An Application of Deep Reinforcement Learning for Algorithmic Trading in the Stock Market

1st Sina Bahrami
Department of Management Sciences
University of Waterloo
Waterloo, Canada
sina.bahrami@uwaterloo.ca

***Abstract*—The use of AI in financial markets (FinTech) has been rapidly expanding. This field is particularly interesting because it provides significant advantages over traditional trading methods by utilizing vast amounts of data and integrating various features such as economic indicators, political news, social trends, and more.**

Prior studies have been conducted on the application of reinforcement learning for adaptive stock trading. In this project, a deep reinforcement learning (DLR) model is implemented by integrating technical indicators, and its performance is evaluated on a separate test data.

This paper explores the application of Deep Reinforcement Learning (DRL) in the stock market, focusing on the utilization of technical indicators for predictive analysis. This study has implemented a deep reinforcement learning model, specifically a Deep Q-Network (DQN), and has assessed its performance using separate test data. The study reviews existing literature and tools available in FinTech, addressing challenges like variable selection and model overfitting

***Index Terms*—Deep Reinforcement Learning, DRL, Deep Q-Learning, DQN, Stock Market, AI, Algorithmic Trading**

I. Introduction

Financial markets are like intricate puzzles where the pieces—individual buying and selling actions—continually influence one another, leading to unpredictable outcomes. These markets are sensitive to even the smallest changes, meaning different starting points can lead to wildly different outcomes, much like how hard it is to predict the weather accurately. Cecconi has provided a comprehensive overview about the application of AI in financial markets without going through the theory and algorithms. He has reviewed the current role of AI in financial markets then has investigiated how it can be used to spot and stop fake news, predict economic trends and create AI agents making intelligent decisions on their own and bulding models that predict people's behavior [1].

Predicting financial time series, particularly in the stock market, remains a compelling field for researchers, investors, and market analysts due to its potential for wealth augmentation. The financial market is a complex system where multiple asset classes such as stocks, bonds, and commodities are exchanged, with equity markets being of paramount interest for predicting market trends, returns, and managing portfolios. Market analysis predominantly revolves around technical and fundamental analysis approaches. The former relies on historical market data like prices and volumes to forecast stock price movements, while the latter relies on financial and macroeconomic factors.

Despite the belief in the unpredictability of stock prices due to the random walk and efficient market hypotheses, a considerable portion of the financial community maintains that stock prices exhibit some level of predictability, attributed to a tendency in investor behaviors and market patterns to repeat itself. The advent of machine learning technologies, coupled with the increase in computational resources, has resulted in novel methodologies in stock market forecasting based on this belief [2].

Another important factor when using stock market data to train a model is the fact the behavior of the stock market may change erratically throughout the historical data. Situations like financial crises are among scenarios that restructure the way that the stock market works. Chen and Yang have tried to simulate the stock market using four kinds of trading agents. Although they have successfully identified similarities in statistical features with those from the real market, these have changed drastically after crises [3]. Such studies highlight that training a model or AI agents that can predict market behavior can be challenging for some time spans but probably possible for the rest.

Diversity of features in stock market datasets necessitates effective feature selection techniques to enhance model performance. The inclusion of features like stock information, technical and economic indicators, and financial news has been explored in different studies, showing that feature selection can significantly improve prediction outcomes [2]. Htun et al. [2] have suggested use of three main types of structured inputs: basic features, technical indicators and economic indicators (fundamental indicators). Additionally, Baumohl has reviewed the most influential economic indicators of US stock market [4]. In another article market regime is predicted by first applying unsupervised machine learning to describe market regimes and then applying XGBoost and LSTM to predict the near future state of the market To do so, they have used four categories of technical indicators as features: Trend, Momentum, Volatility and Return [5]. Kofi Nti, Adekoya and Weyori have used a novel ensemble support vector machine to classify features and predict price of 10 days ahead [6].

Beside price data, economic indicators are traditionally used to predict stock markets but recently, a new type of

economic features called macroeconomic attention indices are introduced by Fisher et al. [7]. They have constructed these indices as a measure of attention to different macroeconomic risks reflected by news article counts. These Macroeconomic Attention Indices (MAI) are compared with the traditional economic indicators by Lu et al. [8] and are shown to be relatively more significant in predicting stock market returns than traditional ones.

The incorporation of news sentiments, or MAIs as mentioned earlier, into prediction has added new features for market prediction. Recent advances in natural language processing (NLP) has enabled automated processing of news to extract a sentiment regarding any specific ticker. Alpha Vantage is an example website that provides free API to access a variety of market data as well as news sentiments or MAIs [9]

As a result, in this study, in addition to using traditional economic indicators, MAIs, as defined in [7], are briefly compared to price data to explore any association with price data. To illustrate better how MAIs are obtained the table, shown in I, represents the search words Fisher et al. have used to evaluate MAI of each macroeconomic fundamentals. Using the search words in the second column, the MAI of the equivalent fundamental is evaluated.

