RL Project Monsoon 2021

Ansh Arora (2019022)
Devansh Gupta (2019160)
Sudarshan Buxy (2019279)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

TD3QN on Algorithmic Trading

Github Link: https://github.com/arora-ansh/DeepRL-AlgoTrading

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Environment, Agent's Description, and Reward

- The environment is a simulation of a given company's stock. It has been modelled as a gym environment. The portfolio of an agent at any given time is the value the agent holds in that one particular stock + cash. Buying and selling operations form the actions, which in this case are cash and share exchanges. The agent interacts with the environment which is presented to it as an order book containing bid and ask quantities with quantities and prices.
- The trading timeline has been discretized, in the form of single day quantums i.e. the agent takes one trading decision everyday.
- The rewards are taken to be strategy daily returns, which makes the reward independent of number of shares held.

$$r_t = \frac{v_{t+1} - v_t}{v_t}$$

Modelling the RL Problem

- The trading strategy is viewed as the policy. Action a_t resulting from its RL policy $\pi(a_t|h_t)$ where h_t is the RL agent's history and receives a reward r_t as a consequence of its action. Agent history can be expressed as $h_t = \{(o_\tau, a_\tau, r_\tau)|\tau = 0,1,\ldots,t\}$.
- The reduced observation space element taken by the environment can be presented as as $o_t = \{S(t'), D(t'), P_{t'}\}_{t'=t-\tau}$ where S(t) represents the state information (current trading position, number of shares owned by the agent, available cash), D(t) is information gathered by the agent at time t in the OHLCV (Open-High-Low-Close-Volume) format. P_t is the current trading position of the agent (long or short).

Modelling the RL Problem

- A single action is modelled as the number of shares to be bought (+ve value), number of shares to be sold (-ve value) and if shares to be held (0 value).
- To reduce the action space, two actions are considered $a_t = Q_t \in \{Q_t^{Long}, Q_t^{Short}\}$, where Q^{long} maximizes the number of shares owned by the agent, by converting as much cash value as possible into share value. Q^{short} maximizes the cash owned by the agent by converting as much share value as possible to cash value.
- The main objective of the agent is to maximize the Sharpe Ratio

$$S_r = \frac{\mathbb{E}[R_s - R_f]}{\sigma_r} = \frac{\mathbb{E}[R_s - R_f]}{\sqrt{\text{var}[R_s - R_f]}} \simeq \frac{\mathbb{E}[R_s]}{\sqrt{\text{var}[R_s]}} \quad (20)$$

where:

- R_s is the trading strategy return over a certain time period, modelling its profitability.
- R_f is the risk-free return, the expected return from a totally safe investment (negligible).
- σ_r is the standard deviation of the trading strategy excess return $R_s - R_f$, modelling its riskiness.

Our Baseline: Trading Deep Q-Network(TDQN)

- The baseline involved the use of Deep Q-Networks to learn a trading strategy in order to maximize the Sharpe Ratio (and a few other performance indicators such as profitability and average return)

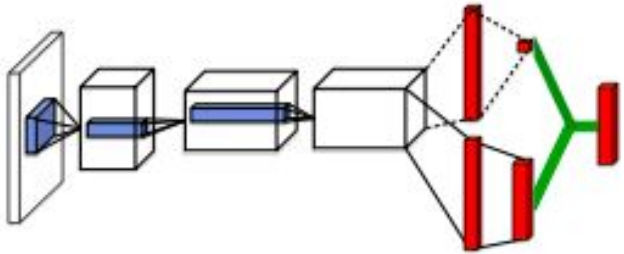
- The Trading DQN was used to determine the optimal trading strategy for given stocks, a starting capital of $100000 and its quantitative performance was measured with the help of the indicators and the plots.

Double Deep Dueling Architectures for Reinforcement Learning

- Introducing a new “Advantage term”

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$$

- We model advantage term and value term as deep neural networks and aggregate it using an aggregation function



- The aggregation function which can be any one of

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \max_{a' \in |\mathcal{A}|} A(s,a';\theta,\alpha)\right).$$

or

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a';\theta,\alpha)\right).$$

Experimental Setup

- We tried three different methods
  - TDQN: Double Deep Q-Network - Baseline
  - TD3QN - max advantage “baseline”: Double Deep Dueling Q-Network with the aggregating function

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \max_{a' \in |\mathcal{A}|} A(s,a';\theta,\alpha) \right).$$

  - TD3QN - mean advantage “baseline”: Double Deep Dueling Q-Network with the aggregating function

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s,a';\theta,\alpha) \right).$$

Experimental Setup

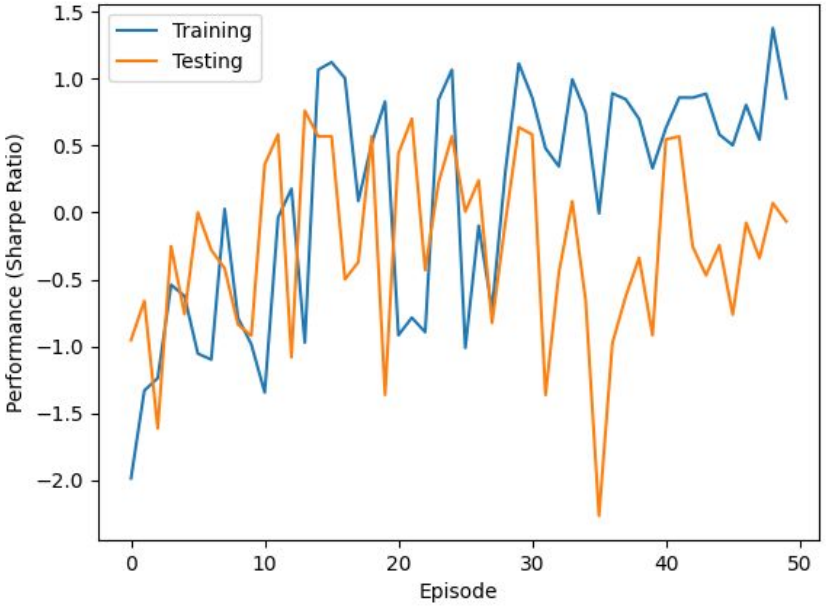
- Ran these algorithms on the price patterns of many stocks and show our detailed results on the four stocks: Google, Microsoft, Twitter, Apple separately which broadly encompasses all the cases encountered when running for other stocks
- Used an environment^ made for stock trading which simulates the stock data from real life datasets given and has a way for creating and adding new strategies and loading them in the system without changing the original codebase*.
- Evaluating the stock trading performance based on Profit & Loss (P&L), Annualized Return, Annualized Volatility, Sharpe Ratio, Maximum Drawdown, Maximum Drawdown Duration, and Profitability

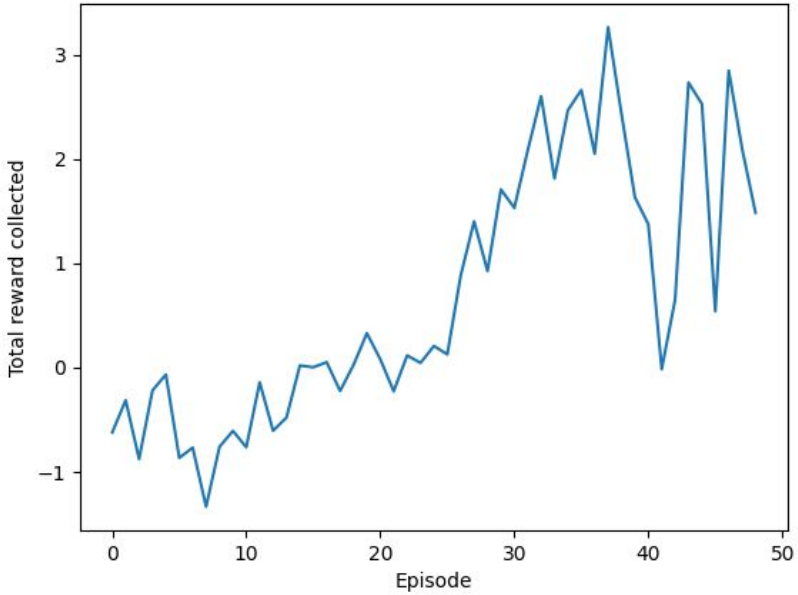
https://github.com/ThibautTheate/An-Application-of-Deep-Reinforcement-Learning-to-Algorithmic-Trading *^

Google Stock - Cumulative Results

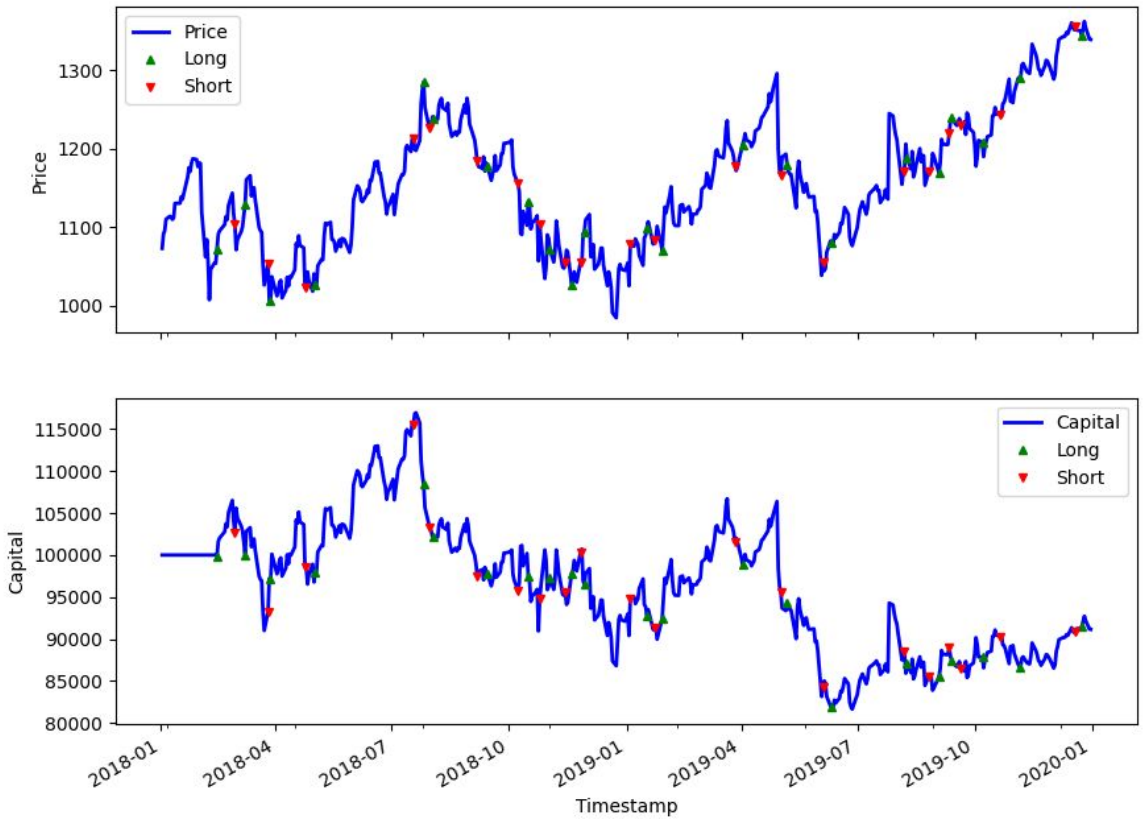
GOOGLE	TD3QN-Max	TD3QN-Average	TDQN
Profit & Loss (P&L)	-14836	**24702**	-8849
Annualized Return	-5.06%	**13.26%**	-1.65%
Annualized Volatility	25.01%	24.79%	**24.56%**
Sharpe Ratio	-0.197	**0.57**	-0.066
Maximum Drawdown	30.82%	**23.37%**	30.18%
Maximum Drawdown Duration	354 days	**104 days**	235 days
Profitability	25.00%	**100.00%**	39.53%

