Project Report
Portfolio Optimization Using Deep Reinforcement Learning

Bishishta Mukherjee, Konanki Sai Charan, Manvita Markala, Romil Rathi

I. Summary

Financial portfolio optimization is the process of redistributing funds into multiple financial vehicles at a given timestamp to maximize returns while minimizing the risk at the same time. The goal of this project is to provide a solution framework which deals with this complex financial engineering problem. The baseline framework is implemented using the concept of Modern Portfolio Theory[1]. The Reinforcement Learning framework is implemented using two machine learning methods - Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). All the frameworks are trained and tested using stocks and cryptocurrency trading data.
The dataset consists of:

1) Historical trading data for 16 Stocks from S&P 500 [2] Portfolio from 2005 to present.
2) Historical trading data for 6 Cryptocurrencies from CoinMarketCap [3] from 2015 to present.

For stocks, the trading period considered is regular trading hours for which the stock exchange is open each day. For cryptocurrencies, the market is open 24/7 which makes it continuous in nature. Thus, the trading period was discretized using two different time frames, first being 24 hour period and the second being 30 minute period. The reason for considering 24 hour period, was to better understand cryptocurrency data considering a day to day comparison and fluctuations in the trading prices. However, to better capture the continuous nature of cryptocurrency data, a smaller period of 30 minutes was also considered.
The price data follows the format of Open, High, Low and Close (OHLC) for a given trading period. Open is the price at which the stock begins trading, High is the highest price value it attains throughout the period, Low is the lowest value it attains throughout the period and Close is the closing price value at the end of the trading period. Usually, open price is equal to close price for the previous period, but cryptocurrency follows high-frequency trading, thus, open price for a particular period may not be the same as the closing price for the previous period.

II. Methods

A. Asset Pre-selection

For selection of stocks, diversification over different sectors has been considered to minimize the risk factor. The different sectors considered were Energy, Basic Materials, Industrials, Consumer Discretionary, Consumer Staples, Healthcare, Financial, Information Technology, Communications and Real Estate. Furthermore, stocks that belong to the same sector have been filtered based on low correlation amongst them and a heat map has been plotted for the same which can be observed in figure 1. Majority of the times the returns (percentage change in closing price) of less correlated stocks move in opposite directions and hence a security factor comes into play i.e., if the portfolio is distributed amongst these assets, the decrease in returns from one asset can be balanced by the increase in the other asset.

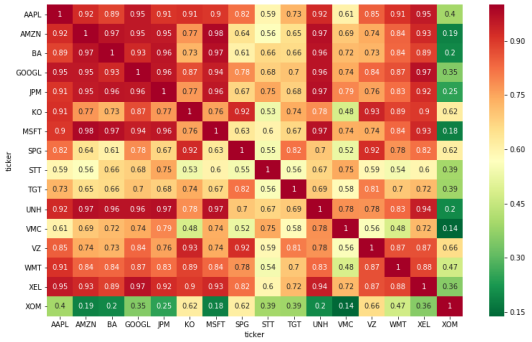

Fig. 1. Correlation plot for selected Stocks

The two criteria mentioned above were used to select 16 assets to be included in the portfolio. Therefore, the portfolio size for stocks is 17 (16 assets + 1 cash).
Six different cryptocurrencies were selected on the basis of the volume of trading i.e., the ones which are performing really good in the market rather than based on the correlation values between them. The basic reason for doing so is that because cryptocurrency has been introduced in recent years, there is not enough data for the majority of them. In such a scenario, only the top trading one's provide us with sufficient data to build a well-trained framework. Therefore, the portfolio size for cryptocurrencies is 7 (6 assets + 1 cash).

B. Data Collection

The trading data for stocks was scraped using a combination of pandas_datareader[4] that takes data from Yahoo! Finance and Selenium[5] that takes data from NASDAQ. The trading data

for cryptocurrency 24 hour period and 30 minute period was pulled using cryptocmd[6] and Poloniex API[7] respectively. For stocks, the total data was collected for 3671 trading days and contained no missing values. Therefore, stocks data needed no further tidying. But in case of cryptocurrency, the total data collected varied from 1500 to 2300 trading days approximately because of the fact that certain cryptocurrencies like Bitcoin were launched as early as 2009 and others like Ethereum were launched much later in the market, around 2015. Due to this wide gap between the launch dates, the time window for each of the cryptocurrency was set to the same dates as that of Ethereum i.e., from August 2015 to present and hence, the data was captured for 1514 trading days. Therefore, the data for the cryptocurrency 24 hour period was for 1514 trading periods whereas, for the cryptocurrency 30 minute period, the data considered was for 1514*48 trading periods.

C. Portfolio Optimization Using Modern Portfolio Theory

Modern Portfolio Theory is an investment theory, which works on the underlying concept of risk-return trade-off i.e greater the probability of higher return, the greater the risk and vice-versa. The theory assumes that investors are risk-averse i.e., they will choose the portfolio with less risk if offered with two portfolios that have the same expected returns. Thus, an investor will take higher risk only if compensated by higher expected returns. This theory also focuses on the assets being diversified instead of allocating resources to a single stock solely. Thus, by optimizing allocation there is a possibility of achieving lower risk than the stock with the lowest risk in the portfolio. The overall return is the return of the weighted average of the individual stocks. However, the overall risk depends on how the movements of the considered stocks correlate with each other. Therefore, the final risk is not equal to the weighted average of the risk related to individual stocks. In this method, random weight vectors are simulated for the portfolio. For this project 30,000 random weight vectors were generated. Corresponding to each weight vector generated, annualised returns and annualised volatility for the portfolio were calculated.

Annualised return is the return over a period of a year from the time t, at which the weight vector w is assigned to the portfolio.

$$\boxed{\text{Annualised Returns}_t = (\text{w}_{(t)}^T.\text{M}) * \text{T}} \tag{1}$$

where,
M = Mean Daily Returns for the individual assets considered.
T = Total trading periods for a year (Stocks - 252 days, Cryptocurrency [24 Hours] - 365 days, Cryptocurrency [30 Min] - 365*48 intervals)

Annualised volatility is the portfolio's standard deviation over a period of a year from the time t, at which the weight vector w is assigned to the portfolio.

$$\boxed{\text{Annualised Volatility}_t = \sqrt{(\text{w}_{(t)}^T(\text{C.w}_{(t)}))}} \tag{2}$$

where,
C= Covariance Matrix for daily returns for the individual assets considered.

