A Deep Deterministic Policy Gradient-based Stock Automated Trading System

*Machine Learning(1001),Group3

Rongcheng LI
Faculty of Science and Technology
Department of Computer Science and Techology
BNU-HKBU United International College, Zhuhai, China
Student ID:2130026074
Email:r130026074@mail.uic.edu.cn

Ruowen Jing
Faculty of Science and Technology
Department of Computer Science and Techology
BNU-HKBU United International College, Zhuhai, China
Student ID:2130026066
Email: r130026066@mail.uic.edu.cn

Jianwen Li
Faculty of Science and Technology
Department of Computer Science and Techology
BNU-HKBU United International College, Zhuhai, China
Student ID:2130026071
Email: r130026071@mail.uic.edu.cn

Jiewen Tan
Faculty of Science and Technology
Department of Computer Science and Techology
BNU-HKBU United International College, Zhuhai, China
Student ID:2130026132
Email: r130026132@mail.uic.edu.cn

Yixuan Ji
Faculty of Science and Technology
Department of Computer Science and Techology
BNU-HKBU United International College, Zhuhai, China
Student ID:2130026059
Email: r130026059@mail.uic.edu.cn

***Abstract*—In recent years, there has been a significant increase in the efforts to develop and apply AI techniques for various aspects of finance research and applications. One of the examples of how AI techniques (e.g., machine learning) can benefit the finance sector is by enabling traders to automate two important tasks in quantitative trading (QT), which is a method of trading that relies on mathematical models and algorithms. These two tasks are: recognizing the current and future market conditions based on the available data and signals, and executing the optimal trading stratetgies that can maximize the profits and minimize the risks.The main objective of this project is to apply some Deep Reinforcement Learning (DRL) algorithms like Deep Deterministic Policy Gradient (DDPG) to the tasks of prediction and trading strategies in the finance domain.We have integrated DDPG into the FinRL architecture and conducted model training on Colab, utilizing DOW30 stock data. Our model achieved a 5% annual return.Throughout our lab experiments, we identified two issues with our model. Firstly, there is room for improvement in its performance. Secondly, the model lacks stability. In future endeavors, we plan to explore the utilization of Quantum Lever Prices (QPLs) to enhance our model and undertake additional efforts in data preprocessing.**

***Index Terms*—Machine learning, Deep Reinforcement Learning, Automated Stock Trading, Deep Deterministic Policy Gradient, Actor-Critic**

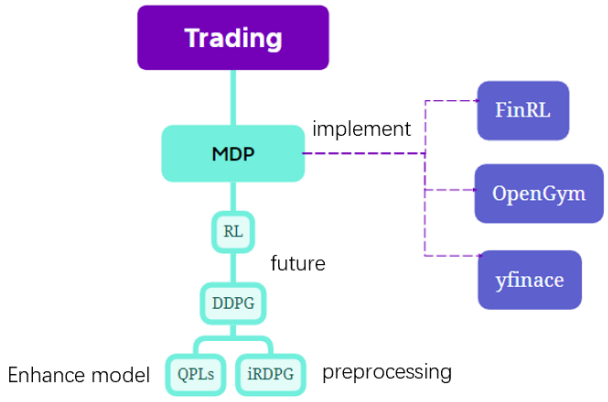

Fig. 1. Flowchart Of Project

I. Introduction

A. Project Purpose

Big data is an area that has been greatly influenced by the development of technology. With the availability and

accessibility of large amounts of financial data, we can use machine learning methods to help people make better use of their portfolio. One of the machine learning methods that we plan to implement is Deep Reinforcement Learning (DRL), which is a type of learning that involves an agent interacting with an environment and learning from its own actions and rewards. We aim to apply some DRL algorithms, such as Deep Deterministic Policy Gradient (DDPG), to the tasks of prediction and trading strategies in the finance domain. We want to gain a deep understanding of how these algorithms work and what are their strengths and weaknesses, in order to prepare ourselves for future work in this field. We will use FinRL, which is a Python library for DRL in finance, to implement these algorithms first, and learn about them.

B. Proposed Method

We opt for the Deep Deterministic Policy Gradient (DDPG) algorithm, a reinforcement learning approach rooted in Policy Gradient methodology.

Compared to Value-based algorithms such as Q-learning, Policy Gradient(PG) do not calculate the value of each action, it just directly random choose one of the actions.Policy Gradient do not calculate loss, it use reward to reinforce the model to have higher possibility to choose the action which is right.

DDPG operates as an actor-critic network, combining aspects of both Policy Gradient and Value-based reinforcement learning. The actor, embodying the Policy Gradient, randomly selects actions, while the critic evaluates whether the chosen actions are advantageous or not.

Traditional reinforcement learning methods, which employ tables to store state-value pairs, prove impractical in environments with numerous states. Deep Neural Networks, proficient at handling extensive input data and producing corresponding output, address this limitation effectively.

Deep Deterministic Policy Gradient‘s advantage is that it makes decisions on continuous actions, so it is more suitable for the complex and changing stock environment.

C. Experiment and Result

Our DDPG stock trading agent performed well in the stock market, generating a profit of 1,830 US dollars from an initial capital of 10,000 US dollars. The performance indicators of our algorithm were as follows:

- Annual return: This measures the percentage change in the value of our portfolio over one year. Our annual return was 0.056101247, which means our portfolio increased by 5.61% in value over one year.
- Cumulative returns: This measures the total percentage change in the value of our portfolio over the entire period. Our cumulative returns were 0.12505537, which means our portfolio increased by 12.51% in value over the entire period.
- Annual volatility: This measures the standard deviation of the daily returns of our portfolio over one year. It reflects the risk or uncertainty of our portfolio. Our annual volatility was 0.186804106, which means our portfolio had a high degree of variation in its daily returns over one year.
- Sharpe ratio: This measures the ratio of the excess return of our portfolio over the risk-free rate to the annual volatility of our portfolio. It reflects the risk-adjusted return of our portfolio. Our Sharpe ratio was 0.386047138, which means our portfolio had a low risk-adjusted return over one year

D. Discussion and Future Work

In contrast to alternative enhanced DDPG algorithms, our approach demonstrates superior performance in both profitability and stability. While we have identified several avenues for enhancing our model, implementation is pending due to time constraints.

Since some research already achieved some results, use Quantum Lever Prices(QPLs) from Quantum Finance Theory (QFT) to upgrade reinforcement learning model. In the future, after we learn more about the theory of quantum finance theory, we will also try to improve and enhance our model, and increase the capabilities of our stock trading AI agent.

