Automated Stock Market Investment Using Reinforcement Learning

Ankan, Ishan, Bijayan, Shreyam, Praveen

May 22, 2024

Contents

Problem statement

- The user starts with no stocks initially and inputs the bot with a maximum initial investment limit.
- Then according to the market condition, the bot should recommend the user which stock to buy, sell or hold.
- The user should be able to make it completely automated, where the bot by itself buys, sells or holds a given stock according to its prediction and the user-inputted investment limit.

Portfolio Management

The equation below represents the portfolio weights at time t, denoted as $\mathbf{w}_t$, where:

- w_t^i represents the weight of asset i in the portfolio.
- M represents the total number of assets in the portfolio, including one risk-free asset (cash).
- $\sum_{i=0}^{M} w_t^i = 1$ ensures that the sum of weights equals 1, indicating that the entire portfolio is fully invested.

The equation is given by:

$$\mathbf{w_t} = \left[w_t^0, w_t^1, \ldots, w_t^M\right]^T \in R^{M+1} \quad \text{and} \quad \sum_{i=0}^{M} w_t^I = 1$$

Holding and Trading period

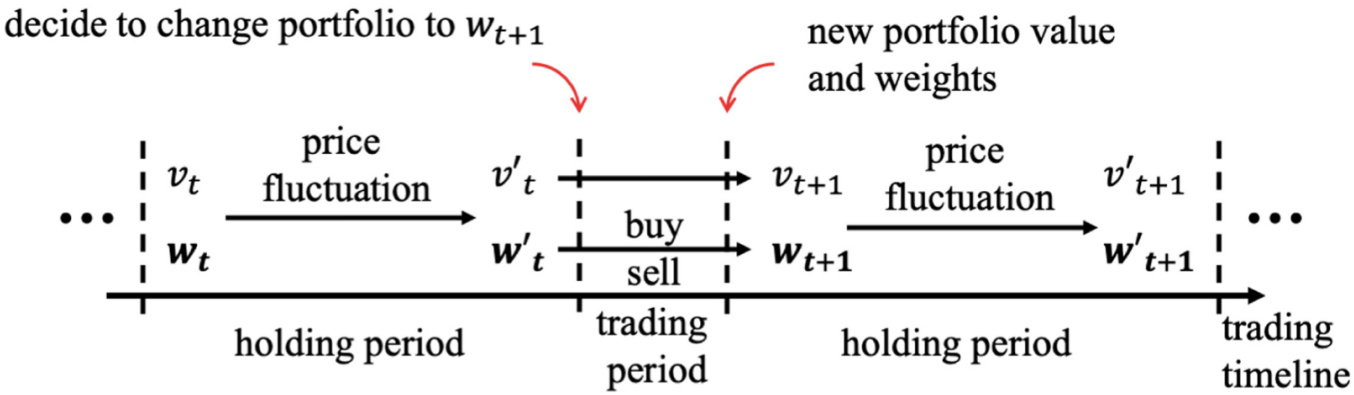

Fig. 2. Portfolio management process.

Existing Approaches to Portfolio Management (PM)

- **Benchmarks**:
  - Benchmarks like Constant Rebalanced Portfolio (CRP) and Uniform Constant Rebalanced Portfolio (UCRP) [25] provide baseline strategies for comparison.
- **Follow-the-Winner Approaches**:
  - Follow-the-Winner strategies such as Exponential Gradient (EG) [54] and Winner [45] focus on following trends and momentum in the market.

Existing Approaches to Portfolio Management (PM)

- **Follow-the-Loser Approaches**:
  - Follow-the-Loser methods like Robust Mean Reversion (RMR) [59], Passive Aggressive Online Learning (PAMR) [79], and Anti-Correlation [14] aim to capitalize on mean reversion and anti-correlation strategies.
- **Pattern-Matching-based Approaches**:
  - Pattern-Matching-based techniques such as correlation-driven nonparametric learning (CORN) [78] and BK [51] leverage pattern recognition to inform trading decisions.
- **Meta-Learning Algorithms**:
  - Meta-Learning algorithms like Online Newton Step (ONS) offer adaptive learning capabilities and flexibility in strategy development.