Sharpe ratio is then calculated which is the risk-adjusted return i.e., it refines the previously calculated annualised returns by measuring how much risk is involved in producing the returns. The ratio describes how much excess returns are being received for the extra volatility that is endured for holding the riskier asset.

$$\boxed{\text{S} = \frac{\text{R}_a - \text{R}_f}{\sigma_a}} \tag{3}$$

where,
S = Sharpe ratio
R_a = Annualised Returns
R_f = Risk free rate of returns that is taken from US department of treasury and is 1.7% approximately as of September 30, 2019, considering 52-week treasury bill rates
σ_a = Annualised Volatility

All the randomly generated portfolios are then plotted with colour map applied to them based on the Sharpe ratio. Two optimal portfolios are located, one with the maximum Sharpe ratio and the other with minimum volatility.

D. Portfolio Management Using Reinforcement Learning

1) **Building Price Tensor**: For each of the stocks and cryptocurrencies, 4 lists containing the scraped raw price values for Open, High, Low and Close (OHLC) over the entire time frame have been created and were then normalized by the opening price since only the changes in the prices will determine the performance of the portfolio management rather than the raw prices themselves. The lists have been normalized as follows:

$$\text{Close values list} = \left[\frac{\text{Close}_{(t-n-1)}}{\text{Open}_{(t-n-1)}}, ..., \frac{\text{Close}_{(t-1)}}{\text{Open}_{(t-1)}}\right]$$

$$\text{High values list} = \left[\frac{\text{High}_{(t-n-1)}}{\text{Open}_{(t-n-1)}}, ..., \frac{\text{High}_{(t-1)}}{\text{Open}_{(t-1)}}\right]$$

$$\text{Low values list} = \left[\frac{\text{Low}_{(t-n-1)}}{\text{Open}_{(t-n-1)}}, ..., \frac{\text{Low}_{(t-1)}}{\text{Open}_{(t-1)}}\right]$$

$$\text{Open values list} = \left[\frac{\text{Open}_{(t-n)}}{\text{Open}_{(t-n-1)}}, ..., \frac{\text{Open}_{(t)}}{\text{Open}_{(t-1)}}\right]$$

At the end of period t, the input price tensor to the neural network is of the shape (x, y, z) where x is the number of features (OHLC in case of portfolio management) and y is the number of non-cash assets and z is the trading periods considered until time t. This price tensor for stocks data is of the shape (4, 15, 3670) and that for cryptocurrencies 24 hour period is of the shape (4, 6, 1513) and 30 minute period is of the shape (4, 5, 72671).

2) **Reinforcement Learning Environment Setup**: Reinforcement learning is one of the basic machine learning paradigms which is concerned with how the software agent should take actions in an environment so as to maximize some notion of cumulative reward and move in the direction of achieving a goal.

In the solution framework presented, the financial market is the environment and the portfolio optimisation software is acting as an agent allocating funds amongst different assets at a particular instance of time. This goal-oriented RL framework approximates the profit function and then trains the before mentioned neural networks - CNN and LSTM, making them learn how to maximize the profit over time.

The action taken by an agent at time t is reallocating the funds between various assets and thus, determines the current weight vector w_t.

$$\boxed{Action_t = w_t = [w_{cash}, w_{asset1}, w_{asset2}, ..., w_{assetn}]} \quad (4)$$

A transaction cost which is normally a constant commission fee involved for each stock/cryptocurrency traded is accounted for while reallocating the funds i.e., when buying or selling assets. Therefore, the total transaction cost[8] at time t will be calculated while distributing the funds according to the new weight vector at time t i.e w_t from weight vector at time $(t-1)$ i.e., w_{t-1}.

$$\boxed{\begin{aligned}&\text{Total transaction cost}_{(t)} = \\ &\text{Portfolio value}_{(t-1)} * \text{Transaction Fee} * (w_{(t)} - w_{(t-1)})\end{aligned}} \quad (5)$$

Therefore, every time the stock portfolio is updated, this transaction cost incurs on the portfolio value i.e., the total funds at timestamp t.

$$\boxed{\begin{aligned}&\text{Portfolio vector}_{(t)} = (\sum(\text{Portfolio value}_{(t-1)}) * w_t) \\ &- (\text{Total Transaction Cost} + [0] * \text{number of assets})\end{aligned}} \quad (6)$$

Portfolio vector is a vector of size: (number of assets + 1) to store the value for each asset at time t.

$$\boxed{\begin{aligned}&\text{Cumulative Portfolio Vector}_{(t)} = \\ &(\sum(\text{Portfolio value}_{(t-1)}) * w_t) - (\text{Total Transaction} \\ &\text{Cost} + [0] * \text{Number of assets}) * \text{Return rate}_{(t)}\end{aligned}} \quad (7)$$

$$\boxed{\begin{aligned}&\text{Final Weight Vector}_{(t)} = \\ &\text{Portfolio vector}_{(t)}/\text{Cumulative Portfolio value}_{(t)}\end{aligned}} \quad (8)$$

Here, return rate at time t contains the rate of interest for each of the asset which the agent will:

1) get at each step if it has a positive amount of money.
2) pay if it has a negative amount of money.

The reward value at time t is the percentage change in the portfolio value and thus depends on Portfolio value$_{(t-1)}$ and Total transaction cost$_{(t)}$.

$$\boxed{\text{Reward}_{(t)} = (\text{Portfolio value}_{(t)}/\text{Portfolio value}_{(t-1)}) - 1} \quad (9)$$

This dependency is encapsulated by considering w_{t-1} and Portfolio value$_{(t-1)}$ as part of the input to the agent at time t. So the state at time t (state$_{(t)}$) consists of the price tensor at time t, weight vector at time $t-1$ (w_{t-1}) and portfolio value at time $t-1$ Portfolio value$_{(t-1)}$ where $W_0 = [1, 0, 0..., 0]^T$

$$\boxed{\text{State}_{(t)} = (\text{Price tensor}_{(t)}, w_{(t-1)}, \text{Portfolio value}_{(t-1)})} \quad (10)$$

Here, two environments are being setup. An environment for the portfolio optimization agent and a baseline environment for the agent assigning equal weights amongst the assets. Using this setup the adjusted reward for time stamp t is calculated as follows:

$$\boxed{\begin{aligned}&\text{Adjusted Reward}_{(t)} = \\ &\text{Reward}_{(t)} - \text{Baseline Reward}_{(t)} * \alpha * max(w_{(t)})\end{aligned}} \quad (11)$$

Where $\alpha * max(w_{(t)})$ is a term proportional to the maximum of the weight vector at time t. This term is taken into account so that the agent avoids investing in a single stock solely.