II. Literature Review

A. Quantitative Trading

Quantitative Trading is a method of conducting financial transactions using mathematical models, statistical analysis, and computer algorithms. It relies on systematic rules and models to extract potential patterns from a large amount of market information, culminating in highly precise and efficient trading decisions. It enables more precise and systematic decision-making compared to the empirical thinking patterns of humans. In recent decades, improvements in computing power and the increased availability of big data have reduced the cost of building and utilizing these models. Machine learning models play a crucial role in quantitative trading. They represent a process that extensively uses statistical models and methods. However, due to their inherent inability to clearly distinguish the relationships established between explanatory variables (inputs) and dependent variables (outputs), they are often regarded as black-box models.

Quantitative trading can improve decision-making efficiency through machine learning and related algorithms, leveraging the enhanced computing power, which significantly boosts overall economic decision-making efficiency. To a large extent, it reduces the likelihood of decision errors. This is achieved by conducting in-depth analysis of market volatility through machine learning algorithms. By learning from historical data, the model can accurately predict future volatility levels. Additionally, machine learning algorithms can analyze asset correlations by establishing specific correlation models. This enables the identification of correlation levels between different elements and facilitates more accurate quantitative transaction predictions.

B. Portfolio management

Dynamic portfolio optimization refers to maximizing investment returns or reducing risks by dynamically adjusting portfolio weights according to market conditions and asset changes in portfolio management. This is an investment strategy based on time changes and market fluctuations. Traditional investment optimization is usually based on a static model, that is, the optimal asset allocation is selected at a fixed point in time, and then left unchanged until the next point in time to rebalance. While dynamic portfolio optimization is a continuous process, compared with static portfolio optimization methods, more flexible, can better adapt to the changing market environment.

Dynamic portfolio optimization typically relies on market predictions. Investors use various methods, including statistical models, time series analysis, machine learning algorithms, etc., to analyze historical data and other market information to forecast future market trends. Investors also need to understand the relationships between assets in different market environments by establishing a relationship model. This helps gain a better understanding of the performance of various assets in the portfolio and make timely adjustments. Dynamic portfolio optimization also emphasizes risk management. Investors must consider the overall risk of the portfolio and implement risk control strategies in times of significant market volatility. Based on market changes and investor predictions, dynamic portfolio optimization involves adjusting the weights of different assets. This may include increasing investments in assets expected to perform well, reducing investments in those expected to perform poorly, or implementing other adjustment strategies to adapt to new market information. While dynamic portfolio adjustments can be made in the short term, regular rebalancing is often necessary. Rebalancing ensures that the portfolio continues to align with the investor's goals and risk preferences over time. Unlike static portfolios, dynamic portfolios typically require more frequent real-time decision-making. This may necessitate more robust technology and systems to support real-time decision-making processes.

C. Deep Learning

Deep Learning (DL) is one of the hottest topics in Machine Learning (ML). Recently, a lot of researchers are trying to figure out Dynamic Portfolio Optimization (DPO) methods through the DL approach. The performance of these DL models varies from good to bad.

Powerful feature learning capability: DL models can extract useful information and characteristics from complex financial data and learn from them, later the predictions of asset prices and stock valuation will be made based on these models. This procedure may improve the effectiveness of the process of DPO. On top of that, DL models can also capture long-term dependencies in time series data, which may help researchers to better predict the movement of the financial market.

Strong adaptability: DL approach DPO methods can meet the requirements of different financial markets, and it show good generalization ability over different financial markets.

High Sharpe Ratio: The DPO methods based on DL can also combine several other ML techniques, such as Fuzzy Learning (FL) and Reinforcement Learning (RL), to improve the overall performance. Compared to traditional Quantitative Trading (QT) strategies DL-based DPO methods usually have higher Sharpe ratios. In other words, it means that the investment portfolio has higher profitability when the risk level are the same.

Altogether, DL models possess strong characteristic learning ability and adaptability, they adapt to different markets and outperform traditional quantitative strategies with higher Sharpe ratios in certain market environments. DPO methods that based on DL have some theoretical advantages, but there are still some shortcomings and shortcomings in practical applications.

Strong data dependency: DL models depend on data of high quality. But in the real financial markets, the data tends to have a lot of noise and instability, this may lead to a decrease in model performance in real-world applications.

Overfitting rick: Because of the noise interference or the high complexity of model, DL models may be overfitting. In financial markets, an overfitting model may generate financial strategies which perform well in the history data or training data, but behave bad in the practical data or the testing data.

Insufficient model interpretations: Currently, the DPO methods which is based on DL may focus on the performance or predictive ability of the model, rather than the explanatory power. That is the reason why DL models are often considered as "black boxes". In financial field, the investors may find it difficult to understand the underlying logic of the DL models. Then, although the DL-based DPO strategies may sometimes profit a lot, but those investors may stubbornly treat them as a kind of luck, which cannot depend on.

A long training time: Due to the complexity of financial data and DL modeling structure, it may be time consuming to come up with the DPO methos based on the DL technique. However, in the real financial markets, real-time demand is usually significant. As a result, using DL-based models while making DPO strategies may cause some unavoidable delay.

Higher complexity: Compared to traditional QT strategies, DL methods may have higher complexity. The training and optimizing process of ML models need a large amount of computational resource, and the selection of model parameters and hyperparameter tuning is relatively more complicated, which increases the difficulty of practical application, and the requirements for researchers are much higher.

Regulatory policy and ethical issues: In financial field, the application of DL models may be limited by the regulatory policy and ethical issues, such as data privacy, algorithmic transparency, and fairness.

Uncertain accuracy: In the practical applications, DPO strategies based on DL may be affected by transaction costs, market frictions, and other factors, which may lead to a decline in strategy performance. Although DL applications have achieved some significant results in some areas, its application in the financial field is still under exploration. DL

models are still struggled to handle some complex financial products, such as financial derivatives and structured products.

D. Reinforcement Learning

Portfolio optimization has been a critical area of interest in the field of finance, aiming to maximize returns while minimizing risk. Traditional methods often rely on mean-variance optimization and other mathematical frameworks, but the dynamic and uncertain nature of financial markets demands more adaptive and sophisticated approaches. Reinforcement Learning (RL) has emerged as a promising paradigm for addressing the challenges of portfolio optimization.The financial markets' complexity and constant evolution pose challenges to traditional portfolio optimization techniques. RL, inspired by behavioral psychology, has gained attention for its ability to learn optimal strategies through interactions with the environment.

