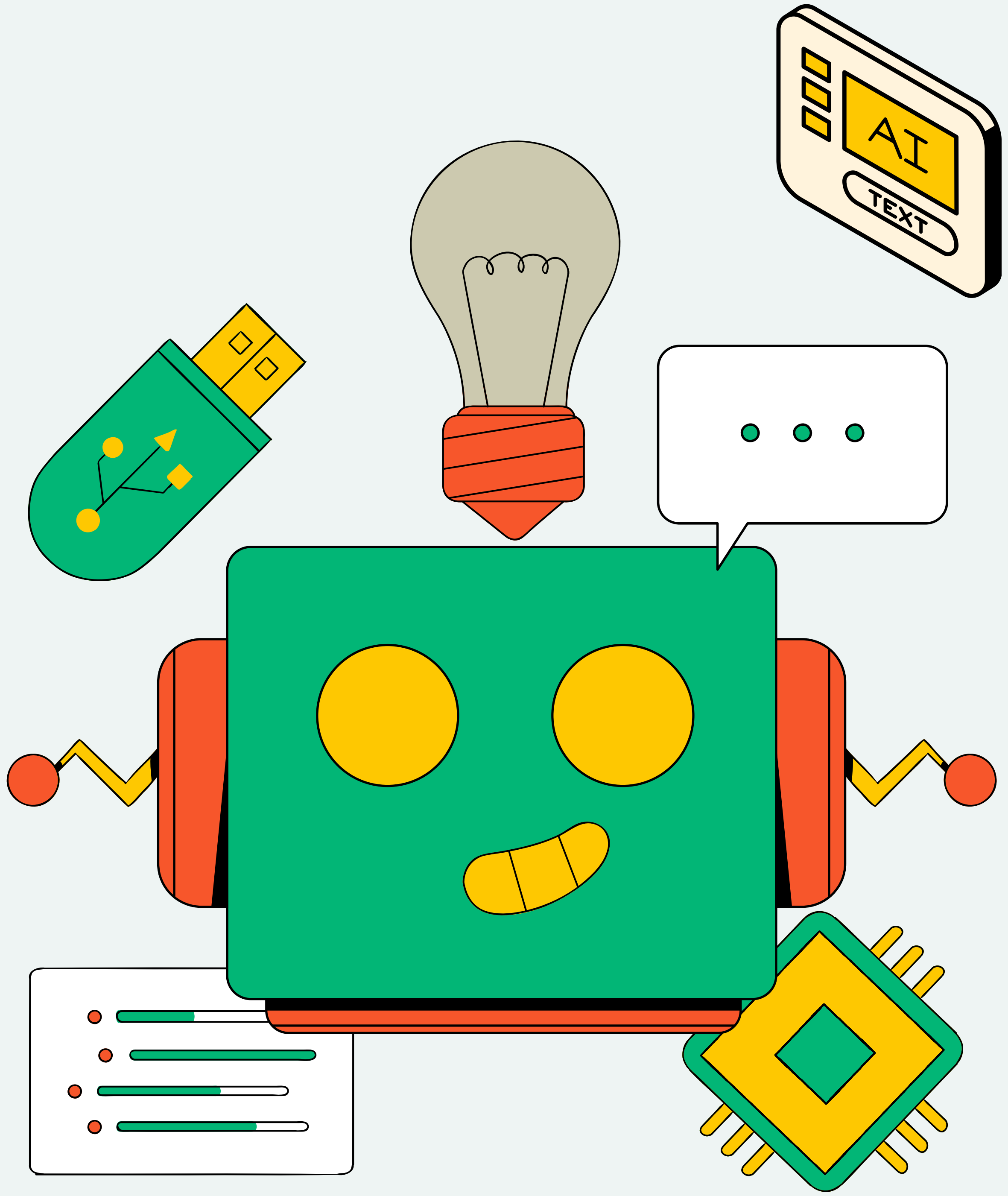
THYNK UNLIMITED

WE LEARN FOR THE FUTURE

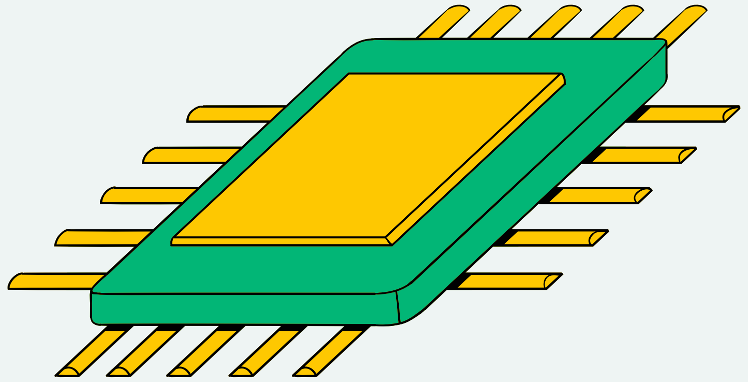
BUILDING A REINFORCEMENT LEARNING MODEL IN STOCK TRADING ACTIVITIE

PRESENTATION



PRESENTED BY:

