivity may result in non-convergence. This simple investment trading approach is further supported by the possibility of returns from price differences across assets that have claims to the same cash flow source.

The chapter by Broussard, J. P., & Vaihekoski (2012) gives a brief overview of the estimation method employed as well as the pairs trading technique. Information and characteristics of Finnish institutions.

The key issue is how to accomplish this while accounting for the uncertainty in her estimates, choosing the appropriate combination of market and limit orders at both ends of the approach, and riding price changes and inventory exposure in between.

The mathematical literature on algorithmic trading has done the bulk of the work on

(i) Optimum liquidation/acquisition and

(ii) Market creation.

Both methods, however, generated discrete trading signals for a single asset, which may not have reflected the best trading option in the real trading environment. To get around this weakness—focusing on a single thing—some study has considered the portfolio dilemma. Market price changes are incredibly unpredictable and challenging to model. The approaches described above as a result are not very generalizable. So a certain model could perform well on the training set but poorly on the testing set. However, these studies just aimed to predict future market patterns rather than focusing on a particular trading decision. Consequently, to expand the range of applications for our portfolio management technique.

A comprehensive RL framework with the following improvements:

• A method for adaptive sampling was proposed to increase computing efficiency.

• Enhanced the model's generalizability

• We used the market representation network to pretrain our model and account for potential hazards during the training process. The maximum drawdown (MDD) was lowered by 40% when using the market representation network model, which also saw a rise in.

In traditional portfolio optimization algorithms, the mean historical return is frequently employed as the anticipated return; however, this strategy has impact behaviour and produces inaccurate estimates of short-term returns. Additionally, it is incorrect to take the specific stock since short-term market mood has a significant impact on stock price. Financial investment is modelled by portfolio optimisation. When creating portfolio optimisation models, many academics prefer projected return to expected return. In several studies, additional predicted outcomes have been added to the original model in an effort to enhance performance.

The process by which findings are achieved, or what was done to arrive at certain conclusions, is described by the research methodology. The chapter provides information on the methodology the researcher employed to gather data and acquire the algorithm's findings. The systematic, organised, and concentrated collecting of data with the goal of learning knowledge to answer research questions is referred to as research methodology. In addition to employing a variety of research tools for data analysis, the researcher tested (backtested) the effectiveness of the LeoTrade algorithm using the dual momentum approach using the Quantopian API and Zipline API. The APIs are essentially identical, however Zipline gets daily data in CSV (comma separated value) files from algorithmic trading servers while Quantopian API operates online. Both are used for backtesting.

The study methodology outlines how conclusions were reached or what was done to get particular results. The chapter details the approach the researcher used to collect data and obtain the results of the algorithm. Research technique is the systematic, structured, and focused collection of data with the aim of gaining information to address research issues. The researcher investigated (backtested) the performance of the LeoTrade algorithm utilising the dual momentum technique using the Quantopian API and Zipline API in addition to using a number of research tools for data analysis. Although Zipline receives daily data in CSV (comma separated value) files from algorithmic trading servers and Quantopian API runs online, the APIs are substantially comparable.

The third chapter's major goal was to examine the methodological decisions used for this inquiry. The research's quantitative character became apparent. The chapter included a review of the tools and data gathering techniques utilised to study the algorithm. This chapter included research ethics, design, and philosophy. The proper data analysis procedure was also taught. The following two chapters 4 and 5, demonstrate how to represent, understand, and analyse the research data used to evaluate the algorithm's performance.

This chapter 2 summarises and analyses the findings on algorithmic trading's efficiency in automating online algorithmic trading. The success of algorithmic trading using Quantopian benchmarks is demonstrated in this chapter utilising line graphs, tables, column charts, bar charts, and other methods of visualisation. Using the returns, trading universe, portfolio holdings, and other criteria described in this research thesis' chapter, the efficacy of algorithmic trading is assessed.

Algorithmic trading, which has existed from the beginning of time, is the transfer of a commodity between individuals based on the supply and demand premise. A supply of a commodity is exchanged for a demand-driven fungible asset. That is foreshadowed by the computerised trading of stock shares for cash on the stock market. Following the launch of online algorithmic trading, AT has grown as a result of the development of new technological initiatives.

The statistics on algorithmic trading's effectiveness in automating online algorithmic trading from the study techniques employed are summarised and analysed in this chapter. This chapter illustrates the effectiveness of algorithmic trading using Quantopian benchmarks using line graphs, tables, column charts, bar charts, and other visualisation techniques. The effectiveness of algorithmic trading is evaluated using the returns, trading universe, portfolio holdings, and other factors specified in this study thesis' chapters.

