EPiC Series in Computing

Volume 89, 2022, Pages 41–51

Proceedings of 35th International Conference on Computer Applications in Industry and Engineering

Deep Reinforcement Learning for Portfolio Management

Y. Ma[1], Z. Liu[2] and C. McAllister[2]

[1] Northern Illinois University, DeKalb, U.S.A.

[2] Southeast Missouri State University, Cape Girardeau, U.S.A.

yuema@niu.edu, zliu@semo.edu, cdmcallister@semo.edu

Abstract

This paper discussed how to build deep reinforcement learning (DRL) agents to determine the allocation of money for assets in a portfolio so that the maximum return can be gained. The policy gradient method from reinforcement learning and convolutional neural network/recurrent neural network/convolutional neural network concatenated with the recurrent neural network from deep learning are combined together to build the agents. With the proposed models, three types of portfolios are tested: stocks portfolio which has a positive influence due to the Covid-19, stocks portfolio which has a negative influence due to the Covid-19, and portfolio of stocks combined with cryptocurrency which are randomly selected. The performance of our DRL agents was compared with that of equal-weighted agent and all the money fully invested on one stock agents. All of our DRL agents showed the best performance on the randomly selected portfolio, which has an overall stable up-ticking trend. In addition, the performance of linear regression model was also tested with the random selected portfolio, and it shows a poor result compared to other agents.

1 Introduction

The advantages that deep reinforcement learning demonstrate in the field like gaming has inspired more and more researchers to apply this framework to other fields, for example, financial market, and expect to achieve the same success. In financial market, portfolio management is about the allocation of financial assets meeting the financial goals of investors. The similar goal to achieve the maximized rewards in deep reinforcement learning has drawn attention from researchers to investigate different deep reinforcement learning methods to manage financial portfolio.

(Liang et al, 2021, 2018) implemented various deep reinforcement learning algorithms including Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO) and Policy Gradient (PG) in portfolio management. (Zhang et al, 2020) proposed a cost-sensitive portfolio selection method with deep reinforcement learning, where a novel two-stream portfolio policy

network was devised to extract both price series patterns and asset correlations. The proposed method also includes a new cost-sensitive reward function to maximize the accumulated return as well as to constrain both costs via reinforcement learning. (Nitin Kanwar, 2019) applied model free reinforcement learning algorithms which directly estimates the optimal policy or value function through policy iteration or value iteration, with emphasis on policy gradient and Actor Critic Methods to build the trading agent in order to maximize its overall return. (Jiang et al, 2017) then presented a model-free convolutional neural network with historic prices on a set of financial assets as its input, which outputs portfolio weights of the set. (Jiang, Xu et al, 2017) proposed portfolio vector machine (PVM) together with the policy network via convolutional neural network (CNN), recurrent neural network (RNN) and the Long Short-Term Memory (LSTM) to rebalance the cryptocurrency portfolio every 30 minutes. (Selim Amrouni et al, 2018) extended the work of Jiang et al, applied the PVM to trade the daily stock and cryptocurrency data with a daily rebalance.

This research was motivated by both Jiang et al and Selim Amrouni et al, and their work were adopted as reference for the basic policy network architecture of convolutional neural network. Recurrent neural network and mixed neural network (RNN+CNN) were experimented in this research to build agents and various financial asset combinations were used for testing. Since Covid-19, stock market has experienced the most volatile ups and downs. In the meanwhile, there is less related research taking the covid-19 into consideration regarding deep reinforcement learning on portfolio management. In order to find some insights on it, this research selected assets with different combinations to build the financial portfolio on testing the performance of deep reinforcement learning agents. To compare the performance of deep reinforcement learning with classic machine leaning approaches in portfolio management, linear regression was also tested for weights allocation in the research.

