IJICTR
International Journal of Information & Communication Technology Research

Volume 17- Number 3 – 2025 (58-69)

Adaptive Portfolio Optimization with Multi-Agent Deep Reinforcement Learning and Short-Term Performance Analysis

Shahin Sharbaf Movassaghpour
Department of Computer Engineering, Islamic Azad University, Tabriz Branch
Tabriz, Iran
sh.movassaghpour@iau.ac.ir

Masoud Kargar*
Department of Computer Engineering, Islamic Azad University, Tabriz Branch
Tabriz, Iran
kargar@iaut.ac.ir

Ali Bayani
Department of Computer Engineering, Islamic Azad University, Tabriz Branch
Tabriz, Iran
alibayani@iaut.ac.ir

Alireza Assadzadeh
Department of Computer Engineering
Islamic Azad University, Tabriz Branch
Tabriz, Iran
alireza.asadzadeh@iaut.ac.ir

Ali Khakzadi
Department of Computer Engineering
Islamic Azad University, Tabriz Branch
Tabriz, Iran
ali.khakzadi@iau.ir

Received: 02 October 2024 – Revised: 27 December 2024 - Accepted: 21 January 2024

Abstract—**This research presents a novel portfolio optimization framework using deep reinforcement learning (DRL). Traditional methods rely on static models or single-agent strategies, which struggle with market dynamics. We propose a dynamic system to address this by selecting the best-performing DRL agent based on recent market conditions. The framework evaluates five DRL agents, A2C, SAC, TD3, DDPG, and PPO, allocating portfolio weights based on short-term performance. A selection mechanism identifies the top agent using cumulative returns over the prior ten days, leveraging multiple agents' strengths. This adaptive approach embraces the philosophy that no single strategy consistently outperforms in all market conditions, making flexibility and continuous learning essential for robust portfolio management. Backtesting on Dow Jones data shows our method enhances cumulative returns and risk-adjusted performance, achieving an 11.43% average annual return, 38.29% cumulative returns, and a 0.832 Sharpe ratio, outperforming individual DRL agents.**

Keywords: Deep Reinforcement Learning, Consulting System, Portfolio Optimization, Tactical Decision, Cumulative Return

Article type: Research Article

Publisher: ICT Research Institute

* Corresponding Author

I. INTRODUCTION

In the dynamic and constantly changing financial markets, effective portfolio management remains essential for investors seeking to maximize returns while managing risks effectively. Conventional portfolio optimization methods often rely on static models, which may fall short of accommodating these markets' fluid and unpredictable nature [1]. To address this, sequence processing techniques such as linear vectors offer a straightforward and efficient way to handle financial datasets. These methods are particularly suited for portfolio optimization tasks requiring low computational complexity, providing a practical approach to analyzing time-dependent patterns in economic data, cryptocurrency, and portfolio signals [2]. By leveraging vectorization and sampling strategies, linear vectors facilitate extracting meaningful temporal patterns essential for informed decision-making [3].

As financial markets grow increasingly complex, electronic consulting systems are pivotal in transforming intricate financial data into actionable investment strategies. The rapid advancement of technology has made it imperative for investors to adopt sophisticated analytical tools to navigate modern market challenges [4]. Among these tools, deep reinforcement learning (DRL) has gained prominence for its ability to adapt and learn in dynamic environments. In financial applications, DRL enables adaptive decision-making by continuously analyzing historical trends and real-time market data, allowing investors to respond proactively to market fluctuations [5], [6]. In addition to finance, DRL has shown significant promise in areas like fog computing, optimizing task offloading between edge and cloud devices for latency-sensitive applications [7], and agricultural land use modeling, which helps simulate climate change adaptation strategies [8].

Financial markets are highly volatile, with prices influenced by economic events, investor sentiment, and global trends. Market analysis primarily relies on two approaches to navigate this complexity: technical analysis, which focuses on historical price patterns and indicators, and fundamental analysis, which incorporates broader economic factors, such as news sentiment and financial reports. This research primarily adopts a technical analysis perspective to evaluate the performance of the proposed multi-agent reinforcement learning framework compared to common DRL-based approaches (A2C [9], PPO [10], DDPG [11], TD3 [12], SAC [13]). By concentrating on technical features, we aim to assess how effectively the model adapts to market fluctuations and enhances portfolio returns. Policy-based DRL methods are particularly well-suited for financial markets because they handle non-stationarity and sequential dependencies in market data. Unlike value-based methods, which struggle with instability in dynamic environments, policy-based approaches optimize decision-making directly, ensuring more robust adaptation to market trends and improving long-term performance.

Despite advancements in financial modeling, portfolio optimization still faces significant challenges. Traditional techniques, such as the Max-Sharpe method, rely on fixed statistical assumptions, making them less effective in dynamic markets [14]. These static models struggle with rapid fluctuations, limiting their real-world applicability [15]. Similarly, many DRL strategies use a single-agent framework, which may underperform in varying market conditions since no single strategy consistently excels across all scenarios. Another key challenge is the lack of short-term adaptability in DRL models, as they often prioritize long-term rewards while failing to capture short-term market trends essential for active portfolio management. Without immediate responsiveness, these models risk missing opportunities or failing to mitigate short-term risks. Furthermore, the effectiveness of DRL models compared to traditional portfolio optimization methods, such as the Equal Weights Method and the Max-Sharpe Model (Markowitz Model), remains underexplored, making it difficult to fully assess their advantages in real-world investment scenarios [1], [2].

To address these challenges, we propose a multi-agent DRL framework that dynamically selects the best-performing agent based on recent market conditions. The key contributions of this work are as follows:

- A novel adaptive portfolio optimization framework that evaluates and selects from multiple DRL agents (A2C, SAC, TD3, DDPG, PPO) based on short-term performance.
- A short-term agent selection mechanism that dynamically chooses the top-performing model over a ten-day window, improving responsiveness to market fluctuations.
- Comprehensive benchmarking against traditional portfolio models, including the Equal Weights Method and the Max-Sharpe Method (Markowitz Model), demonstrates the superiority of our adaptive strategy.
- Extensive empirical evaluation using historical Dow Jones data shows that our approach enhances cumulative returns, risk-adjusted performance, and profitability.

The structure of this paper is as follows: Section II reviews related work on portfolio optimization and reinforcement learning. Section III outlines our methodology, including data preprocessing, model training, and the agent selection mechanism. Section IV discusses the experimental results and system performance evaluation. Section V offers an in-depth discussion of the findings. Finally, Section VI concludes the study and highlights potential directions for future research.

II. RELATED WORKS

Recent advancements in portfolio management have seen a surge in the application of innovative reinforcement learning (RL) architectures. Various studies have explored diverse RL-based methodologies to address dynamic financial markets, optimize portfolio allocation, and enhance risk-adjusted returns. These works highlight the integration of reinforcement learning with advanced techniques such as self-

attention mechanisms, sentiment analysis, multi-agent systems, and deep learning frameworks. A summary of works from 2020 to 2025, illustrating significant contributions in portfolio optimization, is presented in Table I. Each study is categorized based on the year, authors, methods, and key advantages.

TABLE I. SUMMARY OF RELATED WORKS IN PORTFOLIO MANAGEMENT.