3) **Policy Network Architecture for CNN**: Policy function guides the agent what action to take from a particular state i.e., how to reallocate the weights amongst assets so as to maximize profit. For the first model, CNN is used to design the policy function. As seen in figure 2, the input to this network is the price tensor (price tensor$_{(t)}$) and the output is the weight vector $(w_{(t)})$, the portfolio value at t (Portfolio value$_{(t)}$) and adjusted reward at t (Adjusted Reward$_{(t)}$). The first convolution layer is realised in a small tensor and has 2 filters. Based on a brute force approach, the activation function used for the first convolution layer is $Tanh$ for stocks data and $ReLU$ for cryptocurrency data. The second convolution layer results in a second vector (number of assets * 1 * 1) and has 20 filters. The previous output vector is stacked. The activation function used here is $ReLU$ for both stocks and cryptocurrency data. For the neural network to minimize the transaction cost, portfolio vector of the previous time step is inserted before the last layer. The portfolio vector produced by the network at each time step is stored in a Portfolio Vector Memory (PVM) to be used at the next time step. The last layer is a terminate convolution layer resulting in a unique vector of size same as the number of assets. Then a cash bias is added and softmax applied. An important feature of this network is that the network parameters are common for all the assets but they are processed independently. When the softmax function is applied, it is ensured that the weights allotted are non-negative and sum up to unity.

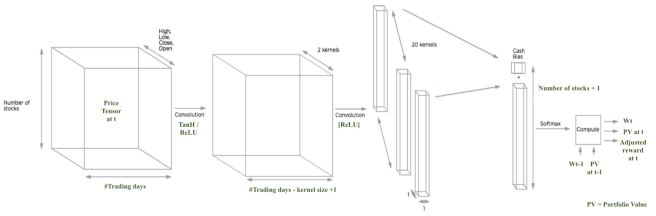

Fig. 2. CNN Architecture for Portfolio Optimization

4) **Policy Network Architecture for LSTM**: For the second model, LSTM is used to design the policy function. The advantage that LSTM has over CNN is that it is better able to process and make predictions on sequential data, in this case, the sequence of historical trading data. As seen in figure 3, the process flow for the LSTM network is similar to the previously mentioned CNN network flow. The difference between the two is realized from the fact that the chain of two convolution layers have been replaced by a basic RNN unrolling layer. RNN unrolling is basically a loop over subsequent time steps where at every time step apart from the input at that time step, the network also takes in information which is remembered or passed on from the previous time step. This chain-like structure makes LSTM intimately related to sequences. Therefore, the price inputs of individual assets are taken by small recurrent subnets and the structure of the ensemble network after the recurrent subnets is the same as the second half of the CNN.

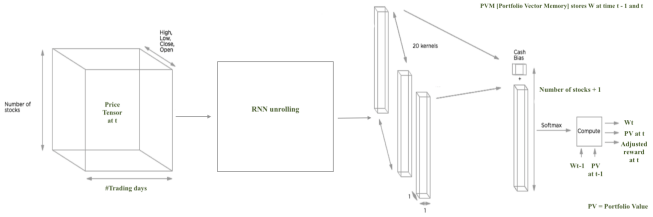

Fig. 3. LSTM Architecture for Portfolio Optimization

E. Evaluation Metrics

1) **Benchmark Testing**: The first evaluation metric is based on benchmark testing[9] where the outcomes produced by built optimizing agent will be compared with outcomes produced by other agents, the first being the agent that distributes the weights equally amongst all the available assets and the second, that does not invest in any of the assets. The outcome here is the cumulative portfolio value returned over test steps. According to the hypothesis, a good optimizer agent should outperform the equiweight agent but this strategy does not accommodate the risk factor involved while calculating the returns. Therefore, other evaluation metrics that involve back testing - Sharpe Ratio and Return over Maximum Drawdown were also considered.

2) **Backtesting**: Backtesting is a term used in financial modelling where predictive strategy is tested on historical data and the viability of the same is assessed by discovering how it would play out on historical data. It is a key component of effective trading system development.
The two key assumptions[10] made in backtesting are:

1) Zero Slippage: the liquidity of all market assets is high enough that each trade can be carried out immediately at the last price.
2) Zero Market Impact: the capital invested by the software trading agent is so insignificant that it has no influence on the market.

These two assumptions are near to reality in the real world trading environment if the trading volume in the market is high enough.
Sharpe ratio is a measure of the risk-adjusted return. It is calculated using the following equation:

$$\boxed{\mathrm{S} = \frac{\mathrm{R}_a - \mathrm{R}_f}{\sigma}} \tag{12}$$

where,
S = Sharpe ratio
R_a = portfolio returns which is nothing but the weighted average of daily returns of all the assets
R_f = risk free rate of return. For general scenarios in the real world, this component is taken as 0
σ = the asset volatility which is calculated as the standard deviation of the portfolio returns
If the Sharpe ratio generated by the current strategy is:

- less than 1, then the strategy developed is considered to be sub-optimal and has a great scope of further optimization
- greater than 1, then it is considered to be fairly good
- greater than 2, then it is considered to be very good
- greater than 3, then it is considered to be excellent

The drawback of using Sharpe Ratio as an evaluation metric is that it uses standard deviation in the denominator which assumes the returns are normally distributed and could be misleading because sometimes the financial returns deviate from a normal distribution. Therefore, an additional metric called Return over Maximum Drawdown (RoMaD) was used which does not assume that the returns follow a normal distribution, rather believes that observed loss patterns over longer periods are the best proxy for actual exposure. The intuition behind this method is that the investor is willing to accept an occasional drawdown of X% to generate a return of Y%.

$$\boxed{\mathrm{RoMaD} = \frac{\mathrm{CPV}}{\mathrm{MDD}}} \tag{13}$$

where,
CPV = Cumulative Portfolio Value
MDD = Maximum Drawdown i.e., the maximum observed loss from a peak to a trough of a portfolio before a new peak is attained

$$\boxed{\mathrm{MDD} = \frac{\text{Peak Value} - \text{Trough Value}}{\text{Peak Value}}} \tag{14}$$

MDD is an indicator of downside risk over a specified time period. If the RoMaD generated by the current strategy is:

- less than 0.5, the strategy developed is considered to be sub-optimal and has a great scope of further optimization
- greater than 0.5, then it is considered to be good
- between the range 3 to 5, then it is considered to be excellent

III. Results

A. Modern Portfolio Theory

1) **Cryptocurrency [24 Hours]**: To understand the evolution of cryptocurrency trading price over time, the closing price

for each cryptocurrency over the considered time period was plotted.

