IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

REINFORCEMENT LEARNING FOR STOCK PORTFOLIO MANAGEMENT

Bhooshi Uhitha Rheya

Department of Computer Science and Systems Engineering,

Andhra University College of Engineering, Visakhapatnam, India.

Abstract: Stock trading techniques play the essential aspect of functioning as a key factor when it comes to investment. In this case Paper, It proposes a way of learning a stock trading strategy that maximizes return on investment using Deep Reintorcement.Researchers developed ways to buy and sell using deep reinforcement learning agents using ensemble trading strategies for actor-critical-based models on these 5 Algorithms: Proximal Policy Optimization (PPO), Actor-Critic Gain (A2C), Deep Deterministic Policy Gradient (DDPG), Twin-Delayed Deep Deterministic Policy Gradient (TD3), Soft Actor-Critic(SAC). Ensemble strategy inherits and integrates the excellent features of these five Algorithms, thereby robustly adapting to the exceptional market situation.To avoid immense memory usage in training networks with an area of continuous movement,For processing massive amounts of data, researchers employ the load-on-demand approach, 30 Dow Jones Stocks with enough liquidity are used to test the algorithms.The overall performance of the trading agency was evaluated with significant enhancement of the Reinforcement algorithm and compared to both the Dow Jones trading stock index and the traditional minimum deviation portfolio allocation procedure. The proposed deep ensemble strategy outperforms baselines and five algorithms in trade-weighted returns by the Sharpe ratio.

Index Terms - Deep reinforcement learning,Markov Decision Process,Automated stock trading, ensemble strategy,actor-critic framework, Proximal Policy Optimization (PPO), Actor-Critic Gain (A2C),Deep Deterministic Policy Gradient (DDPG), Twin-Delayed Deep Deterministic Policy Gradient (TD3),Soft Actor-Critic(SAC).

I.INTRODUCTION

Stock Trading refers to buying and selling of shares and plays a critical role in the investment. A profitable automated stock trading strategy is vital to investment companies and is applied to maximize investment performance, such as expected return. Automated stock trading is also termed "algorithmic trading." Stock trading can be automated using machine learning (ML) techniques such as reinforcement learning (RL) and deep reinforcement learning (DRL) [2]-[3]. However, identifying patterns in a complex and dynamic stock market is quite challenging. Requires a solution that is adaptable to the needs of a dynamic stock market environment. Effective RL agents in the process of portfolio management.Dealing with a limited number of financial tools, primarily equities. Agents will learn how to optimally allocate funds to the aforementioned assets starting with a finite budget and trying to maximize their overall wealth.Agent are trained and tested on true demand data. In general, manual stock trading is a continuously evolving process involving feedback from the financial markets and implementing new ideas to optimize trading strategies.Multiple stock trading is different from single stock trading because as the number of stocks increases, the dimension of the data will increase, and the state and action space in reinforcement learning will grow exponentially. Therefore, accuracy and repeatability are fundamental in this situation. Due to developments in deep reinforcement learning methodologies and advancements in artificial intelligence generally, researchers have recently begun to apply reinforcement learning techniques to the financial market. However, the general public rarely has access to the new algorithm or trading bots developed by financial specialists outside of the academic and professional domains. The possibility of deep reinforcement learning algorithms with the use of automatic stock market trading to support and help regular individual traders in their trading processes and maybe enhance or maximize their profits.

The motivational study question is, "How do we employ modern data technologies like reinforcement learning, deep learning, and machine learning in the field of quantitative finance to enable ordinary traders?" To answer this exploration design, this work develops an ensemble approach predicated on an earlier study (1).

Solution: Ensemble Deep Reinforcement Learning Trading Strategy. This strategy includes five actor-critic-based algorithms: Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), Advantage Actor Critic (A2C), Soft Actor-Critic (SAC), and Twin-Delayed Deep Deterministic Policy Gradient (TD3). It combines the best features of the Five algorithms, thereby robustly adjusting to different market conditions.