2 Methodology

2.1 Deep Learning Neural Networks

Convolutional neural network (CNN) and recurrent neural network (RNN) are two deep learning neural networks. Convolutional neural networks are designed to work with grid-structured inputs, which have strong spatial dependencies in local regions of the grid. Via convolutional layer, pooling layer and fully-connected layer, CNN reduces the dimensionality of a large number of parameters into a small number of parameters before processing. Certain data types such as time-series, text, and biological data contain sequential dependencies among the attributes. Key information about the relationships among the values of these mutually dependable data can be lost if they are treated independently of one another. To prevent information lost, recurrent neural network (RNN) can be used to model sequence data. RNN a class of neural networks which allow previous outputs to be used as inputs while having hidden states.

2.2 Deep Reinforcement Learning

Learning can be divided into supervised learning, unsupervised learning and reinforcement learning. Supervised learning models receive features (X) and labels (y) as training data, use them to get predicted values (y); while unsupervised learning models receive only features (X) and to do further analysis; but reinforcement learning can be different with these learning types. Reinforcement learning (RL) (Henderson et al, 2019) is the study of how an agent can interact with its environment to learn a policy which maximizes expected cumulative rewards for a task. Reinforcement learning methods can be classified as model-free reinforcement learning and model-based reinforcement learning. Model-based learning algorithm uses the transition function (and the reward function) to

estimate the optimal policy. Model-free learning algorithm estimates the optimal policy without using transition function and reward function of the environment. It estimates a "value function" or the "policy" directly from the interaction between the agent and environment, and it includes action-value fitting and policy gradient techniques.

2.3 Portfolio Management

Markowitz portfolio theory, which is also called Modern portfolio theory (MPT), is a theory on how to construct portfolios to maximize expected return based on a given level of market risk (The Investopedia Team, 2021). This paper considers stocks and cryptocurrency as the financial assets and uses their prices to calculate return. The expected rate of return for an individual investment is calculated as shown in the formula below:

$$E(R_{port}) = \sum_{i=1}^{n} w_i R_i$$

where:

w_i = the weight of an individual asset in the portfolio, or the percent of the portfolio in Asset i

R_i = the expected rate of return for Asset i

The expected rate of return for a portfolio of investments is simply the weighted average of the expected rates of return for the individual investments in the portfolio.

3 Research Methodology

3.1 Overall Architecture

In this research, a model building with deep reinforcement learning (DRL) agents was studied to determine the allocation of money for assets in a portfolio so that the maximum return can be gained. The policy gradient method from reinforcement learning and convolutional neural network/ recurrent neural network/ recurrent neural network concatenated with the convolutional neural network from deep learning are combined to build the agents. Figure 1 shows the overall architecture of the deep reinforcement learning agent. Data should be pre-processed based on different choices of neutral networks, and then the pre-processed data was passed to the neural networks. Softmax was applied in order to get the weight distribution of the assets. Input data, neural networks and the weight distribution formed the policy network, and reward was calculated based on the calculated weights. Reward was next fed into an objective function to update the policy network.

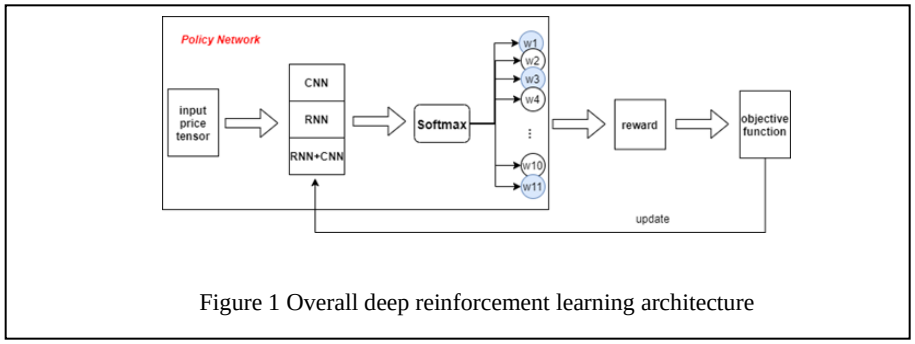

Figure 1 Overall deep reinforcement learning architecture

3.2 Data Collection and Preprocessing

In this research, both of stock and cryptocurrency data between January 25, 2016 to January 25, 2021 were downloaded and they span totally five years, 1258 days for stocks and 1828 days for cryptocurrencies. For the days which cryptocurrency trading occurred while the stock trading did not occur, the price information for cryptocurrency was removed. All the data were collected from yahoo Finance. Considering the impact of the volatility of covid-19 on the financial market, three combinations of stocks and cryptocurrency are selected to be tested: negatively influenced portfolio include stocks which were heavily impacted by the covid-19, positive influenced portfolio include stocks which have a positive influence because of the Covid-19, and randomly selected Portfolio which have neither the positive impact nor the negative impact of covid-19 and include seven random US stocks and three cryptocurrency.