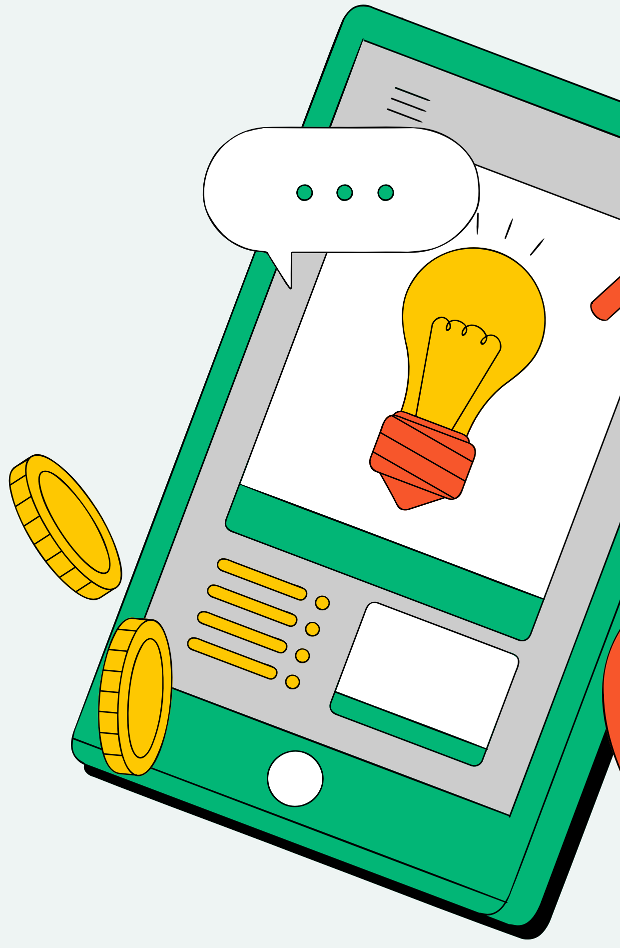
GROUP 5



MEMBER

- Hoàng Thanh Lâm (Leader)
- Trương Phước Trung
- Trương Quyết Thắng
- Trần Tiến Đạt
- Dương Thanh Duy

CONTENTS



INTRODUCTION

- The Vietnamese stock market is developing rapidly, attracting many investors, with over 7.46 million accounts as of July 2023. Investment theories like value investing and the efficient market hypothesis have shaped many financial strategies.

- Algorithmic trading and artificial intelligence (AI) are becoming more prevalent, with 53% of investors believing that AI and machine learning are the future technologies

Deep learning, recurrent neural networks (RNNs), GANs, and reinforcement learning (RL) are new methods being applied to stock prediction. Reinforcement learning research in Vietnam aims to create an automated trading tool useful for investors.

MISSIONS

Deeply study foundational knowledge in finance and reinforcement learning methods for application in stock price trend prediction,

Conduct a comprehensive overview and detailed review of previous research on this topic,

Propose an integrated model to address the problem
& Implement the model.

DEFINITION OF SECURITIES & STOCKS

Securities are assets, including the following types:

- Stocks, bonds, and fund certificates,
- Warrants, covered warrants, rights to purchase shares, depository receipts,
- Derivative securities,
- Other securities as regulated by the government.

Preferred stocks include:

- Dividend preferred stocks,
- Redeemable preferred stocks,
- Voting preferred stocks,
- Other preferred stocks as stipulated in the company's charter and securities law.

STOCK MARKET

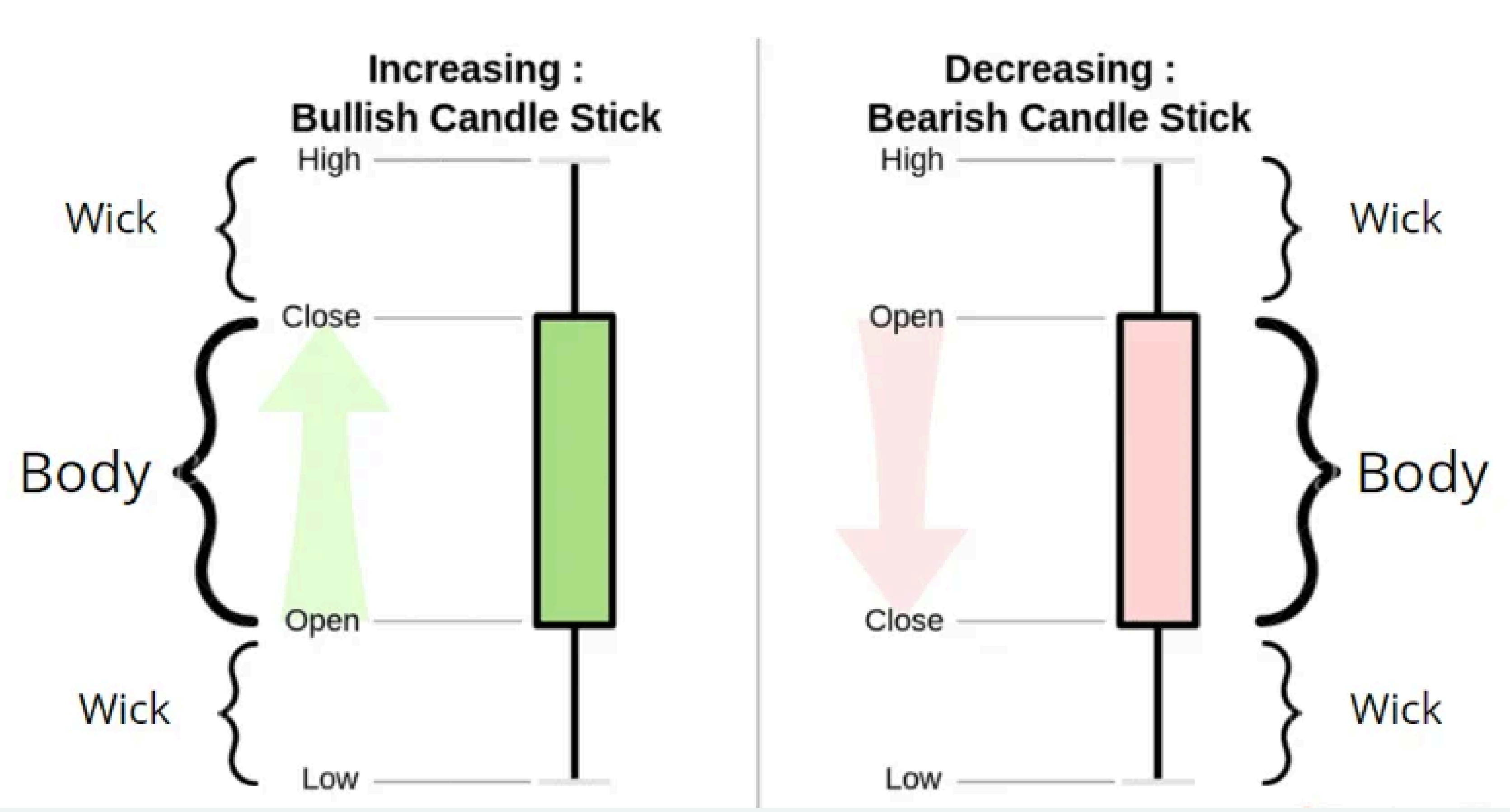
- The stock market is where shares of publicly issued companies are bought and sold, typically operating through a broker-dealer system or through auctions. Today, the market primarily operates online, with the New York Stock Exchange (NYSE) being a notable example.