Our idea

Our idea is to build a bot that predicts stock using the following methods:

1. Using Q-learning to predict stocks.
2. Using real-world data from financial journals of the past 10 years.
3. Using Deep Learning Framework as in DQN [LNC20]

Preliminaries I

Definition of Z-score:
The z-score is a statistical measure that indicates how many standard deviations a data point is from the mean of a dataset. In the context of the stock market, it's commonly employed to assess the deviation of a stock's current price from its historical mean price.
Formula for Z-score:

$$Z = \frac{(X - \mu)}{\sigma}$$

Where:

- X represents the value being evaluated (e.g., current stock price).
- μ denotes the mean of the dataset (e.g., historical average stock price).
- σ signifies the standard deviation of the dataset.

Preliminaries II

Application of z-score in RL:
In reinforcement learning (RL) for stock market prediction, the z-score can serve as a valuable feature to inform trading decisions. Agents can use the z-score of various stock metrics (e.g., price, volume) as input features in their learning process. For instance, an RL agent could learn to buy or sell stocks based on the z-score of their current prices relative to historical averages. A high positive z-score might trigger a sell action, indicating potential overvaluation, while a low negative z-score might prompt a buy action, suggesting potential undervaluation.
By incorporating z-scores as features, RL agents can better capture market dynamics and adapt their strategies to exploit overvalued or undervalued stocks. This integration enables agents to learn more sophisticated trading policies, potentially leading to improved performance in stock market prediction tasks.[Bas+22]

FrameWorks I

Reinforcement learning (RL) frameworks for stock market predictions often utilize various algorithms and methodologies to make decisions based on rewards. Here are a few popular frameworks:

- **Q-Learning**: Q-learning in stock market prediction involves using a reinforcement learning algorithm to iteratively learn the optimal action-value function, enabling the agent to make informed decisions based on historical market data.
- **Deep Q-Network (DQN)**: Utilizes a deep neural network to approximate the Q-value function, enabling the agent to learn from historical stock market data and make decisions to maximize long-term rewards.
- **Policy Gradient Methods**: Directly optimize the policy function to maximize cumulative rewards. Algorithms like REINFORCE or Actor-Critic fall under this category.

FrameWorks II

- **Proximal Policy Optimization (PPO)**: Balances exploration and exploitation by updating the policy function while ensuring that the new policy doesn't deviate significantly from the previous one.
- **Deep Deterministic Policy Gradient (DDPG)**: Suitable for continuous action spaces, DDPG learns a deterministic policy function by leveraging deep neural networks for both the actor and critic.
- **Trust Region Policy Optimization (TRPO)**: Controls the policy updates to ensure that the new policy doesn't diverge too far from the previous one, enhancing stability during training.

These frameworks provide a structured approach for developing RL-based models for stock market predictions, each with its advantages and limitations, depending on the specific characteristics of the problem and the available data.

Z-Score Framework

Algorithm 1: Stock Trading Algorithm using Z-score

```
Input: mov_zscore_close, n, df
Output: money, stocks
begin
    for i ← n to length(df) do
        if mov_zscore_close[i − n] > 1 then
            while stocks > 0 do
                money += df['Close'][i];
                stocks -= 1;
        if mov_zscore_close[i − n] > 0.5 then
            if stocks > 0 then
                money += df['Close'][i];
                stocks -= 1;
        if mov_zscore_close[i − n] < -0.5 then
            if money > df['Close'][i] then
                money -= df['Close'][i];
                stocks += 1;
        if mov_zscore_close[i − n] < -1 then
            while money > df['Close'][i] do
                money -= df['Close'][i];
                stocks += 1;
```

Q-Learning Framework

1. $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma max_{a'} Q(s',a') - Q(s,a))$
2. Given a set of information, we need to decide on an action (Buy, Sell, or Hold).
3. We considered the state space where we have the information about the number of stocks we have presently and the time passed since our first investment.
4. We considered learning rate 0.1, discount factor 0.9 and reward as the difference between the money at time t and the money at time $t-1$.
5. Currenlty in this framework, at every time step, we can buy or sell at most 1 stock.