Year	Authors	Method	Advantages
2021	Betancourt and Chen [16]	Method integrating new assets without retraining	High daily returns achieved
2021	Katongo and Bhattacharyya [17]	Deep RL algorithms with neural network autoencoders for tactical asset allocation	Improved risk-adjusted returns on Dow Jones
2021	Koratamaddi et al. [18]	RL approach incorporating market sentiment	Superior Sharpe ratio and annualized returns compared to baselines
2022	Lim, Cao, and Quek [19]	RL agent combined with LSTM for dynamic portfolio rebalancing	Significant return improvements
2022	Kabbani and Duman [20]	TD3 algorithm for automating stock trading	High Sharpe ratio achieved
2023	Zhao et al. [21]	Deterministic policy gradient RL method with self-attention mechanism	Excellent performance across financial datasets
2023	Li et al. [22]	LSRE-CAAN framework for high-frequency cryptocurrency data	High returns with lower risk metrics
2023	Ma et al. [23]	Multi-agent system with trend consistency regularization for market status switching	Effective results in Chinese Stock Market
2023	Hao et al. [24]	Fuzzy vectors with ensemble RL methods applied to SP100 stocks	Outperformed benchmarks over 11 years
2020	Zhang et al. [25]	Cost-sensitive portfolio selection with a two-stream portfolio policy network	Effective management of transaction and risk costs
2020	Soleymani and Paquet [26]	DeepBreath framework combining autoencoder and CNN with blockchain	Solved settlement issues and enhanced policy enforcement
2020	Wu et al. [27]	GRU-based adaptive stock trading strategies	Outperformed traditional methods
2021	Wu et al. [28]	CNN and RNN models with Sharpe ratio-based reward function	Improved returns and reduced drawdown risks
2021	Carta et al. [29]	Multiple deep neural networks with a reward-based classifier	Superior performance on S&P 500
2022	Lin et al. [30]	Multi-agent RL framework with nested agent structure	Outperformed traditional portfolio selection strategies
2023	Jang and Seong [31]	Modern portfolio theory with RL and Tucker decomposition for multimodal inputs	Outperformed state-of-the-art algorithms
2023	Day et al. [32]	Dynamic trading strategy model incorporating technical indicators and covariance	High annualized and cumulative returns on ESG Select Index ETF
2023	Yu et al. [33]	Nested RL method with weight random selection strategy	Enhanced investor profits across various markets
2024	Li and Hai [34]	Multi-agent deep RL model with additional stock indices and self-attention networks	Improved portfolio management and risk mitigation
2024	Jiang et al. [35]	Model-free DRL framework for portfolio selection	Superior performance, handles high-dimensional data, includes transaction costs
2024	Chen et al. [36]	RL with mean-variance model for dynamic risk preferences	Outperforms buy-and-hold, adjusts risk in volatile markets
2025	Aritonang et al. [37]	Evaluates hidden-layer configurations in RL models	Optimizes portfolios, identifies ideal complexities for algorithms
2025	Song et al. [38]	RL with deterministic state transitions	Enhances feature extraction, excels in cryptocurrency backtesting
2025	Sattar et al. [39]	RMS-Driven DRL for optimized portfolio management	Integrates sentiment analysis, improves risk-return in dynamic markets

III. METHODS

In this research, two algorithms have been independently designed to ensure optimal decision-making for maximizing cumulative returns in stock price prediction. Five RL algorithms, A2C, PPO, DDPG, TD3, and SAC, have been utilized. Each algorithm has strengths and limitations, demonstrating varied performance under different market conditions.

The A2C and PPO agents, leveraging stochastic policies and policy optimization, quickly adapt to market changes and perform better in volatile and turbulent environments. On the other hand, DDPG and TD3 agents, employing neural networks in continuous spaces, excel in gradually shifting markets and are more suitable for long-term and stable scenarios. The SAC agent seeks more diverse policies, enabling it to make robust decisions under uncertainty. By maximizing entropy, SAC proves to be particularly effective in uncertain conditions.

As noted, due to the highly dynamic nature of the market and the structural differences between the agents, it is unclear before trading which agent will perform best for predicting prices at time t+1. Thus, selecting a single agent at the time "t" is challenging, and an incorrect choice can reduce cumulative returns. To address this issue, two algorithms are proposed in this paper:

1. Averaging Approach (Proposed Method 1)

In the first algorithm, the output of all agents is averaged. Specifically, each agent suggests a weight for each stock at time t+1, and the average of these weights is considered the final weight for t+1. The proposed weight by each agent indicates the proportion of that stock in the portfolio. If the final weight equals the weight at time "t," it implies that the stock should remain unchanged in the portfolio (i.e., a "hold" action). If the final weight increases, it indicates a purchase is needed, with the purchase amount calculated based on the weight increase. Conversely, if the final weight decreases, it suggests selling a portion of the stock, with the amount determined by the weight reduction. The total weights of all stocks always sum to 1. Averaging the outputs of multiple agents reduces the error caused by the poor performance of a single agent and leverages the predictive capabilities of each algorithm on complex data.

2. Main Proposed Method

The second proposed algorithm selects a single agent as the decision-maker for predicting stock weights at time t+1. This selection is based on market conditions at that moment. The process involves monitoring agents' historical performance over the past "n" days. In other words, the agent that achieved the highest cumulative return over the last "n" days is chosen to decide for time t+1. Selecting the top-performing agent over the past "n" days ensures that decisions align with current market conditions. This approach is efficient for dynamically adapting to market changes. For instance, if the market transitions to a volatile or stable phase, a new agent suited to these conditions will be quickly selected. Additionally, this method excludes underperforming agents from influencing the final decision, preventing their negative impact. The main proposed method will be further detailed, as illustrated in Figure 1.

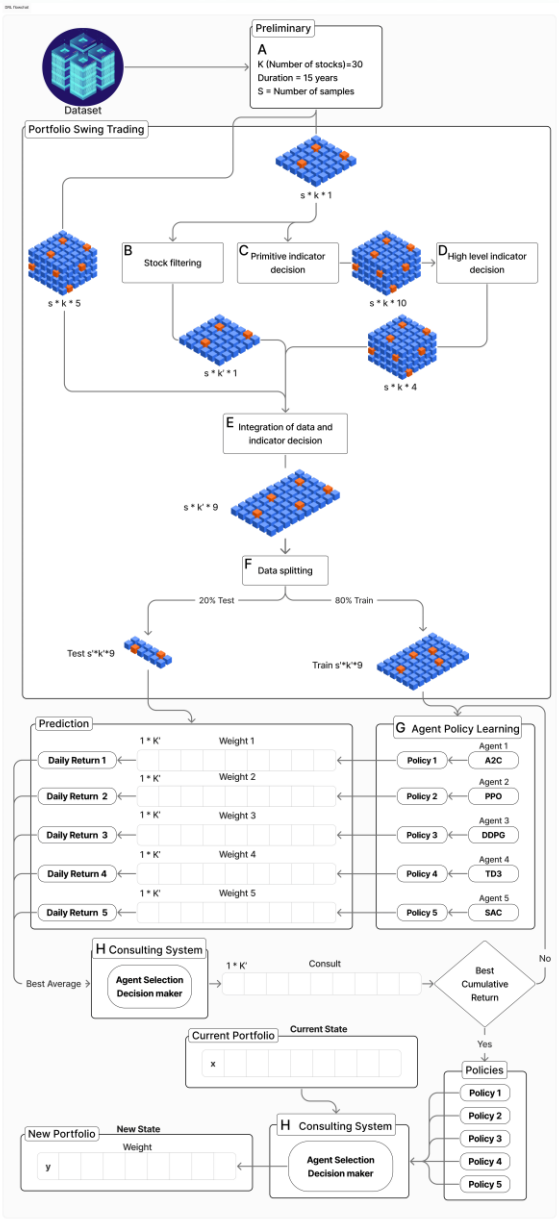

Figure 1. General Schema of the Proposed Consulting System.

A. Stock Data Preparation

The stock data is initially represented in dimensions S×K×5, where S represents the number of samples, each covering a moment t. K is the number of stocks, and 5 illustrates various features of each stock, including the opening price, closing price, highest price, lowest price, and trading volume.

B. Stock Filtering

At this stage, the closing prices of stocks, represented as S×K×1, are compressed using an Autoencoder. During this step, K′ stocks with the lowest volatility are selected. After applying this compression, the output dimensions are reduced to S×K′×1, where K′ is the number of selected stocks with the lowest volatility.