II. Literature Review

Recent deep reinforcement learning applications in the financial sector take into account discrete or continuous state and action domains and employ one of the following learning philosophies: critic-only approach, actor-only approach, or actor-critic approach [4]. Learning models with continuous action houses, as opposed to discrete auction houses, provide better management capabilities. A growing subject within the information processing analysis subfield that's gaining popularity among students from many backgrounds is quantitative finance. With the introduction of varied ways to forecast the trend/position of the securities market to enhance the return/profits, a major amount of research and study has been published. The most effective banking and investment in institutions have made some processes and algorithms private. The foremost prestigious financial and investing institutions revealed several techniques and algorithms that had been kept a secret. The greatest influence on this study comes from the preceding analysis. It was written in November 2020. According to the authors' justifications, this study is part of a stock commerce strategy that focuses on maximizing investment return. It uses an ensemble method that combines three actor-critic-based reinforcement learning algorithms with Deep Deterministic Policy Gradient (DSPG), Proximal Policy Optimisation (PPO), and Advantage Actor-Critic (A2C) (DDPG). In terms of the Sharpe quantitative relationship, which measures risk-adjusted returns, the ensemble methodology outperformed three complete reinforcement learning methods: two baseline methods, the min-variance portfolio allocation strategy, and the Dow Jones Industrial Average. It reaches a median annual comeback of thirteen percent and a cumulative comeback on k 70.4 % between January 2016 and May 2020. The authors believe that developing an automated trading system is critical for investing in corporations and that current strategies are inadequate; thus, their team wishes to create a new automated trading system. This is a significant paper in the discipline because it provides a thorough understanding of the performance of reinforcement learning agents in a stochastic, random probability environment. It does provide future research with a general idea of how reinforcement learning performs in a setting of random probability. Because previous research has focused solely on selecting high-performing stocks rather than allocating trades positions or shares among the selected stocks, the authors wish to develop a model that can model positions.

III. Description of the problem

This section describes how a Markov Decision Process is used to solve the problem of deciding when to BUY/HOLD/SELL a stock portfolio (MDP). The method for solving the MDP by discovering an ideal policy is further explained in this section and characterize our trading goal as maximization of expected return [5]. It should be noted that because the state space with real-valued stock prices is so large, the state space for stock prices has been formulated and solved.

A. MDP Stock Trading Model

To replicate the stochastic nature of the dynamic stock market, researchers employed a Markov Decision Process (MDP) as follows:

- *State s = [p, h, b]* is a vector of stock prices *($p \in R^D_+$)*, share prices *($h \in Z^D_+$)*, and net proceeds *($b \in R+$)*, where D stands for the number of stocks and Z_+ stands for non-negative integers.
- *Action a*, is a series of actions spread across D stocks. The five permitted actions on each stock are selling, purchasing, and holding, which, respectively, cause the number of shares of the stock to decrease, increase, or remain unchanged.
- *Reward (s, a, s0)*: The immediate benefit of acting in state *s* to state s_0.
- Policy(s): The trading strategy at state *s*, represented as the probability distribution of the state *s'* activities-*value Q(s, a)*: the anticipated benefit of acting per policy at state *s*.

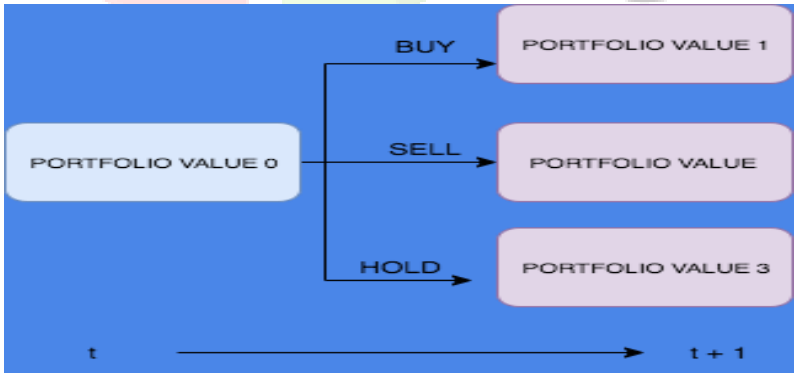


Figure 1. Shows how a starting portfolio value combined with three actions results in three possible portfolios. It should be noted that "hold" may result in different portfolio values due to changing stock prices.