Traditional portfolio optimization relies heavily on assumed statistical distributions and historical data, limiting adaptability to changing market conditions. Model-free RL approaches, such as Q-learning and Deep Q Networks (DQN), empower portfolios to learn from real-time market data, adapting to emerging trends and mitigating the impact of unexpected events. Policy gradient methods, including Proximal Policy Optimization (PPO) and Trust Region Policy Optimization (TRPO), enable agents to directly learn policies without explicit knowledge of the underlying market dynamics. This approach proves beneficial in handling non-linear and complex market structures. RL algorithms excel in dynamically allocating assets based on changing market conditions. By continuously learning from the environment, portfolios can adjust their allocations to optimize returns and manage risk effectively. RL has been successfully applied to develop algorithmic trading strategies. Agents trained through RL can discover and exploit patterns in historical data, leading to improved trading performance and enhanced profitability.

However, RL has some limitations on DPO problem. One of the primary challenges in RL-based portfolio optimization is the requirement for extensive training data. Real-world financial data is often limited, making it challenging to train robust and reliable models. Financial markets are inherently non-stationary, with dynamics changing rapidly. RL models may struggle to adapt quickly to unforeseen events or structural shifts in the market, leading to suboptimal performance. Reinforcement Learning holds great promise in revolutionizing portfolio optimization by providing adaptive, data-driven strategies. While the field has made significant strides, addressing challenges related to data efficiency, non-stationarity, and interpretability remains crucial for the widespread adoption of RL in financial markets. Future research should focus on developing more robust algorithms that can effectively navigate the complexities of dynamic and unpredictable financial environments.

E. Deep Reinforcement Learning

One of the challenges that classical reinforcement learning (RL) methods face is the selection of the appropriate market features that can capture the dynamics and patterns of the problem. Deep learning (DL) methods, on the other hand, have the advantage of being able to handle large input states effectively and efficiently. Deep reinforcement learning (DRL) is a novel approach that integrates RL and DL to solve problems that involve high dimensional data. DRL has demonstrated remarkable performance in complex tasks, such as playing video games at a superhuman level (Mnih et al. 2015). DRL also has the potential to be applied to QT, which is a form of trading that uses mathematical models and algorithms to analyze and execute trades. For instance, Jiang, Xu, and Liang (2017) employed the model-free Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al. 2015) to dynamically optimize cryptocurrency portfolios based on the market conditions and the expected returns. Deng et al. (2016) enhanced financial signal representation by extending fuzzy learning and deep neural networks (DNNs), which are powerful DL models that can learn from complex and nonlinear data. However, these model-free RL methods have a drawback of being sampling inefficient for large state space problems like QT, which means that they require a lot of data and interactions to learn a good policy. Yu et al. (2019) proposed a model-based RL framework for daily frequency portfolio trading, which is a trading frequency that uses daily data to make decisions. However, QT often uses minute-frequent data (Pruitt and Hill 2012), which is a trading frequency that uses data from every minute to make decisions. Minute-frequent data is too fast and too large for human traders to synthesize and process, but not for algorithms that can use DRL to automate and optimize the trading process.

F. Quantum Finance Theory

One of the ways to enhance the performance of DRL model is to use Quantum Price Levels (QPLs), which are derived from Quantum Finance Theory (QFT). QPLs can capture the quantum fluctuations and non-linear dynamics of the financial markets, and provide more accurate and robust signals for the DRL model to learn and act. Two recent studies have applied QPLs to DRL models for stock price prediction and trading strategy optimization, and have achieved good results. The first study by Qiu, Liu and Lee (2023) used QPLs to guide a DRL model. The second study by Lin, Xing, Ma and Lee (2023) used QPLs to enhance a DRL model to optimize the trading strategy of a portfolio. Both studies showed that QPLs can improve the performance of DRL models.

Quantum finance is a branch of finance that applies the principles and methods of quantum theory and quantum mechanics to the study of financial markets and their dynamics. Quantum finance uses the concept of quantum financial particles (QFPs), which are the basic units of financial activity in every market. QFPs are analogous to quantum particles in physics, such as electrons and photons, which have both wave and particle properties. QFPs can also exhibit quantum phenomena, such

as superposition, entanglement, and interference, which can affect the behavior and the outcome of the financial markets. By studying the dynamics of these QFPs, quantum finance aims to better understand the patterns and the future movements of the financial markets, and to develop more accurate and robust models and strategies for financial analysis and decision making. Quantum finance is a relatively new and emerging field, which has the potential to revolutionize the way we view and interact with the financial markets.

III. Problem Statement

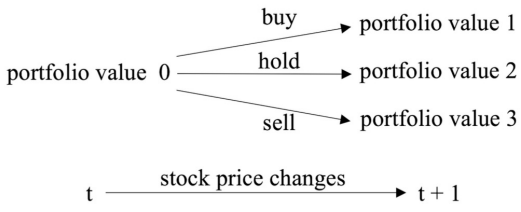

Fig. 2. Markov Decision Process

The objective of this paper is to maximize (expected) cumulative stock returns, transform trading decision problems into optimization problems using MDP mathematical framework, and train agents to learn optimal strategies using reinforcement learning algorithms. MDP is a mathematical framework for describing decision making problems in reinforcement learning. This mathematical framework models an interactive process between an Agent and an environment in which the agent makes decisions, performs actions, and receives rewards or feedback from the environment in different states. The basic elements of MDP include:Status, action, and reward functions. States describe the various situations or configurations that a system may be in. Actions are the actions an agent can take in each state. The reward function is a given state and action, and the reward function defines the reward that the agent receives in time after performing the action. MDP model is widely used in reinforcement learning problems to help agents learn to make optimal decisions in different situations.

IV. Deep Deterministic Policy Gradient-based Stock Automated Trading System

We opt for the Deep Deterministic Policy Gradient (DDPG) algorithm, a reinforcement learning approach rooted in Policy Gradient methodology.

In contrast to Value-based algorithms like Q-learning, Policy Gradient avoids calculating the value of each action and instead directly selects an action randomly from the available choices. This method refrains from computing loss explicitly and relies on rewards to reinforce the model, encouraging a higher likelihood of selecting actions that lead to favorable outcomes.