C. Primitive Indicator Decision

At this stage, technical indicators are calculated for each stock to analyze stock prices better. These indicators include technical tools that help analysts assess market trends and the current state of stocks. The indicators are Average True Range (ATR), Bollinger Bands Width (BBW), On-Balance Volume (OBV), Chaikin Money Flow (CMF), Moving Average Convergence Divergence (MACD), Average Directional Index (ADX), Relative Strength Index (RSI), Commodity Channel Index (CCI), Exponential Moving Average (EMA), and Simple Moving Average (SMA). After calculating these indicators, the data dimensions are adjusted to S×K×10, where S is the number of samples, K is the number of stocks, and 10 represents the number of calculated indicators. This stage is an effective tool for a more precise analysis of price trends and identifying market signals.

D. High-Level Indicator Decision Phase

After extracting the technical indicators, another Autoencoder compresses them to reduce data dimensions and increase model efficiency. Dimension reduction allows the model to focus only on the main and more relevant features instead of processing many complex ones, thereby improving the accuracy and efficiency of agent learning. After this compression, the number of indicators is reduced from 10 to 4, and the data dimensions become S×K×4.

E. Integration of Data and Indicator Decision

The processed and compressed technical indicators and data from each section are integrated at this stage. This process, called "Row Data and Indicators Data Concatenation," combines various data, including stock prices, compressed technical indicators, and selected stocks, into a single dataset. This integration prepares the data for use in reinforcement learning models, enabling the model to utilize all features cohesively for more accurate predictions. After this integration, the output dimensions change to S×K′×9, where S is the number of samples, K′ is the number of selected stocks, and 9 represents the number of different features, including stock prices and indicators for the stocks.

F. Data Splitting

Next, the data is divided into two parts: train and test. 80% of the data is used for training, and the remaining 20% is used for testing. This division is not done randomly because the data is time-series and has temporal dependencies. The data must be split sequentially and chronologically to preserve these dependencies and accurately simulate market trends.

G. Agent Policy Learning

DDPG, A2C, PPO, SAC, and TD3 are all Actor-Critic-based algorithms that leverage state-action pairs to learn optimal policies and Q-values. While DDPG and TD3 are explicitly designed for continuous action spaces, TD3 offers excellent stability by reducing overestimation in Q-value calculations. SAC, also tailored for constant domains, promotes exploration through entropy maximization. On the other hand, A2C and PPO are versatile and can be applied in discrete and continuous spaces. A2C enhances learning through Advantage estimation, whereas PPO ensures excellent stability by constraining policy updates.

Regarding On-Policy versus Off-Policy approaches, A2C and PPO are On-Policy, relying exclusively on data from the current policy. This ensures higher stability but results in lower data

efficiency. Conversely, DDPG, TD3, and SAC are Off-Policy algorithms that utilize Replay Buffers, enabling them to leverage past and recent data. This makes learning in complex environments more efficient, though implementation complexity increases.

Trajectories play a crucial role in these algorithms. Trajectories consist of states, actions, rewards, and following states generated by the agent's interaction with the environment. In On-Policy algorithms like A2C and PPO, trajectories are collected directly from the agent's current policy, ensuring stable learning but requiring frequent environmental sampling. However, off-policy algorithms such as DDPG, TD3, and SAC can store trajectories from previous policies in a Replay Buffer, enhancing data efficiency and facilitating learning in intricate environments.

Additionally, these algorithms adopt various Backup Strategies for value updates. A2C and PPO utilize Monte Carlo methods to accumulate total rewards for a trajectory, whereas DDPG, TD3, and SAC employ Bootstrapping, which estimates next-state values to speed up and optimize learning. SAC combines both methods to strike a balance between stability and accuracy.

Training data is employed to teach agents within the reinforcement learning model at this stage. This data acts as the environment where agents interact. The primary objective is for the agents to learn optimal decision-making strategies to enhance portfolio performance. Key reinforcement learning concepts, State, Action, Reward Function, and Environment, are defined as follows:

1) State

The state of the environment represents the information available to agents at any moment for decision-making. In this problem, the state comprises features related to stock prices, technical indicators, and processed data.

2) Action

The action represents the decision made by the agent at each time step. Here, it refers to assigning weights to each stock in the investment portfolio, where these weights determine the percentage of capital allocated to each stock. The action vector includes weight values for all stocks in the portfolio.

3) Reward

When an agent takes an action, the environment provides a reward indicating the agent's performance. In this case, the reward corresponds to the profit or loss from changes in portfolio weights. Higher profits yield positive rewards, while losses result in negative rewards. This function motivates agents to adopt strategies that maximize cumulative returns in the market.

4) Environment

The training data serves as a simulated environment for agent learning. Agents can evaluate their actions, observe performance results, and measure their impact on the portfolio. The environment reflects real-world stock market conditions, including prices, volatility, and other features.

The goal of training agents is to identify an optimal policy that enables the agent to select the best possible action for each state to maximize cumulative returns. During training, agents observe the market state, perform actions, and improve their policies based on received feedback (rewards). This process iteratively continues until agents develop strategies capable of achieving optimal performance in dynamic and complex market conditions.

H. Consulting System

After training agents and obtaining their respective optimal policies, a Consulting System is used for decision-making at time t+1. This system selects the best-performing agent based on cumulative returns over the past "n" days. The selection criterion is the cumulative income each agent's policy generated during the previous period. The agent with the highest cumulative income is chosen for t+1. The selected agent's proposed weights, representing the percentage of investment in each stock, are used as the final decision. These decisions determine which stocks should be bought, sold, or held.

The selection and decision-making process is applied iteratively on test data to calculate the system's overall cumulative income. The Consulting System aims to outperform the best individual agent by achieving a final cumulative income that surpasses the maximum cumulative income of any single agent. Once this process is complete and satisfactory performance is ensured, the Consulting System is deployed in real-world market conditions. Utilizing trained agents, the system provides recommendations for stock purchases, sales, or holdings, ultimately optimizing investments and maximizing cumulative income.

I. Hyperparameter Configuration

The selection and adjustment of hyperparameters are critical to steering the learning mechanisms of reinforcement learning agents, particularly in the intricate domain of portfolio optimization. By fine-tuning these parameters, the agents can better enhance their decision-making processes. Tables II and III present the tailored hyperparameter setup for the proposed method.

TABLE II. HYPERPARAMETERS USED FOR EACH DRL AGENT.

Hyperparameter	A2C	PPO	DDPG	TD3	SAC
Learning Rate	0.001	0.001	0.001	0.001	0.001
Batch Size	N/A	100	100	100	100
Buffer Size	N/A	N/A	50,000	50,000	50,000
Entropy Coefficient	0.005	0.005	N/A	N/A	auto_0.1
Learning Starts	N/A	N/A	N/A	N/A	100
n_steps	5	5	N/A	N/A	N/A
Total Timesteps	10,000	10,000	10,000	10,000	10,000

TABLE III. STOCK TRADING ENVIRONMENT PARAMETERS.

Parameter	Value
hmax	500
initial_amount	1,000,000
transaction_cost_pct	0.001
stock_dim	20
tech_indicator_dim	4

TABLE IV. PERFORMANCE METRICS AND THEIR MATHEMATICAL FORMULATIONS UTILIZED IN THE EVALUATION OF THE PROPOSED PORTFOLIO OPTIMIZATION MODEL.