DQN FrameWork

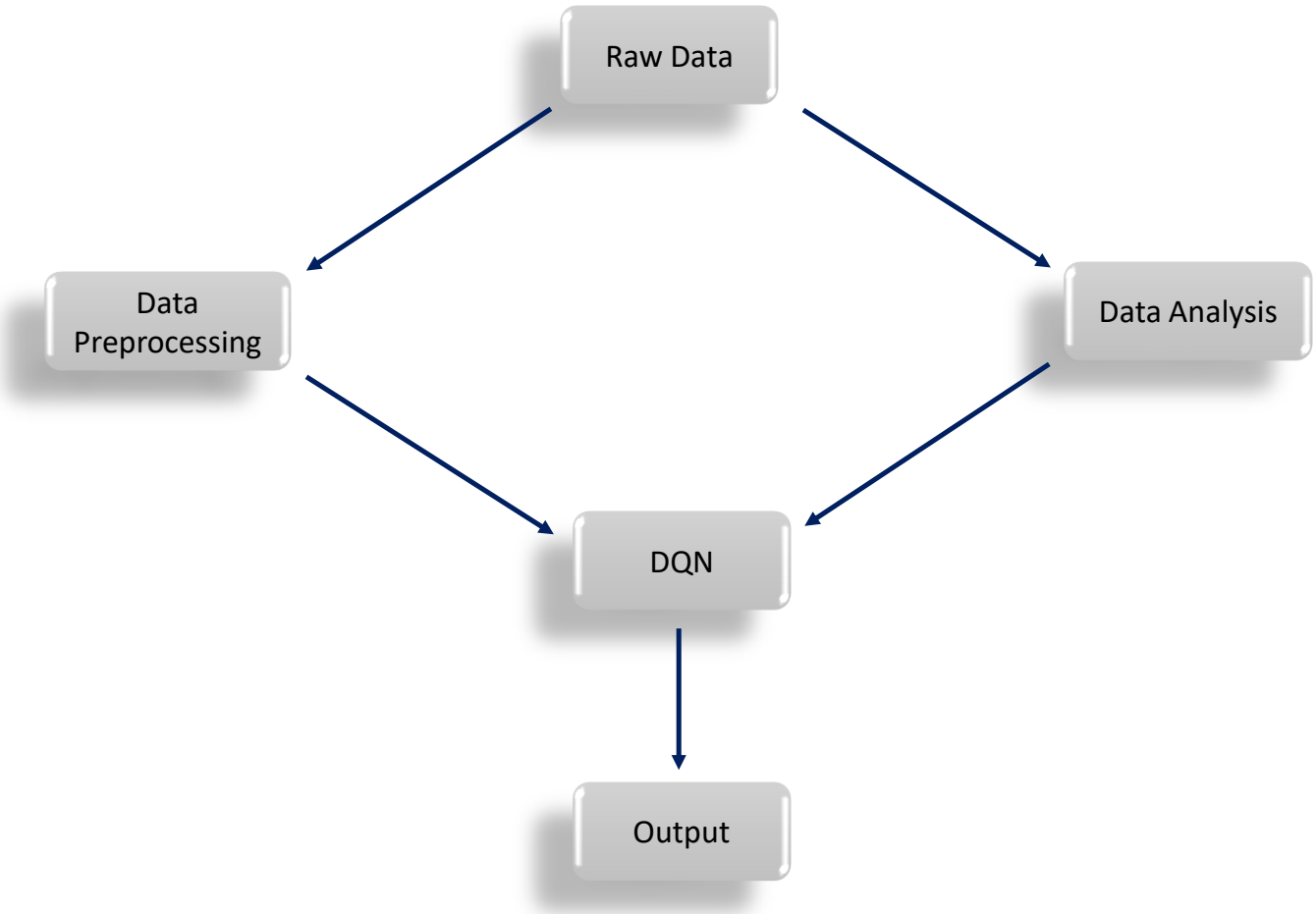

Figure: FrameWork for RL in Deep Q Network

Structure of DQN

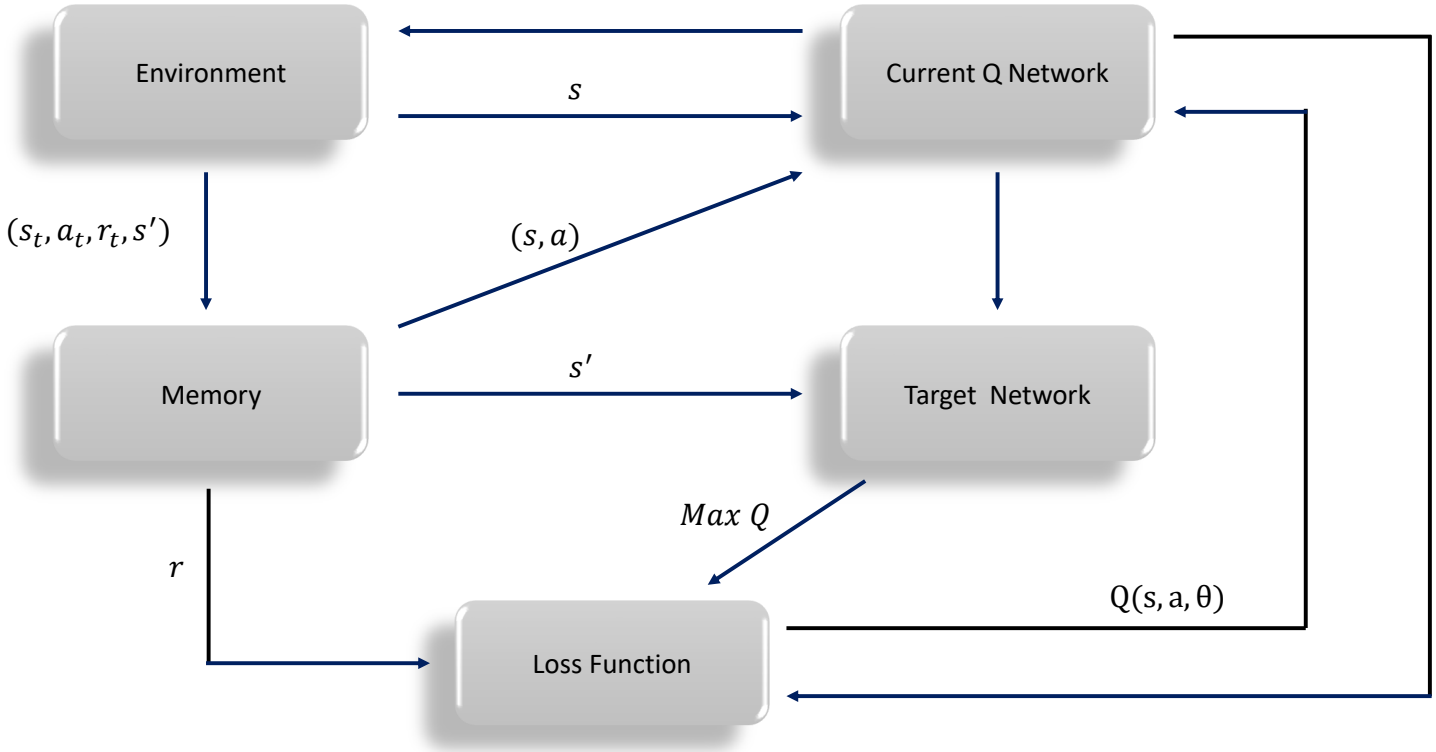

Figure: Structure of Deep Q Network

Results & Obseravtions

The Results of our work which have been implemented by us till now are:

1. Using z-score only (without learning)
2. Using Q-learning

Using Z-score

1. We took the data of the stock prices of Google, Tesla and TCS.
2. We ran the algorithm on the data ranging till today.
3. We then checked how the performances as net income till date and average net income.