Google Stock - TDQN

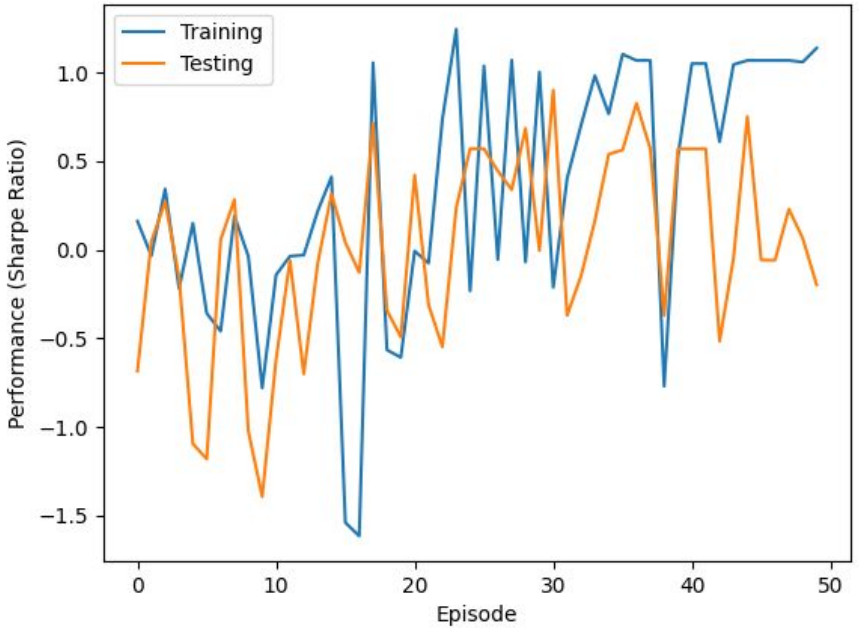


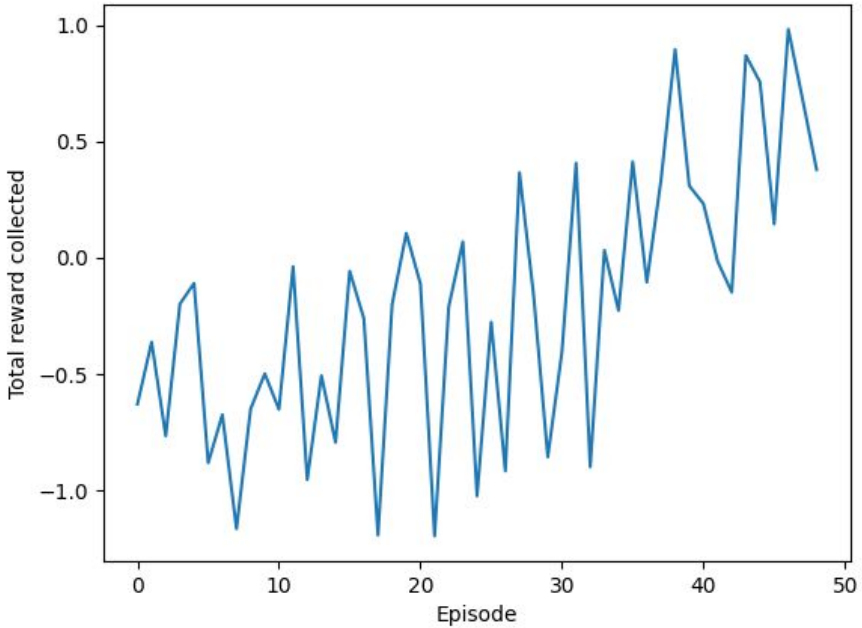


Google Stock - TDQN



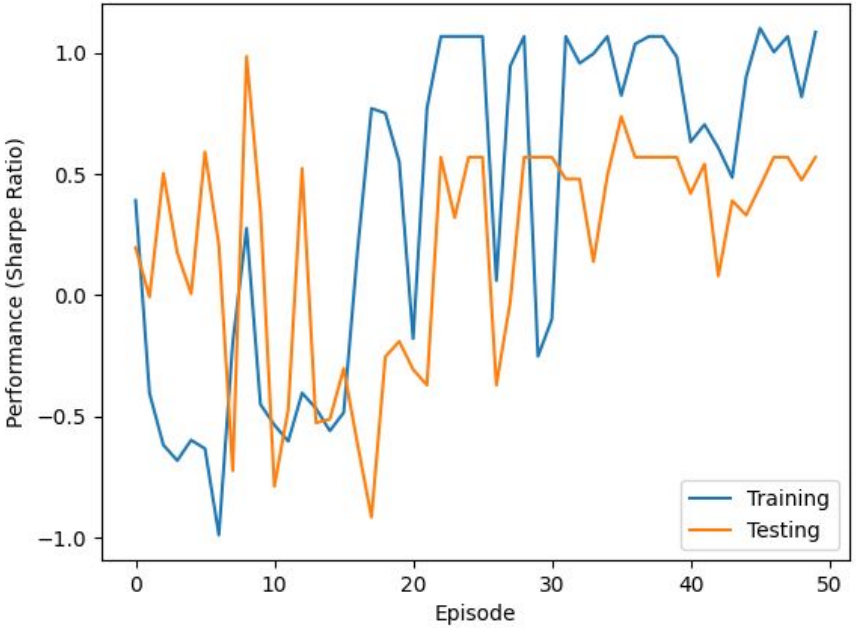
Google Stock - TD3QN with Max-Advantage-Baseline

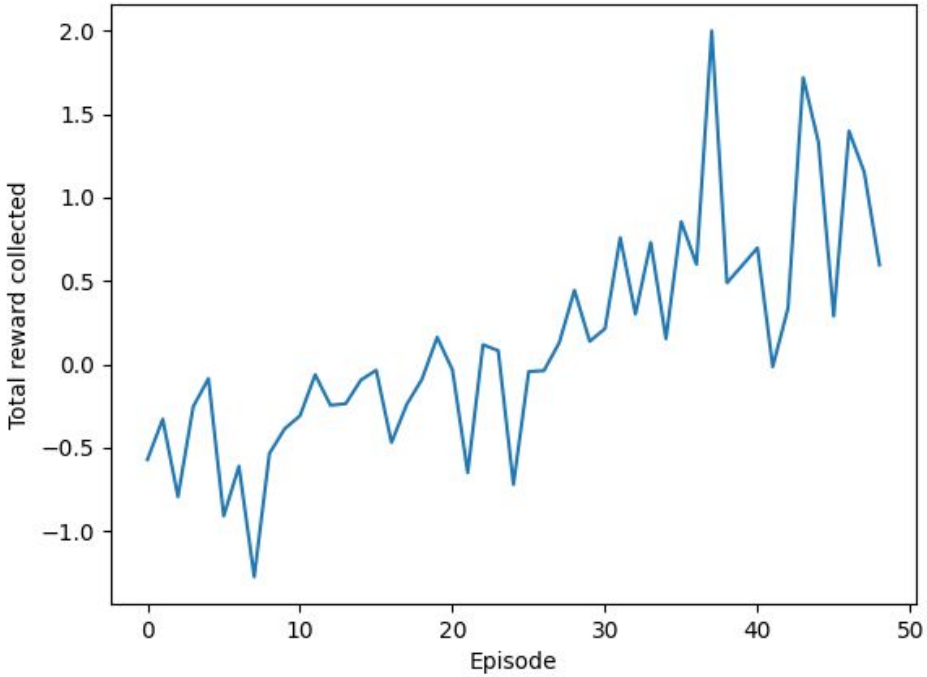




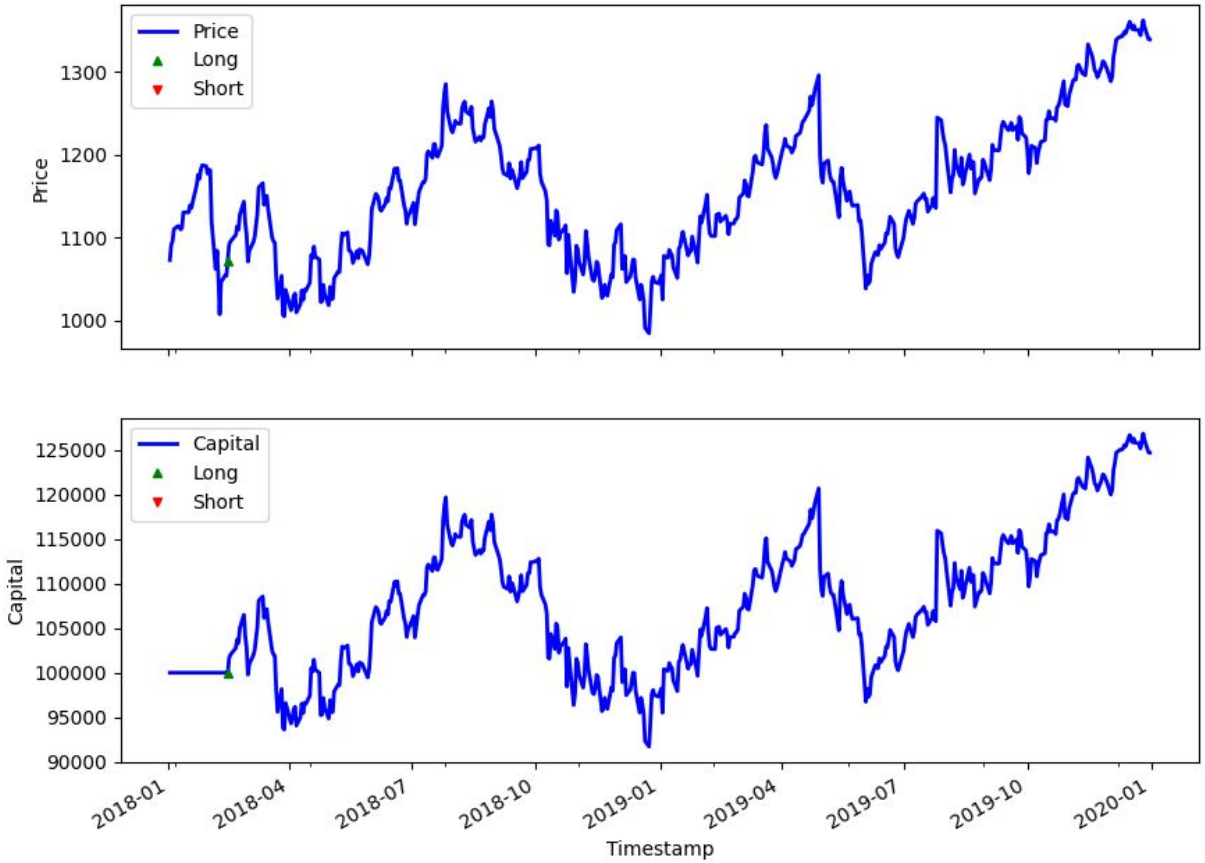
Google Stock - TD3QN with Max-Advantage-Baseline