Metric	Formula	Description
Annual Return	$Annual\ Return = \left(\frac{End\ Value}{Start\ Value}\right)^{\frac{1}{n}} - 1$ (1)	Measures the return achieved over one year.
Cumulative Returns	$Cumulative\ Returns = \frac{End\ Value - Start\ Value}{Start\ Value}$ (2)	Reflects the total return over the evaluation period.
Annual Volatility	$Annual\ Volatility = \sigma \times \sqrt{252}$ (3)	Indicates the standard deviation of returns over a year.
Average Daily Returns	$Average\ Daily\ Returns = \frac{1}{n}\sum_{i=1}^{n} R_i$ (4)	The mean return achieved on a daily basis.
Sharpe Ratio	$Sharpe\ Ratio = \frac{E[R_p] - R_f}{\sigma_p}$ (5)	Evaluates risk-adjusted return relative to volatility.
Calmar Ratio	$Calmar\ Ratio = \frac{Annual\ Return}{Max\ drawdown}$ (6)	Compares annual return to the maximum observed drawdown.
Max Drawdown	$Max\ Drawdown = min_t\left(\frac{Trough_t - Peak_t}{Peak_t}\right)$ (7)	Captures the largest drop in value from a peak to a trough.
Omega Ratio	$Omega\ Ratio = \frac{\int_{R_f}^{\infty}[1-F(x)]dx}{\int_{-\infty}^{R_f}F(x)dx}$ (8)	Measures gains relative to losses using the cumulative distribution function.
Sortino Ratio	$Sortino\ Ratio = \frac{E[R_p] - R_f}{\sigma_d}$ (9)	Focuses on downside risk by considering negative returns only.
Skew	$Skew = \frac{E\left[(R_p - \mu)^3\right]}{\sigma^3}$ (10)	Quantifies the asymmetry of the return distribution.
Kurtosis	$Kurtosis = \frac{E\left[(R_p - \mu)^4\right]}{\sigma^4}$ (11)	Indicates the "tailedness" or extreme values in the return distribution.
Tail Ratio	$Tail\ Ratio = \frac{Average\ of\ Top\ 5\%\ Returns}{Average\ of\ Bottom\ 5\%\ Returns}$ (12)	Compares the best 5% of returns to the worst 5%.
Daily Value at Risk	$VaR_\alpha = -inf\{x \mid F(x) \geq \alpha\}$ (13)	Estimates the maximum potential loss over 24 hours at a specific confidence level.
Alpha	$\alpha = R_p - \left[R_f + \beta\left(R_m - R_f\right)\right]$ (14)	Represents the excess return relative to a benchmark market index.
Beta	$\beta = \frac{Cov(R_p, R_m)}{Var(R_m)}$ (15)	Measures portfolio volatility relative to the market.

J. Aggregation and Evaluation

The aggregation and evaluation processes are fundamental components of our proposed method, ensuring that the combined reinforcement learning agents perform effectively in portfolio asset allocation. This section outlines the key steps involved in integrating agent outputs and assessing their overall performance.

1) Performance Metrics

We employed a range of performance metrics to comprehensively assess the proposed method. These metrics are summarized in Table IV. Each metric provides unique insights into the model's ability to manage risk and optimize returns.

2) Comparative Analysis

The combined agent strategy is benchmarked against individual agents and traditional portfolio allocation methods. Integrating predictions from multiple agents provides a significant advantage, showcasing their collective value in improving outcomes.

K. Time Complexity Analysis

Understanding the computational efficiency of RL algorithms is crucial for evaluating their practical applicability. Table V presents the overall time complexity and the best-case and worst-case scenarios for the DRL methods used in this research. The complexity of each algorithm is influenced by factors such as the total timesteps (T), state space dimension (S), action space dimension (A), neural network layers (L), and number of neurons per layer (N). Among these, PPO exhibits the highest computational complexity due to additional iterative updates per epoch (I), leading to a worst-case complexity of $O(n^4)$. The proposed method executes five agents serially in a loop and has a worst-case complexity of $O(n^4)$, similar to PPO. However, when parallel processing is feasible, the proposed method's complexity improves to $O(n^3)$ in the worst case and $O(n^2)$ in the best case, making it more computationally efficient under optimized conditions.

TABLE V. TIME COMPLEXITY OF THE PROPOSED METHOD.

Algorithm	Overall Time Complexity	Best Case	Worst Case
A2C	$O(T \times (S + A) \times L \times N^2)$	$O(n^2)$	$O(n^3)$
PPO	$O(I \times T \times (LN^2 + BD))$	$O(n^3)$	$O(n^4)$
DDPG	$O(T \times (L \cdot n \cdot D + B \cdot (L \cdot n^2)))$	$O(n^2)$	$O(n^3)$
TD3	$O(T \times (L \cdot n \cdot D + B \cdot (L \cdot n^2)))$	$O(n^2)$	$O(n^3)$
SAC	$O(T \times (L \cdot n \cdot D + B \cdot (L \cdot n^2)))$	$O(n^2)$	$O(n^3)$
Proposed Method	Five agents executed serially in a loop multiple times	$O(n^3)$	$O(n^4)$

IV. RESULTS

This section presents the outcomes of our experiments, detailing the dataset characteristics and analyzing the performance of individual RL agents alongside a comparative assessment of various strategies.

A. Dataset Overview

The dataset used in this research was acquired via the FinRL library [40] using the Yahoo Finance API, which provides free access to data on the Dow Jones Industrial Average (DJIA). The DJIA encompasses 30 leading companies listed on U.S. stock exchanges. From December 31, 2008, to March 30, 2024, the dataset contains 4035 daily records for each stock. These records include key attributes such as opening, closing, highest and lowest prices, and trading volume.

B. Individual Agent Performance

Table VI summarizes the performance metrics for each RL agent and the proposed method. Figure 2 highlights the backtest results on training data, where all agents achieved cumulative income exceeding seven times the initial value. While these figures are unrealistic due to the overlap between training and testing datasets, they provide a valuable benchmark for comparing agent performance. Among the agents, TD3 demonstrated relatively superior performance in this phase.

TABLE VI. PERFORMANCE METRICS FOR DIFFERENT REINFORCEMENT LEARNING AGENTS AND PROPOSED METHODS ON THE DOW JONES INDUSTRIAL AVERAGE DATASET (2009–2024).

Evaluation Metrics	A2C Model	PPO Model	DDPG Model	TD3 Model	SAC Model	Proposed Method 1	Main Proposed Method
Annual return	0.10355	0.088963	0.109103	0.093883	0.104561	0.100179	**0.114276**
Cumulative returns	0.343403	0.2909	0.363759	0.308451	0.347093	0.331145	**0.382903**
Annual volatility	0.139666	0.139456	0.142429	0.147605	0.13928	0.140483	**0.142167**
Sharpe ratio	0.775404	0.680907	0.798369	0.681806	0.783763	0.749933	**0.832301**
Calmar ratio	0.580168	0.578662	0.610732	0.451763	0.59372	0.57494	**0.693455**
Stability	0.439511	0.521288	0.522191	0.253352	0.505337	0.459214	**0.639741**
Max drawdown	-0.17848	-0.15374	-0.17864	-0.20781	-0.17611	-0.17424	**-0.16479**
Omega ratio	1.141502	1.122649	1.144586	1.122642	1.141947	1.135974	**1.153198**
Sortino ratio	1.117814	0.979952	1.146267	0.975362	1.122939	1.076073	**1.20474**
Skew	-0.14533	-0.12615	-0.17648	-0.17439	-0.1918	-0.16957	**-0.13656**
Kurtosis	1.844322	1.724517	1.561056	1.654512	1.784828	1.747927	**1.798994**
Tail ratio	1.048237	1.024539	1.028713	1.077653	0.969984	0.999069	**1.005422**
Daily Value at risk	-0.01717	-0.01719	-0.01749	-0.0182	-0.01711	-0.01728	**-0.01744**
Alpha	0	0	0	0	0	0	**0**
Beta	1	1	1	1	1	1	**1**

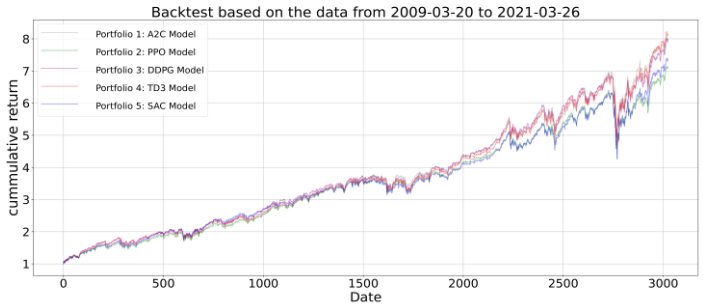

Figure 2. Backtest on Train Data and Cumulative Return by RL Agents.

