A REVISED APPROACH TO CRYPTOCURRENCY PORTFOLIO OPTIMIZATION USING ADVANCED Q-LEARNING AND POLICY ITERATION FRAMEWORKS

A Thesis

by

Ceren Altok

Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in
Department of Data Science

Özyeğin University
September 2023

A REVISED APPROACH TO CRYPTOCURRENCY PORTFOLIO OPTIMIZATION USING ADVANCED Q-LEARNING AND POLICY ITERATION FRAMEWORKS

Approved by:

Asst. Prof. Erinc Albey, Advisor
Dept. of Industrial Eng.
Özyeğin University

Asst. Prof. Mehmet Önal
Dept. of Industrial Eng.
Özyeğin University

Prof. Mehmet Güray Güler
Dept. of Industrial Eng.
Yıldız Teknik University

Date Approved: Aug 14, 2023

To my supportive family and beloved husband

ABSTRACT

Despite all the factors that cause concern among investors, such as volatility and decentralization of crypto world, the popularity of cryptocurrencies continues to grow steadily. The cryptocurrency market still holds its allure for many investors due to the high profit levels it has experienced in the past. With the entrance of numerous altcoins into the market, portfolio management becomes much more challenging. In the literature, we come across numerous studies proposing efficient portfolio management techniques for cryptocurrencies.

This study presents proposed models developed based on policy iteration and Q-learning algorithms. Under Q-learning, three distinct sub-models are introduced: Deep Q-Network (DQN), Double Deep Q-Network (DDQN), and Double Dueling Q-Network (DDDQN). All of these models are trained using 6-month training periods and compared using 10 different training and testing periods. Additionally, to evaluate both of proposed policy iteration and Q-learning models, baseline models were created for each algorithm, and the performance of the proposed models was assessed against these baseline models.

The results indicate that among Policy Iteration models, the proposed model has the highest average ROI value of 3%, making it the top-performing model. Similarly, among Q-learning models, the proposed DQN model surpasses both baseline models and other Q-learning models, with an average ROI value of 2%. Considering all the models, the proposed Policy Iteration model achieves the highest average ROI value, while the proposed DQN and the proposed DDDQN model demonstrates the lowest volatility in terms of ROI standard deviations.

ÖZETÇE

Yatırımcılar arasında volatilite ve merkezsizleşme gibi endişe yaratan tüm faktörlere rağmen, kripto para birimlerinin popülerliği istikrarlı bir şekilde artmaya devam etmektedir. Kripto para piyasası, geçmişte yaşadığı yüksek kar seviyeleri sebebiyle birçok yatırımcı için hala cazibesini korumaktadır. Günlük olarak birçok alternatif kripto para biriminin piyasaya girmesiyle, portföy yönetimi çok daha zorlu hale gelmektedir.

Literatürde, kripto portföylerini verimli bir şekilde yönetmek için önerilen birçok çalışma bulunmaktadır. Bu çalışma, Policy Iteration ve Q-learning algoritmalarından türetilen yeni model önerileri sunmaktadır. Q-learning altında Deep Q-Network (DQN), Double Deep Q-Network (DDQN) ve Double Dueling Q-Network (DDDQN) olmak üzere üç farklı alt model tanıtılmaktadır. Bu modellerin hepsi, 6 aylık eğitim dönemleri kullanılarak eğitilmiştir ve 10 farklı eğitim ve test dönemi kullanılarak karşılaştırmalar yapılmıştır. Ayrıca, hem önerilen politika iterasyonu hem de önerilen Q-learning modellerini değerlendirebilmek için iki algoritma için de basit referans modelleri oluşturulmuş ve önerilen model değerlendirmeleri bu referans modelleri kullanılarak yapılmıştır.

Sonuçlara göre, önerilen Policy Iteration modeli Policy Iteration modelleri arasında %3 ortalama ROI değeri ile en iyi modeldir. Benzer şekilde, önerilen Q-learning modelleri arasında DQN modeli, hem referans modelini hem de diğer Q-learning modellerini %2 ortalama ROI değeri ile geride bırakmaktadır. Tüm modelleri dikkate aldığımızda, önerilen Politika İterasyonu modelinin en yüksek ortalama ROI değerine sahip olduğunu, önerilen DQN ve DDDQN modellerinin ise ROI standart sapması açısından en düşük volatiliteye sahip olduğunu görmekteyiz.

ACKNOWLEDGEMENTS

I want to extend my deep gratitude to my advisor Erinç Albey for his incredible guidance and support throughout this essay. His expertise and encouragement have been in- valuable, and I am truly grateful for his help. I also want to thank my family for their unwavering love and encouragement. Their belief in me has been a constant source of mo- tivation. To my dear spouse, thank you for your endless support and understanding. Your presence has made this journey all the more meaningful. I am also grateful to everyone who has contributed to my growth and success, big or small. Lastly, I want to acknowledge all those who have been part of this journey in any way. Thank you for making this possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Over the past few years, cryptocurrencies have gained tremendous popularity, and although it may not have reached the same level of maturity as Forex or other full developed markets, it is steadily progressing towards becoming a mature market. [1]. As of June 2022, there were around nineteen thousand cryptocurrencies in the market, and 40 of these cryptocurrencies, such as BNB, had a market capitalization exceeding $1 billion. [2]. In this rapidly growing crypto world, there are numerous factors that investors need to pay attention to, which influence their portfolio decisions. Firstly, the cryptocurrency market's evolving nature and the influx of altcoins pose challenges for portfolio management. According to [3]., the introduction of new altcoins has a negative impact on Bitcoin. Therefore, investors need to continuously monitor the entire market and be able to adapt to this changing crypto world. Secondly, high volatility across all crypto stock is affecting all investment behavior. The study [4]. has shown that, except for Ripple and Monero, the entire market is succumbing to this high volatility, resulting in decreased efficiency. In other words, this high volatility discourages investors and creates hesitancy when it comes to entering the cryptocurrency market.

Even in traditional stock markets, portfolio optimization is a challenging subject, but due to characteristics of cryptocurrencies mentioned above, portfolio management becomes even more difficult. We observe that numerous studies have been conducted to cope with the challenging task of cryptocurrency portfolio management. These studies are described in Chapter 2 – Literature Review under the topics of cryptocurrency price prediction and cryptocurrency portfolio optimization.

This study presents proposed models based on the Policy Iteration and Q-learning algorithms. Additionally, the design of these two proposed models incorporates the output of another deep learning model in their optimization process. To evaluate these proposed models, reference baseline models have been created. Therefore, Chapter 3 - Problem Definition provides a preliminary overview of policy iteration, Q-learning, and the underlying neural network models. Chapter 4 – Research Design and Method includes all the details about experiments such as data sets used, target definitions, hyperparameter tuning specifics, and other relevant information.

In Chapter 5 - Results, you will find comparisons of both the benchmark models and the proposed models for policy iteration and Q-learning. This includes evaluations of their individual performances, as well as an overall assessment of all proposed and benchmark models together. The chapter aims to provide insights into the implications derived from this comprehensive evaluation, offering a conclusive analysis of the overall picture.

General summaries of the study and insights, along with comments and suggestions for future research, are provided In Chapter 6 - Conclusion. This section offers a comprehensive overview of the key findings, highlighting the significance of the study's contributions to the field.

CHAPTER II

LITERTURE REVIEW

The cryptocurrency market is constantly evolving, with the emergence of numerous altcoins affecting even established cryptocurrencies like Bitcoin. This complexity makes portfolio management challenging for all investors. To effectively manage their portfolios in this volatile space, investors must continuously adapt their strategies and remain vigilant. In parallel with the growing cryptocurrency world, we observe an increasing number of studies dedicated to addressing various challenges in cryptocurrency portfolio management.

This study focuses on two specific topics, and therefore, this section of the report will provide a summary of relevant studies in the respective areas. The first part will cover research conducted on price predictions, while the second part will delve into portfolio management studies that examine the decision-making process for buying and selling assets.

2.1 Price Prediction Model Studies in Literature

An excellent example of a fundamental study on cryptocurrency price prediction models is the research conducted by [5]. This study focuses on a specific cryptocurrency, ETH, and utilizes Linear Regression and Support Vector Machine algorithms to develop a price prediction model. One significant aspect of this study, which is also relevant to our research, is that the models were trained on different time periods to compare the performance of the algorithms across various periods. In a similar vein, in the [6] study, we also observe a comparison of two algorithms: Xgboost and LSTM. However, in this study, two important aspects have been questioned alongside. The first question examines the optimal period for feeding the models and it is observed

that LSTM performs significantly better when trained on longer periods. The second aspect being investigated is the comparison between cryptocurrency-specific prediction models and making collective predictions by considering the entire market. The results show that cryptocurrency-specific models perform better for the entire portfolio. Additionally, there are numerous similar studies available in the literature on this topic. [7],[8],[9],[10],[11].

In [12], it is observed that Zcash outperforms Ethereum and Litecoin in predicting Bitcoin prices for following four algorithms used: Linear, Gradient Boosting, Random Forest, and Support Vector Regressors. Such studies highlight that another important consideration in price prediction models is the presence of hidden relationships between cryptocurrencies that may not be evident at first glance. Indeed, this showcases the complexity and high-dimensional nature of the cryptocurrency world. It demonstrates that in price prediction models, considering not only the features of individual cryptocurrency but also market features as a whole and the characteristics of other related cryptocurrencies can positively impact the results.

In price prediction models, we observe that the next stage of algorithms comparison studies are ensemble models. In [13]'s study, we see how a model designed by combining LSTM and GRU outperformed the baseline model for Monero and Litecoin. In the literature, we come across numerous hybrid model studies for forecasting prices, such as the combination of Markov Models and LSTM [14], ARIMA and Artificial Neural Networks (ANN) [15].

2.2 Portfolio Optimization Studies in Literature

There are numerous studies in portfolio management that focus on multi-agent structures. In the study conducted by Patel [16], macro agents make decisions on holding, buying, and selling based on tick data, while the micro agent is responsible for placing limit orders. The optimization process in this work incorporates Q-learning to

improve overall performance. This approach allows for a comprehensive and dynamic optimization strategy by leveraging the combined efforts of multiple agents in the cryptocurrency market. Another interesting study involves the utilization of Jae Won Lee and Jangmin O, where four agents are employed. These agents are the sell signal, sell, buy signal, and buy agents, respectively. Based on the information received from signals, the next decision of the buy and sell agents is determined, either acting or remaining inactive. As a result, it has been demonstrated and conveyed in this study that this arrangement yields profitable outcomes. [17]

Alongside multi-agent studies, ensemble approaches are also prevalent in price prediction models. In their study, [18], researchers merged Markov Decision Process (MDP) with Genetic Algorithm (GA) for stock market. This combination offers a promising approach for optimizing portfolios. This study utilizes the Markov Decision Process (MDP) in real-time to achieve accurate timing and strategy. Concurrently, Genetic Algorithms (GA) are employed to enhance the optimal stock selection strategy and capital allocation. By leveraging the strengths of both MDP and GA, they provide valuable insights into improving portfolio management strategies.

[19] is another highly significant study where they propose a deep learning-based model that examines all cryptocurrencies in the market to determine a portfolio management strategy. This approach allows for the inclusion of newly emerging cryptocurrencies directly into the portfolio.