DDPG operates as an actor-critic network, combining aspects of both Policy Gradient and Value-based reinforcement

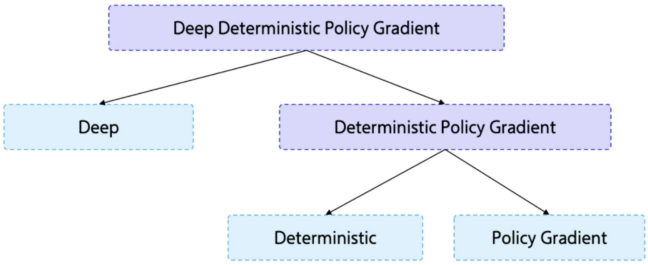

Fig. 3. DDPG

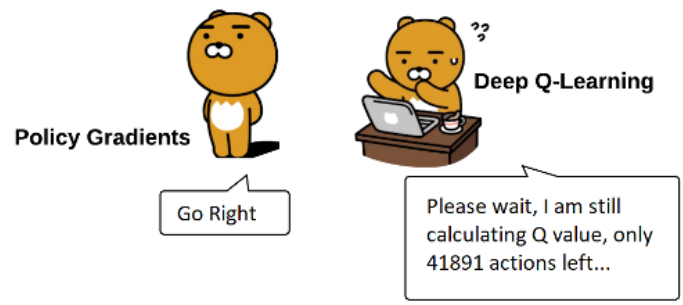

Fig. 4. Policy Gradient

learning. The actor, embodying the Policy Gradient, randomly selects actions, while the critic evaluates whether the chosen actions are advantageous or not.

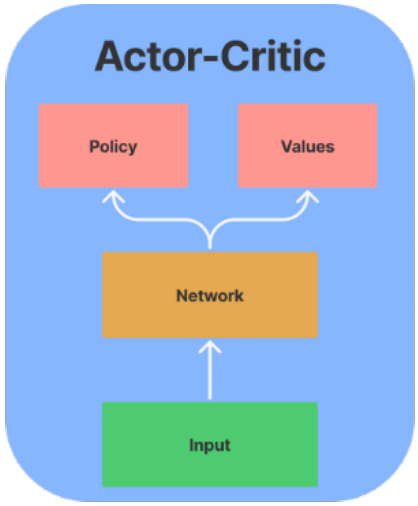

Fig. 5. Actor-Critic

Traditional reinforcement learning methods, which employ tables to store state-value pairs, prove impractical in environments with numerous states. Deep Neural Networks, proficient at handling extensive input data and producing corresponding output, address this limitation effectively.

The distinct advantage of DDPG lies in its capability to make decisions involving continuous actions, rendering it well-suited for navigating the intricate and dynamic stock market environment.

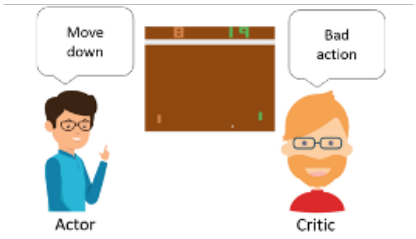

Fig. 6. Actor-Critic Example

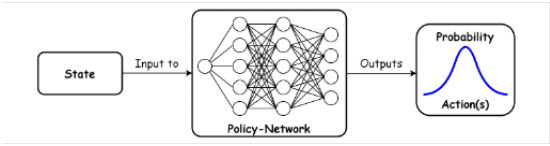

Fig. 7. Deep Nerual Network in Reinforcement Learning

In order to achieve the highest possible return on investment, we use a DDPG algorithm, which is an enhanced version of the Deterministic Policy Gradient (DPG) algorithm . The DPG algorithm is a hybrid approach that integrates the ideas of both Q-learning and policy gradient methods. Unlike DPG, DDPG employs neural networks to approximate the functions involved in the algorithm. The DDPG algorithm that we use in this section is tailored for the MDP model that represents the stock trading market.

Q-learning is a method to learn the environment. Rather than using the expected value of $Q(s_{t+1}, a_{t+1})$ to update $Q(s_t, a_t)$, Q-learning uses the action a_{t+1} that gives the highest $Q(s_{t+1}, a_{t+1})$ for the next state s_{t+1}, that is,

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{s_{t+1}}\left[r(s_t, a_t, s_{t+1}) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})\right]$$

DDPG consists of an actor network and a critic network. The actor network , maps states to actions, representing the actor network parameters. On the other hand, the critic network, calculates the value of an action within a given state, being the parameters of the critic network. To enhance exploration for improved actions, noise is introduced to the output of the actor network. This noise is obtained by sampling from a random process.

V. Experimental Results and Analysis

A. Experiment Set Up

In this study, we utilized the colab platform to execute our project, employing the jupyter notebook as the main tool.
We plan to use Financial Reinforcement Learning(FinRL) to get the data and implement the algorithm first, later we may do some adjustment to the algorithm.

FinRL is a Python library that enables us to apply deep reinforcement learning (DRL) algorithms to the tasks of prediction and trading strategies in the finance domain. FinRL supports various markets, such as stock, forex, crypto, etc., and provides a unified interface for different DRL algorithms, such as DDPG, PPO, SAC, etc. FinRL also offers benchmarks of many quant finance tasks, such as portfolio management, multi-agent trading, etc., and allows us to test our models in live trading environments. FinRL is an open source project that aims to facilitate the research and development of financial reinforcement learning.

B. Data Pre-processing

1) Loading Data: In the data acquisition stage of this project, we first selected six representative stocks in the technology and electric vehicle fields, namely: AAPL (Apple), GOOGL (Alphabet Inc., the parent company of Google), AMZN (Amazon), TSLA (Tesla), and MSFT (Microsoft).

In order to ensure the accuracy and completeness of the data, we have chosen Yahoo Finance as the data source because it has the characteristics of providing financial data quickly, accurately, and comprehensively. Meanwhile, we have set a time range for the data, including training data (2010-2021) and trading data (2021-2023).

2) Pre-processing: After obtaining the data, we carried out data cleaning and preprocessing steps, briefly handling missing values and outliers. Then add technical indicators in DataFrame to describe the changes in stock prices. Technical indicators are quantitative indicators calculated using data such as price, trading volume, and time to predict future stock price trends. In our example, we selected several commonly used technical indicators, such as Moving Average Convergence Divergence, Relative Strength Index, Volatility Index, Simple Moving Average, and Commodity Channel Index.

After simple preprocessing of the data, we used a Savitzky Golay filter to smooth the data. Savitzky Golay filter is a filter used for smoothing signals, which finds the best fitting curve of data by minimizing a quadratic function. This method is more complex than a simple moving average filter, but it can provide higher smoothness. In this experiment, our data achieved good smoothing results, which is very helpful for directly observing data trends and subsequent model training.