TABLE I
MACROECONOMIC ATTENTION INDEXES AND FUNDAMENTALS [7]

Category	Newspapers search words	Fundamental
Credit rating	credit rating or bond rating	Corporate bond spread
GDP	(U.S. or United States) and (gross domestic product or gdp or gnp)	Quarter-to-quarter real GDP
Housing market	housing market or house sale or new home start or home construction or residential construction or housing sale or home price	National home price
Inflation	inflation or consumer price index or producer price index and (U.S. or United States)	Consumer Price Index
Monetary	(federal reserve or federal open market committee or fomc) and (interest rate or monetary or inflation or economy or economic or unemployment)	Federal fund rate and balance sheet
Oil	oil	Crude oil spot price
U.S. dollar	U.S. dollar or U.S. exchange rate or U.S. currency	Trade-weighted U.S. dollar index
Unemployment	(unemployment or jobless) and (U.S. or United States) and (economy)	Unemployment rate

With features being selected and pre-processed, they can be fed into a model training algorithm to create a model, representing a stock trading strategy. reinforcement learning as branch of machine learning is explained in detail in [10]. The book covers details of deep learning and Markov Decision Processes (MDPs) and presents practical examples using OpenAI Gym, a library with uniform API for reinforcement learning agents and environments. Additionally, in [11], an ensemble strategy that utilizes deep reinforcement learning is used to train a trading strategy model. They have integrated three

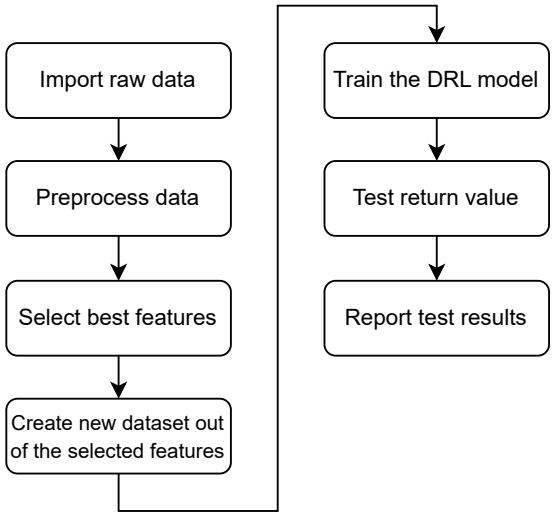

Fig. 1. Flowchart of the overall process

algorithms: Proximal Policy Optimization (PPO), Advanced Actor Critic (A2C), and Deep Deterministic Policy Gradient (DDPG). Their results show that the ensemble strategy outperforms the three individual algorithms and two baselines. In another study by Namyeong et al. RL is used for automated stock trading based on Transformer Actor-Critic with Regularization (TACR) to train a model [12]. The flowchart of the process in this study is presented in Figure 1. Han et al. have developed an intelligent stock trader and gym (ISTG) model based on DRL that utilizes historical price data, macroeconomic indicators and technical indicators to improve trading performance [13].

II. Data Collection

Fortunately, numerous online sources provide stock data, news, and economic factors required for the prediction of stock prices. A list of free stock and economic data resources is gathered and provided in the table below. Among these resources, Alpha Vantage provides free API to select and obtain all types of data including news indicators, economic indicators as well as historical stock price which is the reason it was selected for this study. In this study, News sentiments scores, macroeconomic indicators and the price data of S&P 500 ETF (SPY) are obtained from the Alpha Vantage website.

III. Feature Selection

The environment of RL specifies all of the random and decision variables that the agent is acting in. These features can be price features like open, close, maximum and minimum prices, trading volume, momentum indicators like RSI, economic indicators like unemployment rate, and Consumer Price Index, or sentiment scores obtained from news called MAIs as briefly explained before. However, using too many features for training the RL agent can be very computationally costly as many iterations are needed to train an RL agent particularly when a neural network is used to learn the Q values. Using a

TABLE II
OVERVIEW OF ECONOMIC AND STOCK DATA RESOURCES

Name	Features
Federal Reserve Economic Data	Offers a vast collection of economic data, including interest rates, inflation, employment numbers, and more.
World Bank Open Data	Provides access to global economic indicators, such as GDP, GNI, and development indicators across various countries.
Bureau of Economic Analysis	Offers detailed U.S. economic accounts data, including GDP, consumer spending, and corporate profits.
Bureau of Labor Statistics	Source for U.S. labor market information, including unemployment rates, job growth, and wage data.
Trading Economics	Provides economic indicators, exchange rates, stock market indexes, government bond yields, and commodity prices for over 200 countries.
Yahoo Finance	Offers historical stock prices, volumes, and financial summaries. Data can be downloaded in CSV format for individual stocks.
Google Finance	Provides real-time and historical stock prices, market summaries, and financial news. Useful for integrating directly with spreadsheets.
Alpha Vantage	Offers free APIs for historical stock data, including time series data, technical indicators, and sector performances.
Quandl	Provides financial, economic, and alternative datasets. Some data is free, but extensive datasets may require a subscription.
Investing.com	Provides historical data, news, and financial information on stocks, indexes, currencies, and commodities.

high capacity model without enough data can also lead to an overfitted model which will not perform well on test data or and agent that under-performs in real trading situation.

In this study, different types of features - macroeconomic indicators, news sentiment score and price technical indicators - are visually analyzed to explore opportunities for future price prediction. Finally, the selected features are introduced.

The relationship between the closing prices of the SPY ETF and the overall sentiment score from financial market news is explored, as depicted in Figure 2. No Association is discernible from the visual representation. This observation is corroborated by the correlation metrics outlined in Table III, which indicate a negligible correlation between the prices, and sentiment scores.