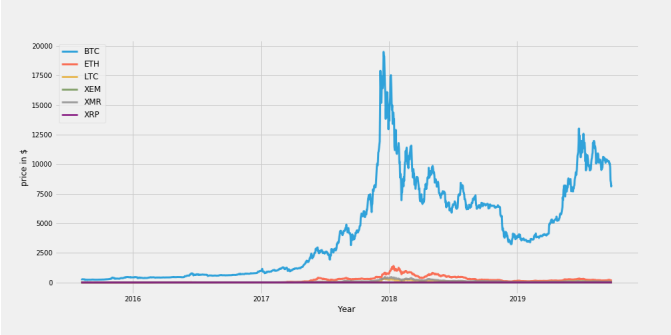

Fig. 4. Variation of closing price over time for Cryptocurrency [24 Hours]

From figure 4, it is observed that Bitcoin's (BTC) price is relatively higher when compared to other assets and shows a lot of variation in recent years. However, all the other cryptocurrencies lie in a similar range which makes it difficult to understand the corresponding variation over time by observing the plot obtained. Therefore, plotting the variation of daily returns over time for each asset is a better way to observe the volatility of each stock. Daily returns is nothing but the percentage change in current day's closing price as compared to the previous day's closing price.

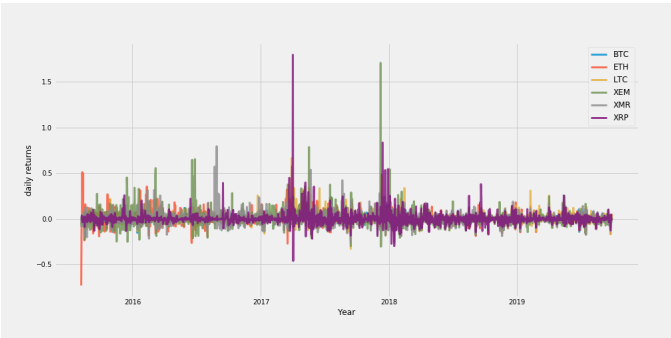

Fig. 5. Variation of Daily Returns over time for Cryptocurrency [24 Hours]

From figure 5, it is observed that Ripple (XRP) and NEM (XEM) have shown a lot of extreme variations in daily returns whereas BTC has not shown any extreme variation over time.

After plotting the randomly generated portfolios using colour map based on the Sharpe Ratio, it can be observed in figure 6, that a curved boundary is formed by the darkest blue dots towards the left of the cluster obtained. This boundary formed is called the Efficient Frontier. The green star in the plot represents the portfolio with minimum risk (volatility) and the red star represents the portfolio with maximum Sharpe Ratio. For a given level of return, the bluer the dot representing the portfolio, the higher the Sharpe Ratio for that portfolio. Therefore, if two portfolios with the same returns are given to a risk-averse investor, he would choose the one with the most bluer dot i.e., the one with least risk.

By looking at the weight vectors allocated in the figure 7 for minimum volatility portfolio allocation, BTC is allocated

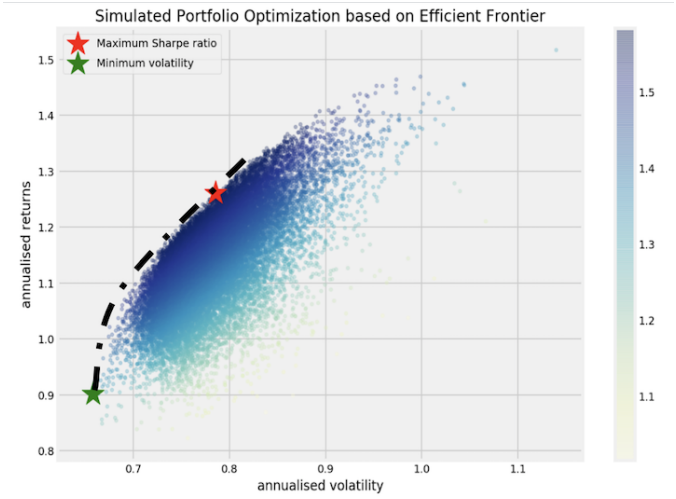

Fig. 6. Efficient Frontier plot for Cryptocurrency [24 Hours]

```
--------------------------------------------------------------------------------
Maximum Sharpe Ratio Portfolio Allocation

Annualised Return: 1.26
Annualised Volatility: 0.79

              BTC    ETH   LTC    XEM    XMR    XRP
allocation   12.6  30.21  1.79  22.29  17.26  15.86
--------------------------------------------------------------------------------
Minimum Volatility Portfolio Allocation

Annualised Return: 0.9
Annualised Volatility: 0.66

              BTC    ETH  LTC   XEM   XMR    XRP
allocation  62.78  19.83  0.3  2.41  2.36  12.31
```

Fig. 7. MPT: Weight Allocation for Cryptocurrencies [24 Hours]

the highest weight which aligns with the fact that it is least volatile in nature as seen from figure 5, investing in which poses the least risk intuitively. For maximum Sharpe Ratio portfolio allocation, Ethereum (ETH) and XEM are allocated the highest weights which have high variations when compared to BTC as seen from figure 5.

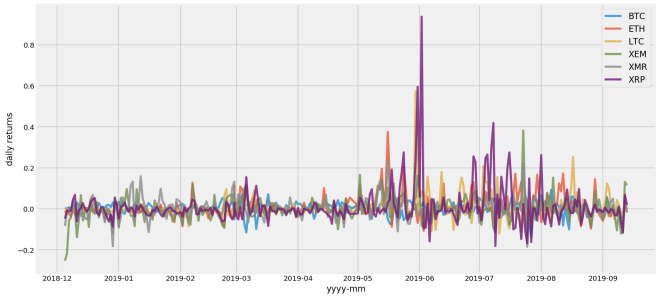

Fig. 8. Variation of Daily Returns over time for Cryptocurrency [30 Min]

2) **Cryptocurrency [30 Min]**: For better readability, figure 8 was generated using data for the most recent year and

mean of all the data points for each day was taken. From figure 8, it is observed that BTC has shown the least variation in daily returns whereas other cryptocurrencies have shown considerable variations over time with XRP being the most extreme.