C. Building Reinforcement Learning Environment

The goal of Reinforcement Learning is to find an optimal strategy within a limited time to maximize the cumulative reward. The environment is very important for Reinforcement Learning, because the agent need to get feedback for each episode. The environment can also provide the agent to learn and practice in a controlled manner, and then apply to the real world.

There are lots of environments for RL training. For example, Carla is an open-source simulator for autonomous driving research, and AirSim is another simulator developed by Microsoft for research on drones and robots. For our research, we choose to use OpenAI GYM. OpenAI GYM provides a series of standard environments for algorithms testing, this helps to reduce the difference in the experiment, so that

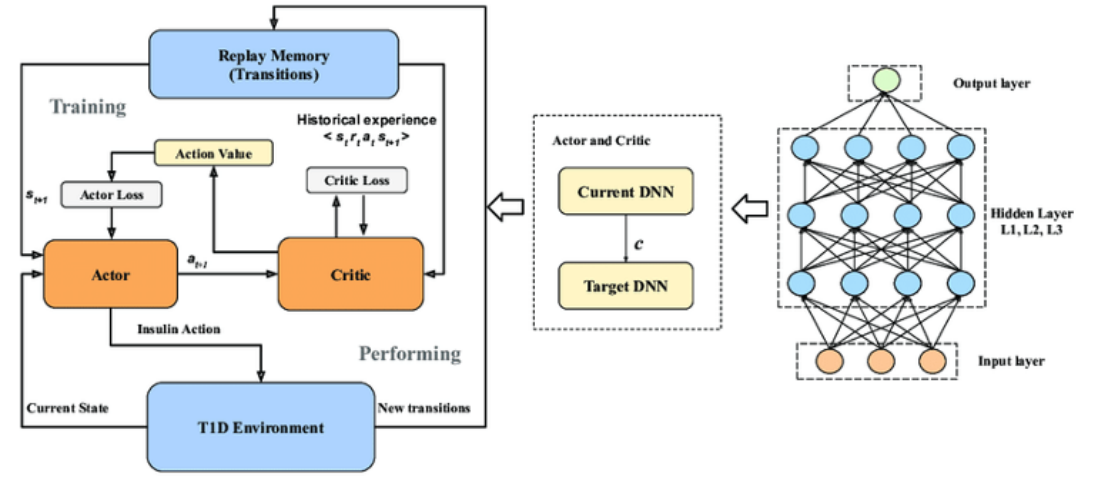

Fig. 8. DDPG Architecture

Fig. 9. Experiment Environment

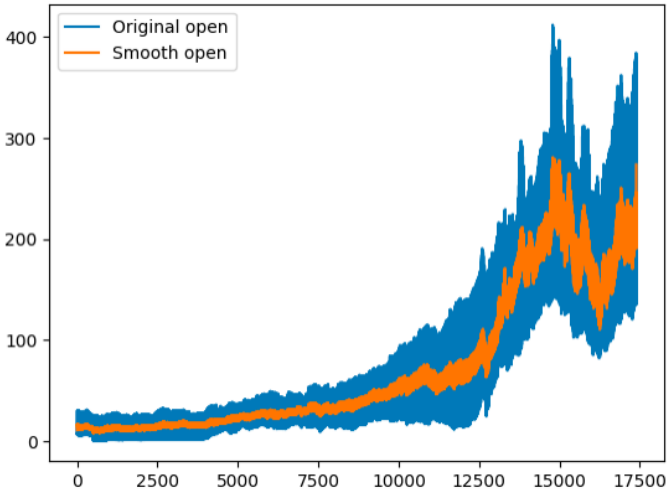

Fig. 10. pre-processed data example

different algorithms can be directly compared. OpenAI GYM is relatively famous in RL area, so there are many tutorials and communities on the Internet, which is more user friendly.

D. Experiment Matrix

1) Annual Return: Definition: The annual yield is a metric used to measure the profitability level of an investment within a one-year period. It represents the percentage of returns achieved by the investment relative to the initial investment. Annual yield is a key indicator in the evaluation of investment performance, providing a concise way to understand the investment's performance over a relatively short time.

$$\text{Annual Return} = \left(\frac{\text{final value} - \text{initial value}}{\text{initial value}}\right) \times 100$$

2) Cumulative Returns:

$$\text{Cumulative Return} = \left(\frac{\text{final value} - \text{initial value}}{\text{initial value}}\right) \times 100$$

This signifies that, within this year, the investment yielded a positive return of 44.08%. A positive value of the annual yield indicates that the investment generated profits, whereas a negative value suggests that the investment incurred losses.

3) Annual Volatility: Definition: Annual Volatility is a metric that measures the volatility of an investment or asset price, reflecting the extent of price fluctuations within a one-year period. For investors and risk managers, understanding the volatility of assets is paramount, as it directly relates to the risk level of a portfolio. The computation of annual volatility typically involves utilizing historical data of asset or portfolio prices and employing statistical methodologies to estimate future price fluctuations. The most common calculation method involves the use of the standard deviation of prices, a statistical measure of the degree of dispersion in data.

General steps for calculating annual volatility:

1. Historical Price Data Collection: Acquire historical

price data for the asset or investment portfolio, typically encompassing daily, weekly, or monthly price points.
2. Calculate the rate of return: Calculate the rate of return for each adjacent price point, that is, the percentage change of the current price relative to the previous price. The formula for calculating the rate of return is:
3. Calculate Standard Deviation: Utilize the computed return data to calculate their standard deviation, representing the measure of volatility. Standard deviation signifies the degree of dispersion of data points relative to the mean.
4. Annualization: Multiply the computed standard deviation by the square root of the number of trading days, typically 252. This adjustment is made considering that there are approximately 252 trading days in a year.
5. Get the annual volatility: The final result is the annual volatility, usually expressed as a percentage.

$$\text{Rate of Return} = \left(\frac{\text{Final value} - \text{Initial value}}{\text{Initial value}} \right) \times 100$$

Explanation of Annual Volatility: If the annual volatility is 20%, it implies that within one year, the standard deviation of price fluctuations for the asset or investment portfolio accounts for 20% of its mean. Higher annual volatility typically indicates higher risk, as the magnitude of price fluctuations is greater, while lower annual volatility suggests lower risk. Investors can utilize annual volatility to assess and compare the risk levels of different assets or investment portfolios, incorporating it into risk management decisions.