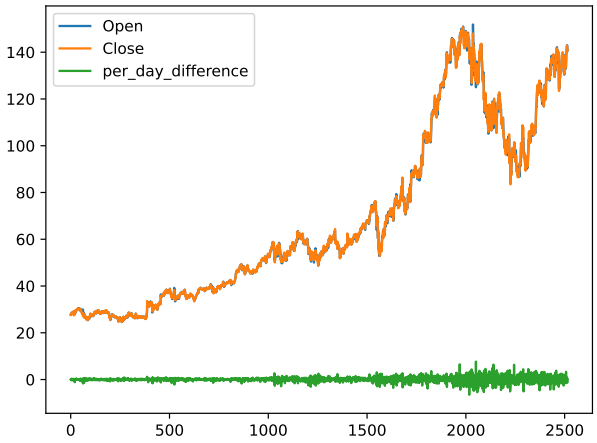

Figure: Google

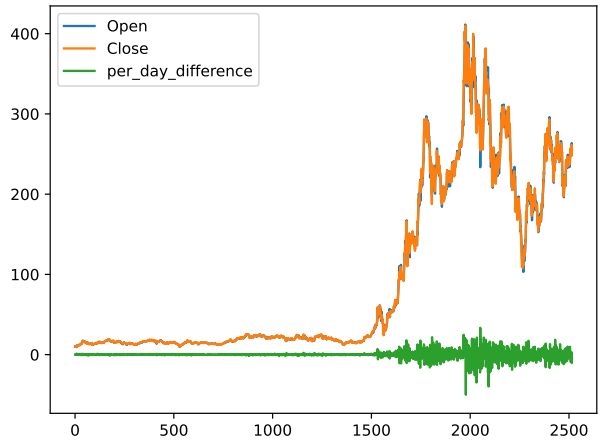

Figure: Tesla

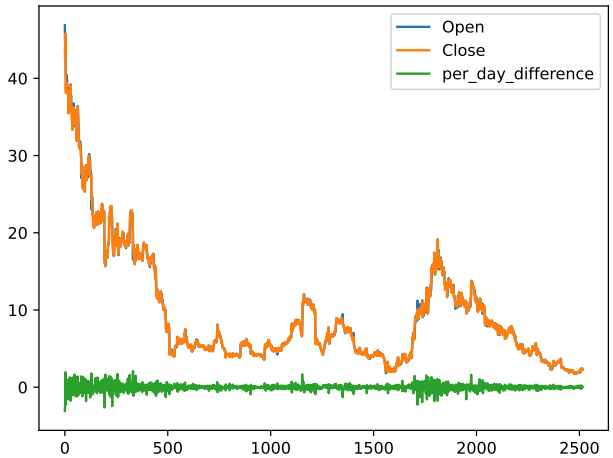

Figure: TCS

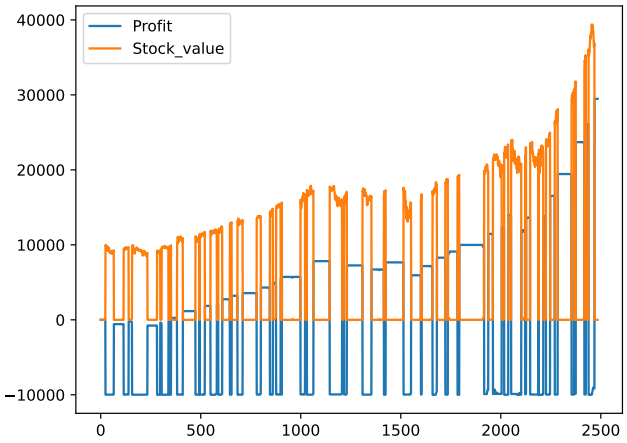

Figure: Google

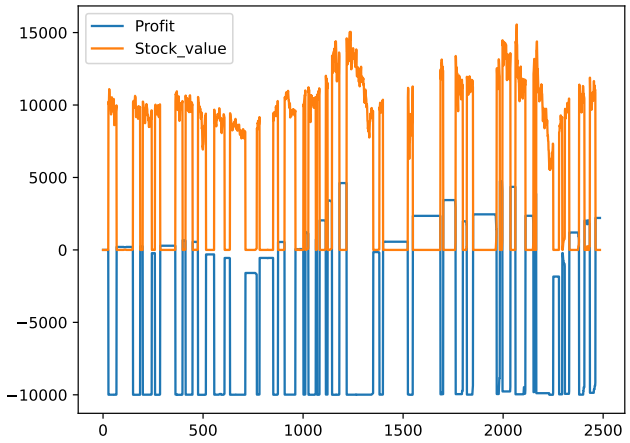

Figure: Tesla

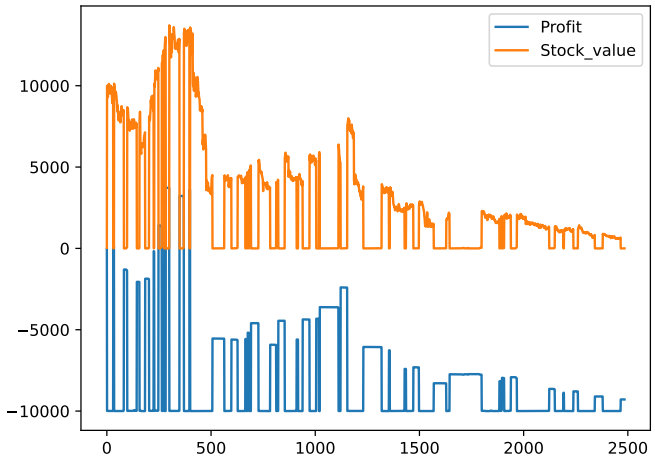

Figure: TCS