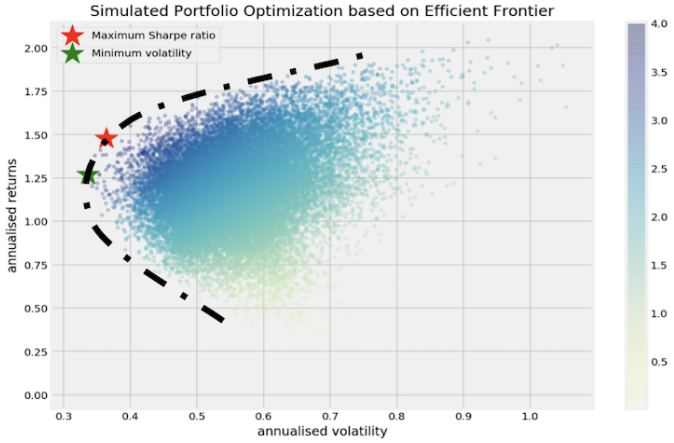

Fig. 9. Efficient Frontier plot for Cryptocurrency [30 Min]

```
--------------------------------------------------------------------------------
Maximum Sharpe Ratio Portfolio Allocation

Annualised Return: 1.48
Annualised Volatility: 0.36

crypto_ticker     BTC   ETH    LTC    XEM    XMR    XRP
allocation      12.68  17.4  11.65  3.46   0.38  54.43
--------------------------------------------------------------------------------
Minimum Volatility Portfolio Allocation

Annualised Return: 1.27
Annualised Volatility: 0.34

crypto_ticker     BTC    ETH    LTC    XEM    XMR    XRP
allocation      60.69  10.04  12.63  2.84   10.35  3.45
```

Fig. 10. MPT: Weight Allocation for Cryptocurrencies [30 Min]

The efficient frontier obtained for cryptocurrency [30 Min] can be seen in figure 9. By looking at the weight vectors allocated in the figure 10 for minimum volatility portfolio allocation, BTC and Litecoin (LTC) have the highest allocation which aligns with the fact that they are the least volatile in nature as seen from figure 8, investing in which poses the least risk intuitively. For maximum Sharpe Ratio portfolio allocation, ETH and XRP are allocated the highest weights which exhibits high variation in daily returns.

3) **Stocks**: From figure 11, it is observed that Apple (AAPL), Google (GOOGL), Amazon (AMZN), Xcel (XEL), State Street (STT) have shown high variations in daily returns over time whereas Coca-Cola Co (KO), Verizon (VZ) and Boeing Co (BA) have shown comparatively lesser variations in daily returns.

The efficient frontier obtained for stocks can be seen in figure 12. By looking at the weight vectors allocated in figure 13 for minimum volatility portfolio allocation, KO and VZ have

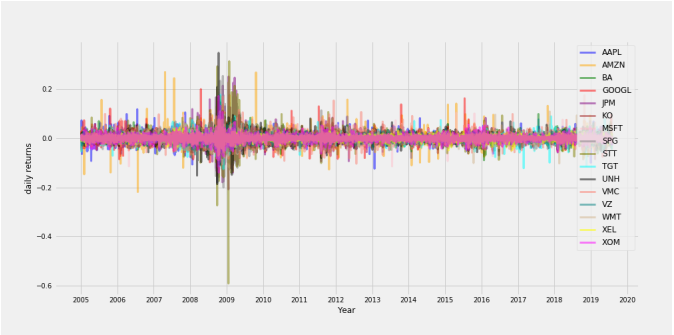

Fig. 11. Variation of Daily Returns over time for Stocks

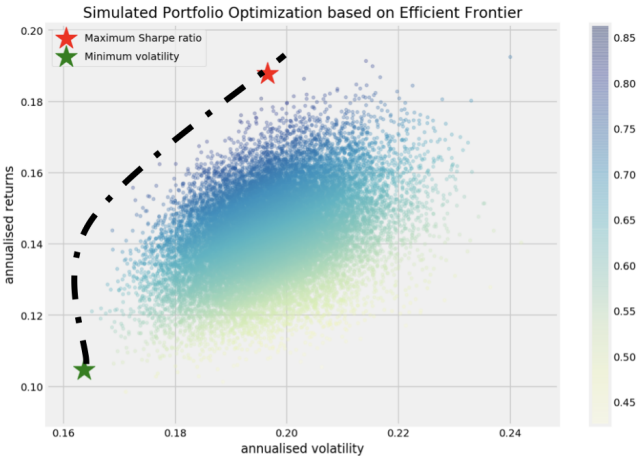

Fig. 12. Efficient Frontier plot for Stocks

```
--------------------------------------------------------------------------------
Maximum Sharpe Ratio Portfolio Allocation

Annualised Return: 0.19
Annualised Volatility: 0.2

            AAPL   AMZN    BA   GOOGL   JPM    KO   MSFT  SPG   STT   TGT  \
allocation  14.84  12.99  8.46  16.87  0.45  2.61  1.52  1.2  3.36  5.18

             UNH   VMC    VZ    WMT    XEL   XOM
allocation  3.84  6.41  2.32  1.45  16.22  2.26
--------------------------------------------------------------------------------
Minimum Volatility Portfolio Allocation

Annualised Return: 0.1
Annualised Volatility: 0.16

            AAPL  AMZN    BA  GOOGL   JPM    KO  MSFT  SPG   STT    TGT  \
allocation  3.46  2.83  1.04   5.05  0.37  16.09  1.44  1.12  1.97  14.33

             UNH   VMC    VZ    WMT    XEL   XOM
allocation  1.37  8.52  16.19  7.17  13.62  5.44
```

Fig. 13. MPT: Weight Allocation for Stocks

Results	Mean Sharpe Ratio	Mean RoMaD
MPT	1.26	-
CNN (RL)	1.652	1.239
LSTM (RL)	1.815	0.983

TABLE I
BACKTESTING RESULTS FOR CRYPTOCURRENCY [24 HOURS]

the highest allocation which aligns with the fact that they are amongst the lesser volatile stocks as seen from figure 11, investing in which poses the least risk intuitively. For maximum Sharpe Ratio portfolio allocation, GOOGL, XEL and AAPL are allocated the highest weights which exhibits high variation in daily returns.

B. Portfolio Optimization using Reinforcement Learning

1) **Cryptocurrency [24 Hours]**: To visualize the benchmark results, cumulative portfolio values over the test steps were plotted for optimizing agent, equiweight agent and when no investment is done. As can be seen from figure 14 and figure 15, the optimizing agent performs the best for both CNN and LSTM. Between CNN and LSTM, the plots show that LSTM policy is giving higher returns when compared to the CNN policy.

To verify the benchmark results, Mean Sharpe Ratio and

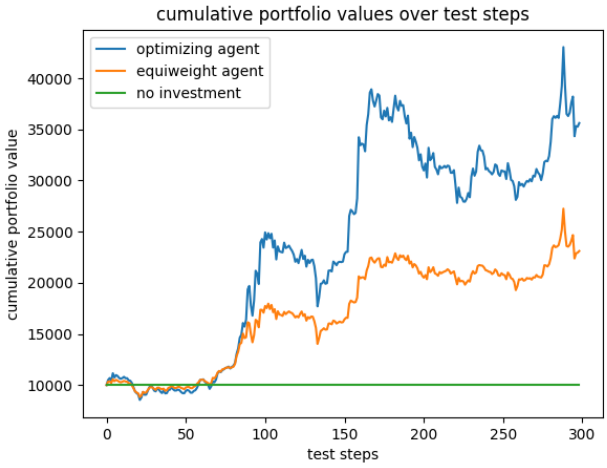

Fig. 14. CNN: Cumulative Portfolio Value over Test steps for Cryptocurrency [24 Hours]

mean RoMaD were calculated. As seen from the table I, the least mean Sharpe ratio was obtained for MPT policy while the highest was obtained for LSTM policy. This aligns with the obtained benchmark results. However, the mean RoMaD contradicts this conclusion and indeed a lower mean RoMaD for LSTM policy suggests that CNN policy is performing better in this case. The reason behind this can be either the returns from the financial data were not normally distributed giving misleading mean Sharpe Ratio or 24 hour period data did not capture the sequential behaviour of cryptocurrencies well which led to a poorly learnt LSTM policy.