6.11. A Deep Reinforcement Learning Approach to Financial Trading

Any market practitioner would benefit from an automated method that reliably produces profit from the financial market. A basis for fully training such trading agents is provided by recent advances in deep reinforcement learning. In this study, we provide a Markov Decision Process (MDP) model appropriate for financial trading tasks and solve it using the cutting-edge deep recurrent Q-network (DRQN) approach.

Huang made numerous changes to the current learning algorithm to make it better suited for use in financial trading. These changes significantly reduced the amount of replay memory required, which was only a few hundred bytes compared to the millions used by more recent deep reinforcement learning algorithms. Additionally, he developed an action augmentation technique that provides additional feedback signals to the agent for all tasks in order to decrease the need for random exploration.

Because of this, we may use the greedy strategy while learning, and our empirical performance is better than that of the more typical greedy exploration. However, given precise market assumptions, this strategy is only applicable to financial trading. For the purpose of training recurrent neural networks, we sample a longer sequence. We can now train the agent for each T step using this technique. Due to the T-fold reduction in overall calculation, training time is significantly reduced. We put all of the aforementioned information together into a comprehensive online learning system and put it to the test on the live foreign exchange market.

The researcher used a quantitative research approach. The reason the researcher chose quantitative research is that it tries to employ certain methods to make data processing simpler. The researcher picked the quantitative research methodology because it permits investigation of the efficacy of algorithms using mathematical Python tools built inside the Quantopian API. A combination of research methods may be required to analyse algorithms in order to get beyond the drawbacks of employing them alone. The researcher used six methods to create the algorithm.

6.11.1. Drawbacks

The procedure is fundamentally subjective and vulnerable to personal prejudices and flaws. However it can be a useful addition to other research techniques.

6.12. Trading Algorithms when Learning

In these chapters, we show how an algorithmic trader may use market-wide price trends to learn how to trade around directional projections of the price of an asset or collection of assets. Through round-trip trade margins and the capitalization of inventory, the approach makes money. When price volatility impairs the quality of the information, the approach relies less on betting on the direction of the asset and more on two-sided quotes to make a profit. The strategy functions as a market maker who is only willing to accept a small amount of inventory risk in dire situations, such as when the trader is uniformed or the IT receives insufficient high-quality information.

We also show how trading strategies might become more profitable by borrowing from assets that have performed better. Earnings increase noticeably when a trader takes positions in a volatile asset while improving the quality of the information the IT receives by examining the dynamics of other assets that are performing well. We also show how the trader's strategy is modified to take into account the expenses associated with unfavourable selection. Finally, we show that the single viscosity solution of the QVI derived from the dynamic programming concept is really attained by the numerical approach.

AT is rising everywhere, but particularly in the US, which is home to the largest stock exchange in the world. The adoption of autoquotes by the NYSE in 2003, which permitted quicker quoting of security prices, is at least partially responsible for the creation of AT. The scope of the article concentrated on a tiny portion of the US equities market, however AT expands to other sectors including the foreign currency (FX) and derivative markets. The broad range of AT is connected to new technological innovations, where the speedier business or trader in terms of light speed measurement has the greatest benefit.

6.13. Future Work

This section summarises intriguing insights to bring this thesis to a close and suggests prospective lines of inquiry. Though an extension to a number of other markets is anticipated in the future, for the time being the focus will be on the stock market. Although the Sharpe ratio maximisation is the ultimate objective, the DRL approach utilised in this scientific study actually maximises the expected discounted sum of rewards (daily returns) over an indefinite time horizon. One may think of this optimisation criterion as a loosening up of the Sharpe ratio criterion because it comes incredibly close to but falls short of profit maximisation. Future study in this area should focus on bridging the gap between these two objectives. Nevertheless, this practise is an intriguing area for further study.

The optimal adjustment of this parameter in the context of this algorithmic trading problem is not straightforward due to the significant future uncertainty. On the other hand, it wouldn't be a smart idea to place too much stock in the stock market's future. It mimics the notion that a trading agent has to have enough confidence in the future to balance the additional risk brought on by the cost of trading. A low value for the parameter will inescapably reduce the RL agent's propensity to change its trading position, which decreases the TDQN algorithm's trading frequency. The importance given to the future is determined by the discount factor.

Second, despite the fact that the distribution of the daily returns is always changing, the past must be sufficiently predictive of the future in order for the TDQN algorithm to produce reliable results. The output of a purely predictive system, on the other hand, would just consist of forecasts on the direction or price of the market in the future, with no suggestions for profitable trading methods that take trading costs into account. The final method is less successful by design even if it offers more flexibility and could lead to profitable trading techniques.

