Portfolio Optimization with Reinforcement Learning

By
Yinglei Xu

Senior Honors Thesis
Statistics & Operations Research
University of North Carolina at Chapel Hill

December 04, 2024

Approved:

Chuanshu Ji, Thesis Advisor

Abstract

This study explores the application of deep reinforcement learning (DRL) in portfolio optimization, addressing the limitations of traditional models in dynamic financial markets. Using algorithms like Advantage Actor-Critic (A2C), Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), and Twin Delayed Deep Deterministic Policy Gradient (TD3), we evaluate their ability to balance risk and returns. Through data processing, feature engineering, and advanced modeling, the study implements DRL agents to manage a portfolio of 30 stocks with low pairwise correlations. Results show SAC's impressive risk-adjusted returns and TD3's stable performance, though limitations such as constrained action space and tiny excess returns emphasize areas for further improvement. This work shows the potential of DRL in advancing portfolio optimization through adaptive, data-driven strategies.

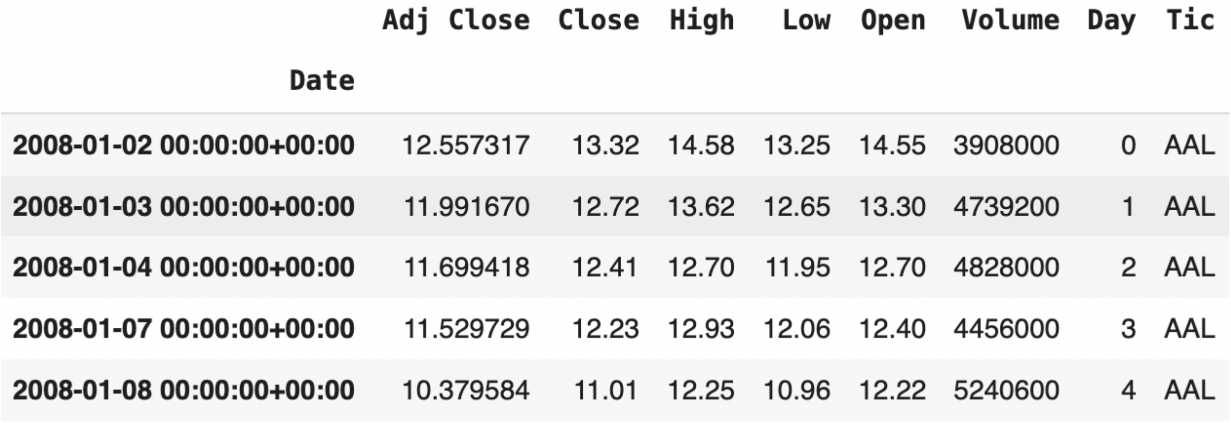
1 Introduction

Portfolio optimization is at the core of wealth management, yet traditional methods face limitations in today's volatile markets. Established models such as the Capital Asset Pricing Model (CAPM) and Modern Portfolio Theory (MPT) assume stable relationships between assets, linear dependencies, and efficient markets. The use of reinforcement learning (RL) in financial trading and portfolio optimization has become increasingly popular in recent years. Early research introduced creative ways to improve RL trading agents. The FinRL library by Liu et al. (2021) is a key development. FinRL is an open-source library built for automated stock trading. The FinRL provides customizable settings, including trading fees, market activity, and risk levels, allowing traders to create trading strategies with different RL models in an environment that mimics the real-world financial market.

2 Data

2.1 Background Data

This study uses historical stock data for all S&P 500 constituent stocks, including adjusted closing prices, closing prices, opening prices, highs, lows, volumes, and trading days. The dataset covers a period of over 16 years (January 1, 2008, to October 1, 2024) and was retrieved from the Yahoo Finance API. It provides a view of market activity with metrics such as trading volumes and day-by-day indicators for each stock, as shown in the sample data.



	Adj Close	Close	High	Low	Open	Volume	Day	Tic
Date								
2008-01-02 00:00:00+00:00	12.557317	13.32	14.58	13.25	14.55	3908000	0	AAL
2008-01-03 00:00:00+00:00	11.991670	12.72	13.62	12.65	13.30	4739200	1	AAL
2008-01-04 00:00:00+00:00	11.699418	12.41	12.70	11.95	12.70	4828000	2	AAL
2008-01-07 00:00:00+00:00	11.529729	12.23	12.93	12.06	12.40	4456000	3	AAL
2008-01-08 00:00:00+00:00	10.379584	11.01	12.25	10.96	12.22	5240600	4	AAL

Figure 1: Sample Data of S&P 500 Stock Metrics

2.2 Adjusted Closing Price

In this research, we focus on the adjusted closing price to exclude the impacts of sudden fluctuations in raw closing prices caused by dividends, stock splits, or other corporate actions.

