PORTFOLIO OPTIMIZATION USING REINFORCEMENT LEARNING AND EIGEN VESTING TECHNIQUE

Pinaki Das, Badal Raj Bijawat, Aditya Singla, Gargi Singh

February 14, 2022

School of Engineering

Jawaharlal Nehru University

under the guidance of

Dr. Sakshi Sharma

Atal Bihari Vajpayee School of Management & Entrepreneurship(ABVSME)

Jawaharlal Nehru University, New Delhi

in the partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

(a part of Five-Year Dual Degree Course)

Declaration

We declare that the project work entitled **"PORTFOLIO OPTIMISATION USING REINFORCEMENT LEARNING AND EIGEN VESTING TECHNIQUE**" which is submitted by us in partial fulfilment of the requirement for the award of degree B.Tech. (a part of Dual-Degree Programme) to School of Engineering, Jawaharlal Nehru University, Delhi comprises only our original work and due acknowledgement has been made in the text to all other material used.

Group Members Signatures

Pinaki Das

Aditya Singla

Badal Raj Bijawat

Gargi Singh

Certificate

This is to certify that the project work entitled "**PORTFOLIO OPTIMIZATION USING REINFORCEMENT LEARNING AND EIGEN VESTNIG TECHNIQUE**" being submitted by Mr. **Pinaki Das** (Enrolment No. **18/11/EC/026**), Mr. **Aditya Singla** (Enrolment No. **18/11/EC/038**), Ms. **Gargi Singh** (Enrolment No. **18/11/EC/030**) in fulfilment of the requirements for the award of the Bachelor of Technology (part of Five-Year Dual Degree Course) in Computer Science and Engineering, and Mr. **Badal Raj Bijawat** (Enrolment No. **18/11/EE/040**) in fulfilment of the requirements for the award of the Bachelor of Technology (part of Five-Year Dual Degree Course) in Electronics and Communication Engineering, will be carried out by them under my supervision. In my opinion, this work fulfils all the requirements of an Engineering Degree in respective streams as per the regulations of the School of Engineering, Jawaharlal Nehru University, New Delhi. This thesis does not contain any work, which has been previously submitted for the award of any other degree.

Signature of the Supervisor

Dr. Sakshi Sharma

Assistant Professor

Atal Bihari Vajpayee School of Management and Entrepreneurship(ABVSME)

Jawaharlal Nehru University, Delhi

Acknowledgement

Foremost, we would like to express our sincere gratitude to our project mentor Dr. Sakshi Sharma for her continuous guidance and support for our B.Tech Project work, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped us throughout the project. We would also like to thank Hrishidev Unni, PhD, SCIS for his constructive suggestions.

.

Group Members Signatures

Pinaki Das

Aditya Singla

Badal Raj Bijawat

Gargi Singh

Abstract

The advent of artificial intelligence in investing has revolutionised stock market trading. Owing to the non-stationary nature of financial time series and the influence of both exogenous and endogenous factors affecting the same, there is a need to develop comprehensive strategies taking help of artificial intelligence and machine learning. Coronavirus pandemic was an eye-opener in this regard as the artificial intelligence techniques so used have been able to give better results in the same period as compared to the conventional techniques such as Markowitz portfolio optimization and minimum variance portfolio. We put our algorithms to the test on the 28 stocks of Dow Jones Industrial Average Indices (DJIA) using various models of Deep Reinforcement Learning to test its effectiveness. The resulting portfolio is then compared and correlated with eigen vectors derived using Eigenvesting technique which is another method to optimise the portfolio. This leads to some important conclusions.

Contents

I INTRODUCTION and THESIS OVERVIEW 9

1 Introduction 9

2 Thesis Objective 10

II LITERATURE SURVEY 12

3 About Hurst Exponent 12

4 Momentum Based Trading Strategy 12

5 Deep Reinforcement Learning 12

6 Eigenvesting 13

III PROPOSED WORK AND METHODOLOGY 14

7 Using Relative Strength Index(RSI) Strategy 14

8 Trading Agent based on Deep Reinforcement Learning 15

8.1 Advantage Actor Critic (A2C) 16
8.2 Deep Deterministic Policy Gradient (DDPG) 16
8.3 Proximal Policy Optimization (PPO) 17
8.4 Ensemble Strategy 17

9 Analysis Based on Eigenvectors 18

IV RESULT DISCUSSION 19

10 Introduction 19

11 Stock Data Preprocessing 19

12 Hurst and Momentum 20

13 Ensemble Strategy and DRL Algorithms Performance 20

14 Eigenvesting 21

V CONCLUSION AND FUTURE SCOPE 24

15 Conclusion 24

16 Future Scope 24

List of Tables

1 Test to check for Data Stationarity 15
2 Comparision of Performance 20
3 Comparision between Traditional RSI Methods and Different Machine Learning Techniques 21
4 Working of Ensemble Strategy by Calculating the Sharpe Ratio 22