Google Stock - TD3QN with Avg.-Advantage-Baseline





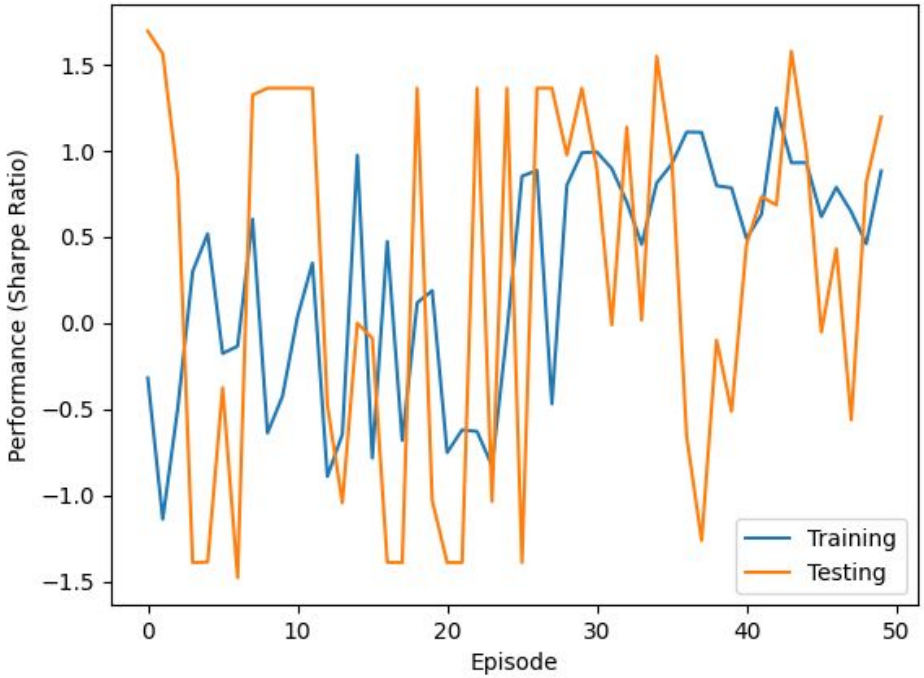
Google Stock - TD3QN with Avg.-Advantage-Baseline

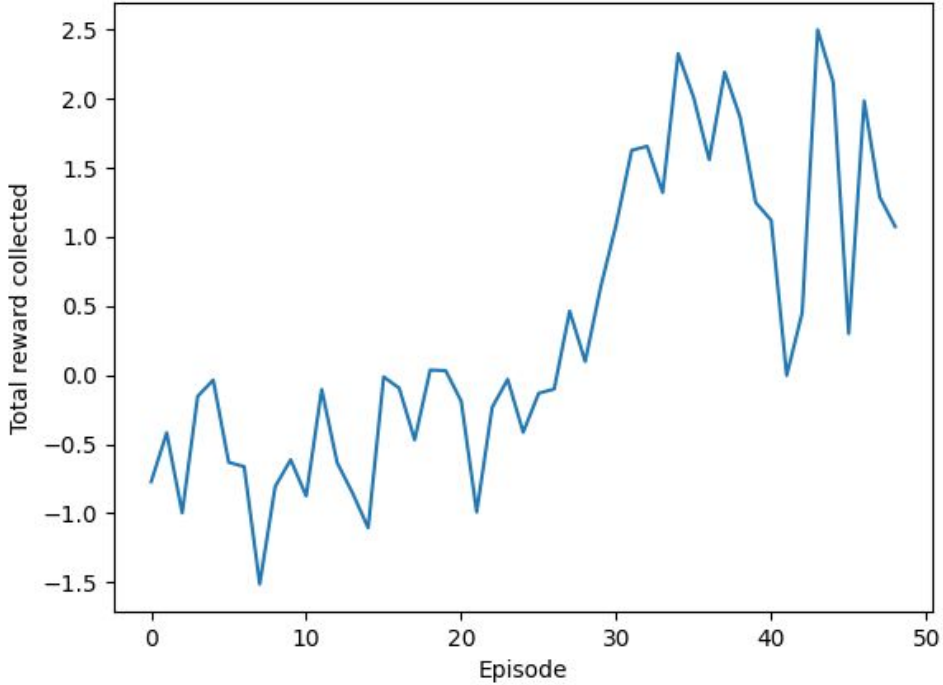


Microsoft Stock - Cumulative Results

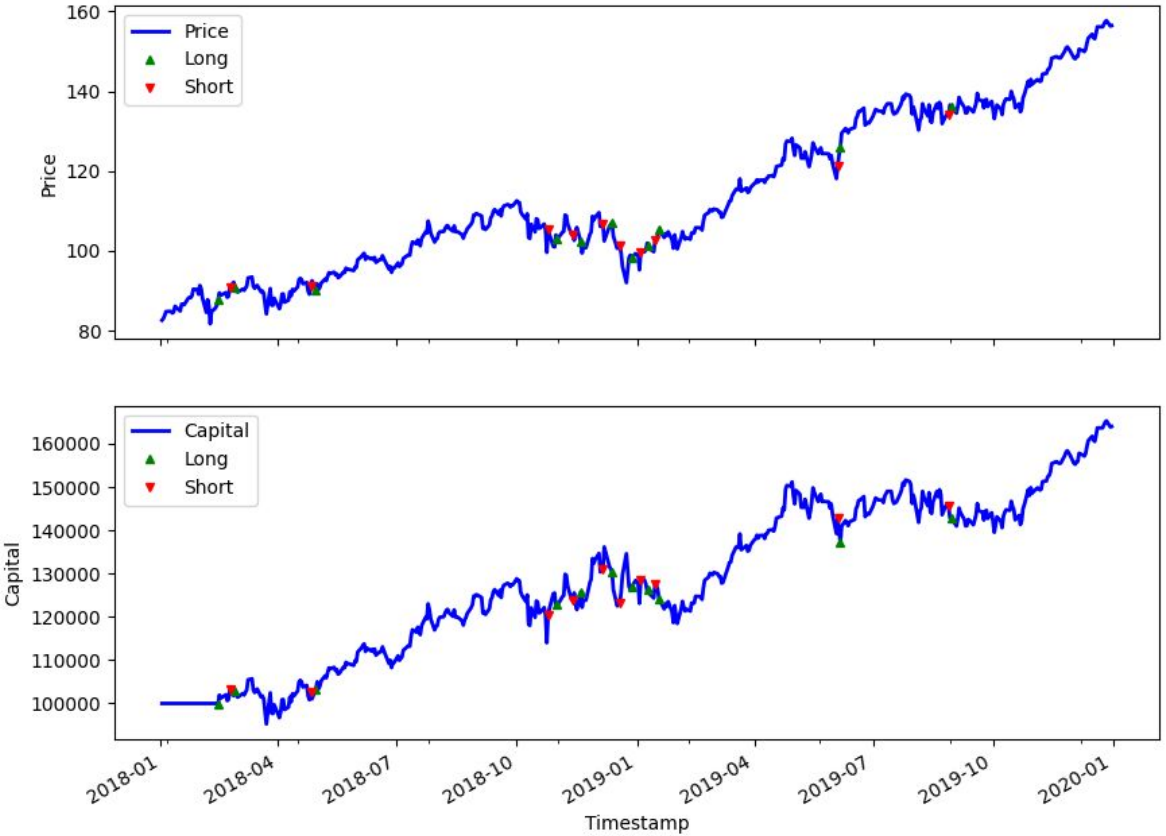
MICROSOFT	TD3QN-Max	TD3QN-Average	TDQN
Profit & Loss (P&L)	**78236**	**78236**	63975
Annualized Return	**27.83%**	**27.83%**	24.45%
Annualized Volatility	23.21%	23.21%	**22.89%**
Sharpe Ratio	**1.364**	**1.364**	1.197
Maximum Drawdown	**18.22%**	**18.22%**	12.98%
Maximum Drawdown Duration	58 days	58 days	**37 days**
Profitability	**100%**	**100%**	61.90%

