Dynamic Reinforced Ensemble using Bayesian Optimization for Stock Trading

Arishi Orra
Indian Institute of Technology Mandi
India
arishi.orra98@gmail.com

Aryan Bhambu
Indian Institute of Technology Guwahati
India
a.bhambu@iitg.ac.in

Himanshu Choudhary
Indian Institute of Technology Mandi
India
ch.himanshu1199@gmail.com

Manoj Thakur
Indian Institute of Technology Mandi
India
manoj@iitmandi.ac.in

Abstract

In the world of automated stock trading, Deep Reinforcement Learning (DRL) techniques have become highly effective due to their inherent capability of learning optimal trading strategies through trial and error. However, a single DRL agent lacks the flexibility to adapt to the complex and evolving market dynamics, yielding a suboptimal strategy. This paper proposes a Bayesian optimization-based dynamic ensemble approach utilizing various model-free DRL algorithms for multi-stock trading. The ensemble method enhances decision-making by leveraging the policies of diverse models while mitigating the risk of overfitting. The proposed model uses recent historical data to compute dynamic time-varying weights for the ensemble and employs Bayesian optimization for hyperparameter tuning. The effectiveness of our proposed approach is assessed using two global stock market indices: Dow Jones from the U.S. and Sensex from India. The empirical results demonstrate that the proposed methodology outperformed the market indices and several other benchmark trading strategies, taking into consideration the risk and return measures. Our ensemble approach demonstrated the ability to achieve higher profits with reduced risks and rapid recovery after corrections.

CCS Concepts

• **Computing methodologies** → **Markov decision processes**; **Reinforcement learning**; **Ensemble methods**.

Keywords

Deep reinforcement learning, Bayesian Optimization, Ensemble Learning, Advantage Actor-Critic, Proximal Policy Optimization, Stock Trading, Quantitative Finance

ACM Reference Format:
Arishi Orra, Aryan Bhambu, Himanshu Choudhary, and Manoj Thakur. 2024. Dynamic Reinforced Ensemble using Bayesian Optimization for Stock Trading. In *5th ACM International Conference on AI in Finance (ICAIF '24), November 14–17, 2024, Brooklyn, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3677052.3698595

ICAIF '24, November 14–17, 2024, Brooklyn, NY, USA

ACM ISBN 979-8-4007-1081-0/24/11
https://doi.org/10.1145/3677052.3698595

1 Introduction

A practical automated stock trading system is crucial for hedge funds and investment firms. It helps them execute trades far faster than humans and eliminates emotional bias. Machine learning and deep learning (DL) techniques are rising in prominence for developing automated stock trading methods due to their ability to analyze intricate data patterns [13, 27, 28, 32]. DL methods are often the first choice of practitioners as they can learn rich feature representations and nonlinear relationships from the data. Long Short-Term Memory (LSTM) networks and several other DL models have frequently been employed in the literature to predict stock prices [3, 4, 6, 9]. However, these methods lack robustness due to the noisy and nonstationary nature of the stock market data. They are also prone to overfitting because of their complex architecture [5].

An alternate approach for designing an optimal trading strategy is to model it as a Markov Decision Process (MDP) and apply Reinforcement learning (RL) techniques to solve it [12, 23, 25, 26, 30]. RL techniques try to learn the complex dynamics of the market by interacting with it through trial and error [34]. Rapid advancements in deep neural networks extend the conventional RL techniques to Deep Reinforcement learning (DRL) by approximating the value functions and policies for high dimensional state and action spaces [16, 22]. Lu [19] employed an LSTM network along with the policy gradient method for designing a Forex trading model to optimize risk measures such as the differential Sharpe ratio and downside deviation ratio. The agent effectively mitigated the downside risks driven by exchange rate volatility. Wu et al. [36] proposed a Gated Recurrent Unit (GRU) and DRL-based system to make adaptive trading decisions. They used GRU to extract informative features from stock data, integrated it with Deep Q-Learning (DQN) [22] and Deep Deterministic Policy Gradient (DDPG) [17] agents, and tested this approach in a trendy and volatile market. Zhang et al. [40] utilized DRL algorithms to craft trading strategies for continuous futures contracts to enhance the reward function by incorporating volatility scaling. Their methodology exhibited the capability to capture significant market swings and adjust to periods of consolidation with flexibility. Kabbani and Duman [14] formulated the stock trading problem as a Partially Observed Markov Decision Process

(POMDP) and employed the Twin Delayed Deep Deterministic Policy Gradient (TD3) [10] algorithm for generating profitable trades. Avramelou et al. [2] presented a novel multi-modal embedding-based method for combining the features from price and sentiment data. A Proximal Policy Optimization (PPO) [31] agent then utilized these embedded features to trade cryptocurrencies.

Each of the DRL algorithms has a different architecture and diverse characteristics. Due to its dynamic nature, the stock market exhibits a variety of trends during a trading cycle. As such, a single DRL model fails to capture distinct market movements and lacks generalizability. To overcome this issue, practitioners employ an ensemble of various DRL agents harnessing their intelligence to apprehend patterns in the volatile market. Yang et al. [38] proposed an ensemble approach that retrains three DRL agents, namely, Advantage Actor-Critic (A2C) [21], DDPG, and PPO for a window of three months, and then select the best agent stemming from the Sharpe ratio to trade for the following quarter. Carta et al. [5] trained multiple instances of DQN using the same training data and suggested a threshold-based scheme to make the final trading decision. More recently, a nested RL methodology integrating A2C, DDPG, and Soft Actor-Critic (SAC) [11] models was proposed by Yu et al. [39] that dynamically chooses agents adaptive to the state of the market.