List of Figures

1 Overview of Reinforcement Learning based Stock Trading Strategy . 10
2 Proposed RSI Strategy 14
3 Model Flowchart for RSI Momentum Strategy 15
4 Comparisions between A2C, PPO, DDPG 19
5 Ensembled Strategy Comparisions with Min-Variance and Baseline DJIA 21
6 Weight Distribution of Eigenvectors(Time on X Axis) 22
7 Weight Distribution of Eigenvectors(Time on Y Axis) 22

Part I
INTRODUCTION and THESIS OVERVIEW

1 Introduction

Portfolio optimization is the process of selecting the best portfolio among all portfolios available, so that the returns on that portfolio are maximized and the risk is minimized in the given holding period. Modern portfolio theory (MPT) given by Harry Markowitz in 1951 gives us an initial understanding of optimizing portfolios[13]. MPT emphasizes the trade off between risk and returns to arrive at an efficient portfolio. The portfolio, constructed using efficient frontier, is well diversified, therefore, is agnostic to the extreme fluctuation in the market.

Minimum variance portfolio is another well known investing method. It tries to make a portfolio where the correlation between the constituents of the portfolio is very low. This method is robust and trustworthy. However, it is not immune to sudden fluctuations in the market. This strategy is hard and costly to implement since portfolio managers may want to adjust their selections at each time step and consider other factors such as transaction cost. More precisely we require a methodology which is self adjusting its strategy according to the changing behaviour of the market. Such behaviour of the financial market was observed recently during the coronavirus pandemic, therefore, it is important to develop methodologies which are able to steer a portfolio through such crises by avoiding a blood bath.

Based on the value of the Hurst exponent (explained in Part II) we solve such the problem either using Momentum-based Strategies or Mean-Reversion strategies.

Momentum Based strategies:

Investors that use momentum-based techniques strive to profit from the continuation of a current market trend. For example, during the last period of interest, a particular firm was performing well and its stock price was constantly growing. In this instance, investors may choose to gamble on the price continuing to climb and thereby take a long position. For a short position, do the opposite. Naturally, the methods are not so straightforward, and the majority of entry and exit decisions are based on a set of technical indications.

Mean reversion strategy: In finance, mean reversion implies that many phenomena of interest, such as asset prices and return volatility, eventually revert to long-term average levels. Many financial strategies, ranging from stock trading tactics to options pricing models, are based on the mean reversion principle. One of the most promising techniques in this field is the application of Deep Reinforcement Learning(DRL) in automating trading strategies. Deep reinforcement learning is the subfield of machine learning that combines reinforcement learning and deep learning. Reinforcement learning involves an agent which makes decisions by trial and error. In other words it is a process in which an agent approaches its goal by learning from its own previous experience. Deep learning is a type of machine learning that uses artificial neural networks to transform a set of inputs into a set of outputs. Deep learning methods, which often employ supervised learning with labelled datasets, have been shown to solve tasks involving complex, high-dimensional raw input data, such as images, with less manual feature engineering than previous methods. As a result, large datasets containing historical stock data can be elegantly handled by deep learning models. DRL methods are today widely used across financial markets to predict all kinds of metrics ranging from optimization to prediction.[1, 3]

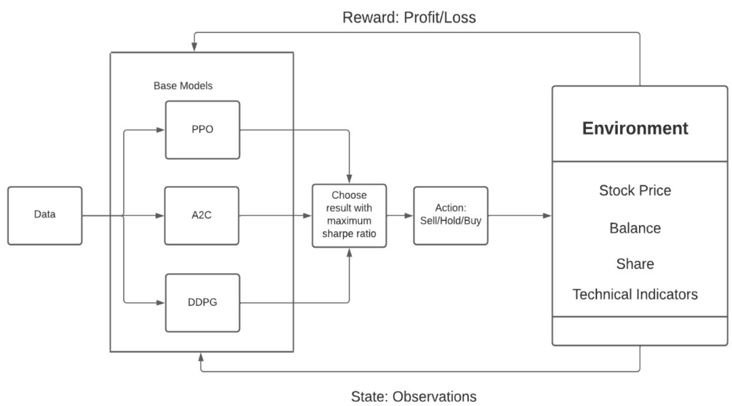

Figure 1: Overview of Reinforcement Learning based Stock Trading Strategy

In this report, we offer an ensemble technique for determining the best trading strategy in a complicated and dynamic stock market by combining three deep reinforcement learning algorithms. Proximal Policy Optimization (PPO), Advantage Actor Critic (A2C), and Deep Deterministic Policy Gradient (DDPG) are the three actor-critic methods of Deep Reinforcement Learning. Figure 1 depicts our deep reinforcement learning strategy. We improve the trading method's robustness and reliability by using the ensemble strategy. Our method can adapt to changing market conditions and maximise return while keeping risk to a minimum..The three algorithms that take actions in the environment are then trained in the second step. Third, we use an ensemble strategy to combine the three algorithms on the basis of a metric called sharpe ratio and observe that this strategy gives superior returns with respect to three algorithms individually.