6.13.1. Recommendations for future work

This research article is suggested for further study since algorithmic trading is becoming more and more popular worldwide and because the LeoTrade algorithm design may be used to create new algorithms that can forecast market trends. A more effective algorithm may be made by using artificial intelligence techniques to develop the algorithm.

APPENDIX A

SURVEY COVER LETTER

From

Dr. George E. latridis,

To

The Dean, Academic Research,

Respected Sir,

Sub: Submission of Thesis for Virendra Kumar Yadav

Register No:

I wish to inform you that my scholar Virendra Kumar Yadav is going to submit his thesis of DBA. Degree entitled "AUTOMATING THE CREATION OF TRADING STRATEGIES USING DEEP REINFORCEMENT LEARNING: ALGORITHMIC TRADING**"** on 22/02 /2024.

Thanking you

Yours Sincerely

(Dr. George E. latridis)

APPENDIX B

INFORMED CONSENT

TITLE OF STUDY

AUTOMATING THE CREATION OF TRADING STRATEGIES USING DEEP REINFORCEMENT LEARNING: ALGORITHMIC TRADING

PRINCIPAL INVESTIGATOR

[Name]

[Department]

[Address]

[Phone]

[Email]

PURPOSE OF STUDY

It is requested that you participate in a research project. It's crucial that you are aware of the goals and procedures of the study before deciding whether or not to take part. Please take your time reading the following material. If there is anything unclear or if you want additional information, do ask the researcher.

This study's main finding is that the RL state space's dimensionality is probably too low for the agent to learn a trading algorithm that is highly adaptable and appropriate for a variety of volatility regimes. The VG training data suggest that in this case, doing "more of the same" would be the best course of action. The main problem will now be persuading the RL agent to utilise the additional information that the 25 multi-asset futures contracts definitely provide. Third, further study is required to fully comprehend why evaluation frequency is essential for preventing disasters like the one in 2008, with semi-annually serving as the sad exception.

There are many strange things about this. Fourth, the one strategy that has worked. Given that investors are usually risk averse, the objective is to maximise a risk-adjusted performance function, such as the Sharpe ratio.

Actually, a concave utility function is produced by this. After learning the distribution, we may select actions that have the highest predicted Q-value and the lowest standard deviation from it to maximise the Sharpe ratio. By altering the action spaces to include weights for specific contracts in a portfolio, we expand our methods for optimising portfolios. To have a great predicted algorithmic trade return with minimum volatility, the reward functions and mean-variance portfolio theory can be used.

CONFIDENTIALITY

Your answers to this [survey] will remain private. Do not include any personal information in your [survey]. Your responses won't be kept anonymous for the purposes of this research project. The researcher will make every attempt to protect your privacy, including the following:

• Giving participants code names or numbers that will appear on all study notes and papers

• Storing the researcher's own notes, interview transcripts, and any other information that may be used to identify participants.

Except in circumstances where the researcher is required by law to report particular instances, participant data will be kept private. Instances of abuse and suicide risk are among these instances, although they are not necessarily confined to them.

CONTACT INFORMATION

Contact the researcher using the information on the first page if you have any concerns about this study at any point or if taking part in it has caused you to experience any negative consequences. Please call the Institutional Review Board, if you have any queries about your rights as a study participant or if you run into issues that you don't feel comfortable discussing with the primary investigator.

VOLUNTARY PARTICIPATION

You are not required to take part in this study. You have the choice of participating or not in this study. You will have to sign a permission form if you choose to participate in the study. You have the right to withdraw your approval at any moment and without providing a reason even after signing the consent form. The interaction you have, if any, with the researcher will not be impacted if you decide to leave this study. Your data will either be returned to you or deleted if you drop out of the research before it is done collecting it.

CONSENT

I've read, comprehend, and have had a chance to ask questions about the material presented. I am aware that my participation is entirely optional and that I can stop at any moment, for any reason, and without incurring any fees. I am aware that a copy of this permission form will be provided to me. I freely consent to participate in this study.

Participant's signature ______________________________ Date __________

Investigator's signature _____________________________ Date __________

APPENDIX C

INTERVIEW GUIDE

Research project title:

Research investigator:

Research Participants name:

Thank you for agreeing to be interviewed as part of the above research project. Ethical procedures for academic research require that interviewees explicitly agree to being interviewed and how the information contained in their interview will be used. This consent form is necessary for us to ensure that you understand the purpose of your involvement and that you agree to the conditions of your participation. Would you therefore read the accompanying information sheet and then sign this form to certify that you approve the following:

- the interview will be recorded and a transcript will be produced
- you will be sent the transcript and given the opportunity to correct any factual errors
- the transcript of the interview will be analysed by Virendra Kumar Yadav as research investigator access to the interview transcript will be limited to Virendra Kumar Yadav and academic colleagues and researchers with whom he might collaborate as part of the research process

Quotation Agreement

I also understand that my words may be quoted directly. With regards to being quoted, please initial next to any of the statements that you agree with:

	I wish to review the notes, transcripts, or other data collected during the research pertaining to my participation.
	I agree to be quoted directly.
	I agree to be quoted directly if my name is not published and a made-up name (pseudonym) is used.
	I agree that the researchers may publish documents that contain quotations by me.