Examples even more complex studies can be found in the field of multi-objective portfolio management. Studies like [20] explore the application of multi-objective evolutionary (MOE) algorithms. This innovative approach allows for a comprehensive exploration of optimization possibilities and provides valuable insights into the effectiveness of different techniques in the subject of cryptocurrency portfolio management. Following MOE algorithms are compared in this study: Non-Dominated Genetic Algorithm II, Non-Dominated Genetic Algorithm III, Speed-Constrained

Multi-objective Particle Swarm Optimization, Multi-objective Evolutionary Algorithm based on Decom- position, Generalized Differential Evolution, Steady-state Epsilon-dominance.

CHAPTER III

PROBLEM DEFINITIONS

This study primarily aims to introduce proposed models derived from reinforcement learning sub algorithms. In this section of the report, we will provide a brief description of used algorithms and explain problem definitions of proposed models and baseline models. Below is a flowchart that provides a concise overview to gain a general understanding of reinforcement learning.[21] Our proposed model is derived from Q-learning and Policy Iteration, from temporal – difference learning and dynamic programming sections in Figure 1 respectively.

Our proposed models are more complex models that have been modified in terms of both their architecture and the input data, which differs slightly from the classical understanding. Firstly, in the proposed models, we run a pioneer neural network model to be used as input data. Thus, we have 2 proposed models, 2 baseline models and 1 pioneer neural network model explained in this part of the report.

In other words, in this section of the report, a brief explanation is given about the pioneer neural network model, policy iteration models, and q-learning models in

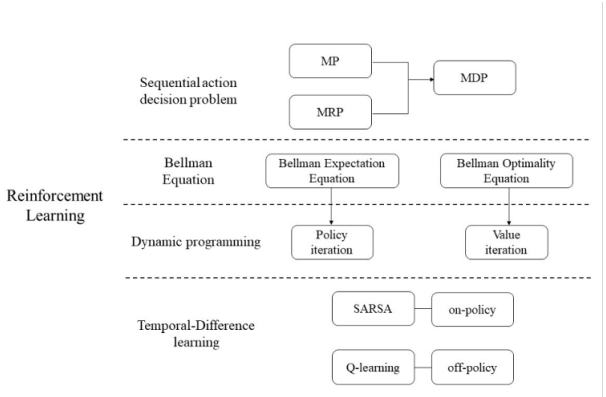

Figure 1: Reinforcement Learning Algorithms Diagram

sequential order, followed by the problem definition.

3.1 Pioneer Neural Network

As we have seen in the [5],[6],[7],[9] studies, there are numerous model studies in the literature that use neural networks, boosting, and similar algorithms to predict the next hour's price for each cryptocurrency. In this study, the neural network model outcomes will be used as input data for proposed models. Therefore, this neural network model is designed to predict whether the price will exceed or not exceed 5% of the current hour's price in the next 36 hours, and this binary value will be used as indicator for proposed models.

A classical neural network model with multiple layers has been preferred. The input dataset consists of rows that are hourly and crypto-specific. It provides various features related to the respective cryptocurrency and the market.

You can see the input matrix in Equation [1] below. Input data (I) includes (M) number of Features of (N) cryptocurrencies for (T) time period long. Time period column (t) is a datetime format hourly value while symbol column (s) shows basically symbols of cryptocurrencies.

$$I = \begin{bmatrix} t_1 & s_1 & FI_{11} & ... & FM_{11} \\ t_2 & s_1 & FI_{12} & ... & FM_{12} \\ t_T & s_1 & FI_{1T} & ... & FM_{1T} \\ ... & ... & ... & ... & ... \\ t_1 & s_2 & FI_{21} & ... & FM_{21} \\ t_2 & s_2 & FI_{22} & ... & FM_{22} \\ t_T & s_2 & FI_{2T} & ... & FM_{2T} \\ ... & ... & ... & ... & ... \\ t_T & s_N & FI_{NT} & ... & FM_{NT} \end{bmatrix} \quad (1)$$

Target value is a binary value for every crypto (N) in input data for every hour

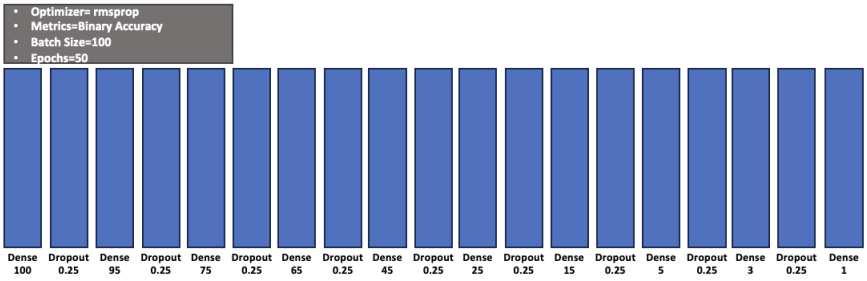

Figure 2: Pioneer Neural Network Structure

of (T) time period. On Equation 2, you can see the presentation of target- output matrix of our pioneer neural network model.

$$T = \begin{bmatrix} T_1 \\ T_2 \\ \dots \\ T_{t+1} \\ T_{t+2} \\ \dots \\ T_{NT+1} \end{bmatrix} \quad (2)$$

Lastly, please see on Figure 2 for information about pioneer neural network structure. Hyperparameter tuning details will be given on Chapter IV – Research Design and Method.

3.2 Dynamic Programming – Policy Iteration

3.2.1 Short description of Policy Iteration

Policy iteration is a dynamic strategy used for optimizing the Bellman Equation [3]. Here, R refers to reward of actions a on state. γ is the discount factor and P is the probability of transition from current state s to next state s'. V(s') is the value of next state after taking action a.

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s')\right) \quad (3)$$

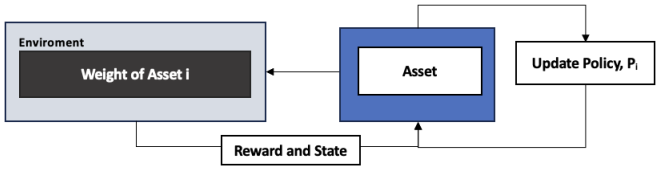

Figure 3: Baseline Policy Iteration Model Diagram

Summarily, R(s,a) gives reward of action a on state s and it is multiplied with next state value by including its discount factor. Thus, it calculates all possible action values and decide optimal action for best value for current states. [22]

It begins with random or default state values and gradually improves its performance by penalizing incorrect actions. Over time, it acquires knowledge about state transi- tions and becomes more effective. However, policy iteration is not well-suited for complex environments. As the number of states increases, the computational effort required also grows, making it less efficient. [23]

3.2.2 Problem Definition of Baseline Policy Iteration Model

In the preliminary version of the Policy Iteration approach, structures that are directly fed with asset-related features and there are no micro and macro agents. Please see a conventional model design as shown in Figure 3. PI 1 is an abbreviation for the Policy Iteration baseline model. For the pseudocode of the Baseline Policy Model – P1, please refer to Figure A1 in the Appendix.

3.2.2.1 States

State Space Set: $[W_{0t}, \ldots, W_{it}, \ldots, W_{nt}]$

- W_{it}: The weight of crypto i in the portfolio on time t

This parameter shows money allocation to crypto i on time t, in every time step weight will be updated according to taken actions. Since weight can get discrete values from set of [0,5,10], every coin can get 3 different values for their states. In

other words, investor can invest maximum 10$ to every cryptocurrency.

For instance, state set can be kept as [0,5,0,10], it means there is no money on BTC and CHZ while there is 5$ on DOGE and 10$ on SAND.

3.2.2.2 Actions

Action Set [$B1_{it}$, $B2_{it}$, $S1_{it}$, $S2_{it}$, H_{it}]

- $B1_{it}$: Buying action for cryptocurrency i on time t
- $B2_{it}$: Strong buying action for cryptocurrency i on time t
- $S1_{it}$: Selling action for cryptocurrency i on time t
- $S2_{it}$: Strong selling action for cryptocurrency i on time t
- H_{it}: Holding the money for cryptocurrency i on time t

Action space set includes 5 action types and buying and selling actions have two levels to show how strong investment or withdraw should occur. While normal buy or sell actions equal to 5$, strong buy or sell actions equal to 10$.

3.2.2.3 Reward

Reward equals to sum of nominal return of all cryptos in portfolio on time t.

3.2.2.4 Transition Probability

The transition probability is a matrix that represents the probability values of all state transitions. It is updated in every time step t. In this case, since each state consists of N cryptocurrencies in the portfolio, and each cryptocurrency can have a weight value of up to W, the shape of the transition probability matrix is W^N X W^N. In our case this array size is 3^4*3^4 since there are 4 cryptocurrencies in portfolio and 3 possible weight values.

3.2.3 Problem Definition of Proposed Policy Iteration Model

We aimed to go beyond the basic understanding of the Policy Iteration framework and present a new framework. We established a sophisticated framework that integrates

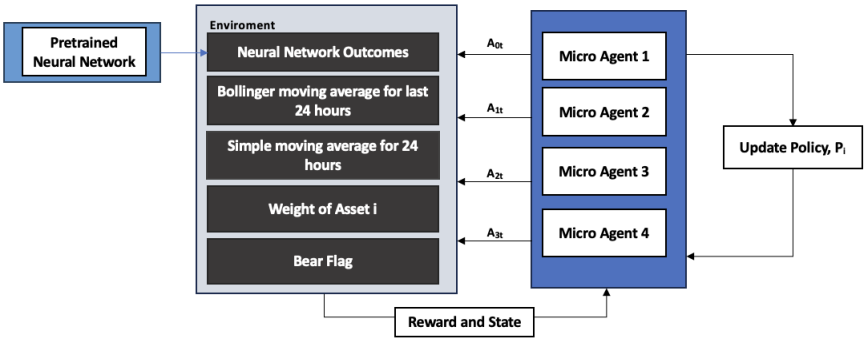

Figure 4: Proposed Policy Iteration Model Diagram

both macro market indicators and a pretrained neural network model outcome for a more comprehensive analysis. Please see Figure 4 to understand the framework of proposed Policy Iteration model – PI2.

Neural Network prediction, Bollinger moving average values, Simple moving average values, weights and Flag feature showing if the market is in bull or bear are parts of states. Micro agents act specific to their coins and change the environment.

3.2.3.1 States

State Space Set: $[W_{it}, M1_{1t}, M2_{1t}, M3_{it}]$

- W_{it}: The weight of crypto i in the portfolio on time t
- M_{1t}: Bear&Bull binary value showing if cryptocurrency market is on bear period, or bull on time t
- $M2_{1t}$: The binary value shows if Bollinger moving average value for last 24 hours on time t for cryptocurrency i is more than Bollinger moving average value on time t-1 for cryptocurrency i.
- $M3_{it}$: The binary value shows if simple moving average for last 24 hour at time t for cryptocurrency i is higher than simple moving average at time t-1 for cryptocurrency i.
- NN_{it}: The binary value showing the outcomes of our neural network model for

cryptocurrency i on time t

Metrics are selected based on their ability to serve various signal purposes. M_{1i} provides a macro-level overview of the cryptocurrency market, offering insights on its overall dynamics. M_{2i} is a crucial indicator for assessing crypto volatility, while the last signal offers information on cryptocurrency trends. M_{3i} is one of commonly used trend indicators.

In summary, an attempt was made to provide different views by incorporating a volatility feature, a macro-level feature, and a trend feature. In addition, a more complex and high-dimensional indicator was created and incorporated into the model using the output of a neural network model.