Diversifying the base learners across various algorithmic fields enhances the input space, providing the ensemble with a broader selection and facilitating effective combinations. Furthermore, the static optimal forecast combination determines model weights through various techniques, including error-based methods [1] and an equally weighted model combination approach [7]. Subsequently, Du et al. [8] introduced a Bayesian optimization-based ensemble method, incorporating an adaptive combination technique that assigns time-varying weights based on modified in-sample training-validation pair neural networks. We introduced modifications to [8] and proposed a Dynamic Reinforced Ensemble method using Bayesian optimization [29] (DREB) that assigns the time-varying adaptive weights to the base learners. The proposed model also utilizes the dynamic weight strategy. In particular, DREB utilizes five model-free Deep RL techniques: A2C [21], DDPG [17], TD3 [10], SAC[11], and PPO [31] as base learners and then employs Bayesian Optimization to assign time-varying weights for developing automated stock trading strategies. The effectiveness of the proposed approach in generating higher risk-averse returns is demonstrated across two global stock market indices.

The main contributions of our paper are as follows:

(1) We have proposed a novel Dynamic Reinforced Ensemble method using Bayesian Optimization (DREB) for learning automated stock trading strategy.
(2) We utilize five diversified DRL techniques across various algorithmic frameworks as base learners to constitute the ensemble. The Bayesian optimization technique is employed to fine-tune hyperparameters of time-varying weights.
(3) The empirical results and simulation studies across two global stock markets demonstrated the superiority of our proposed ensemble over various benchmark trading strategies.

The remainder of this paper is structured as follows. Section 2 introduces the proposed ensemble methodology and also provides a brief description of the stock trading problem and the environment used for simulation. The experimental settings and the performance comparison are reported in section 3. Section 4 concludes the paper and suggests some direction for future work.

2 Proposed Methodology

The stock trading problem as an RL objective is addressed in this section. We also present the stock trading environment used for simulation and the proposed ensemble approach.

2.1 Problem Definition

We formulated the problem of multi-stock trading as an MDP. The dynamic and stochastic nature of the stock market makes it an ideal environment for DRL algorithms. The uncertainty of the future prices transforms it into a stochastic control problem, which can be solved using model-free DRL methods. Stock trading aims to maximize the long-term cumulative return, analogous to the RL objective of finding a policy that maximizes the discounted future cumulative rewards. In our problem, the agent interacts with the environment at discrete time steps and performs suitable action. Stock prices represent the state and satisfy the Markov property. At step t, the agent observers state s_t, which comprises t^{th} day stock prices, and takes action $a_t \in \{buy, sell, hold\}$. Then, the environment gives the agent a numeric reward feedback and the next day's stock price in the form of the following state s_{t+1}. The goal of the trading agent is to learn a policy $\pi(a_t|s_t)$ that maximizes the expected cumulative return by interacting with the environment.

2.2 Environment for stock trading

We employed a stock trading environment provided by the open-source library FinRL [18], which simulates a practical trading scenario. The environment includes various data that an agent requires to learn to trade by interacting with it, such as stock prices, technical indicators, current holdings, and more. The agent observes the information of stocks as a state, takes suitable action, and consequently obtains a reward from the environment.

2.2.1 State Space. A state of the environment consists of OHLC (Open, High, Low, and Close) prices, technical indicators, current holdings of shares, and the remaining balance. The representation of the state space is motivated from Xiong et al. [37]. In our multi-stock trading problem of n stocks, the agent observes a $(13n + 1)$ dimensional state vector at each step. The components of the state vector are briefly described as follows:

(1) Remaining balance amount with the agent.
(2) Total number of shares that the agent owns of each stock.
(3) OHLC prices of each stock.
(4) Eight technical indicators (30 and 60 day Simple Moving Averages (SMA), Moving Average Convergence Divergence (MACD), Upper and lower Bollinger bands, Relative Strength Index (RSI), Commodity Channel Index (CCI), and Average Directional Index (ADX) [24]) corresponding to each of the n stocks.

2.2.2 Action Space. This work considers a continuous action space in the interval $[-1, 1]$. For each stock, at step t, action

$$a_t \in \{-h, \ldots, 0, \ldots, h\},$$

where $-h$ and h denotes the number of shares to sell or buy, respectively. Therefore, for a n stock trading problem, the complete action space is of dimension $(2h+1)^n$, which is subsequently normalized to $[-1, 1]$. We also constrained the maximum number of shares to buy or sell in a single step as $h \leq 100$.

2.2.3 Reward. For this multi-stock trading problem, we define the reward function at step t as a change in the portfolio value following an action a_t. Let I_t be the portfolio value at step t, and I_{t+1} is the portfolio's value after taking action a_t, then the reward at this step is given by:

$$r(t) = I_{t+1} - I_t - c_t$$

where c_t is the transaction cost of executing action a_t. As defined by Thakur and Kumar [35], a transaction cost of 0.05% of the overall transaction is assumed. As an example, at step t, if a_t is the vector containing the number of shares of each stock to be bought or sold and p_t is the vector of the price of each stock, then $c_t = |a_t^T p_t \times 0.05\%|$. This reward function drives the trading agent to select actions to maximize the portfolio's total return while minimizing transaction costs.