Further exploratory data analysis investigates the impact of macroeconomic factors, specifically the US unemployment rate and retail sales, on the SPY ETF. These factors are chosen based on their recognized influence on stock market dynamics as noted by Baumohl in "The Secrets of Economic Indicators" [4]. An analysis over a four-year span, as illustrated in Figure 2, reveals a relatively good correlation between these macroeconomic indicators and the SPY's performance with positive correlation with unemployment rate and negative correlation with retail sales as corroborated by Table III. However, the there is no daily data from these macroeconomic indicators. As a result, these indicators should be used for long-term investment instead of daily trading, which is the subject of this study.

So far, based on the data exploratory analysis, no noticeable relationship between measurable macroeconomic indicators

TABLE III
CORRELATION COEFFICIENTS

	Close	**Sentiment**	**Unemployment**	**Retail Sales**
Close	1.00	0.20	0.74	-0.67
Sentiment	0.20	1.00	0.18	-0.05
Unemployment	0.74	0.18	1.00	-0.66
Retail Sales	-0.67	-0.05	-0.66	1.00

or news sentiment with the price is observed. In fact these relationship are complicated and the effects of a wide range of factors need to be taken into account which needs further and more in depth study. However, a more traditional and simpler methodology to predict the price is using technical indicators.

A. Technical Indicators

The stock market's dynamics are influenced by traders, many of whom rely on technical analysis to interpret the stochastic price series. Technical indicators, classified into trend, momentum, and volatility categories help traders and investors understand market regimes. To avoid overfitting, a selection of indicators from each category is essential. Here are the chosen indicators for identifying market states. For each indicators specific thresholds are used to turn them into signals for the trading agent. The definition of each indicator is explained as follows [5], [14].

B. Trend Indicator

Commodity Channel Index (CCI) The CCI is calculated as:

$$CCI = \frac{TP - SMA(TP, N)}{0.015 \times MD}$$

Where:

- TP (Typical Price) = $\frac{High+Low+Close}{3}$
- SMA is the Simple Moving Average over N periods
- MD is the Mean Deviation

Signals:

- 1 when $CCI < -100$
- -1 when $CCI > 100$
- 0 otherwise

C. Momentum Indicator

Relative Strength Index (RSI) The RSI is defined as:

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

Where:

$$RS = \frac{\text{Average gain of up periods}}{\text{Average loss of down periods}}$$

Signals:

- 1 when $RSI < 30$
- -1 when $RSI > 70$
- 0 otherwise

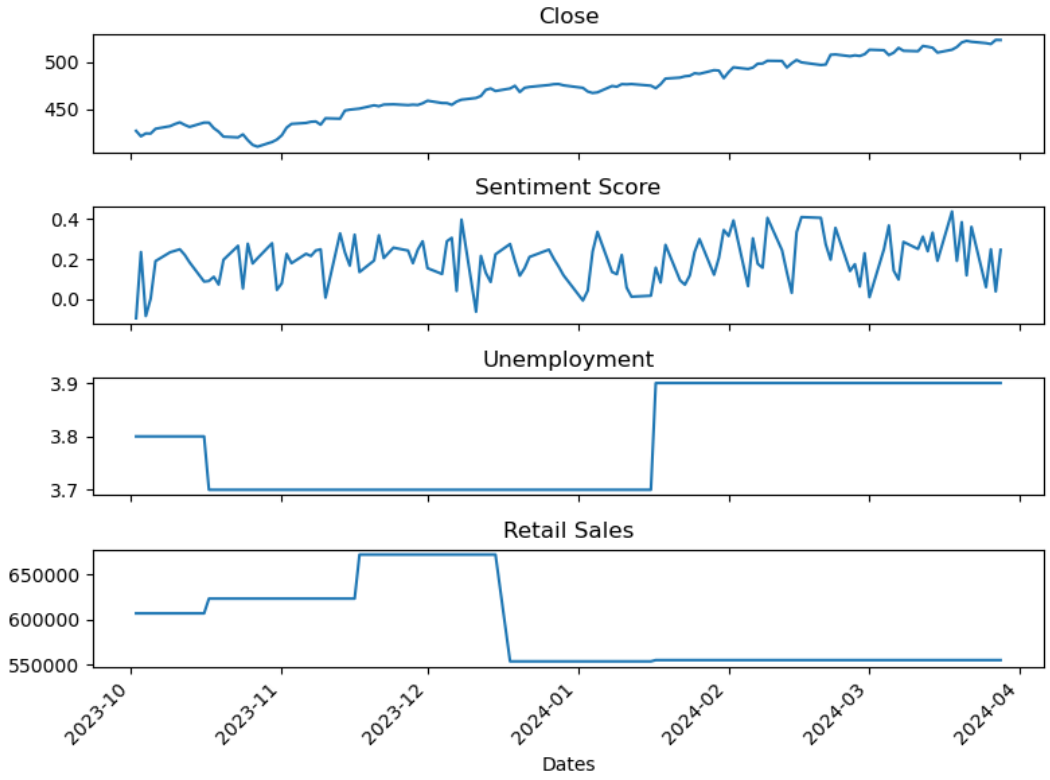

Fig. 2. Closing price of SPY symbol (S&P 500 ETF) together with overall sentiment score of the financial markets (in scale of -1 to 1), unemployment rate and retail sales in US in the time span between 2023-10 and 2024-04.