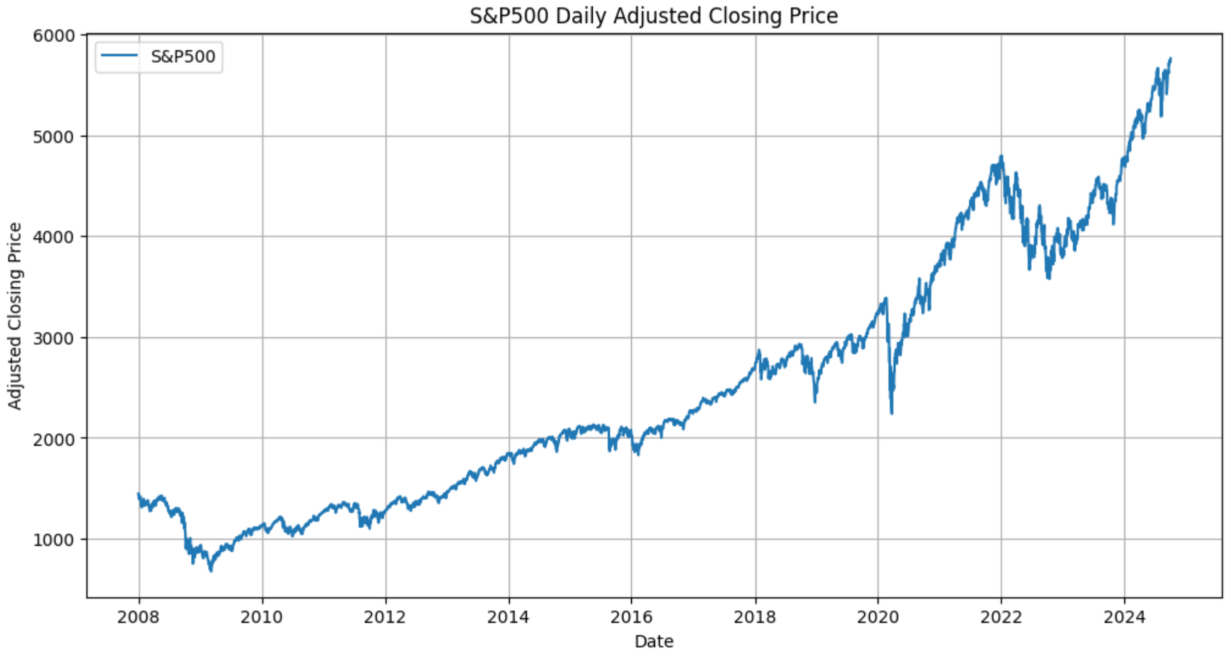


Figure 2: S&P 500 Adjusted Closing Price Over Time

2.3 Data Preprocessing

To prepare the dataset for experimental purposes and ensure its quality, the following data preprocessing steps were applied:

- **Imputation**: Missing values were removed to ensure the consistency and reliability of data. Stocks with prolonged trading suspensions or significant missing data were excluded from the dataset to avoid introducing bias into the analysis.

- **Daily Returns**:

$$\text{Daily Return}_t = \frac{P_t - P_{t-1}}{P_{t-1}} \times 100\%$$

  was calculated, where P_t is the adjusted closing price at day t. A shifting operation was applied to align daily returns with the corresponding trading days for consistency across the dataset.

- **Covariance Matrix**: Rolling covariance matrices of stock returns were calculated to capture inter-stock dependencies:

$$\Sigma_{i,j} = \mathrm{Cov}(R_i, R_j)$$

- **Feature Engineering**: Technical indicators listed in the below table were calculated for the next-step state space design purpose.

Indicator	**Description**
`macd`	Moving Average Convergence Divergence (MACD), a trend indicator.
`boll_ub`	Bollinger Upper Band, showing the upper price range boundary.
`boll_lb`	Bollinger Lower Band, showing the lower price range boundary.
`rsi_30`	30-period Relative Strength Index (RSI), a momentum indicator.
`cci_30`	30-period Commodity Channel Index (CCI), used for identifying trends.
`dx_30`	30-period Directional Index (DX), indicating trend strength.
`close_30_sma`	30-period Simple Moving Average (SMA), smoothing price fluctuations.
`close_60_sma`	60-period Simple Moving Average (SMA).

Table 1: Technical Indicators and Descriptions.

Table 1 summarizes the technical indicators. The MACD and SMA are referenced from Murphy (1999), Bollinger Bands from Bollinger (2001), RSI and DX from Wilder (1978), and CCI from Lambert (1980).

2.4 Data Splitting

As shown in Figure 3, the data is split into training and trading sets.

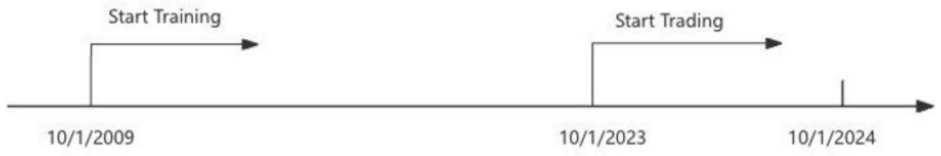


Figure 3: Data Splitting: The dataset is divided into two periods: a training period (10/1/2009 to 10/1/2023) and a trading/testing period (10/1/2023 to 10/1/2024).

3 Methodology

3.1 Pipeline

Our methodology follows the outlined pipeline. It begins with a stock selection process to identify a subset of 30 stocks with low pairwise daily return correlations. This subset forms the basis of the environment, which simulates the financial market by providing state information such as stock prices, technical indicators, and portfolio values. The reinforcement-learning agents, including PPO, A2C, TD3, and SAC, interact with the environment by performing actions—adjusting portfolio weights based on the observed states. The environment, in turn, evaluates these actions and calculates rewards based on the scaled portfolio value. The agents progressively refine their decision-making strategies to optimize portfolio performance through this iterative feedback loop.

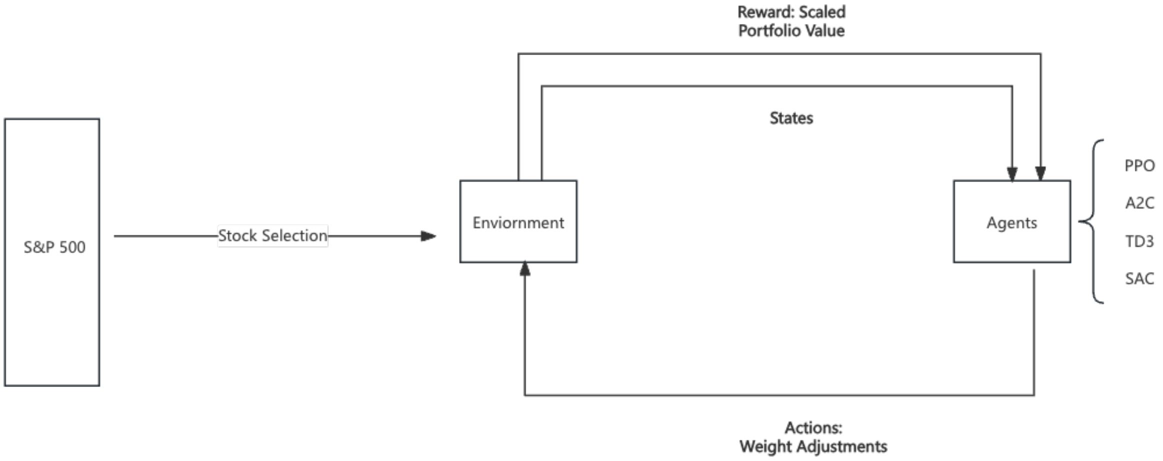


Figure 4: Reinforcement Learning-Based Portfolio Optimization Pipeline.