2.3 Bayesian Optimization

Bayesian Optimization (BO) [29] is a systematic approach based on Bayes' theorem that maps hyperparameters to their probability of performing well on the objective function. Unlike grid or random search, BO maintains a record of all past evaluations, thus avoiding the computational waste of evaluating poor hyperparameters. Additionally, an acquisition function identifies the most promising hyperparameters to evaluate in the next iteration. In this work, we utilize the Tree-structured Parzen Estimator (TPE) method [33] for probabilistic modeling of the surrogate function. BO techniques begin by constructing the initial configuration of the hyperparameter space. As the number of iterations progresses, the technique optimizes and identifies the best hyperparameters through a series of steps: creating a surrogate probability model for the objective function, determining the best hyperparameters for the surrogate, applying these hyperparameters to the objective function, and updating the surrogate model with new data.

2.4 Proposed Ensemble

We propose an ensemble approach utilizing various model-free DRL techniques for learning automated stock trading strategies as illustrated in Figure 1. In our work, we employ five DRL algorithms, namely, A2C, DDPG, TD3, SAC, and PPO, and each of them has its distinctive flair in terms of underlying principles, optimization, and exploration-exploitation tradeoffs. A2C [21] intertwines the elegance of policy gradients with the value functions and utilizes the advantage function to improve policy updates. DDPG [17] excels in continuous action spaces by preserving a deterministic policy, while TD3 [10] advances DDPG by using twin critics and delayed policy updates to enhance training stability. SAC [11] benefits from an entropy maximization objective, balancing the exploration-exploitation tradeoff while maintaining robust policy learning. PPO [31] avoids large policy updates and guarantees stable training by clipped probability ratios.

This proposed methodology leverages the weighting scheme introduced in [8] for assigning the dynamic weights for the base model. We introduce the flexibility of incorporating a maximum number, denoted as n_m, of models for forecast combination. This approach employs the cold-start index i_c during testing to start with the specific model M_{i_c}. Additionally, a dynamic weight strategy is implemented for optimal combination. Let at time step t, $\hat{y}_{i_M}(t)$ be the return of the i_M^{th} base model, and $y(t)$ is the target return that the agent is expected to achieve. The weight $\hat{w}_{i_M}(t)$ for each model i_M at time t is calculated using the following formula:

$$\hat{w}_{i_M}(t) = \begin{cases} \frac{\sum_{i=t-1}^{t-e_w} \phi(\gamma^i(\hat{y}_{i_M}(i) - y(i)))}{\sum_{k_M=1}^{n_m} \sum_{i=t-1}^{e_w} \phi(\gamma^i(\hat{y}_{k_M}(i) - y(i)))} & \text{if } i_M \in N_m(t) \\ k_M \in N_m(t) & \\ 0 & \text{else.} \end{cases}$$

Here, e_w represents the evaluation window for calculating the weight, ϕ signifies the chosen weighting scheme: inverse mean squared error (IMSE), softmax, or inverse mean absolute error (IMAE), γ acts as the discount factor emphasizing the importance of nearer observations, n_m is the maximum number of models for ensemble selection, and $N_m(t)$ is the set of models choosing the top n_m based on their rank in terms of accumulated error.

The final weight, denoted as $w_{i_M}(t)$, is determined by the following equation:

$$w_{i_M}(t) = \begin{cases} 1 & \text{if } i_M = i_c \in N_m(t), \quad i = 1 \\ l w_{i_M}(t-1) + (1-l)\hat{w}_{i_M}(t) & \text{if } i_M = 1, 2, \ldots \end{cases}$$

where $l \in [0, 1]$ serves as the update parameter for the weights corresponding to each model, allowing for a balance between historical weights and the current dynamic weight. This ensemble approach, incorporating a variety of models and dynamic weights, aims to enhance the stock trading strategy through the combination of diverse forecasting perspectives. The hyperparameters of the ensemble approach, such as i_c, e_w, ϕ, γ, and l, are fine-tuned using Bayesian optimization [29]. The proposed methodology requires specifying the target returns that the trading agent is expected to achieve at each step. To determine these target returns, expert trajectories are created using the available historical training data. We have extended Lee and Moon [15]'s formulation from designing expert actions for portfolio optimization to trading, where the action for the i^{th} stock at step t is defined as follows:

$$action_{i,t} = \begin{cases} \frac{-e^{\rho_{i,t}.z}}{\sum_i e^{|\rho_{i,t}.z|}} & \rho_{i,t} < 0 \\ \frac{e^{\rho_{i,t}.z}}{\sum_i e^{|\rho_{i,t}.z|}} & \rho_{i,t} \geq 0, \end{cases} \tag{1}$$

where $\rho_{i,t}$ is the percentage change in the close price of stock i at step t, and z is a constant ranging from 1 to 5. These actions are then utilized to compute the target return for our agent.