The data is divided as below:
Total days: 1258
Training days: 60% of total days
Validation days: 20% of total days
Test days: 20% of total days

For CNN price tensor pre-processing, the input data are arranged into 3D in shape of (4,10,1258) as shown in figure 2. The first dimension 4 represents the number of features, and they are:

- Close(t-1)/Open(t-1)
- High(t-1)/Open(t-1)
- Low(t-1)/Open(t-1)
- Open(t)/Open(t-1)

Here the prices are normalized to the same scale. The second value for the dimension is 10, where it represents ten asset items (stocks+cryptocurrency) in the portfolio. The third value for the dimension is 1258, which represents the time steps (days). 1258 was sliced to 50 days each before passing to the network.

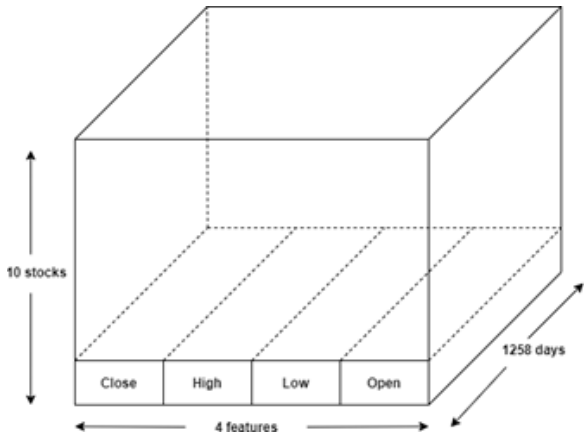

Figure 2 CNN data pre-processing

For RNN and RNN+CNN price tensor pre-processing, there are only one feature (the mean value for the open price, close price, high price and the low price.) is used. And the mean prices are passed to MinMaxScaler which helps to make all the value in the same scale. After data scaling, data was grouped to 50 days each. In the previous section's CNN policy network, the length of tensor was set to be 50 days. In order to do a comparative analysis with the CNN policy network, the tensor should be processed as the same length for batch processing. After transposing and dimension expanding, pre-processed input data has the shape (25,1,10,50), which is illustrated in figure 3. The diagram shows that there are 25 batches of (1,10,50), for each step, one batch (1,10,50) from (25,1,10,50) was passed to the policy network to calculate the weight distribution. 1 means there is only one feature– mean value, 10 represents 10 stocks, and 50 is the length of the tensor (50 days).

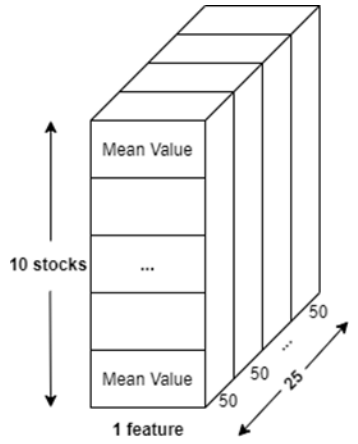

Figure 3 RNN and RNN+CNN data pre-processing

Data processing for linear regression is similar as for RNN. However, in linear regression, it needs X (features) and y (target values). In this research, the initial investment and mean value (open price, close price, high price and low price) of 10 stocks are chosen for X, which has totally 11 features. Initial investment is set to 10000 for every data time. y is the portfolio value which is randomly generated between values of (10000,13000). After setting the portfolio value for each data time, the initial investment of each day will be adjusted to the previous portfolio value. Figure 4 shows the shape of pre-processing data for linear regression model.