3.2 Stock Selection

Stock selection is important as it determines the pool of assets available for allocation. In the stock selection process, we explored the following two methods.

1. **Correlation Analysis (Primary Method)**: We selected stocks with the lowest pairwise correlations to ensure diversification. Pairwise correlations were calculated using:

$$\rho_{i,j} = \frac{\text{Cov}(R_i, R_j)}{\sigma_i \sigma_j}$$

where:

- $\text{Cov}(R_i, R_j)$: Covariance between stock i and stock j.
- σ_i: Standard deviation of returns for stock i.
- σ_j: Standard deviation of returns for stock j.

2. **Sharpe Ratio-Based Filtering(Alternative Method)**: We identified 30 stocks with high Sharpe ratios, prioritizing those with superior risk-adjusted returns. The Sharpe ratio formula is given by:

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

where:

- R_p: Portfolio return.
- R_f: Risk-free rate.
- σ_p: Standard deviation of portfolio returns.

While this method selects highly promising stocks, further analysis of their performance and implications is presented in the Discussion section.

3.2.1 Top 30 Stocks With the Least Correlated Daily Returns

Table 2 presents the 30 stocks with the lowest pairwise daily returns' correlations.

Tickers	Company Name	Industry
AAL	American Airlines Group Inc.	Airlines
ALK	Alaska Air Group, Inc.	Airlines
APA	APA Corporation	Oil & Gas E&P
CCL	Carnival Corporation	Hotels, Resorts & Cruise Lines
CLX	The Clorox Company	Household & Personal Products
CMA	Comerica Incorporated	Regional Banks
CMG	Chipotle Mexican Grill, Inc.	Restaurants
CNC	Centene Corporation	Health Care Providers & Services
CPB	Campbell Soup Company	Packaged Foods & Meats
DAL	Delta Air Lines, Inc.	Airlines
DLTR	Dollar Tree, Inc.	General Merchandise Stores
DVN	Devon Energy Corporation	Oil & Gas E&P
HBAN	Huntington Bancshares Incorporated	Regional Banks
JWN	Nordstrom, Inc.	Department Stores
KR	The Kroger Co.	Food & Staples Retailing
KSS	Kohl's Corporation	Department Stores
LUV	Southwest Airlines Co.	Airlines
M	Macy's, Inc.	Department Stores
MGM	MGM Resorts International	Hotels, Resorts & Cruise Lines
NEM	Newmont Corporation	Gold
NKTR	Nektar Therapeutics	Biotechnology
OXY	Occidental Petroleum Corporation	Oil & Gas E&P
PNC	The PNC Financial Services Group, Inc.	Diversified Banks
PVH	PVH Corp.	Apparel, Accessories & Luxury Goods
RF	Regions Financial Corporation	Regional Banks
SCHW	The Charles Schwab Corporation	Investment Banking & Brokerage
UAL	United Airlines Holdings, Inc.	Airlines
ULTA	Ulta Beauty, Inc.	Specialty Stores
WYNN	Wynn Resorts, Limited	Hotels, Resorts & Cruise Lines
ZION	Zions Bancorporation, N.A.	Regional Banks

Table 2: Selected Stocks and Their Industries

3.3 Correlation Heatmap

To validate the diversification, we generated a correlation heatmap (Figure 5) visualizing the pairwise correlations among the selected stocks' adjusted closing price.

