Capstone Project

Reinforcement Learning in Cryptocurrency Trading: An Investigation of Effective Strategies

Francisco Heshiki de las Casas

Supervisor: Professor Manoel Fernando Alonso Gadi

School of Science and Technology
IE University
Madrid, Spain
May 3, 2024

Abstract

This research paper explores the application of Deep Reinforcement Learning (DRL) to cryptocurrency trading, seeking to evaluate its effectiveness compared to traditional machine learning and algorithmic trading techniques. The study centers on the volatile cryptocurrency markets and the advanced capabilities of machine learning technologies, offering a comprehensive comparison of different trading strategies. Specifically, the focus is on understanding how DRL can enhance trading algorithms to better adapt and perform in the dynamically changing and often unpredictable market environments characteristic of cryptocurrency trading. This investigation leverages theoretical frameworks and simulation models rather than methodical implementations, focusing on conceptual advancements and theoretical validation of DRL strategies fine-tuned to the high-stakes cryptocurrency financial sector.

Keywords: Reinforcement Learning, Cryptocurrency Trading, Algorithmic Trading, Deep Neural Networks, Technical Analysis, Blockchain.

Contents

1. Introduction

In the year 2009, a new technology called Bitcoin was introduced to the public by the illusive figure named Satoshi Nakamoto. This sparked the emergence of the cryptocurrency market which slowly grew into a massive global industry currently valued at around $2 trillion USD (*Forbes, 2024*). The decentralized nature and cryptographic foundations of Bitcoin attracted a large audience since early adopters popularized it due to its potential for technological applications and investment.

One of the major applications that spawned with the rise of technological advancements was the field of algorithmic trading and high frequency trading. This involves powerful computers with advanced mathematics designed to make trading decisions that maximize returns and minimize risk without the need for much human intervention. Traditionally, algorithmic trading involved humans cherry picking trading strategies and methodologies that were hard-coded and tested to verify their usability. Instead of the average investment of buying and holding over longer periods of time, the algorithms seek to exploit the short time price fluctuations to make smaller but more frequent profits. While these methods may be profitable and can perform operations at a higher frequency to their human competitors they introduced a lot of biases and rigidness. Meaning, the traditional trading algorithm can't easily adapt to a highly volatile market with many changing parameters. The volatile nature of the cryptocurrency market caused by its strong online presence, hype, and user trust are some of the main reasons for the sudden bursts of investments and price swings (*Bakas et al., 2022*). Therefore, while the underlying structure of the crypto market is alike to other markets, traditional trading algorithms used on the crypto market don't perform at the level they would in a more stable environment like the US stock market.

The need for better fined-tuned algorithms leads us to the development of new techniques for state-of-the-art algorithmic trading in the modern era. Newer algorithms incorporate advanced Machine Learning (ML) models that process immense amounts of historical data in order to find patterns and automatize the entire learning process. The result is a model capable of performing tasks such as price prediction, risk management, decision making, and more. A promising sub-field of ML in the world of algotrading is Reinforcement Learning (RL). RL algorithms learn to optimize their reward (results) by using a process similar to trial-and-error. The cryptocurrency market can be used as an environment upon which the RL algorithm will learn and adapt based on feedback from interacting with said environment.

This report aims to delve deeper into the intersection between the volatile cryptocurrency market and deep reinforcement learning algorithms to see if effective results can be obtained. It is also important to compare with other traditional ML techniques and trading strategies. Through a systematic analysis of said techniques, the research seeks to not only address the effectiveness of incorporating RL but also to compare and optimize diverse trading strategies to navigate the challenges presented by the ever-changing cryptocurrency market.

2. Literature Review

In recent years, there has been much work done regarding algorithmic trading and the integration of Artificial Intelligence into the field. This is partially due to the expanding markets and perceived profitability of investing in stocks, foreign exchange, and cryptocurrencies. One article estimates that algotrading accounted for 80% of the entire equity turnover in the U.S. in 2018, compared to 50% in 2011 (*Burgess, 2019*). While they all have differences, their overall structure is similar. Meaning, most algotrading techniques involving AI can be applied to various markets after some fine-tuning and access to the right data. Many researchers are searching to optimize various aspects of the algotrading field in order to maximize returns and accuracy. In recent years, these algorithms have been applied to the cryptocurrency market more due to its large potential profitability. Although, this comes at the cost of higher risk due to the large amounts of speculation and volatility (*Woebbeking, 2021*).

Due to the many different approaches proposed in the literature it is important to understand algorithmic trading and all its aspects at a deep level. This is necessary to analyze the full potential of ML and RL in this field without coming to abrupt conclusions. Relevant material will be reviewed to have a draw solid results of the current state of algotrading systems and their differences. Additionally, while the interest in the crypto market has been exponentially increasing, the standard stock markets still receive most of the attention (*Fang et al., 2022*). Therefore, algotrading for the crypto market still has some gaps to be filled.

Research in the field demonstrates that while the best performing models have been implemented for stocks, not all have been fine-tuned for the crypto market. Meaning there is a lot of room for experimentation by applying these advanced models to cryptocurrencies specifically and potentially increase performance. This potential will be explored with a technical implementation of some ML and RL models and a final analysis comparing it to other performing models. Further research will also be conducted into the advantages and disadvantages between ML models and RL models for cryptocurrency algotrading. This literature review will focus on gathering information about algorithmic trading, reinforcement learning, and technical aspects for a fully automatized trading algorithm.

2.1 Algorithmic Trading for Cryptocurrency

As defined by *Kolm et al.* in *2010*, Algorithmic Trading refers to the use of computer programs to automate pretrade analysis, trading signal generation, and trade execution. Having all the required elements is what separates algorithmic trading from informed trading. The focus of this study will be fully automatized systems that are capable of analyzing, generating, and performing each action completely independently with the use of computer programs. The definition also gives us the three main components which serve as base requirements for the algorithm to work in a trading environment.

First, pretrade analysis, requires a quality source of data that includes historical data for training/backtesting and real-time data for ongoing deployment. Assurance of data quality is exceedingly important in algorithmic trading since its dealing with real assets and could lead to big losses. Proper data standards should be enforced before analysis and trading can occur (*Kilkenny & Robinson, 2018*). Overall, the pretrade analysis component is responsible for the training of the algorithm and extraction of evaluation metrics from the input.

Second, trading signal generation refers to taking the inputs provided by the pretrade analysis and computationally translating that information into outputs. This important component serves as the brain of the program responsible for decisions and predictions of the model. There are many approaches for implementing this component, but the ones producing the best results recently use ML algorithms and Neural Networks. However, the features and indicators need to be carefully crafted in order to optimize profitability and efficiency. In some cases, over analyzing and computing may lead to slow results which is undesirable when dealing with real-time data (*Cohen & Qadan, 2022*).

Finally, the trade execution component of the program will handle the interaction with the environment. In the case of the cryptocurrency market, it will buy, sell, or hold varying quantities of assets. Further investigation will explore the possibility of deciding between different cryptocurrencies. Additionally, the trading execution component will pass the actions through a set of constraints and logic flow to make final adjustments to trade size, execution time, and more (*Kolm & Maclin, 2010*).

2.1.1 Trading Signals, Strategies, & Evaluation Metrics

Trading signals are indicators or triggers that guide an algotrading system in decision making. These signals are generated through technical analysis and/or the algorithmic model. They are important indicators of potential changes in the market and provide actionable insights to the system. Some trading signals can be derived from equations, obtained by chart patterns, or directly generated by the algorithm. A popular trading signal is the "Moving Average Crossover" generated when a short-term moving average intersects with a long-term moving averages (*Granville, 1963*). However, in terms of reinforcement learning this may alter depending on the methods employed.

Trading strategies are systematic approaches that guide an algotrading system in executing decisions in the given market. Essentially, strategies are sets of rules and control flow that will help the algorithm reach a final decision. These strategies may be altered over time depending on various parameters drawn from market analysis, risk, objectives, and others. While standard trading strategies like trend following and momentum trading are useful, in algorithmic trading these simple strategies tend to be of less importance since the decision-making process lies within the algorithm itself (*Lv et al., 2019*).

Standard trading evaluation metrics are still used intensively in algotrading to evaluate the performance, effectiveness, and risk. These metrics serve an important role for the algorithm to make informed decisions, optimize results, and refine their strategy. Evaluation metrics are employed during the development, backtesting, and optimization phases of algorithmic trading (*Norman, 2019*). Continuous monitoring and adjustment based on these metrics are integral to maintaining a successful and adaptive algorithmic trading strategy. Evaluation metrics will also be used in the final analysis of the results to indicate the overall performance of the algorithms. Some important examples of evaluation metrics include the following:

1. Net Profit and Loss: Measure for the profit or loss over a period of time of the algorithm

2. Maximum Drawdown: Risk measure that is the difference between a local maximum and local minimum

3. Sharpe Ratio: Measures the risk-adjusted return where a higher Sharpe Ratio means better risk-adjusted returns.

Blockchain specific metrics and indicators can also be derived for more specific fine-tuning (*King et al., 2024*).

2.1.2 Machine Learning in Trading

Machine Learning has become the primary field of research for algotrading due to its wide range of applications. ML is utilized in algotrading to better decision-making processes, optimize trading strategies, and to adapt to the dynamic market conditions. There are many examples of ML models showing positive results that support the claim that ML can be applied effectively for algotrading and even accurately predict cryptocurrency prices and devise profitable trading strategies despite the high volatility and risk. Common ML algorithms used include decision trees, support vector machines (SVMs), and neural networks (*Sebastião & Godinho, 2021*). ML models are being used for price prediction as its one of the main indicators of profitability (*Alessandretti et al., 2018*). Other researchers are using multi-model approaches that combine fields like sentiment analysis with other indicators in order to get a more complete market analysis (*Colianni et al., 2015*). There are mainly opensource frameworks and resources surrounding ML for algotrading that are readily available to test in your own environment. Therefore, a ML algorithms will be used for mere comparison and not implemented.

2.1.3 Backtesting

Backtesting is a crucial step in the development of a trading algorithm that evaluates the performance of the strategy. It does so by having the algorithm operate with a representative sample of historical data and simulates the trading decisions the algorithm makes based on its strategy. The backtesting environment will should closely mimic real-world trading conditions. To achieve this the backtesting simulation will add transaction cost, variability, and slippage to evaluate how the algorithm performs in a dynamic market. Various performance metrics like the ones mentioned previously are used to asses the algorithms effectiveness. At the same time, a benchmark model will be used to compare and evaluate whether the algorithm can outperform a simple strategy. Creating this sort of tool may seem challenging, but there are many open source solutions that reduce the complexity (*Spörer, 2020*).

2.2 Reinforcement Learning (RL) in Trading

Reinforcement Learning is a Machine Learning sub-field that seeks to mimic the way individuals learn from trial-and-error by employing a self-improving algorithm capable of learning from dynamic environment. It revolves around an agent, an entity tasked with making sequential decisions, and an environment that responds to the agent's actions. The agent learns through feedback in the form of rewards or penalties based on the results from the performed action. This feedback is called the reward function and serves the purpose of calculating how good the given agent's action was in comparison to the outcome. The fundamental concept is based on trial-and-error, where the agent maps the different actions it can take, calculated the expected reward and next state, and adjusts its strategy to maximize the total reward over time. RL is particularly effective in scenarios where explicit programming of a solution is challenging, and the optimal strategy can evolve with changing conditions (*Sutton & Barto, 1998*).

This approach has found success in many applications including algorithmic trading since its main purpose is to adapt and optimize its decision-making process. Therefore, with a properly defined agent, strategy, and reward function an RL algorithm theoretically can adapt to volatile markets like the crypto market. Additionally, it could potentially outperform some of the generalized machine learning methods that are already proven to be effective.

2.2.1 Applied RL in Trading

In the literature, two reinforcement learning algorithms seem the most prevalent when it comes to algotrading. These include SARSA and Q-Learning.The two algorithms (SARSA and Q-Learning) will be important to the research since they have shown promising results (*Comis, 2021*).

Both algorithms are derived from a classification of RL methods called Temporal Difference (TD). TD is a concept that focuses on updating the value function of states or state-value pairs based on the observed and predicted outcomes. The fundamental idea behind TD learning is to adjust the value estimates by considering the difference between predicted and observed values. Essentially, TD learning allows the algorithm to learn each time step and improve their strategy over time. This is particular useful in scenarios where the final outcome and reward is not known and needs to iteratively refine the strategy through ongoing interactions with the environment. Many of the reward functions are based on the Sharpe ratio that was discussed earlier as a common evaluation metric for algotrading. This method has been applied to many financial trading markets including the crypto market (*Corazza et al., 2021*).