Figure 1 represents the state change of the stock trading process. At each stage, one of three possible actions is applied to the portfolio stock *d (d = 1,..., D).*

- Selling $k[d] \in [1, h[d]]$ shares leads in $h_{t+1}[d] = h_t[d] - k[d]$, where $k[d] \in Z_+$ and d = 1,..., D.
- Holding constant, $h_t[d] = h_{t+1}[d]$.
- Purchasing $k[d]$ shares leads in $h_{t+1}[d] = h_t[d] + k[d]$.

As shown in Figure 1, when an action is made at time *t* and the stock prices are updated at time *t+1*, the portfolio values may change from “portfolio value 0” to “portfolio value 1,” “portfolio value 2,” or “portfolio value 3” as appropriate. As seen, the portfolio value is

$p^T h + b.$

B. INCORPORATION STOCK CONSTRAINTS

The following assumptions and constraints reflect practical concerns:

1. Stock Market Liquidity Calculation
    - Multidimensional market liquidity measurement using measures such as depth, breadth, immediacy, and transaction cost
    - comparison between dimensional liquidity measures
    - New Liquidity Measures Proposal
    - Identifying patterns and relationships between liquidity dimensional measures
2. Stock Market Liquidity Influencing Factors
    - Liquidity and regulatory policy announcements
    - Corporate Announcements, disclosures, and liquidity
    - Practices of corporate governance and liquidity
    - Stock exchange mergers and trading system developments Company-specific company-specific liquidity
3. Stock Market Liquidity Calculation
    - Market liquidity is measured in multiple dimensions, including depth, breadth, immediacy, and transaction cost.
    - Comparison of liquidity dimensional measures
    - New Liquidity Proposal Measures Identifying Patterns and Relationships Between Dimensional Liquidity Measures
4. Risk of Stock Market Liquidity and Expected Returns
    - Market Liquidity Risk on Expected Returns in Different Market Conditions
    - The Impact of Stock Market Liquidity Risk on Expected Returns in Different Stock Market Sectors
    - Factors influencing the significance of stock market liquidity risk on expected returns

C. Return maximization as a trading objective:

The reward function is defined as the change in portfolio value when action *a* is executed at state s and results in a new state *s0*.

The purpose is to devise a trading strategy that maximizes the portfolio's value change:

$$r(s_t, a_t, s_{t+1}) = (b_{t+1} + p^T_{t+1} h_{t+1}) - (b_t + p^T_t h_t) - c_t \quad (1)$$

where the initial & secondary terms represent the portfolio value at time *t+1* and time To dissect the return further, the transition of the shares that is defined as h_{t+1} equals h_t.

$$h_{t+1} = h_t - k^S_t + k^B_t \quad (2)$$

and balance, but the transition is specified in (1). (4) can thus be rewritten as

$$r(s_t, a_t, s_{t+1}) = r_H - r_S + r_B - c_t, \quad (3)$$

where

$$r_H = (p^H_{t+1} - p^H_t)^T h^H_t, \quad (4)$$

$$r_S = (p^S_{t+1} - p^S_t)^T h^S_t, \quad (5)$$

$$r_B = (p^B_{t+1} - p^B_t)^T h^B_t, \quad (6)$$

where r_H, r_S, and r_B signify the change in portfolio value caused by holding, selling, and buying shares from time to time t + 1. According to Equation (9), By purchasing and holding companies whose prices will climb in the upcoming time step, investors can increase the positive change in portfolio value and reduce the negative change by selling equities whose prices will decline. Index of Turbulence *turbulence* is combined with the reward function to address the fear of a market crash. When the index rises above a certain level, Equation (5) changes to resale = $(p_{t+1}, p_t)^T k_t$, suggesting that the goal is to sell all holding stocks to minimize the negative change in portfolio value because all stock values will decrease. The setup of the model is as follows. p_0 is the stock price at time 0, while b_0 is the beginning fund amount. h and $Q\pi(s, a)$ are both zero, and (s) are distributed evenly over all actions for each state. Then, by engaging with the stock market environment, $Q(s_t, a_t)$ is updated. The Bellman Equation reveals the best course of action, with the expected benefit of acting the state s_t equaling the total of the immediate reward ($r(s_t, a_t, s_{t+1})$) and the future reward in state s_{t+1}. To achieve convergence, designers need to discount future gains by a factor of 0 to 1. If we continue in this direction, designers will eventually profit.