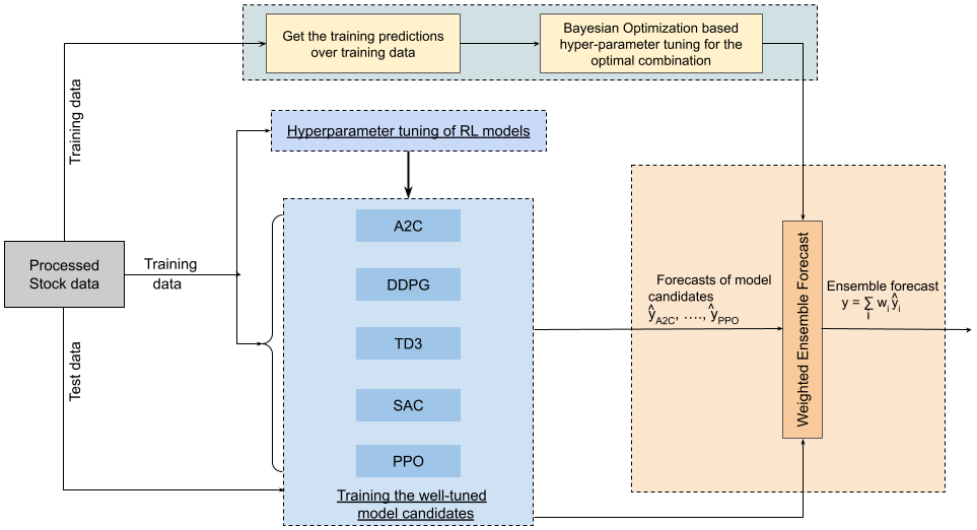

Figure 1: The schematic diagram depicting the workflow of the DREB algorithm. The stock data is partitioned into training and test sets. The training data is employed for model training and hyperparameter tuning, while the test data is utilized by the ensemble to determine the optimal course of action.

3 Results and Discussion

In this section, we present the dataset utilized for evaluation, as well as the performance measures and baseline trading strategies for assessing the effectiveness of our proposed ensemble method.

3.1 Data and Experimental Setting

The effectiveness of the proposed DREB model is assessed across two global market indices: the Dow Jones Index (DJI) from the U.S. and the Sensex from India. DJI is one of the oldest and most popular equity indexes, comprised of the top 30 companies traded in the U.S. stock market, while Sensex, the oldest stock index of India, comprises the top 30 stocks traded on the Bombay Stock Exchange. For experimentation, we select daily data of all the available constituents of these indices from 01/01/2010 to 31/03/2024. During this period, data for only 29 out of 30 stocks exists for DJI, while all the 30 stock data is available for Sensex. The dataset consists of OHLC prices collected from Yahoo Finance. In-sample data from 01/01/2010 to 31/12/2021 is utilized for training the agents, and their trading performance is assessed on out-of-sample data spanning from 01/01/2022 to 31/03/2024. All the employed DRL agents are adopted from the open-source FinRL [18] library.

3.2 Performance Measures and Baselines

The efficacy of the proposed approach is assessed using the following metrics:

(1) **Cumulative Return:** The overall return generated at the end of the trading period.
(2) **Annualized Return:** It measures average annual returns yielded during the entire trading period.
(3) **Sharpe Ratio:** It is defined as the excess return that a trader makes per unit risk.
(4) **Annualized Volatility:** It measures the dispersion of returns and quantifies the risk associated with the investment.
(5) **Maximum Drawdown:** It is a measure of downside risk and is calculated as the highest recorded loss from any peak of trading capital to a trough.
(6) **Average Profit per Trade:** The average profit or loss a trader incurs per trade.

Using these aforementioned metrics, the efficacy of our proposed ensemble model is assessed against the base models and the following benchmark trading strategies:

(1) **Market Index:** It is a hypothetical portfolio representing a subset of the stock market, used to measure the market's performance. This study uses the DJI and Sensex indexes to represent the U.S. and Indian markets. Both indices constitute the price-weighted average of the top 30 stocks of the exchange.
(2) **Buy-and-Hold Strategy:** In the traditional Buy-and-Hold trading strategy, the agent buys at the beginning of the trading session and holds it through to the end.
(3) **Mean Trading:** In this strategy, the agent trades by selecting the mean action of all the base models.
(4) **Random Trading:** In random trading, the agent takes a trading position randomly at each step.
(5) **Adaptive Ensemble:** Existing ensemble method provided by Yang et al. [38].
(6) **Mean-Variance Optimization (MVO):** The MVO model [20] is a mathematical framework that determines the most efficient portfolio for the underlying assets by assessing the expected risk and returns.

3.3 Results and Analysis

We backtested the base models and the benchmark methods for the trading period from 01/01/2022 to 31/03/2024. An initial capital of 100, 000 was allocated to all the agents to trade with. The performance comparison of all the models based on the abovementioned performance metrics for the DJI and Sensex data is reported in Table 1 and Table 2, respectively. The analysis of the results is twofold. Firstly, the proposed ensemble approach is assessed against the base models. Afterward, the ensemble model is evaluated against the benchmarks.

