**SmartTrader**

ISS SLS PROJECT REPORT

ONN WEI CHENG A0092201X

YANG JIESHEN A0003901Y

NIRAV JANAK PARIKH A0213573J

ISA-PM-SLS-2021-01-09-IS02PT

Image from: https://static.vecteezy.com/system/resources/previews/002/062/633/original/business-candle-stick-graph-chart-of-stock-market-investment-trading-on-white-background-design-bullish-point-trend-of-graph-illustration-vector.jpg

**TABLE OF CONTENTS**

[1. Problem Statement 2](#_Toc72072460)

[1.1. PRoblem description 2](#_Toc72072461)

[1.2. approach to solution 2](#_Toc72072462)

[2. model overview and implementation 3](#_Toc72072463)

[2.1. state 3](#_Toc72072464)

[2.2. reward 3](#_Toc72072465)

[2.3. actions 3](#_Toc72072466)

[3. model evaluation 4](#_Toc72072467)

[3.1. features used for evaluation 4](#_Toc72072468)

[3.2. phase 1 4](#_Toc72072469)

[3.3. phase 2 4](#_Toc72072470)

[3.4. phase 3 4](#_Toc72072471)

[4. conclusion 5](#_Toc72072472)

**APPENDICES**

APPENDIX I:

1. Problem Statement

* 1. PRoblem description
  2. approach to solution

1. model overview and implementation

The analysis and work done in this project aims to improve on an existing Deep Q-Network reinforcement learning model which is available on Kaggle: <https://www.kaggle.com/itoeiji/deep-reinforcement-learning-on-stock-data/notebook>

The model divides the data set into the training dataset and the test dataset. The agent learns using the training dataset and is used to predict buy/sell/hold actions to maximize profits.

A deep neural network is used to predict the action value based on input states. Due to the nature of stock markets, prices are somewhat correlated sequentially. Hence, an experience relay is deployed to stabilize the network. The agent uses a memory to store batches of historical data which is parsed in randomly to train the neural network. The target Q network is updated periodically to further reduce correlation.

The hyper-perimeters are listed in Table xx, and the structure of the neural network is displayed in Table xx

* 1. state
  2. reward
  3. actions
  4. user interface

One way for users to try out the trained model in actual trading scenarios is to use the wrapper python script (Refer to Appendix xx for user guide and demonstration). The script starts by asking user to input historical data for a selected stock. Once completed, the program will run the model and apply actions to the historical data and simulates the profits made based on the environment.

After which, user is prompted to input current date and stock information. Based on the new information, the action value will be generated for each step and the action with the highest action value is recommended to the user.

The program will also advance the model by applying the action to calculate the reward and profit of the next state.

The current state of the program only runs based on a model which only requires user input of the closing price. Once there are other models generated using other technical analysis, the program is easily scalable to include user inputs for that information.

1. model evaluation
   1. features used for evaluation
   2. phase 1
   3. phase 2
   4. phase 3
2. conclusion

**Appendix:**