$$Q_\pi(s_t, a_t) = E_{st+1}[r(s_t, a_t, s_{t+1}) + \gamma E_{at+1} \sim \pi (s_{t+1})[Q_\pi(s_{t+1}, a_{t+1})]]. \quad (7)$$

The aim is to create a trading strategy that maximizes the portfolio's positive cumulative change. In the dynamic environment, In order the problem of ($r(s_t, a_t, s_{t+1})$), researchers adopt the deep reinforcement learning method.

IV. ENVIRONMENT OF THE STOCK MARKET

Before training a deep reinforcement trading agent, researchers carefully designed the environment to mimic real-world trading. This allows the agent to interact with and absorb information from its surroundings. Various information, such as historical stock prices, current holding shares, technical indicators, and so on, must be considered in real trading. The environment must provide this information to its trading agent, who must then carry out the tasks listed in the previous section. Let's build an environment and use the OpenAI gym to train the agents [6], [7], and [8].

A. Multiple Stocks Environment

To model the trading of several equities, researchers use an ongoing used activity space. The portfolio contains, according to analysts' estimates, 30 stocks in total.

1) State Space: To describe the state space of numerous stock trading environments, Make use of a 181-dimensional vector made up of seven information components: [b_t, p_t, h_t,M_t, R_t, C_t, X_t]. Each component is described below:

- $b_t \in R_+$: the available balance at time step *t.*
- $p_t \in R^{30}{}_+$: each stock's adjusted closing price.
- $h_t \in Z^{30}{}_+$: the number of shares owned in each stock.
- $M_t \in R^{30}$: MACD (Moving Average Convergence Divergence) is calculated using the close price. MACD is a momentum indicator that identifies moving averages that are widely utilized [35].
- $R_t \in R^{30}$ +: The Relative Strength Index (RSI) is computed using the closing price. The size of the recent price changes has been evaluated by RSI. If the price oscillates around the support line, the stock is oversold and investors should consider buying. If the price oscillates around the resistance, the stock is overbought and should be sold.
- $C_t \in R^{30}{}_+$: The Commodity Channel Index (CCI) is calculated using the open, high, and low prices. CCI to compare the current price to the average price over a while to determine whether to buy or sell.
- $X_t \in R^{30}$: The Average Directional Index (ADX) is generated using the open, high, and low prices. The ADX indicator detects trend strength by quantifying price movement.

2) Action Space: The action space is defined for a single stock as "-k,⋯, 1, 0, 1,⋯, k," where " k" and "-k" is the number of shares individuals can purchase and sell, respectively and k ≤ h max, where the maximum number of shares for each decision made is specified by the predetermined h max parameter. As a result, the total action space is $(2k + 1)^{30}$. The action space is then normalized to [1] because the RL algorithms A2C and PPO establish the policy directly on a Gaussian distribution, which must be normalized and symmetric [8].

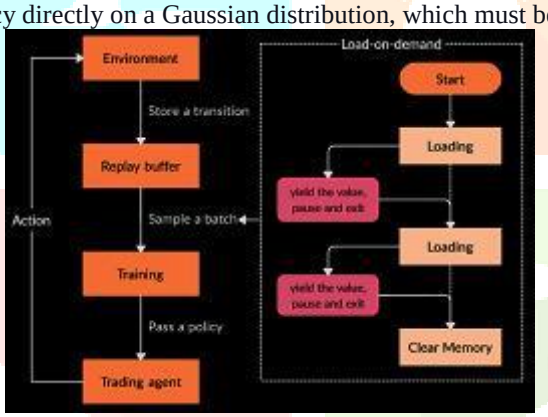


Figure 2. An illustration of the load-on-demand approach.