- The stock market allows companies to raise capital by offering stocks and bonds, while also helping investors earn profits from stock prices and dividends. It transforms savings and investments into effective investment opportunities, contributing to economic growth.

- In addition to stocks, market indices like the VN-INDEX and VN30-INDEX are also of interest to investors, reflecting the condition of the stock market.

CANDLESTICK CHART

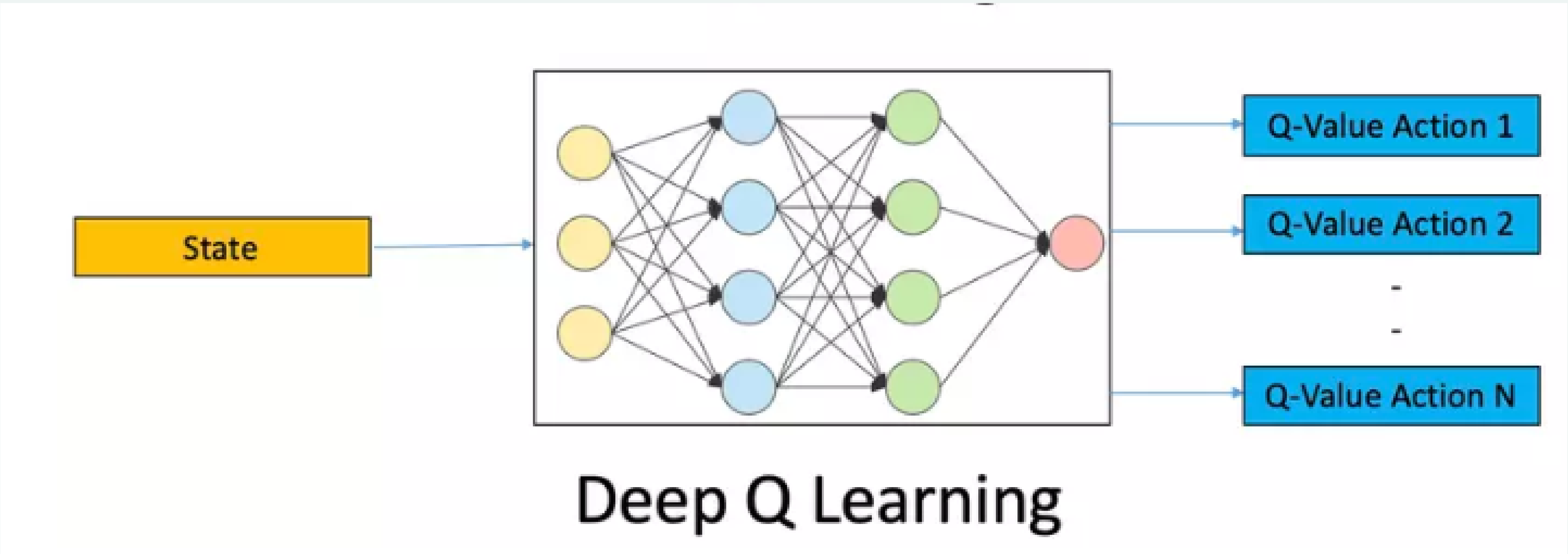
- **Candlestick Chart:** Displays the highest, lowest, opening, and closing prices of a stock at a specific point in time.
- **Real Body:** The area between the opening and closing prices.
- **Bullish Candle** (green): Closing price is higher than the opening price.
- **Bearish Candle** (red): Closing price is lower than the opening price.
- **Upper Shadow:** The area between the highest price and the closing price (bullish candle) or the opening price (bearish candle).
- **Lower Shadow:** The area between the lowest price and the opening price (bullish candle) or the closing price (bearish candle).



PROPOSED MODEL

In this mini project, we explore the application of Deep Q-learning, a reinforcement learning algorithm . Using historical closing prices, we train a Deep Q-learning agent to make buy, sell, or hold decisions, with the objective of maximizing profits within a simulated trading environment.

From DangQuan's Thesis we choose strategy to use Deep Q-learning framework and focus prepare our dataset to improve the result.



PROPOSED MODEL

Deep Q-Learning Model

$$Q^{new}(S_t, A_t) \leftarrow (1 - \underbrace{\alpha}_{\text{learning rate}}) \cdot \underbrace{Q(S_t, A_t)}_{\text{current value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \underbrace{\bigg(\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(S_{t+1}, a)}_{\text{estimate of optimal future value}} \bigg)}_{\text{new value (temporal difference target)}}$$

where R_{t+1} is the reward received when moving from the state S_t to the state S_{t+1}, and α is the learning rate $(0 < \alpha \leq 1)$.