Moreover, Figure 3 examines the backtesting results on unseen data from 2021 to 2024, comparing five RL agents, the proposed algorithm, and a strategy averaging the weights of all agents. Although TD3 excelled during training, DDPG performed better during testing. The weight-averaging algorithm exhibited average performance, whereas the proposed algorithm, which dynamically selects the best-performing agent based on the previous 10 days, outperformed all methods. It achieved a cumulative profit of 38.29%, the highest among all strategies.

Figures 4 through 10 collectively highlight the effectiveness and robustness of the proposed algorithm in dynamic portfolio management. Figure 4 illustrates daily portfolio weight adjustments, detailing buy, hold, and sell actions, while Figure 5 provides a weekly aggregation of these actions. Figure 6 demonstrates the percentage usage of agents throughout each working week, showcasing the dynamic adaptability of the proposed approach. Table VII further details the usage percentages of each agent over the 765-day backtest period, showing that the proposed method utilized the DDPG algorithm, the top-performing agent, 21.48% of the time, and the PPO algorithm, the least effective, 12.21% of the time. Figure 7 showcases the percentage of stock selections by the agents on testing data, emphasizing the method's adaptability in leveraging diverse strategies.

TABLE VII. USAGE PERCENTAGES OF EACH ALGORITHM.

Algorithm	A2C	PPO	DDPG	TD3	SAC
Usage Percent	19.33%	12.21%	21.48%	30.47%	16.51%

Figure 8 demonstrates the algorithm's capability to deliver consistent returns through its monthly and annual profit percentages on testing data. Finally, Figure 9 compares the cumulative returns of the proposed algorithm with the Dow Jones index, clearly underscoring the superiority of the proposed method in achieving higher profits and effectively managing risk over the testing period.

C. Performance Evaluation Against Traditional Portfolio Models

1) Equal Weights Method

In this approach, all assets in the portfolio are assigned equal weights. It is one of the most straightforward asset allocation strategies, as it does not involve any optimization based on asset characteristics. The main advantage of this method lies in its simplicity and the fact that it does not require historical data to estimate asset returns and risks. However, this approach may not yield optimal performance, as assigning equal weights to all assets can result in an inappropriate risk profile portfolio.

2) Max-Sharpe Method (Markowitz Model)

The second approach is based on the Markowitz model and aims to maximize the Sharpe ratio defined in Equation 5 (Table IV). Where $E[R_p]$ represents the expected return of the portfolio, R_f is the risk-free return rate, and σ_p denotes the standard deviation of the portfolio return, which serves as a measure of risk.

In this method, the optimal portfolio weights are determined using training data. These weights are then applied to test data to evaluate actual performance. The optimization process involves calculating the mean return and the covariance matrix of assets based on the training dataset. Once the optimal weights are determined, they are applied to the test dataset, and cumulative returns are computed.

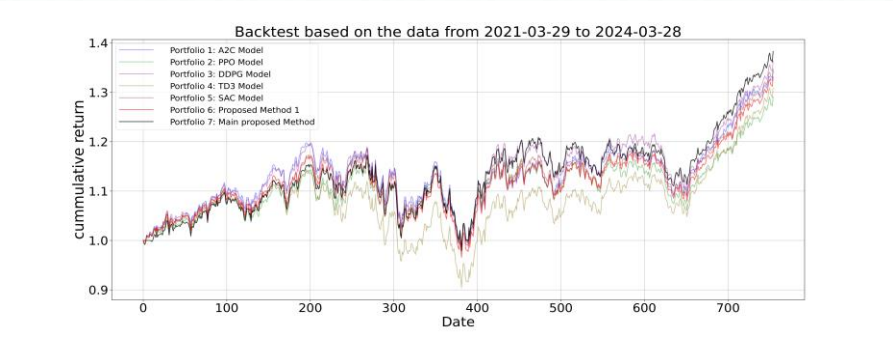

Figure 3. Backtest on Test Data and Cumulative Return by RL Agents and Proposed Methods.

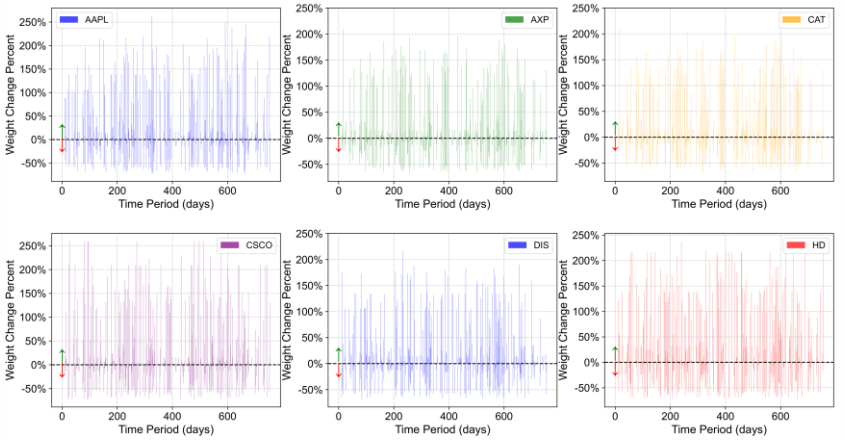

Figure 4. Daily Portfolio Weight Changes: 6 Example Stocks.

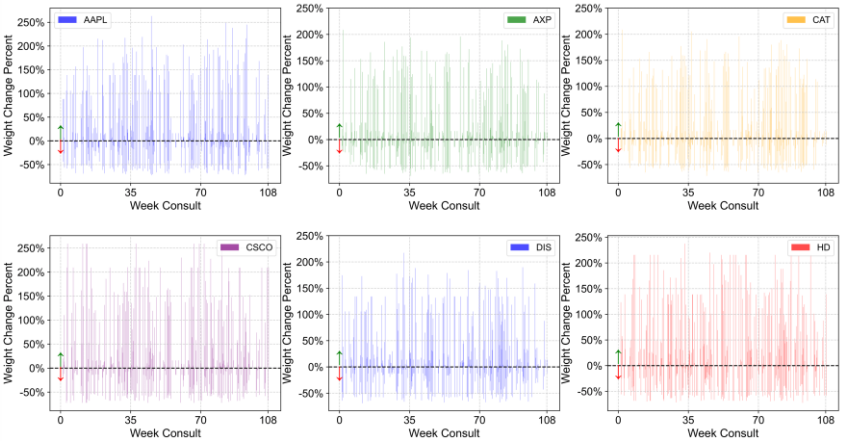

Figure 5. Weekly Portfolio Weight Changes: 6 Example Stocks.

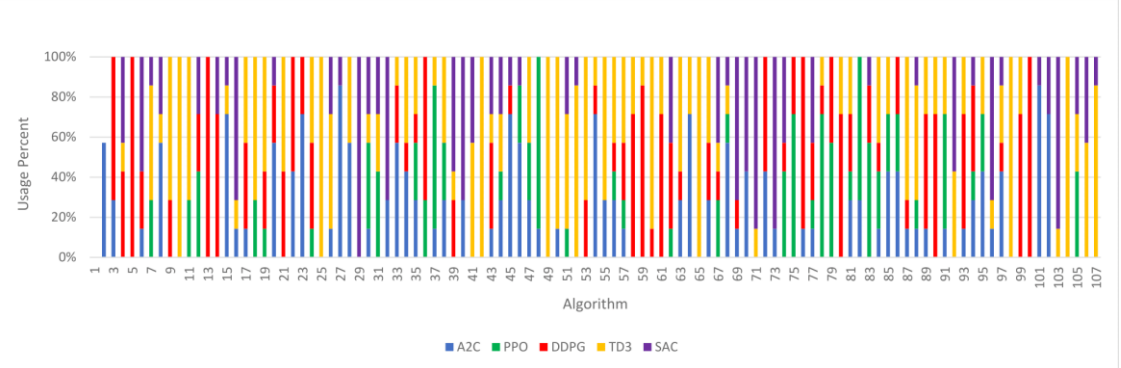

Figure 6. The Percentage Usage of Agents Throughout Each Working Week.