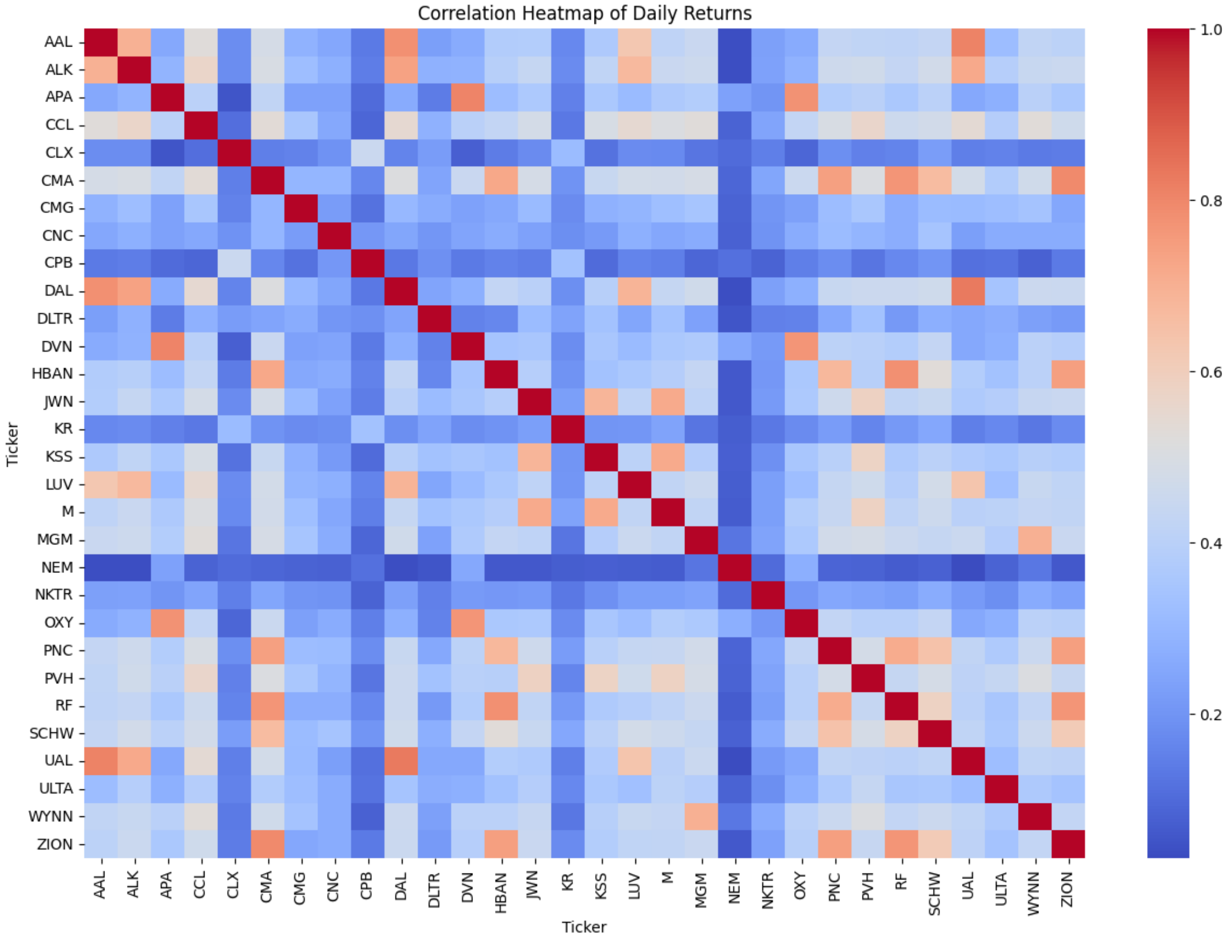


Figure 5: Correlation Heatmap of Selected Stocks

3.4 Environment

The environment is the system that interacts with the trading agents, responding to their actions and providing feedback through rewards. Following the stock selection process, the environment is narrowed down from the S&P 500 universe to 30 stocks with the lowest pairwise daily return correlations. We further modeled the environment using the OpenAI Gym framework, where it provides the state to the agent and receives actions (portfolio weight allocations) from the agents.

3.5 State Space

As Liu et al. (2021) discussed, the state space represents all the information from the environment that the agent uses to make decisions. At trading time t, the state space comprises two types of features: **short-term information** and **long-term information**.

- **Short-Term Information**: These features capture the current state of the portfolio and market conditions. The cash balance (b_t) indicates the available capital after each trade, while the number of shares owned (h_t) tracks the portfolio's current composition. Adjusted closing prices (p_t), along with daily opening, high, and low prices (o_t, h_t, l_t), provide insights into intraday price movements. Trading volume (v_t) reflects market liquidity, while technical indicators such as Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI) help identify trends and momentum in the short term.

- **Long-Term Information**: These features provide historical context to enhance decision-making. The covariance matrix (Σ) captures inter-stock relationships based on returns over a 252-day lookback period, while historical returns (R) represent daily returns for each stock over the same timeframe. These features allow the agent to incorporate patterns and relationships observed over the long term.

3.6 Reward Function

We set the scaled portfolio value as the reward function for the experiments. The raw portfolio value encourages the agent's learning objective with maximizing cumulative portfolio value over time. However, the raw portfolio value, often in the range of hundreds or thousands, can create significant challenges for reinforcement learning algorithms. Thus, we scale down the raw portfolio value by a factor of 10^{-4} . The final reward function involves the following transformation steps:

1. **Portfolio Value at t**: The portfolio value at day t is updated based on the portfolio value at the previous time step ($t-1$) and the portfolio return at t:

$$\text{Portfolio Value}_t = \text{Portfolio Value}_{t-1} \times (1 + \text{Portfolio Return}_t).$$

2. **Portfolio Return at t**: The portfolio return at time t is calculated as:

$$\text{Portfolio Return}_t = \sum_{i=1}^{n} w_i \cdot \frac{\text{Price}_{i,t} - \text{Price}_{i,t-1}}{\text{Price}_{i,t-1}},$$

   where:

   - n: The number of selected stocks in the portfolio, which is 30.
   - w_i: Weight of stock i in the portfolio, with $\sum_{i=1}^{n} w_i = 1$.
   - $\text{Price}_{i,t}$: Closing price of asset i at day t.
   - $\text{Price}_{i,t-1}$: Closing price of asset i at day $t-1$.

3. **Reward Calculation**: The reward is calculated using the updated portfolio value scaled by a predefined factor:

$$\text{Reward} = \text{Portfolio Value}_t \times \text{Reward Scaling},$$

   where the scaling factor is:

$$\text{Reward Scaling} = 10^{-4}.$$

3.7 Action Space

In the context of portfolio management, in a trading day t, action space corresponds to the weight allocation for each of the $n = 30$ selected stocks in the portfolio. Actions are continuous values between 0 and 1, representing the proportion of total capital allocated to each asset, ensuring that the sum of weights equals 1. To achieve this, the raw actions output by the agent, which consists of arbitrary continuous values, is transformed through a softmax transformation process. With the transformation, we linked the action space directly to its role in portfolio rebalancing.

1. **Raw Actions**: The agent initially outputs a vector of raw actions, $\mathbf{a} = [a_1, a_2, \ldots, a_n]$,

where $\mathbf{a} \in \mathbb{R}^n$ and $a_i \in \mathbb{R}$. These values represent unnormalized preferences for allocating weights to each stock.

2. **Softmax Transformation**: To convert the raw actions into valid portfolio weights, a softmax transformation is applied. The transformation is defined as:

$$\tilde{a}_i = \frac{e^{a_i}}{\sum_{j=1}^{n} e^{a_j}},$$

where:

- a_i: Raw action for stock i,
- $\tilde{a}_i$: Normalized weight for stock i,
- $n = 30$: Total number of selected stocks.

This transformation ensures that:

- Each weight $\tilde{a}_i$ lies within $[0, 1]$,
- The sum of all weights equals 1 :$\sum_{i=1}^{n} \tilde{a}_i = 1$.

3. **Normalized Actions (True Portfolio Weights)**: After applying the softmax transformation, the normalized actions $\tilde{\mathbf{a}} = [\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n]$ represent the proportion of total capital allocated to each stock. These weights are then used to rebalance the portfolio during each trading period.

By defining the action space in this manner, the agent's decisions remain constrained to valid portfolio allocations, aligning with practical financial requirements and promoting stable training in the reinforcement learning framework.

3.8 Agents

Agents are decision-making entities that interact with an environment based on the environment's feedback, the updated value of the rewards function. In this study, We implemented four deep reinforcement learning algorithms: Advantage Actor-Critic (A2C), Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), and Twin Delayed Deep Deterministic Policy Gradient (TD3) as the agents.