Microsoft Stock - TDQN

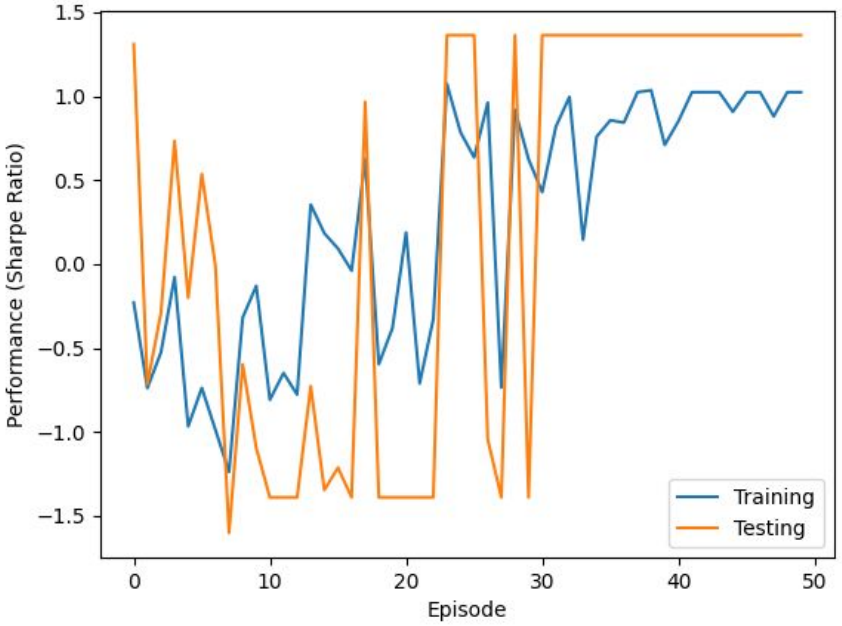


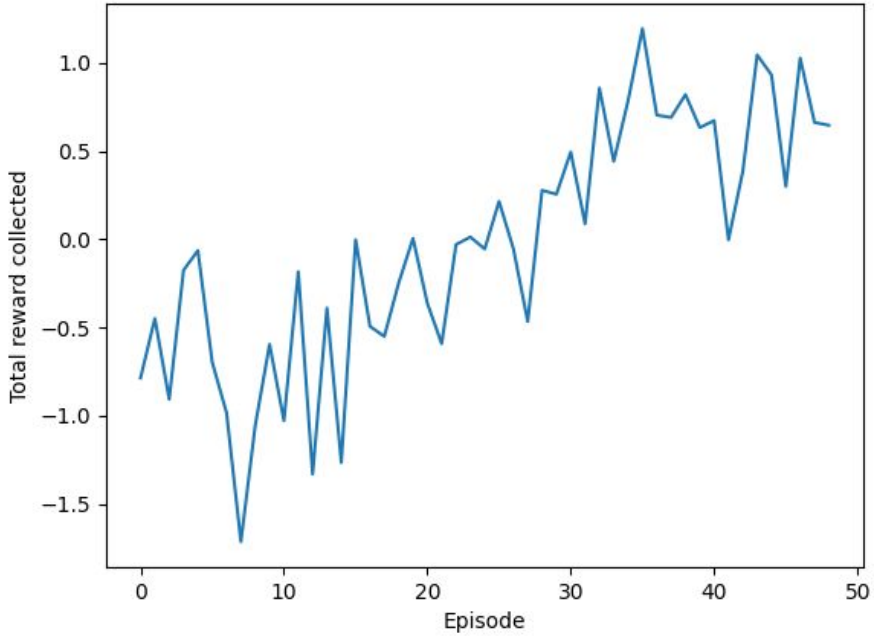


Microsoft Stock - TDQN

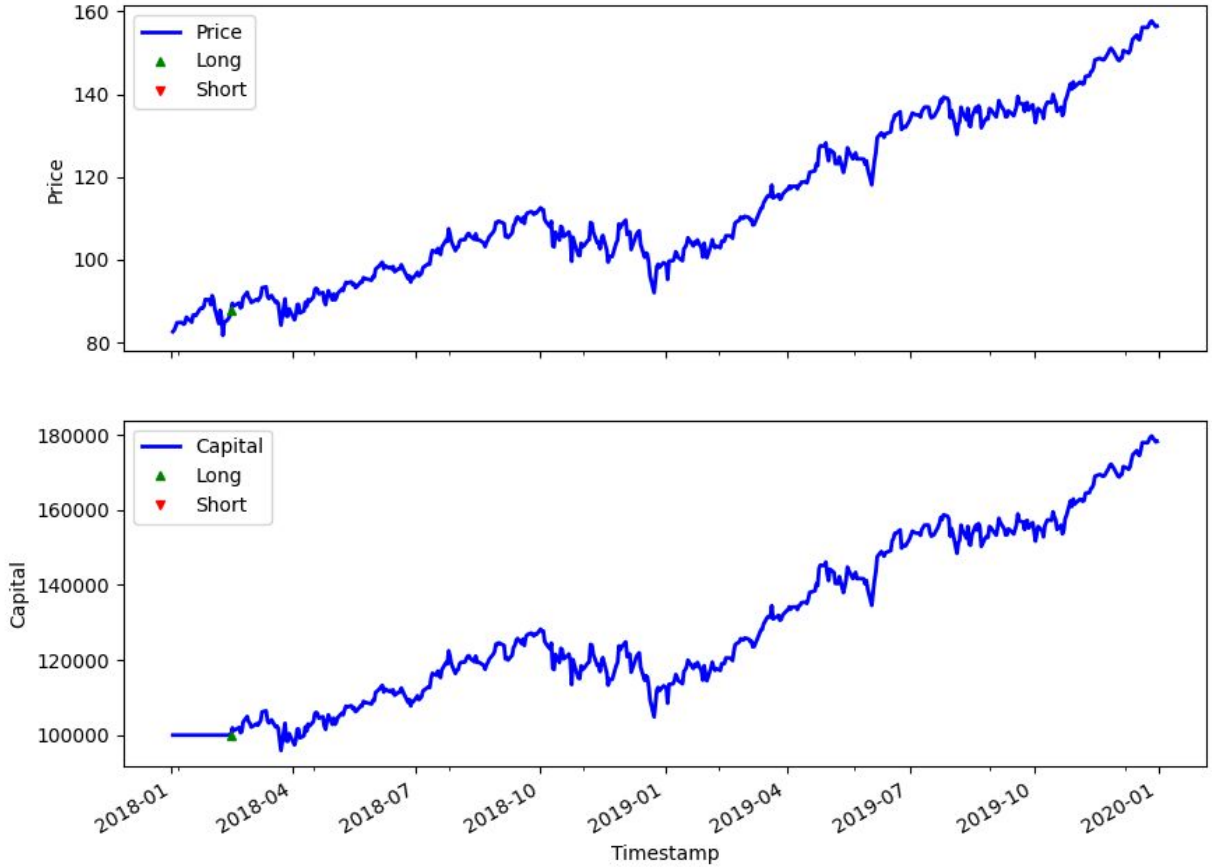


Microsoft Stock - TD3QN with Max-Advantage-Baseline

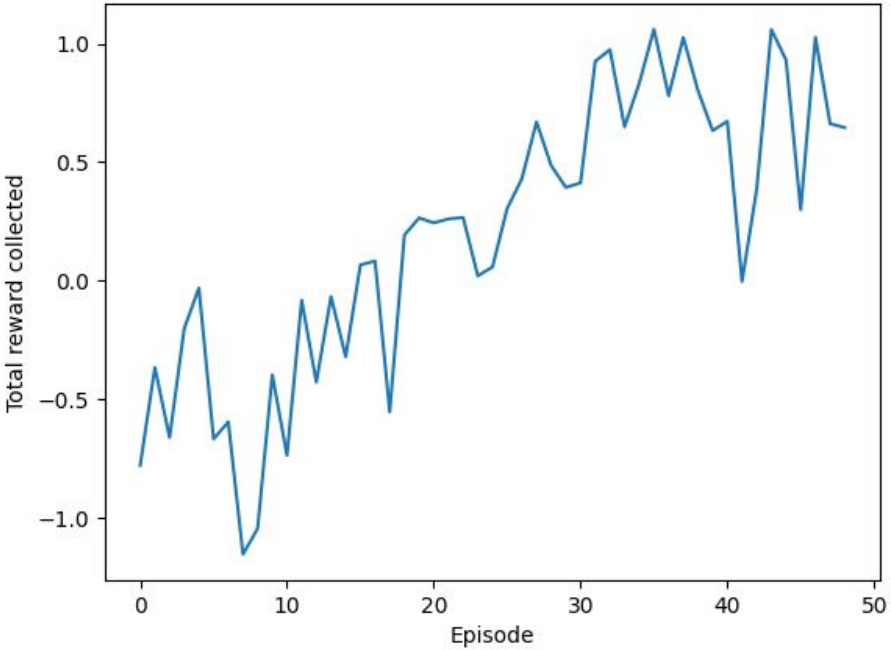


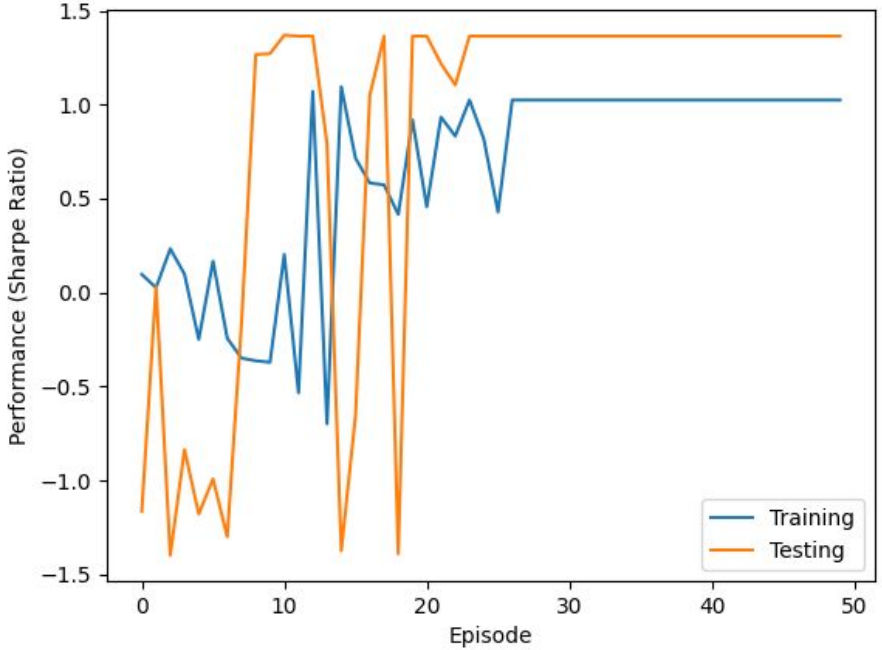


Microsoft Stock - TD3QN with Max-Advantage-Baseline

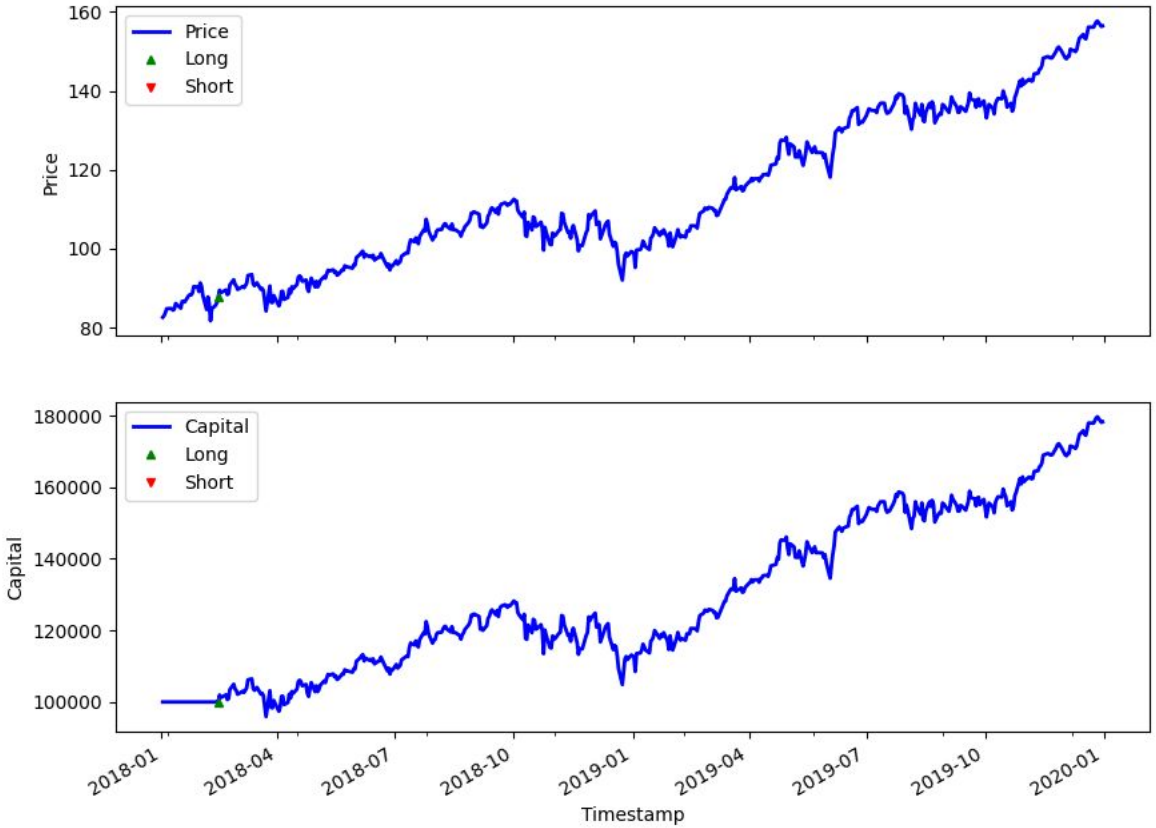


Microsoft Stock - TD3QN with Avg.-Advantage-Baseline





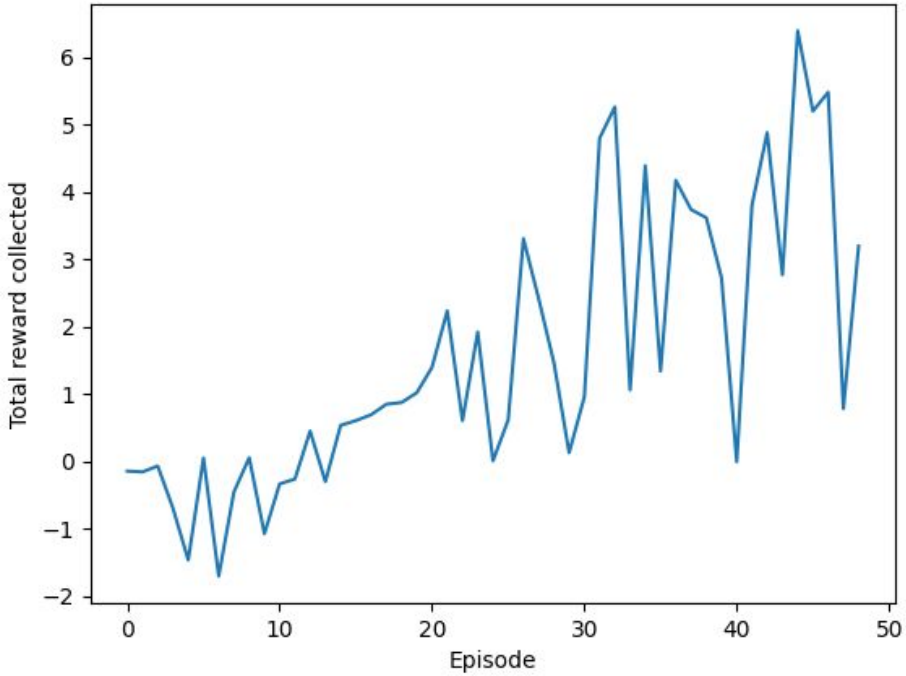