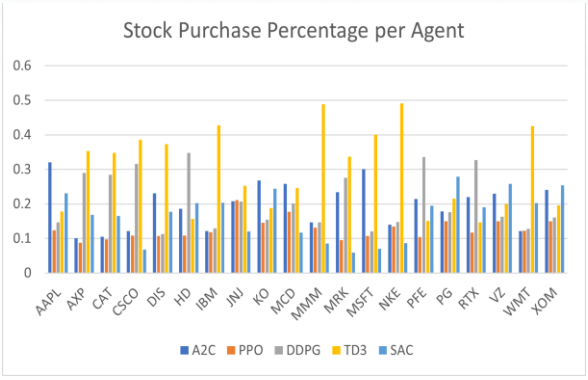

Figure 7. Stock Purchase Percentage per Agent.

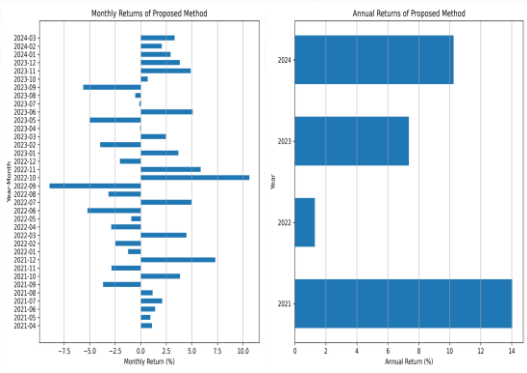

Figure 8. Monthly and Annual Returns of Proposed Method.

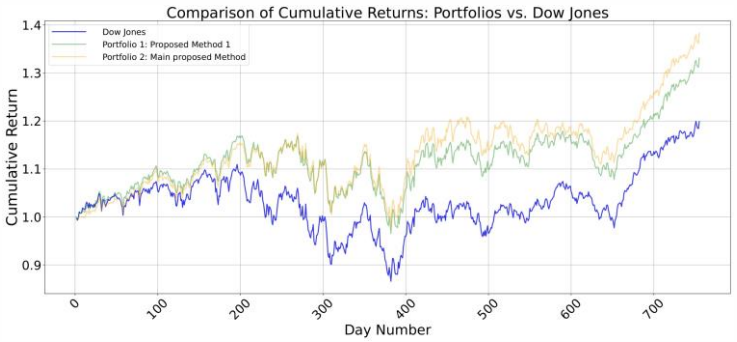

Figure 9. Comparison of Cumulative Returns of Proposed Methods and Dow Jones.

Figure 10 illustrates the asset weights assigned in both methods. As observed, the Equal Weights method distributes identical weights to all assets. In contrast, the. In contrast, the Max-Sharpe method assigns varying weights based on training data, adjusting allocations according to the optimization process.

Table VIII presents the cumulative returns of the proposed method compared to the Equal Weights and Max-Sharpe strategies on test data from March 29, 2021, to March 28, 2024. The results indicate that the proposed method significantly outperforms both alternative approaches, achieving a cumulative return of 38.2%, compared to 28.2% for the Max-Sharpe model and 27.3% for the Equal Weights strategy.

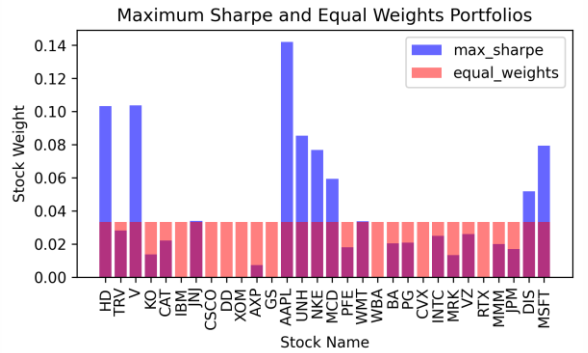

Figure 10. Asset Weights Allocation in Equal Weights and Max-Sharpe Methods.

TABLE VIII. CUMULATIVE RETURNS OF DIFFERENT PORTFOLIO STRATEGIES ON TEST DATA.

Method	Cumulative Return
Equal Weights	27.3%
Max-Sharpe	28.2%
Proposed Method	38.2%

V. DISCUSSION

A. *Effectiveness of a Dynamic Multi-Agent DRL Approach*

This paper highlights the substantial potential of DRL in investment portfolio optimization. By employing a framework that dynamically identifies and selects the most effective DRL agent over short-term intervals, the system can adapt swiftly to fluctuating market conditions. This adaptability capitalizes on the unique strengths of various DRL models in real-time, which collectively contribute to improved portfolio outcomes.

The choice of policy-based DRL methods in this framework is motivated by their ability to address key financial market challenges, particularly non-stationarity and sequential dependencies in market data. Unlike value-based approaches, which struggle with instability due to violating the independent and identically distributed (I.I.D.) assumption, policy-based methods optimize decision-making directly, ensuring smoother learning. This characteristic is particularly crucial in dynamic and high-volatility markets, where abrupt changes in asset prices require continuous adaptation. By leveraging policy optimization techniques, these methods enable more stable and responsive trading strategies, improving overall portfolio resilience.

This research enhances adaptability to shifting market conditions by integrating multiple policy-based DRL agents in a dynamic selection framework. Switching between policy-based agents allows for a more seamless transition between market regimes, improving long-term stability and returns.

Notably, the multi-agent DRL strategy employed herein offers greater resilience and robustness than relying on a single agent, as evidenced by enhanced risk-adjusted performance metrics. Key indicators, including higher Sharpe and Sortino ratios, underscore the practical viability of this method, which achieves a favorable balance between return generation and risk mitigation. This makes the approach appealing for investors pursuing growth and stability in volatile markets.

This research utilized five DRL agents, A2C, PPO, DDPG, TD3, and SAC, to optimize portfolio allocations. Each algorithm excels under specific financial conditions due to its distinctive attributes. Table IX summarizes these advantages, demonstrating how agents leverage their unique strengths to perform effectively across diverse market environments. After training and backtesting using historical data, the results shown in Table X indicate the cumulative returns achieved by each agent. The challenge lies in accurately selecting the top-performing agent daily to maximize overall portfolio returns.

TABLE IX. ADVANTAGES OF REINFORCEMENT LEARNING ALGORITHMS FOR PORTFOLIO OPTIMIZATION.

Advantages	Algorithms
Stable, cost-effective, faster, and works better with large batch sizes	A2C
Improve stability, less variance, simple to implement	PPO
Better at handling high-dimensional continuous action spaces	DDPG
Improve network stability in complex environments	TD3
Improve stability	SAC

TABLE X. CUMULATIVE RETURNS BY MODEL (MARCH 29, 2021 – MARCH 28, 2024).

Model	Cumulative Return (%)
PPO Model	29.0%
TD3 Model	30.8%
A2C Model	34.3%
SAC Model	34.7%
DDPG Model	36.3%

One solution implemented was a weighted average approach, whereby all agents distributed the portfolio's allocation daily. This method resulted in cumulative returns between the best-performing agent (DDPG, 36.3%) and the lowest-performing agent (PPO, 29%), yielding an intermediate result (Proposed Method 1, 33.1%). An alternative approach involved selecting the agent with the best performance over the preceding 10 days for each working day. As Table XI illustrates, this Main Proposed Method significantly improved cumulative returns, reaching 38.2%, surpassing both the weighted average approach and the best individual agent.