As seen from figure 16, the highest weights were allocated to

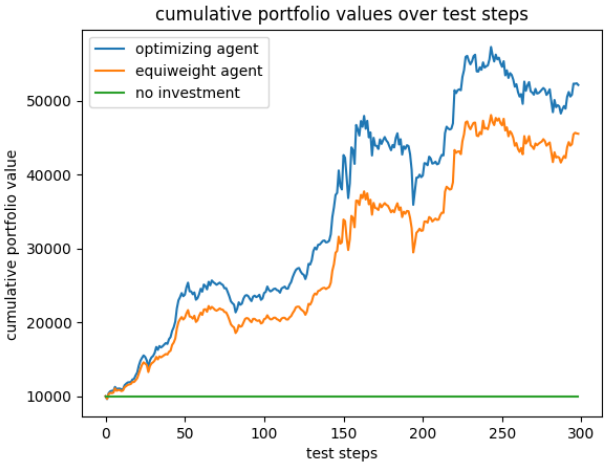

Fig. 15. LSTM: Cumulative Portfolio Value over Test steps for Cryptocurrency [24 Hours]

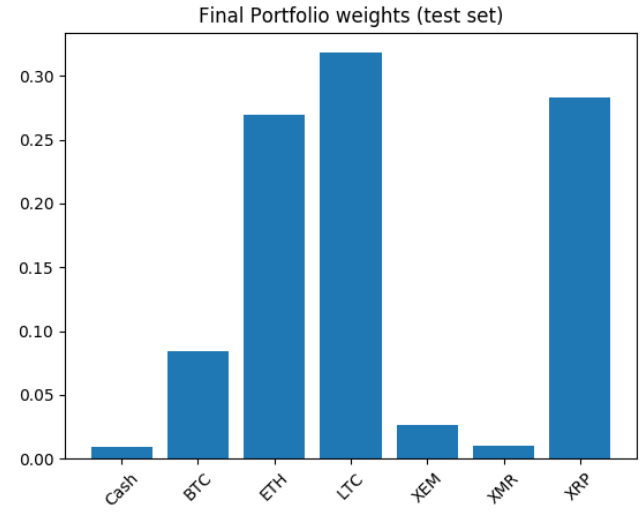

Fig. 16. CNN: Weight Allocation for Cryptocurrencies [24 Hours]

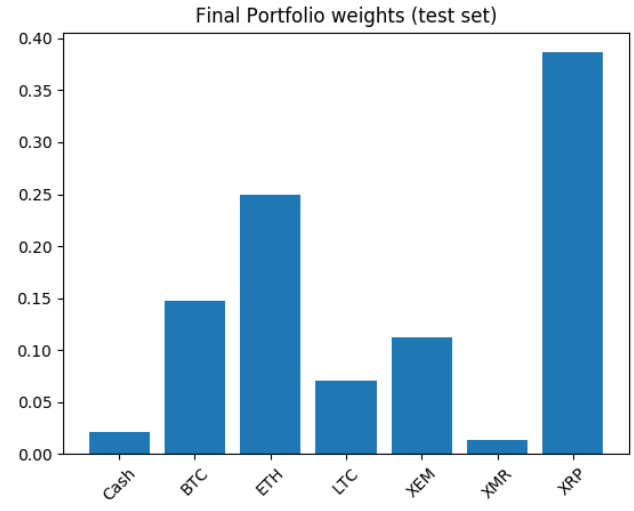

Fig. 17. LSTM: Weight Allocation for Cryptocurrencies [24 Hours]

LTC, ETH and XRP for the CNN policy whereas for LSTM policy, the highest weights were allocated to ETH and XRP as can be seen from figure 17.

2) **Cryptocurrency [30 Min]**: The cumulative portfolio values over test steps plots as shown in figure 18 and figure 19 indicates that for both CNN and LSTM, the optimizing agent performs the best. Between CNN and LSTM the plots show that LSTM is giving higher returns when compared to CNN.

As seen from table II, the least mean Sharpe Ratio was obtained for MPT policy while the highest was obtained for LSTM policy. This aligns with the benchmark results. Mean RoMaD also confirms that the LSTM policy is doing a better job in optimizing the portfolio value when compared to the other methods implemented. The amount of data available for training the agent for 30 minute period was much more than the 24 hour period which could be a reason that LSTM was better able to capture the continuous nature of cryptocurrencies giving higher mean Sharpe Ratio and mean RoMaD than the one obtained in 24 hour period.

Results	Mean Sharpe Ratio	Mean RoMaD
MPT	1.49	-
CNN (RL)	1.734	1.204
LSTM (RL)	2.003	1.973

TABLE II
BACKTESTING RESULTS FOR CRYPTOCURRENCY [30 MIN]

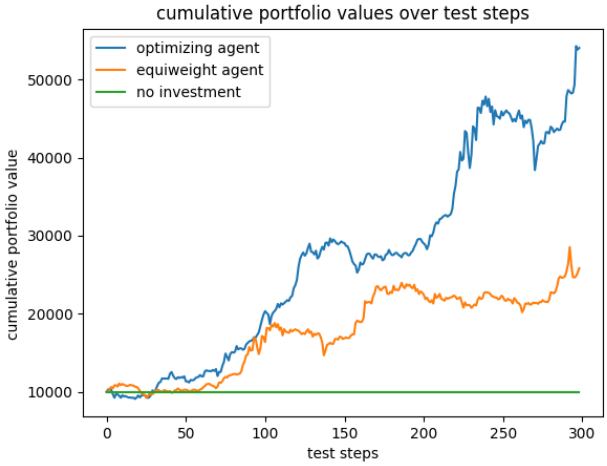

Fig. 18. CNN: Cumulative Portfolio Value over Test steps for Cryptocurrency [30 Min]