All or part of the content of your interview may be used;

- In academic papers, policy papers or news articles
- On our website and in other media that we may produce such as spoken presentations
- On other feedback events
- In an archive of the project as noted above by

signing this form I agree that;

1. I am voluntarily taking part in this project. I understand that I don't have to take part, and I can stop the interview at any time;
2. The transcribed interview or extracts from it may be used as described above;
3. I have read the Information sheet;
4. I don't expect to receive any benefit or payment for my participation;
5. I can request a copy of the transcript of my interview and may make edits I

feel necessary to ensure the effectiveness of any agreement made about confidentiality;

6. I have been able to ask any questions I might have, and I understand that I am free to contact the researcher with any questions I may have in the future.

____________________________ ________________

Participants Signature Date

____________________________ ________________

Researchers Signature Date

Contact Information

This research has been reviewed and approved by the University Research Ethics Board. If you have any further questions or concerns about this study, please contact:

Virendra Kumar Yadav

Tel: 0648449521

E-mail: smilingvirendra@gmail.com

REFERENCES

AbdelKawy, R., Abdelmoez, W. M., & Shoukry, A. (2021). A synchronous deep reinforcement learning model for automated multi-stock trading. *Progress in Artificial Intelligence*, *10*(1), 83-97.

Asgari, M., & Khasteh, H. (2021). Application of Three Different Machine Learning Methods on Strategy Creation for Profitable Trades on Cryptocurrency Markets. *ArXiv Preprint ArXiv:2105.06827*. https://arxiv.org/abs/2105.06827

Attanasio, G., Cagliero, L., Garza, P., & Baralis, E. (2019). *Quantitative cryptocurrency trading*. 1–6. https://doi.org/10.1145/3336499.3338003

Bajpai, S. (2021). *Application of deep reinforcement learning for Indian stock trading automation*. http://arxiv.org/abs/2106.16088

Beverungen, A. (2019). *Algorithmic Trading, Artificial Intelligence and the Politics of Cognition*. 77–94. https://doi.org/10.14361/9783839447192-005

Bloch, D. A. (2018). Recipe for Quantitative Trading with Machine Learning. *SSRN Electronic Journal*, *April*. https://doi.org/10.2139/ssrn.3232143

Brini, A., & Tantari, D. (2021). Deep Reinforcement Trading with Predictable Returns. *arXiv preprint arXiv:2104.14683*.

Briola, A., Turiel, J., Marcaccioli, R., & Aste, T. (2021). *Deep reinforcement learning for active high frequency trading. arXiv preprint arXiv:2101.07107*.

Broussard, J. P., & Vaihekoski, M. (2012). Profitability of pairs trading strategy in an illiquid market with multiple share classes. *Journal of International Financial Markets,* *Institutions and Money, 22*(5), 1188-1201.

Bertoluzzo, F., & Corazza, M. (2012). Testing different reinforcement learning configurations for financial trading: Introduction and applications. *Procedia Economics and Finance, 3*, 68-77.

Cartea, Á., Jaimungal, S., & Kinzebulatov, D. (2016). ALGORITHMIC TRADING with LEARNING. *International Journal of Theoretical and Applied Finance*, *19*(4). https://doi.org/10.1142/S021902491650028X

Cartea, Á., Jaimungal, S., & Sánchez-Betancourt, L. (2021). Deep reinforcement learning for algorithmic trading. *Available at SSRN*.

Casgrain, P., & Jaimungal, S. (2016). Trading Algorithms with Learning in Latent Alpha Models. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2871403

Charupat, N., & Miu, P. (2016). Trading Strategies. *Leveraged Exchange-Traded Funds*, 111–138. https://doi.org/10.1057/9781137478214_7

Chen, H. B., Liang, D. L., & Zhao, L. L. (2019). A quantitative trading method using deep convolution neural network. *IOP Conference Series: Materials Science and Engineering*, *490*(4). https://doi.org/10.1088/1757-899X/490/4/042018

Chen, S., Luo, W., & Yu, C. (2021). *Reinforcement Learning with Expert Trajectory For Quantitative Trading*. http://arxiv.org/abs/2105.03844

Chen, Y., Liu, K., Xie, Y., & Hu, M. (2020). Financial Trading Strategy System Based on Machine Learning. *Mathematical Problems in Engineering, 2020*. https://doi.org/10.1155/2020/3589198

Computing, M. (2019). *Algorithmic Trading using Mean Reversion Indicators*. *8*(6), 7–13.