Note that $Q^{new}(S_t, A_t)$ is the sum of three factors:

- $(1-\alpha)Q(S_t, A_t)$: the current value (weighted by one minus the learning rate)
- $\alpha\, R_{t+1}$: the reward R_{t+1} to obtain if action A_t is taken when in state S_t (weighted by learning rate)
- $\alpha\gamma \max_a Q(S_{t+1}, a)$: the maximum reward that can be obtained from state S_{t+1} (weighted by learning rate and discount factor)

```
#
p
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_4 (Dense)	(None, 64)	704
dense_5 (Dense)	(None, 32)	2,080
dense_6 (Dense)	(None, 8)	264
dense_7 (Dense)	(None, 3)	27

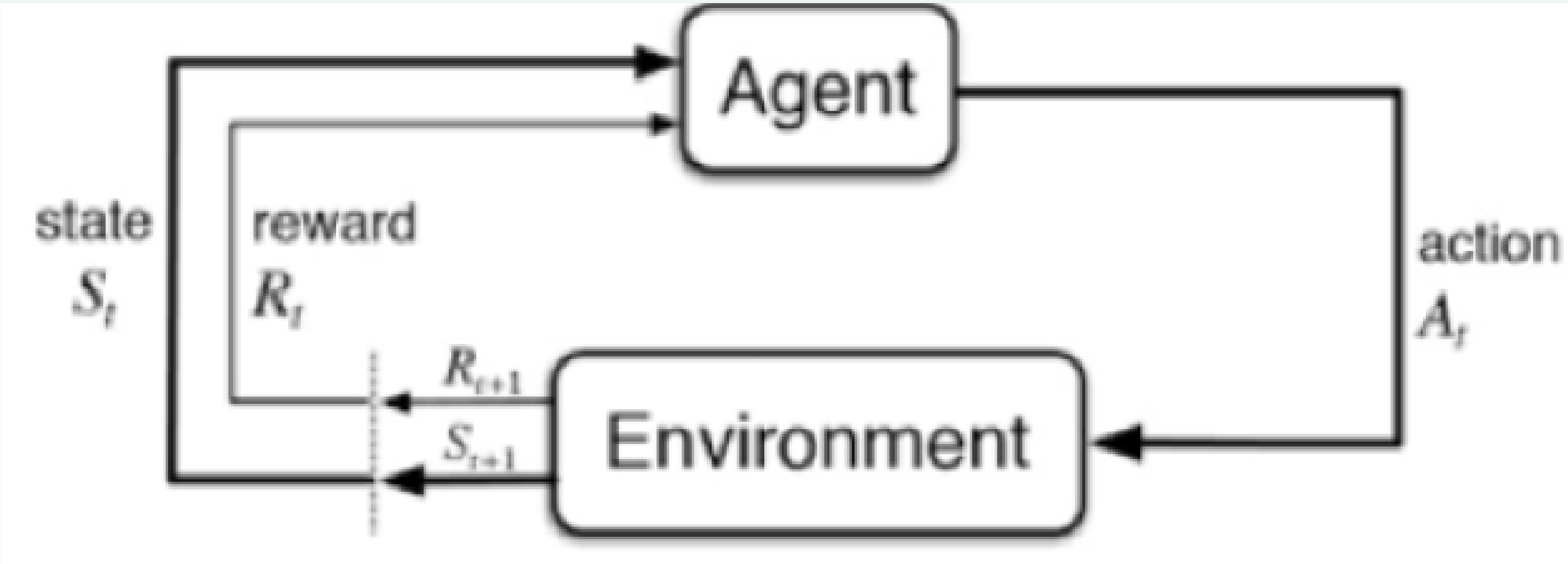
Total params: 3,075 (12.01 KB)
Trainable params: 3,075 (12.01 KB)
Non-trainable params: 0 (0.00 B)
None

PROPOSED MODEL

ENVIRONMENT FOR STOCK PREDICTION:

MDP (MARKOV DECISION PROCESS) FOR STOCK PRICE PREDICTION:

- AGENT – AN AGENT A THAT WORKS IN ENVIRONMENT
- ACTION – BUY/SELL/HOLD
- STATES: DATA VALUES OF 10 CLOSE PRICES BEFORE
- REWARDS – PROFIT / LOSS

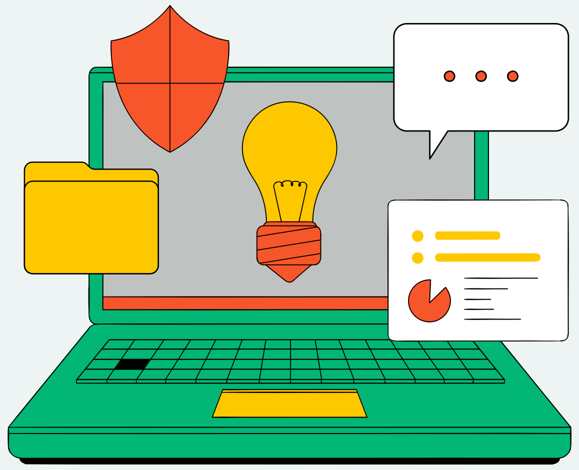


PROPOSED MODEL

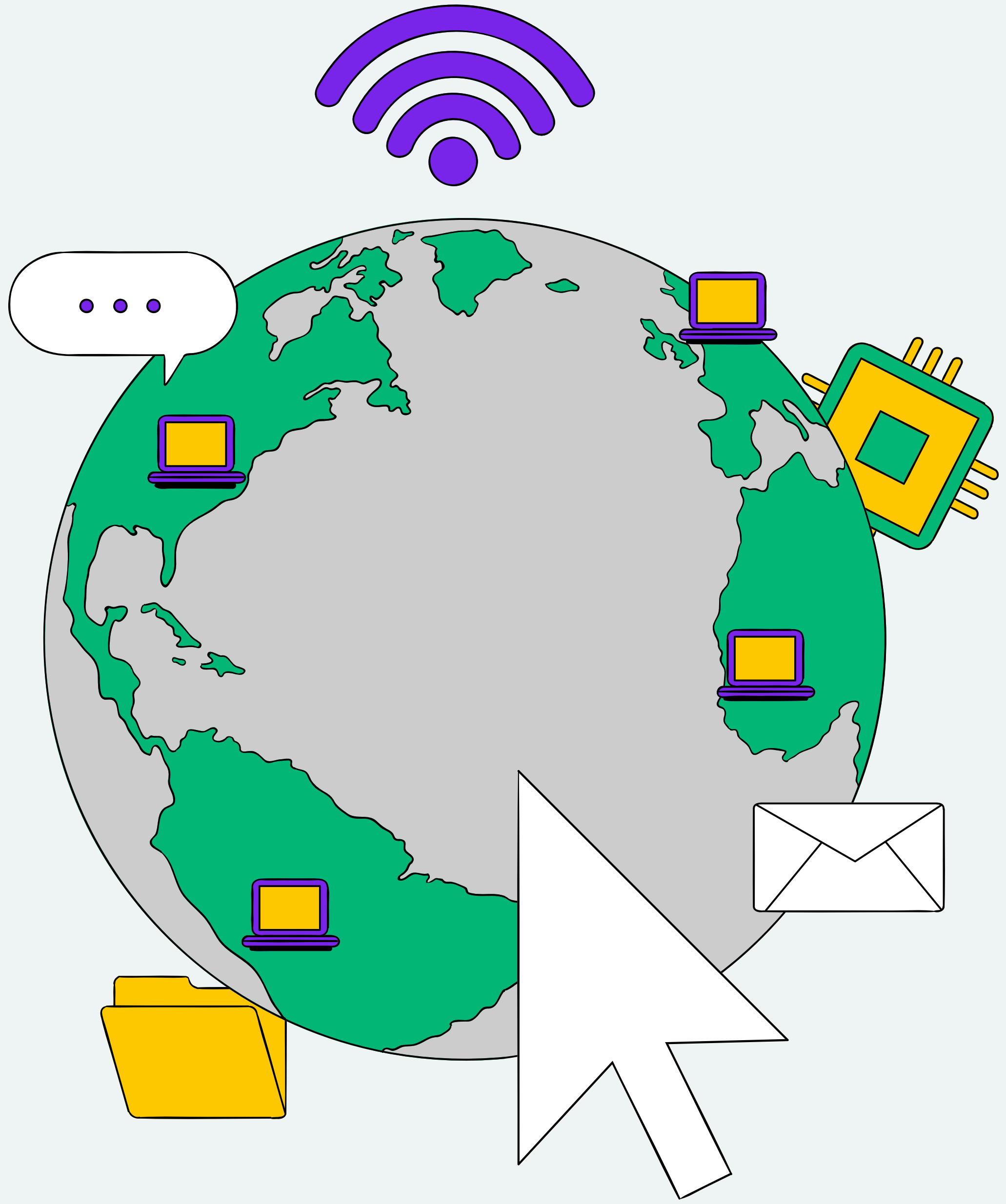
Main paramaters of Model:

```
self.state_size = state_size
self.action_size = 3
self.memory = deque(maxlen=1000)
self.inventory = []
self.model_name = model_name
self.is_eval = is_eval
self.gamma = 0.95
self.epsilon = 1.0
```

```
#epsilon-greedy approch is implemented to find action
    def act(self, state):
        # If not in evaluation mode or with probability epsilon, choose a random action
        if not self.is_eval and random.random() <= self.epsilon:
            return random.randrange(self.action_size)A
        #exploitation
        # Otherwise, predict Q-values for each action and choose the action with the highest Q-value
        options = self.model.predict(state)
        return np.argmax(options[0])
```



PROPOSED MODEL



Collect data strategy

- According to Dang Quan's article, there will be 30 data sets, but we decided to get external data including 3 companies: ACB, FPT, VCB
- Data will be taken from https://finance.yahoo.com/
- Each data set will start from November 1, 2014, ending on October 31, 2024 (Excluding Saturdays, Sundays and holidays)

OVERVIEW

Overview

Visualize

Processing

Experiments

Metrics

Evaluations

OVERVIEW

- ACB: 2489 days
- FPT: 2494 days
- VCB: 2487 days

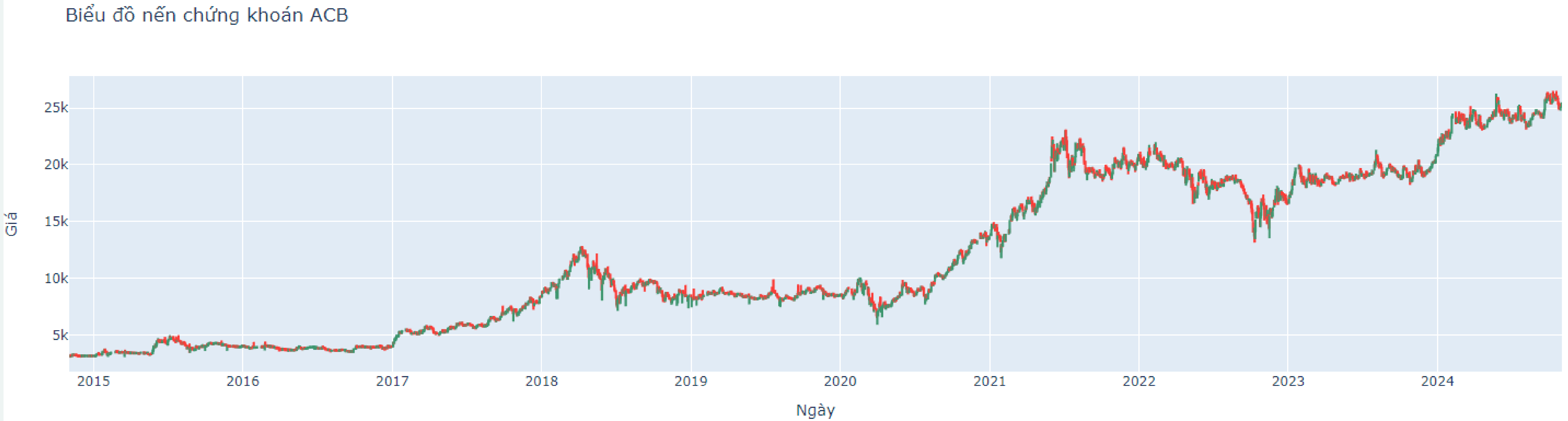
- Date: trading date
- Open: opening price
- High: highest price
- Low: lowest price
- Close: closing price
- Volume: total volume of matched transactions

Example: ACB

	Close	High	Low	Open	Volume
Date					
2014-11-03 00:00:00+00:00	3148.515381	3148.515381	3127.936768	3148.515381	2087723
2014-11-04 00:00:00+00:00	3148.515381	3148.515381	3107.358154	3127.936768	227737
2014-11-05 00:00:00+00:00	3107.358154	3148.515381	3107.358154	3127.936768	1027120
2014-11-06 00:00:00+00:00	3210.250977	3210.250977	3127.936768	3127.936768	2950652
2014-11-07 00:00:00+00:00	3210.250977	3230.829346	3189.672363	3230.829346	529432
...	...	...	...	...	...
2024-10-24 00:00:00+00:00	25000.000000	25450.000000	25000.000000	25400.000000	12246700
2024-10-25 00:00:00+00:00	24900.000000	25100.000000	24850.000000	25000.000000	10970400
2024-10-28 00:00:00+00:00	25150.000000	25150.000000	24750.000000	24900.000000	5315602
2024-10-29 00:00:00+00:00	25200.000000	25300.000000	25100.000000	25300.000000	5561461
2024-10-31 00:00:00+00:00	25400.000000	25450.000000	25050.000000	25100.000000	5191604

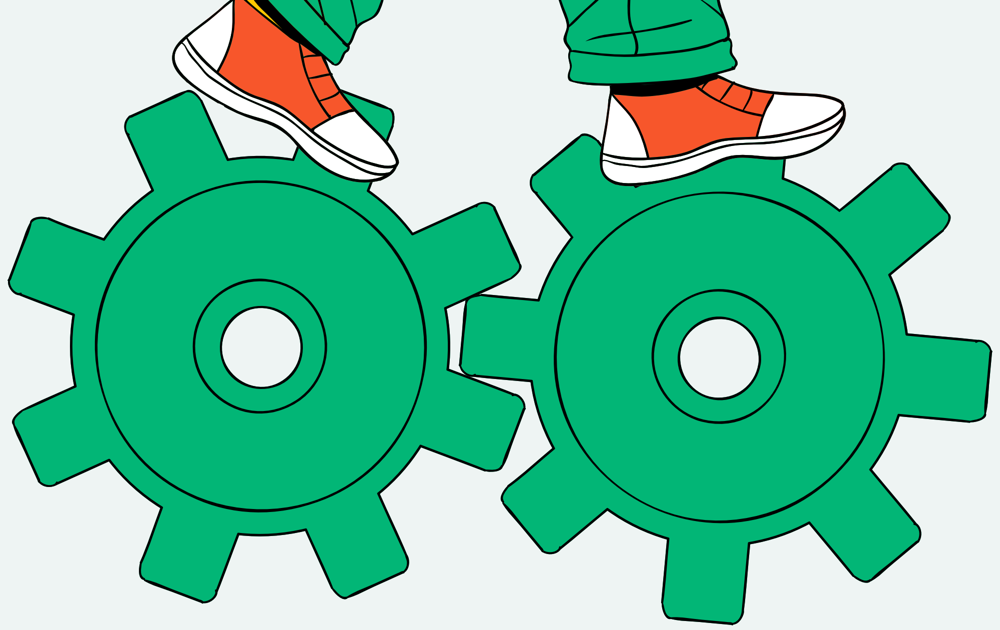
2489 rows × 5 columns

VISUALIZE