2 Thesis Objective

The primary objective of this thesis is to

1. Differentiate between Short Term and Long Term Strategy
2. Integrating ML in traditional relative strength index (RSI) momentum strategy
3. Portfolio allocation through different DRL algorithms and an ensemble strategy.

4. Evaluating and Correlating the result of Ensemble Strategy with Eigenvesting to come to useful conclusions.

Part II
LITERATURE SURVEY

3 About Hurst Exponent

Given the characteristics (stylized facts) of financial data it is natural to consider their degree of predictability rather than the capabilities of a specific model. In relation to this, economists are divided into two main schools of thoughts. The group of those who believe that market data are random walks fulfilling the market efficiency hypothesis, thus financial data cannot be predicted . On the other hand there are those who accept that prices may move in trends and past prices can be used to forecast future price changes to some degree. Several authors have proposed the Hurst parameter H, as a measure of the stock market efficiency and predictability.

We use the Hurst exponent (H) as a measure for long-term memory of a time series,

- $H < 0.5$ — a mean-reverting (anti-persistent) series. The closer the value is to 0, the stronger the mean-reversion process is. In practice, it means that a high value is followed by a low value and vice-versa.
- $H = 0.5$ — a geometric random walk.
- $H > 0.5$ — a trending (persistent) series. The closer the value is to 1, the stronger the trend. In practice, it means that a high value is followed by a higher one .

4 Momentum Based Trading Strategy

Momentum investing is a trading strategy where we try to capitalize on existing market trends, by buying stocks when it looks like they are going to rise in price and to sell them before their price starts decreasing. To predict the future stock prices, traders look at various indicators, one of them is RSI (Relative Strength Index), which is a 14-day momentum oscillator and gives us an idea of whether the stock is overbought or oversold. For time series forecasting, models like ARIMA and NBEATS have long been used. ARIMA also known as autoregressive integrated moving average is a statistical model that is used to predict future values of a time series on the assumption that present and future values have had some residual effect from past values while NBEATS is a model that uses neural basis expansion for time series forecasting analysis.

5 Deep Reinforcement Learning

Recent applications of Deep reinforcement learning in trading use three approaches namely actor only, critic only or actor-critic approach[5].

Critic only approach, it is the most common approach to solve discrete action spaces problems. Some of the examples of its algorithms are DQN (Deep Q-Learning) and its further improvements. It is a value based approach so its main goal is to maximise the future reward given the current state. DQN works on estimating the Q value near to the expected Q value, it uses neural networks for performing this task. But as we know it is applicable only on discrete spaces it limits its application in stock trading as stock trading works in continuous space.

Actor approach, unlike the critic approach, is a policy based approach[4]. It works by using neural networks to optimise over a single policy in contrast with critic approach which uses neural networks to estimate the Q-value. A policy defines a strategy of how an action can be taken in a given state, using probability. In actor only approach we use Recurrent reinforcement learning to avoid the curse of dimensionality and improve trading efficiency. It is useful in continuous action spaces and consequently used in stock trading.

The most recently applied algorithm in trading scenario however is the actor-critic approach[2]. The working of this algorithm as the name suggests combines the working of both actor approach and critic approach. The actor generates an action using its policy based approach, then that action gets evaluated by critic network. Critic network sends a signal back to the actor network about how well the action is in the given state. Actor learns by that signal and improves that action. With the time critic network gets better in evaluating actions using value based approaches and actor gets better in generating action. The actor-critic approach has proved with the time that it can handle complex environments and produce good results. Game of doom[17] is one such example, which uses the application of actor-critic approach. Thus, the actor-critic approach is one of the best algorithms for stock trading.

6 Eigenvesting

The typical interpretation of the eigenvalue decomposition of a covariance matrix is this:

- The eigenvalues give the “variance” of each “factor” or eigenvector
- The variance associated with each factor is “uncorrelated” with the others (If we use the correlation matrix, this would be uncorrelated.)

Each eigenvalue corresponds to the risk of a portfolio and that the eigenvectors can represent an allocation of weights. So in this context, we can make the interpretations

- Eigenvectors are the “eigen portfolios”, strategy weight allocations which are uncorrelated to other eigen portfolios
- Eigenvalues are the “risk” of the given eigenportfolio

Part III
PROPOSED WORK AND METHODOLOGY

This part attempts to throw light on the different methodologies applied to our data set consisting of a portfolio of 28 stocks. First, we use RSI to evaluate the performance. Subsequently the Deep Reinforcement Learning Models are used to optimize the portfolio of 28 stocks of the Dow Jones Industrial Average(DJIA) index and the result is co related with vectors obtained using Eigenvesting.