Da Costa, J., & Gebbie, T. (2020). Learning low-frequency temporal patterns for quantitative trading. *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, 1091–1099. https://doi.org/10.1109/SSCI47803.2020.9308232

Dabérius, K., Granat, E., & Karlsson, P. (2019). Deep Execution - Value and Policy Based Reinforcement Learning for Trading and Beating Market Benchmarks. *SSRN Electronic Journal*, *November 2018*. https://doi.org/10.2139/ssrn.3374766

Dai, H. (2020). Interval Validity Testing and Optimization in Quantitative Trading. *ACM International Conference Proceeding Series*, *7*, 305–312. https://doi.org/10.1145/3414752.3414807

Dempster, M. A., & Leemans, V. (2006). An automated FX trading system using adaptive reinforcement learning. *Expert Systems with Applications*, *30*(3), 543-552.

De Spiegeleer, J., Madan, D. B., Reyners, S., & Schoutens, W. (2018). Machine learning for quantitative finance: fast derivative pricing, hedging and fitting. *Quantitative Finance*, *18*(10), 1635–1643. https://doi.org/10.1080/14697688.2018.1495335

Diego, S. (2019). *A Backtesting Protocol in*. 1–11.

Dixon, M. F., Klabjan, D., & Bang, J. H. (2017). Classification-Based Financial Markets Prediction Using Deep Neural Networks. *SSRN Electronic Journal*, 1–20. https://doi.org/10.2139/ssrn.2756331

Dutta, E., Diwan, S. and Chakrabarty, S.P., 2024. ESG driven pairs algorithm for sustainable trading: Analysis from the Indian market. *arXiv preprint arXiv:2401.14761*.

Elliot, J. R., Hoek, J. V., & Malcolm, W. P., (2005). Pairs Trading, Quantitative Finance *5*(3), 271-276.

Evan, G., William, N. & Rouwenhorst, K. G. (2006), Pairs Trading: Performance of a RelativeValue Arbitrage Rule, *The Review of Financial Studies, 19*(3).

Emerson, S., Kennedy, R., O'Shea, L., & O'Brien, J. (2019). Trends and Applications of Machine Learning in Quantitative Finance. *8th International Conference on Economics and Finance Research (ICEFR)*.

Fengqian, Di., & Chao, L. (2020). An Adaptive Financial Trading System Using Deep Reinforcement Learning with Candlestick Decomposing Features

Fernandez-Tapia, J. (2015). High-Frequency Trading Meets Reinforcement Learning: Exploiting the Iterative Nature of Trading Algorithms. *Ssrn*, 1–28. https://doi.org/10.2139/ssrn.2594477

es. *IEEE Access*, *8*, 63666–63678. https://doi.org/10.1109/ACCESS.2020.2982662

Gao, X. (2018). Deep reinforcement learning for time series: playing idealized trading games. *arXiv preprint arXiv:1803.03916*.

Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies, 19*(3), 797-827.

Ge, M., Zhou, S., Luo, S., & Tian, B. (2022). *3D Tensor-based Deep Learning Models for Predicting Option Price*. 1–6. https://doi.org/10.1109/icisct52966.2021.9670100

Golub, A., Glattfelder, J. B., & Olsen, R. B. (2018). The alpha engine: Designing an automated trading algorithm. *High-Performance Computing in Finance: Problems, Methods, and Solutions*, 49–76. https://doi.org/10.1201/9781315372006

Hansen, K. B. (2020). The virtue of simplicity: On machine learning models in algorithmic trading. *Big Data and Society*, *7*(1). https://doi.org/10.1177/2053951720926558

Hirsa, A., Hadji Misheva, B., Osterrieder, J., & Posth, J. A. (2021). Deep reinforcement learning on a multi-asset environment for trading. *Available at SSRN 3867800*.

Huang, S. H., Miao, Y. H., & Hsiao, Y. T. (2021). Novel Deep Reinforcement Algorithm with Adaptive Sampling Strategy for Continuous Portfolio Optimization. *IEEE Access*, *9*, 77371–77385. https://doi.org/10.1109/ACCESS.2021.3082186

Huang, C. Y. (2018). Financial trading as a game: A deep reinforcement learning approach. *arXiv preprint arXiv:1807.02787*.