Microsoft Stock - TD3QN with Avg.-Advantage-Baseline

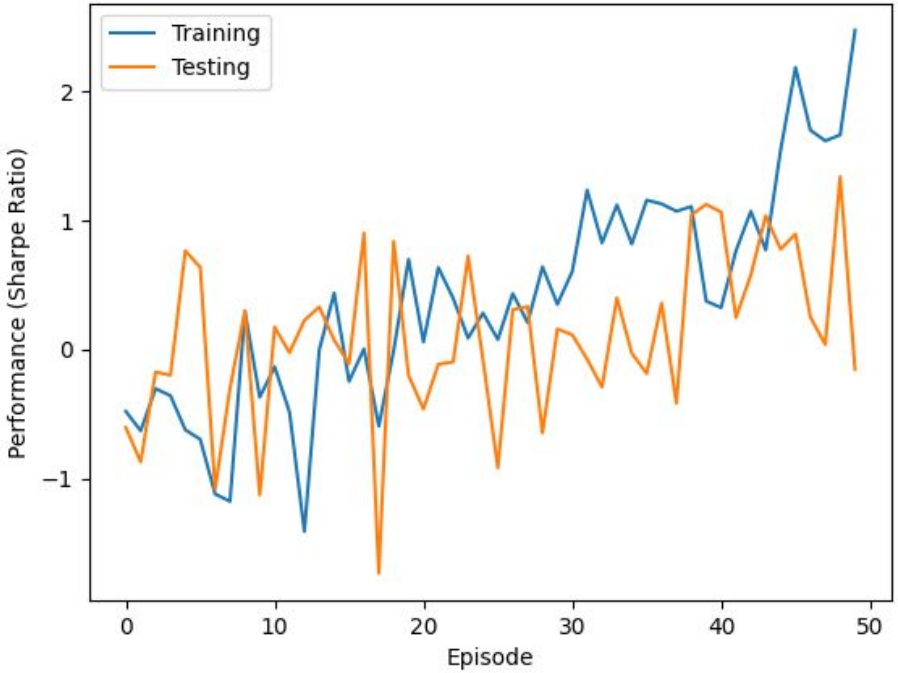


Twitter Stock - Cumulative Results

TWITTER	TD3QN-Max	TD3QN-Average	TDQN
Profit & Loss (P&L)	-3240	**14493**	-30104
Annualized Return	9.21%	**16.06%**	-7.38%
Annualized Volatility	47.37%	46.41%	**46.29%**
Sharpe Ratio	0.203	**0.373**	-0.153
Maximum Drawdown	65.59%	**40.67%**	61.91%
Maximum Drawdown Duration	319 days	**60 days**	249 days
Profitability	45.61%	**48.84%**	48.78%