Other RL algorithms included in other resources are also worth implementing. For example, many articles mention a more advanced model incorporating both Q-Learning and deep neural networks called DQN (Deep Q-Network) (*Taghian et al., 2022*). One more prominent example uses another RL algorithm called Proximal Policy Optimization (PPO) which was developed in 2017 by OpenAI (*Park et al., 2024*).

2.2.2 Proof of Concept

Although limited due to certain constraints imposed by corporations and regulations, the literature surrounding RL applied to algotrading shows promises in the field. One comparative survey conducted by *Felizardo et al.* in *2022* analyzes results between benchmark models and different RL approaches which provide good insights into their potential. This particular research paper outlines a comprehensive review of the literature, between the years 2016 and 2022, on the application of RL in trading systems. Additionally, the study provides a systematic investigation into the state-of-the-art and an integration of theoretical RL components with the classic trading problem. *Felizardo et al.* highlighted the increasing adoption and advancements of modern RL models such as Actor-Critic and Deep Q-Networks. Their result provide important insights, addresses potential gaps, and gives perspective on the potential uses of RL techniques in financial trading (*Felizardo et al., 2022*). Overall, an important contribution to the RL community inside the trading market.

3. Methodology

This chapter will dive into the intricacies of the algorithmic trading field and the various approaches taken to solve the complexities of the problem at hand. Section 3.1 will define the cryptocurrency market and its unique technicalities that have brought about a surge of attention in recent years. Knowing the differences in features and components that make up the market will be crucial to separate it from the conventional stock market and to adapt certain aspects of algorithmic trading systems. Additionally, an initial exploratory data analysis will be performed to gather statistical insights into the chosen cryptocurrencies. Section 3.2 will cover algorithmic trading techniques that are unrelated to reinforcement learning. Moreover, they will serve an important role as benchmarks that will assist the comparison and validation of reinforcement learning based approaches. Section 3.3 will begin by providing foundational knowledge about reinforcement learning and furthermore define algorithmic trading as a reinforcement learning problem. This problem definition will lead into Section 3.4 and will serve as the basis for reinforcement learning systems to understand the numerous elements of the crypto-trading world. This section will also explain in detail the algorithms and processes behind the application of reinforcement learning to cryptocurrency algorithmic trading.

3.1 Cryptocurrency Market

Cryptocurrencies represent a radical shift in how we define financial investments and securities. Since the introduction of Bitcoin in 2008, cryptocurrencies has developed into a new type of financial asset that intrigues investors with its market volatility and blockchain technology. The sudden attention drawing can also partially be attributed to the growing online activity that has propelled cryptocurrencies into the social media spotlight in recent years. The cryptocurrency market not only encompasses digital currencies but also a vast range of tokens and assets that are derived from blockchain technology. Furthermore, the environment surrounding the crypto-market is characterized by technological innovation, speculation, and significant price volatility. This presents both risks and opportunities for investors and participants. However, despite the risks, there are many cases of governments, organizations, charities, and individuals participating and contributing. The cryptocurrency market continues to evolve with the introduction of derivative products, which provide new mechanisms for risk management and future price movements. Overall, the attention from investors combined with its potential for high returns are increasing the crypto-market's integration into the financial landscape.

3.1.1 Trading Features

There are many features that can be extracted specific to the cryptocurrency market that will need to be considered for crypto-trading. Perhaps the most important aspect, the decentralized nature of blockchain technology changed many market conditions that are unconventional. Unlike the centralized frameworks of tradition markets, where transactions are recorded and processed by specific institutions, cryptocurrency transactions are collectively validated by a network of peers. Not only does this reduces the influence of any central authorities but also means the market is operational on a 24/7 basis allowing for trades at any time. This continuous trading cycle contributes to the market's high volatility and significant price swings. Therefore, this further opens the field for algorithmic trading. High-frequency trading systems are able to continuously operate, obtain more data, and possibly increase returns. However, this also implies that a winning algorithm will need short interval time-series data and significant computing power.

Market volatility, specifically cryptocurrency, is also significantly fueled by various outside factors, some of which include regulatory news, social media, and technological innovation. Moreover, another consequence of high volatility and competition within the cryptocurrency market leads to trading techniques that often combine many variables for a holistic trading decision. One of the methods with promising results is using sentiment analysis on popular social media and news platforms that might affect investors trading decisions. Additionally, this has been observed as a powerful indicator in a multitude of markets and is common in the algorithmic trading community. However, this is out of the scope of the research and instead a there will be a focus on concrete indicators and metrics.

Trading metrics and technical indicators have been widely applied to traditional markets. While some can reliably be applied to the cryptocurrency market, fine-tuning an algorithmic trading system to the cryptocurrency market with cryptocurrency specific metrics may prove beneficial. In a 2024 publication by J.C King et al., the authors propose and validate multiple blockchain indicators and metrics that proved useful to machine learning algorithmic trading systems (*King et al., 2024*). They mention block hash rates, mining difficulty, transaction costs, and blockchain ribbons. In the application of reinforcement learning, only transaction costs will be incorporated into the algorithm. The reason for this is to validate the system with a robust selection of metrics and indicators on multiple cryptocurrencies that may not benefit from certain metrics.

Another difference the cryptocurrency market has with traditional markets is the flow and form of assets. In the conventional stock market, most assets serve purely as investments and pass through centralized nodes. However, in the cryptocurrency market, assets can serve as investments, digital currency, non-fungible tokens (NFTs), and even more with the emerging Web 3.0 technology, each offering different rights and functionalities. This brings uncertainties about about market liquidity and supply that wouldn't be found in traditional markets. With so many variables to account for there needs to be a careful selection to avoid over stimulating the algorithmic trading system with insignificant information.

3.1.2 Cryptocurrencies

In this research, five cryptocurrencies were selected based on their relevance in the market, quality of historic data, and transaction traffic. Each cryptocurrency chosen for the research has different functions and are considered staples in the cryptocurrency market. Therefore, they can be considered a representative sample. Due to accessibility, the cryptocurrency market is highly inflated with ”scam-coins”, ”meme-coins”, and other less active alternatives which are entirely unreliable and serve little purpose in this study. Data for the preliminary pre-processing was obtained from Yahoo Finance and includes 1826 prices from 2019 to 2024. The final selection is the following five cryptocurrencies:

1. **Bitcoin (BTC)**: The first and most widely recognized cryptocurrency, is considered the gold standard of digital currencies. It is often praised for its pioneering blockchain technology and serves as a store of value and investment asset, though it can be susceptible to dramatic price swings due to market sentiment and regulatory news.

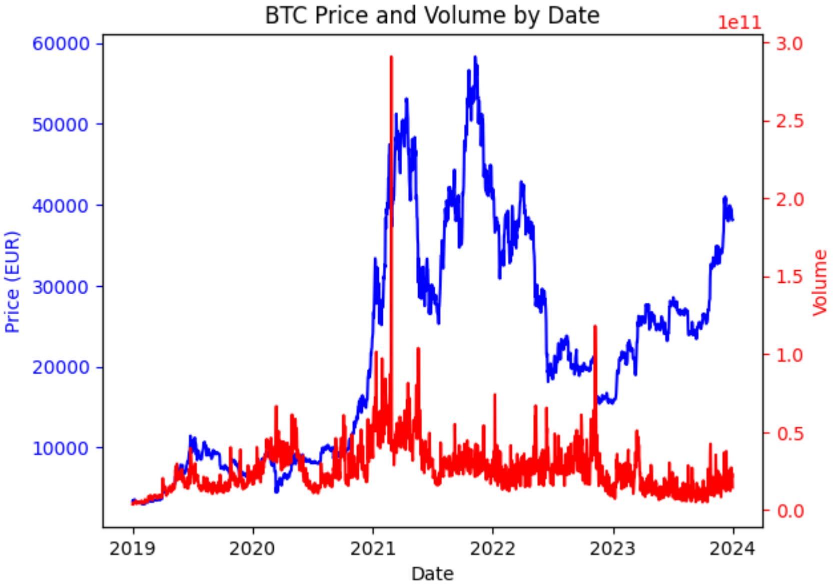


Figure 3.1: Bitcoin prices and volume by date. Source: Python

2. **Ethereum (ETH)**: Notable for its smart contract functionality, which has become a foundational technology for various decentralized applications. Ethereum’s

transition to proof-of-stake in an upgrade known as Ethereum 2.0 aims to address scalability and energy efficiency, potentially affecting its market performance positively by attracting environmentally conscious investors.

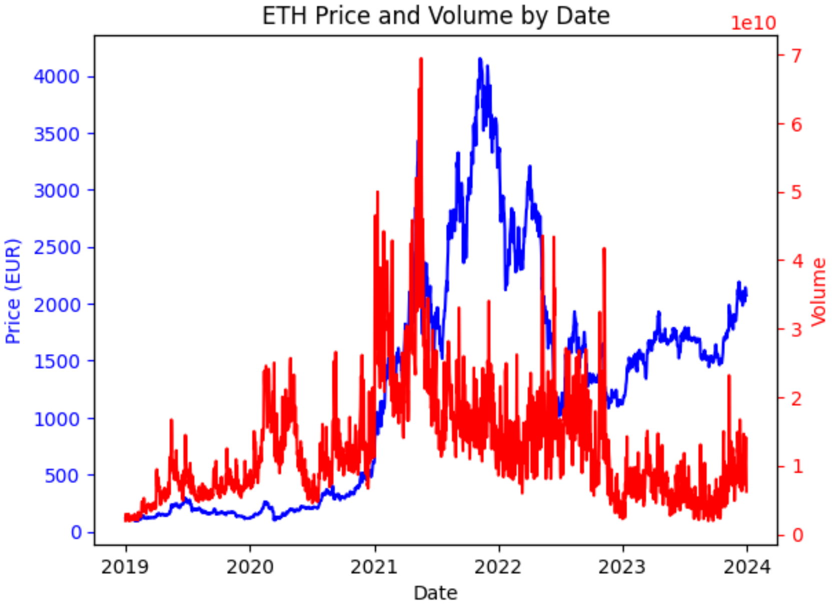


Figure 3.2: Ethereum prices and volume by date. Source: Python

3. **Ripple (XRP)**: Operates distinctly, targeting financial institutions with rapid, low-cost cross-border payment solutions. Ripple's market performance often relies on its utility among financial entities and decentralized technology rather than individual investor speculation. However, many of the common trends and patterns that are prevalent in the cryptocurrency market are still identifiable.

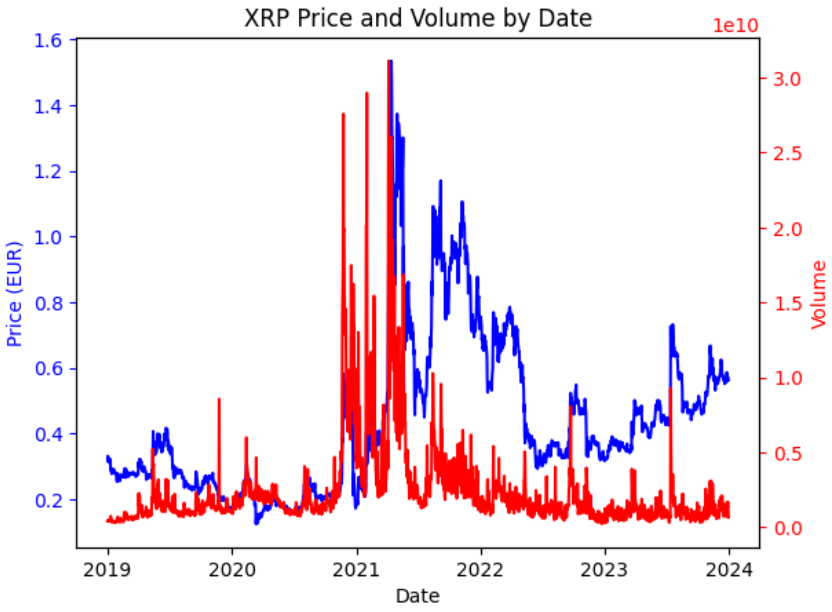


Figure 3.3: Ripple prices and volume by date. Source: Python

4. **Litecoin (LTC)**: Litecoin is often referred to as the ”silver” to the Bitcoin’s ”gold”, which is often considered the monetary standard in the cryptocurrency market. The faster processing times (ledger closing, consensus protocols, etc.) and a larger supply of total tokens makes Litecoin attractive for payment systems and trading. The market trends of Litecoin can be often seen mirroring Bitcoin’s market trends although at lower price points and volume.