TABLE XI. CUMULATIVE RETURNS FOR PROPOSED METHOD 1 AND MAIN PROPOSED METHOD (MARCH 29, 2021 – MARCH 28, 2024).

Model	Cumulative Return (%)
PPO Model	29.0%
TD3 Model	30.8%
Proposed Method 1	**33.1%**
A2C Model	34.3%
SAC Model	34.7%
DDPG Model	36.3%
Main Proposed Method	**38.2%**

The Proposed Method 1 offers a balanced decision-making approach using the average weight of all factors. This method performs reasonably well with a cumulative return of 0.33, a Sharpe ratio of 0.74, and a Sortino ratio of 1.07, indicating a relative balance between returns and risk. Additionally, with a maximum drawdown of -0.17, it has managed to control potential losses. However, due to the fixed weighting, its flexibility in adapting to market changes is limited. In this method, the Calmar ratio of 0.57 and the Tail ratio of 0.99 suggest that compared to other models, its performance is weaker when facing long-term and low-volume risks, as seen in Table VI.

By adjusting the weights of factors based on past performance in each time step, the Main Proposed Method shows greater adaptability to market conditions. This method increases cumulative returns to 0.38 and exhibits a better risk-adjusted performance with a Sharpe ratio of 0.83. Although some metrics, such as the annual volatility of 0.14, the Calmar ratio of 0.69, and the Omega ratio of 1.13, are not significantly different from other models, its ability to reduce maximum drawdown to -0.16 and improve stability to 0.63 demonstrates a more effective use of market opportunities. Furthermore, the Tail ratio of 1.00 and the Sortino ratio of 1.20 show significant improvements in reducing downside risks and increasing stability. Additionally, the kurtosis of 1.79 and skew of -0.13 have adapted more favorably to market fluctuations and managed potential risks more effectively, as seen in Table VI.

The system's efficacy, demonstrated through backtesting with Dow Jones data, indicates its potential for broader application across various markets and asset classes. However, additional research is required to evaluate its performance under differing economic conditions and ensure scalability for real-world implementation. Future studies should consider incorporating more advanced DRL agents and exploring refined decision-making frameworks to enhance portfolio optimization. These findings underscore the importance of dynamic, adaptive strategies in modern portfolio management, showcasing the benefits of leveraging multiple DRL agents for superior investment performance.

B. Optimal Time Window for Market Adaptation

The A2C algorithm, due to its Actor-Critic structure, performs better in markets with low volatility or weak price trends. It effectively handles gradual and continuous market changes, allowing for optimal decision-making. In contrast, PPO, known for its stability in policy updates, achieves the best results in highly volatile and unpredictable markets. By preventing drastic policy changes, PPO adapts well to sudden market fluctuations. The DDPG algorithm, which follows a deterministic policy, performs best in trending markets where changes occur gradually according to recognizable patterns. TD3, an improved version of DDPG, excels in markets with clear trends but significant noise. By employing two Critic networks to minimize errors and noise, TD3 can make more precise decisions under such conditions. Finally, leveraging entropy to balance exploration and exploitation, the SAC algorithm demonstrates superior

performance in unstable and complex markets. SAC is particularly effective in environments with sudden and intricate fluctuations, offering high flexibility and responsiveness.

Given the performance of these algorithms across different market conditions, a 10-day time window was selected for determining the optimal agent. Analyzing cumulative returns over various time windows (5, 10, 15, 20, 25, and 30 days) revealed that the 10-day window outperformed the others. The results are summarized in Table XII.

TABLE XII. CUMULATIVE RETURNS FOR DIFFERENT TIME WINDOWS.

Days	Cumulative Return
5	1.34
10	1.38
15	1.30
20	1.34
25	1.31
30	1.27

As shown in Table XII, the 10-day window achieved the highest cumulative return among all tested periods. This result indicates its adaptability and efficiency in both short-term and long-term market conditions. In short-term markets characterized by rapid fluctuations, the 10-day window enables the model to react quickly to new conditions and make optimal decisions, enhancing its resilience to extreme volatility. This window effectively captures gradual market trends in long-term markets, ensuring the model remains consistently aligned with evolving conditions.

Ultimately, the findings suggest that the 10-day time window provides superior performance in high-volatility and trend-driven markets. This selection enables the model to adapt efficiently across different market environments, leading to more robust and optimal decision-making.

C. Research Limitation

One limitation of this research is the exclusion of legal and regulatory constraints, which vary based on market conditions and financial regulations in different countries. These factors primarily impact the operational deployment of a trading strategy rather than the development of the underlying reinforcement learning framework. As this research focuses on designing an adaptive decision-making model applicable across various market environments, regulatory considerations were not explicitly incorporated. Instead, such constraints are more relevant to the practical implementation of the strategy on specific trading platforms, where compliance with market regulations is essential.

VI. CONCLUSION AND FUTURE WORKS

This paper introduced a novel consulting system for portfolio asset allocation that leveraged deep reinforcement learning to enhance investment decision-making. By integrating five distinct DRL agents (A2C, PPO, DDPG, TD3, and SAC) and employing a short-term agent selection mechanism based on cumulative returns, the system dynamically adapted to shifting market conditions and significantly improved portfolio performance. Backtesting on historical Dow Jones index data demonstrated that the approach achieved an 11.43% average annual return, 38.29% cumulative returns, and a Sharpe ratio of 0.832, outperforming individual agents and conventional strategies.

Future enhancements to the proposed system include incorporating additional financial indicators and alternative data sources, such as sentiment analysis from news and social media, to achieve more comprehensive market insights. Expanding the system to accommodate other asset classes, including bonds, commodities, and cryptocurrencies, would increase its versatility. Evaluating the system's robustness and adaptability in fuzzy environments characterized by high uncertainty and market volatility presents another valuable avenue for research. Additionally, investigating the use of transfer learning to adapt models trained on one market to other markets could enhance cross-market generalization and improve system effectiveness. Finally, developing strategies for real-time implementation would enable the system to operate efficiently in live trading environments, bridging the gap between theoretical advancements and practical applications.

DATA AND CODE AVAILABILITY

This research utilizes DJIA daily stock data from Yahoo Finance (https://finance.yahoo.com), covering open, high, low, and close prices and trading volumes for 30 major U.S. companies. The source code and dataset are publicly available at https://github.com/MasoudKargar/APO-MADRL-STA.

REFERENCES

[1] A. Afonin, D. Bredin, K. Cuthbertson, C. Muckley, and D. Nitzsche, “Carbon portfolio management,” Int. J. Finance Econ., vol. 23, pp. 349– 361, 2018.

[2] A. Gunjan and S. Bhattacharyya, "A brief review of portfolio optimization techniques," Artificial Intelligence Review, vol. 56, no. 5, pp. 3847–3886, 2023.

[3] A. Bayani and M. Kargar, “LDCNN: A new arrhythmia detection technique with ECG signals using a linear deep convolutional neural network,” Physiol. Rep., vol. 12, no. 17, p. e16182, 2024.

[4] M. Karegar, T. Saderi, A. Isazadeh, and F. Fartash, “Electronic consulting in marketing,” in Proc. 3rd Int. Conf. Inf. Commun. Technol. Theory Appl., 2008.

[5] C. Huang et al., “Multi-hop RIS-empowered terahertz communications: A DRL-based hybrid beamforming design,” IEEE J. Sel. Areas Commun., vol. 39, pp. 1663–1677, 2021.

[6] S. S. Movassaghpour, M. Kargar, A. Bayani, A. Assadzadeh, and A. Khakzadi, "A consulting system for portfolio assets allocation by selecting the best agent in the short term based on cumulative returns with deep reinforcement learning," in Proc. 2024 11th Int. Symp. Telecommun. (IST), 2024, pp. 141-149.