7 Using Relative Strength Index(RSI) Strategy

Based on the value of Hurst Exponent, we use Momentum Based Strategies. RSI is a kind of momentum indicator that indicates whether the asset is oversold or overbought. A traditional way to use RSI in trading is to buy when the 14-period $RSI < 30$ and to sell when the 14-period $RSI > 70$. We propose to use the time series prediction to confirm that the RSI will rise within the next five days. The proposed strategy is represented in Figure 2. This gives us a more specific approach because along with the initial parameters we also factor in the 5-day prediction mean of RSI. The flow of the algorithm is mentioned in Figure 3.

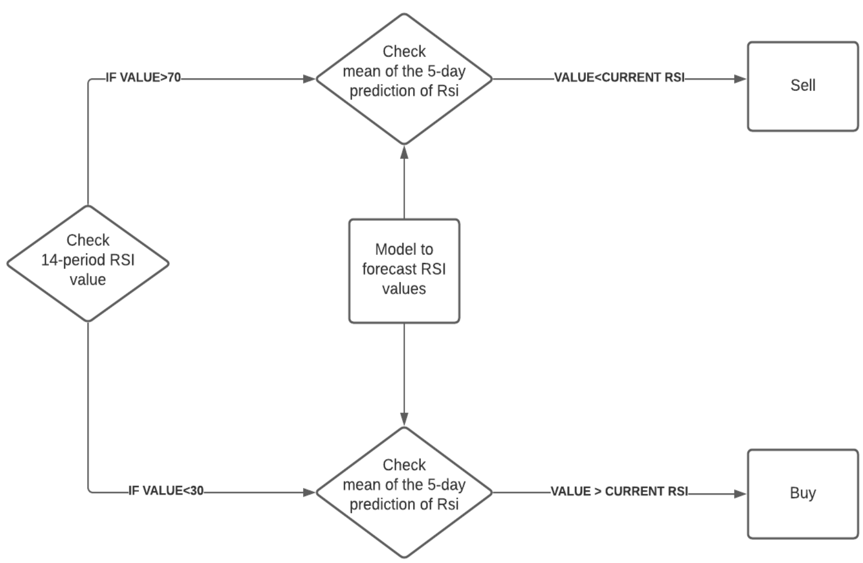

Figure 2: Proposed RSI Strategy

Making the model to predict values of RSI :

ADF Test Statistic	15.42540343675543
p-value	2.9964768475462023e-28
No. of. Lags Used	1
Number of Observations	6141

Table 1: Test to check for Data Stationarity

1. Checking whether the data is stationary: In order to find out whether our data is stationary or not, we ran the *Augmented-Dickey Fuller* test (ADF Test) on our data. On receiving a p value less than 0.05 as shown in Table 1 we came to the conclusion that our data is stationary.

2. Choosing the parameters for the model: After trying out a few standard models of ARIMA like ARIMA(1,1,1) and ARIMA(2,2,0), we used the pmdarima library to figure out the best model for our use-case. The ARIMA model selected was ARIMA(1,0,0).

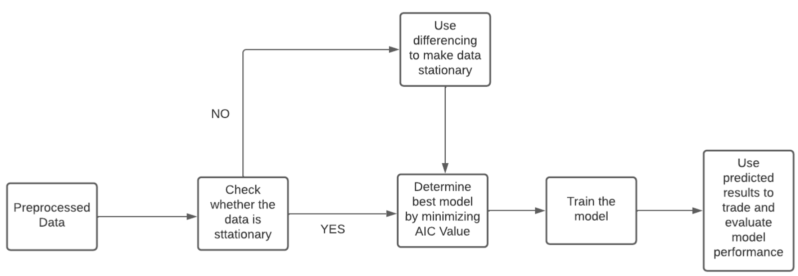

Figure 3: Model Flowchart for RSI Momentum Strategy

8 Trading Agent based on Deep Reinforcement Learning

Our aim is to find a robust deep reinforcement learning based strategy which gives the maximum returns on a portfolio. Therefore, while trading with help of a DRL agent, we propose to use an ensemble method which combines the different DRL agents and gives us a better understanding of optimizing the process of portfolio allocation. The three DRL methods used here are *Proximal Policy Optimization* (PPO), *Advantage Actor Critic* (A2C), and *Deep Deterministic Policy Gradient* (DDPG). The trading agent in these models is evaluated against the standards of *Dow Jones Industrial Average Index*(DJIA) which is a price weighted stock market index of 30 major companies listed in the United States, and classic min-variance portfolio. Metrics used to evaluate the performance include the sharpe ratio and volatility.