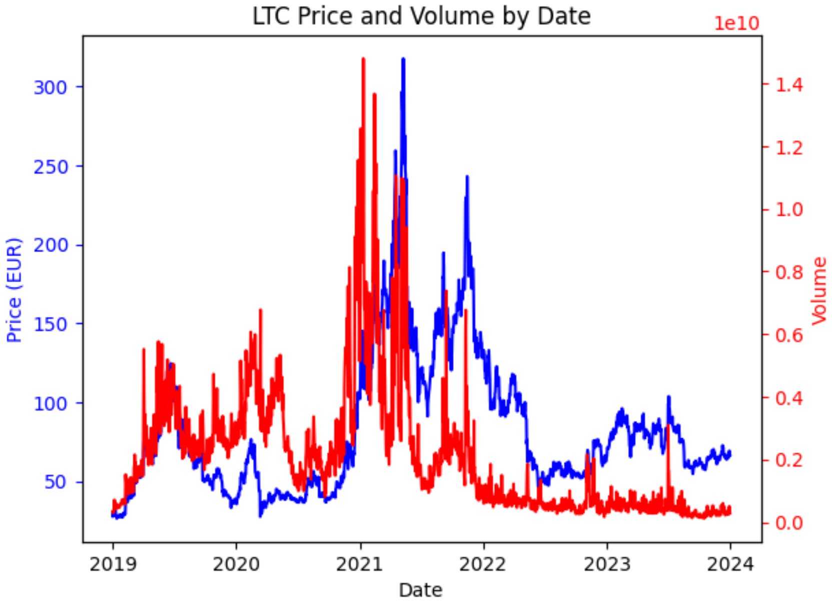


Figure 3.4: Litecoin prices and volume by date. Source: Python

5. **Monero (XMR)**: Monero differs from the rest of the cryptocurrencies and has unique features that cause it to stand out in the market. Monero emphasizes privacy and security features which make it favorable for the blockchain community. It accomplishes this by using advanced cryptography to mask transactions, adding an extra layer to the anonymity of the involved parties. This focus on anonymity appeals to users that care about decentralization and discretion. Even though the data is not representative of the overall market conditions, Monero can be considered representative for the cryptocurrencies with a smaller audience (implying lower volume) that still hold valuable technological applications and investment opportunities. For example, the dark web communities have largely moved over to Monero as a payment method for the previously mentioned reasons.

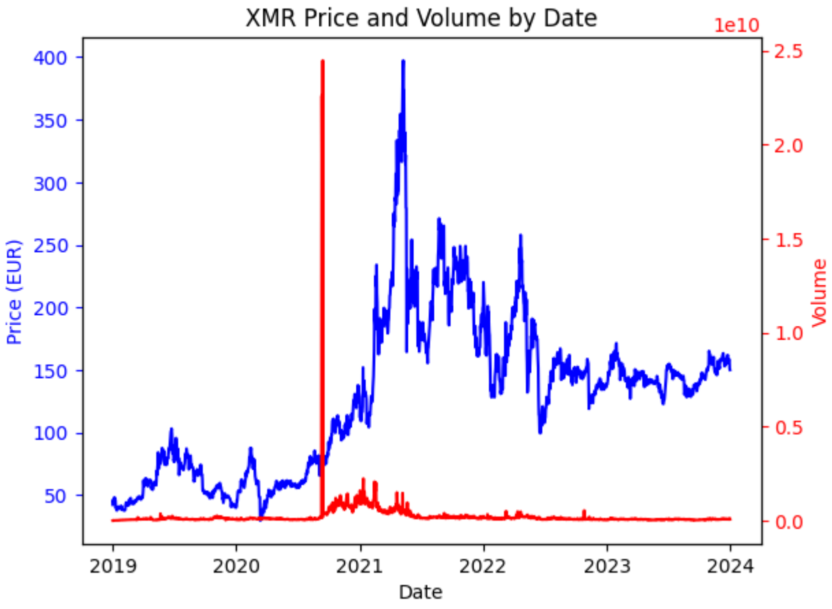


Figure 3.5: Monero prices and volume by date. Source: Python

The overall selection of these five cryptocurrencies was determined based on criteria aimed at covering the different variations of cryptocurrency trading options. Additionally, the popularity in the literature by means of analysis and implementation of these cryptocurrencies to algorithmic trading system show that they are valuable assets. Bitcoin and Ethereum are both considered staples in the market due to their innovation and investment opportunities. Their high-volume/high-price dictate large portions of the market and, although highly volatile, can be considered a stable foundation for the rest of the market. Ripple and Litecoin were both picked due to their technical and financial link to the Ethereum and Bitcoin respectively. Both are built on their predecessors technology and therefore possibly hold price and volume correlations with them. However, they trade on a smaller scale than Ethereum and Bitcoin. Finally, Monero was intentionally chosen as an outlier to the rest as a representative candidate of low-volume cryptocurrencies which make up a large portion of the market. It is important to include a cryptocurrency of this type to show that cryptocurrency algorithmic trading systems can perform even in sub-optimal conditions, meaning its not one of the leading cryptocurrencies which are often classified as the ”best” trading options in this market. This also helps avoid the cryptocurrencies, like pump-and-dump scams or ”meme-coins”, that often negatively impact the market and public opinions. More exploration into the statistical worth, features, and correlations will be explored in the EDA section.

3.1.3 Data Requirements

Cryptocurrency market data comes in the form of time series data which by definition is simply observations taken at different time intervals. Each variable should be time dependent in order to work for the algorithm. The crypto market dataset should have the following structure:

1. Timestamp: Indicates the specific time and date the observation was taken (can vary from seconds to years)

2. Name: Name of the cryptocurrency is provided

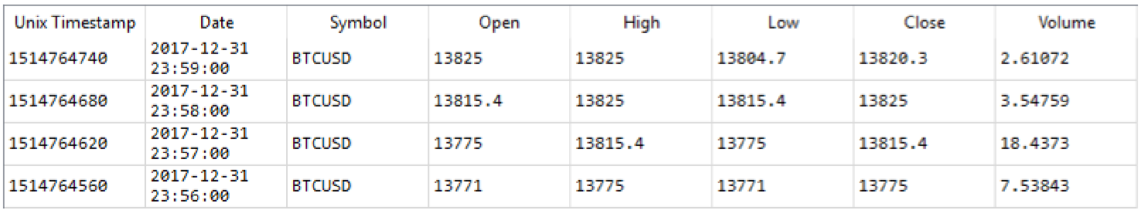
3. Open: Starting price for the given time period

4. Close: Closing price for the given time period

5. High: Highest price reached

6. Low: Lowest price reached

7. Volume: Total volume of executed orders during the period



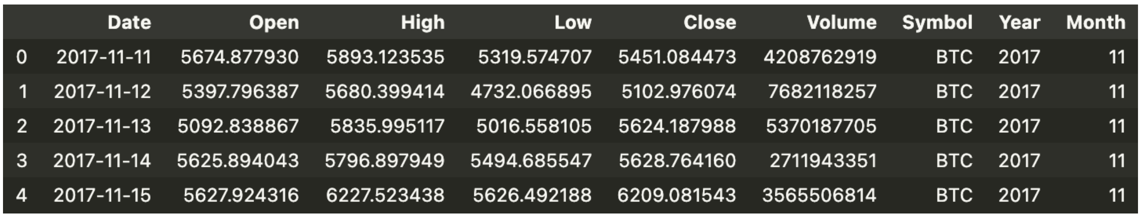
Unix Timestamp	Date	Symbol	Open	High	Low	Close	Volume
1514764740	2017-12-31 23:59:00	BTCUSD	13825	13825	13804.7	13820.3	2.61072
1514764680	2017-12-31 23:58:00	BTCUSD	13815.4	13825	13815.4	13825	3.54759
1514764620	2017-12-31 23:57:00	BTCUSD	13775	13815.4	13775	13815.4	18.4373
1514764560	2017-12-31 23:56:00	BTCUSD	13771	13775	13771	13775	7.53843

Figure 3.6: Example of Bitcoin data set entries

For the initial training of the model, historical time series data of various cryptocurrencies should be obtained. The more historical data obtained the better the results may be (training times also increase), because we are more likely to remove some biases and large variance caused by uncharacteristic trends and large price fluctuations. For example, obtaining data from 2020 when Bitcoin experienced a 300% surge during COVID-19 will largely skew the data and result in a sub-optimal biased model (*Dardouri et al., 2023*). The proposed method for collecting historical data and obtaining real-time data would be connecting to a reliable data source API in order to feed it directly to the model or store it for training (*Zhang et al., 2022*).

3.1.4 Exploratory Data Analysis

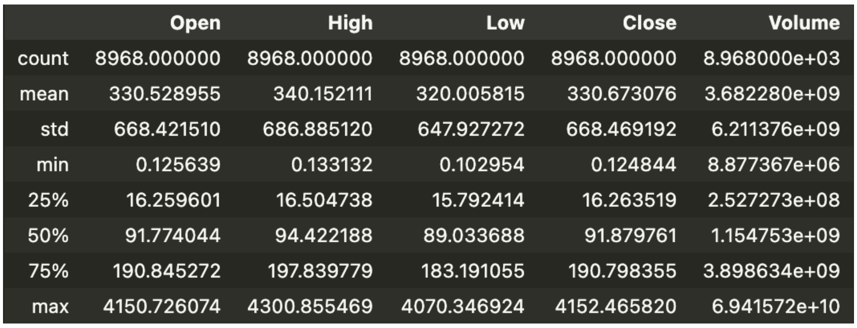
The exploratory data analysis phase seeks to obtain a deeper understanding of the cryptocurrencies and the various feature points of the data. After gathering the data from Yahoo Finance from the five selected cryptocurrencies, an initial data preparation and cleaning was performed to optimize the data set for analysis and training/testing further down the line. This process also allows for finding outliers, patterns, and possible correlations in the data. As mentioned in the previous section, all cryptocurrencies contain OHLCV time-series data with the starting period being 2017-11-11 and the ending period being 2024-01-01. These specific dates were chosen to have the same amount of data records between all the cryptocurrencies. In this case, adjusted price wasn't used and different variables were added for EDA purposes. The final clean and prepared dataset had 2242 records with 9 variables for each cryptocurrency.



	Date	Open	High	Low	Close	Volume	Symbol	Year	Month
0	2017-11-11	5674.877930	5893.123535	5319.574707	5451.084473	4208762919	BTC	2017	11
1	2017-11-12	5397.796387	5680.399414	4732.066895	5102.976074	7682118257	BTC	2017	11
2	2017-11-13	5092.838867	5835.995117	5016.558105	5624.187988	5370187705	BTC	2017	11
3	2017-11-14	5625.894043	5796.897949	5494.685547	5628.764160	2711943351	BTC	2017	11
4	2017-11-15	5627.924316	6227.523438	5626.492188	6209.081543	3565506814	BTC	2017	11

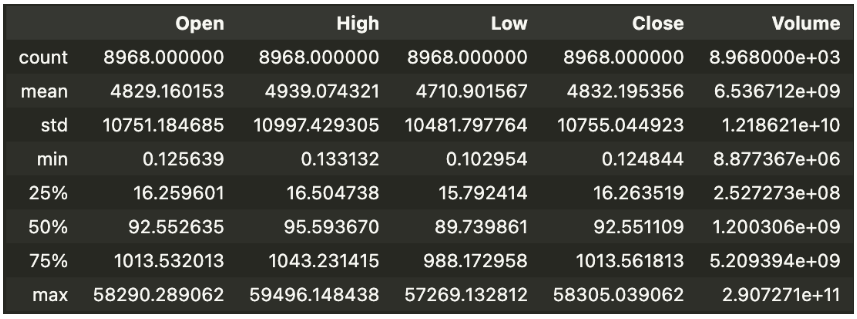
Figure 3.7: Example of cleaned data set

Data exploration confirmed the high volatility with a significant standard deviation. Across all the cryptocurrencies, outliers were discovered that largely skewed certain periods of the cryptocurrency market prices and volume.