3.2.3.2 Actions

Actions Set: [$B1_{it}$, $B2_{it}$, $S1_{it}$, $S2_{it}$, H_{it}]

- $B1_{it}$: Buying action for cryptocurrency i on time t
- $B2_{it}$: Strong buying action for cryptocurrency i on time t
- $S1_{it}$: Selling action for cryptocurrency i on time t
- $S2_{it}$: Strong selling action for cryptocurrency i on time t
- H_{it}: Holding the money for cryptocurrency i on time t

3.2.3.3 Reward

Reward equals to sum of nominal return of all cryptos in portfolio on time t.

3.2.3.4 Transition Probability

The transition probability is a matrix that represents the probability values of all state transitions. In this case, since each state consists of N cryptocurrencies in the portfolio, 4 binary indicators added to state set and each cryptocurrency can have a weight value of up to W, the shape of the transition probability matrix is $W*2^4$ X $W*2^4$ which is equal to 48*48. Please see Figure A2 on Appendix for pseudocode of

Proposed Model.

3.3 Deep Q – Learning

3.3.1 Short description of Q-learning

3.3.1.1 Deep Q Network

Q-Learning is differentiated from Dynamic Programming from different aspects. Please see equation [4] for Q equation formula. Addition to Equation [3], it has a learning rate a and Q value is presented instead of value multiplied with probability since there is not transition probability anymore. The reason behind of this is that Q learning doesn't create a transition matrix and it learns how to calculate Q value with a deep leaning algorithm and it improves its deep learning model in every epoch and try to minimize its loss. In other words, Q-learning learns from the outcomes of actions rather than constructing a transition map like in MDP. Watkin describes the Q network as an agent that anticipates an immediate reward upon executing the recommended action and subsequently transitions to a state that holds value for the agent, based on a probability specific to this policy. [24]

$$\text{New } Q(s) = Q(s,a) + \alpha \cdot [R(s,a) + \gamma \cdot \max Q'(s',a') - Q(s,a)] \quad (4)$$

As illustrated in the equation (4) above [25], the Q-value is updated at each step by taking the chosen policy's result, multiplying it by the discount rate, and adding the reward to this element. This process represents a high-level overview of the Deep Q-Network (DQN) algorithm.

During the actual deployment phase, a different and often simplified approach may be taken, prioritizing exploitation over the careful trade-off between exploration and exploitation. [21]

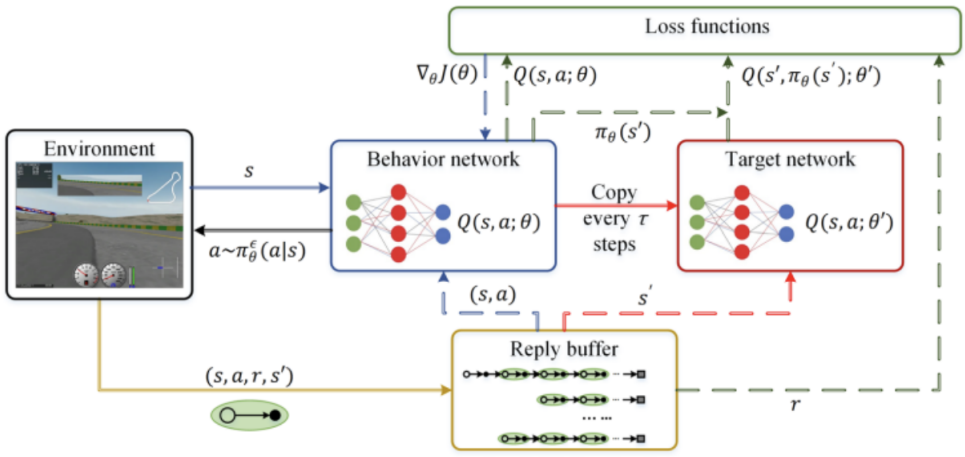

Figure 5: Double Deep Q-Learning Diagram

3.3.1.2 Double Deep Q Network

To address the issue of Q-value overestimation observed in DQN (Deep Q- Network), the double Q-learning approach was developed. In double Q-learning, both the target and main models are updated using the other model, resulting in them having different experiences. When selecting an action, both the main model and target model are utilized, allowing for a more reliable estimation of Q-values and mitigating the problem of overestimation. [21] Please see Figure 5 explaining Double deep q network flow presented in [26].

3.3.1.3 Dueling Double Deep Q Network

Estimating state-action values in reinforcement learning using the concept of advantage. Instead of directly estimating the state-action values, it suggests estimating the state values and relative advantages of each action separately. The advantage of an action is defined as the difference between the state-action value and the state value, representing how a particular action is better or worse compared to other actions in a given state.

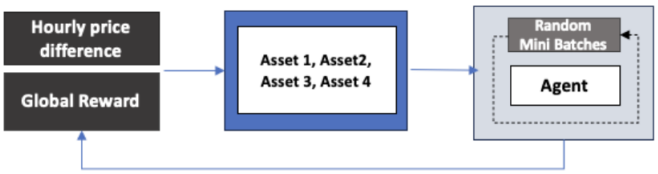

Figure 6: Baseline Q-Learning Model Diagram

To implement this idea, a neural network architecture called Dueling Networks is introduced. It consists of two computation streams: the value stream and the advantage stream. The value stream produces a single scalar output representing the state value, while the advantage stream outputs a vector with dimensions equal to the number of actions, representing the advantages of each action in that state. Both streams share the same convolutional encoder, allowing them to extract relevant features from the input.

By separating the estimation of state values and advantages, the Dueling Networks architecture enables more efficient and accurate learning. It allows the agent to focus on estimating the advantages of different actions while maintaining a shared representation of the state. This approach has been shown to be effective in various reinforcement learning tasks. [26]

To sum up, we enhanced Q-learning design by using another model output as input and set a different relations between micro and macro agents.

3.3.2 Problem Definition of Baseline Q-learning Model

Please see Figure 6, a baseline Q-learning framework implemented, which serves as a benchmark for our work. This framework provides a foundation for our research and allows us to compare and evaluate the performance of our proposed methods.

3.3.2.1 States

State Space Set: [[P diff_{it}], ..., [P diff_{it}] , ... , [P diff_{Nt}]]

• P $diff_{it}$: The price difference of close price at time t and close price at time t- 1 for cryptocurrency i for last 25 hour

3.3.2.2 Actions

Action Set: [$B1_{it}$, $B2_{it}$, $S1_{it}$, $S2_{it}$, H_{it}]

• $B1_{it}$: Buying action for cryptocurrency i on time t

• $B2_{it}$: Strong buying action for cryptocurrency i on time t

• $S1_{it}$: Selling action for cryptocurrency i on time t

• $S2_{it}$: Strong selling action for cryptocurrency i on time t

• H_{it}: Holding the money for cryptocurrency i on time t

Likely in policy iteration, there are 5 possible actions as strongly buy, buy, sell, strongly and hold the money.

3.3.2.3 Reward

Reward equals to sum of nominal return of all cryptos in portfolio on time t. Please see Figure A3 on Appendix for pseudocode of benchmark DQN model.

3.3.3 Problem Definition of Proposed Q-Learning Model

Unlike the structure of the basic Q-learning framework, our Q-learning algorithm assigns coins to different clusters by shuffling them. The main reason for adopting such a design is to handle the increased responsibility and potential performance degradation that an agent would face when dealing with a larger number of coins within a single cluster. To address this, we create subclusters to reduce the cluster size for micro agents, enabling the model to receive input from multiple agents and gather multiple perspectives on each coin at every step. We questioned if model's ability to make faster and more accurate decisions by incorporating the insights from multiple agents across different clusters for same coin. Please see Figure 7 the framework picture of proposed Q learning model framework. Please see Figure A4 on Appendix

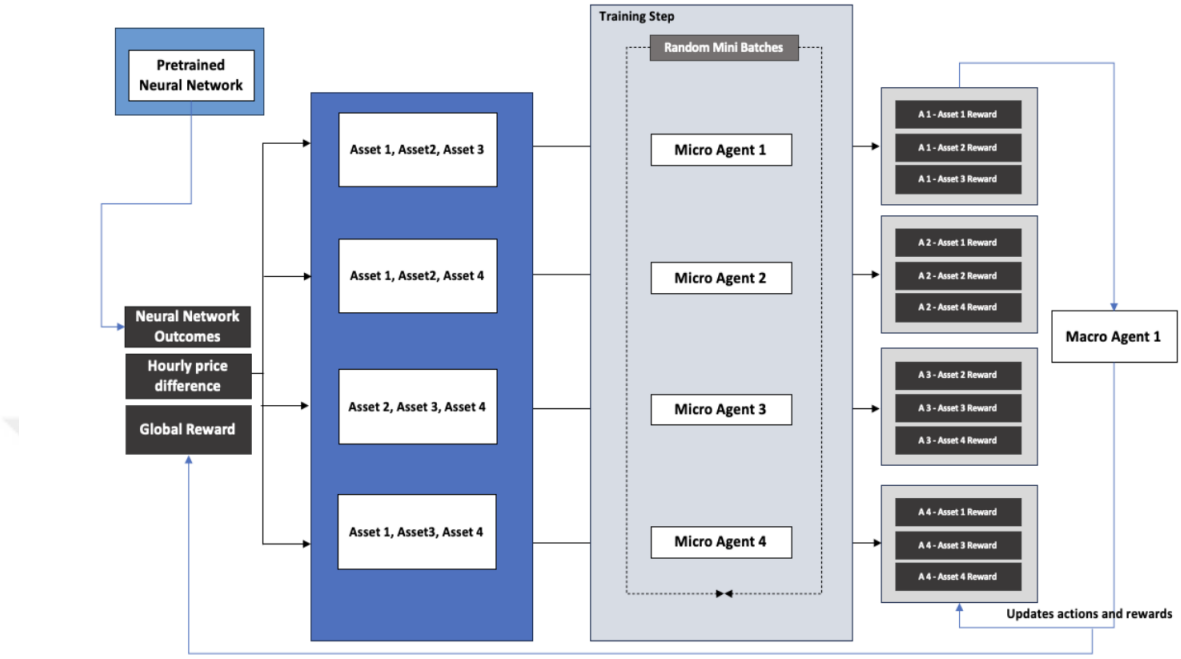

Figure 7: Proposed Q-Learning Model Diagram

for pseudocode for proposed DQN Model.

Our micro-agents, responsible for the subclusters, provide output regarding the reward values of each asset. The macro agent, on the other hand, evaluates the rewards assigned to the same coin by different micro-agents and considers the top two agents with the highest cumulative rewards. Based on this evaluation, the final actions and reward values for the four coins are adjusted without reflecting the results in the environment context.

Last difference from primitive model is proposed model has states including outcomes of a pretrained Neural Network which is trained to show signal if there is probability of price will go 5% higher of current close price.

3.3.3.1 States

State Space: [[N_{1t}, P $diff_{1t}$], ..., [N_{it}, P $diff_{it}$], [NN_t, P $diff_{Nt}$]]

- N_{it}: Binary value shows pretrained Neural Network prediction for coin i

at time t

• P $diff_{it}$: The price difference of close price at time t and close price at time t- 1

Close price difference is hold for last 24 hours and Neural Network model has only 1 binary outcomes. Therefore, state space is an array 1*25 for all micro agents in portfolio.