Money	Stock1	Stock2	...	Stock10	Portfolio Value
10000	12.3	25.6		28.8	12500
12500	12.9	22.6		15.8	12907
12907	17.8	11.1		15.9	11008
...	...	...		...	...

Figure 4 Linear regression data pre-processing

3.3 Reinforcement Learning Framework

In the studied reinforcement learning, the state of the agent is the input matrix X_t and previous portfolio weights at time t-1:

state = $S_t = (X_t, w_{t-1})$

The action of the agent is the weight vector of portfolio at time t:

action = $w_t = (w_1, ..., w_n)$

The reward function is defined such as it is the agent's return minus a baseline's return minus a term proportional to the maximum of the weight (this term is setup to make the agent avoid to invest fully in one stock), and baseline is an equal weighted agent which divides the investment to same proportion.

$$r_t = \sum w_i' y_i - \frac{1}{m} \sum y_i - \alpha \max(w_1, \dots, w_m)$$

where:

y_i is the return of single stock or cryptocurrency.

$\sum w_i' y_i$ is the sum of each stock or cryptocurrency's return. The formula for calculating the expected rates of return was used as reference.

m is number of items (stocks and cryptocurrency) in the portfolio.

α is the parameter of regularization, which can avoid to invest fully on one stock.

And this formula can be described like this:

AdjustedReward = Reward − BaselineReward − α ∗ MaximumPortfolioWeight

3.4 Network Structure

The convolutional neural network structure was designed by (Selim Amrouni et al, 2018). Motivated by the work in (Selim Amrouni et al, 2018), this research also studied recurrent network, and a mix-used network with RNN+CNN. The network structure design for the recurrent neural network and the mixed-used network was a new attempt for the training process.

The policy network which uses convolutional neural network comprises input layer and three convolution layers where the conv3 will output the last feature map, which concatenated with the cash bias. After the concatenation with the cash bias, the tensor got squeezed, and in the end, softmax function is used to get the probability distribution which is also the weight distribution of the portfolio. The convolutional neural network model is shown in figure 5.

The output of the network is then used to calculate the adjusted reward. At first it needs to calculate the instantaneous reward as shown in the formula below:

$$InstantaneousReward = \frac{PortfolioVaoue - PreviousPortfolioValue}{PreviousPortfolioValue}$$

In the formula, the trading cost is also taken into consideration when the portfolio value is calculated.

As for the objective function, since there is no maximize function to maximize the reward, and minimizing the negative original value will be same as maximize the original value. Hence, minimize (-adjusted reward) was chosen to maximize the reward over the batch.

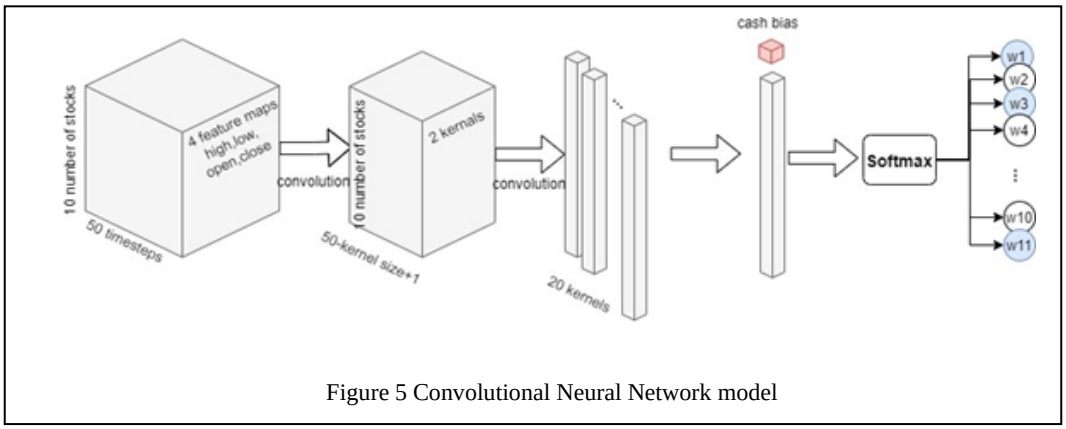

Figure 5 Convolutional Neural Network model

For the recurrent neural network model has two recurrent layers, for the first recurrent layer, the number of recurrent units was set to 20, and it came from the inspiration from figure 5's CNN of filters=20. For the second recurrent layer, the number of the recurrent units was set to 1. The output dimension after the two recurrent layers was expanded to 4D. And after that it can be concatenated with a 4D-shape cash-bias. The model is shown in figure 6.