[7] H. Mashal and M. H. Rezvani, "Multiobjective offloading optimization in fog computing using deep reinforcement learning," J. Comput. Netw. Commun., vol. 2024, no. 1, Art. no. 6255511, 2024.

[8] C. Stetter, R. Huber, and R. Finger, "Agricultural land use modeling and climate change adaptation: A reinforcement learning approach," Appl. Econ. Perspect. Policy, vol. 46, no. 4, pp. 1379–1405, 2024.

[9] V. Mnih et al., “Asynchronous methods for deep reinforcement learning,” in Proc. Int. Conf. Mach. Learn., pp. 1928–1937, 2016.

[10] Y. Wang, H. He, and X. Tan, “Truly proximal policy optimization,” in Proc. 35th Uncertainty Artif. Intell. Conf., vol. 115, pp. 113–122, Jul. 2020. [Online]. Available: https://proceedings.mlr.press/v115/wang20b.html

[11] T. Lillicrap et al., “Continuous control with deep reinforcement learning,” arXiv preprint arXiv:1509.02971, 2015.

[12] S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in Proc. Int. Conf. Mach. Learn., pp. 1587–1596, 2018.

[13] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Offpolicy maximum entropy deep reinforcement learning with a stochastic actor,” in Proc. Int. Conf. Mach. Learn., pp. 1861–1870, 2018.

[14] P. N. Kolm, R. Tütüncü, and F. J. Fabozzi, "60 years of portfolio optimization: Practical challenges and current trends," Eur. J. Oper. Res., vol. 234, no. 2, pp. 356–371, 2014.

[15] M. Abbasi, M. Kargar, F. Ahmadian, D. NoormohammadZadehMaleki, A. Arandan, and N. S. Hosseini, "GN-CNN-LSTM: Financial market prediction with Gaussian noise embedded CNN LSTM," in Proc. 11th Int. Symp. Telecommun. (IST), 2024, pp. 287–294.

[16] C. Betancourt and W. Chen, “Deep reinforcement learning for portfolio management of markets with a dynamic number of assets,” Expert Syst. Appl., vol. 164, p. 114002, 2021.

[17] M. Katongo and R. Bhattacharyya, “The use of deep reinforcement learning in tactical asset allocation,” SSRN Electron. J., 2021.

[18] P. Koratamaddi, K. Wadhwani, M. Gupta, and S. Sanjeevi, “Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation,” Eng. Sci. Technol. Int. J., vol. 24, pp. 848–859, 2021.

[19] Q. Lim, Q. Cao, and C. Quek, “Dynamic portfolio rebalancing through reinforcement learning,” Neural Comput. Appl., vol. 34, pp. 7125–7139, 2022.

[20] T. Kabbani and E. Duman, “Deep reinforcement learning approach for trading automation in the stock market,” IEEE Access, vol. 10, pp. 93564–93574, 2022.

[21] T. Zhao, X. Ma, X. Li, and C. Zhang, “Asset correlation based deep reinforcement learning for the portfolio selection,” Expert Syst. Appl., vol. 221, p. 119707, 2023.

[22] J. Li, Y. Zhang, X. Yang, and L. Chen, “Online portfolio management via deep reinforcement learning with high-frequency data,” Inf. Process. Manage., vol. 60, p. 103247, 2023.

[23] C. Ma, J. Zhang, Z. Li, and S. Xu, “Multi-agent deep reinforcement learning algorithm with trend consistency regularization for portfolio management,” Neural Comput. Appl., vol. 35, pp. 6589–6601, 2023.

[24] Z. Hao, H. Zhang, and Y. Zhang, “Stock portfolio management by using fuzzy ensemble deep reinforcement learning algorithm,” J. Risk Financial Manage., vol. 16, p. 201, 2023.

[25] Y. Zhang et al., “Cost-sensitive portfolio selection via deep reinforcement learning,” IEEE Trans. Knowl. Data Eng., vol. 34, pp. 236–248, 2020.

[26] F. Soleymani and E. Paquet, “Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder— DeepBreath,” Expert Syst. Appl., vol. 156, p. 113456, 2020.

[27] X. Wu et al., “Adaptive stock trading strategies with deep reinforcement learning methods,” Inf. Sci., vol. 538, pp. 142–158, 2020.

[28] M. Wu, J. Syu, J. Lin, and J. Ho, “Portfolio management system in equity market neutral using reinforcement learning,” Appl. Intell., vol. 51, pp. 8119–8131, 2021.

[29] S. Carta, A. Corriga, A. Ferreira, A. Podda, and D. Recupero, “A multilayer and multi-ensemble stock trader using deep learning and deep reinforcement learning,” Appl. Intell., vol. 51, pp. 889–905, 2021.

[30] Y. Lin, C. Chen, C. Sang, and S. Huang, “Multiagent-based deep reinforcement learning for risk-shifting portfolio management,” Appl. Soft Comput., vol. 123, p. 108894, 2022.

[31] J. Jang and N. Seong, “Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory,” Expert Syst. Appl., vol. 218, p. 119556, 2023.

[32] M. Day, C. Yang, and Y. Ni, “Portfolio dynamic trading strategies using deep reinforcement learning,” Soft Comput., 2023.

[33] X. Yu, W. Wu, X. Liao, and Y. Han, “Dynamic stock-decision ensemble strategy based on deep reinforcement learning,” Appl. Intell., vol. 53, pp. 2452–2470, 2023.

[34] H. Li and M. Hai, “Deep reinforcement learning model for stock portfolio management based on data fusion,” Neural Process. Lett., vol. 56, p. 108, 2024.

[35] Y. Jiang, J. Olmo, and M. Atwi, "Deep reinforcement learning for portfolio selection," Glob. Finance J., vol. 62, Art. no. 101016, 2024.

[36] T.-F. Chen, X.-J. Kuang, S.-L. Liao, and S.-K. Lin, "Portfolio allocation with dynamic risk preferences via reinforcement learning," Comput. Econ., vol. 64, no. 4, pp. 2033–2052, 2024.

[37] P. K. Aritonang, S. K. Wiryono, and T. Faturohman, "Hidden-layer configurations in reinforcement learning models for stock portfolio optimization," Intell. Syst. Appl., vol. 25, Art. no. 200467, 2025.

[38] G. Song, T. Zhao, X. Ma, P. Lin, and C. Cui, "Reinforcement learning-based portfolio optimization with deterministic state transition," Inf. Sci., vol. 690, Art. no. 121538, 2025.

[39] A. Sattar et al., "A novel RMS-driven deep reinforcement learning for optimized portfolio management in stock trading," IEEE Access, 2025.

[40] X. Liu et al., “FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance,” arXiv preprint arXiv:2011.09607, 2020.

Shahin Sharbaf Movassaghpour is a computer engineer and Ph.D. candidate at Islamic Azad University, Tabriz Branch, specializing in artificial intelligence applications in healthcare and economics. He is a core member of the Robotics and Soft Technologies Research Center at the same university.

Masoud Kargar, Assistant Professor at Islamic Azad University, Tabriz Branch, specializes in artificial intelligence with deep learning, reinforcement learning, and GANs applied to finance and healthcare. He leads the AI Research Group in Healthcare and Economics at the Robotics and Soft Technologies Research Center, has authored several books, including Deep Learning Clues and C++, and serves as an Associate Editor and Reviewer for computer science and AI journals.

Ali Bayani is a computer engineer specializing in machine learning and artificial intelligence. He is a core member of the Robotics and Soft Technologies Research Center at Islamic Azad University, Tabriz Branch. His research focuses on advanced AI methodologies for economics, healthcare, and medicine.

Alireza Assadzadeh is a computer engineer with expertise in artificial intelligence. He is a member of the Robotics and Soft Technologies Research Center at Islamic Azad University, Tabriz Branch.

Ali Khakzadi is a computer engineer focused on machine learning and economics. He contributes to projects at the Robotics and Soft Technologies Research Center, Islamic Azad University, Tabriz Branch.