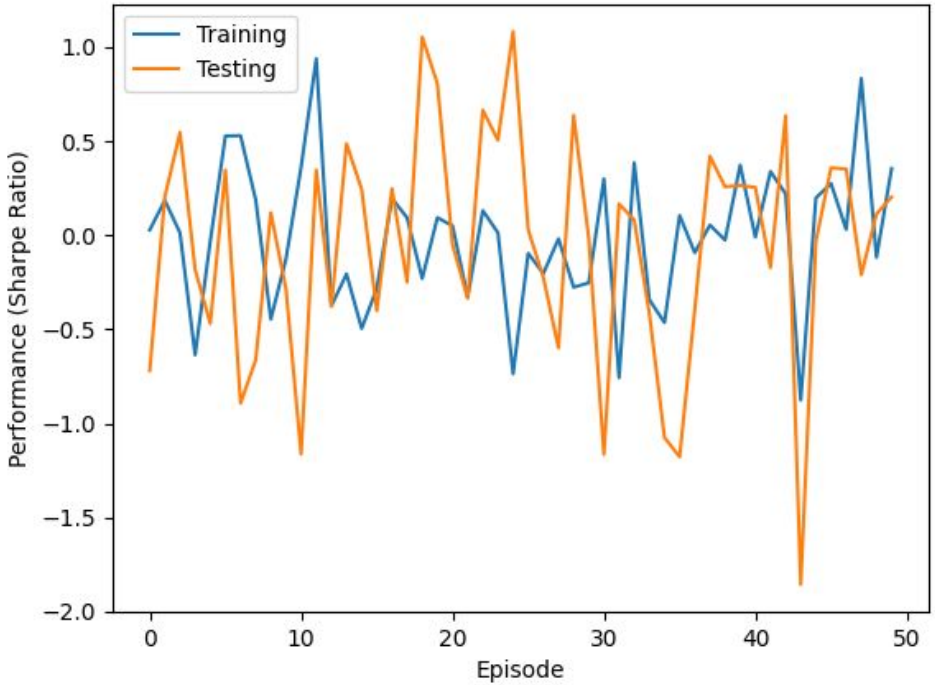
Twitter Stock - TDQN

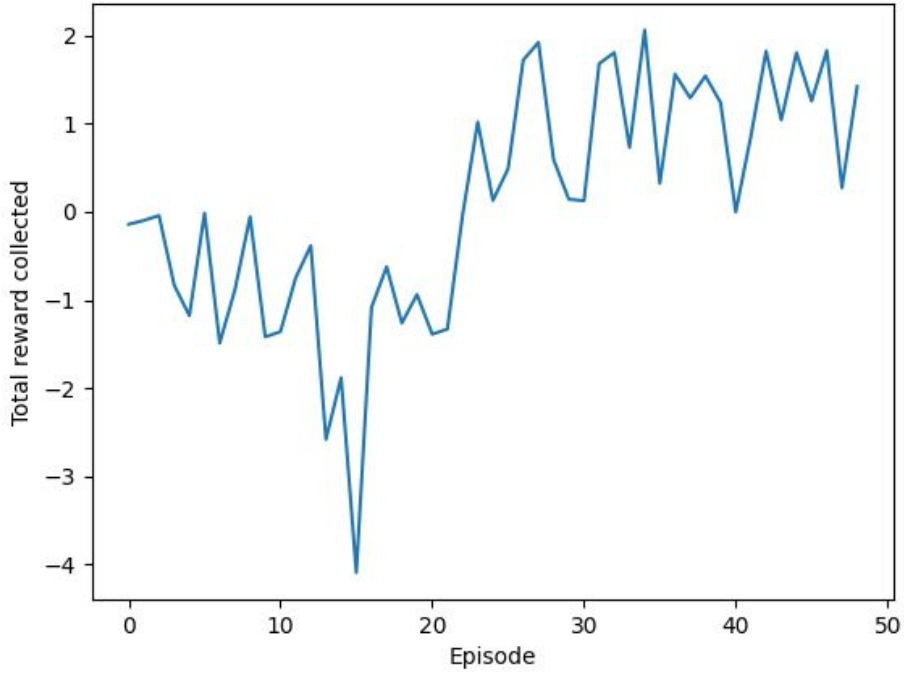




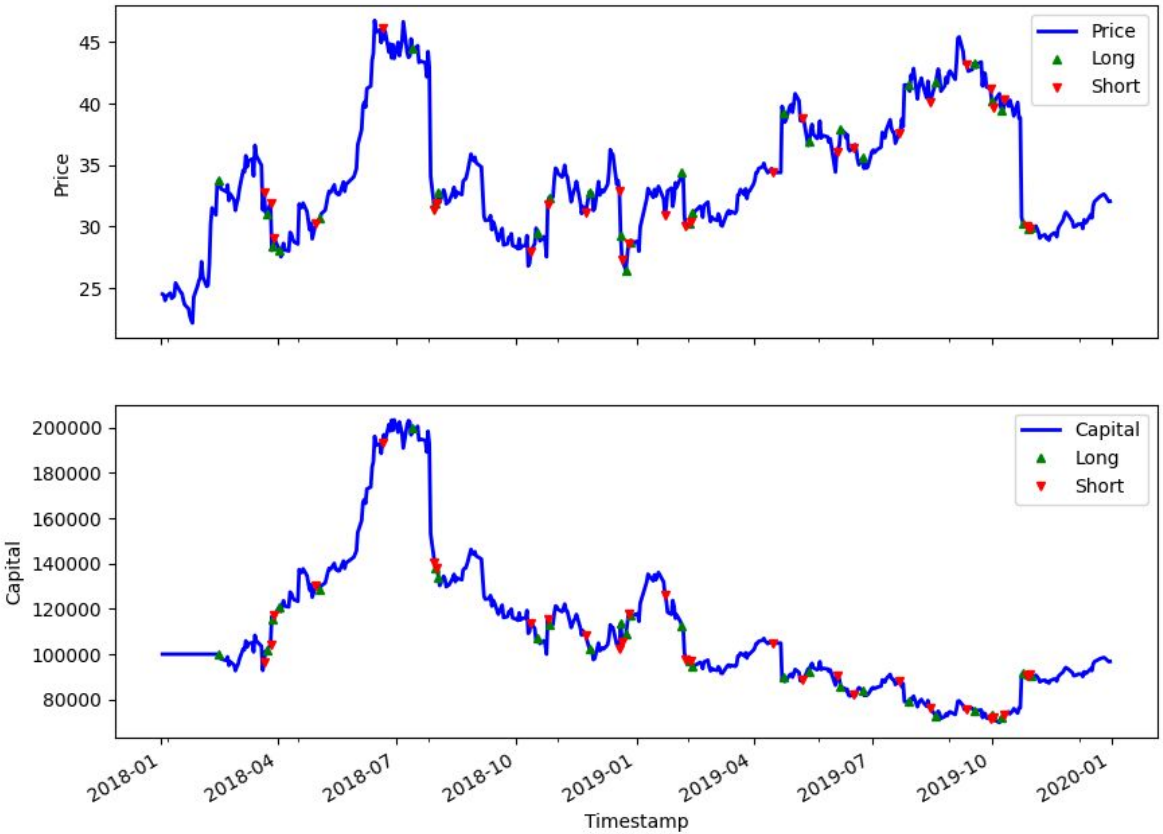
Twitter Stock - TDQN

Twitter Stock - TD3QN with Max-Advantage-Baseline

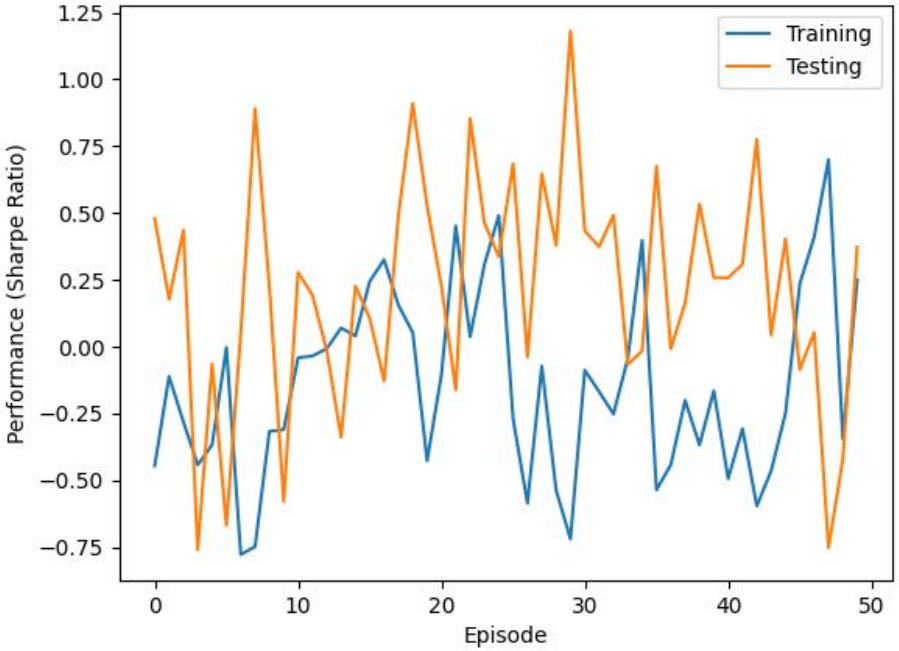


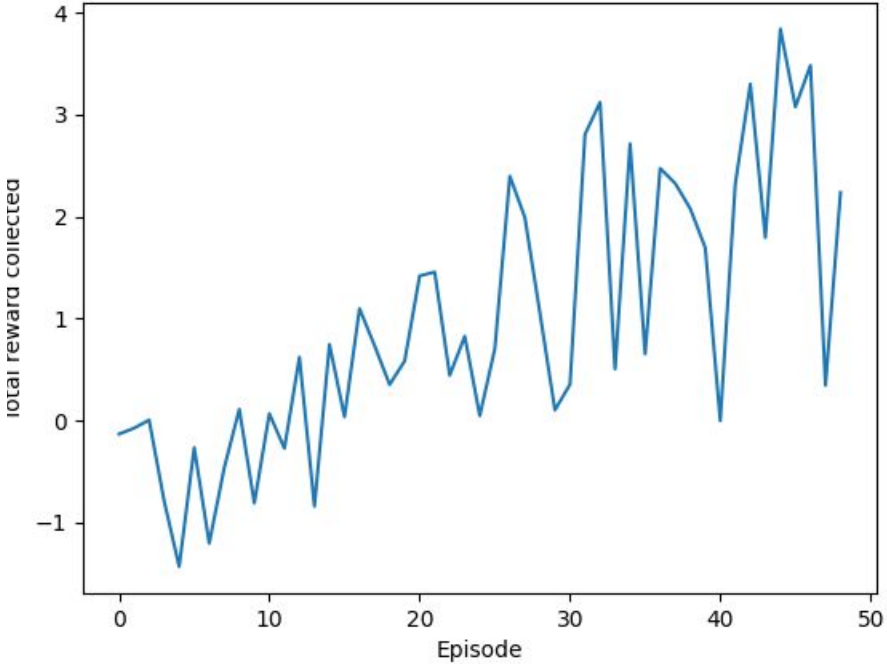


Twitter Stock - TD3QN with Max-Advantage-Baseline

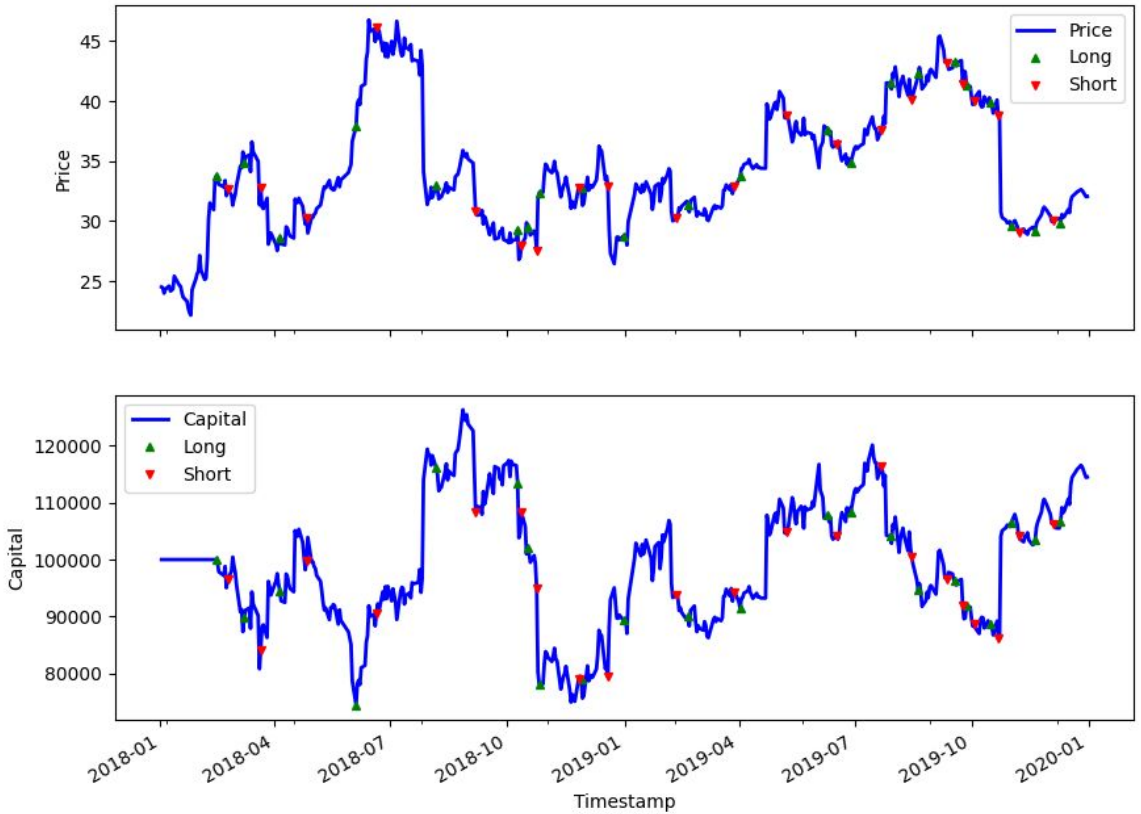


Twitter Stock - TD3QN with Avg.-Advantage-Baseline





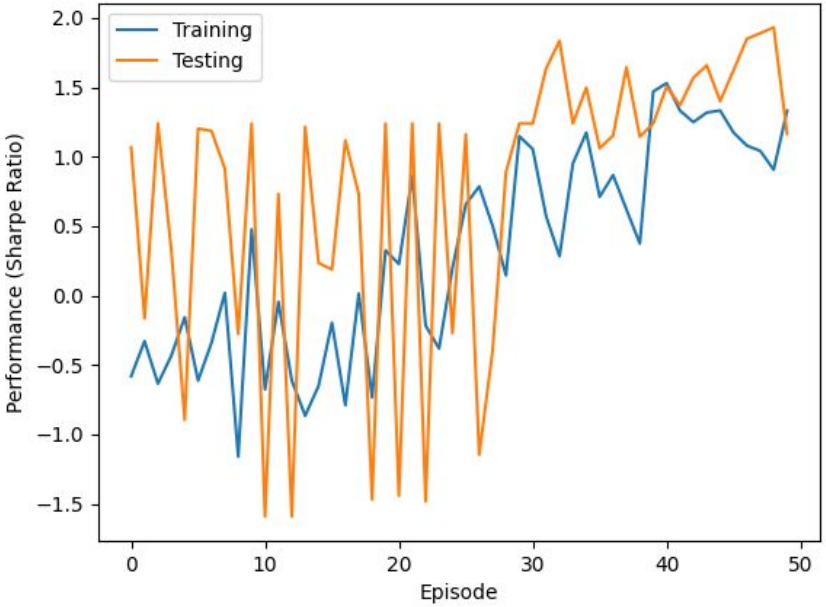