Table 1 exhibits a detailed evaluation of the base DRL agents and the proposed ensemble approach on the DJI dataset. All the base DRL agents generated higher returns than the DJI index and consequently attained a higher Sharpe ratio except for TD3. However, these agents also exhibited high annualized volatility and drawdown, indicating a significant risk level. Among them, PPO achieved the highest cumulative and annualized returns of 13.46% and 5.77%, respectively. DDPG, TD3, and SAC produced higher average profitability than PPO but also had high volatility and drawdown. Therefore, in terms of risk-adjusted returns, PPO outperformed all other base agents, achieving the highest Sharpe ratio of 0.43. Our proposed DREB model surpassed the PPO agent by generating superior returns and a higher Sharpe ratio. It also attained the highest average profitability with the least volatility among the base agents. This underscores the effectiveness of our proposed ensemble model in delivering superior performance and greater stability compared to the base learners. A plot of cumulative returns of the base trading agents and the ensemble method is depicted in Figure 2. The initial trading period until October 2022 can be seen as bearish, during which all the base agents incurred substantial losses. Our proposed DREB model exhibited minimal losses during this period and experienced the least drawdown, highlighting its risk-averse capability. The second trading period is mostly sideways till October 2023. In this volatile market, these base agents are primarily indecisive. They achieved profits over shorter periods but then experienced intermittent losses. In contrast, the DREB model effectively captured short-term trends and consistently achieved small profits. Thus, it recovered its initial loss and also started to accumulate wealth. The last trading period was primarily bullish, and all agents produced significant profits. PPO generated the highest returns among the base agents, followed by DDPG and SAC. Our proposed DREB model outperformed them by generating consistent returns throughout the entire trading period.

Figure 3 shows the cumulative return plot of the ensemble model and the benchmark trading strategies for the DJI data. The random trading strategy is the worst performer as it takes actions randomly and has an equal chance of winning or losing at each step. Hence, it does not follow any trend and randomly makes small profits and losses. The DJI represents the behavior of the stock market during the trading period. It initially had a bearish movement, followed by a sideways trend, and then followed a bullish trend towards the end. The Buy-and-Hold strategy invests in the index components and then holds till the end. Therefore, it mimics every movement of the index and slightly exceeds it. The mean trading strategy selects an average action of the DRL agents. Due to the high variability and drawdown of the individual agents, the mean actions suffered substantial setbacks and performed nearly similarly to the Buy-and-Hold strategy. This indicates how arduous it is to profit in this dynamic and volatile market. The adaptive ensemble approach selects an optimal trading agent for three months, failing to capture the short-term market fluctuations entirely. Although it beats the market index, it underperformed the mean trading strategy. The MVO model determines the optimal asset allocations based on risk and return tradeoffs. It experienced the smallest drawdowns among the baselines during bearish markets while achieving higher profits during sideways markets. It exhibited the least variability and surpassed all baselines to achieve superior returns.

Our ensemble approach, which considers the weighted actions of all the DRL agents, outperforms all the benchmarks by a considerable margin. The ensemble method endured the least capital loss even during the market's bearish trend. Also, during the bullish period, it accumulates quick wealth. As seen from Table 1, the ensemble method has the highest returns, Sharpe ratio, and average profitability among the benchmarks. Random trading has minimal volatility and maximum drawdown of 0.24% and 2.76%, respectively, because of its small average profits per trade and hence yields subpar long-term returns. Nonetheless, this trade-off between risk and reward is acceptable, given the more significant cumulative return and the bearish state of the market. Overall, our ensemble approach demonstrates a caliber of obtaining higher profits with reduced risks and quick recovery after corrections.

Table 2 describes the performance comparison of the proposed DREB model against the base agents and benchmarks for the Sensex data. Similar performance to the DJI data is also evident here. PPO attained the highest average returns of 16.75% among the base agents, followed by TD3 with 16.35%. A2C achieved the maximum Sharpe ratio of 1.32 with the least volatility of 1.11%. All the base DRL agents surpassed the market index by a considerable margin. However, our proposed ensemble model significantly outperformed the base agents concerning all six performance metrics. Among the benchmarks, the MVO model generated the highest average returns of 52.35% and a Sharpe ratio of 1.69, nearly doubling the market index's performance. The proposed DREB model achieved slightly higher returns than the MVO model, along with better average profitability. Random trading showed a minimum volatility of 0.19% and a maximum drawdown of 1.52% due to its subpar returns. Otherwise, DREB outperformed all other benchmarks in exhibiting the least volatility and thus produced the highest Shape ratio of 1.82. Overall, the DREB model demonstrated its effectiveness by generating higher risk-adjusted returns, outperforming both the base agents and the benchmarks. The cumulative return plots of the DREB model versus the base agents and the benchmarks are exhibited in Figure 4 and 5, respectively. Figure 4 showcases the efficacy of our proposed ensemble model in attaining higher cumulative returns than the base DRL agents. It incurred a loss on its initial capital only once and, after that, accumulated greater wealth. Figure 5 shows that the MVO model outperformed the other benchmarks by generating higher risk-free returns. It performed nearly as well as DREB but with slightly more variability. At times, it outperformed the DREB model but experienced more frequent losses. On the other hand, the DREB model generated more consistent returns with lower volatility. Considering the risk and return

Table 1: Performance evaluation of the proposed approach against the base learners and the benchmarks for the DJI data.