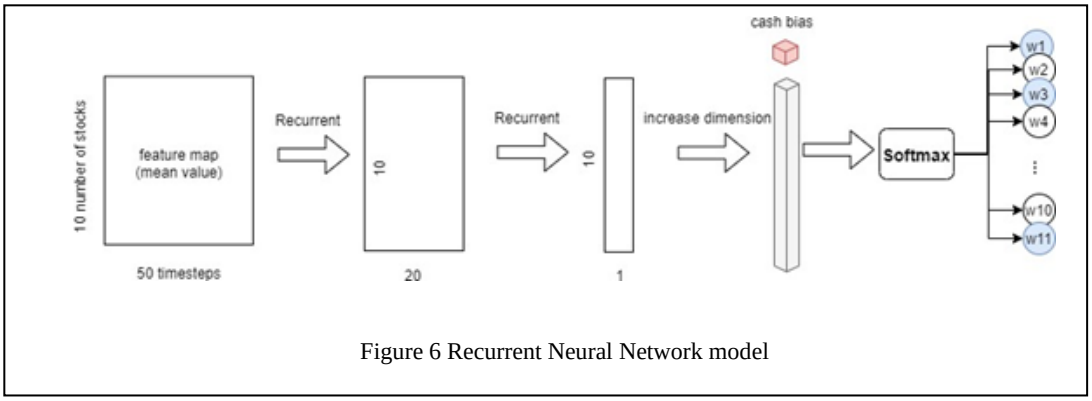

Figure 6 Recurrent Neural Network model

The mixed neural network model combines both the convolution layers and recurrent layers when processing the price tensor. The input tensor has the same shape as the one of the recurrent neural network. First, it was passed to the recurrent layer, which has 20 recurrent units. The output of the recurrent layer's dimension is then expanded in order to be processed by the convolutional neural network. For the convolution layer, there is one filter. After the dimension expansion, all the following steps are same with the convolution neural network's. The mixed neural network structure is shown in figure 7.

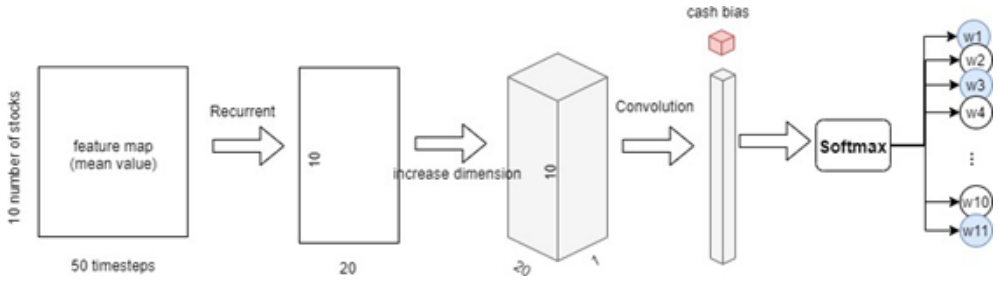

Figure 7 Mixed Neural Network model

In linear regression model, the training data is passed to the standard scaler first to be processed into same scale. Next, the processed data passes into the linear regression model in order to get the coefficient list which is also the weight list. The accumulated value for each value in the portfolio weight list should be 1, so the proportional value is calculated. The linear regression model is shown in figure 8.

weight_list = [|coef1|/sum_coef, |coef2|/sum_coef, ... |coef11|/sum_coef]