Twitter Stock - TD3QN with Avg.-Advantage-Baseline

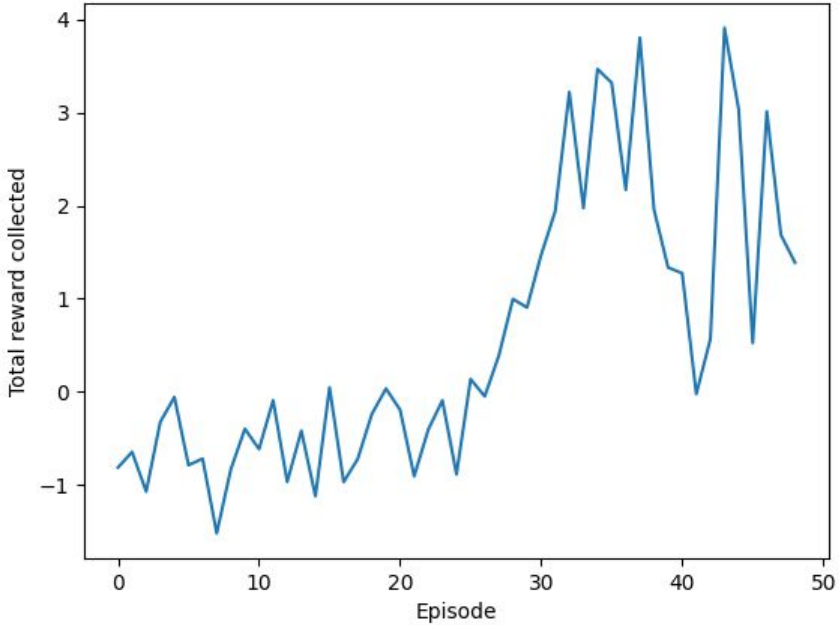


Apple Stock - Cumulative Results

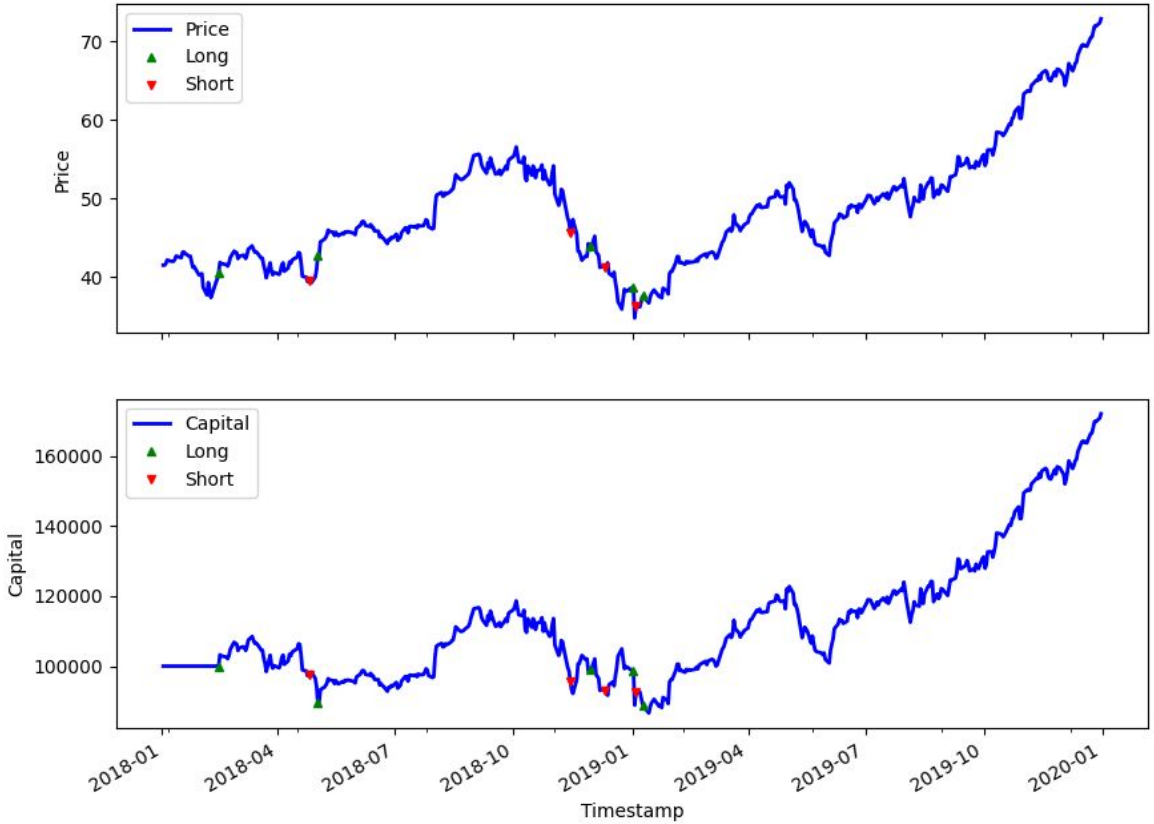
APPLE	TD3QN-Max	TD3QN-Average	TDQN
Profit & Loss (P&L)	**79823**	**79823**	72072
Annualized Return	**28.86%**	**28.86%**	27.08%
Annualized Volatility	26.62%	26.62%	**26.38%**
Sharpe Ratio	**1.239**	**1.239**	1.164
Maximum Drawdown	38.51%	38.51%	**27.00%**
Maximum Drawdown Duration	**62 days**	**62 days**	69 days
Profitability	**100%**	**100%**	44.44%