Hu, G., 2023. Advancing Algorithmic Trading: A Multi-Technique Enhancement of Deep Q-Network Models. *arXiv preprint arXiv:2311.05743*.

Hu, Z., Teresa, S., & Garcia, P. (2020). Algorithmic quoting, trading, and market quality in agricultural commodity futures markets. *Applied Economics*, *00*(00), 6277–6291. https://doi.org/10.1080/00036846.2020.1789060

Huck, N. (2009). Pairs selection and outranking: An application to the S&P 100 index. *European Journal of Operational Research, 196*(2), 819-825.

Huck, N. (2010). Pairs trading and outranking: The multi-step-ahead forecasting case. *European Journal of Operational Research, 207*(3), 1702-1716.

Huck, N., & Afawubo, K. (2015). Pairs trading and selection methods: is cointegration superior? *Applied Economics, 47*(6), 599-613.

Jacobs, H., & Weber, M. (2015). On the determinants of pairs trading profitability. *Journal of Financial Markets, 23*, 75-97.

Joshi, J. (2018). Developing an Artificial Neural Network Algorithmic Trading System. *Available at SSRN 3889600*.

Jansen, S. (2020). Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python. Packt Publishing Ltd.

Kanamura, T., Rachev, S. T., & Fabozzi, F. J. (2009). A profit model for spread trading with an application to energy futures. *The Journal of Trading, 5*(1), 48-62.

Kim, J. G., & Lee, B. (2020). Automatic P2P energy trading model based on reinforcement learning using long short-term delayed reward. *Energies*, *13*(20), 5359.

Kole, H. J. W. G. (2017). *Pairs trading using machine learning: An empirical study*.

Lei, K., Zhang, B., Li, Y., Yang, M., & Shen, Y. (2020). Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. *Expert Systems with Applications*, *140*, 1–14. https://doi.org/10.1016/j.eswa.2019.112872

Li, Y., Burns, C., & Hu, R. (2016). Representing stages and levels of automation on a decision ladder: The case of automated financial trading. *Proceedings of the Human Factors and Ergonomics Society*, *0*, 328–332. https://doi.org/10.1177/1541931213601074

Li, X., Deng, X., Zhu, S., Wang, F., & Xie, H. (2014). An intelligent market making strategy in algorithmic trading. *Frontiers of Computer Science, 8*(4), 596–608. https://doi.org/10.1007/s11704-014-3312-6

Li, Y., Ni, P., & Chang, V. (2020). Application of deep reinforcement learning in stock trading strategies and stock forecasting. *Computing*, *102*(6), 1305–1322. https://doi.org/10.1007/s00607-019-00773-w

Li, Y., Zheng, W., & Zheng, Z. (2019). Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access*, *7*, 108014-108022.

Lin, H., Zhou, D., Liu, W., & Bian, J. (2021). Learning Multiple Stock Trading Patterns with Temporal Routing Adaptor and Optimal Transport. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Vol. 1, Issue 1). Association for Computing Machinery. https://doi.org/10.1145/3447548.3467358

Liu, X.-Y., Yang, H., Chen, Q., Zhang, R., Yang, L., Xiao, B., & Wang, C. (2021). FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance. *SSRN Electronic Journal*, *NeurIPS*, 1–11. https://doi.org/10.2139/ssrn.3737859

Liu, Y., Liu, Q., Zhao, H., Pan, Z., & Liu, C. (2020). Adaptive quantitative trading: An imitative deep reinforcement learning approach. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2128–2135. https://doi.org/10.1609/aaai.v34i02.5587

Lu, N. (2016). *A Machine Learning Approach to Automated Trading*. 41.

Lv, D., Yuan, S., Li, M., & Xiang, Y. (2019). An Empirical Study of Machine Learning Algorithms for Stock Daily Trading Strategy. *Mathematical Problems in Engineering*, *2019*(2). https://doi.org/10.1155/2019/7816154

Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, *165*(June 2020), 113973. https://doi.org/10.1016/j.eswa.2020.113973

Madan, I., Saluja, S., & Zhao, A. (2015). Automated bitcoin trading via machine learning algorithms. *URL: http://cs229. stanford. edu/proj2014/Isaac% 20Madan*, *20*.

Moffitt, S. (2018). How Two Profitable Trading Systems for Eurodollar Futures Were

Developed. *SSRN Electronic Journal*, *1993*, 1–61. https://doi.org/10.2139/ssrn.3179171

Ntakaris, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2020). Mid-price prediction based on machine learning methods with technical and quantitative indicators. In *PLoS ONE* (Vol. 15, Issue 6 June). https://doi.org/10.1371/journal.pone.0234107

Paiva, F. C. L., Felizardo, L. K., Bianchi, R. A. D. C., & Costa, A. H. R. (2021). Intelligent Trading Systems: A Sentiment-Aware Reinforcement Learning Approach. *arXiv preprint arXiv:2112.02095*.