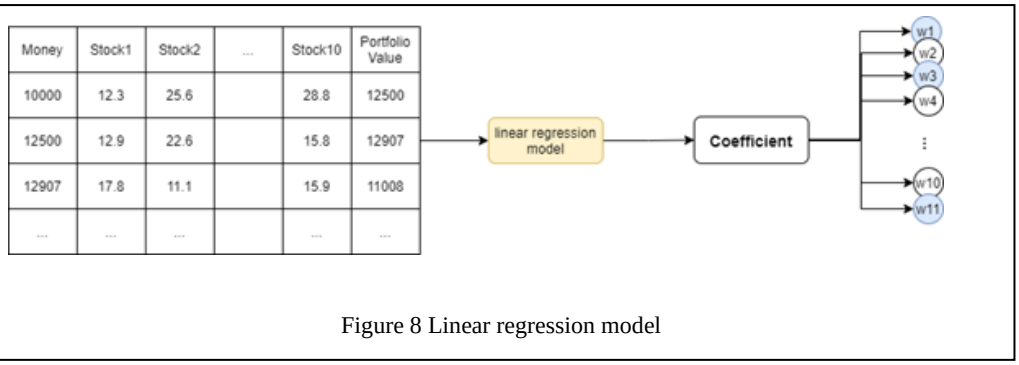

Figure 8 Linear regression model

4 Results

Three different types of portfolio and three different networks were tested. For linear regression, the random selected portfolio was tested. The portfolio value evolutions during testing are presented For the testing result figures which presented in the next subsections, except three deep reinforcement learning agents, there are other financial portfolio management agents are tested. Equal-weighted agent which is an agent equally allocated weights for all the financial assets. Initial investment is the 10000 dollars which acts as a baseline to show a clear performance comparison. And for the other full stock AAL, DAL, LUV..., which means the money fully invest on the single stock.

Figure 9 shows the negatively influenced portfolio value evolution for 253 days. For both three types of deep reinforcement learning agents, there exist certain periods in which portfolio value is

greater than the initial investment. But overall, they do not show an increasing trend. Among the three models, RNN model has the best performance (most return) at the end of the testing day, however it still loses 45.58% initial investment.

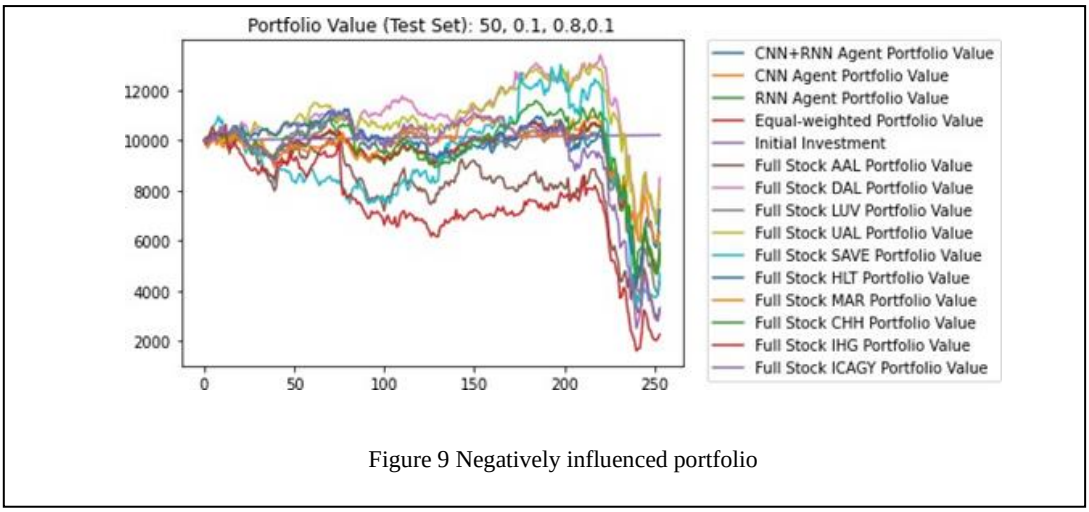

Figure 9 Negatively influenced portfolio

Figure 10 shows the positively influenced portfolio value evolution for 253 days. Due to the explosive increasing trend for the full stock on BABA portfolio value, the graph did not show a clear trend for the other agents. Actually, for the three types of deep reinforcement learning agents, more than half part of testing days, the portfolio value seldomly exceeds the initial investment, but at the end of testing days, all the agents shows an increasing trend. For CNN model which has the best performance (most return) at the end of testing days, it gains 110.87% initial investment.