B. Management of Memory

Memory usage for training may increase exponentially as the number of stocks, data kinds, state space variables, number of layers and neurons in neural networks, and batch size increase. To address the issue of memory requirements, for memory effectiveness, researchers used a load-on-demand approach. The load-on-demand approach, as displayed in Figure 2, does not keep all results in memory; rather, it creates them on demand. Because the memory is only utilized when a result is sought, memory utilization is decreased.

V. TRADING SYSTEM BASED ON DEEP REINFORCEMENT LEARNING

In this trading agent's performance, five actor-critic algorithms are used. DDPG, PPO, A2C, TD3, and SAC are the five algorithms, respectively. An ensemble method is presented for combining the five agents to create a strong trading strategy.

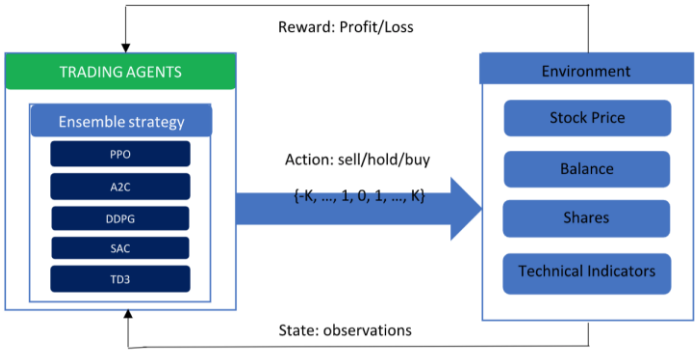


Figure 3. An Outline of a Stock trading strategy based on Reinforcement learning

1. *Deep Deterministic Policy Gradient (DDPG)*

DDPG [18] is used to induce the highest possible investment return. DDPG integrates the frameworks of Q-learning [9,10], as well as policy gradient [9,10], and employs neural networks as function approximators. In contrast to DQN, which learns indirectly through Q-value tables and is plagued by the curse of dimensionality [40], DDPG learns directly from observations via policy gradient. To better match the continuous action space environment, it is proposed to deterministically map states to actions.

The DDPG agent acts as each time step, receives a reward, and then appears at time $s_{t+1.}$ Replay buffering R is used to store changes (s_t, a_{t+1}, r_t). A cluster of N advances is taken from R, and the Q-value y_i is updated as follows:

$$y_i = r_i + \gamma Q^0(s_{i+1}, \mu^0(s_{i+1}|\theta^{\mu 0}, \theta^{Q0})), i = 1, \cdots, N. \quad (8)$$

After that, the critical network is updated by minimizing the loss function, which is the difference between the outputs of the target critical network Q′ and the earlier critical network Q. DDPG is helpful for stock trading since it can handle continuous action space.

$$L(\theta^Q) = E_{st,a_t,r_t,s_{t+1}\sim \text{buffer}}[(y_i - \mathcal{Q}(s_t, a_t|\theta^Q))^2]. \quad (9)$$

Because DDPG is good at dealing with continuous action space, it is suitable for stock trading.

2. *Proximal Policy Optimization (PPO)*

PPO is studied and utilized as a segment in ensemble techniques. PPO [11] is used to regulate arrangement for gradient updates and ensure that the new approach doesn't stray too far from the old one. PPO seeks to further the goal of Trust Region Policy Optimization by introducing section terms with aim work (TRPO). Suppose that the probability ratio between the old and new techniques is as follows:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \quad (10)$$

The clipping and traditional goals are utilized to calculate PPO target efficiency [12]. Major strategic alterations are discouraged by PPOs outside of the trimmed span. As a result, by restricting arrangement updates at each preparation phase, PPO enhances the strength of strategy network preparation. Because PPO is more dependable, swift, and user-friendly, experts prefer it for stock trading.