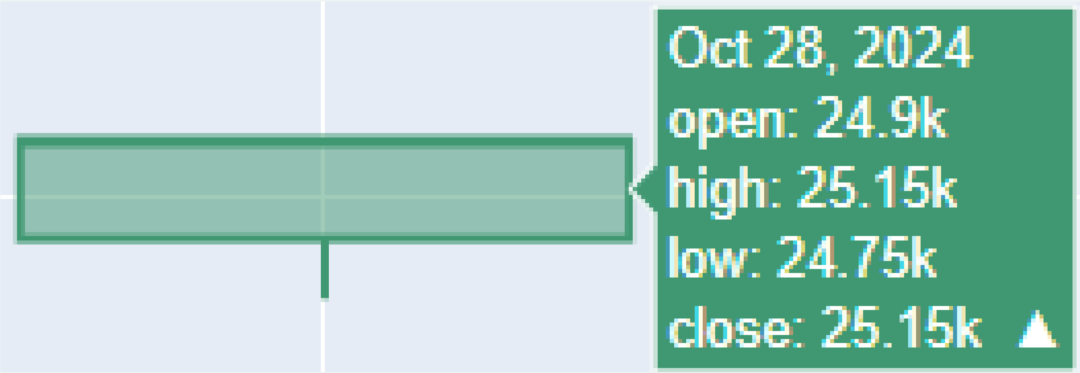


Biểu đồ nến chứng khoán ACB
25k
20k
15k
10k
5k
Giá
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
Ngày

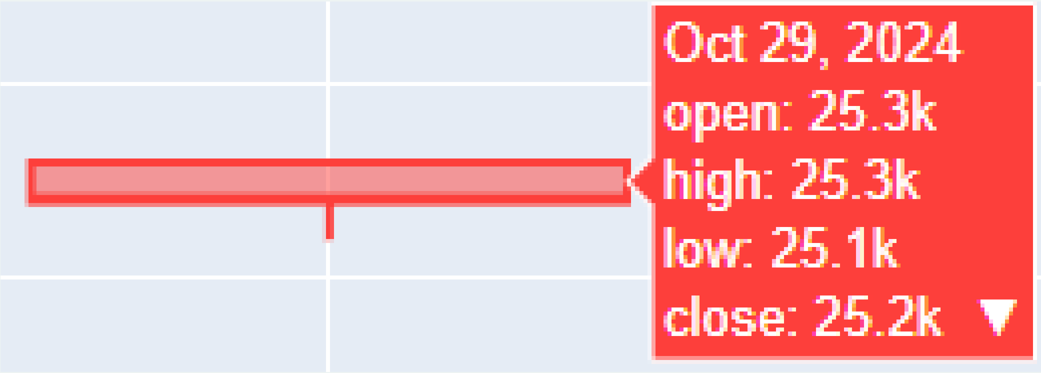
VISUALIZE



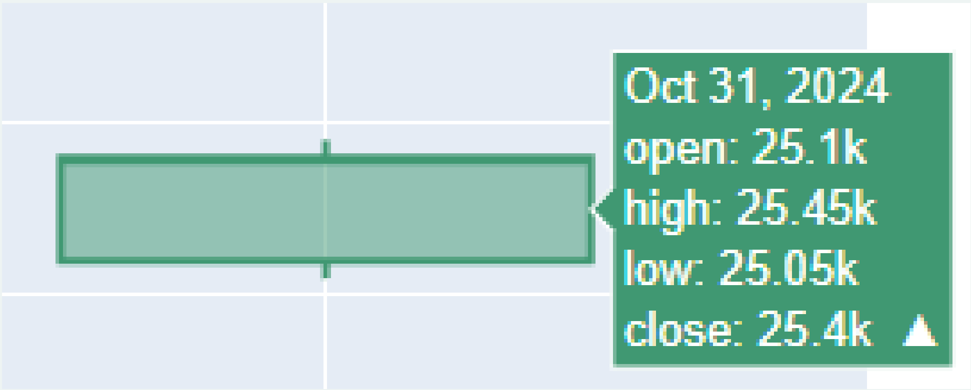
28/10/2024



29/10/2024



31/10/2024



PROCESSING

$$\text{label} = \frac{\text{today\_close} - \text{yesterday\_close}}{\text{yesterday\_close}}$$

	Close	High	Low	Open	Volume	label
Date						
2014-11-03 00:00:00+00:00	3148.515381	3148.515381	3127.936768	3148.515381	2087723	hold
2014-11-04 00:00:00+00:00	3148.515381	3148.515381	3107.358154	3127.936768	227737	sell
2014-11-05 00:00:00+00:00	3107.358154	3148.515381	3107.358154	3127.936768	1027120	sell
2014-11-06 00:00:00+00:00	3210.250977	3210.250977	3127.936768	3127.936768	2950652	buy
2014-11-07 00:00:00+00:00	3210.250977	3230.829346	3189.672363	3230.829346	529432	sell
...	...	...	...	...	...	...
2024-10-24 00:00:00+00:00	25000.000000	25450.000000	25000.000000	25400.000000	12246700	sell
2024-10-25 00:00:00+00:00	24900.000000	25100.000000	24850.000000	25000.000000	10970400	sell
2024-10-28 00:00:00+00:00	25150.000000	25150.000000	24750.000000	24900.000000	5315602	buy
2024-10-29 00:00:00+00:00	25200.000000	25300.000000	25100.000000	25300.000000	5561461	sell
2024-10-31 00:00:00+00:00	25400.000000	25450.000000	25050.000000	25100.000000	5191604	sell

2489 rows × 6 columns

EXPERIMENTS

- **Model:** Q-learning agent with actions (buy, sell, hold) to maximize profit on ACB, FPT, VCB stock data.
- **Training Setup:** Trained on 80% of data with 1000 episodes, using a 10-day price window as input state.
- **Hyperparameters**:
  - Episode = 1000
  - batch_size = 32
  - Discount factor = 0.95
  - Exploration rate = 1.0

Dataset Used

DATASET	TIME PERIOD	LENGTH	DESCRIBE
ACB, FPT, VCB	03/11/2014 - 31/10/2022	1991 DAYS	TRAIN
ACB, FPT, VCB	01/11/2022 - 31/10/2024	498 DAYS	TEST

EXPERIMENTS

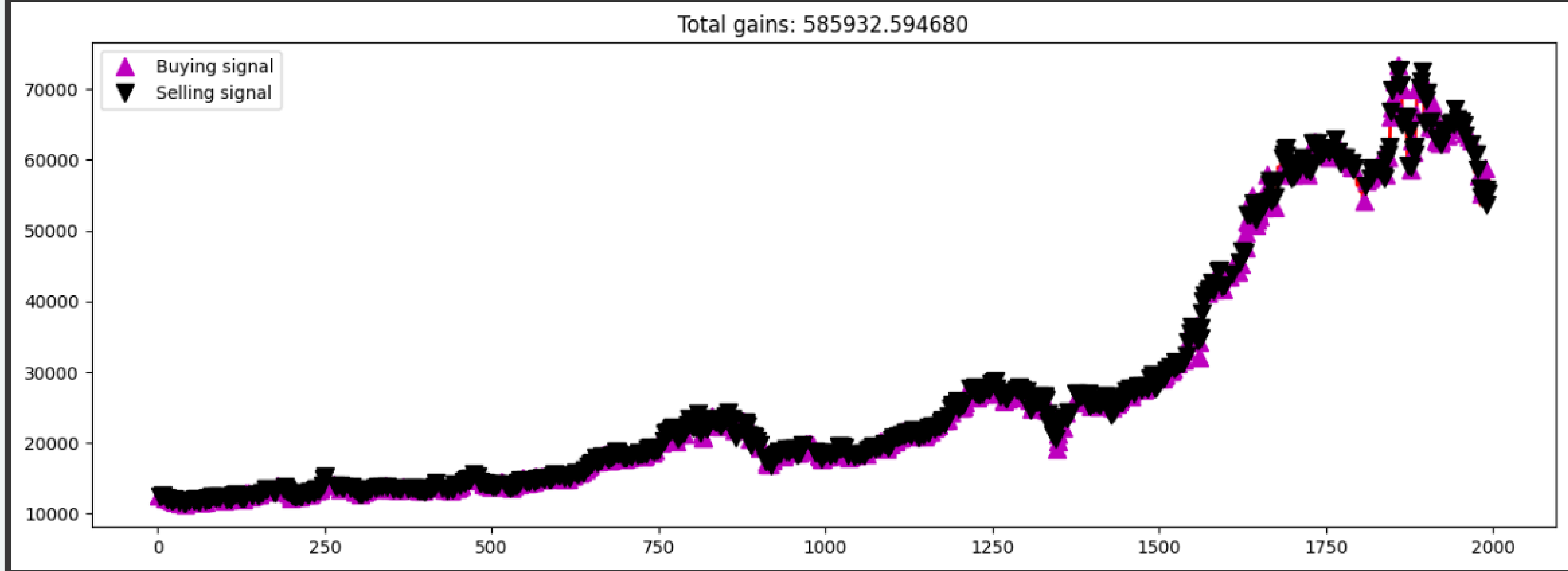
- **Model:** Q-learning agent with actions (buy, sell, hold) to maximize profit on ACB, FPT, VCB stock data.
- **Training Setup:** Trained on 80% of data with 1000 episodes, using a 10-day price window as input state.
- **Hyperparameters**:
  - Episode = 1000
  - batch_size = 32
  - Discount factor = 0.95
  - Exploration rate = 1.0

Label distribution

STOCK CODE	SELL	BUY	HOLD
ACB	76.46	23.50	0.04
FPT	78.23	21.73	0.04
VCB	78.23	21.73	0.04

EVALUATIONS