Method/Benchmark	Cummulative Return	Annualized Return	Sharpe Ratio	Annualized Volatility	Maximum DrawDown	Average Profit per Trade
A2C [21]	10.05%	4.35%	0.34	1.56%	19.66%	501.99
DDPG [17]	12.95%	5.56%	0.40	1.67%	22.81%	564.88
TD3 [10]	9.29%	4.03%	0.31	1.72%	25.28%	553.97
SAC [11]	11.63%	5.01%	0.36	1.71%	25.29%	537.96
PPO [31]	13.46%	5.77%	0.43	1.52%	20.13%	519.72
DJI	8.81%	3.82%	0.32	1.47%	21.94%	-
Buy-hold	12.44%	5.34%	0.37	1.78%	26.79%	-
Mean Trading	11.98%	5.16%	0.37	1.73%	24.31%	435.08
Random Trading	5.06%	2.21%	0.85	**0.24%**	**2.76%**	127.60
Adaptive Ensemble [38]	10.79%	4.66%	0.35	1.69%	20.68%	520.83
MVO [20]	14.47%	6.19%	0.55	1.18%	14.67%	490.03
DREB (ours)	**15.24%**	**6.51%**	**1.05**	0.59%	6.56%	**569.49**

Table 2: Performance evaluation of the proposed approach against the base learners and the benchmarks for the Sensex data.

Method/Benchmark	Cummulative Return	Annualized Return	Sharpe Ratio	Annualized Volatility	Maximum DrawDown	Average Profit per Trade
A2C [21]	38.12%	15.43%	1.32	1.11%	14.53%	610.18
DDPG [17]	37.05%	15.04%	1.27	1.13%	15.36%	610.28
TD3 [10]	40.58%	16.35%	1.30	1.18%	13.96%	661.92
SAC [11]	35.23%	14.36%	1.19	1.15%	14.33%	650.98
PPO [31]	41.69%	16.75%	1.01	1.62%	22.18%	770.34
Sensex	26.67%	11.08%	0.85	1.31%	16.47%	-
Buy-hold	32.54%	13.34%	0.99	1.32%	18.76%	-
Mean Trading	28.83%	11.92%	0.89	1.35%	21.03%	648.86
Random Trading	6.27%	2.74%	1.50	**0.19%**	**1.52%**	91.89
Adaptive Ensemble [38]	38.91%	15.73%	0.91	1.73%	20.66%	680.19
MVO [20]	52.35%	20.57%	1.69	1.13%	11.05%	749.59
DREB (ours)	**56.05%**	**21.87%**	**1.82**	1.04%	9.54%	**791.84**

Figure 2: Cumulative returns of the proposed DREB model and the base DRL agents for the DJI data over the entire trading period.

tradeoff, the proposed DREB approach excels at the MVO and the other benchmarks.

Additionally, we observed that the DRL algorithms induce a bias for some selective stocks. They learn to trade solely in those stocks and neglect others for the entire trading period. This phenomenon was perceived in both datasets. Having a bias in some stocks can be detrimental during stock crashes or any unfavorable news about the company. An ideal portfolio must be diversified enough to accumulate the maximum possible return while mitigating the downside risk. The proposed ensemble method assigns weights to the actions of all base agents and takes positions in every stock chosen by the union of base models. Thus, the proposed ensemble approach selects all stocks for trading and overcomes this limitation of taking biased trading actions.

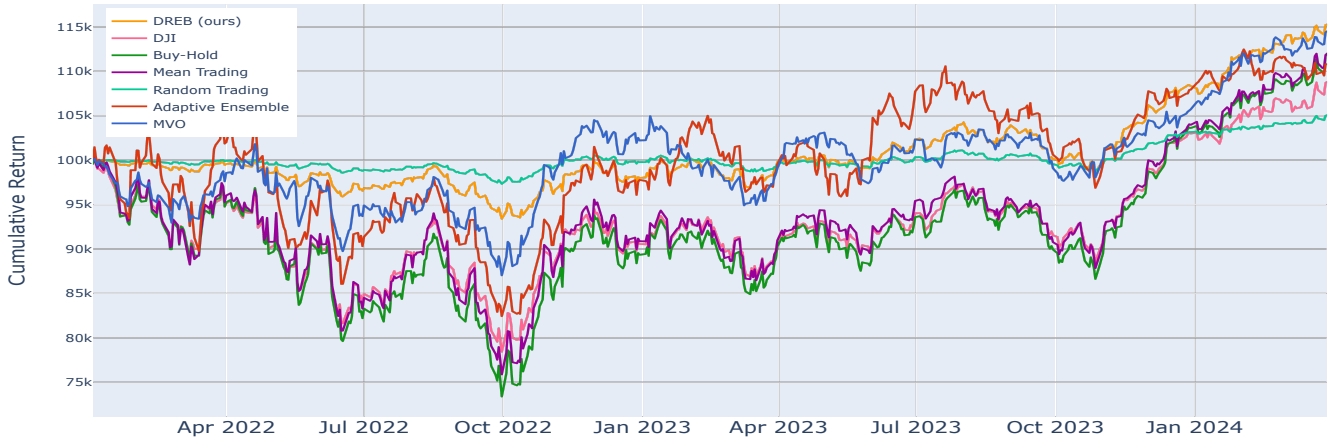

Figure 3: Cumulative returns of the proposed DREB model and the benchmark trading strategies for the DJI data over the entire trading period.