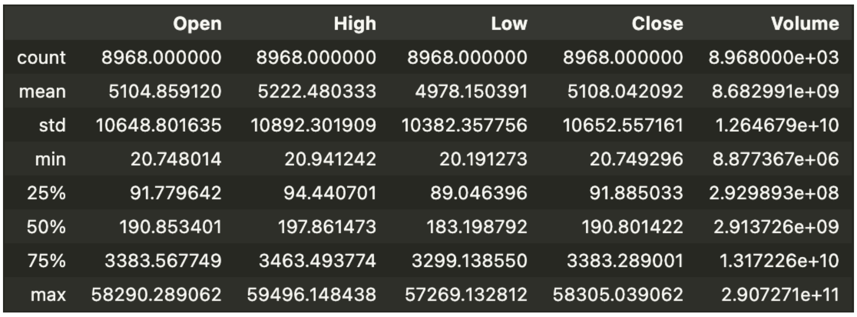
	Open	High	Low	Close	Volume
count	8968.000000	8968.000000	8968.000000	8968.000000	8.968000e+03
mean	330.528955	340.152111	320.005815	330.673076	3.682280e+09
std	668.421510	686.885120	647.927272	668.469192	6.211376e+09
min	0.125639	0.133132	0.102954	0.124844	8.877367e+06
25%	16.259601	16.504738	15.792414	16.263519	2.527273e+08
50%	91.774044	94.422188	89.033688	91.879761	1.154753e+09
75%	190.845272	197.839779	183.191055	190.798355	3.898634e+09
max	4150.726074	4300.855469	4070.346924	4152.465820	6.941572e+10

Figure 3.8: BTC data statistics



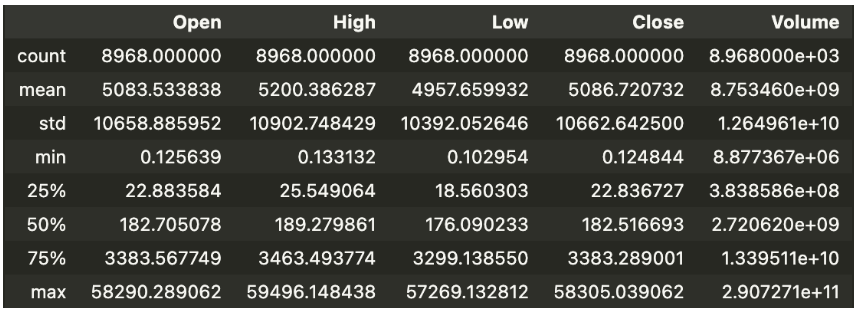
	Open	High	Low	Close	Volume
count	8968.000000	8968.000000	8968.000000	8968.000000	8.968000e+03
mean	4829.160153	4939.074321	4710.901567	4832.195356	6.536712e+09
std	10751.184685	10997.429305	10481.797764	10755.044923	1.218621e+10
min	0.125639	0.133132	0.102954	0.124844	8.877367e+06
25%	16.259601	16.504738	15.792414	16.263519	2.527273e+08
50%	92.552635	95.593670	89.739861	92.551109	1.200306e+09
75%	1013.532013	1043.231415	988.172958	1013.561813	5.209394e+09
max	58290.289062	59496.148438	57269.132812	58305.039062	2.907271e+11

Figure 3.9: ETH data statistics



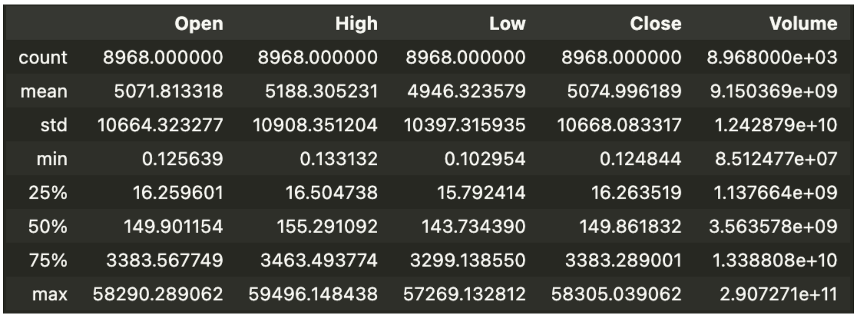
	Open	High	Low	Close	Volume
count	8968.000000	8968.000000	8968.000000	8968.000000	8.968000e+03
mean	5104.859120	5222.480333	4978.150391	5108.042092	8.682991e+09
std	10648.801635	10892.301909	10382.357756	10652.557161	1.264679e+10
min	20.748014	20.941242	20.191273	20.749296	8.877367e+06
25%	91.779642	94.440701	89.046396	91.885033	2.929893e+08
50%	190.853401	197.861473	183.198792	190.801422	2.913726e+09
75%	3383.567749	3463.493774	3299.138550	3383.289001	1.317226e+10
max	58290.289062	59496.148438	57269.132812	58305.039062	2.907271e+11

Figure 3.10: XRP data statistics



	Open	High	Low	Close	Volume
count	8968.000000	8968.000000	8968.000000	8968.000000	8.968000e+03
mean	5083.533838	5200.386287	4957.659932	5086.720732	8.753460e+09
std	10658.885952	10902.748429	10392.052646	10662.642500	1.264961e+10
min	0.125639	0.133132	0.102954	0.124844	8.877367e+06
25%	22.883584	25.549064	18.560303	22.836727	3.838586e+08
50%	182.705078	189.279861	176.090233	182.516693	2.720620e+09
75%	3383.567749	3463.493774	3299.138550	3383.289001	1.339511e+10
max	58290.289062	59496.148438	57269.132812	58305.039062	2.907271e+11

Figure 3.11: LTC data statistics



	Open	High	Low	Close	Volume
count	8968.000000	8968.000000	8968.000000	8968.000000	8.968000e+03
mean	5071.813318	5188.305231	4946.323579	5074.996189	9.150369e+09
std	10664.323277	10908.351204	10397.315935	10668.083317	1.242879e+10
min	0.125639	0.133132	0.102954	0.124844	8.512477e+07
25%	16.259601	16.504738	15.792414	16.263519	1.137664e+09
50%	149.901154	155.291092	143.734390	149.861832	3.563578e+09
75%	3383.567749	3463.493774	3299.138550	3383.289001	1.338808e+10
max	58290.289062	59496.148438	57269.132812	58305.039062	2.907271e+11

Figure 3.12: XMR data statistics

The EDA further confirmed the previous idea that Bitcoin, Ethereum, and most of the other cryptocurrencies are highly correlated in their price fluctuations. Over multiple periods the five cryptocurrencies all experienced similar rising spikes in volatility and inflation and decreased in the same short intervals. The biggest spikes in both price and volume occurred in 2020, which confirms the claims explained in the literature. information.

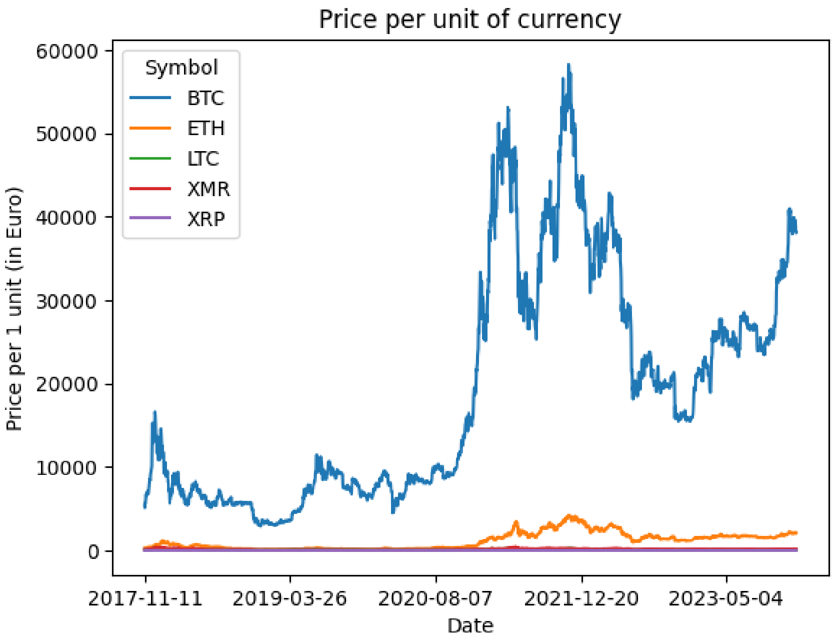


Figure 3.13: Chosen Cryptocurrencies price per unit trend

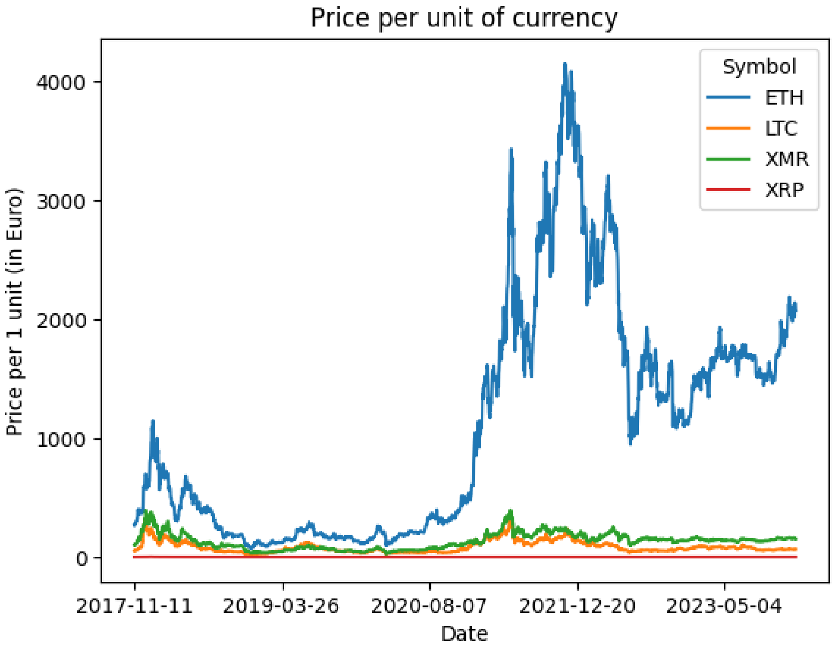


Figure 3.14: ETH, XRP, LTC, XMR price per unit trend

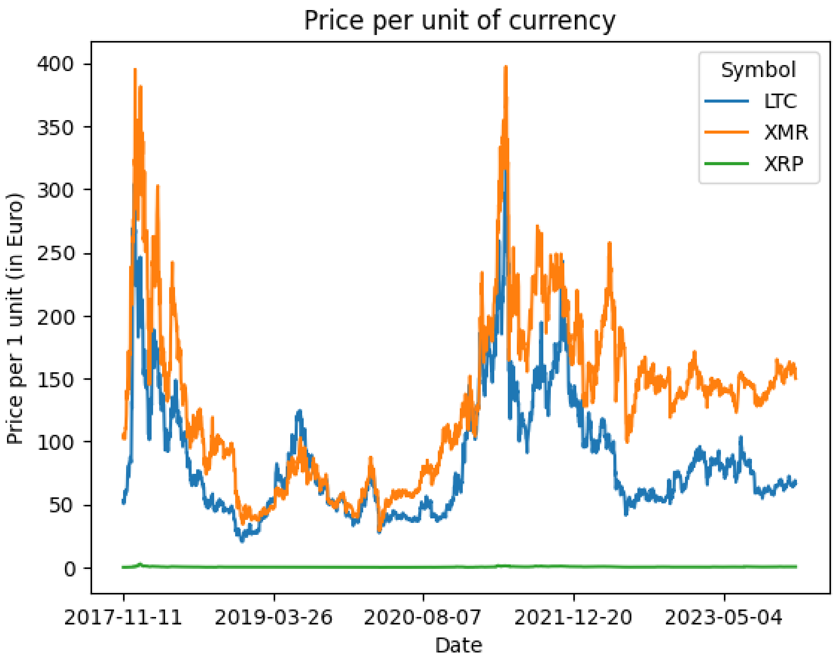


Figure 3.15: XRP, LTC, XMR price per unit trend

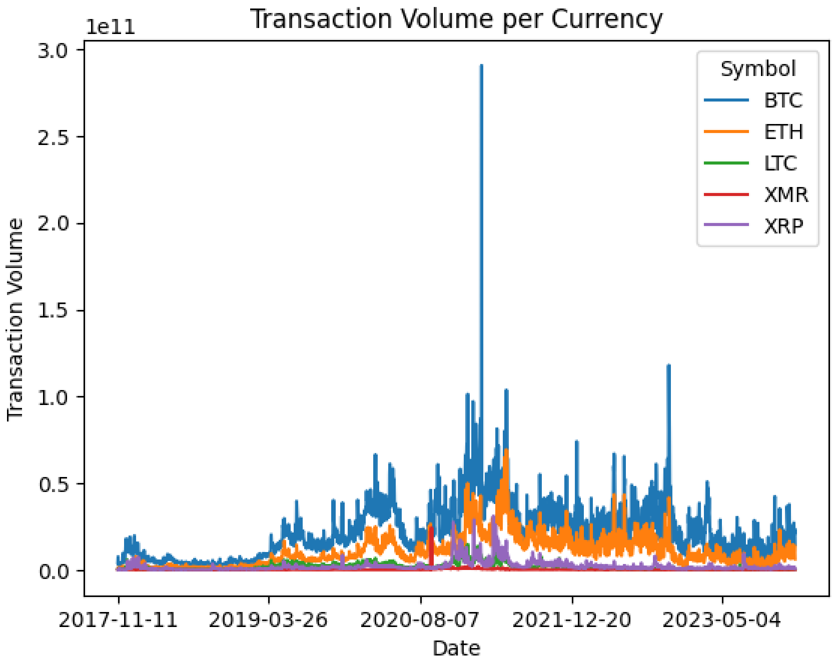


Figure 3.16: Chosen Cryptocurrencies volume trend

Analyzing the trends reveals that Bitcoin and Ethereum had the largest increases in price while Ripple had the lowest. Additionally, Bitcoin experienced massive spikes in volume compared to the rest during certain periods while Monero experienced the least. However, except for Monero, the other cryptocurrencies follow Bitcoin in its volume fluctuations. This information has been confirmed by the literature and economics metrics before, but it is important to confirm that the data also accepts or rejects this

3.2 Benchmarks

Algorithmic trading benchmarks have in important role as a tool for assessing and comparing the effectiveness of algorithmic trading systems. The benchmarks provide a standard for analyzing newly developed and experimental trading strategies. For a comprehensive evaluation of reinforcement learning trading systems, multiple strategies and algorithms may be used.