3.3.3.2 Actions

Actions Set: [$B1_{it}$, $B2_{it}$, $S1_{it}$, $S2_{it}$, H_{it}]

• $B1_{it}$: Buying action for cryptocurrency i on time t

• $B2_{it}$: Strong buying action for cryptocurrency i on time t

• $S1_{it}$: Selling action for cryptocurrency i on time t

• $S2_{it}$: Strong selling action for cryptocurrency i on time t

• H_{it}: Holding the money for cryptocurrency i on time t

Likely in policy iteration, there are 5 possible actions as strongly buy, buy, sell, strongly and hold the money.

3.3.3.3 Reward

Reward equals to sum of nominal return of all cryptos in portfolio on time t.

CHAPTER IV

RESEARCH DESIGN AND METHOD

4.1 Data Set Preparation

4.1.1 Selection of Coins in Portfolio

In order to enhance computational efficiency and enable a comprehensive comparative analysis of various architecture designs, the portfolio has been strategically composed to include four prominent coins: BTC, DOGE, CHZ and SAND. To select these coins from many on the market, firstly we select 16 coins as following: BTC, ADA, AXS, BNB, CAKE, CHZ, DOGE, DOT, ENJ, ETH, FARM, HNT, LINK, MANA, SAND, UNI. We assessed the market capitalization of the coins and linked it to their level of risk. Specifically, a coin with a higher market capitalization is considered to be less risky compared to a coin with lower capitalization.

Please find below bar chart of capitalization values of these cryptocurrencies in Jan of 2021 by getting Coinmarketcap. [27] In summary, four subgroups were created based on the coin market cap rankings, and one coin was selected from each risk group. For example, the least risky group appears to consist of BTC, ETH, DOT, and ADA coins, and only BTC was selected from this group. BNB, LINK, UNI, and DOGE coins are from the low-risk group, and DOGE was selected from this group. The risky group comprises MANA, ENJ, CHZ, and HNT coins, and CHZ was chosen from this group. Lastly, the highly risky group includes CAKE, FARM, AXS, and SAND coins, and SAND coin was selected from this group.

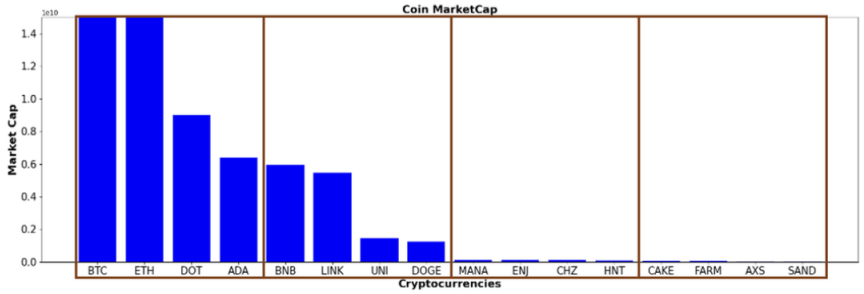

Figure 8: Cryptocurrencies Market Capitalization

4.1.2 Time Period Decision

The selection of an appropriate time period for analysis poses a distinctive challenge due to the inherently volatile nature of the coin market. The market's pronounced fluctuations make it essential to carefully consider the timeframe under examination. To ensure fairness to all stakeholders involved, it becomes necessary to incorporate data from previous years into the experimentation process. This approach acknowledges the significant divergence in the market's performance between 2023 and the preceding year, as visually illustrated for Bitcoin on the Figure 9.[28] Please see Figure A6,A7,A8 on Appendix for line graphs of DOGE, CHZ and SAND close prices. The change that occurred after the first half of 2022 is also evident in these cryptocurrencis. By incorporating historical data from previous years, a more holistic and representative assessment can be conducted, offering a comprehensive understanding of the market dynamics.

Consequently, our dataset is between beginning of 2020 and till to 2022. Through this approach, the effectiveness and robustness of the trained models can be accurately measured and compared, offering valuable insights into their overall performance and

Figure 9: Bitcoin Close Price Chart

potential for success.

While Policy Iteration and Q-learning models were trained using the same time periods, different time periods were used for the Neural Network model. The Neural Network model was pre-trained using data from the year 2020, whereas the 10 training periods created for Policy Iteration and Q-learning models were obtained from the year 2021. The table below shows the training and test periods for Policy Iteration and Q- learning.

Table 1: Train and Test Time Periods

Periods	Train Data Set Time Ranges	Test Data Set Time Ranges
#1	01.01.2021-31.07.2021	01.08.2021-31.08.2021
#2	01.02.2021-31.08.2021	01.09.2021-30.09.2021
#3	01.03.2021-30.09.2021	01.10.2021-31.10.2021
#4	01.04.2021-31.10.2021	01.11.2021-30.11.2021
#5	01.05.2021-30.11.2021	01.12.2021-31.12.2021
#6	01.06.2021-31.12.2021	01.01.2022-31.01.2022
#7	01.07.2021-31.01.2022	01.02.2022-28.02.2022
#8	01.08.2021-28.02.2022	01.03.2022-31.03.2022
#9	01.09.2021-31.03.2022	01.04.2022-30.04.2022
#10	01.10.2021-30.04.2022	01.05.2022-31.05.2022

Summarily, for pioneer neural network model, we used dataset between 01.01.2020 – 31.12.2020 and train and test datasets are randomly select from this period.

We used 10 training periods for Policy Iteration and Q-learning, each spanning 6 months in length, and followed each of them with a 1-month test period as showed on Table 1.

4.1.3 Feature Details and Data Sources

We can categorize the features we have into subgroups as volume indicators, trend indicators, volatility indicators, momentum indicators, macro market indicators, coin-specific features, and other indicators. In Python, the technical analysis library [29] was used for most of the technical indicators, while the remaining features were obtained from [30],[31].

4.1.3.1 Volume Indicators

• Accumulation Distribution Index: It is a technical indicator used to assess the flow of money in and out of a security over a specific time period. It combines price and volume data to gauge buying and selling pressure.

• Chaikin Money Flow Indicator: The alternative version of the Accumulation Distribution Index sums up the Money Flow Volume instead of taking the cumulative total like the traditional Accumulation Distribution Index does. This variation focuses on adding up the Money Flow Values without considering their cumulative progression over time.

• Ease of Movement: The Ease of Movement indicator is a valuable tool for assessing the strength of a trend, even when the trend itself has been identified. It helps to gauge the intensity or magnitude of the trend, providing insights into how strong the trend actually is.

• Force Index: It is summarily measures the pressure of taking action for investors and calculated by multiplying volume with close price difference in sequential periods.

• Money Flow Index: The Ease of Movement indicator serves as an additional

metric for evaluating buying and selling pressure in a market by quantifying positive and negative money flow. It assigns ratings to these flows, allowing for a more comprehensive assessment of the dynamics between buyers and sellers.

• Negative Volume Index Indicator: This metric is associated with the concept of smart money and refers to intelligent investors who are capable of interpreting technicalindicators along with smart money. When this metric shows negativity, it indicates that the smart money group is actively involved.

• On Balance Volume: It cumulatively take sum of volume by comparing close price between time periods. [29],[32].

4.1.3.2 Trend Indicators

• Average Directional Movement Index: The metric is computed by analyzing the difference between consecutive lows, and its negative or positive value indicates the direction of the trend slope.

• Trix Indicator: The Trix indicator is derived by calculating a series of Exponential Moving Averages, starting with a Single Smoothed EMA and progressing to a Triple Smoothed EMA. The final Trix value is obtained by measuring the percentage change in the Triple Smoothed EMA over a single period.[29],[32].

4.1.3.3 Volatility Indicators

• Bollinger Moving Average: Bollinger Bands are valuable tools for traders to identify market volatility and potential breakouts, signaling overbought or oversold conditions, and potential price reversals, helping them make informed trading decisions when combined with other indicators.

• Average True Range: Crucial indicator for assessing price volatility. A larger ATR suggests a higher potential for significant price movements in the cryptocurrency market, indicating the possibility of strong and impactful market activity.

• Donchian Channels: Similar to the Bollinger Bands, the Keltner Channel provides insights into the potential extension of bullish or bearish trends by establishing a 27 relationship between current prices and trading ranges. It serves as a valuable tool for assessing market conditions and determining the potential duration of market seasons.

• Keltner Channel: Unlike Bollinger Bands, the Keltner Channel utilizes Average True Range (ATR) and Exponential Moving Average (EMA) values instead of standard deviation to define its price ranges. This approach provides a distinct methodology for assessing volatility and identifying potential trading opportunities. [29],[32].

4.1.3.4 Momentum Indicators

• Simple Moving Average: The Simple Moving Average (SMA) calculates the average price of an asset during a specific time frame, smoothing price data and identifying trends.

• Percentage price oscillator: It is calculated by taking the difference between two exponential moving averages of different time periods and dividing it by the exponential moving average of the larger period. This calculation provides a measure of the price's rate of change and helps identify the momentum and strength of the current trend.

• Percentage Volume Oscillator: Only difference from percentage price oscillator, it does same calculation from volume based moving average.

• Rate of Change: It is very popular indicator, also known as Momentum, measures the percentage change in price over a specific period. It fluctuates above and below the zero line, providing signals such as centerline crossovers and overbought/oversold readings. Divergences are less reliable. It helps assess price momentum and identify potential trading opportunities.

• Relative Strength Index: It is designed to identify overbought or oversold conditions by evaluating the magnitude of price gains and losses. It measures the speed of price changes and helps determine the momentum in the market. Its primary goal is to capture potential buying or selling opportunities based on the rate at which prices are changing.

• Stochastic RSI: Similar to the traditional Relative Strength Index (RSI) but provides more specific insights about an asset rather than general market outcomes. It helps traders and investors understand the relative strength or weakness of an asset's price compared to its own historical performance

• Ultimate Oscillators: It is enhanced version of momentum oscillators by focusing on larger time period and if it increases we interpret this as buying pressure is strong and decreasing shows vice versa. [29],[32].

4.1.3.5 Other Indicators

• Daily return: It is well known metric and basically, it gives the percentage of close price increase or decrease which is acquired from [30].

• Day flag: It is a feature shows the transaction day number in week. This feature is acquired from date column from [30].

4.1.3.6 Macro Market Indicators

These features are composed since to get information about macro market and make model to find relation this with macro dynamics in the market.

• Bitcoin Dominance Rate Feature: It shows bitcoin ratio on the market and it is kept to understand how Bitcoin dominates market and maybe to catch the effect of altcoins. [30]

• Ethereum Average Gas Cost: The Ethereum Average Gas Cost refers to the average amount of gas consumed per transaction on an hourly basis within the Ethereum network. Gas is the unit of measurement used to quantify the computational effort

required to perform operations on the Ethereum blockchain. By calculating the average gas cost, users and developers can assess the typical gas expenditure for executing transactions and smart contracts, providing insights into the network's efficiency and transaction costs. [31]

• Bear and Bull Flag: This feature shows if period is in bear or bull, this feature is created from close prices get from [30].

4.1.3.7 Coin Specific Features Indicators

Features specific to coins and which is on more micro level.

• Close Price: close prices of cryptocurrencies on hourly basis.

• Volume Rate: typically refers to the total rate of coins traded for a specified asset within a given time period in the market.

• Resistance: Resistance refers to a price level in a market where an asset's upward movement is often met with selling pressure, causing the price to face difficulty in surpassing that level. It is identified by analyzing historical price data, particularly the close prices, to determine the point at which the asset has historically struggled to sustain higher prices. This feature has been obtained through feature engineering from close price feature.