D. Volatility Indicator

Donchian Channel The Donchian Channel upper and lower bands are calculated as:

$$\text{Upper Band} = \max(High, N)$$

$$\text{Lower Band} = \min(Low, N)$$

Signals:

- 1 when $Close \geq$ Upper Band
- -1 when $Close \leq$ Lower Band
- 0 otherwise

E. Return

One-Month/Three-Month Returns One-Month and Three-Month Returns are calculated as the percentage change in the closing price over 21 and 63 days, respectively.

$Return = \frac{One-MonthReturn}{Three-MonthReturn}$

Signals:

- 1 when $Return > 0$
- -1 when $Return < 0$
- 0 otherwise

The aforementioned indicators signals and their relationship with the closing price, for a dataset of SPY symbol are applied and the signals are visualized in 3. As shown, the Donchian signal provides little information to the trading agent for the illustrated time span, but it is still used for training the DRL model in this study. The correlation study between signals and the price is not investigated as signals are not expected to reflect the level of price. For example, a signal might switch at the same level of price, when there is a change in trend.

IV. Deep Reinforcement Learning (DRL) Model

Deep Q-learning is an advanced reinforcement learning algorithm that addresses the limitations of traditional Q-learning when dealing with environments characterized by large or continuous state spaces. One classic example of such an environment is Atari games, where the variety of possible screen images (states) is vast, especially if one considers raw pixels as individual states. This complexity makes it infeasible to track and approximate Q-values for every possible state-action pair, as traditional Q-learning attempts to do [10]. The reason for using such a method to train the trader agent is due to the presence of continuous or unbounded features among the state variables: stock price, current balance, and number of shares held.

The core of the Q-learning algorithm is to adjust our network to estimate the Q-function accurately. We define the Q-function estimate as dependent on the parameters of our model, presented in $\hat{Q}_\theta(s,a) \approx Q^*(s,a)$ where s and a represent state and action correspondingly. It is important to note that Q-learning is focused on value estimation rather than policy learning. The Q-function approximation, represented

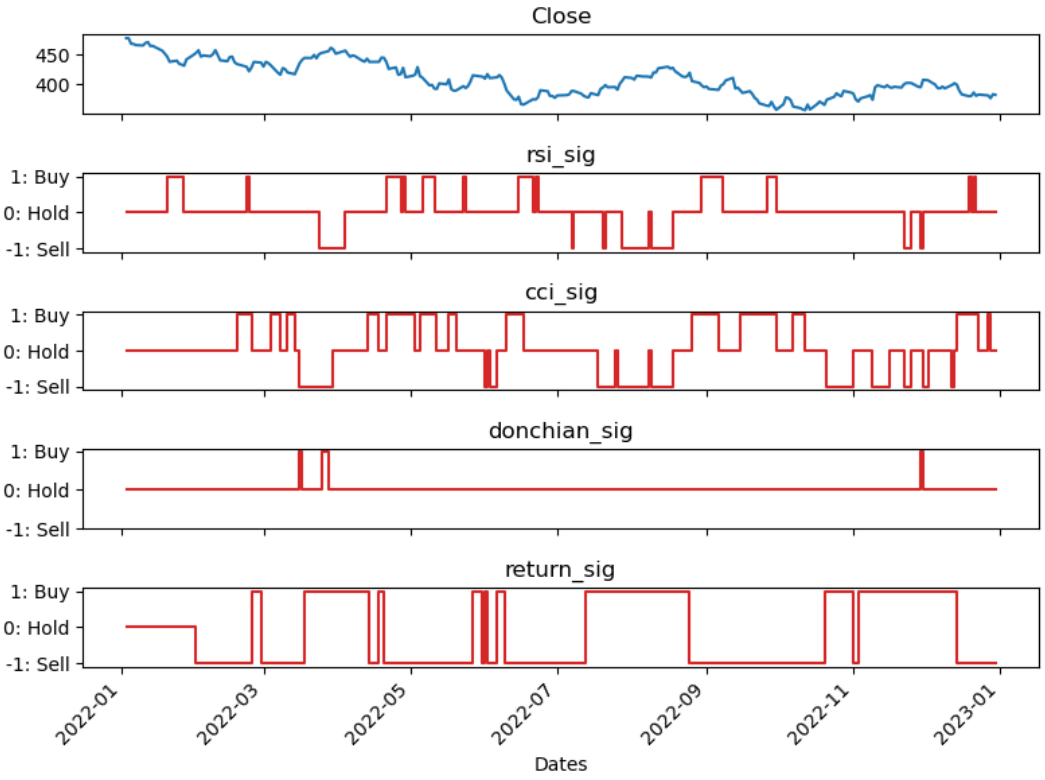

Fig. 3. Closing price, RSI, CCI, Donchian and Return signals of SPY ETF in year 2022

as Qtheta, has been formulated, and the intent is for it to approximate the expected future rewards. The Bellman Equation allows formulating this anticipated reward as in 1 [15].

$$R_t^* = r_t + \gamma \max_{a'} Q(\hat{s}_{t+1}, a'|\theta') \tag{1}$$

Where:

- R_t: Target Q-value for the current state-action pair.
- r_t: Reward received after taking action a in state s.
- γ: Discount factor, which determines the importance of future rewards.
- $Q(s_{t+1}, a'; \theta')$: Approximated Q-value for the next state s_{t+1} and all possible actions a', as predicted by the target network with parameters θ'.
- s_{t+1}: Next state after taking action a in state s.
- a': Any possible action in the next state s_{t+1}.
- θ': Parameters (weights) of the target neural network, which are periodically updated to match θ and are kept constant between updates.

$$\min_{\theta} \sum_{e \in E} \sum_{t=0}^{T} \left(\hat{Q}_\theta(s_t, a_t|\theta) - R_t^* \right)^2 \tag{2}$$

By substituting R_t^* from 1 in 2, the objective function (loss function) to be minimized is obtained:

$$\min_{\theta} \sum_{e \in E} \sum_{t=0}^{T} \left(\hat{Q}_\theta(s_t, a_t|\theta) - r_t + \gamma \max_{a'} \hat{Q}(\hat{s}_{t+1}, a'|\theta') \right)^2 \tag{3}$$

where:

- E: Set of experiences (state, action, reward, next state) tuples, also known as the replay buffer.
- T: Time horizon or the last time step of each episode.
- s_t: Current state at time step t.
- a_t: Action taken at time step t.
- θ: Parameters (weights) of the current neural network used to approximate the Q-function.

The function inside the Σ in equation 3 is entirely differentiable with respect to model parameters, allowing the utilization of gradient-based optimization methods to minimize the loss function in 3 [15].

The stock environment class is designed to simulate a simplified stock trading environment. It is based on the OpenAI Gym framework, which provides a standard API for developing and comparing reinforcement learning algorithms [16]. However, in this study most of the methods are customized. The class simulates the experience of trading stocks, allowing an agent (such as a reinforcement learning algorithm) to interact with the market and make decisions based on observed data. The environment encapsulates the complexities of stock trading into a simplified model:

1) Action Space: The environment defines a discrete action space with three possible actions: selling a stock, holding (or doing nothing), and buying a stock. This action space is a simplified representation of the decisions a trader can make at any given time. Also the number of shares in each transaction is limited to one share.

2) Observation Space: The observation space is constructed from both market data and the agent's financial state. This includes the external signal data, the agent's current balance, the number of shares currently held, and the most recent stock price. The agent receives a normalized state representation, combining the external data with the ratio of the current balance to the current stock price and the number of shares held. This comprehensive observation space equips the agent with the necessary information to make informed decisions.

3) Steps and Dynamics: The core interaction with the environment happens through the step method which acts as a transition function and simulates the passing of one time step in the trading world. Given an action from the agent (buy, sell, or hold), the environment updates the agent's portfolio, including the number of shares held and the current balance. It also advances to the next time step, updating the stock price based on the provided data.

The result of an action is a combination of the new observation (reflecting the updated state of the market and the agent's portfolio), a reward (or penalty) calculated based on the change in the portfolio's value, a flag indicating whether the simulation has ended (for example, when there are no more data points to process), and an optional info dictionary that can provide additional debug information.

4) Rewards: The reward structure aims to motivate the agent to maximize its portfolio value. The reward at each step is determined by the potential profit or loss, measured by the difference in the value of the shares held, scaled by the current stock price. The proposed reward formula aims to incentivize a relatively long-term profit from the shares held by the agent.

$$R = \frac{SH \times (MFP - CP)}{CP}$$

Where:

- R is the reward at the current time step
- SH is the shares held by the trader at the current time step
- MFP is the average price over a set number of data points after the current time step.
- CP is the price at the current time step

5) Deep Q-Learning Implementation: The DQN agent learns to make decisions by predicting the future rewards of its actions in a given state, using a neural network to approximate the Q-function. The neural network employed in this study is a fully connected feedforward network, designed to approximate the Q-value function within a reinforcement learning framework. The architecture of the network is composed of three layers, interconnected by rectified linear unit (ReLU) activation functions, which introduce non-linearity and enable the network to learn complex decision boundaries.

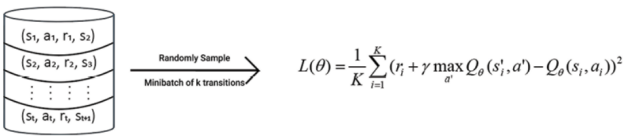

Fig. 4. Minibatch in DQN for updating Q by minimizing the loss function $L(\theta)$ [17]

V. Training Process

The training process involves multiple episodes, where each episode encapsulates a complete sequence of trading actions, allowing the agent to experience a full cycle of decision-making within the stock market environment. An episode spans from the start to the end of a pre-defined training period, which, according to the code, is set between January 2020 and January 2023.

At the beginning of each episode, the environment initializes to a state reflective of starting with an initial balance and a predetermined number of shares.

To prevent overfitting, two forms of stochasticity are incorporated into the training process, enhancing the robustness of the agent's learning. The training loop incorporates a mechanism for random resets with a specific probability at each time step. The other form of stochasticity is through resetting the agent's state from a random time step in the given training data. This reset is performed at the end of each training episode.

During the training loop, the agent performs experience replay whenever its memory accumulates enough data. In experience replay, a random minibatch of transitions is created from the agent's memory to update the neural network and prevent instability in learning the Q function as depicted in Figure 4.