The first benchmark should involve traditional technical analysis strategies. Implementing these traditional methods as a benchmark allows for comparison of how well the reinforcement learning system performs against time-tested conventional trading strategies. These strategies have been widely used and rigorously testing over time and across various markets. This comparison can reveal whether the reinforcement learning approaches offer significant improvements in predictive accuracy or profitability under similar market conditions.

The second benchmark consists of the best performing machine learning models that have been adapted for cryptocurrency trading. With the added computational power of machine learning over technical analysis, it will provide valuable insights into the capabilities of advanced computer techniques. Additionally, it poses as a powerful opponent to reinforcement learning systems especially since most research has gone into machine learning algorithmic trading and its variations.

Both benchmarks will be implemented in a way that supports their role as validators instead of a detailed case study of themselves. The focus will be on using these benchmarks to validate the effectiveness of the reinforcement learning strategies under real-world trading scenarios. This approach ensures that the primary analysis remains on the reinforcement learning models, while still providing a clear and effective comparison to established methods.

3.2.1 Technical Analysis

Technical Analysis is still a popular method for financial trading despite the advancements in algorithmic trading systems. By using historic market data, including mainly price and volume, technical analysis can be used to find trends and predicts price movements. Analysts use technical analysis to generate trading signals on a short-term basis and may therefore obtain a slight advantage over the market.

Researchers over time have developed technical indicators that support technical analysis trading. There are many indicators that each focus on different aspects of trading. These include price trends, and chart patterns, moving averages, and many more. Other cryptocurrency specific indicators are like the ones mentioned in Section 3.1 (hash rates, mining speed, etc.). Technical analysis adapted for the cryptocurrency market has been tested and validated before and has shown promising results (*Resta et al., 2020*).

Strategy Selection

Three technical analysis strategies should be selected for the benchmark. The indicators included moving average convergence divergence (MACD), relative strength index (RSI), average directional index (ADX), exponential moving averages (EMA), and directional indicators (DI). Mathematical formulas are included in Appendix A and an overview of each strategy is included in below.

Moving average convergence (MACD) crossover generates a buy signal when the difference between two exponential moving averages (EMA) crosses above the signal line and generates a sell signal when it crosses below. This line is called the MACD line and a crossover of two lines shows a possible reversal in the trend.

Average directional index (ADX) and Directional Indicator (DI) Crossover is the second trading method. Without taking into account the trend's direction, the ADX indicator is used to evaluate a trend's strength. Generally speaking, an ADX score above 25 implies a strong trend and a value below 20 a weak trend. Two directional indications, $DI+$ and $DI-$, accompany ADX in the ADX and DI crossing. Buying may be advised when the ADX is above 20 and the $DI+$ crosses above the $DI-$, both of which point to a strengthening uptrend. When the ADX stays above 20 and the $DI-$ crosses over the $DI+$, a sell signal is issued, signifying a strengthening a downtrend.

Relative strength index (RSI) can be used to look for overbought and oversold levels. It accomplishes this by measuring the rate of change of price movements on a scale of 0 to 100. Traditionally, when the RSI values are over 70 an asset is implied to be overbought and selling should be considered. One the other hand, when the RSI values are under 30 an asset is implied to be oversold and a price increase may occur. Traders use these levels as an indicator of whether they should enter or exit the market.

Implementation

With the focus of the research being reinforcement learning, pre-made tools and software can be utilized for creating the technical analysis benchmark. Open-source software in the programming language Python can be used for implementing automated trading system to test and analyze strategies. Freqtrade provides the necessary framework for optimizing trading strategies, backtesting, and deployment of trading systems.

3.2.2 Machine Learning

In recent years, machine learning has emerged as a powerful force in the realm of algorithmic trading, disrupting the financial sector with robust models capable of trading at a level beyond what was thought possible within the financial sector. As financial corporations seek methods to gain a competitive edge in the digital marketplace, machine learning is the leading technology, enhancing trading strategies by processing and learning from massive amounts of historical data. This allows machine learning algorithms to rapidly adapt to market conditions and successfully operate.

The most common application of machine learning in the algorithmic trading field is price prediction and risk management. Other uses include sentiment analysis of media surrounding the market, market research, and even a combination of many in what is called a multi-model approach. While these approaches show promising results, they are not included in the scope of this research since it would add many external factors that might not be representative of the underlying technology.

It is important to note that although the literature on machine learning algorithmic trading is extensive, a standard framework or technique has not clearly been defined yet. Financial institutions developing algorithmic trading systems are actually incentivized to not share their information as it could dull their competitive edge over the market. This challenge, along with other difficulties, could be the cause for conflicting results and information among the research.

Model Selection

One type of machine learning model that is prevalent in the field of algorithmic trading is the long short-term memory (LSTM). LSTMs networks are a specialized type of recurrent neural networks (RNN) with a unique architecture that enables them to effectively capture long-term dependencies in sequential data while ignoring information it deems irrelevant. Therefore, making them highly suitable for tasks such as time-series predictions and more specifically, financial market predictions.

At the core of an LSTM network are the memory cells. Their purpose is to maintain information in memory for extended periods. Within each cell, three components called gates regulate the flow of information.

1. Forget Gate: Decide if information should be discarded from the cell state. It accomplishes this by looking at the previous state and current input, then passes it through a sigmoid function which outputs a number between 0 and 1. A 0 and 1 means forget and remember information respectively.

2. Input Gate: Decide what information will be stored in the cell state. It contains a sigmoid layer that is capable of deciding what values to update and a

tanh layer that vectorizes new candidate values.

3. Output Gate: Decide what the next hidden state is, which holds the information of the previous inputs. This information is is regulated by a sigmoid function before it is used.

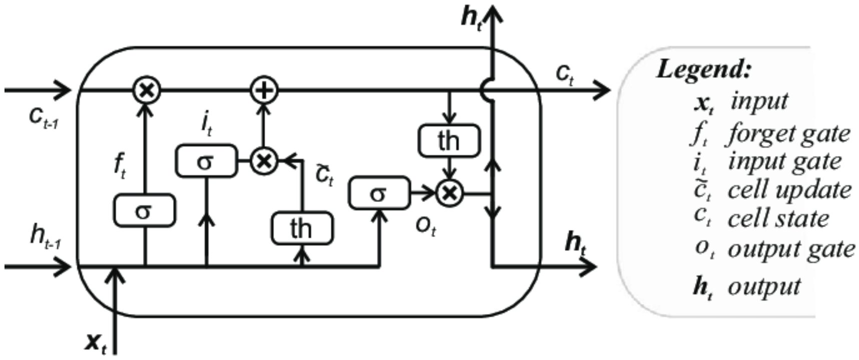


Figure 3.17: LSTM cell with its internal structure. Source: Hrnjica and Bonacci, 2019

Due to LSTMs ability to recall and forget information based on their importance, they make a good candidate for markets that experience highly volatile "hype" periods like the cryptocurrency market. In general, LSTMs have a powerful architecture that excels at handling sequential data with temporal dependencies, making them an effective tool for financial problems.

Implementation

By using open-source software, a simple implementation of an LSTM model for generating trading signals may be used. There are many approaches to this method. For example, an LSTM model can be used for predicting trends and generating trade signals and can be applied with trading strategies (*Botunac et al., 2024*). An assortment of open-source tools and software can be used for data preparation, implementation, and training. These include Sklearn, Tensorflow, data processing tools and more. Another tool for backtesting, provided by Spörer in his research, called quantbacktest is a specialized tool for testing algorithmic trading strategies applied to cryptocurrencies (*Spörer, 2020*).

3.3 Algorithmic Trading in Reinforcement Learning

Reinforcement learning is inspired by how humans learn via trial-and-error in their daily lives. RL is capable of adapting to a variety of situations by using algorithms that are founded on this idea of "learn by experience". RL employs principles of psychological conditioning, situating the algorithm within an environment where it interacts with an interpreter and a reward system. The results of each algorithm iteration are evaluated by the interpreter to determine their desirability. If the algorithm produces a satisfactory outcome, it is positively reinforced through rewards. On the other hand, if the outcome is unfavorable, the algorithm iteratively adjusts its approach until improvements are achieved. Typically, the reward system is scaled to the effectiveness of the outcome. Thus, the algorithm is incentivized to pursue the best possible solution in exchange for the maximum reward.

3.3.1 Reinforcement Learning Concepts

This methodology shows promise for applications in algorithmic trading due to the dynamic environment and and strict parameters. Rewards for RL applied to trading, in some cases, can be as simple as returns on investment. Other examples include financial metrics such as Sharpe or Calmar ratio incorporated into the reward system (*Peng et al., 2022*). There has been many attempts at defining a framework for RL in this field, but it is still in its development phase and can be considered behind machine learning application (*Felizardo et al., 2022*).The following section will cover reinforcement learning concepts and how they can be adapted for cryptocurrency algorithmic trading. Much of the background knowledge on RL covered is directly inspired from the book "*Reinforcement Learning: An Introduction*" written by *Sutton* and *Barto*.

Agent-Environment Model

In reinforcement learning, the agent-environment model is foundational. It describes the interaction between an agent, an autonomous interaction system, and the environment, which occurs in discrete time steps. At each time step t, the agent receives a state s_t from the environment. The agent then chooses an action a based on the policy π. The policy is a function that maps states to actions. In response, the environment provides the next state s_{t+1} and a reward r_t. This agent-environment interaction is guided by the dynamics of the environment itself. Typically, formalized as a Markov Decision Process (MDP).

Markov Decision Process is a mathematical framework used for modeling decision-making in situation where the next state is partly unknown and may be affected by the decision maker (agent). MDPs are therefore used frequently in RL to provide a formal definition of the agent-environment model. An MDP is characterized by the following components:

1. States S: Set of states that represent all the possible situations that might occur in the environment. In single agent problem, an agent can only be in one state at a given time.

2. Actions A: Given each state, there are actions available to the agent that may potentially affect the state. Actions are decisions taken by the agent and transition it to another state.

3. Transition Probability P: Function that calculates the probability of moving from one state to another state given an action. The following function represents the probability of transitioning to state s_{t+1} from state s_t after taking action a: $P(s_{t+1}|s_t, a_t)$

4. Rewards R: Reward function that assigns a numerical value to each transition between states given the specific actions. The following function typically denotes the reward obtained after transitioning state s_t from state s_{t+1} after taking action a: $R(s, a, s')$

5. Discount Factor γ: Factor between 0-1 used to discount from future rewards as compared to immediate rewards. This reflects the importance of sooner rewards rather than later. It helps manage the trade-off between immediate and future rewards while also ensuring the cumulative reward does not increase indefinitely.

The overall goal of an MDP is for the agent to find a policy π that maximizes the expected cumulative reward r over time t. This is often expressed as a value function ($V(s)$) or an action-value function ($Q(s,a)$) for state-action pairs. For an agent to achieve the highest long-term rewards, an optimal policy will try to maximize these functions.

Optimal Policy Estimation & Value Function

In order to estimate the optimal policy, the value functions are critical for evaluating how good it is to be in a given state or how good it is to perform a certain action given the state.

1. State-Value Function: This function gives the expected return when stating in state s_t and then following policy π:
   $V_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$

2. Action-Value Function: This function gives the expected return starting at state s_t, taking an action a, and then following policy π:
   $Q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$

3. Bellman Equations: The previous two functions can also be written as Bellman equations. In this form they provide recursive relationships that are essential for computational methods. The equations can be found in the Appendix.

Exploration v.s Exploitation Trade-off

In RL, there is a fundamental problem based on the trade-off between exploration and exploitation. Exploration is the agent's ability to discover more information about its surrounding environment. Exploitation is the agent's ability to use the information its already knows in order to maximize it's rewards. A sub-optimal policy due to undiscovered actions may occur when exploration is low. On the other hand, reward maximization may not occur when exploitation is low. By tweaking this parameter ϵ between exploration and exploitation, given the unique agent-environment model, the agent is capable of improving its performance and reaching a near optimal policy. Luckily, there exists techniques like ϵ-greedy where the agent chooses a random action with probability ϵ and the best-known action with probability $1 - \epsilon$. Other more advanced techniques like Upper Confidence Bound or Thompson Sampling dynamically adjust the level of exploration based on the uncertainty of the action-value estimation. By employing these kinds of methods to balance the trade-off, the agent can learn an optimal policy that maximizes its rewards.