3. *Advantage Actor Critic (A2C)*

As part of the ensemble approach, researchers used A2C, a popular actor-critic computation. The A2C debut incorporates enhanced layout gradient upgrades. Benefit capacity is used by A2C [13] by decreasing the gradient of the strategy difference. A critical network judges the worth of the work rather than merely estimating it. As a result, the evaluation of activity takes into account both its current quality as well as its potential for improvement. Reduce the high disparity of the arrangement organization and strengthen the model. A2C is a fantastic stock exchange model because of its stability. A2C's intended work is [13]:

$$\nabla J_\theta(\theta) = \mathbb{E}[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t)],$$

(11)

Where A(st, at) is the policy network and (at |st) is The advantage function is expressed as:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t),$$

$$A(s_t, a_t) = r(s_t, a_t, s_{t+1}) + \gamma V(s_{t+1}) - V(s_t).$$

(12)

4. *Soft Actor-Critic (SAC)*

The Soft Actor-Critic (SAC) technique bridges the gap between stochastic policy optimization and DDPG-style methods by optimizing a stochastic policy in an off-policy manner. Although it was released roughly concurrently with TD3, it is not a straight replacement for TD3. Nevertheless, it features the clipped double-Q technique and gains from target policy smoothing because of the policy's intrinsic stochastic. Entropy regularization is one of SAC's key characteristics. Entropy, a gauge of the policy's unpredictability, and anticipated returns are trade-offs that the policy is trained to optimize. This is closely related to the exploration-exploitation trade-off: when entropy rises, exploration increases, which speeds up learning later on. Additionally, it can stop the strategy from prematurely achieving a suboptimal local optimum. Entropy is a measurement that, in general, indicates how unpredictable a random variable is. Low entropy refers to a coin that nearly usually lands heads; high entropy refers to a coin that is uniformly weighted and has a 50% probability of any result. The objective reward function is changed by soft Q-learning by including an entropy regularizing component.

The objective reward function of Q-learning

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \sum_t \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [r(s_t, a_t)]$$

(13)

Objective Soft Q-learning reward function

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \sum_t \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [r(s_t, a_t) + \alpha \mathcal{H}(\pi(.|s_t))]$$

(14)

5. *Twin Delayed DDPG (TD3)*

TD3 is a modified version of DDPG. Sometimes, DDPG can perform well, but a lot relies on the hyperparameters used. To increase the stability of the training process, TD3 employs three crucial methods. A method called Twin Delayed DDPG (TD3) resolves this problem by using three crucial techniques:

1: Clipped Double-Q Learning. To generate the targets in the Bellman error loss functions, TD3 learns two Q-functions rather than one (thus, "twin"), and employs the lesser of the two Q-values.

$$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\phi_{\text{t arg},i}}(s', \tilde{a}'(s))$$

(15)

2: "Delayed" Policy Updates. Compared to the Q-function, TD3 changes the policy (and target networks) less frequently. One policy modification is suggested for every two Q-function revisions in the study.
3: Target Policy Smoothing. To make it more difficult for the policy to take advantage of Q-function flaws by smoothing out Q along with changes in action, TD3 introduces noise into the target action.
Together, these three techniques produce a performance that is noticeably better than DDPG's base level.

$$\max_{a'} Q_\phi(s', a') = Q_{\phi_{\text{targ}}}(s', \pi_{\theta_{\text{targ}}}(s') + \varepsilon)$$

(16)

6. *Ensemble Strategy*

The aim is to create a highly reliable trading strategy. To choose the best performing agent among DDPG,PPO,A2C,SAC, and TD3 to trade based on the Sharpe ratio, designers use an ensemble approach. The ensemble procedure is explained as follows:

*Step 1:*All five of the agents will be simultaneously retrained over an increasing number of n months. In this study, all five agents underwent a three-month cycle of retraining.

Step 2: To decide which agent performs the best and has the highest Sharpe ratio, researchers validate all five agents using a rolling validation window that lasts three months after the training window.[14].The formula for the Sharpe ratio is:

$$Sharpe\ ratio = \frac{\bar{r}_p - r_f}{\sigma_p}, \quad (17)$$

where r_f is the risk-free rate, σp is the portfolio standard deviation, and rp is the anticipated portfolio return. A turbulence indicator was also used by researchers to alter risk aversion throughout the validation process.

Step 3. The best agent is chosen to forecast and execute trades for the upcoming quarter, researchers use it. This decision was made since every trading agent is responsive to various trends. One agent behaves poorly in a negative trend but well in a bullish one. A different agent is more used to a choppy market. The better an agent's returns have been in comparison to the level of investment risk it has assumed, the greater its Sharpe ratio. As a result, Researchers determine the trading methods that can increase gains while adjusting for increased risk.