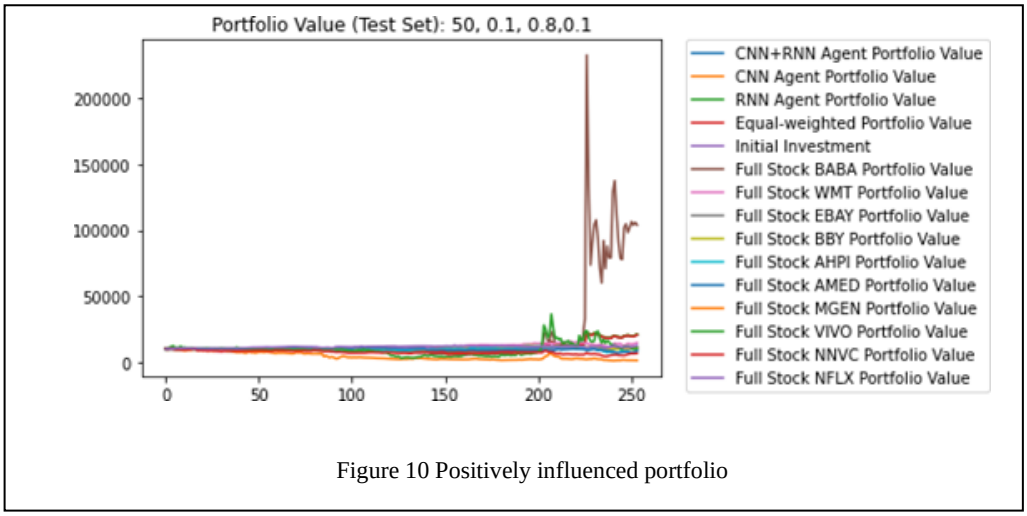

Figure 10 Positively influenced portfolio

Figure 11 shows the randomly selected portfolio value evolution for 253 days. For the three types of deep reinforcement learning agents, where a stable increasing trend shows. The deep reinforcement learning agents have the best performance on this type of portfolio compared to the other two portfolios. RNN model which has the best performance (most return) gained 47.72% initial investment during 253 days.

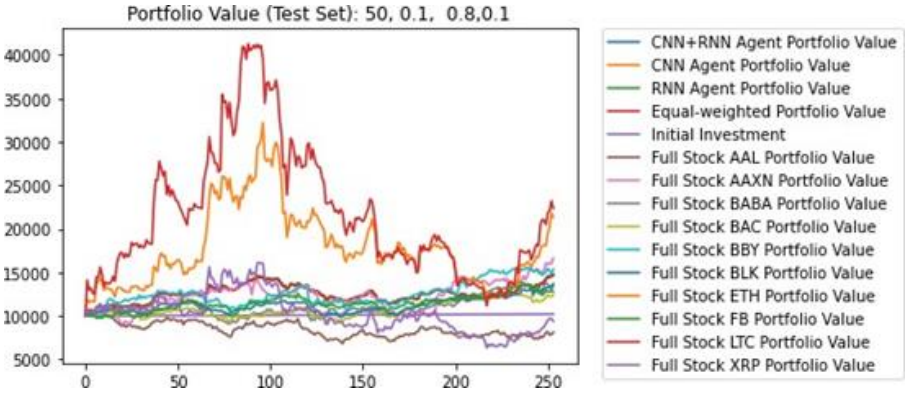

Figure 11 Randomly selected portfolio

Since randomly selected portfolio shows a stable increasing performance, we tested linear regression using the randomly selected portfolio. The linear regression agent's result is shown in figure 12 and is compared with the equal-weighted portfolio value and the initial investment. From the comparison, it can be seen that linear regression model shows a bad performance which has a clear downward trend.