3.3.2 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) is an advanced deep learning technique that incorporates deep neural networks into the decision-making framework of reinforcement learning. With the use of DRL, agents are more capable of learning optimal policies when dealing with high-dimensional action-states spaces and complex environments. By using the efficiency of deep neural networks to estimate policy and value functions problems involving continuous spaces and large dimensions. Therefore, DRL offers significant potential in the field of algorithmic trading. Financial markets are characterized by non-linear relationship and noisy data making a complex environment with a massive state-action space. Additionally, the cryptocurrency market can be considered more complex due to its 24/7 open market, unique trends, and high volatility. Section 3.1. Despite the many challenges, DRL may be capable of discovering profitable trading strategies. Furthermore, an optimized DRL trading strategy suggest high returns with minimized costs and real-time portfolio management. The continuously learning ability of RL systems make it a valuable tool for enhancing the performance of automated trading systems. The main application of DRL for algorithmic trading will be covered in Section 3.4.2.

3.4 Applied RL for Cryptocurrency Algorithmic Trading

In order to apply reinforcement learning to algorithmic trading, algorithmic trading needs to be formulated into the proper format. In essence, algorithmic trading is a complex sequential decision-making problem. It provides time-series data, has a continuous environment at set time steps, and clearly outlines an observation and action space. The challenge involved in with this RL application is the design of a trading policy that can compete with other algorithmic trading strategies. Particularly, the strategies mentioned previously in the research.

3.4.1 Problem Formalization

The setup of the components of algorithmic trading with reinforcement learning are crucial for the agent-environment model to function correctly. The literature revealed a multitudes of approaches to this problem (*Felizardo et al., 2022*). Many problem formalization over complicate themselves which is not beneficial for many technical applications due to the exponential increases in dimentionality. For this reason, the formalization provided by *Théate* in *2020* was used and then further adapted to adjust for the characteristics of the cryptocurrency market (*Théate & Ernst, 2021*). This will be described in detail in the following section.

Defining Components

In the scope of this research, each cryptocurrency will be considered independently by the trading system. Meaning, the same trading system will need to be trained multiple times independently on the data of each cryptocurrency. A portfolio value v_t is defined, in this case containing a single cryptocurrency, and is composed of the agents cash value v_t^c and the share value v_t^s, where t is time.

The trading agent will interface with the cryptocurrency market through an order book. The order book is the set of all buy and sell actions the agent performs. Each order will consist of a side s (buy or sell), quantity q, and price p. Essentially, the agent in the system will perform actions that include quantity q given the time-step and the actions are then recorded in the order book for exchanging with the market.

For a trade to be executed, the maximum sell price needs to be greater than or equal to the maximum buy price ($p_{max}^{buy} \geq p_{max}^{sell}$). The time interval t on which the environment will operate will be defined by Δt. Therefore, a trade can only occur every $1/\Delta t$. This needs to be managed to fit with the data time intervals and the technical capabilities need to be considered.

The generalized trading strategy of the agent will first update its state with the new market information i_t. Second, the policy will output an action given the new market information and the action. The trade will then be executed and the process can start again. For a more complete definition involving reinforcement learning, terms related to the agent and environment need to be defined. The RL agent, at time t will initially retrieve an observation o_t from its environment. This will become the new internal state of the environment s_t. Afterwards, the policy $\pi(a_t|h_t)$ determines the action a_t that the agent performs. h_t refers to the agent's history, which includes the past observations, actions, and rewards. The mathematical representation of the history is the following: $h_t = \{(o_\tau, a_\tau, r_\tau)|\tau = 0, 1, ..., t\}$. The history component is crucial for the agent to learn from its past and identify trends more efficiently.

The criterion for obtaining the optimal policy π^*, in this case, is the expected discounted sum of rewards. The mathematical representation is the following:

$$\pi^* = \underset{\pi}{argmax} \mathbb{E}[R \mid \pi]$$

$$R = \sum_{t=0}^{\infty} \gamma^t r_t$$

Observation Space

As mentioned in previous sections, the features on which a trading system can operate are vast, but limited at the same time. Traditional datasets used for algorithmic trading include OHLCV (open, high, low, close, volume) data and while this is essential for most trading systems, it leaves room for improvement with more indicators. However, expressing such indicators in a quantitative format while maintaining a low-dimensionality is a major challenge. Ideally, cryptocurrency specific metrics would be added to the observation space (*King et al., 2024*).

The agent begins by collecting information contained in the internal state ($s_t \in S$) trading environment. The observation space ($o_t \in O$) contains all the relevant information the agent will need. The more information the observation contains it is likely the agent would perform better. A reduced implementation of an observation at time t can be represented as follows: $o_t = \{\{p_{t'}^O, p_{t'}^H, p_{t'}^L, p_{t'}^C, V_{t'}\}_{t'=t-\tau}^t, P_t\}$. The total observation includes the OHLCV data taking into account the history and the state information. P_t being the trading position of the agent.

Action Space

As mentioned previously, the agent executes an action $a_t \in A$ based on the current policy $\pi(a_t|h_t)$. The agent not only needs to determine whether to buy, sell, or hold, but also needs to determine the quantity of the trade. This quantity will be represented by Q_t. With this in place there are three conditions for Q_t. If Q_t is greater than 0 the agent will buy. Else if Q_t is less than 0 the agent will sell. Finally, if Q_t is equal to 0 the agent will hold.

The actions taken by the agent will affect the cash and share values we had previously defined. Therefore, they will be updated using the following functions, with n_t being the amount of cryptocurrency owned by the agent:

$$v_{t+1}^{c} = v_{t}^{c} - Q_t p_t$$

$$v_{t+1}^{s} = \underbrace{(n_t + Q_t)}_{n_{t+1}} p_{t+1}$$

A major constraint that needs to be accounted for is the costs associated with trading. In a simulation scenario, an algorithmic trading system could be profitable. However, when connected to an exchange and the system is deployed it might now perform as expected. This is due to two factors, explicit and implicit costs, that were defined by *Théate*. Explicit costs include transaction fees, taxes, exchange rates, and more that can be easily calculated and accounted for by the algorithm. Implicit costs include spread costs, market impact costs, and timing costs caused by latency in the system (*Théate & Ernst, 2021*).

These additional costs are mitigated by integrating a constant C heuristic in the form of a percentage into the cost value $v_t^c$and action set A. This value needs to be tested with varying amount to find the best fitting for the cryptocurrency market. Additionally, the action space needs to be further reduced in order to reduce time complexity and high-dimensionality. The following is the simplified version of the action space with only two actions: $a_t = Q_t \in \{Q_t^{long}, Q_t^{short}\}$

A clear explanation of the final reduced action space and definitions can be found in the publication "*An Application of Deep Reinforcement Learning to Algorithmic Trading*".

Rewards

For a simple yet effective reward system, returns on investment can be used to incentivize the system to increase its profitability. This can be modeled as followed: $r_t = \frac{v_{t+1} - v_t}{v_t}$. Ideally, the reward system should include risk as a factor in order for the agent to learn how to mitigate risk. Metrics such as the previously mentioned Sharpe and Calmar ratio are good indicators of risk. They have both shown to be valuable to trading systems for both performance and testing. It is also important to mention that reward systems can not always be generalized to all models. This is due to different models having different characteristics that might need fine-tuning like discussed in Section 3.3.

3.4.2 Applied Algorithm

In the literature, there are many examples of reinforcement learning algorithms applied to algorithmic trading. Among these, algorithms related to the sub-field Temporal Difference (TD) Learning have shown prevalence. For example, there is state-action-reward-state-action (SARSA) and Q-Learning. TD learning is a method that iteratively updates the value functions based on discrepancies between predicted and observed outcomes. This approach is promising when applied to financial market trading. Furthermore, the exploration of other more advanced deep reinforcement learning strategies, such as Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO), introduces additional dimensions to algorithmic trading, promising even more sophisticated solutions in fields as diverse as traditional finance and the burgeoning crypto market.

This research explores the implementation of a specialized DQN tailored for algorithmic trading. The benefits of employing such algorithms is the enhanced efficiency that comes with deep reinforcement learning and formalized environments that allow the agent to perform effectively. This implementation is inspired by the work of *Théate*, because the results showed the most promise from the reviewed literature. The publication came with open-source software for simple testing of their system.

Trading Deep Q-Network (TDQN)

The Trading Deep Q-Network (TDQN) algorithm is an approach tailored for algorithmic trading, adapted from the Deep Q-Network (DQN) framework. DQN, which itself extends the traditional Q-learning methodology, employs deep neural networks (DNNs) to estimate the state-action value function (Q-function). This estimation involves learning the parameters of the DNN to optimally predict the expected rewards of taking certain actions in given states. For the sake of context within the following explanation, a ”trajectory” refers to a sequence of states, actions, and rewards experienced by a reinforcement learning agent over a series of time steps within the simulated trading environment. Essentially, it represents a specific path that the agent takes through its decision space from the beginning to the end of an episode, influenced by the policy it follows. This is crucial for learning optimal trading strategies in the stock market context the paper addresses.

TDQN's training is primarily grounded in the generation of artificial trajectories from historical market data (specifically daily OHLCV data), avoiding the need for a complete environment model. Each trajectory is represented as follows:

$$\tau = \left(\left\{o_0, a_0, r_0\right\}, \left\{o_1, a_1, r_1\right\}, \ldots, \left\{o_T, a_T, r_T\right\}\right)$$

This is a sequence of observations (o_T), actions (a_T), and rewards (r_T) from an RL agent over a number T of trading time steps. These trajectories simulate the market interactions, assuming the market remains uninfluenced by the agent's actions due to their minimal market impact.

For trading adaptations, TDQN modifies the original DQN's convolutional neural network (CNN) architecture to a more suitable feed forward DNN with Leaky ReLU activation functions, addressing the distinct nature of time-series input data. Furthermore, enhancements such as the Double DQN are implemented to mitigate issues with overestimation in value predictions. This is achieved by decomposing the target max operation into separate action selection and evaluation processes:

$$Q(s, a) = r + \gamma Q\left(s', \underset{a'}{\operatorname{argmax}} Q(s', a', w), w^{-}\right)$$

To improve exploration, TDQN introduces a technique wherein each selected action and its opposite are executed in the actual and a copied environment, respectively. This not only aids in balancing the exploration-exploitation trade-off but also does so with minimal additional computational cost.

Due to the complexity of the training data and the volatility of cryptocurrency market data in general, the DNN was adapted to meet certain requirements for stability and the convergence on an optimal policy. The Adaptive Moment Estimation (ADAM) optimizer was employed which known for its efficiency and effectiveness across a wide range of applications. ADAM is able to maintain EMA of past gradients which helps to stabilize updates and therefore converges faster (*Duchi et al., 2011*). It combines AdaGrad and RMSProp which are two optimization methods that adjust learning rates for each parameter based on gradient magnitudes. Secondly, gradient clipping is used to manage the high volatility and potential erratic behaviors observed in financial market data (*Goodfellow et al., 2016*). Additionally, gradient clipping is particularly crucial due to sudden large movements or "shocks" in market data can lead to unusually large gradients. By applying gradient clipping, the TDQN ensures that each training step makes a bounded update, thus protecting the network against instability induced by these abrupt changes in the input data.

Finally, in the context of this TDQN batch normalization is employed slightly differently from its standard use (*Ioffe & Szegedy, 2015*). While it typically aims to reduce internal covariate shift in most deep learning tasks, in this TDQN, its use is more critically focused on handling the peculiarities of financial time-series data, which can be noisy and highly volatile. The normalization process helps the network adjust to the erratic nature of market data, promoting a faster and more robust training phase. It aids in generalization which is an important aspect when dealing with financial datasets that may not display consistent patterns. This specialized application in TDQN highlights the adaptability of batch normalization to different domains, emphasizing its role in enhancing learning stability and improving model performance under varied data conditions.

The implementation of robust regularization strategies in the TDQN model is instrumental in combating overfitting, a common challenge with deep neural networks, especially in fluctuating markets like those of cryptocurrencies. Methods such as Dropout, L2 regularization, and Early Stopping have been integrated based on insights from initial experiments (*Hinton, 2012*). Dropout randomly disables a portion of the neurons during each training iteration, preventing any single neuron from becoming overly influential on the outcome. This randomness helps ensure that the network remains robust to variations in input data. L2 regularization, on the other hand, adds a penalty on the magnitude of network coefficients to the loss function, encouraging the model to maintain smaller weights, thereby simplifying the model and avoiding overfitting. Early Stopping halts training as soon as the model's performance on a validation set starts to deteriorate, despite ongoing improvements on the training set, ensuring the model does not overtrain and generalize poorly on new, unseen data.