V. Performance Assessments

- Researchers evaluate the five algorithms. The results demonstrate that, in terms of return, the five agents recommended techniques surpass both the Dow Jones Industrial Average and the traditional min-variance portfolio allocation strategy.
- Load Python packages
  1. Install Packages
  2. Check Additional Packages
  3. Import Packages
  4. Create Folders
- Download Data
- Preprocess Data
  1. Technical Indicators
  2. Perform Feature Engineering
- BuBuildingnvironment
  1. Training & Trade Data Split
  2. User-defined Environment
  3. Initialize the Environment
- Implement DRL Algorithms
  1. A2C
  2. PPO
  3. DDPG
  4. SAC
  5. TD3
- Backtesting Performance
  BackTestStats & BackTestPlot
  Baseline Stats

VI. RESULTS

```
[ ] df_summary
```

	Iter	Val Start	Val End	Model Used	A2C Sharpe	PPO Sharpe	DDPG Sharpe	TD3 Sharpe	SAC Sharpe
0	126	2020-12-02	2021-03-05	TD3	0.000292	-0.045627	0.159413	0.265174	0.145403
1	189	2021-03-05	2021-06-04	A2C	0.419722	0.31768	0.339015	0.12845	0.25351
2	252	2021-06-04	2021-09-02	A2C	0.196097	0.060442	0.023344	0.119876	0.145247
3	315	2021-09-02	2021-12-02	PPO	-0.015346	0.012213	-0.232606	-0.235419	-0.157318
4	378	2021-12-02	2022-03-04	DDPG	-0.13029	-0.207384	-0.031456	-0.237358	-0.097931

```
==============Get Backtest Results===========
Annual return          0.116711
Cumulative returns     0.147958
Annual volatility      0.169595
Sharpe ratio           0.737854
Calmar ratio           0.655508
Stability              0.673635
Max drawdown          -0.178047
Omega ratio            1.137402
Sortino ratio          1.043090
Skew                        NaN
Kurtosis                    NaN
Tail ratio             1.053905
Daily value at risk   -0.020870
dtype: float64
```

```
==============Get Baseline Stats=========
[**********************100%***************
Shape of DataFrame:  (314, 8)
Annual return          0.033420
Cumulative returns     0.041812
Annual volatility      0.148495
Sharpe ratio           0.296312
Calmar ratio           0.221733
Stability              0.000918
Max drawdown          -0.150722
Omega ratio            1.051277
Sortino ratio          0.405307
Skew                        NaN
Kurtosis                    NaN
Tail ratio             0.979751
Daily value at risk   -0.018534
dtype: float64
```

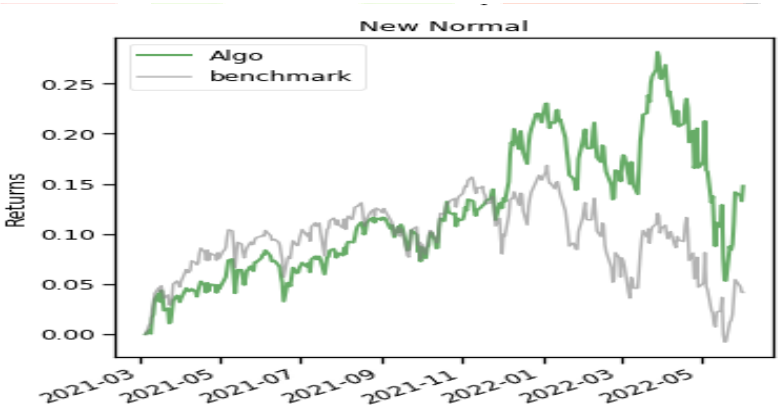
```
==============Compare to DJIA===========
[**********************100%**********************]  1 of 1 completed
Shape of DataFrame:  (314, 8)
```