3.8.1 Advantage Actor-Critic (A2C)

A2C is a synchronous version of the Actor-Critic method that combines policy gradients with value function estimation (Mnih et al., 2016).

- **Policy Gradient:** The policy is updated by calculating the gradient of the policy, $\nabla_\theta \log \pi_\theta(a_t|s_t)A_t$. This allows the agent to learn to prioritize actions that result in higher rewards, as weighted by the advantage A_t.
- **Advantage Function:** The advantage function, $A_t = r_t + \gamma V(s_{t+1}) - V(s_t)$, measures how much better an action is compared to the average value of the current state. It consists of:
  - The immediate reward (r_t).
  - The discounted value of the next state $(\gamma V(s_{t+1}))$.
  - Subtracting the current state's value $(V(s_t))$, which stabilizes training.
  - Discount Factor (γ): γ determines how much importance is placed on future rewards. A larger value encourages the agent to focus on long-term returns, while a lower value focuses on current rewards.

3.8.2 Proximal Policy Optimization (PPO)

PPO improves upon traditional policy gradient methods by enhancing stability and efficiency through a clipped surrogate objective function (Schulman et al., 2017).

Clipped Surrogate Objective Function:

$$L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[\min\left(r_t(\theta)A_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)A_t\right)\right]$$

where:

- $\hat{\mathbb{E}}_t$: The average over a batch of timesteps t during training.

- $r_t(\theta)$: The ratio of probabilities between the new policy and the old policy:

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

- A_t: The advantage function. It measures how much better an action is compared to the average action in that state.

- ϵ: A small value that limits how much the new policy can change. Common values set at 0.1 or 0.2.

3.8.3 Soft Actor-Critic (SAC)

Soft Actor-Critic (SAC) is an off-policy reinforcement learning algorithm. Unlike the traditional reinforcement learning methods that focus more on maximizing expected rewards, SAC seeks a balance between expected rewards and entropy (Haarnoja et al., 2018).

Maximum Entropy Objective Function: The Soft Actor-Critic (SAC) algorithm optimizes a maximum entropy objective to balance reward maximization and policy exploration. The objective function is given as:

$$J(\pi) = \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{T} r(s_t,a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))\right],$$

where α is a hyper-parameter controlling the trade-off between the reward $r(s_t, a_t)$ and entropy $\mathcal{H}(\pi(\cdot|s_t))$, encouraging exploration.

Soft Q-Value Update: The **soft Q-value update** is computed as:

$$Q_\theta(s_t,a_t) = r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}\sim p}\left[V_\psi(s_{t+1})\right],$$

where $V_\psi(s_{t+1})$ is the soft value function. The algorithm employs two Q-functions, Q_{θ_1} and Q_{θ_2}, and uses the minimum of the two Q-values during updates.

Policy Update: The policy update is performed using the reparameterization trick:

$$\nabla_\phi J_\pi(\phi) = \nabla_\phi \log \pi_\phi(a_t|s_t) - \nabla_\phi Q_\theta(s_t, a_t),$$

where the policy optimizes the log-probability of actions while minimizing the Q-value of suboptimal actions.

3.8.4 Twin Delayed Deep Deterministic Policy Gradient (TD3)

TD3 addresses overestimation bias in Q-values, which occurs in DDPG (Deep Deterministic Policy Gradient) by using two critic networks and delayed policy updates. During training, the minimum value from the two critics is used for the target value. This method mitigates overestimated Q-values.(Fujimoto et al., 2018).

Critic Update: The target for the critic is computed as:

$$y = r(s_t, a_t) + \gamma \min_{i=1,2} Q'_{\theta_i}(s_{t+1}, \pi_{\phi'}(s_{t+1}) + \epsilon),$$

where:

- y: The target Q-value used for critic updates.
- $r(s_t, a_t)$: The reward received at time t.
- γ: The discount factor, which determines the importance of future rewards.
- Q'_{θ_i}: The target Q-function for critic network i.
- $\pi_{\phi'}$: The target policy network.
- ϵ: Small noise added to the action to promote exploration.

Actor Update: The actor is updated to maximize the expected Q-value for the current policy:

$$\nabla_\phi J \approx \frac{1}{N} \sum_i \nabla_a Q_{\theta_1}(s_i, a)|_{a=\pi_\phi(s_i)} \nabla_\phi \pi_\phi(s_i),$$

where $\nabla_\phi J$ represents the gradient of the objective function for the actor with respect to its parameters.

4 Experiment

4.1 Trading Constraints

We applied several trading constraints to our experiments to better mimic real financial market conditions. These constraints include:

- **Turbulence Index:** To account for market instability, we incorporated the turbulence index when training our trading models. According to Kritzman and Li (2010), the turbulence index is defined as:

  $$d_t = (\mathbf{y}_t - \boldsymbol{\mu})\, \boldsymbol{\Sigma}^{-1}\, (\mathbf{y}_t - \boldsymbol{\mu})'$$

  Within this turbulence index formula, $\boldsymbol{d}_t$ represents the turbulence at time t, which is a scalar value. $\mathbf{y}_t$ is the vector of asset returns for time t, having dimensions $1 \times n$. $\boldsymbol{\mu}$ denotes the sample mean vector of historical returns, also of size $1 \times n$. Finally, $\boldsymbol{\Sigma}$ refers to the sample covariance matrix of historical returns with dimensions $n \times n$. This index penalized trades executed during periods of heightened market turbulence, encouraging the agents to avoid risky allocations in unstable conditions.

- **Transaction Cost:** We also account for the transaction cost, set at 0.1% (one per thousand), applied to each trade during portfolio rebalancing in our experiments. This ensures the model reflects real-world trading conditions, where transaction fees are a significant factor in profitability.