4) Sharpe Ratio: Preface: A prevalent characteristic in investment is the conventional understanding that as the expected returns of an investment target increase, investors are more willing to tolerate higher levels of volatility risk. Conversely, when expected returns are lower, volatility risk tends to decrease. Therefore, the primary objective for rational investors in selecting investment targets is either to pursue the maximum returns within a fixed level of acceptable risk or to minimize risk while maintaining a fixed expected return.

Definition: The Sharpe Ratio is an indicator proposed by Nobel Prize winner William F. Sharpe for measuring portfolio performance. The Sharpe ratio is the ratio of portfolio excess return to portfolio volatility and is designed to evaluate the excess return earned by a portfolio for each unit of total risk it bears.

$$\text{Sharpe Ratio} = \frac{E(R_p) - R_f}{\sigma_p}$$

Explanation of the Sharpe Ratio:
Positive Value: A positive Sharpe Ratio indicates that the investment portfolio has achieved excess returns, and the higher the excess returns, the higher the Sharpe Ratio. This is considered a favorable evaluation of the investment portfolio.

Negative Value: A negative Sharpe Ratio signifies that the portfolio's returns are below the risk-free rate. This may suggest that investors have taken on excessive risk without corresponding returns.

Comparison: The Sharpe Ratio can be used to compare the performance of different investment portfolios because it takes into account the level of risk. A higher Sharpe Ratio typically indicates that the portfolio has achieved better excess returns under the same level of risk.

Risk-Adjusted: Since the Sharpe Ratio adjusts excess returns for the level of risk undertaken, it serves as a risk-adjusted performance measure. This is beneficial in assessing the capability of investment managers, particularly in terms of risk control.

5) Calmar Ratio: Definition: The Calmar Ratio is a metric designed to evaluate the performance of an investment portfolio, introduced by fund manager Terry W. Young in 1970. The name "Calmar" is derived from the Calmar number used in the calculation formula. The Calmar Ratio primarily focuses on risk-adjusted returns of the investment portfolio, aiming to assess the level of returns achieved by an investment strategy while accounting for the associated level of risk.

$$\text{Calmar Ratio} = \frac{\text{Annual Return}}{\text{Maximum Drawdown}}$$

Annualized Return: The average annual return of a portfolio during the examination period.

Maximum Drawdown: The maximum loss magnitude of a portfolio from any point in time to any other point in time within the examination period.

Positive Value: A positive Calmar Ratio indicates that the portfolio has achieved positive risk-adjusted returns. A higher Calmar Ratio suggests that the portfolio has achieved a higher average annualized return per unit of risk.

Negative Value: A negative Calmar Ratio indicates that the portfolio experienced negative returns during the examination period. A lower Calmar Ratio may suggest a higher level of risk, and the relatively lower annualized return may not offset this elevated risk.

Usage of Calmar Ratio: Investors can utilize the Calmar Ratio to compare the performance of different portfolios and choose those with a higher Calmar Ratio. A higher Calmar Ratio implies a higher average annualized return per unit of risk. Additionally, as the Calmar Ratio incorporates the maximum drawdown as the denominator, it places emphasis on the potential maximum loss the portfolio may face, allowing investors to consider risk more comprehensively.

6) Stability: Portfolio Stability: Stability typically refers to the volatility of a portfolio's value. For an AI decision model, different portfolios exhibit varying levels of return volatility. The goal is to identify a portfolio that strikes a balance between favorable returns and risk. Therefore, stability metrics are crucial for assessing an AI decision model, particularly in the context of finding a portfolio that optimally manages the trade-off between return and risk.

Asset Price Stability: For a single asset, stability refers to the relative steadiness of its price. This is associated with the volatility of the asset, where smaller price fluctuations are typically considered a sign of greater stability. In decision-making, there may be a preference for investing in assets with relatively stable prices rather than those with larger price fluctuations. Hence, this too is an important metric within the model.

E. Experiment Result

1) Result: Using DOW30

The summary table of DDPG compare to Dow Jones Industrial Average:

	ddpg	dji
Annual return	0.056101247	0.021609648
Cumulative returns	0.12505537	0.047322959
Annual volatility	0.186804106	0.160609239
Sharpe ratio	0.386047138	0.213590896
Calmar ratio	0.230104211	0.098490685
Stability	0.061879728	0.009037599
Max drawdown	-0.243807996	-0.219408038
Omega ratio	1.06738887	1.036753918
Sortino ratio	0.553071511	0.302426901
Skew		
Kurtosis		
Tail ratio	1.03390391	1.012362275
Daily value at risk	-0.023248934	-0.020098732

TABLE I
TABLE OF RESULT

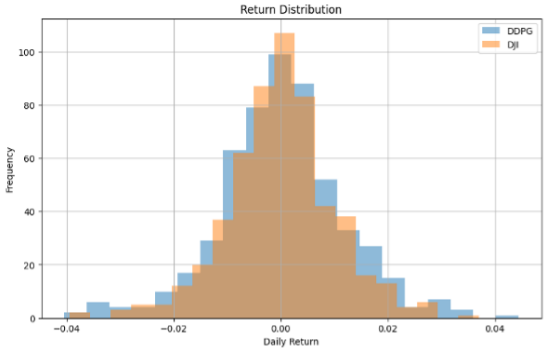

Fig. 11. Result of Distribution

2) Robustness Test: We reduced the number of stocks from 30 to five, and kept the number of iterations, the learning rate, and other conditions unchanged. We tested the robustness of the model by changing the data in different ways.

The summary table of DDPG compare to Dow Jones Industrial Average:

	ddpg	dji
Annual return	0.047937391	0.021609648
Cumulative returns	0.106365193	0.047322959
Annual volatility	0.302886214	0.160609239
Sharpe ratio	0.306100203	0.213590896
Calmar ratio	0.112961033	0.098490685
Stability	0.007240789	0.009037599
Max drawdown	-0.424371041	-0.219408038
Omega ratio	1.051904398	1.036753918
Sortino ratio	0.43647382	0.302426901
Skew		
Kurtosis		
Tail ratio	0.985659858	1.012362275
Daily value at risk	-0.037792165	-0.020098732

TABLE II
RESULT OF 5 STOCKS

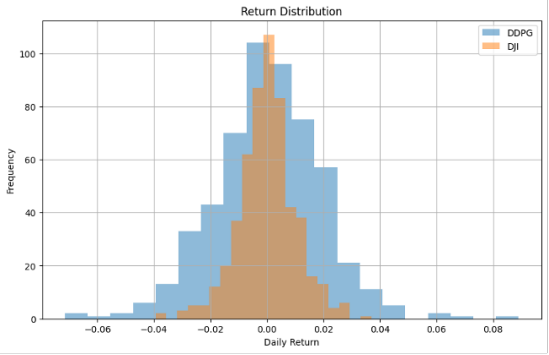

Fig. 12. Result of Distribution of 5 Stocks

VI. FUTURE WORK

In this study, we conducted an experiment to test our model based on Deep Deterministic Policy Gradient (DDPG). However, we encountered some problems and limitations of our model. We compared our model with other enhanced versions of DDPG and found that they achieved better performance in terms of profitability and stability.