8.1 Advantage Actor Critic (A2C)

A2C algorithm that is Advantage Actor Critic as the name suggests consists of two parts: actor and critic. It has an advantage function which calculates the agent's prediction error. The actor network chooses an action at each time step by its policy based approach. Then the critic network uses its value based approach to evaluate the quality of the action taken by the actor network. The critic network predicts which states are good or bad and gives that signal back to the actor network to learn and increase the probability of good states and decrease the probability of bad states.

The objective function for A2C is:

$$\nabla J_\theta(\theta) = \mathbb{E}\left[\sum \nabla_\theta \log \pi_\theta (a_t|s_t) A(s_t, a_t)\right], \tag{1}$$

where $\pi_\theta (a_{t,}|s_t)$ is the policy network, $A(a_t, s_t)$is the Advantage Function which can be written as:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t), \tag{2}$$

or

$$A(s_t, a_t) = r(s_t, a_{t,} s_{t+1}) + \gamma V(s_{t+1}) - V(s_t) \tag{3}$$

8.2 Deep Deterministic Policy Gradient (DDPG)

The Q network and policy network is very much like simple Advantage Actor-Critic, but in DDPG, the Actor directly maps states to actions (the output of the network directly the output) instead of outputting the probability distribution across a discrete action space. The target networks are time-delayed copies of their original networks that slowly track the learned networks. Using these target value networks greatly improves stability in learning. At each time step, the DDPG agent performs an action a_t at s_t , receives a reward r_t and arrives at s_{t+1}. The transitions (s_t, a_t, s_{t+1}, r_t) are stored in the replay buffer R. A batch of N transitions are drawn from R and the Q-value y_i is updated as:

$$y_i = r_i + \gamma Q^{'}\left(s_{i+1}, \mu^{'}\left(s_{i+1}|\theta^{\mu^{'}}, \theta^{Q^{'}}\right)\right), i = 1, ..., N \tag{4}$$

The critic network is then updated by minimizing the loss function $L(\theta^Q)$ which is the expected difference between outputs of the target critic network $Q^{'}$and the critic network Q, i.e,

$$L(\theta^Q) = \mathbb{E}_{s_t, a_t, r_t, s_{t+1}} \sim \text{buffer}\left(y_i - Q\left(s_t, a_t|\theta^Q\right)\right) \tag{5}$$

DDPG is effective at handling continuous action space, and so it is appropriate for stock trading.

8.3 Proximal Policy Optimization (PPO)

PPO is used to manage the policy gradient update and guarantee that the new policy does not deviate too much from the old. By introducing a clipping component to the objective function, PPO attempts to simplify the goal of Trust Region Policy Optimization (TRPO).[15, 16]Let us assume the probability ratio between old and new policies is expressed as:

$$r_t(\theta) = \frac{\pi_\theta\left(a_t|s_t\right)}{\pi_{\theta_{old}}\left(a_t|s_t\right)} \tag{6}$$

The clipped surrogate objective function of PPO is:

$$J^{CLIP}(\theta) \widetilde{= \mathbb{E}_t}[min\left(r_t(\theta)\hat{A}(s_t, a_t)\right), clip\left(r_t\left(\theta\right), 1-\varepsilon, 1+\varepsilon\right)\hat{A}\left(s_t, a_t\right)] \tag{7}$$

where $r_t(\theta)\hat{A}(s_t, a_t)$ is the normal policy gradient objective, and $\hat{A}(s_t, a)_t$ is the estimated advantage function. The function $clip\left(r_t\left(\theta\right), 1-\varepsilon, 1+\varepsilon\right)$ clips the ratio $r(\theta)$ to be within $[1-\varepsilon, 1+\varepsilon]$. The objective function of PPO takes the minimum of the clipped and normal objective. PPO discourages large policy change moves outside of the clipped interval. Therefore, PPO improves the stability of the policy networks training by restricting the policy update at each training step. We select PPO for stock trading because it is stable, fast, and simpler to implement and tune.

8.4 Ensemble Strategy

Our goal is to predict the best algorithm among the three algorithms i.e. A2C, DDPG and PPO. Different stocks have different trends for different periods of a year. We use ensemble strategy to pick one of the algorithms that fits best according to the trend of the stock.

STEP 1. We take a window of 3 months of a year and apply all the three algorithms on our portfolio.

STEP 2. After applying each algorithm in a period of 3 months we pick the best performing among them on the basis of sharpe ratio. The sharpe ratio is calculated as:

$$\text{Sharpe Ratio} = \frac{\bar{r_p} - r_f}{\sigma_p} \tag{8}$$

where $\bar{r_p}$ is the expected portfolio return, r_f is the risk free rate, and σ_p is the portfolio standard deviation.

STEP 3. After the best performing algorithm is chosen, we apply it to predict and trade for the next quarter or a period of 3 months.

The idea of ensemble strategy is effective because each algorithm is sensitive to different stock trends in different periods. Hence, following a single algorithm for optimizing our portfolio will not give optimised results. So, we pick the best performing algorithm among them for different time periods of our portfolio and accumulate their returns.