• Symbol: To make the model aware of the specific coin it is processing, we include the symbol. [30]

Close price, Bollinger moving average, Simple moving average and Bear&Bull flag features are used in proposed policy iteration model, the rest of features were used in pioneer neural network model.

4.2 Experimental Setup

We have developed two novel frameworks for implementing value iteration and Q-learning algorithms. These frameworks are designed to integrate with a pioneering neural network. In simpler terms, we first create a neural network and then utilize its

outputs to feed into our proposed models. We will explain in this section, sequentially for the neural network, policy iteration, and Q-learning algorithms, how model setups are established.

4.2.1 Pioneer Neural Network Model

Our objective was to develop an enhanced prediction model by leveraging the neural network structure, thus augmenting the performance of our optimization models. By incorporating neural networks into our framework, we sought to leverage their powerful capabilities for pattern recognition and predictive analysis. This approach allows us to harness the potential of neural networks in improving the accuracy and effectiveness of our optimization models. Through this integration, we aimed to unlock new insights and achieve better predictions, ultimately enhancing the overall performance of our models.

4.2.1.1 Target Definition

The Target Definition in this context is developed through a methodical analysis. Recognizing the difficulty in predicting the next hour's price accurately, the target is formulated as a binary parameter which will be chosen for a time period in future.

Considering the challenges in forecasting within the cryptocurrency domain, our focus lies in achieving a balanced dataset. Given that the largest rate in the table accounts for 42%, we aim to maintain a more equitable distribution for our binary target. As a result, a target percentage of 5% is selected for this model, along with a time period of the subsequent 36 hours. By incorporating these choices, we aim to mitigate the inherent difficulties of cryptocurrency forecasting and promote a more balanced and reliable approach.

To sum up, if target value equals to 1 that means close price will be higher 5% of current time close price in 36 hours.

Table 2: Target percent distribution for different next time setups

Target Percent	Next 3hrs	Next 6 hrs	Next 12 hrs	Next 24 hrs	Next 36 hrs
20%	0.2%	6%	1.5%	3.5%	5.6%
10%	1.4%	3.2%	6.7%	13%	19%
5%	6%	12%	22%	34%	42%

4.2.1.2 Neural Network Model Design

To ensure the effectiveness of our dynamic and Q-learning models in the years 2021 and 2022, we have collected the training, validation, and test datasets exclusively from the year 2020. This deliberate choice allows us to align the data with the target timeframe of our models.

The dataset has been split into training, testing, and validation sets, with the proportions of 60%, 30%, and 10% respectively, ensuring a comprehensive evaluation of the model's performance.

Here are the respective numbers of observations present in each dataset, as provided below:

Table 3: Train, Test and Validation Dataset Summary of Pioneer NN Model

	Train dataset	Test dataset	Validation dataset
Number of observations	52,303	26,152	8,717
Percentage of target	39.9%	39.5%	39.6%

After train test and validation split, different Neural Network structures are experienced and best responsive framework is chosen. Moreover, a total of 18 iterations were conducted, consisting of 3 epoch sizes, 3 batch sizes, and 2 optimizers. The number of layers was determined based on the initial iterations, and the best-performing architecture was selected.

In order to fine-tune the model's hyperparameters, we implement the following

settings.

Table 4: Hyperparameter Tuning Settings for Pioneer Neural Network Model

Hyper Parameter Tunes	Parameter Ranges
Epoch size	10,20,50
Batch size	100,150,200
Optimizer	Rmsprop, Adam

After careful evaluation of the F1 score and accuracy score, the setting numbers with the highest F1 and accuracy scores are 13 and 14 and setting 15 has the highest precision. Considering F1 values, since it is weighted metric of recall and precision, we decided to choose between settings 13 and 14. We prioritize precision and, therefore, selected setting 13, which has a higher precision between 13 and 14. In summary, we chose a model with an epoch size of 50, a batch size of 100, and an optimizer of rmsprop, and then used it on training datasets of proposed models. Please see Table A1 on Appendix for result of validation dataset.

Table 5: Hyperparameter Tuning Results for Test Dataset of NN model

	Epoch Size	Batch Size	Optimizer	F1	Accuracy	Precision	Recall
1	10	100	Rmsprop	0.7	0.7	0.59	0.81
2	10	150	Rmsprop	0.73	0.73	0.66	0.68
3	10	200	Rmsprop	0.71	0.71	0.62	0.7
4	10	100	Adam	0.7	0.71	0.67	0.53
5	10	150	Adam	0.45	0.6	0.001	0.001
6	10	200	Adam	0.65	0.67	0.67	0.36
7	20	100	Rmsprop	0.77	0.78	0.76	0.65
8	20	150	Rmsprop	0.78	0.78	0.72	0.74
9	20	200	Rmsprop	0.77	0.77	0.78	0.6
10	20	100	Adam	0.76	0.77	0.73	0.7
11	20	150	Adam	0.7	0.71	0.66	0.56
12	20	200	Adam	0.74	0.74	0.72	0.57
13	50	100	Rmsprop	0.82	0.82	0.8	0.74
14	50	150	Rmsprop	0.82	0.82	0.77	0.79
15	50	200	Rmsprop	0.8	0.81	0.87	0.61
16	50	100	Adam	0.81	0.81	0.81	0.7
17	50	150	Adam	0.79	0.79	0.77	0.68
18	50	200	Adam	0.82	0.82	0.86	0.66

4.2.2 Policy Iteration and Q-Learning Models

4.2.2.1 Constraints and Assumptions

In this section of the report, we will outline the assumptions and constraints that were considered during the analysis.

- Hourly data is utilized, and it is assumed that every action is taken at the last minute of the respective hour. The models are not permitted to intervene in the portfolio during the duration of one hour.

- Take-profit and stop-loss limits are set at 5% in relation to the weighted buying cost of the respective asset. If the price increases or decreases by 5%, the corresponding action is triggered.

- Transactions between different assets are not permitted. For example, it is not possible to convert CHZUSDT to BTCUSDT directly. One asset can be

converted to USDT, or vice versa.

- The maximum investment allowed for a single asset is limited to $10. Any investment beyond this amount is not permitted.

4.2.2.2 Policy Iteration Model Details

First 3 months are chosen for hyperparameter tuning, please see time periods as following:

- Train1: 01-01-2021 - 30-06-2021
- Train2: 01-02-2021 – 31-07-2021
- Train3: 01-03-2021 – 31-08-2021

Proposed policy iteration models are tunned by consideration of discount rates for 0.65, 0.75, 0.85. Please find below ROI outcomes of these tunes.

Table 6: Hyperparameter Tuning Settings for Policy Iteration Model

Discount Rates	Train 1	Train 2	Train 3	ROI Avg	ROI Sigma
0,65	0,92	1,07	1,05	1,01	0,08
0,75	3,4	1,4	1,14	1,98	1,24
0,85	1,8	0,94	1,3	1,35	0,43

There is a trade of between ROI average and ROI standard deviation. That's why it is a bit challenging to decide the best tuning. Since 0.01 increase is really low, 1st setting is eliminated even though its ROI sigma is the smallest one. Second setting is also not chosen even though its ROI average value is the highest one , its deviation value is really high. As a result, 3rd setting is seem better between these 2 settings and seem fair even though it has 0,4 deviation.

4.2.2.3 Q Learning Model Details

Hyperparameter tuning is applied to DQN model and selected settings are applied to all Q-network models. Please find on Table 6 tunning parameters for proposed DQN and these ROI values for train datasets 1,2,3 and their ROI average and deviation.

- Epoch size: 20,40
- Gamma: 0,85;0,95
- Alpha: 0,0001; 0,005

Table 7: Hyperparameter Tuning Settings for Q-Learning Model

	Epoch	Gamma	Alpha	Train 1	Train 2	Train 3	ROI Avg	ROI Sigma
1	20	0.95	0.0001	1,5	1,13	1,14	1,26	0,21
2	20	0.95	0.005	2,8	0,99	1,3	1,70	0,97
3	20	0.85	0.0001	2,55	1,31	1,04	1,63	0,81
4	20	0.85	0.005	2,12	1,5	1,16	1,59	0,49
5	40	0.95	0.0001	2,2	1,14	1,4	1,58	0,55
6	40	0.95	0.005	1,45	1,04	1,13	1,21	0,22
7	40	0.85	0.0001	2,8	1,21	1,14	1,18	0,94
8	40	0.85	0.005	1,05	1,6	0,96	1,20	0,35

The table above highlights the highest ROI values, with different model parameters demonstrating the best performance during each training period. The decision is based on considering the average and standard deviation values of ROI. While the second setting appears to have the highest ROI value, the first setting exhibits the lowest ROI deviation. Despite the second setting having the highest ROI mean value, its deviation is around 1. Given our objective of finding a model with high profit and low volatility, we ultimately prefer the first setting. It boasts an impressive ROI average of 1.26, which is significant for a one-month period, and its Sigma value is exceptionally small.

After tuning the DQN model we continued with epoch size 20, Discount value with 0.95 and alpha value with 0.0001. DDQN, DDDQN also is created according to this setting for proposed model.Please see Figure A9-A18 on Appendix, how loss value is getting decrease during epochs for DQN with this setting on all training periods.

CHAPTER V

RESULT

5.1 The Evaluation of Policy Iteration Models

Firstly, we would like to evaluate our proposed models within their algorithm groups. PI 1 refers to our benchmark model while PI 2 refers to proposed model for Policy Iteration Algorithms on. Please see on Table 7 Policy Iteration models ROI values for test periods.

When we look at each period individually, it becomes challenging to determine the winning model. For instance, in Period 1, PI 2 has a better ROI value, while in Period 4, PI 1 shows a higher ROI value. Therefore, it makes more sense to consider the average and deviation of the ROI for all periods. In this regard, when examining the results, we find that the PI 2 model performs better with a 3% ROI mean and a 6% ROI deviation. This conclusion suggests that PI 2 appears to be a better option both in terms of returns and to hedge against volatility.

Table 8: ROI values for Policy Iteration Models

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
PI1	1,1	0,94	1,13	1,13	0,97	0,94
PI2	1,2	1,03	1,06	1,03	0,97	0,99

	Period 7	Period 8	Period 9	Period 10	ROI Avg	ROI Sigma
PI1	0,96	1,04	0,95	1,002	1,02	0,08
PI2	1,01	1,01	0,99	1,01	1,03	0,06

To better visualize the volatility difference, the line graph below has been provided. Please review it.

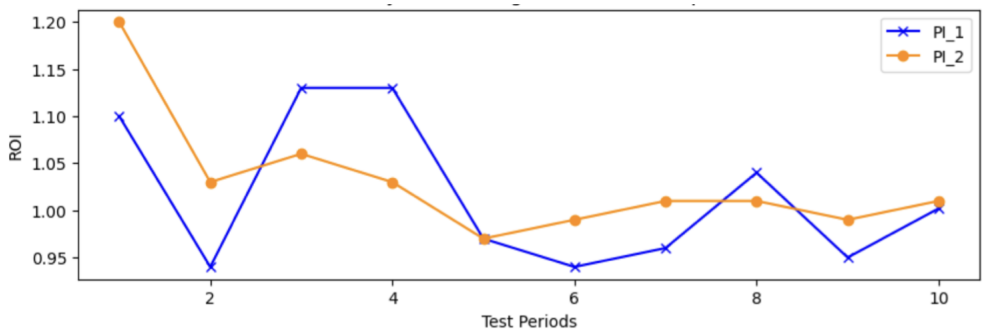

Figure 10: ROI values of Policy Iteration Algorithms for Test Periods

In summary, we can make the following assessment: The PI 1 model takes more aggressive decisions and can achieve ROI values as high as 13% in Period 2. On the other hand, the PI 2 model appears to be more cautious and refrains from making significant investment decisions unless confident. PI 2 observes a maximum 6% increase and a 3% loss, whereas PI 1 experiences a maximum 13% increase and a 6% loss. If the investor has a risk-seeking profile, they could continue with the PI 1 model. However, our intention was to create a model with low volatility and profitability outcomes. The reason for this is that high volatility is one of the significant drawbacks of cryptocurrency exchanges. Hence, all our hyperparameter tuning and model evaluations were based on this criteria.