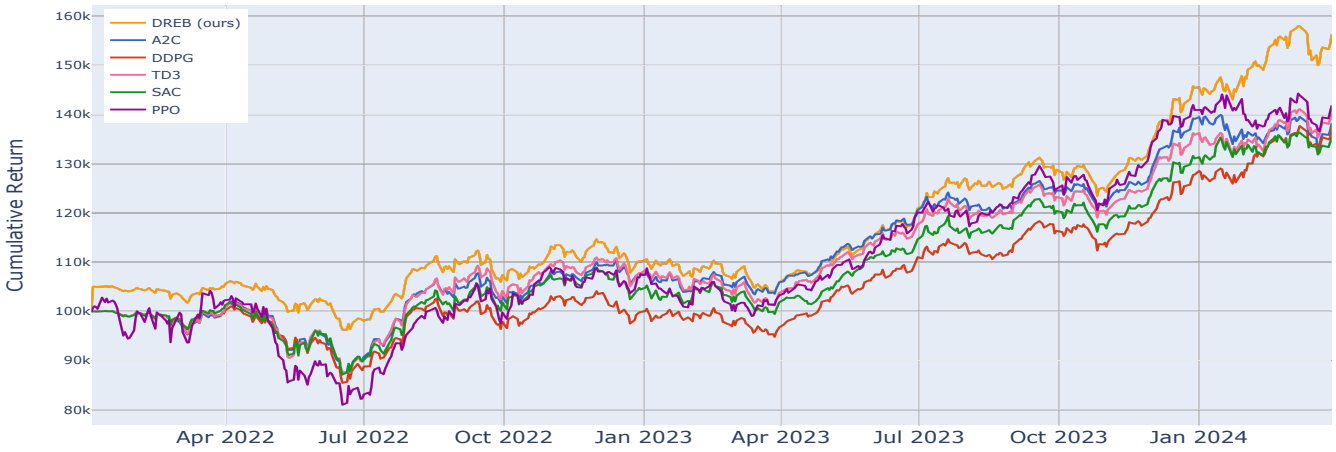

Figure 4: Cumulative returns of the proposed DREB model and the base DRL agents for the Sensex data over the entire trading period.

4 Conclusion

This paper introduces a dynamic reinforced ensemble method using the Bayesian optimization technique (DREB) for developing an automated trading strategy. The ensemble leverages diverse Deep Reinforcement learning (DRL) models, broadening the algorithmic input space. The proposed model demonstrates resilience to random specification errors in individual DRL models by utilizing dynamic time-varying weights assigned through Bayesian Optimization. The optimization problem is strategically transformed into a hyperparameter tuning challenge, efficiently addressed by a model-based Bayesian optimization algorithm. A comparative analysis considering the risk and return measures was conducted across two global stock market indices, Dow Jones from the U.S. and Sensex from India, against several benchmark trading strategies. A comprehensive simulation study confirmed the robustness and efficacy of our proposed methodology in optimizing stock trading strategies across varying market fluctuations. Additionally, our approach effectively overcame the limitation of individual DRL agents taking biased trading actions in some selective stocks.

Even though the current results are encouraging, we propose expanding the base DRL model space for future work to introduce large diversification for possible improvements. Also, incorporating advanced data-preprocessing techniques and additional technical indicators can further augment the performance of the proposed model. Exploring different reward functions like the Sharpe ratio or risk measures like CVaR offers chances to improve the trading system's stability.

References

[1] Ratnadip Adhikari and RK Agrawal. 2014. Performance evaluation of weights selection schemes for linear combination of multiple forecasts. *Artificial Intelligence Review* 42 (2014), 529–548.

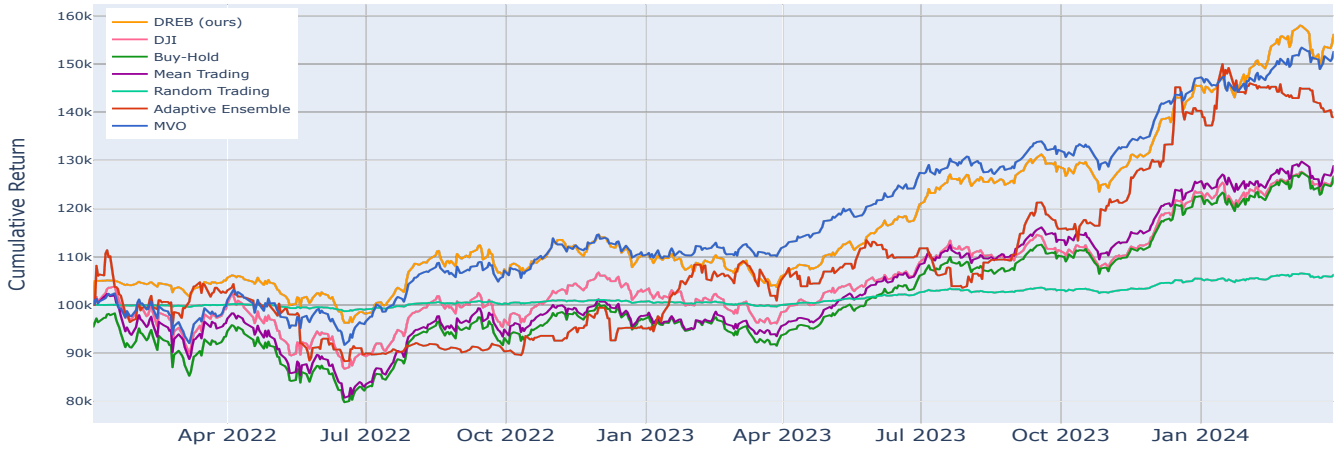

Figure 5: Cumulative returns of the proposed DREB model and the benchmark trading strategies for the Sensex data over the entire trading period.

[2] Loukia Avramelou, Paraskevi Nousi, Nikolaos Passalis, and Anastasios Tefas. 2024. Deep reinforcement learning for financial trading using multi-modal features. *Expert Systems with Applications* 238 (2024), 121849.