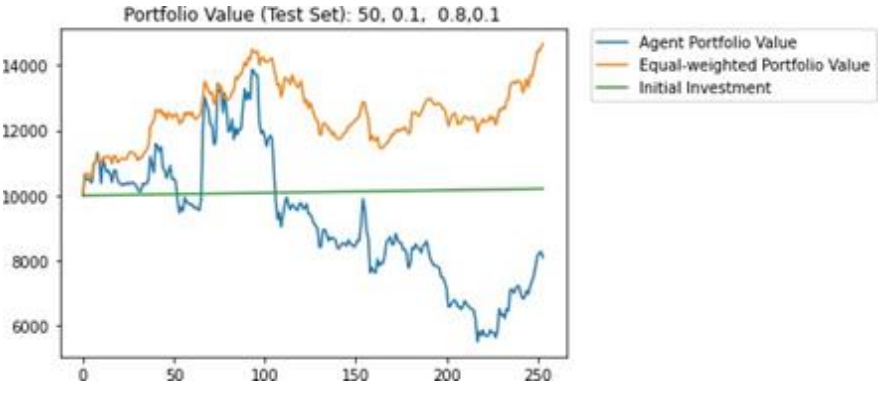

Figure 12 Linear regression on randomly selected portfolio

5 Conclusion

This research implemented deep reinforcement learning method to allocate weights for stocks and cryptocurrency in portfolio management and compared different policy network models. The training results from various policy networks constructed with convolutional neural network, recurrent neural network, or the mixed one show similar results. In this research, three different portfolios were considered, and they include stocks that have been negatively influenced by the Covid-19, the stocks that have been positively influenced by Covid-19, the stocks, and cryptocurrency that were randomly selected. Among the three portfolios, it was found that all the agents have the best performance on the randomly selected portfolio, which presents overall stable moving-up portfolio values as time goes on. The average return rate for all three deep reinforcement learning agents reached 47.38%. It is known that the financial market is filled with uncertainties and unpredictability due to many different factors, so the training results from this research also show the instability when dealing with positively influenced portfolio and negatively influenced portfolio. The linear regression model for portfolio optimization shows poor results. But the weight allocation changed more when compared with deep reinforcement learning agents. Overall, it does not meet the goal of financial portfolio management. On the other hand, the deep reinforcement learning agents show a better performance when cryptocurrency was added to the portfolio.

References

Z.Liang, K.Jiang, H.Chen, J.Zhu and Y.Li. (2021), *Deep Reinforcement Learning in Portfolio Management*. Retrieved from https://ar5iv.labs.arxiv.org/html/1808.09940

Z.Liang, H.Chen, J.Zhu, K.Jiang and Y.Li. (2018). *Adversarial Deep Reinforcement Learning in Portfolio Management.* Retrieved from https://arxiv.org/pdf/ 1808.09940.pdf

Y.Zhang, P.Zhao, Q.Wu, B.Li, J.Huang and M.Tan. (2020). *Cost-Sensitive Portfolio Selection via Deep Reinforcement Learning*. Retrieved from https://arxiv. org/pdf/2003.03051.pdf.

N.Kanwar.(2019). *Deep Reinforcement Learning-based Portfolio Management*. Retrieved from https://rc.library.uta.edu/uta-ir/bitstream/handle/10106/28108/KANWAR-THESIS-2019.pdf?sequence=1&isAllowed=y

Z.Jiang and J.Liang. (2017). *Cryptocurrency Portfolio Management with Deep Reinforcement Learning. Retrieved from* https://arxiv.org/pdf/1612.01277.pdf

Z.Jiang, D.Xu and J.Liang. (2017). *A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem*. Retrieved from https://arxiv.org/pdf/1706.10059.pdf

S.Amrouni, A.Moulin and P.Mizrahi. (2018). *A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem*. Retrieved from https://github.com/selimamrouni/Deep-Portfolio-Management-Reinforcement-Learning

P.Henderson, R.Islam, P.Bachman, J. Pineau, D.Precup and D. Meger. (2019). *Deep Reinforcement Learning that Matters*. Retrieved from https://arxiv.org/pdf/ 1709.06560.pdf

The Investopedia Team. (2021). *Modern Portfolio Theory (MPT).* Retrieved from https://www.investopedia.com/terms/m/modernportfoliotheory.asp