In order to prevent the model from learning irrelevant patterns, preprocessing steps are taken to circumvent the potentially misleading information derived from stock market data. Low pass filtering is an excellent example of this as it reduces the high-frequency noise typically found within such convoluted data. Normalization also offers solutions for problems of scale in which collections of data operate are displayed within ranges that are traditionally comparable, it does this by "normalizing" the data on an equivalent scale, typically 0-1, which is most often vital for the training of deep learning models.

Data augmentation has become a key preliminary process for financial data given the inherently limited quantity and quality. Shifting, filtering and artificial noise are techniques that are frequently implemented in order to artificially expand the dataset and help create varied training samples that mimic real financial phenomena. The connection between the simulated financial data and the real-world observed financial data is paramount as the model will only perform as well as its training data, cementing the demand for the transfer from theoretical to practical applications. The model's accuracy and robustness are, in turn, significantly boosted through the unity of data augmentation with preprocessing and regularization building an exhaustive approach that prepares it to handle the unpredictable nature of the financial markets.

Cryptocurrency Market Adjustments

Several modifications and enhancements need to be applied to the TDQN algorithm to account for the unique features of the cryptocurrency market. As mentioned in Section 3.1.1, the incorporation of cryptocurrency-specific metrics can be used to enhance the observation space, reward system, and cost functions (*King et al., 2024*). Some of these metrics include hash rates, mining speed, transaction volumes, and blockchain analytics. The successful application of DRL and specifically TDQN to different stocks and options indicates the potential for fine-tuning the model.

There are many costs that significantly impact the ROI of specific trading systems. This is all the more important to consider when dealing with transaction costs in the cryptocurrency market. Some examples include exchange platform fees, network fees, currency exchange rates, and perhaps most importantly, the transaction logging latency of the blockchain. Therefore, a heuristic representative of the cryptocurrency market should be added to the cost value (¿0.2 is recommended). In summary, adding this heuristic to the model assists the algorithm in staying profitable despite the additional expenses.

Risk management techniques are also essential when dealing with high volatility. Mechanisms like stop-loss can help mitigate losses by selling assets when a certain threshold is reached (*Lei & Li, 2009*). Another example is dynamic position sizing, which, based on conditions and volatility of the market, adjusts the size of investments. Using measures like the Sharpe ratio or the Calmar ratio directly in the reward function and not just as a form of evaluation can help the algorithm balance returns with risk (*Peng et al., 2022*). TDQN can employ these techniques to maximize returns all the while managing risks and avoiding obstacles cryptocurrency trading may present.

Moreover, the continuous 24/7 nature of cryptocurrency markets requires the TDQN algorithm to operate without downtime. This constant trading cycle presents both opportunities and challenges. To address this, the algorithm needs to be designed for continuous operation, possibly incorporating mechanisms for real-time data processing and decision-making. This design consideration will allow the TDQN algorithm to respond swiftly to market changes, maintaining its effectiveness in a constantly active market environment.

In summary, by incorporating cryptocurrency-specific metrics, adjusting for transaction costs, managing volatility with advanced techniques, and ensuring continuous operation, the TDQN algorithm can be effectively tailored for cryptocurrency trading. These modifications will enhance its ability to maximize returns and manage risks, making it a robust tool for navigating the complexities of cryptocurrency markets.

3.5 Results

In this research, several key challenges meant that a fine-tuned implementation of a TDQN for the cryptocurrency market could not be performed. The main reason being the exceedingly high computational demand to train and simulate this sort of system. In order to accurately model and validate the TDQN, an extensive amount of data and training epochs is required. While theoretical frameworks and simulation models offer valuable insights, the lack of access to high-quality, real-time data limits the ability to fully test and deploy a practical TDQN system. Therefore, while the conceptual exploration and theoretical validation of the TDQN provide a solid foundation, implementing a complete system for cryptocurrency trading was not feasible.

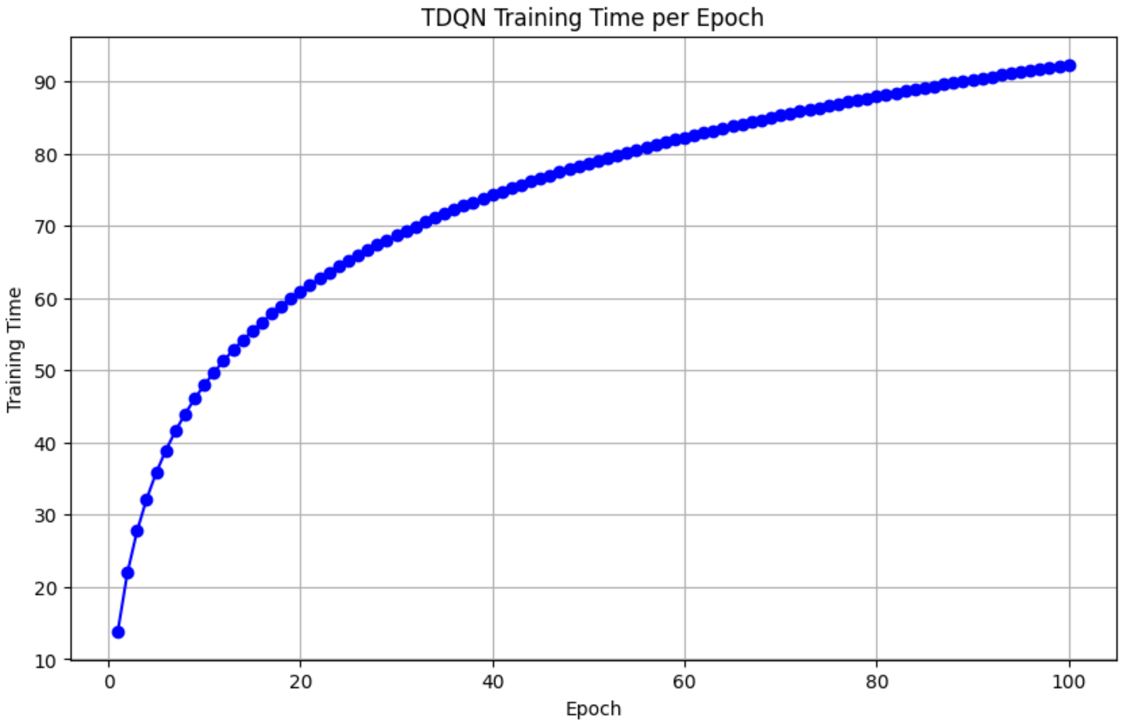


Figure 3.18: Estimated TDQN training times (APPL stocks) Source: Théate and Ernst, 2021

However, the effectiveness of the algorithm is not solely dependent on quantitative results but also on the strengths in its architecture that allow it to navigate complex trading scenarios efficiently. The TDQN algorithm still shows significant promise in various market environments. It's adaptability is particularly relevant for the cryptocurrency market. By implementing targeted modifications it can effectively handle the unique challenges posed by this market, providing a robust and adaptive trading strategy. The continuous improvement and scalability of the TDQN algorithm position it as a cutting-edge solution for navigating the complexities of cryptocurrency trading. With the added metrics and fine-tuned parameters a satisfactory result should be achieved.

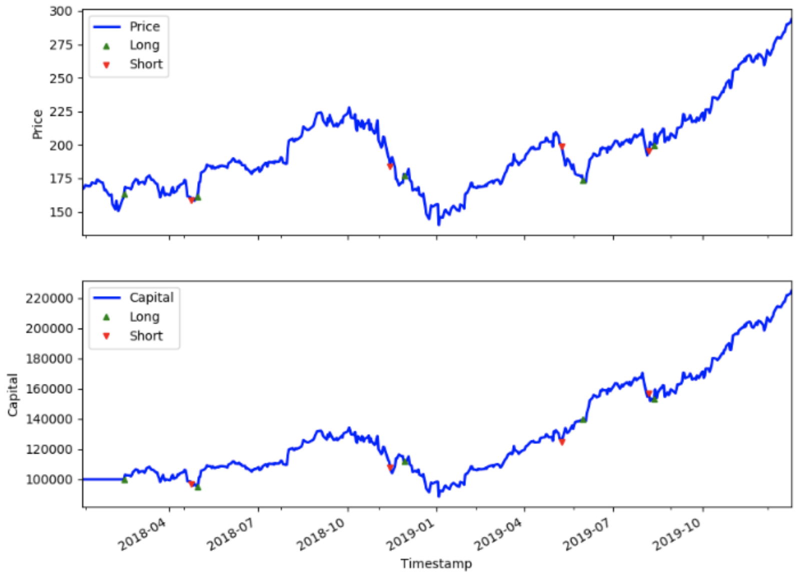
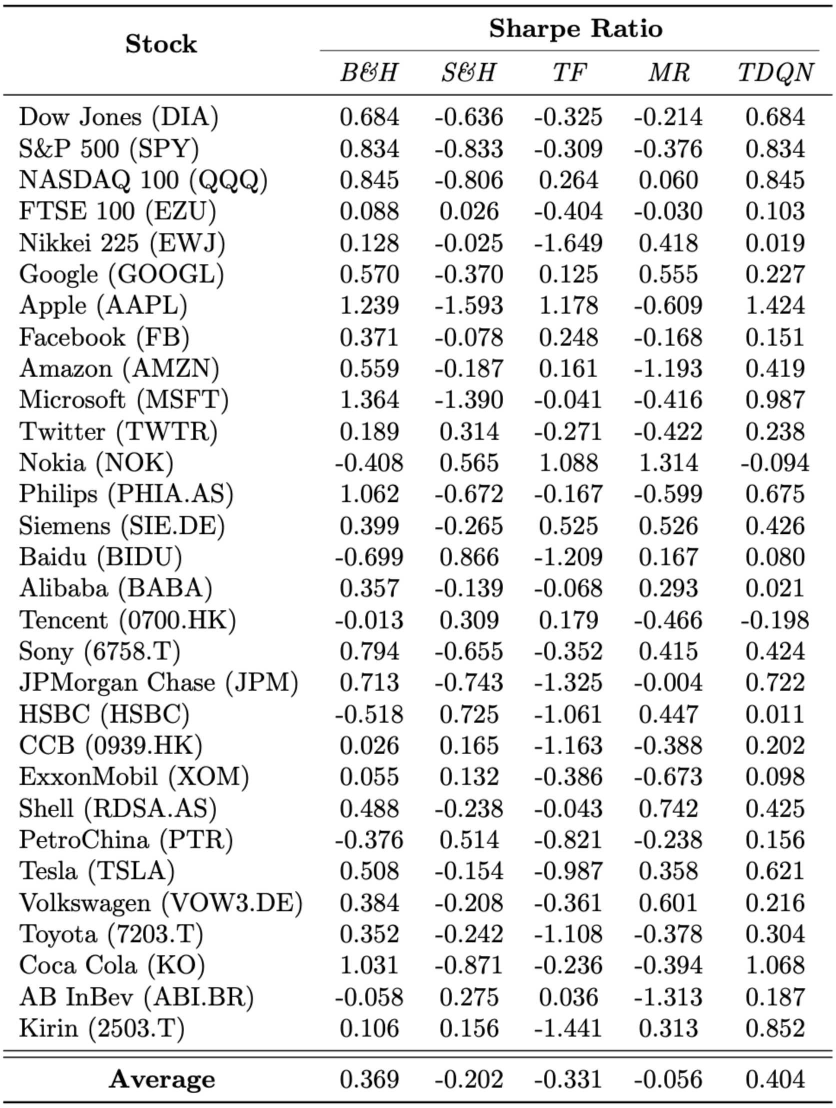


Figure 3.19: TDQN expected results 0.2 transaction cost (APPL stocks) Source: Théate and Ernst, 2021

In the original research paper introducing TDQN, they tested and validated across multiple stocks and indexes. It managed to surpass the the benchmark trading strategies (*Théate & Ernst, 2021*).