Apple Stock - TDQN

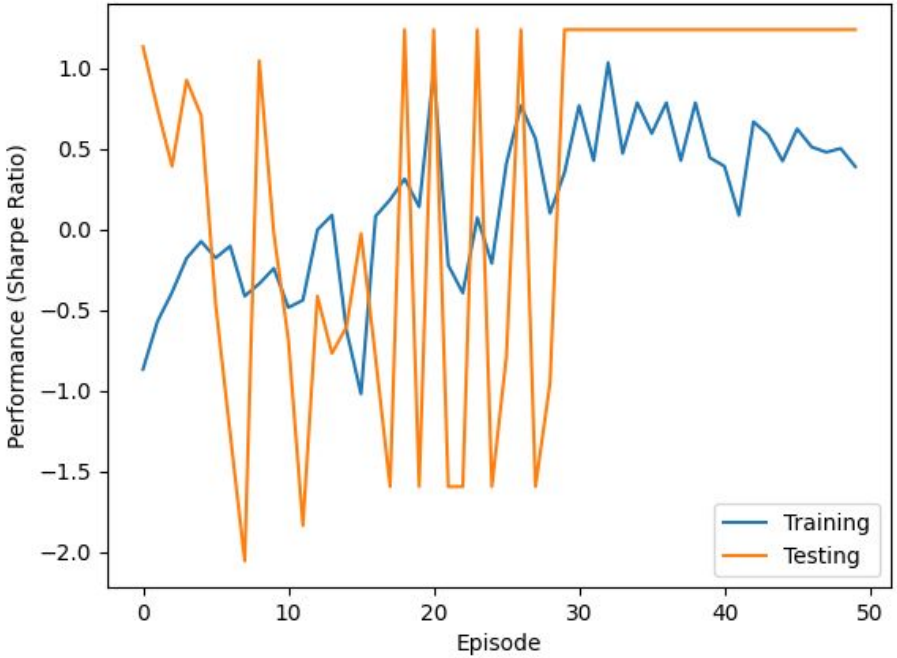


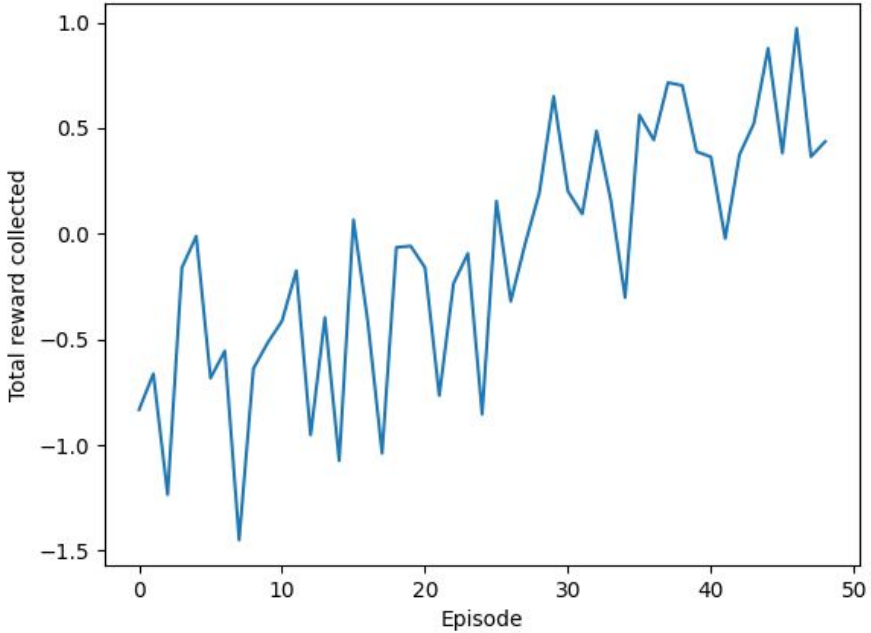


Apple Stock - TDQN

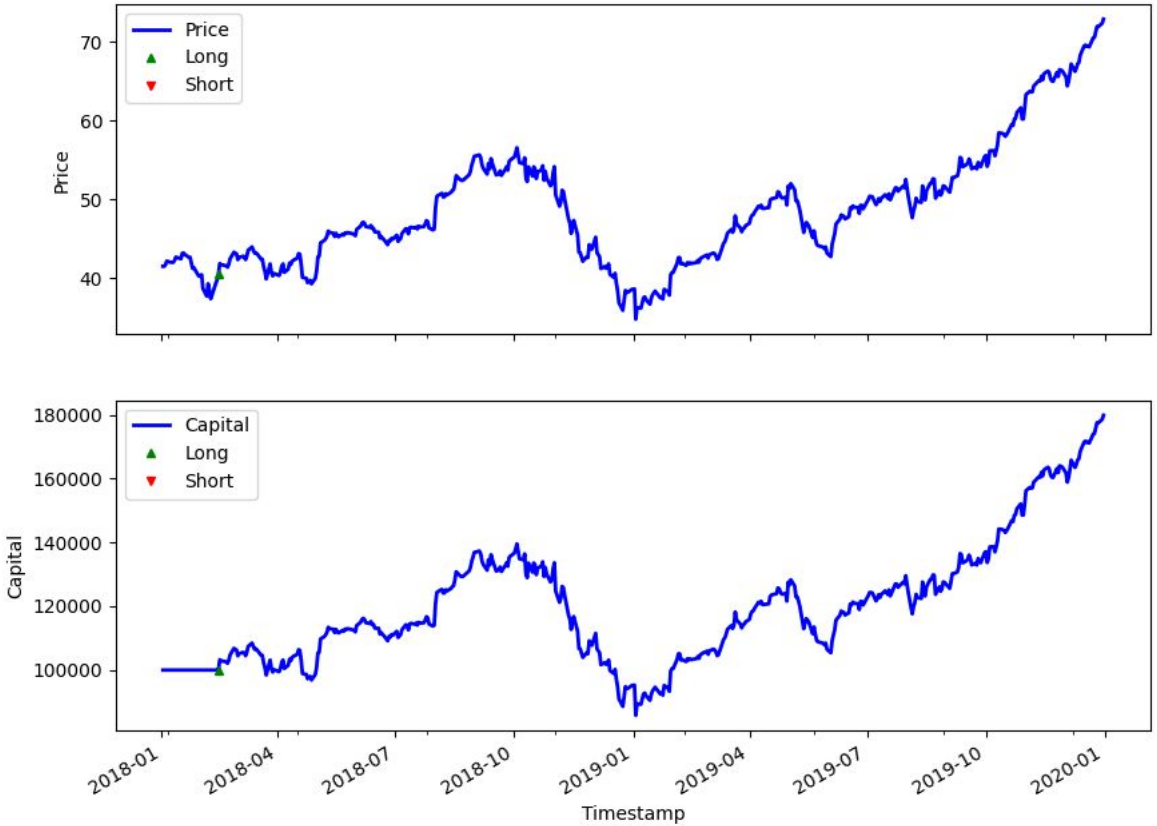


Apple Stock - TD3QN with Max-Advantage-Baseline

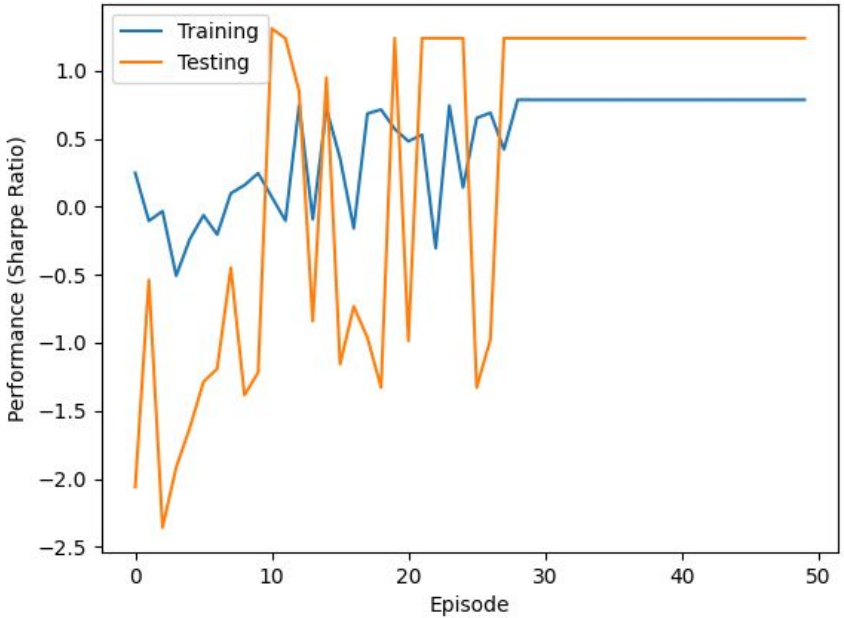


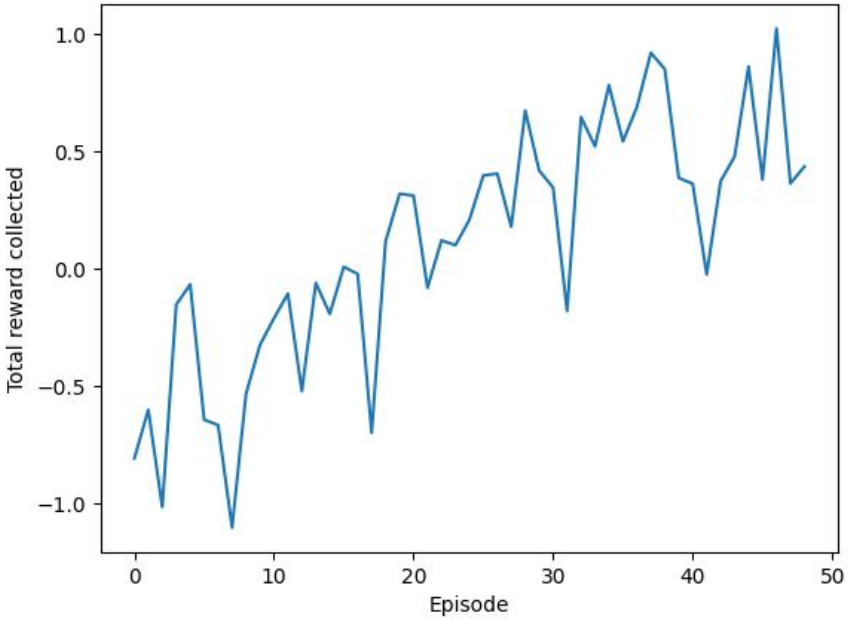


Apple Stock - TD3QN with Max-Advantage-Baseline

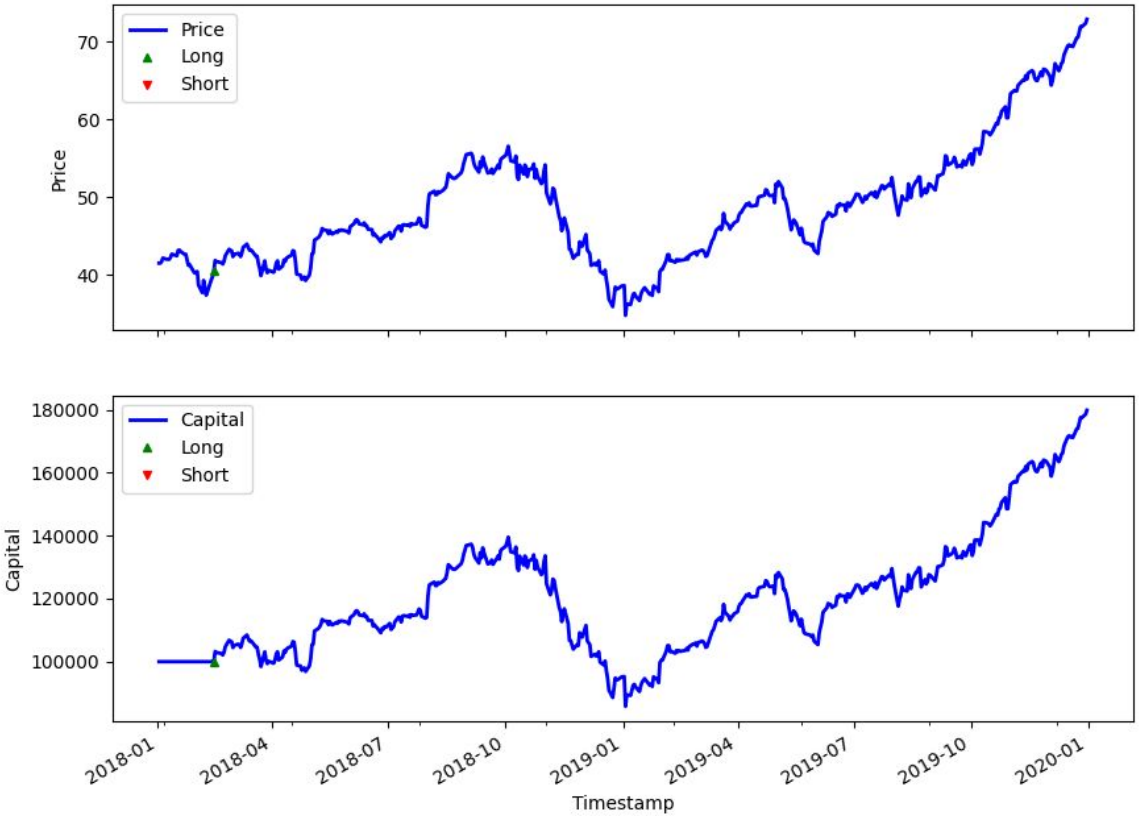


Apple Stock - TD3QN with Avg.-Advantage-Baseline





Apple Stock - TD3QN with Avg.-Advantage-Baseline



Observations

- For each stock, one of the D3QN methods fare slightly better than the baseline method in terms of the Sharpe ratio and the returns obtained
- Has a tendency to take less number of actions over the stock market and mainly takes action when there is a sharp change in the price of the stock owing to the dueling architecture of the QNetwork
    - Hence, it was subject to greater market volatility as can be seen from the provided observations
    - For stock prices which were almost monotonic(like Microsoft and Apple), it took only one action at the start and accordingly the capital had the same trend as the stock price and gave the same results in case of both the variants of D3QN
- Average Advantage Baseline over advantages of the D3QN was highly superior for stock trading as compared to the max-advantage

Thank You