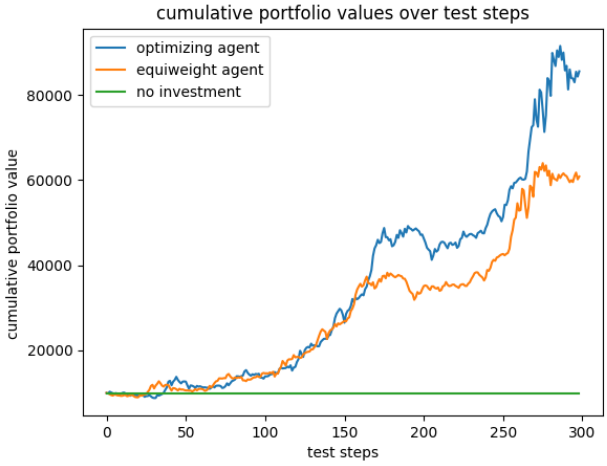

Fig. 19. LSTM: Cumulative Portfolio Value over Test steps for Cryptocurrency [30 Min]

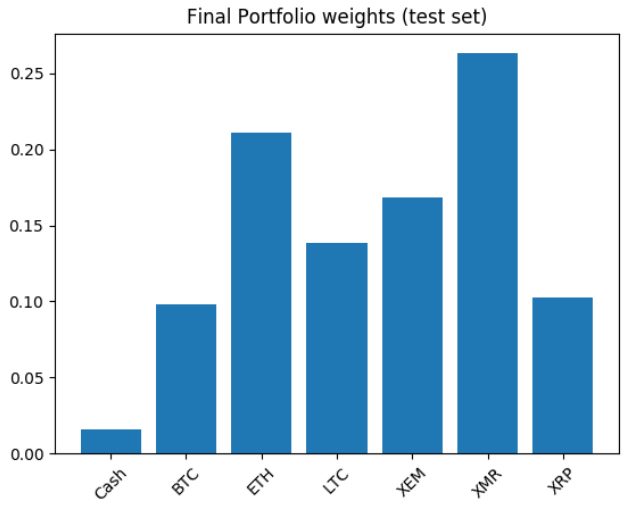

Fig. 20. CNN: Weight Allocation for Cryptocurrencies [30 Min]

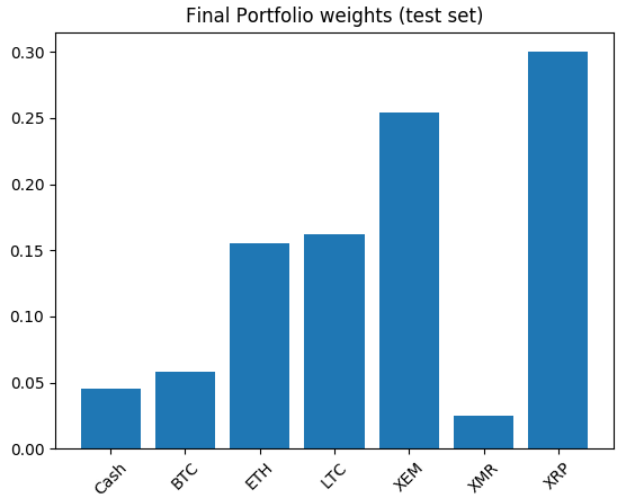

Fig. 21. LSTM: Weight Allocation for Cryptocurrencies [30 Min]

As seen from figure 20, highest weights were allocated to ETH and XMR for the CNN policy whereas for LSTM policy, the highest weights were allocated to XEM and XRP as can be seen from figure 21.

3) **Stocks**: The cumulative portfolio values over test steps plots as shown in figure 22 and figure 23 indicates that for both CNN and LSTM, the optimizing agent performs the best.

Results	Mean Sharpe Ratio	Mean RoMaD
MPT	0.19	-
CNN (RL)	0.635	0.472
LSTM (RL)	1.169	0.807

TABLE III
BACKTESTING RESULTS FOR STOCKS

Between CNN and LSTM the plots show that LSTM is giving higher returns when compared to CNN.

As seen from table III, the least mean Sharpe Ratio was

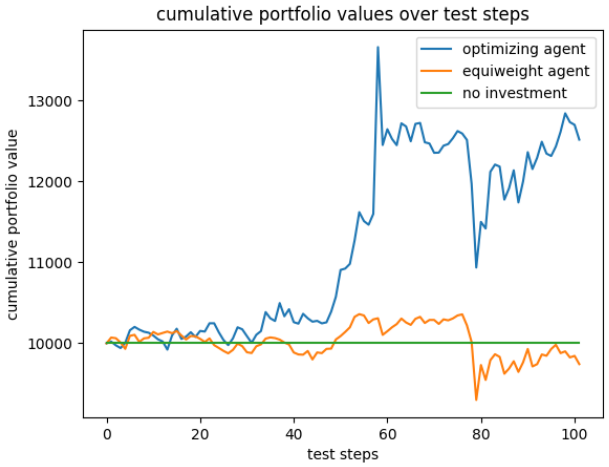

Fig. 22. CNN: Cumulative Portfolio Value over Test steps for Stocks

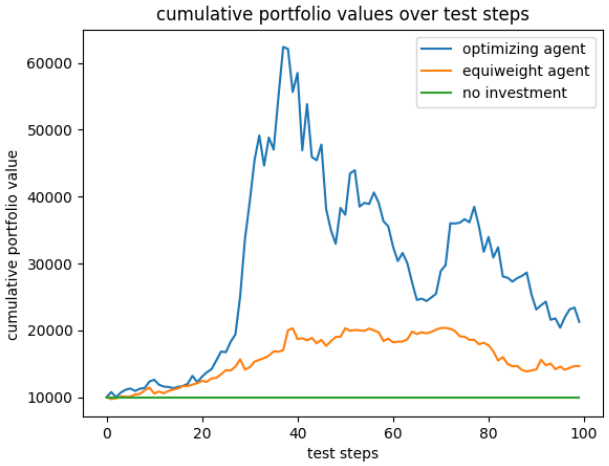

Fig. 23. LSTM: Cumulative Portfolio Value over Test steps for Stocks

obtained for MPT policy while the highest was obtained for LSTM policy. This aligns with the benchmark results. Mean RoMaD also confirms that the LSTM policy is doing a better job in optimizing the portfolio value when compared to the other methods implemented. However for CNN, the cumulative portfolio plots were giving positive returns over initial investments but a lower Sharpe Ratio (less than 1) conveys that the learnt CNN policy is still inefficient and hence, the benchmark results for the same were misleading.