4.2 Hyperparameter Learning Results

We conduct hyperparameter tuning using grid search. The optimal hyperparameters are:

Model	Learning Rate	Batch Size	Discount Factor	Other Parameters
A2C	0.001	128	0.99	N/A
PPO	0.0003	256	0.99	$\epsilon = 0.2$
SAC	0.0003	256	0.99	$\alpha = 0.2$
TD3	0.001	256	0.99	$\epsilon = 0.2$

Table 3: Optimal Hyperparameters for RL Models

5 Results

5.1 Performance Metrics

We compare the performance of the RL models with a baseline buy-and-hold strategy, where the weight for each stock is set at $\frac{1}{30}$, as the portfolio consists of 30 stocks.

Metric	A2C	PPO	SAC	TD3	Baseline
Cumulative Return (%)	28.41	27.07	30.84	**31.07**	29.90
Annual Return (%)	28.54	27.19	30.99	**31.21**	30.17
Sharpe Ratio	1.47	1.35	**1.52**	1.46	1.47
Max Drawdown (%)	**12.25**	12.95	12.69	12.29	12.55

Table 4: Performance Metrics for RL Models vs. Baseline

5.2 Asset Growth Over Time

The following figures show the asset growth over time for each RL model compared to the buy-and-hold baseline.

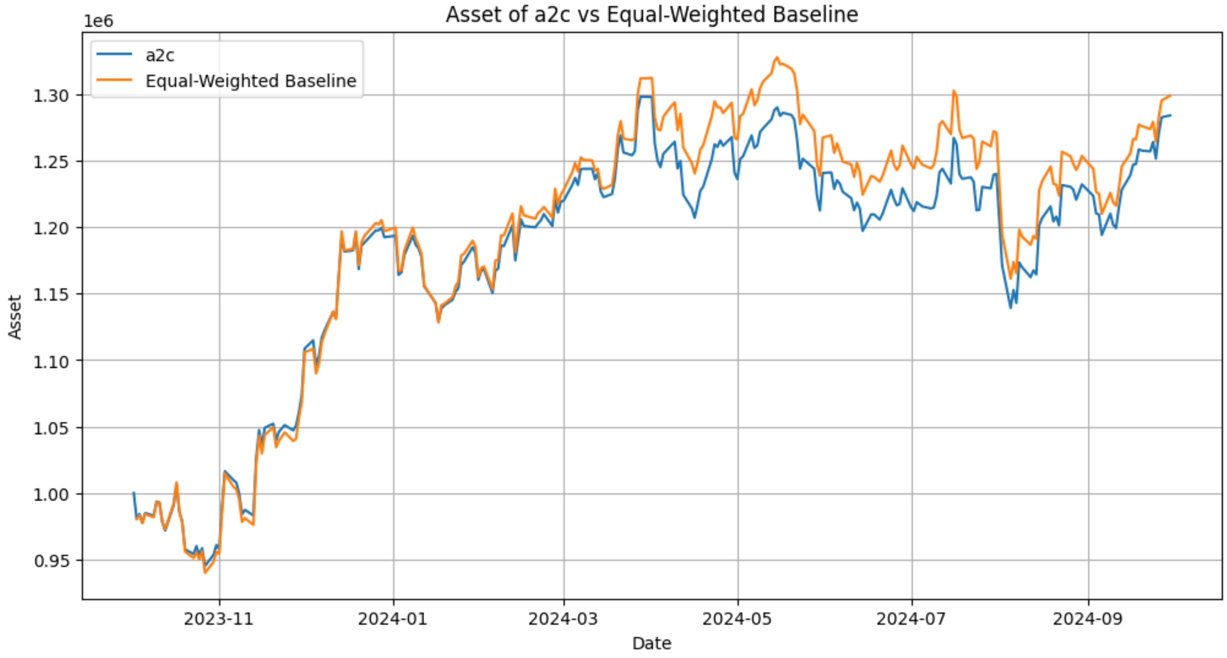


Figure 6: Asset Growth: A2C vs. Baseline

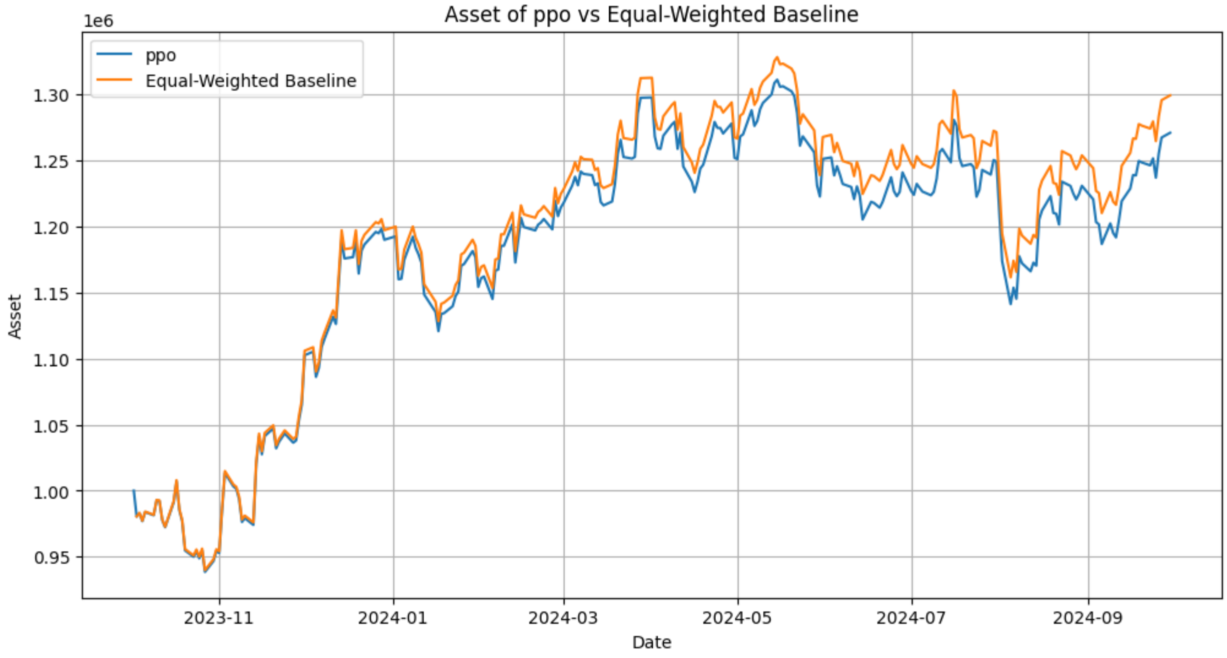


Figure 7: Asset Growth: PPO vs. Baseline

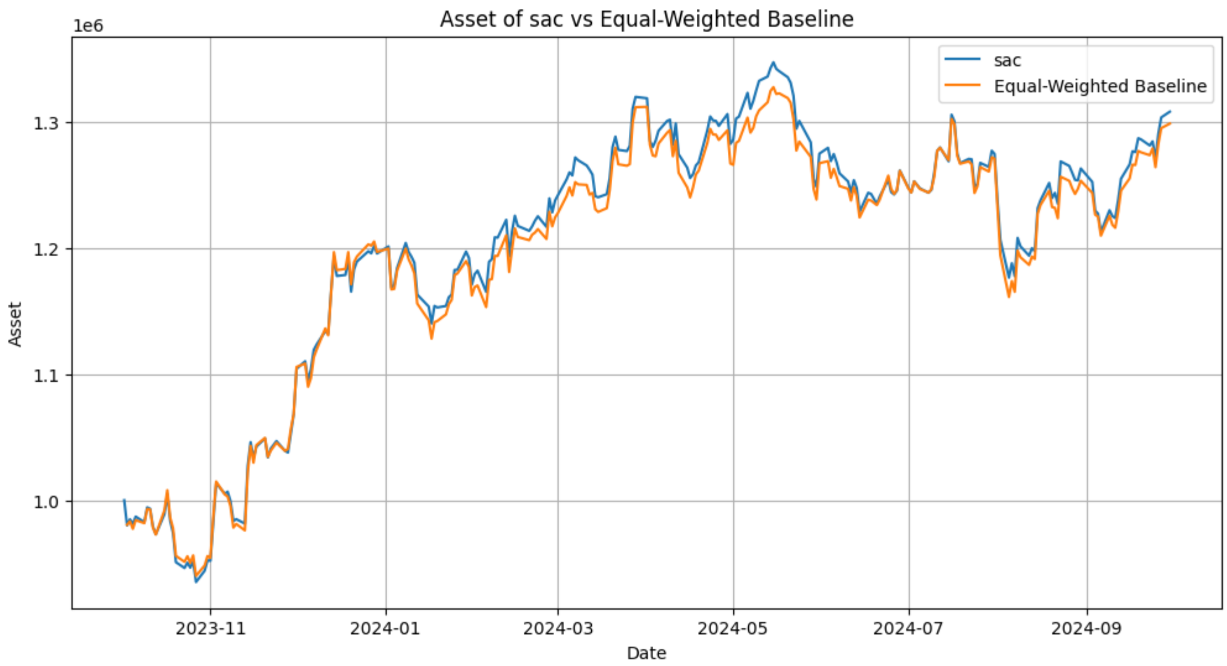


Figure 8: Asset Growth: SAC vs. Baseline

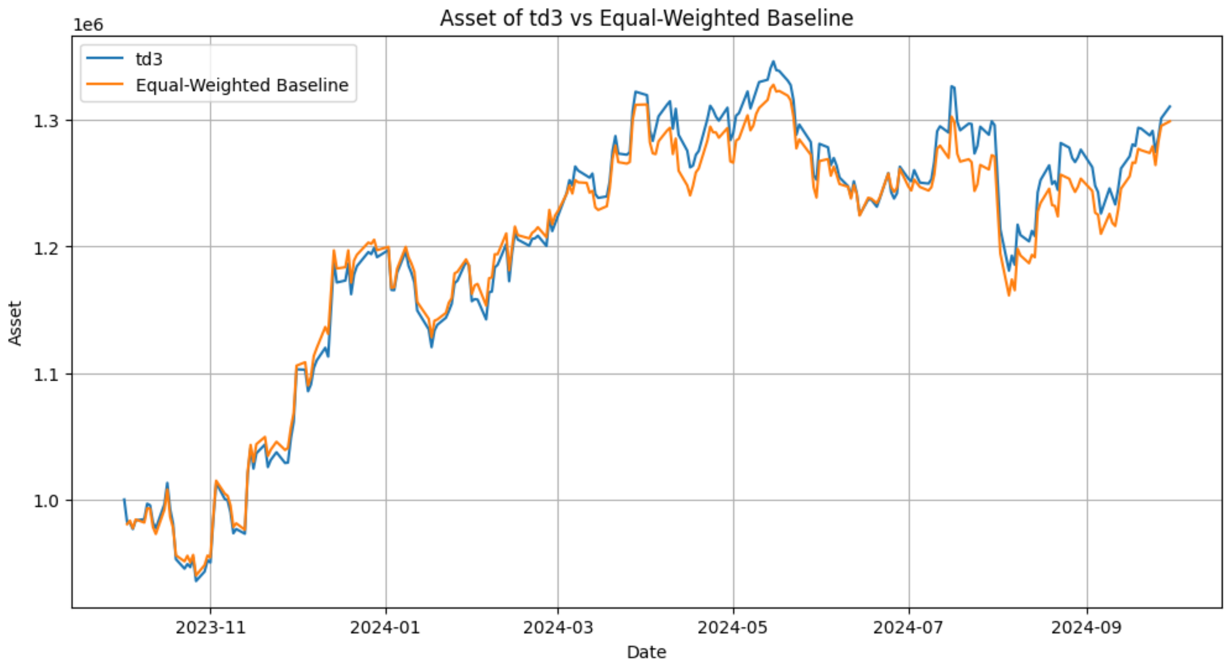


Figure 9: Asset Growth: TD3 vs. Baseline

6 Discussion

6.1 Analysis of Results

Returns Performance TD3 shows a strong performance in encouraging the portfolio value growth, achieving the highest cumulative return (31.07%) and annual returns (31.21%). SAC closely follows with slightly lower returns, benefiting from its entropy-regularized policy that balances exploration and exploitation. A2C and PPO underperform, likely due to their more conservative policy updates and clipping mechanisms, which may limit adaptability to dynamic market conditions. The baseline strategy achieves a cumulative return of 29.90% and an annual return of 30.17%, performing competitively but lacking the flexibility to adapt to changing market conditions.