The exploration rate (epsilon) is adjusted at the end of each episode to balance exploration and exploitation. As training progresses, the agent shifts from exploring the environment randomly to exploiting its learned knowledge by decaying the ϵ to near zero. The decay factor is calculated based on the number of episodes.

After each episode, various metrics, including the final portfolio value, are logged to monitor the agent's performance. The training loop concludes with saving the agent's state and plotting the change in portfolio value over episodes, providing a visual measure of the agent's learning progress.

VI. Testing Process

The testing phase evaluates the trained agent's performance on unseen data. The environment is reinitialized with test data, and the agent's exploration rate is set to 0 to disable random actions, focusing solely on the exploitation of the learned policy.

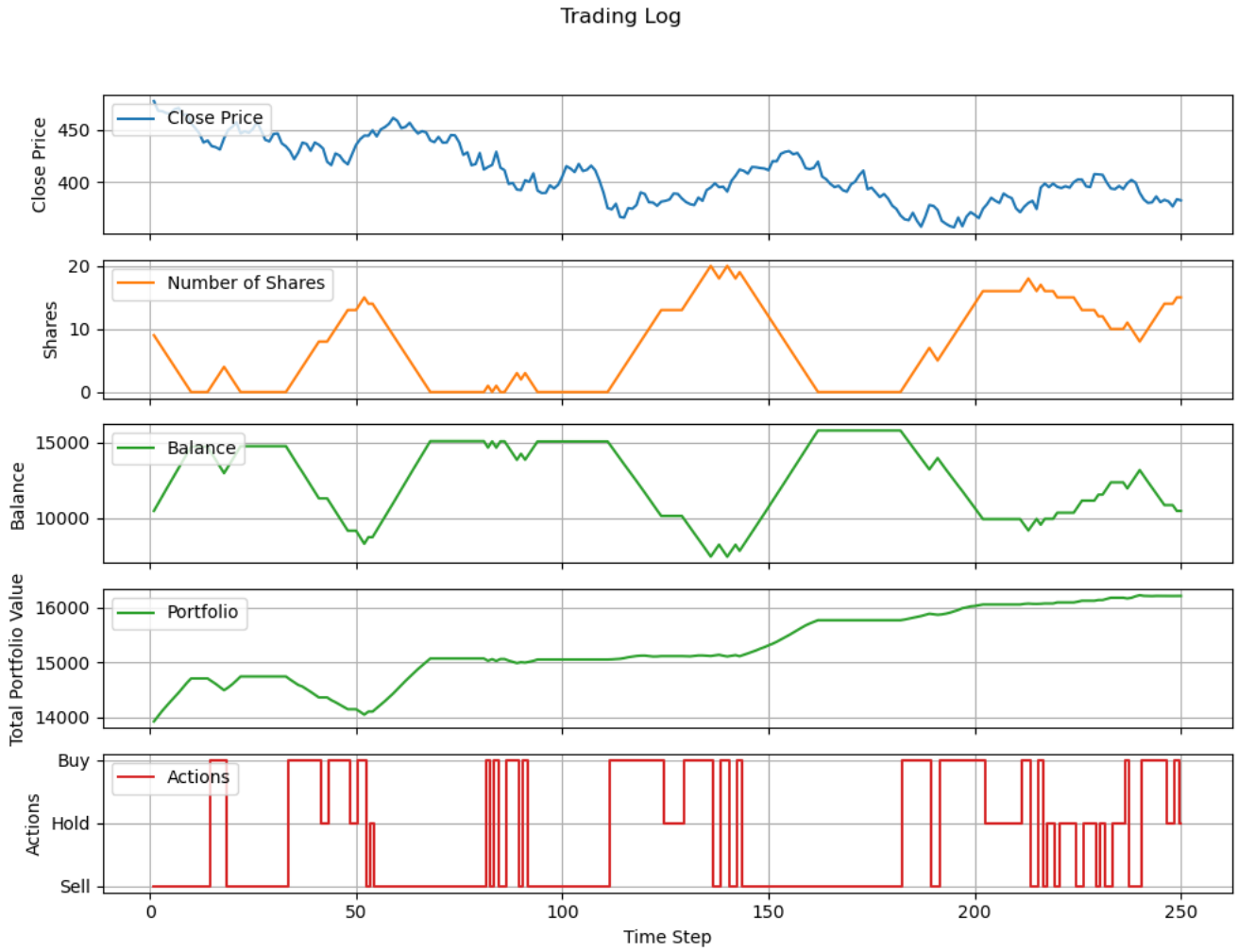

Fig. 5. Trained agent trading log on SPY price for the time period between 2022-01 to 2023-01

The testing loop is similar to the training loop but typically does not include the replay step since the goal is to evaluate the learned policy rather than further train the agent. The agent takes actions based on the current state, processes the results, and moves to the next state until the end of the testing period or an episode concludes.

VII. Results and Discussion

The trading strategy of the trained DQN RL agent is presented in figure 5. The agent operates within this dynamic environment, making buy or sell decisions to maximize its rewards. Several tests for different periods are conducted, which show satisfactory results for the test periods. The figure shows the Close price, number of shares held by the trader, the available balance by the trader, the total portfolio value and the trading action the agent has taken in each time step. The total portfolio value is calculated as:

$$PV = CB + SH \times CP$$

- PV is the portfolio value at the current time step
- CB is the balance the current time step

The evaluation of the trained agent implemented using Deep Q-Networks (DQN) Reinforcement Learning (RL) is performed by comparing the portfolio value with those from individual signals. For this purpose, for each trading signal (RSI, CCI, Donchian, and Return) a trading agent is implemented that acts only based on the corresponding signal. The comparison of the performance on the SPDR S&P 500 ETF Trust (SPY) stock shows better performance of the trained DQN RL agent for the period shown in figure 6.