We have identified some possible ways to improve our model, but we have not implemented them yet due to the time constraint. One way is to incorporate Quantum Lever Prices (QPL) from Quantum Finance Theory (QFT) into our model. QPL is a method for risk control that can enhance the DDPG model. This approach was proposed by Jiang, Z.; Xu, D.; and Liang, J. and they demonstrated its effectiveness in their paper.

Another way is to perform better data pre-processing on the stock data. Stock data contains a lot of noise and fluctuations that can affect the model's accuracy and robustness. Liu, Y., Liu, Q., Zhao, H., Pan, Z., & Liu, C. developed a model of adaptive trading model, called iRDPG, that can automatically generate Quantum Trading (QT) strategies by using an intelligent trading agent. Their model showed good results in handling noisy stock data.

VII. SUMMARY

In this project, we first collected literature and conducted in-depth research on the literature because we have a strong interest in AI-assisted financial assistants in the AI era. After that, we determined our direction to try to use deep reinforcement learning to implement an artificial intelligence agent for automatic stock trading strategies.

Then we selected the model. We chose the DDPG algorithm, which is a reinforcement learning algorithm based on Policy

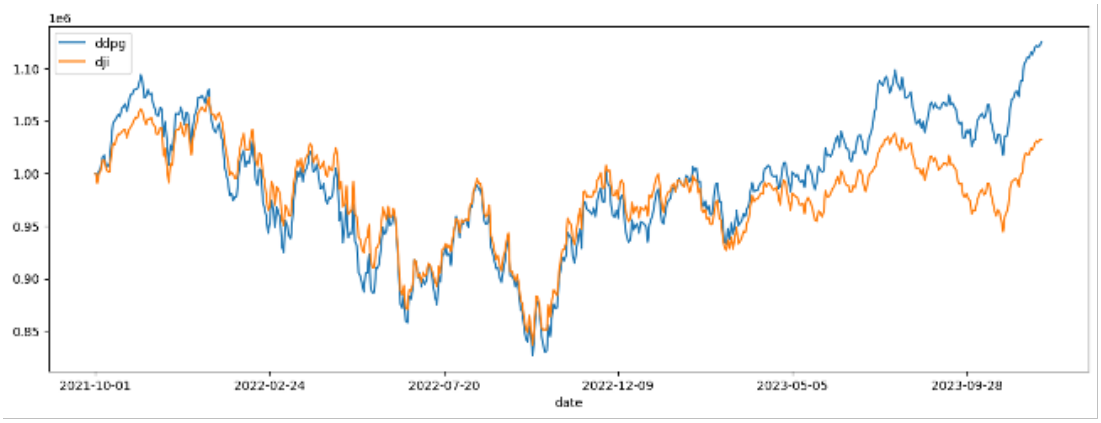

Fig. 13. Result Curve Of Experiment

time/episodes	time/time_elapsed	train/reward	train/learning_rate	train/n_updates	time/fps	time/total_timesteps	train/actor_loss	train/critic_loss
4	78	-8.59422	0.001	8871	151	11828	243.1889393	2516.914355
8	173	-8.59422	0.001	20699	136	23656	33.38629986	130.317245
12	268	-8.59422	0.001	32527	131	35484	-38.27470124	84.45886326
16	363	-8.59422	0.001	44355	129	47312	-51.10953447	70.71166328

TABLE III
TABLE OF TRAINING INFORMATION OF DDPG

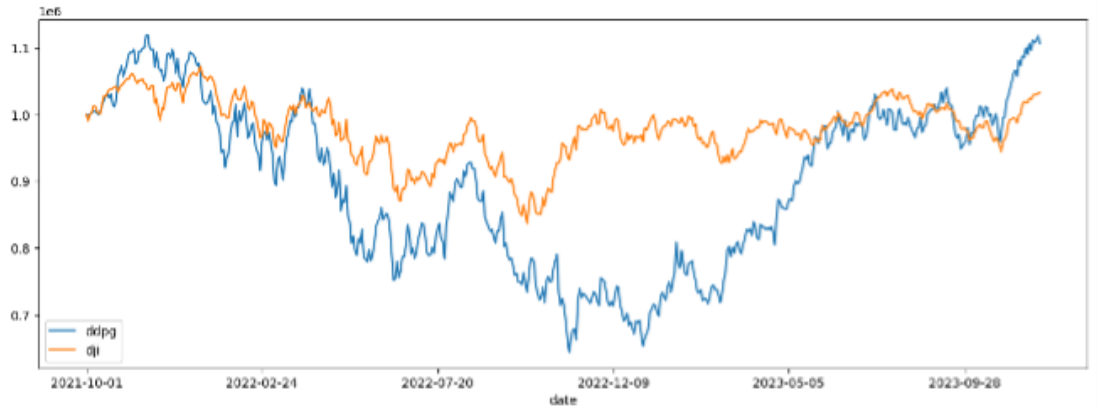

Fig. 14. Result Curve Of Experiment Using 5 Stocks

time/episodes	time/time_elapsed	train/reward	train/learning_rate	train/n_updates	time/fps	time/total_timesteps	train/actor_loss	train/critic_loss
4	78	-8.59422	0.001	8871	151	11828	243.1889393	2516.914355
8	173	-8.59422	0.001	20699	136	23656	33.38629986	130.317245
12	268	-8.59422	0.001	32527	131	35484	-38.27470124	84.45886326
16	363	-8.59422	0.001	44355	129	47312	-51.10953447	70.71166328

TABLE IV
TABLE OF TRAINING INFORMATION OF DDPG OF 5 STOCKS

Gradient. Compared with Value-based algorithms such as Q-learning, DDPG's advantage is that it makes decisions on continuous actions, so it is more suitable for the complex and changing stock environment.