```
Running episode 1000/1000
--------------------------------
Total Profit: $585932.59
--------------------------------
```

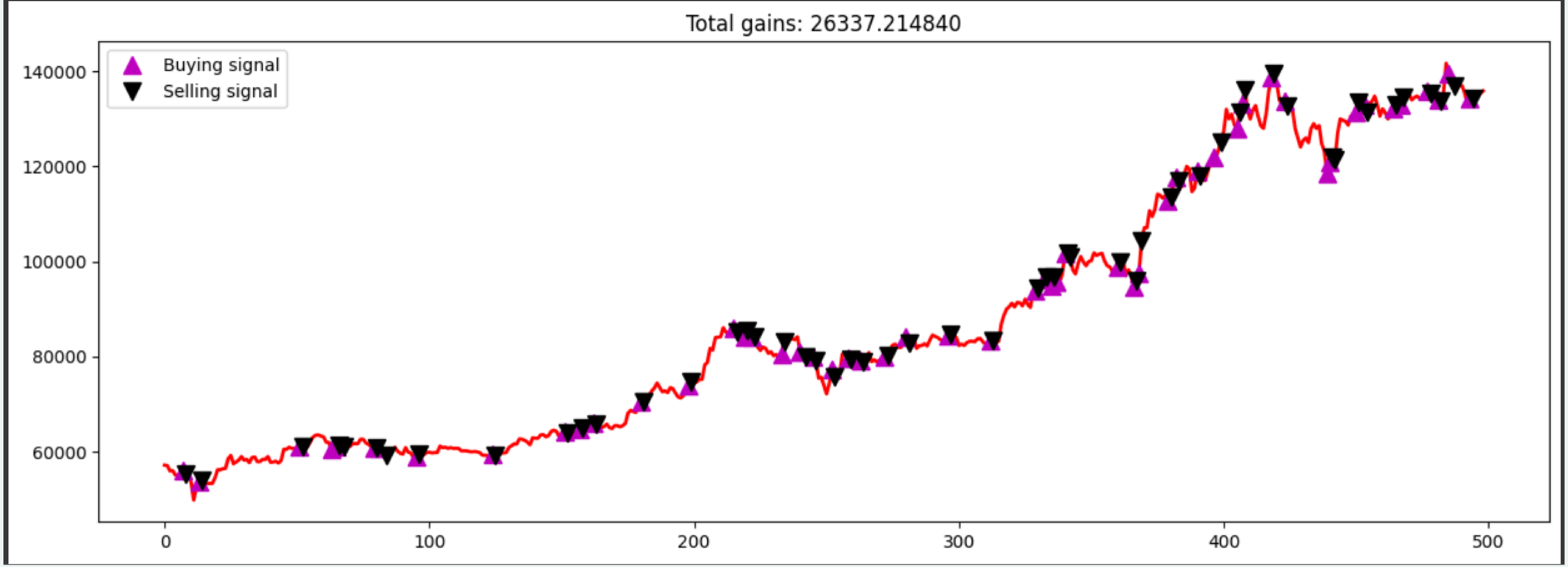