The main difference between these two models stems from their state definitions. PI 1 relies solely on weight information, while PI 2 considers market indicators, the predictions of the Neural Network model, and weight values. Having more information about the environment seems to have protected the PI 2 model from more significant downturns. Additionally, on Figure 11, which displays the maximum loss values for each period, confirms this theory. In PI 1, larger loss values are observed, whereas these values are reduced even further in PI 2, providing further evidence for our hypothesis.

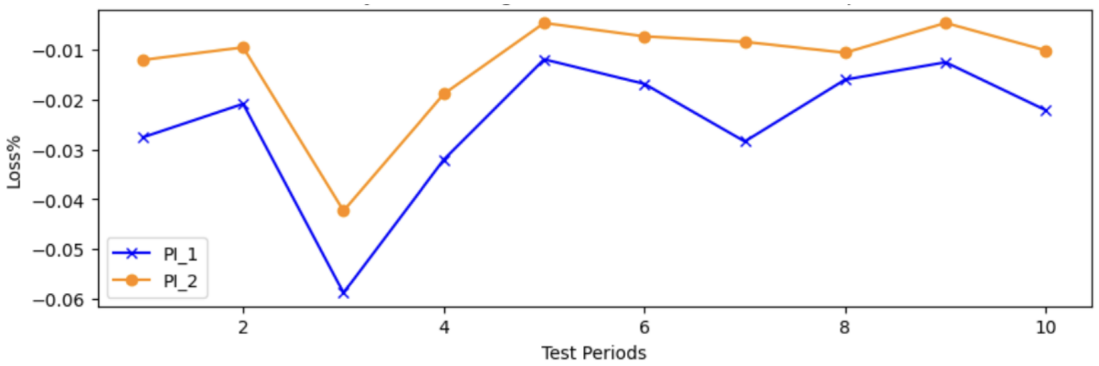

Figure 11: Max Loss Percent Values for Policy Iteration Algorithms

5.2 The Evaluation of Q-Learning Models

Periodically, the Q1 models may exhibit very high ROI values; however, when considering the overall average and deviation, we can conclude that the Q2 - DQN model is the best model with a 2% ROI mean value. As for the ROI standard deviation, both the Q2 - DQN and Q2 - DDDQN models perform the best with a value of 3%.

Table 9: ROI values for Q-Learning Models

		Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Q1	DQN	1,15	0,71	1,3	0,9	0,86	0,76
	DDQN	1,14	0,74	1,17	0,9	0,93	0,82
	DDDQN	1,05	0,75	1,12	0,97	0,97	0,87
Q2	DQN	1,05	0,99	1,03	1,07	0,99	0,99
	DDQN	1,06	0,98	1,13	1,02	0,99	0,99
	DDDQN	1,01	0,99	1,08	0,99	0,99	1,01

		Period 7	Period 8	Period 9	Period 10	ROI Avg	ROI Sigma
Q1	DQN	1,2	1,4	0,7	1,1	1,01	0,25
	DDQN	1,02	1,3	0,77	1,03	0,98	0,18
	DDDQN	1,01	1,01	0,75	1,02	0,97	0,16
Q2	DQN	1,01	1,04	0,98	1,01	1,02	0,03
	DDQN	0,99	1,03	0,99	0,96	1,01	0,05
	DDDQN	1,01	1,01	0,99	1,01	1,01	0,03

We observe that the Q1 models exhibit very high ROI standard deviation values, such as 25% or 18%. As a result, we can say that Q2 models are significantly better

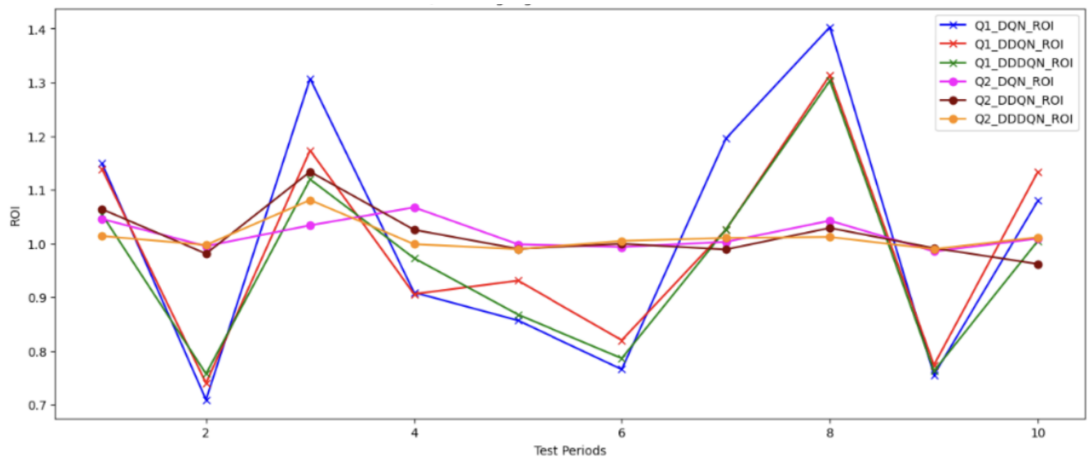

Figure 12: ROI Values for Q-Learning algorithms

due to their profitable average and low volatility. The difference in volatility can be seen more clearly in the line graph below. Lines with 'x' belongs to Q1 models and it makes big move while lines with 'o' makes more small changes which belongs to Proposed Q- Learning models.

In the proposed Q2 models, we observe a more cautious approach compared to the Policy Iteration models. When examining the architectural differences, there could be two reasons why the Q2 models perform better. Firstly, similar to the Policy Iteration models, the Q2 models may have more information about the market due to being fed with data from the Neural Network model. This is because the Q2 model is receiving additional inputs from the Neural Network.

The second reason could be that the Q2 model consists of multiple micro-agents, and their decisions are not directly taken but are selected by the macro-agent based on high rewards. This may lead to the model behaving more conservatively, as it considers a broader range of possible actions and outcomes before making decisions.

Overall, these two factors may contribute to the Q2 models' superior performance and more cautious behavior compared to the Q1 models.

5.3 The Evaluation of all models

When we look at all models collectively, the proposed Policy Iteration model (PI 2) stands out with the highest ROI mean value at 3%, while the proposed DQN and proposed DDDQN models (Q2) have the lowest ROI standard deviation values. Considering that there is not a significant difference between a 2% ROI and a 3% ROI, and the fact that the Q2-DQN model has the lowest ROI standard deviation at 3%, it can be selected as the best model.

Additionally, when examining Table 9, we notice a significant trade-off between mean and standard deviation values. While there is not a considerable difference in ROI mean values, there is a substantial gap between the ROI deviation values, ranging from 25% to 3%. This indicates that the models may take significant risks even to increase ROI by just 1%. Furthermore, we anticipate that the proposed models generally the benchmarks due to their access to more market information.

Table 10: ROI values of all models

		Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Q1	DQN	1,15	0,71	1,3	0,9	0,86	0,76
	DDQN	1,14	0,74	1,17	0,9	0,93	0,82
	DDDQN	1,05	0,75	1,12	0,97	0,97	0,87
Q2	DQN	1,05	0,99	1,03	1,07	0,99	0,99
	DDQN	1,06	0,98	1,13	1,02	0,99	0,99
	DDDQN	1,01	0,99	1,08	0,99	0,99	1,01
PI 1		1,1	0,94	1,13	1,13	0,97	0,94
PI 2		1,2	1,03	1,06	1,03	0,97	0,99

		Period 7	Period 8	Period 9	Period 10	ROI Avg	ROI Sigma
Q1	DQN	1,2	1,4	0,7	1,1	1,01	0,25
	DDQN	1,02	1,3	0,77	1,03	0,98	0,18
	DDDQN	1,01	1,01	0,75	1,02	0,97	0,16
Q2	DQN	1,01	1,04	0,98	1,01	1,02	0,03
	DDQN	0,99	1,03	0,99	0,96	1,01	0,05
	DDDQN	1,01	1,01	0,99	1,01	1,01	0,03
PI 1		0,96	1,04	0,95	1,002	1,02	0,08
PI 2		1,01	1,01	0,99	1,01	1,03	0,06

CHAPTER VI

CONCLUSION

In this study, two novel models with enhanced architectures have been proposed for the Policy Iteration and Q learning algorithms under the Reinforcement Learning framework. To compare these models within their respective algorithm groups, primitive models have been created for both algorithm types as well. In Chapter 5 - Results, the reason why the proposed models outperform the other models has been evaluated by examining the ROI mean and ROI standard deviation values. As a result, it has been observed that the new designs, with access to more market information and the transformation from a single-agent flow to a multi-agent flow, have yielded positive results.

In this study, the opportunity to compare the Policy Iteration and Q learning algorithms with each other was also present. As a result, it was concluded that the Q learning algorithms take bigger risks - especially evident in the benchmark Q learning model, where the ROI reaches 25%. While there may not be significant differences in the average ROI returns, when looking at the deviation values, we can state that the benchmark Q learning models have been unsuccessful.

This study is pioneering in comparing both Policy Iteration and Q learning algorithms with each other, and at the same time, it opens new possibilities with the introduction of novel model proposals and their designs. Additionally, both proposed models involve running a neural network model, which provides a different perspective to the analysis.

Apart from these, these proposed models can be further enhanced by increasing the number of cryptocurrencies in the portfolio, increasing the market indicators, or

selecting different indicators. Moreover, experimenting with different neural network architectures could lead to more complex and market-savvy model designs, both in terms of the pioneer neural network and Q learning neural networks. This opens the possibility of exploring more sophisticated model structures that better understand the market dynamics.