Perlin, M. S. (2009). Evaluation of pairs-trading strategy at the Brazilian financial market. *Journal of Derivatives & Hedge Funds*, *15*(2), 122-136.

Ponomarev, E. S., Oseledets, I. V., & Cichocki, A. S. (2019). Using Reinforcement Learning in the Algorithmic Trading Problem. *Journal of Communications Technology and Electronics*, *64*(12), 1450–1457. https://doi.org/10.1134/S1064226919120131

Pricope, T. V. (2021). Deep reinforcement learning in quantitative algorithmic trading: A review. *arXiv preprint arXiv:2106.00123*.

Qin, M., Sun, S., Zhang, W., Xia, H., Wang, X. and An, B., 2023. Earnhft: Efficient hierarchical reinforcement learning for high frequency trading. *arXiv preprint arXiv:2309.12891*.

Ritter, G. (2017). Machine Learning for Trading. *SSRN Electronic Journal*, 1–19. https://doi.org/10.2139/ssrn.3015609

Rundo, F., Trenta, F., di Stallo, A. L., & Battiato, S. (2019). Machine learning for quantitative finance applications: A survey. *Applied Sciences (Switzerland)*, *9*(24). https://doi.org/10.3390/app9245574

Rundo, F., Trenta, F., Di Stallo, A. L., & Battiato, S. (2019). Advanced Markov-based machine learning framework for making adaptive trading system. *Computation*, *7*(1). https://doi.org/10.3390/computation7010004

Sarmento, S. M., & Horta, N. (2020). Enhancing a pairs trading strategy with the application of machine learning. *Expert Systems with Applications*, *158*, 113490.

Sawhney, R., Wadhwa, A., Agarwal, S., & Shah, R. R. (2021). *Quantitative Day Trading from Natural Language using Reinforcement Learning*. 4018–4030. https://doi.org/10.18653/v1/2021.naacl-main.316

Si, W., Li, J., Ding, P., & Rao, R. (2017, December). A multi-objective deep reinforcement learning approach for stock index future's intraday trading. In *2017 10th International symposium on computational intelligence and design (ISCID)* (Vol. 2, pp. 431-436). IEEE.

Snow, D. (2020). Machine Learning in Asset Management— Part 2: Portfolio Construction—Weight Optimization . *The Journal of Financial Data Science*, *2*(2), 17–24. https://doi.org/10.3905/jfds.2020.1.029

Soares Koshiyama, A., Firoozye, N., & Treleaven, P. (2020). Algorithms in Future Capital Markets. *SSRN Electronic Journal*, 1–23. https://doi.org/10.2139/ssrn.3527511

Sokolovsky, A., Arnaboldi, L., Bacardit, J., & Gross, T. (2021). Explainable Machine Learning-driven Strategy for Automated Trading Pattern Extraction. *arXiv preprint arXiv:2103.12419*.

Srihith, I.D., Rajitha, L., Thriveni, K., Donald, A.D. and Blessy, P. (2023), *Trading on Autopilot: The Rise of Algorithmic Trading*.

Suri, K., Shi, X. Q., Plataniotis, K., & Lawryshyn, Y. (2021). *TradeR: Practical Deep Hierarchical Reinforcement Learning for Trade Execution*. http://arxiv.org/abs/2104.00620

Ta, V. D., Liu, C. M., & Addis, D. (2018). Prediction and portfolio optimization in quantitative trading using machine learning techniques. *ACM International Conference Proceeding Series*, 98–105. https://doi.org/10.1145/3287921.3287963

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

Tsantekidis, A., Passalis, N., Toufa, A. S., Saitas-Zarkias, K., Chairistanidis, S., & Tefas, A. (2021). Price Trailing for Financial Trading Using Deep Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(7), 2737–2846. https://doi.org/10.1109/TNNLS.2020.2997523

Vats, P., & Samdani, K. (2019). Study on machine learning techniques in financial markets. *2019 IEEE International Conference on System, Computation, Automation and Networking, ICSCAN 2019*, 3–7. https://doi.org/10.1109/ICSCAN.2019.8878741

Vinitnantharat, N., Inchan, N., Sakkumjorn, T., Doungjitjaroen, K., & Worasucheep, C. (2019). Quantitative Trading Machine Learning Using Differential Evolution Algorithm. *JCSSE 2019 - 16th International Joint Conference on Computer Science and Software Engineering: Knowledge Evolution Towards Singularity of Man-Machine Intelligence*, 230–235. https://doi.org/10.1109/JCSSE.2019.8864226

Volta, V. (n.d.). *Profitable Algorithmic Trading Strategies in Mean-Reverting Markets*.