Start date	2021-03-05
End date	2022-06-02
Total months	15

	Backtest
Annual return	11.671%
Cumulative returns	14.796%
Annual volatility	16.959%
Sharpe ratio	0.74
Calmar ratio	0.66
Stability	0.67
Max drawdown	-17.805%
Omega ratio	1.14
Sortino ratio	1.04
Skew	NaN
Kurtosis	NaN
Tail ratio	1.05
Daily value at risk	-2.087%
Alpha	0.08
Beta	1.05

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	17.80	2022-03-29	2022-05-20	NaT	NaN
1	7.77	2022-01-04	2022-02-23	2022-03-22	56
2	4.70	2021-06-04	2021-06-18	2021-07-26	37
3	3.92	2021-09-01	2021-09-30	2021-10-19	35
4	3.15	2021-05-10	2021-05-12	2021-06-01	17

Stress Events	mean	min	max
New Normal	0.05%	-4.75%	3.39%



VII. CONCLUSION & FUTURE WORKS

This research explores the use of Deep Deterministic Policy Gradient (DDPG), Advantage Actor Critic (A2C), Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), and Twin Delayed DDPG (TD3) agents as deep reinforcement algorithms in a learning stock exchanging system.To adjust to changing market conditions, researchers use an ensemble technique to automatically select the best performing agent to trade based on the Sharpe ratio. The study demonstrates that by balancing risk and return while accounting for transaction costs, The five distinct strategies perform better than the combined strategy., the Dow Jones Industrial Average, and the min-variance portfolio allocation method in terms of the Sharpe ratio. Under various market conditions, researchers will use an exchange approach that eventually chooses the best performing agent through exchange dependent on the Sharpe ratio.

Future Prospects

- Improve Space and Time Complexity of the Algorithm
- More Variables like Alternative Data, Company Account Details, Sentiment, etc
- Results for all Stocks Companies
- Comparison of different company stock informaGiveving more suggestions or results using financial information
- More assured stock data from other APIs using the APICreateking a software or app using GUI
- Createaking a package to generate result reported ports on GitHub
- Tuning more time series models and other models

REFERENCES

1. Yang, Hongyang and Liu, Xiao-Yang and Zhong, Shan and Walid, Anwar, Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy (September 11, 2020). Available at SSRN: http://dx.doi.org/10.2139/ssrn.3690996
2. ASHISH SHARMA, DINESH BHURIYA, UPENDRA SINGH In "Survey of Stock Market Prediction Using Machine Learning Approach"
3. M. H. Pesaran and A. Timmermann, "Predictability of stock returns: Robustness and economic significance," The Journal of Finance, vol. 50, no. 4, pp. 12011228, 1995.
4. Thomas G. Fischer, "Reinforcement learning in financial markets - a survey," FAU Discussion Papers in Economics 12/2018, Friedrich Alexander University Erlangen-Nuremberg, Institute for Economics, 2018.
5. A. Ilmanen, "Expected returns: An investor's guide to harvesting market rewards," 05 2012.
6. Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, "Openai gym," 2016.
7. Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Nicholsichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov, "Openai baselines," https://github.com/openai/baselines, 2017.
8. Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Nicholsichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu, "Stable baselines," https://github.com/hill-a/stable-baselines, 2018.
9. Richard Sutton and Andrew Barto, "Reinforcement learning: an introduction," *IEEE Transactions on Neural Networks*, vol. 9, pp. 1054, 02 1998.
10. Richard Sutton, David McCallister, Satinder Singh, and Yishay Man sour, "Policy gradient methods for reinforcement learning with function approximation," *Conference on Neural Information Processing Systems (NeurIPS), 1999*, 02 2000.
11. Timothy Lillicrap, Jonathan Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra, "Continuous control with deep reinforcement learning," *International Conference on Learning Representations (ICLR) 2016*, 09 2015.
12. Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and A. Elwalid, "Practical deep reinforcement learning approach for stock trading," *NeurIPS Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy, 2018.*, 2018.
13. Thomas G. Fischer, "Reinforcement learning in financial markets - a survey," FAU Discussion Papers in Economics 12/2018, Friedrich Alexander University Erlangen-Nuremberg, Institute for Economics, 2018.
14. W.F. Sharpe, "The Sharpe ratio," *Journal of Portfolio Management*, 01 1994.