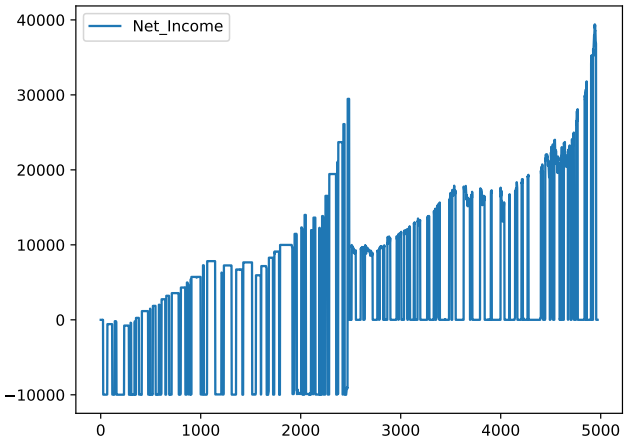

Figure: Google

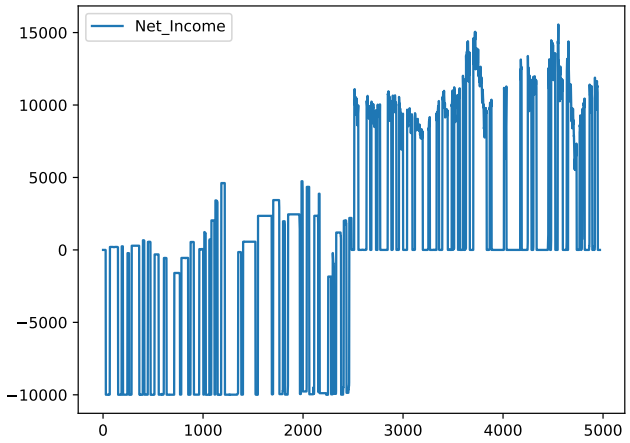

Figure: Tesla

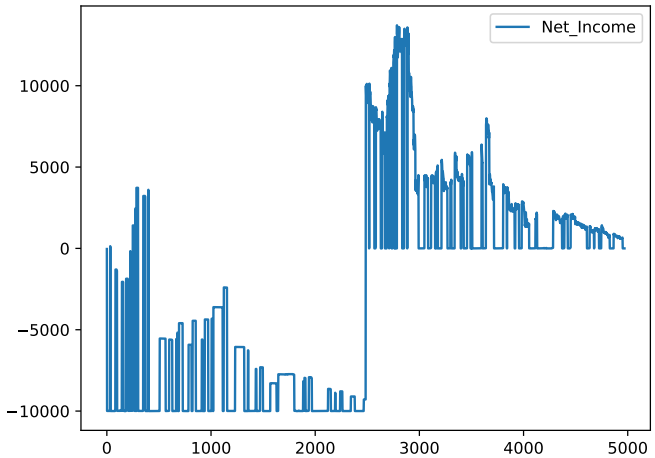

Figure: TCS

In Q-Learning FrameWork

1. We took the data of the stock prices of Google.
2. We ran the algorithm on the data ranging from 2014 to 2019.
3. We then checked how the Q learning performs on data after 2019.