Vidyamurthy, G. (2004). Pairs Trading: quantitative methods and analysis. John Wiley &

Sons.

Wang, Q., Zhou, Y., & Shen, J. (2021). *Intraday trading strategy based on time series and machine learning for Chinese stock market*. 1–47. http://arxiv.org/abs/2103.13507

Wang, Z., Lu, W., Zhang, K., Li, T., & Zhao, Z. (2021). A parallel-network continuous quantitative trading model with GARCH and PPO. In *Proceedings of ACM Conference (Conference'17)* (Vol. 1, Issue 1). Association for Computing Machinery. https://doi.org/10.1145/XX

Wray, A., Meades, M., & Cliff, D. (2020). Automated Creation of a High-Performing Algorithmic Trader via Deep Learning on Level-2 Limit Order Book Data. *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, *December*, 1067–1074. https://doi.org/10.1109/SSCI47803.2020.9308266

Wu, X., Chen, H., Wang, J., Troiano, L., Loia, V., & Fujita, H. (2020). Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, *538*, 142–158. https://doi.org/10.1016/j.ins.2020.05.066

Wu, Jia, Wang, C., Xiong, L., & Sun, H. (2019). Quantitative Trading on Stock Market Based on Deep Reinforcement Learning. *Proceedings of the International Joint Conference on Neural Networks*, *2019-July*(July), 1–8. https://doi.org/10.1109/IJCNN.2019.8851831

Wu, Junran, Xu, K., Chen, X., Li, S., & Zhao, J. (2022). Price graphs: Utilizing the structural information of financial time series for stock prediction. *Information Sciences*, *588*, 405–424. https://doi.org/10.1016/j.ins.2021.12.089

Xiong, Z., Liu, X.-Y., Zhong, S., Yang, H., & Walid, A. (2018). *Practical Deep Reinforcement Learning Approach for Stock Trading*. *i*. http://arxiv.org/abs/1811.07522

Xu, M., Lan, Z., Tao, Z., Du, J. and Ye, Z., 2023. Deep Reinforcement Learning for Quantitative Trading. *arXiv preprint arXiv:2312.15730*.

Yang, X., Liu, W., Zhou, D., Bian, J., & Liu, T.-Y. (2020). *Qlib: An AI-oriented Quantitative Investment Platform*. http://arxiv.org/abs/2009.11189

Yang, H., Liu, X. Y., Zhong, S., & Walid, A. (2020). *Deep reinforcement learning for automated stock trading: An ensemble strategy.* In *Proceedings of the First ACM International Conference on AI in Finance* (pp. 1-8).

Yang, H., Liu, X.Y. and Wang, C.D., 2023. FinGPT: Open-Source Financial Large Language Models. arXiv. *arXiv preprint arXiv:2306.06031*.

Yu, C., & Xie, T. (2021). *Multivariate Pair Trading by Volatility & Model Adaption Trade-off*. 1–15. http://arxiv.org/abs/2106.09132

Zarkias, K. S., Passalis, N., Tsantekidis, A., & Tefas, A. (2019, May). Deep reinforcement learning for financial trading using price trailing. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 3067-3071). IEEE.

Zhang, J., & Maringer, D. (2016). Using a Genetic Algorithm to Improve Recurrent Reinforcement Learning for Equity Trading. *Computational Economics*, *47*(4), 551–567. https://doi.org/10.1007/s10614-015-9490-y

Zhang, Z., Zohren, S., & Roberts, S. (2020). Deep reinforcement learning for trading. *The Journal of Financial Data Science*, *2*(2), 25-40.

Zhang, H. (2021). A Comparative Evaluation of Predominant Deep Learning Quantified Stock Trading Strategies. *2021 IEEE 4th International Conference on Big Data and Artificial Intelligence, BDAI 2021*, 15–25.

https://doi.org/10.1109/BDAI52447.2021.9515244

Zhao, C. (2019). Comparing the effectiveness of multiple quantitative trading strategies. *ACM International Conference Proceeding Series*, *336*, 47–51. https://doi.org/10.1145/3307363.3307391

Zhuang, X.-N., Chen, Z.-Y., Wu, Y.-C., & Guo, G.-P. (2021). *Quantum Quantitative Trading: High-Frequency Statistical Arbitrage Algorithm*. 1–10. http://arxiv.org/abs/2104.14214.