9 Analysis Based on Eigenvectors

Based on historical returns, we find the eigenvectors representing the data. These eigenvectors correspond to different eigenvalues. It has been claimed that the portfolio allocation corresponding to the maximum eigenvector will have a maximum return. Portfolio allocation corresponding to minimum eigenvalue corresponds to minimum risk.

Since there are 28 stocks, therefore, we will get 28 linearly independent basis eigenvectors. Let us denote the eigenvectors as $V_1, V_2, \ldots\ldots\ldots, V_{27}, V_{28}$ which are linearly independent. Then any portfolio generated can be represented as a linear combination of these 28 eigenvectors. That is, $c_1V_1 + c_2V_2 + \ldots\ldots\ldots + c_{27}V_{27} + c_{28}V_{28}$ where $c_1, c_2, \ldots\ldots, c_{27}, c_{28}$ represent the coefficients of these eigenvectors which can be calculated using Reinforcement Learning. Based on the analysis of the coefficients and the distribution of eigenvectors we can further analyse the stock market and make situation specific portfolio allocation.

Part IV
RESULT DISCUSSION

10 Introduction

The performance evaluation of our proposed scheme is presented in this section. Table shows that our ensemble technique outperforms the three agents, the Dow Jones Industrial Average, and the classical min-variance portfolio allocation strategy in terms of Sharpe ratio.

11 Stock Data Preprocessing

Figure 4: Comparisions between A2C, PPO, DDPG

Twenty Eight stocks are selected from Dow Jones 30 index as on *2020-01-01* in our trading stock pool. We evaluate the performance of our DRL algorithms on the basis of the training with the help of the historical daily data from *2009-01-01* to *2021-10-31*. YAHOO FINANCE API was used to accumulate the historical stock data. Our dataset consists of two periods: IN-SAMPLE period and OUT-OF-SAMPLE period. In-sample period contains data for training stage and out-of-sample period data consists data for trading stage.

TRAINING PERIOD : *2009-01-01* to *2020-07-01*

TESTING PERIOD : *2020-01-01* to *2021-10-31*

	A2C	PPO	DDPG	DJIA(Baseline)	Ensemble	Min Variance
Cumulative Returns	0.33009	0.369228	0.36576	0.3918402	0.416702	0.279764
Annual Returns	0.237854	0.263509	0.260639	0.279047	0.261399	0.202566
Sharpe Ratio	1.645981	1.776876	1.726857	1.844560	1.442334	1.754163
Annual Volatilty	0.135233	0.136952	0.139828	0.139129	0.174200	0.108546
Max Drawdown	-0.087796	-0.091962	0.139828	-0.091962	-0.094090	-0.071326

Table 2: Comparision of Performance

We train three DRL agents employing PPO, A2C, and DDPG, respectively, in the training stage. Then, we analyse the profitability of each algorithm throughout the trading stage. Finally, we put our agent to the test on trading data, which is unseen out-of-sample data from *January 1, 2020* to *October 31, 2021.* The results of the same can be seen in Table 2 and it's plot in Figure 4.

Metrics used to evaluate them are:

- *Cumulative Return*: It is calculated by first finding the difference between the final value and the initial value of the portfolio, and then dividing it by the initial value.
- *Annualized Return*: Geometric average Amount of money made by the agent over a time period.
- *Annualized Volatility*: It is the annualized standard deviation of portfolio return
- *Max Drawdown*: It is the maximum observed loss from a peak to a trough of a portfolio, before a new peak is attained

12 Hurst and Momentum

According to our results in Table 3, we can see that the *ARIMA* model outperformed the *N-BEATS* model and the traditional RSI method by capturing more trade opportunities as observed from the hit-ratio and limiting the loss by resisting extreme dips.

13 Ensemble Strategy and DRL Algorithms Performance

Ensemble strategy, as mentioned in section, works to pick up the best performing algorithm in the time span of three months. It compares the sharpe ratio achieved by the three DRL algo(PPO, A2C, DDPG) in a holding period of three months and selects the model which has the highest sharpe ratio. Table 4 shows sharpe ratios of the DRL algorithms used and the Model picked up by the Ensemble strategy in the total holding period from *2020-01-02* to *2020-07-02.* Figure 5 compares the Ensemble strategy with min variance portfolio and DJIA portfolio which is taken as a baseline.

Ensemble Technique achieved a cumulative return of $41.6 percent$ according to Table 2 which is the best among all. It outperforms the A2C, DDPG, and PPO significantly. It also outperforms the Min Variance portfolio and the baseline DJIA portfolio.