CHAPTER VII

APPENDIX A

Table 11: Validation Data Set Hyperparameter tuning Results

	Epoch Size	Batch Size	Optimizer	F1	Accuracy	Precision	Recall
1	10	100	Rmsprop	0.69	0.69	0.58	0.8
2	10	150	Rmsprop	0.73	0.73	0.65	0.66
3	10	200	Rmsprop	0.71	0.71	0.62	0.7
4	10	100	Adam	0.7	0.7	0.67	0.5
5	10	150	Adam	0.45	0.6	0.001	0.001
6	10	200	Adam	0.64	0.7	0.66	0.34
7	20	100	Rmsprop	0.77	0.77	0.75	0.64
8	20	150	Rmsprop	0.78	0.78	0.72	0.73
9	20	200	Rmsprop	0.76	0.76	0.76	0.58
10	20	100	Adam	0.76	0.76	0.72	0.65
11	20	150	Adam	0.7	0.7	0.65	0.54
12	20	200	Adam	0.73	0.74	0.72	0.56
13	50	100	Rmsprop	0.82	0.82	0.79	0.74
14	50	150	Rmsprop	0.82	0.82	0.76	0.78
15	50	200	Rmsprop	0.8	0.8	0.87	0.6
16	50	100	Adam	0.81	0.81	0.8	0.7
17	50	150	Adam	0.79	0.79	0.67	0.67
18	50	200	Adam	0.82	0.82	0.85	0.66

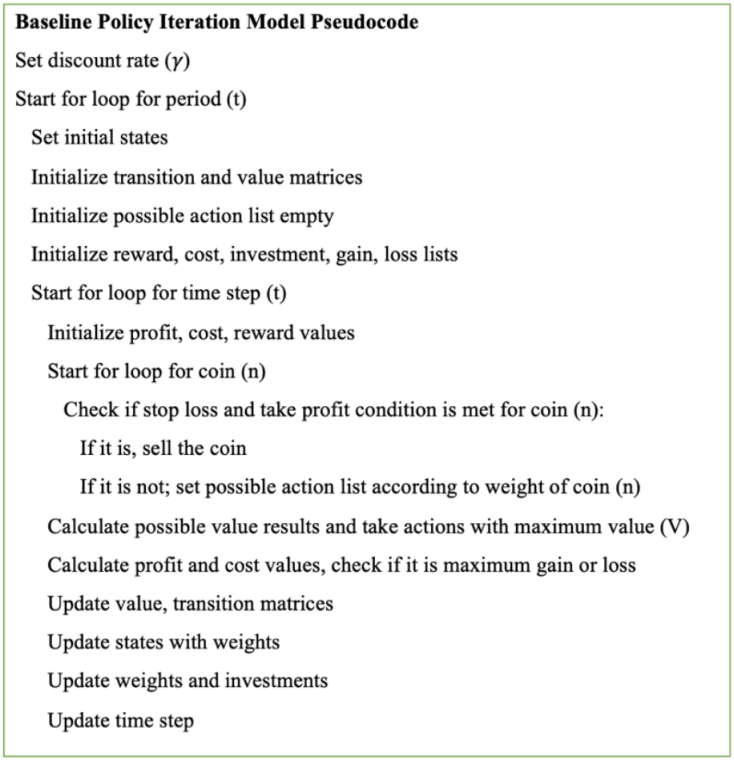

Figure 13: Pseudocode for baseline policy iteration model

Proposed Policy Iteration Model Pseudocode

Set discount rate (γ)

Start for loop for period (t)

  Set initial states for cryptos separately

  Initialize transition and value matrices for all coins

  Initialize possible action list empty for all coins

  Initialize reward, cost, investment, gain, loss lists

  Start for loop time step (t)

    Initialize profit, cost, reward values

    Start for loop for coin (n)

      Check if stop loss and take profit condition is met for coin (n):

        If it is, sell the coin (n)

        If it is not; set possible action list according to weight of coin (n)

      Calculate possible value results and take actions with maximum value (V) for coin (n)

      Calculate profit and cost values, check if it is maximum gain or loss for coin (n):

      Update value, transition matrices for coin (n):

      Update states with market indicators, NN outcomes and weight for coin (n)

      Update weights and investments for coin (n):

    Calculate total reward, total costs

    Update time step

Figure 14: Pseudocode for proposed policy iteration model

Benchmark DQN Algorithm Pseudeocode

Define main DQN model structure and set deepcopy of target model

Set initial states and weights

Set epsilon (ε) and discount rate (γ)

Set total rewards, costs, losses, main model and target model lists for all agents

Start epoch (e) for loop:

  Start for loop for time step (t):

    Initialize states and profits for all coins

    Start for loop for coin (n):

      Call DQN outcome for agent n and keep actions

      Check if stop loss and take profit condition is met for coin (n):

        If it is; sell the coin

        If it is not:

          If random number is bigger than epsilon:

            Get action with highest Q value

          Else:

            Randomly select and action

      Calculate Reward and Append memory with beginning states, ending states, action, global agent reward

      If memory length is longer or equal to 200:

        Create batches with 20 lenghts from memory and shuffle them

        Find Q values for mini matches and keep loss values

        Update all network according to loss values

        In every 20 step update target network

      If epsilon is bigger than 0.1; decrease this with 0,0001

      Update total reward, cost values and update time step

Figure 15: Pseudocode for baseline DQN model

Proposed DQN Algorithm Pseudeocode

Define main DQN model structure and set deepcopy of target model

Set initial states and weights

Set epsilon (ε) and discount rate (γ)

Set total rewards, costs, losses, main model and target model lists for all agents

Start epoch (e) for loop:

  Start for loop for time step (t):

    Initialize states and profits for all coins

    Start for loop for agent (n):

      Start for loop for coin (m) in agent (n) cluster:

        Call DQN outcome for agent n and keep actions

        Check if stop loss and take profit condition is met for coins in agent n's responsibility:

          If it is; sell the coin

          If it is not:

            If random number is bigger than epsilon:

              Get relevant action for coin (m) from action list

            Else:

              Randomly select an action for coin (m)

      Check all possible actions for all agents and choose winner and second agents with the highest reward by macro agent

      Decide actions and update all weights and actions parameters

      Calculate Reward and Append memory with beginning states, ending states, action, global agent reward

      If memory length is longer or equal to 200:

        Start for loop for agent (n):