After that, we carried out our experiments. We used Colab as the experimental environment, implemented our algorithm based on FinRL's financial reinforcement learning framework, used yfinance to obtain stock data and perform data preprocessing, then used Openai Gym to create a reinforcement learning environment, used Baseline3 to call the reinforcement learning algorithm, and then trained and backtested.

Our algorithm started with 10,000 US dollars and ended up with 11,830 US dollars. The annual return was 0.056101247, the cumulative returns were 0.12505537, the annual volatility was 0.186804106, and the Sharpe ratio was 0.386047138.

Compared to other enhanced DDPG algorithms, ours achieved better results in terms of profitability and stability.We have found some ways to improve our model, but we have not

finished implementing them yet due to time constraints.

Since some research already achieved some results, use Quantum Lever Prices(QPLs) from Quantum Finance Theory (QFT) to upgrade reinforcement learning model. In the future, after we learn more about the theory of quantum finance theory, we will also try to improve and enhance our model, and increase the capabilities of our stock trading AI agent.

REFERENCES

[1] R. Lin, Z. Xing, M. Ma and R. S. T. Lee, "Dynamic Portfolio Optimization via Augmented DDPG with Quantum Price Levels-Based Trading Strategy," 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 2023, pp. 1-8, doi: 10.1109/IJCNN54540.2023.10191785.

[2] R. S. Lee, Quantum finance. Springer, 2020.

[3] Liu, Y., Liu, Q., Zhao, H., Pan, Z., & Liu, C. (2020, April). Adaptive quantitative trading: An imitative deep reinforcement learning approach. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 02, pp. 2128-2135).

[4] Jiang, Z.; Xu, D.; and Liang, J. 2017. A deep reinforce- ment learning framework for the financial portfolio manage- ment problem. arXiv preprint arXiv:1706.10059.

[5] Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In ICML.

[6] L. Zha, L. Dai, T. Xu, and D. Wu, "A hierarchical reinforcement learning framework for stock selection and portfolio," in 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022, pp. 1–7.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013. [Online]. Available: https://arxiv.org/abs/1312.5602

[8] R. Wang, H. Wei, B. An, Z. Feng, and J. Yao, "Commission fee is not enough: A hierarchical reinforced framework for portfolio management," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 1, 2021, pp. 626–633.

[9] H.Yang,X.-Y.Liu,S.Zhong,andA.Walid,"Deepreinforcementlearn- ing for automated stock trading: An ensemble strategy," in Proceedings of the First ACM International Conference on AI in Finance, 2020, pp. 1–8.

[10] Amit Agarwal, Elad Hazan, Satyen Kale, and Robert E Schapire. Algorithms for portfolio management based on the newton method. In Proceedings of the 23rd international conference on Machine learning, pages 9–16. ACM, 2006.

[11] Puja Das and Arindam Banerjee. Meta optimization and its application to portfolio se- lection. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11, page 1163, 2011. doi: 10.1145/2020408.2020588. URL http://dl.acm.org/citation.cfm?doid=2020408.2020588.

[12] Parzen, E. (1961). An approach to time series analysis. The Annals of Mathematical Statistics, 32(4), 951-989.

[13] Nason, G. P., Sachs, R. V. (1999). Wavelets in time-series analysis. Philosophical transactions of the royal society of London. Series A: Mathematical, Physical and Engineering Sciences, 357(1760), 2511-2526.

[14] Kehoe, T. J., Pujolas, P. S., Rossbach, J. (2017). Quantitative trade models: Developments and challenges. Annual Review of Economics, 9, 295-325.

[15] Arthur, W. B. (1995). Complexity in economic and financial markets. Complexity, 1(1), 20-25.

[16] Yu, P., Lee, J. S., Kulyatin, I., Shi, Z., Dasgupta, S. (2019). Model-based deep reinforcement learning for dynamic portfolio optimization. arXiv preprint arXiv:1901.08740.

[17] Filos, A. (2019). Reinforcement learning for portfolio management. arXiv preprint arXiv:1909.09571.

[18] Yang, S. (2023). Deep reinforcement learning for portfolio management. Knowledge-Based Systems, 278, 110905

[19] Puder, A., Markwitz, S., Gudermann, F., Geihs, K. (1995). AI-based trading in open distributed environments. In Open Distributed Processing: Experiences with distributed environments. Proceedings of the third IFIP TC 6/WG 6.1 international conference on open distributed processing, 1994 (pp. 157-169). Springer US.

[20] Jaravel, X., Sager, E. (2019). What are the price effects of trade? Evidence from the US and implications for quantitative trade models.

[21] Jordan, M. I., Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

[22] Kaelbling, L. P., Littman, M. L., Moore, A. W. (1996). Reinforcement learning: A survey. Journal of artificial intelligence research, 4, 237-285.

[23] García, Salvador, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez, and Francisco Herrera. "Big data preprocessing: methods and prospects." Big Data Analytics 1, no. 1 (2016): 1-22.

[24] Mishra, Puneet, Alessandra Biancolillo, Jean Michel Roger, Federico Marini, and Douglas N. Rutledge. "New data preprocessing trends based on ensemble of multiple preprocessing techniques." TrAC Trends in Analytical Chemistry 132 (2020): 116045.

[25] Baresa, Suzana, Sinisa Bogdan, and Zoran Ivanovic. "Strategy of stock valuation by fundamental analysis." UTMS Journal of Economics 4, no. 1 (2013): 45-51.

[26] Liu, Ming, Qianqiu Liu, and Tongshu Ma. "The 52-week high momentum strategy in international stock markets." Journal of International Money and Finance 30, no. 1 (2011): 180-204.

[27] Silver, David, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. "Deterministic policy gradient algorithms." In International conference on machine learning, pp. 387-395. Pmlr, 2014.

[28] Sutton, Richard S., David McAllester, Satinder Singh, and Yishay Mansour. "Policy gradient methods for reinforcement learning with function approximation." Advances in neural information processing systems 12 (1999).

[29] Luo, Suyuan, Xudong Lin, and Zunxin Zheng. "A novel CNN-DDPG based AI-trader: Performance and roles in business operations." Transportation Research Part E: Logistics and Transportation Review 131 (2019): 68-79.

[30] Yang, Hongyang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. "Deep reinforcement learning for automated stock trading: An ensemble strategy." In Proceedings of the first ACM international conference on AI in finance, pp. 1-8. 2020.