The Number of Shares panel indicates the quantity of SPY shares the agent holds at any given time. Notably, the agent's position changes infrequently, signifying a strategy that does not rely on high-frequency trading. This is in fact due to using Mean Future Price, as described before, to calculate the reward at each time step.

The Balance panel reflects the agent's balance (cash reserves) apart from its investment in SPY shares. The balance shows sharp drops followed by periods of relative stability, implying that the agent makes lump-sum investments rather

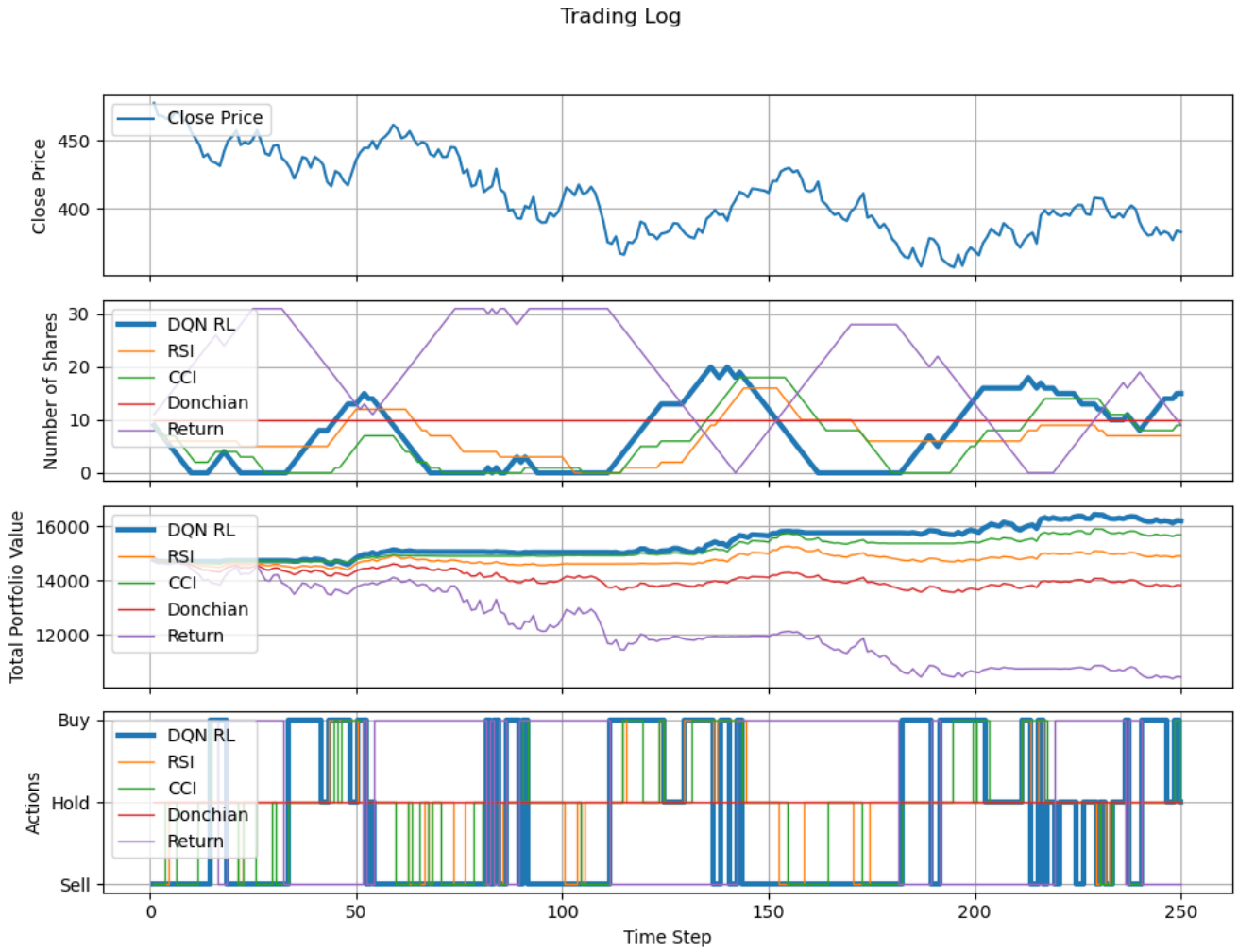

Fig. 6. Comparison of trading agents performance for SPY symbol for the period between 2022-01 to 2023-01

than gradual or continuous purchases. However such sharp drops or rises are moderated by the limit of one share per transaction.

The Portfolio panel tracks the total value of the agent's portfolio. The overall trend is upward but with fluctuations, indicating that the agent's policy is successfully leading to an increase in total assets over time. This upward trajectory is a positive sign of the agent's performance, especially if it outpaces the market's performance.

The final panel, Actions, captures the decisions made by the agent—whether to buy, sell, or hold the SPY stock. The sell actions appear to be well-timed, often occurring when the stock's price is higher, and buy actions occur when the stock's price shows a dip, demonstrating the agent's capacity to exploit opportunities to buy low and sell high.