Stock	Sharpe Ratio				
	B&H	*S&H*	*TF*	*MR*	*TDQN*
Dow Jones (DIA)	0.684	-0.636	-0.325	-0.214	0.684
S&P 500 (SPY)	0.834	-0.833	-0.309	-0.376	0.834
NASDAQ 100 (QQQ)	0.845	-0.806	0.264	0.060	0.845
FTSE 100 (EZU)	0.088	0.026	-0.404	-0.030	0.103
Nikkei 225 (EWJ)	0.128	-0.025	-1.649	0.418	0.019
Google (GOOGL)	0.570	-0.370	0.125	0.555	0.227
Apple (AAPL)	1.239	-1.593	1.178	-0.609	1.424
Facebook (FB)	0.371	-0.078	0.248	-0.168	0.151
Amazon (AMZN)	0.559	-0.187	0.161	-1.193	0.419
Microsoft (MSFT)	1.364	-1.390	-0.041	-0.416	0.987
Twitter (TWTR)	0.189	0.314	-0.271	-0.422	0.238
Nokia (NOK)	-0.408	0.565	1.088	1.314	-0.094
Philips (PHIA.AS)	1.062	-0.672	-0.167	-0.599	0.675
Siemens (SIE.DE)	0.399	-0.265	0.525	0.526	0.426
Baidu (BIDU)	-0.699	0.866	-1.209	0.167	0.080
Alibaba (BABA)	0.357	-0.139	-0.068	0.293	0.021
Tencent (0700.HK)	-0.013	0.309	0.179	-0.466	-0.198
Sony (6758.T)	0.794	-0.655	-0.352	0.415	0.424
JPMorgan Chase (JPM)	0.713	-0.743	-1.325	-0.004	0.722
HSBC (HSBC)	-0.518	0.725	-1.061	0.447	0.011
CCB (0939.HK)	0.026	0.165	-1.163	-0.388	0.202
ExxonMobil (XOM)	0.055	0.132	-0.386	-0.673	0.098
Shell (RDSA.AS)	0.488	-0.238	-0.043	0.742	0.425
PetroChina (PTR)	-0.376	0.514	-0.821	-0.238	0.156
Tesla (TSLA)	0.508	-0.154	-0.987	0.358	0.621
Volkswagen (VOW3.DE)	0.384	-0.208	-0.361	0.601	0.216
Toyota (7203.T)	0.352	-0.242	-1.108	-0.378	0.304
Coca Cola (KO)	1.031	-0.871	-0.236	-0.394	1.068
AB InBev (ABI.BR)	-0.058	0.275	0.036	-1.313	0.187
Kirin (2503.T)	0.106	0.156	-1.441	0.313	0.852
Average	0.369	-0.202	-0.331	-0.056	0.404

Figure 3.20: Performance assessment Source: Théate and Ernst, 2021

4. Discussion & Limitations

During this research, the disparity in the literature between machine learning algorithms and reinforcement learning algorithms applied to algorithmic trading became apparent. While traditional methods and machine learning models have been widely applied, documented reinforcement learning applications are sparse. More advanced and potentially effective DRL systems are largely missing from the literature. Particularly there is a lack of practical deployment of DRL in real-time trading environments. The optimization and validation of the DRL systems is loosely defined and no foundational framework has been determined as of yet. Additionally, there is a lack of studies that focuses on DRL model scaling for real-time operations and, aside from theoretical perspectives, there is a lack of research on incorporating risk management within DRL frameworks. Overall, the three main limitations found during this research and from the literature is the computational intensity, representative cryptocurrency market simulations, and the obscurity surrounding technical details.

While the theoretical aspects of DRL to dynamically adapt and generate optimized trading strategies is intriguing, practical implementations of such sophisticated models come with many challenges. Intensive simulations that require large amounts of data are increasingly computationally demanding. Particularly, the simulated nature of reinforcement learning required for training accurate DRL models and the high-stakes involved with algorithmic trading proves a significant hurdle for researchers and enthusiasts. Furthermore, this could be a reason why the research surrounding RL algorithmic trading seems to be obscured and potentially misleading. Large financial institutions are of the few that have access to such large amounts of compute power and investment capital, which is often required to have a competitive lead over the market.

In order to address the limitations, there are several areas for improvement that may prove effective. Future efforts should look into optimization of DRL systems for computational efficiency. This can be potentially be achieved by developing new models that are less resource intensive and also possibly using advancements in parallel processing and hardware acceleration. Another, more ambitious approach, would be to investigate hybrid models that combine the decision-making and simulation capabilities of DRL with the computational efficiency of machine learning models. This could provide a balanced approach for algorithmic trading that helps manage the trade-off between performance and compute power. It's possible that a more sophisticated simulation of the market environment would improve results. Moreover, a deeper understanding of the financial theory economic features of the cryptocurrency market could lead to innovations in the development and fine tuning of applied DRL systems.

5. Conclusion

This research was aimed at providing valuable insights into the application of deep reinforcement learning in the algorithmic trading field. Specifically, tasked with navigating the complexities of the cryptocurrency market. The literature mentioned in the research shows evidence that DRL is a viable competitor for the traditional trading strategies and machine learning algorithms that are widely employed in the market today. The TDQN architecture discussed has shown the most promising results so far by using various strategies that are able to find an optimal policy gradient and maintain a positive Sharpe ratio despite unpredictable environment conditions. Additional to the TDQN architecture, the research suggests applying cryptocurrency specific metrics and indicators, like hash rates, transaction costs, and blockchain analytics to the RL model's reward system and cost function. Therefore, DRL systems can be adapted to the unique characteristics and complications faced in the cryptocurrency market. These additional parameters along with the short time intervals and continuous data show promise for an innovative method for fine tuning applications of DRL for cryptocurrency trading. However, modern algorithmic trading systems are largely complex and require many components that are built for specific markets and conditions. RL systems for algorithmic trading tend to achieve sub-optimal policies when generalized to many markets, stocks, cryptocurrencies, etc. This being said, DRL models applied specifically to cryptocurrencies still requires heavy testing, fine tuning, and architectures specifically made to tackle the reinforcement learning problem.

Despite these challenges, this research demonstrates the inherent advantages of DRL in navigating highly uncertain and volatile environments such as the cryptocurrency market. Even though significant quantitative results could not be obtained from the DRL system, an important methodology and outline for implementing the many components and deploying a satisfactory system are provided. Including data gathering and processing, benchmarking, backtesting, and maintaining. Upon overcoming the significant challenges that as of now delay real-time large scale applications the potential of the proposed method's may be realized.

In conclusion, while DRL presents promising opportunities for algorithmic trading in the world of the cryptocurrency market, a full implementation of such systems must be approached with an awareness of the present logistical and computational challenges. These challenges require technical solutions and a deep understanding of financial theory and the trading dynamics that inhibit market behaviors. As more research into the field is conducted, the adaptation of DRL systems with robust frameworks will drive forward the capabilities of financial technology.

6. Future Work

Future work in this field of research should focus developing efficient DRL trading systems that are not only computationally accessible but also accurate and effective. By leveraging hardware and software optimization advancements such as distributed computing systems and data pipelines a real-time trading system incorporating DRL models can be achieved. Additionally, hardware such as GPUs and TPUs can significantly accelerate the training process. Further experimentation into data acquisition and processing techniques may enhance training phases, which can potentially reveal trading characteristics in other markets or assets. Most importantly, getting a effective TDQN model fine-tuned for cryptocurrency algorithmic trading is a priority. Fulfilling this task will clearly define deep reinforcement learning algorithm potential for adaptability in various conditions.

Another promising area of future work is the implementation of Proximal Policy Optimization (PPO) algorithms. PPO was developed by OpenAI and is known for its stable and reliable performance in reinforcement learning tasks. It strikes a balance between complexity and performance by using a clipped objective function, which prevents large policy updates that can destabilize training. Implementing PPO for cryptocurrency trading could yield effective models capable of handling the high volatility and dynamic nature of the market. Also, by taking a multi-model approach, the integration of sentiment analysis from social media and news sources, a common approach for informed trading strategies, could further enhance the DRL algorithm's predictive capabilities.

By addressing these areas and further expanding based on related technology, future research can pave the way for more practical and scalable DRL applications in cryptocurrency trading, ultimately leading to more efficient and profitable trading strategies.

Code & Data Availability

For future work purposes, the code used for this research and the datasets are provided in the following GitHub repository: github.com/paches00/rl-crypto-trading.

Bibliography

Forbes. (2024). Cryptocurrency prices, market cap and charts.

Bakas, D., Magkonis, G., & Oh, E. Y. (2022). What drives volatility in Bitcoin market? *Finance Research Letters*, *50*, 103237.

Burgess, N. (2019). An introduction to algorithmic trading: Opportunities & challenges within the systematic trading industry. *SSRN*.

Woebbeking, F. (2021). Cryptocurrency volatility markets. *Digital Finance*, *3*(3-4), 273–298.

Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., & Li, L. (2022). Cryptocurrency trading: A comprehensive survey. *Financial Innovation*, *8*(1).

Kolm, P. N., & Maclin, L. (2010). Algorithmic Trading. *Encyclopedia of Quantitative Finance*.

Kilkenny, M. F., & Robinson, K. M. (2018). Data quality: "garbage in – garbage out" [PMID: 29719995]. *Health Information Management Journal*, *47*(3), 103–105.

Cohen, G., & Qadan, M. (2022). The Complexity of Cryptocurrencies Algorithmic Trading. *Mathematics*, *10*(12), 2037.

Granville, J. E. (1963). Prentice- Hall.

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*, 1–30.

Norman, C. T. (2019). Reinforcement LEARNING APPROACH FOR ALGORITHMIC TRADING OF BITCOIN A THESIS. *THE UNIVERSITY OF OKLAHOMA GRADUATE COLLEGE.*

King, J. C., Dale, R., & Amigó, J. M. (2024). Blockchain metrics and indicators in cryptocurrency trading. *Chaos, Solitons & Fractals*, *178*, 114305.

Sebastião, H., & Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation*, *7*(1).

Alessandretti, L., ElBahrawy, A., Aiello, L. M., & Baronchelli, A. (2018). Anticipating Cryptocurrency Prices Using Machine Learning. *Complexity*, *2018*, 1–16.

Colianni, S., Rosales, S., & Signorotti, M. (2015). Algorithmic trading of cryptocurrency based on Twitter sentiment analysis. *CS229 Project*, *1*(5), 1–4.

Spörer, J. F. (2020). Backtesting of Algorithmic Cryptocurrency Trading Strategies. *Frankfurt School of Finance and Management.*

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning - an introduction* (Vol. 9). MIT Press.

Comis, E. (2021). Reinforcement Learning for financial trading: An application to the cryptocurrency market.

Corazza, M., Fasano, G., Gusso, R., & Pesenti, R. (2021). *Comparing RL Approaches for Applications to Financial Trading Systems.* Springer International Publishing.

Taghian, M., Asadi, A., & Safabakhsh, R. (2022). Learning financial asset-specific trading rules via deep reinforcement learning. *Expert Systems with Applications*, *195*, 116523.

Park, J.-H., Kim, J.-H., & Huh, J.-H. (2024). Deep Reinforcement Learning Robots for Algorithmic Trading: Considering Stock Market Conditions and U.S. Interest Rates. *IEEE Access*, *12*, 20705–20725.

Felizardo, L. K., Paiva, F. C. L., Costa, A. H. R., & Del-Moral-Hernandez, E. (2022). Reinforcement Learning Applied to Trading Systems: A Survey. *arXiv*.

Dardouri, N., Aguir, A., & Smida, M. (2023). The Effect of COVID-19 Transmission on Cryptocurrencies. *Risks*, *11*(8), 139.

Zhang, L., Wu, T., Lahrichi, S., Salas-Flores, C.-G., & Li, J. (2022). A Data Science Pipeline for Algorithmic Trading: A Comparative Study of Applications for Finance and Cryptoeconomics. *2022 IEEE International Conference on Blockchain (Blockchain)*.

Resta, M., Pagnottoni, P., & Giuli, M. E. D. (2020). Technical Analysis on the Bitcoin Market: Trading Opportunities or Investors' Pitfall? *Risks*, *8*(2), 44.

Hrnjica, B., & Bonacci, O. (2019). Lake level prediction using feed forward and recurrent neural networks. *Water Resources Management*, 1–14.

Botunac, I., Bosna, J., & Matetić, M. (2024). Optimization of Traditional Stock Market Strategies Using the LSTM Hybrid Approach. *Information*, *15*(3), 136.

Peng, A., Ang, S. L., & Lim, C. Y. (2022). Automated Cryptocurrency Trading Bot Implementing DRL. *Pertanika Journal of Science and Technology*, *30*(4), 2683–2705.

Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, *173*, 114632.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

Hinton, G. (2012). Neural networks for machine learning - lecture 6a: Overview of mini-batch gradient descent.

Lei, A. Y., & Li, H. (2009). The value of stop loss strategies. *SSRN Electronic Journal*.

A. Appendix

MACD Crossover

$EMA_t = \left[V_t \times \left(\frac{s}{1+d}\right)\right] + EMA_y \times \left[1 - \left(\frac{s}{1+d}\right)\right]$

$MACD = 12{-}PeriodEMA - 26{-}PeriodEMA$

Bellman Equations

State-Value Function:

$$V^{\pi}(s) = \sum_{a \in A} \pi(a|s) \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^{\pi}(s')]$$

Action-Value Function:

$$Q^{\pi}(s,a) = \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma \sum_{a'} \pi(a'|s') Q^{\pi}(s',a')]$$