Metric	Simple RSI	N-BEATS	ARIMA
Hit Ratio	41.3%	50%	53.75%
Net Profit	289.18	394.85%	460.61 $
Expectancy	-6.29$ per trade	3.93$ per trade	7.54$ per trade
Profit Return	0.75	0.92	0.83
Total Return	711.82$	1394.85$	1460.61$
Average Gain	46.3$ per trade	85.44$ per trade	89.43$ per trade
Average Loss	-43.3$ per trade	-93.3 $ per trade	-79.84$ per trade
Largest Gain	390.95$	517.32$	517.32$
Largest Loss	-188.85$	-209.69	-188$
Realised RR	1.07	0.92	1.07
Minimum	-291.27$	144.53$	142.21$
Maximum	307.61$	450.25$	387.97$

Table 3: Comparision between Traditional RSI Methods and Different Machine Learning Techniques

Figure 5: Ensembled Strategy Comparisions with Min-Variance and Baseline DJIA

14 Eigenvesting

Figure 6 shows the distribution of weights of each eigenvector with respect to the final answer as reached through the reinforcement Learning. It is interesting to note that *data-1*, the first principal eigenvector, has got the highest weight allocation continu-

Itr. No.	Start	End	Model Picked Up	A2C	PPO	DDPG
1	2020-01-02	2020-04-02	PPO	-0.385534	**-0.331629**	-0.418476
2	2020-04-02	2020-07-02	DDPG	0.183964	0.241241	**0.252796**
3	2020-07-02	2020-10-01	A2C	**0.319927**	0.201723	0.17129
4	2020-10-01	2020-12-31	DDPG	0.087727	0.110573	**0.269406**
5	2020-12-31	2021-04-05	PPO	0.145914	**0.244942**	0.226651
6	2021-04-05	2021-07-02	A2C	**0.123671**	0.031212	0.067533

Table 4: Working of Ensemble Strategy by Calculating the Sharpe Ratio

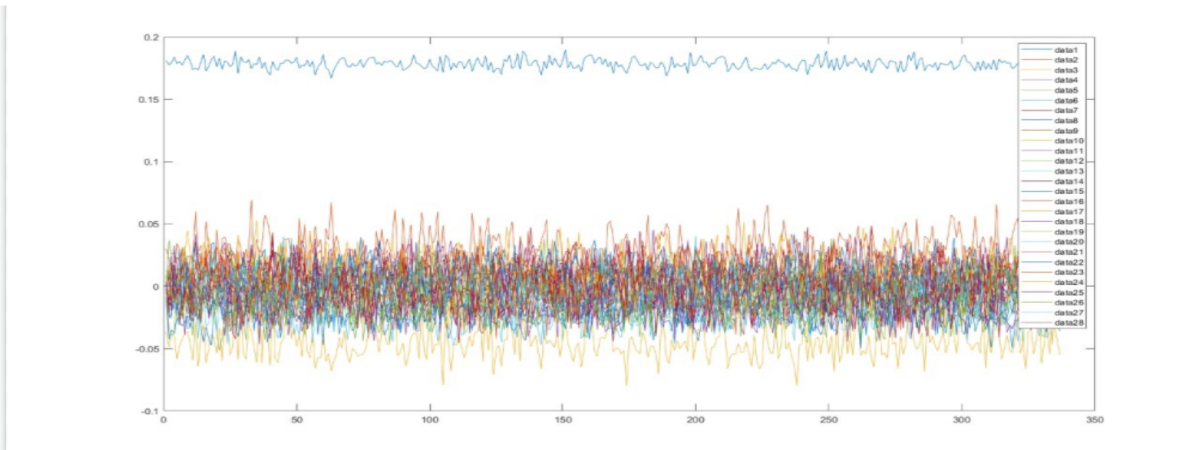

Figure 6: Weight Distribution of Eigenvectors(Time on X Axis)

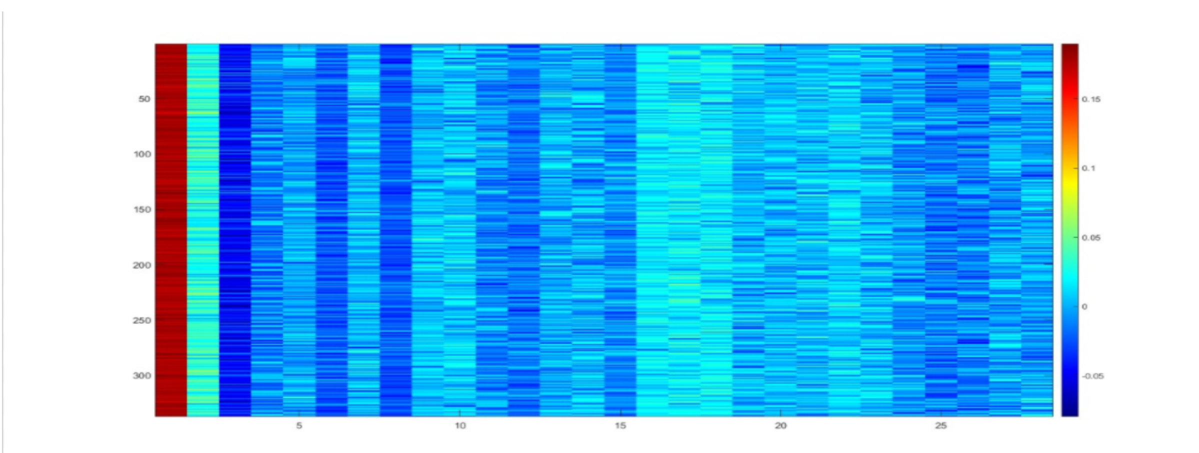

Figure 7: Weight Distribution of Eigenvectors(Time on Y Axis)

ously which is in agreement with our initial hypothesis.