In Q Learning Framework

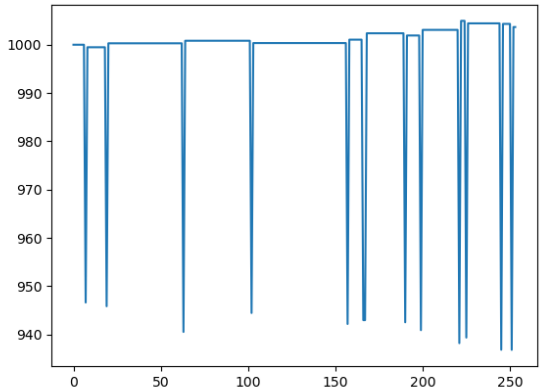

Figure: Performance for 2019-2020

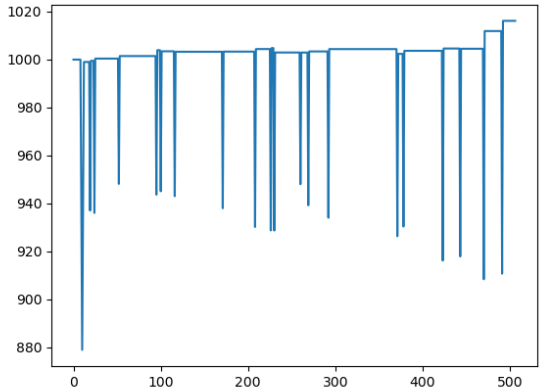

Figure: Performance for 2019-2021

In Q Learning Framework

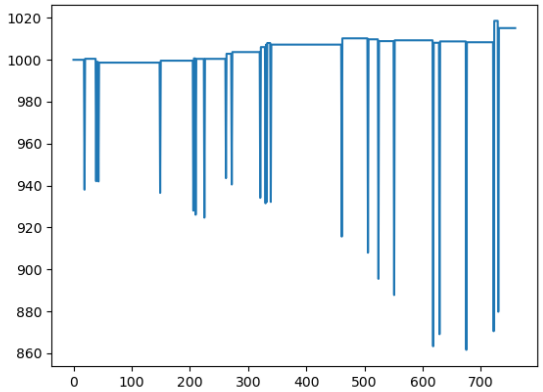

Figure: Performance for 2019-2022

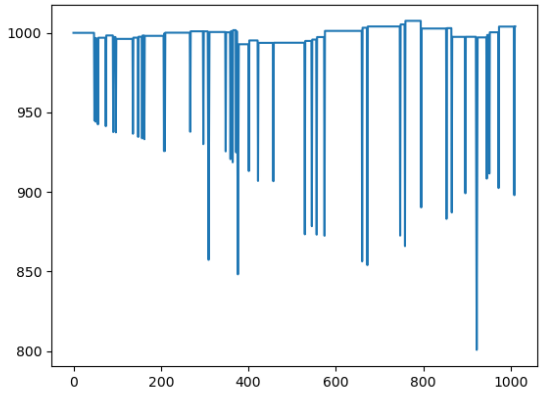

Figure: Performance for 2019-2023

References I

[Bas+22] Prakash Basanna et al. "Relevance of Ratios in Z-score Model for Predicting Bankruptcy: Study of Nifty PSES". In: *International Journal of Mechanical Engineering* 7 (2022).

[LNC20] Yuming Li, Pin Ni, and Victor Chang. "Application of deep reinforcement learning in stock trading strategies and stock forecasting". In: *Computing* 102 (2020), pp. 1305–1322.