[3] Aryan Bhambu. 2023. Stock Market Prediction Using Deep Learning Techniques for Short and Long Horizon. In *Soft Computing for Problem Solving: Proceedings of the SocProS 2022*. Springer, 121–135.

[4] Aryan Bhambu, Ruobin Gao, and Ponnuthurai Nagaratnam Suganthan. 2024. Recurrent ensemble random vector functional link neural network for financial time series forecasting. *Applied Soft Computing* 161 (2024), 111759.

[5] Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Antonio Sanna. 2021. Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting. *Expert systems with applications* 164 (2021), 113820.

[6] Kai Chen, Yi Zhou, and Fangyan Dai. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)*. IEEE, 2823–2824.

[7] Gerda Claeskens, Jan R Magnus, Andrey L Vasnev, and Wendun Wang. 2016. The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting* 32, 3 (2016), 754–762.

[8] Liang Du, Ruobin Gao, Ponnuthurai Nagaratnam Suganthan, and David ZW Wang. 2022. Bayesian optimization based dynamic ensemble for time series forecasting. *Information Sciences* 591 (2022), 155–175.

[9] Thomas Fischer and Christopher Krauss. 2018. Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research* 270, 2 (2018), 654–669.

[10] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 1587–1596.

[11] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[12] Chien Yi Huang. 2018. Financial trading as a game: A deep reinforcement learning approach. *arXiv preprint arXiv:1807.02787* (2018).

[13] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. 2005. Forecasting stock market movement direction with support vector machine. *Computers & operations research* 32, 10 (2005), 2513–2522.

[14] Taylan Kabbani and Ekrem Duman. 2022. Deep reinforcement learning approach for trading automation in the stock market. *IEEE Access* 10 (2022), 93564–93574.

[15] Namyeong Lee and Jun Moon. 2023. Offline Reinforcement Learning for Automated Stock Trading. *IEEE Access* (2023).

[16] Yuxi Li. 2017. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274* (2017).

[17] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[18] Xiao-Yang Liu, Ziyi Xia, Jingyang Rui, Jiechao Gao, Hongyang Yang, Ming Zhu, Christina Dan Wang, Zhaoran Wang, and Jian Guo. 2022. FinRL-Meta: Market Environments and Benchmarks for Data-Driven Financial Reinforcement Learning. *NeurIPS* (2022).

[19] David W Lu. 2017. Agent inspired trading using recurrent reinforcement learning and lstm neural networks. *arXiv preprint arXiv:1707.07338* (2017).

[20] Harry M Markowitz. 1991. Foundations of portfolio theory. *The journal of finance* 46, 2 (1991), 469–477.

[21] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1928–1937.

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[23] John Moody and Matthew Saffell. 2001. Learning to trade via direct reinforcement. *IEEE transactions on neural Networks* 12, 4 (2001), 875–889.

[24] John J Murphy. 1999. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin.

[25] Ralph Neuneier. 1995. Optimal asset allocation using adaptive dynamic programming. *Advances in neural information processing systems* 8 (1995).

[26] Ralph Neuneier. 1997. Enhancing Q-learning for optimal asset allocation. *Advances in neural information processing systems* 10 (1997).

[27] Arishi Orra, Kartik Sahoo, and Himanshu Choudhary. 2023. Machine Learning-Based Hybrid Models for Trend Forecasting in Financial Instruments. In *Soft Computing for Problem Solving: Proceedings of the SocProS 2022*. Springer, 337–353.

[28] Jigar Patel, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications* 42, 1 (2015), 259–268.

[29] Martin Pelikan, David E Goldberg, Erick Cantú-Paz, et al. 1999. BOA: The Bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99*, Vol. 1. Citeseer.

[30] Antonio Riva, Lorenzo Bisi, Pierre Liotet, Luca Sabbioni, Edoardo Vittori, Marco Pinciroli, Michele Trapletti, and Marcello Restelli. 2022. Addressing non-stationarity in FX trading with online model selection of offline rl experts. In *Proceedings of the Third ACM International Conference on AI in Finance*. 394–402.

[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[32] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. 2017. Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*. IEEE, 1643–1647.

[33] Keyan Shen, Hui Qin, Jianzhong Zhou, and Guanjun Liu. 2022. Runoff probability prediction model based on natural Gradient boosting with tree-structured parzen estimator optimization. *Water* 14, 4 (2022), 545.

[34] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA.

[35] Manoj Thakur and Deepak Kumar. 2018. A hybrid financial trading support system using multi-category classifiers and random forest. *Applied Soft Computing* 67 (2018), 337–349.

[36] Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. 2020. Adaptive stock trading strategies with deep reinforcement learning

methods. *Information Sciences* 538 (2020), 142–158.

[37] Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. 2018. Practical deep reinforcement learning approach for stock trading. *arXiv preprint arXiv:1811.07522* (2018), 1–7.

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

[39] Xiaoming Yu, Wenjun Wu, Xingchuang Liao, and Yong Han. 2023. Dynamic stock-decision ensemble strategy based on deep reinforcement learning. *Applied Intelligence* 53, 2 (2023), 2452–2470.

[40] Zihao Zhang, Stefan Zohren, and Roberts Stephen. 2020. Deep reinforcement learning for trading. *The Journal of Financial Data Science* (2020).