As seen in figure 24, highest weights were allocated to stocks

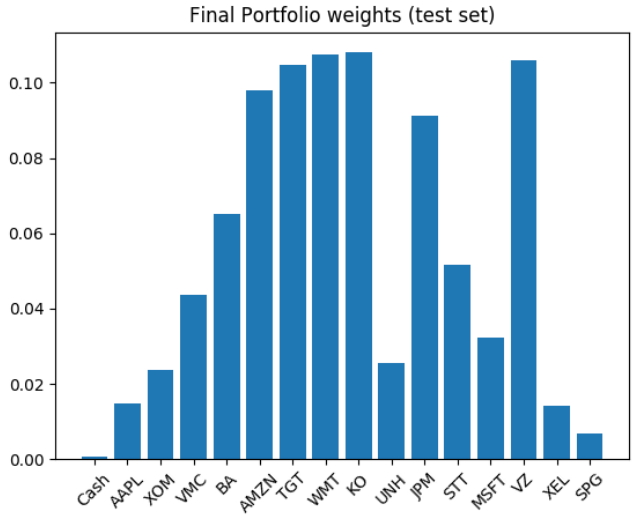

Fig. 24. CNN: Weight Allocation for Stocks

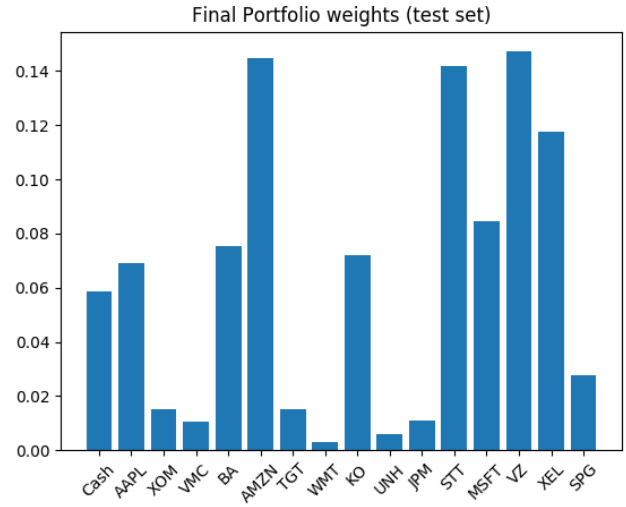

Fig. 25. LSTM: Weight Allocation for Stocks

like AMZN, Target (TGT), KO, Walmart (WMT) and VZ for the CNN policy whereas for the LSTM policy, highest weights were allocated to stocks like AMZN, STT, VZ and XEL as can be seen in figure 25.

IV. DISCUSSION

The models implemented using Reinforcement Learning framework have proven to work better than traditionally existing baseline methods like Modern Portfolio Theory. Amongst the machine learning techniques implemented using RL, LSTM has shown to do a better job in cases where the data is more continuous in nature (in the case of 30 minute period cryptocurrency data) or a huge amount of sequential data is available (in the case of stocks data) rather than CNN.

Methodologies that have been implemented using Reinforcement Learning in the project have given fairly good results with a decent balance between portfolio returns and volatility. In future, the derived modules can be integrated into an automatic trading system which sells and buys assets from the market according to the best portfolio allocation obtained. Since the policies learnt through the RL trading framework were stable and exhibited good performance, the framework built with slight modifications can be applied to general reinforcement learning domains, like gaming applications and recommendation systems.

V. Statement of Contribution

Stage of Project	Member Contribution
Data Scraping	Romil Rathi, Sai Charan Konanki
Data tidying and pre-processing	Bishista Mukherjee, Manvita Markala
Implementing Modern Portfolio Theory	Romil Rathi, Sai Charan Konanki
Building price tensor	Bishista Mukherjee, Manvita Markala
Building the RL environment	Romil Rathi, Sai Charan Konanki
Building the Policy Network Architecture(CNN and LSTM)	Bishista Mukherjee, Manvita Markala, Romil Rathi, Sai Charan Konanki
Tuning hyperparameters of networks for stocks	Bishista Mukherjee, Manvita Markala
Tuning hyperparameters of networks for cryptocurrency	Romil Rathi, Sai Charan Konanki
Calculating and plotting Benchmark and Backtesting results	Bishishta Mukherjee, Manvita Markala, Romil Rathi, Sai Charan Konanki
Documentation (Creating report, presentation and proposals)	Bishishta Mukherjee, Manvita Markala, Romil Rathi, Sai Charan Konanki

VI. Appendix

A. Git link for code

Portfolio Optimization using Deep Reinforcement Learning

B. Ticker Values for selected assets

1) **Stocks Ticker**:

- AAPL: Apple
- XOM: Exxon Mobil Corporation
- VMC: Vulcan Materials Company
- BA: Boeing Co
- AMZN: Amazon.com, Inc
- TGT: Target Corporation
- WMT: Walmart Inc
- KO: Coca-Cola Co
- UNH: UnitedHealth Group Inc
- JPM: JP Morgan Chase Co
- STT: State Street Corp
- MSFT: Microsoft Corporation
- VZ: Verizon Communications Inc
- XEL: Xcel Energy Inc
- SPG: Simon Property Group Inc

2) **Cryptocurrencies Ticker**:

- BTC: Bitcoin
- ETH: Ethereum
- LTC: Litecoin
- XEM: NEM
- XMR: Monero
- XRP: Ripple

References

[1] Modern Portfolio Theory https://towardsdatascience.com/efficient-frontier-portfolio-optimisation-in-python-e7844051e7f

[2] Historical stocks data of 15 assets from SP 500 portfolio https://www.barchart.com

[3] Historical data for 6 Cryptocurrencies from CoinMarketCap https://coinmarketcap.com

[4] Scraping Financial Data with Pandas DataReader https://pandas-datareader.readthedocs.io/en/latest/

[5] Scraping Financial Data with Selenium https://datarebellion.com/blog/scraping-financial-data-with-selenium/

[6] Scraping cryptocurrency data with CryptoCMD https://pypi.org/project/cryptocmd/0.3.2/

[7] Scraping cryptocurrency data with Poloneix API https://docs.poloniex.com/#introduction

[8] Chi Zhang, Corey Chen, Limian Zhang. Deep Reinforcement Learning for Portfolio Management. https://www-scf.usc.edu/~zhan527/post/cs599/

[9] Olivier Jin, Hamza El-Saawy Portfolio Management using Reinforcement Learning Dept. of Computer Science, Stanford, USA. https://pdfs.semanticscholar.org/90ad/8fed17daebaee0a743f9f9847e022739a534.pdf?_ga=2.116929670.1018518321.1572860456-1698436213.1572860456

[10] Zhengyao Jiang, Dixing Xu, and Jinjun Liang A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. Xi'an Jiaotong-Liverpool University, Suzhou, SU 215123, P. R. China. https://arxiv.org/pdf/1706.10059.pdf

[11] Environment Reference RL Environment for Portfolio Management