The visualization of the agent's actions in the context of stock price and portfolio value offers a clear indication that the DQN RL agent is learning a strategy aimed at capitalizing on market movements to grow its portfolio value. The periodic nature of the transactions suggests that the agent is identifying patterns or signals within the market data to inform its decisions rather than reacting to single-point fluctuations.

VIII. Conclusion

This study successfully demonstrates the potential of Deep Reinforcement Learning (DRL), specifically through the use of a Deep Q-Network (DQN), in navigating the complexities of the stock market with a focus on the SPDR S&P 500 ETF Trust (SPY). The implemented DRL model was able to devise trading strategies that capitalized on patterns in price movements, technical indicators, and macroeconomic signals to optimize portfolio returns under simulated market conditions.

There are areas of improvement, however, that a future study and pursue, as summarized below:

- The features used to represent the price regime are limited to three price signals. As explained in the introduction macroeconomic and news sentiments can also be included as well. However, based on the preliminary data analysis, no specific association between the price and these

features was observed. A more sophisticated study might be required to study their impact on the price and see whether they provide any leading signal for a transaction.
- The importance of the signals on prediction is not studied in this article, but it is suggested to investigate this in a future study.
- Number of shares sold or bought in every transaction is limited to 1. DQN allows the removal of this limit, however, it will require more learning episodes
- Increasing the number of episodes has shown significant improvement in the behavior of the trading agent, however, due to limited computational resources further increase in the number of episodes was not easily possible. Such a study is required because the increase in portfolio value did not slow down. As a result, more training episodes might be required to reach a convergence of the final portfolio value.
- The range of stock data used for training agents is limited to less than a year. This has been demonstrated to adversely affect the performance of the trading agent in other time periods. So it is recommended to consider using other time periods from earlier years when different market regimes prevailed.
- The trained agent's performance should be compared with the trading performance of RL agents that utilize other learning algorithms as well.

REFERENCES

[1] F. Cecconi, *AI in the Financial Markets : New Algorithms and Solutions*, first edition. ed., ser. Computational Social Sciences Series. Cham, Switzerland: Springer, 2023.

[2] H. H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," *Financial innovation (Heidelberg)*, vol. 9, no. 1, pp. 26–26, 2023.

[3] H. Yang and S. Chen, "A heterogeneous artificial stock market model can benefit people against another financial crisis," *PloS one*, vol. 13, no. 6, pp. e0 197 935–e0 197 935, 2018.

[4] B. Baumohl, *The secrets of economic indicators : hidden clues to future economic trends and investment opportunities*. Upper Saddle River N.J: Wharton School Publishing, 2005.

[5] "Market regime determination and prediction," Unpublished Manuscript, No Date, available from NYU Tandon School of Engineering.

[6] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "Efficient stock-market prediction using ensemble support vector machine," *Open Computer Science*, vol. 10, no. 1, pp. 153–163, 2020. [Online]. Available: https://doi.org/10.1515/comp-2020-0199

[7] A. Fisher, C. Martineau, and J. Sheng, "Macroeconomic Attention and Announcement Risk Premia," *The Review of Financial Studies*, vol. 35, no. 11, pp. 5057–5093, 02 2022. [Online]. Available: https://doi.org/10.1093/rfs/hhac011

[8] F. Ma, X. Lu, J. Liu, and D. Huang, "Macroeconomic attention and stock market return predictability," *Journal of International Financial Markets, Institutions and Money*, vol. 79, p. 101603, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S104244312200083X

[9] "Alpha vantage: Free apis for realtime and historical stock data, forex, and cryptocurrency," https://www.alphavantage.co/, accessed: [March 18th 2024].

[10] M. Lapan, *Deep reinforcement learning hands-on : apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more*, second edition. ed. Birmingham ;: Packt Publishing, 2020.

[11] H. Yang, X.-Y. Liu, S. Zhong, and A. Walid, "Deep reinforcement learning for automated stock trading: An ensemble strategy," in *ICAIF 2020 - 1st ACM International Conference on AI in Finance*, 2020.

[12] N. Lee and J. Moon, "Offline reinforcement learning for automated stock trading," *IEEE Access*, vol. 11, pp. 112 577–112 589, 2023.

[13] D. Han, J. Zhang, Y. Zhou, Q. Liu, and N. Yang, "Intelligent trader model based on deep reinforcement learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ser. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, vol. 11817, pp. 15–21.

[14] M. J. Pring, *Technical analysis explained : the successful investor's guide to spotting investment trends and turning points*, fifth edition. ed. New York: McGraw-Hill Education, 2014 - 2014.

[15] N. Buduma and N. Locascio, *Fundamentals of deep learning : designing next-generation machine intelligence algorithms*, first edition. ed. Sebastopol, CA: O'Reilly Media, 2017.

[16] Farama Foundation, "Gymnasium by farama foundation," 2024, accessed: March 22nd, 2024. [Online]. Available: https://gymnasium.farama.org/

[17] S. Ravichandiran, *Deep reinforcement learning with Python : master classic RL, deep RL, distributional RL, inverse RL, and more with OpenAI Gym and TensorFlow*, second edition. ed., ser. Expert insight. Birmingham, England ;: Packt Publishing, 2020.