In Figure 7 each column represents 1 eigenvector and each row represents 1 timestep. It shows a similar result, that is the first principal eigenvector has highest weight allocation and it is interesting to note that the third eigenvector has negative weight allocation for the whole time period.

Part V
CONCLUSION AND FUTURE SCOPE

15 Conclusion

- Hurst exponent values give us an accurate understanding of the time series under consideration.
- If the Hurst Exponent is high, then using RSI trading strategy with ARIMA forecasting gives the best result.
- In case of long term investments, using Reinforcement Learning gives best results which are in conformity with the eigenvesting technique, that is, maximum weight allocation to eigenvectors corresponding to the maximum eigenvalue.
- The training dataset used coincided with the covid period. However, the portfolio still generated greater return in comparison to the conventional techniques such as minimum variance portfolio.
- The results will be beneficial for the policy makers and emphasizes the need of computational methods in the financial markets.

16 Future Scope

The ensemble strategy used can be applied to more diverse portfolios which will contain not just stocks but also bonds and free cash. Future research can take place on how to offset the influence of external features like transaction costs levied during trading of stocks. The reinforcement learning agents can be made more robust and trustworthy by continuously feeding them cleaned and real time data.

References

[1] Qian Chen and Xiao-Yang Liu. 2020. Quantifying ESG alpha using scholar big data: An automated machine learning approach. ACM International Conference on AI in Finance, ICAIF 2020 (2020)

[2] Stelios Bekiros. 2010. Heterogeneous trading strategies with adaptive fuzzy Actor-Critic reinforcement learning: A behavioral approach. Journal of Economic Dynamics and Control 34 (06 2010), 1153–1170

[3] Yunzhe Fang, Xiao-Yang Liu, and Hongyang Yang. 2019. Practical machine learning approach to capture the scholar data driven alpha in AI industry. In 2019 IEEE International Conference on Big Data (Big Data) Special Session on Intelligent Data Mining. 2230–2239

[4] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. 2016. Deep direct reinforcement learning for financial signal representation and trading. IEEE Transactions on Neural Networks and Learning Systems 28 (02 2016), 1–12

[5] Thomas G. Fischer. 2018. Reinforcement learning in financial markets - a survey. FAU Discussion Papers in Economics 12/2018. Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics

[6] Jinke Li, Ruonan Rao, and Jun Shi. 2018. Learning to Trade with Deep Actor Critic Methods. 2018 11th International Symposium on Computational Intelligence and Design (ISCID) 02 (2018), 66–71

[7] Vijay Konda and John Tsitsiklis. 2001. Actor-critic algorithms. Society for Industrial and Applied Mathematics 42 (04 2001)

[8] Mark Kritzman and Yuanzhen Li. 2010. Skulls, financial turbulence, and risk management. Financial Analysts Journal 66 (10 2010)

[9] Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, and Xiao-Yang Liu. 2019. DP-LSTM: Differential privacy-inspired LSTM for stock prediction using financial news. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019) Workshop on Robust AI in Financial Services: Data, Fairness, Explainability, Trustworthiness, and Privacy, December 2019 (12 2019)

[10] Zhipeng Liang, Kangkang Jiang, Hao Chen, Junhao Zhu, and Yanran Li. 2018. Adversarial deep reinforcement learning in portfolio management. arXiv: Portfolio Management (2018)

[11] Timothy Lillicrap, Jonathan Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. International Conference on Learning Representations (ICLR) 2016 (09 2015)

[12] Mansoor Maitah, Petr Procházka, Michal Čermák, and Karel Šrédl. 2016. Commodity Channel index: evaluation of trading rule of agricultural Commodities. International Journal of Economics and Financial Issues 6 (03 2016), 176–178

[13] Harry Markowitz. 1952. Portfolio selection. Journal of Finance 7, 1 (1952), 77–91

[14] Volodymyr Mnih, Adrià Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. arXiv:1707.06347 (07 2017)

[16] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. arXiv:1707.06347 (07 2017)

[17] Yuxin Wu and Yuandong Tian. 2017. Training agent for first-person shooter game with actor-critic curriculum learning. In International Conference on Learning Representations (ICLR), 2017.