Risk-Adjusted Returns SAC has the highest Sharpe Ratio (1.52), indicating that it can generate consistent returns with relatively low volatility. TD3 and A2C follow closely, maintaining a balance between returns and risk, while PPO lags with the lowest Sharpe Ratio (1.35), suggesting less efficient risk management. The baseline strategy achieves a Sharpe Ratio of 1.47, showing a decent balance between return and risk but falling short of the top-performing SAC.

Risk Management A2C shows the smallest max drawdown (12.25%), suggesting better resilience during market downturns, which is valuable for risk-averse strategies. TD3 and SAC also show competitive drawdowns, reflecting effective risk control while delivering strong returns. The baseline strategy records a max drawdown of 12.55%, slightly better than PPO (12.95%) but higher than the RL models with stronger risk management capabilities. PPO, with the largest drawdown, indicates its limitations in handling turbulent environments.

6.2 Sharpe Ratio-Based Selection

We also experiment with Sharpe Ratio-based filtering as a stock selection method. Table 5 lists the top 30 stocks ranked by their Sharpe ratios.

Stock Ticker	**Sharpe Ratio Rank**
AAPL	1
REGN	2
MA	3
TJX	4
ORLY	5
BKNG	6
CTAS	7
ROP	8
AZO	9
ODFL	10
COR	11
DPZ	12
CMG	13
UNH	14
TYL	15
ROST	16
FICO	17
NFLX	18
ACN	19
CPRT	20
NVDA	21
INTU	22
SHW	23
AMZN	24
MSCI	25
APH	26
HD	27
BR	28
MCO	29
CDNS	30

Table 5: Top 30 Stocks Ranked by Sharpe Ratio

Metric	**A2C**	**PPO**	**SAC**	**TD3**	**Baseline**
Cumulative Return (%)	35.92	39.33	**41.99**	39.67	40.95
Annual Return (%)	36.09	39.51	**42.19**	39.86	41.34
Sharpe Ratio	2.74	2.71	2.80	2.77	**2.81**
Max Drawdown (%)	**6.31**	7.05	6.87	7.18	7.03

Table 6: Performance Metrics (Sharpe Ratio-Based Stock Selection)

As observed in Table 6, the Sharpe ratio-based portfolio shows strong performance due to its inherent focus on risk-adjusted returns. Even with a simple buy-and-hold strategy, the selected high-Sharpe-ratio stocks deliver impressive results. While reinforcement-learning-based strategies generally struggle to outperform this baseline, SAC achieves better cumulative and annual returns, showing its robustness even changing the reward function.

6.3 Sharpe Ratio as Reward Function

To explore further, we also implement the Sharpe Ratio as the reward function. We keep the original 30 selected stocks with the lowest daily return correlations during the stock selection phase to reduce overfitting and ensure a diverse portfolio.

However, as shown in Table 7, the baseline model consistently outperforms the reinforcement learning models across all evaluated metrics.

Metric	**A2C**	**PPO**	**SAC**	**TD3**	**Baseline**
Cumulative Return (%)	25.44	26.25	26.89	26.67	**29.90**
Annual Return (%)	25.55	26.37	27.01	26.79	**30.17**
Sharpe Ratio	1.30	1.33	1.41	1.31	**1.47**
Max Drawdown (%)	14.77	12.67	12.88	14.89	**12.55**

Table 7: Performance Metrics (Reward Function: Sharpe Ratio)

7 Limitations and Future Work

Limitations This study has several limitations. The action space is restricted to portfolio weights ranging from 0 to 1, which prevents the agent from performing short-selling. As a result, the trading strategy is limited to buying and holding stocks, reducing its flexibility and making it less reflective of real-world trading conditions. Additionally, while some improvement was observed compared to the baseline model, the excess return achieved by the reinforcement learning algorithms remains relatively small, highlighting the need for further optimization and refinement of the approach.

Future Work Future research could address these limitations by expanding the action space to include short-selling, thereby enabling the agent to mimic real-world trading strategies better. Furthermore, refining the reward function is important. We used scaled portfolio value as the reward function in our study, but ignoring the risk term. The reward function could incorporate maximum drawdown (MDD) as a risk term to better balance risk and return. A potential reward function is given as:

$$r_t = \text{Portfolio Return}_t - \lambda \cdot \text{MDD}_t,$$

where r_t is the adjusted reward at trading day t , Portfolio Return$_t$ represents the return at time t, MDD$_t$ is the maximum drawdown at time t, and λ is a hyperparameter controlling the trade-off between return and risk. By integrating such adjustments, reinforcement learning agents could achieve more stable and robust portfolio performance, particularly in volatile market conditions.

8 Conclusion

This study shows the potential of using DRL for portfolio optimization, with SAC and TD3 performing well. SAC stood out for its ability to balance exploration and exploitation, achieving the highest Sharpe ratio among the RL models. With proper Q-value estimation, TD3 showed strong and steady performance. However, the RL strategies had some challenges, such as only small gains compared to the baseline model. The action space was also limited, as it did not allow short-selling, reducing flexibility. Future work could focus on improving the reward functions to include risk measures like maximum drawdown and expanding the action space to better match real-world trading. Overall, this study highlights the potential of DRL to make adaptive decisions in portfolio optimization.

References

Bollinger, J. (2001). *Bollinger on Bollinger Bands*. McGraw-Hill.

Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1587–1596.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870.

Kritzman, M., & Li, Y. (2010). Skulls, Financial Turbulence, and Risk Management. *Financial Analysts Journal*, *66*(5), 30–41. Taylor & Francis. Retreived from https://ssrn.com/abstract=1691756.

Lambert, D. R. (1980). Commodity channel index: Tool for trading cyclic trends. *Commodities Magazine*. Reprinted from Commodities Magazine, 219 Parkade, Cedar Falls, IA 50613.

Liu, X.-Y., Yang, H., Chen, Q., Zhang, R., Yang, L., Xiao, B., & Wang, C. D. (2021). FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. *arXiv preprint arXiv:2011.09607*.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783*.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Wilder, J. W. (1978). *New concepts in technical trading systems*. Trend Research.