          Create batches with 20 lenghts from memory and shuffle them

          Find Q values for mini matches and keep loss values

          Update relevant DQN network according to loss values

          In every 20 step update target network

      If epsilon is bigger than 0.1; decrease this with 0,0001

      Update total reward, cost values and update time step

Figure 16: Pseudocode for proposed DQN model

Figure 17: Ethereum close price chart

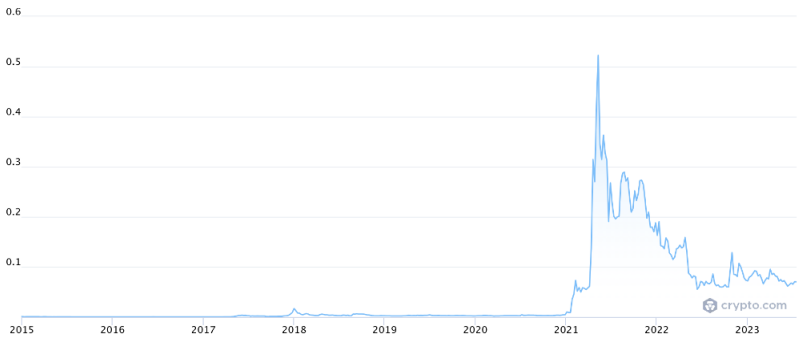

Figure 18: Dogecoin close price chart

Figure 19: Sandbox close price chart

Figure 20: Chiliz close price chart

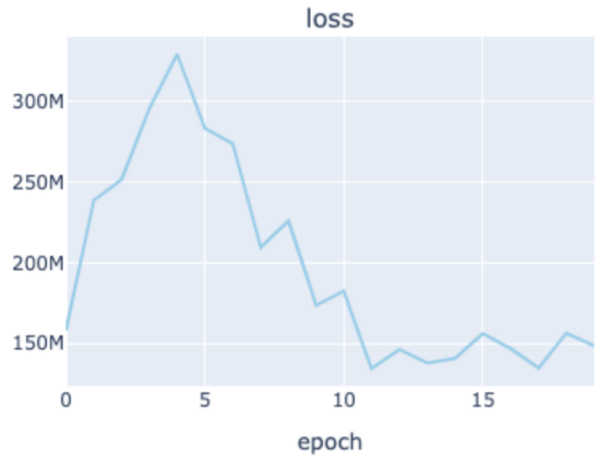

Figure 21: MSE loss graph of proposed DQN for training period 1

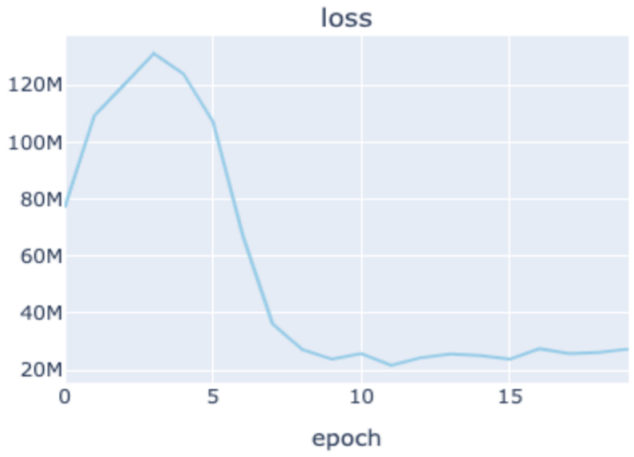

Figure 22: MSE loss graph of proposed DQN for training period 2

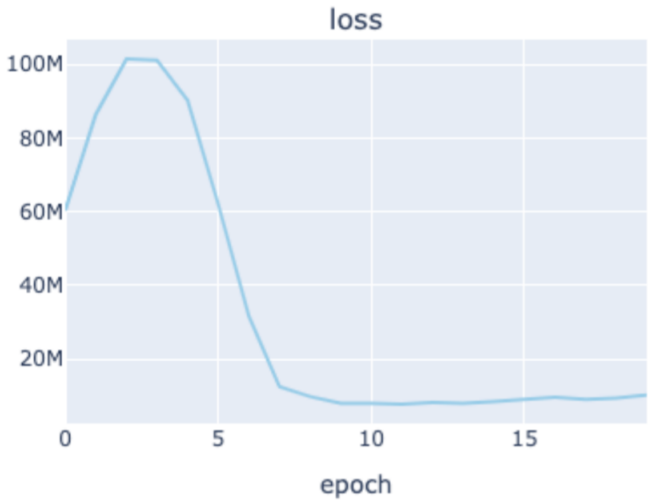

Figure 23: MSE loss graph of proposed DQN for training period 3

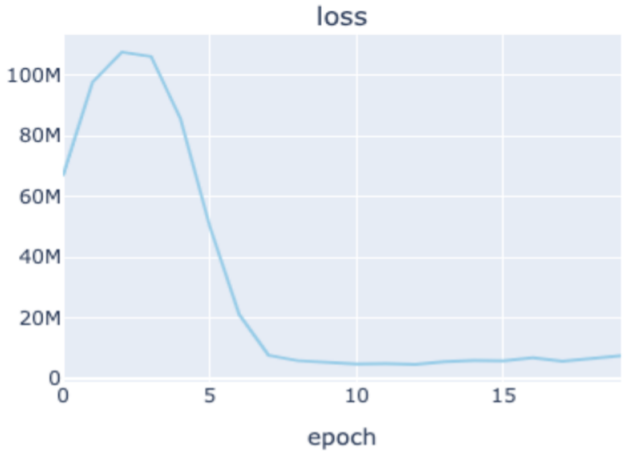

Figure 24: MSE loss graph of proposed DQN for training period 4

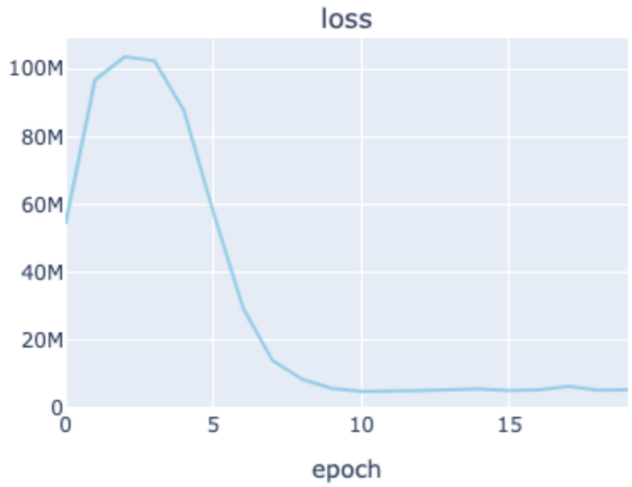

Figure 25: MSE loss graph of proposed DQN for training period 5

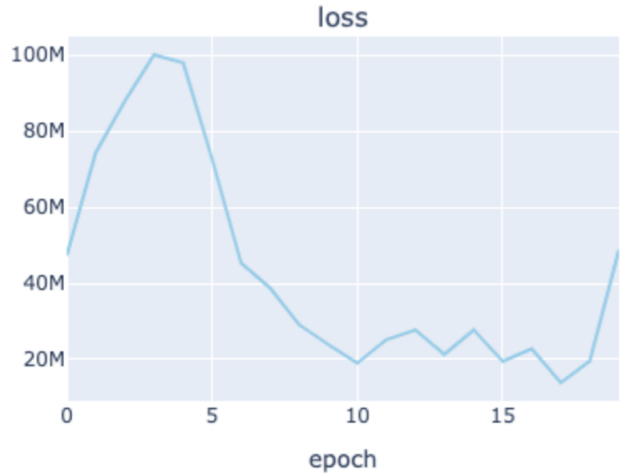

Figure 26: MSE loss graph of proposed DQN for training period 6

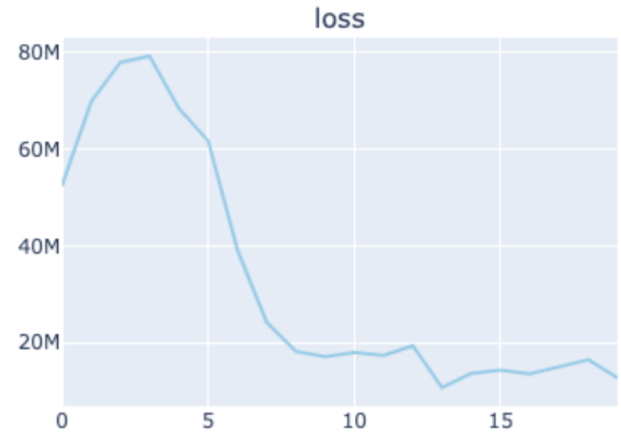

Figure 27: MSE loss graph of proposed DQN for training period 7

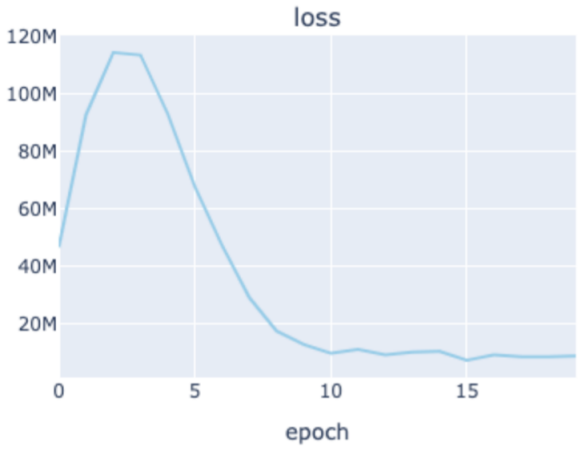

Figure 28: MSE loss graph of proposed DQN for training period 8

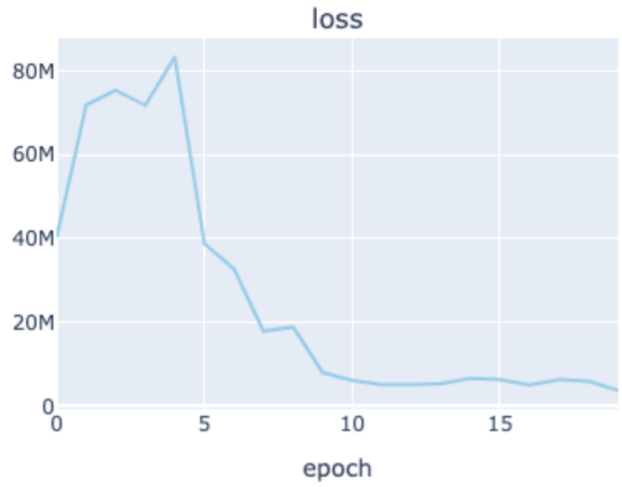

Figure 29: MSE loss graph of proposed DQN for training period 9

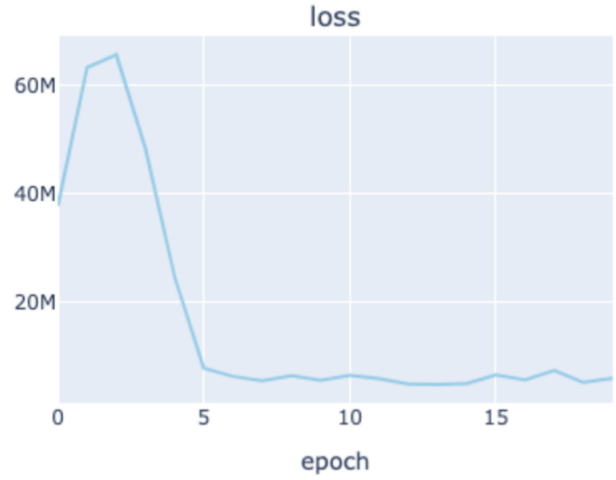

Figure 30: MSE loss graph of proposed DQN for training period 10

REFERENCES

[1] M. Watorek, S. Drożdż, J. Kwapień, L. Minati, P. Oświecimka, and M. Stanuszek, "Multiscale characteristics of the emerging global cryptocurrency market," *Physics Reports*, vol. 901, pp. 1–82, 2021.

[2] Anonymous, "Cryptocurrencies: What are they?." https://www.schwab.com/learn/story/ cryptocurrencies-what-are-they. Accessed on June 13, 2022.

[3] T. V. H. Nguyen, B. T. Nguyen, T. C. Nguyen, and Q. Q. Nguyen, "Bitcoin return: Impacts from the introduction of new altcoins," *Research in International Business and Finance*, vol. 48, pp. 420–425, 2019.

[4] K. H. Al-Yahyaee, W. Mensi, H.-U. Ko, S.-M. Yoon, and S. H. Kang, "Why cryptocurrency markets are inefficient: The impact of liquidity and volatility," *The North American Journal of Economics and Finance*, vol. 52, p. 101168, 2020.

[5] M. Poongodi, A. Sharma, V. Vijayakumar, V. Bhardwaj, A. P. Sharma, R. Iqbal, and R. Kumar, "Prediction of the price of ethereum blockchain cryptocurrency in an industrial finance system," *Computers & Electrical Engineering*, vol. 81, p. 106527, 2020.

[6] L. Alessandretti, A. ElBahrawy, L. M. Aiello, and A. Baronchelli, "Anticipating cryptocurrency prices using machine learning," *Complexity*, vol. 2018, pp. 1–16, 2018.

[7] H. Jang and J. Lee, "An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information," *Ieee Access*, vol. 6, pp. 5427–5437, 2017.

[8] S. Karasu, A. Altan, Z. Saraç, and R. Hacioğlu, "Prediction of bitcoin prices with machine learning methods using time series data," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, May 2018.

[9] K. Rathan, S. V. Sai, and T. S. Manikanta, "Crypto-currency price prediction using decision tree and regression techniques," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 190–194, IEEE, 2019.

[10] S. Alonso-Monsalve, A. L. Suárez-Cetrulo, A. Cervantes, and D. Quintana, "Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators," *Expert Systems with Applications*, vol. 149, p. 113250, 2020.

[11] A. A. Oyedele, A. O. Ajayi, L. O. Oyedele, S. A. Bello, and K. O. Jimoh, "Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction," *Expert Systems with Applications*, vol. 213, p. 119233, 2023.

[12] N. Maleki, A. Nikoubin, M. Rabbani, and Y. Zeinali, "Bitcoin price prediction based on other cryptocurrencies using machine learning and time series analysis," *Scientia Iranica*, vol. 30, pp. 285–301, 2020.

[13] M. M. Patel, S. Tanwar, R. Gupta, and N. Kumar, "A deep learning-based cryptocurrency price prediction scheme for financial institutions," *Journal of Information Security and Applications*, vol. 55, p. 102583, 2020.

[14] I. A. Hashish, F. Forni, G. Andreotti, T. Facchinetti, and S. Darjani, "A hybrid model for bitcoin prices prediction using hidden markov models and optimized lstm networks," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 721–728, IEEE, 2019.

[15] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and arima models for time series forecasting," *Applied Soft Computing*, vol. 11, no. 2, pp. 2664–2675, 2011.

[16] Y. Patel, "Optimizing market making using multi-agent reinforcement learning," *arXiv preprint arXiv:1812.10252*, 2018.

[17] J. W. Lee and J. O, "A multi-agent q-learning framework for optimizing stock trading systems," in *International Conference on Database and Expert Systems Applications*, pp. 153–162, Springer Berlin Heidelberg, 2002.

[18] Y.-H. Chang and M.-S. Lee, "Incorporating markov decision process on genetic algorithms to formulate trading strategies for stock markets," *Applied Soft Computing*, vol. 52, pp. 1143–1153, 2017.

[19] C. Betancourt and W.-H. Chen, "Deep reinforcement learning for portfolio management of markets with a dynamic number of assets," *Expert Systems with Applications*, vol. 164, p. 114002, 2021.

[20] I. Estalayo, J. Del Ser, E. Osaba, M. N. Bilbao, K. Muhammad, A. Gálvez, and A. Iglesias, "Return, diversification and risk in cryptocurrency portfolios using deep recurrent neural networks and multi-objective evolutionary algorithms," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 755–761, IEEE, 2019.

[21] B. Jang, M. Kim, G. Harerimana, and J.-W. Kim, "Q-learning algorithms: A comprehensive classification and applications," *IEEE access*, vol. 7, pp. 133653–133667, 2019.

[22] S. Tanwar, "Bellman equation and dynamic programming," February 10 2022. Retrieved from https://medium.com/analytics-vidhya/bellman-equation-and-dynamic-programming-773ce67fc6a7.

[23] M. Sewak, *Deep Reinforcement Learning - Frontiers of Artificial Intelligence*. Springer, 2019.

[24] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[25] C. Shyalika, "A beginners guide to q-learning." Medium. (15. 11. 2019.), 2019. Retrieved from https://12ft.io/proxy.

[26] B. Peng, Q. Sun, S. E. Li, D. Kum, Y. Yin, J. Wei, and T. Gu, "End-to-end autonomous driving through dueling double deep q-network," *Automotive Innovation*, vol. 4, pp. 328–337, 2021.

[27] "Historical snapshot - 01 january 2021 — coinmarketcap," n.d.

[28] "Crypto.com." Crypto Website, 2023.

[29] T. A. L. in Python, "Technical analysis library in python 0.1.4 documentation," 2023.

[30] "Binance api." Binance Website, 2023.

[31] "Flipside." Flipside Website, 2023.

[32] "School.stockcharts." School Stockchart Website, 2023.

VITA

Ceren Altok graduated from Bilecik Fen High School in 2011 and obtained her degree from the Middle East Technical University's Industrial Engineering department in 2017. Since her graduation, she has worked in various sectors in roles such as Data Analyst and Data Scientist. Currently, she is employed as a Senior Data Scientist at Beymen Group.