```
The episode with the highest profit is 670 with a profit of $4389564.92

Top 10 episodes with the most transactions:
Episode 356: 1381 transactions, Total Profit: $2095578.38
Episode 975: 1379 transactions, Total Profit: $1767491.47
Episode 431: 1375 transactions, Total Profit: $2091897.90
Episode 865: 1375 transactions, Total Profit: $2807179.48
Episode 214: 1370 transactions, Total Profit: $2893487.65
Episode 290: 1370 transactions, Total Profit: $3323619.31
Episode 707: 1368 transactions, Total Profit: $1002626.98
Episode 905: 1367 transactions, Total Profit: $1423268.78
Episode 742: 1366 transactions, Total Profit: $1741421.18
Episode 781: 1366 transactions, Total Profit: $2487592.73
```

EVALUATIONS

```
------------------------------------------
Total Profit: $26337.21
Sharpe Ratio: 0.1271575868527757
Win Rate: 0.5283018867924528
------------------------------------------
```



EVALUATIONS

Experimental results on Testset

STOCK CODE	ACCURACY	F1-SCORE
ACB	52.50	55.90
FPT	49.06	54.16
VCB	55.65	59.48
AVERAGE	52.40	56.51

CONCLUSION

While Deep Q-learning has potential for developing adaptive and sequential decision-making strategies in stock trading, its practical challenges, especially related to the stock market's volatility and complexity, make it difficult to deploy effectively without significant adaptation.

CONCLUSION

The main takeaway is the high complexity and unpredictability of financial markets, which are influenced by numerous factors like investor sentiment, global events, and economic indicators. This complexity highlights the limitations of traditional Deep Q-learning in effectively modeling such multifaceted, real-world data.

CONTRIBUTION

Topic	Team Effort	Hoàng Thanh Lâm	Trương Phước Trung	Trương Quyết Thắng	Trần Tiến Đạt	Dương Thành Duy
Backgrounds of Stock Trading	100%	20%	20%	20%	20%	20%
Backgounds and related works of RL techniques you used	100%	20%	20%	20%	20%	20%
Your proposed model	100%	20%	30%	15%	15%	20%
Implementation and evaluation	100%	30%	20%	15%	20%	15%
Documentation (technical report, slides)	100%	20%	20%	20%	20%	20%

THANK YOU VERY MUCH

Any question ?