Đại học Quốc gia TPHCM
Trường Đại học Công nghệ Thông tin

BÁO CÁO ĐỒ ÁN CUỐI KỲ

CS106.M11.KHTN

DEEP REINFORCEMENT LEARNING FOR AUTOMATED STOCK TRADING AN ENSEMBLE SRATEGY

Sinh viên thực hiện:

19520257 Hứa Thanh Tân
19522155 Phạm Viết Tài
19520218 Nguyễn Minh Phú

Giáo viên hướng dẫn

TS. LƯƠNG NGỌC HOÀNG

TPHCM, Tháng 12 năm 2021

Mục lục

Chương 1

Tổng quan

Thị trường chứng khoán đã và đang trở nên lớn mạnh hơn bao giờ hết, thu hút rất nhiều sự quan tâm của các nhà đầu tư. Người ta đã cố gắng phát triển các hệ thống thông minh tự động giao dịch chứng khoán hòng thu về lợi nhuận cho mình. Tuy nhiên, chứng khoán lại là một môi trường vô cùng phức tạp và biến chuyển liên tục, điều này làm cho việc xây dựng một hệ thống hiệu quả trở nên vô cùng khó khăn.

Với sự ra đời của Khoa học dữ liệu và Máy học, rất nhiều nghiên cứu đã đề ra phương hướng tiếp cận để giải quyết bài toán này. Một hướng tiếp cận hiệu quả đối với bài toán là sử dụng các thuật toán học tăng cường hay Reinforcement learning (RL). Bài báo cáo này tập trung nghiên cứu các thuật toán RL như A2C, DDPG, PPO, và một chiến lược tổng hợp. Chiến lược tổng hợp được tích hợp những điểm mạnh của cả 3 thuật toán nên có thể dễ dàng thích ứng với tình hình thị trường phức tạp. Sau đó là kết quả khi tiến hành thử nghiệm khi chạy chiến lược trên thị trường tiền ảo.

1.1 Giới thiệu

1.1.1 Giao dịch chứng khoán

Chứng khoán là một bằng chứng xác nhận sự sở hữu hợp pháp của người sở hữu đó với tài sản hoặc phần vốn của công ty hay tổ chức đã phát hành. Chứng khoán bao gồm các loại như cổ phiếu, trái phiếu, chứng chỉ quỹ, ... Chứng khoán cũng được coi là một phương tiện hàng hóa trừu tượng có thể thỏa thuận và có thể thay thế được, đại diện cho một giá trị tài chính.

Giao dịch chứng khoán là hành động mua và bán chứng khoán trên thị trường chứng khoán. Mục tiêu của người tham gia là tối đa hóa lợi nhuận của mình bằng việc mua và bán chứng khoán khi thị trường chứng khoán biến đổi. Lợi nhuận có được khi ta mua vào cổ phiếu trong thời điểm nó có giá thấp và bán ra khi giá của nó tăng lên cao.

1.1.2 RL

RL là một phương pháp học máy làm việc với environment (môi trường) và các agents (tác tử). Các agents sẽ đưa ra những hành động để nhận rewards (những phần thưởng). Các thuật

toán RL sẽ tập trung vào mục tiêu là tìm được một chiến lược hành động mà tối đa reward nhận được, đồng thời tránh được những rủi ro tiềm ẩn. Ưu điểm của RL là không cần nhiều dữ liệu được gán nhãn sẵn, điều này giúp tiết kiệm được rất nhiều thời gian và công sức khi xây dựng mô hình.

Deep Reinforcement learning là RL sử dụng kết hợp với neural network (mạng nơ ron nhân tạo). Bởi vì môi trường thị trường chứng khoán thì rất phức tạp, nên việc tạo ra một thuật toán có thể đánh giá được toàn bộ các yếu tố ảnh hưởng đến việc biến động của thị trường. Việc này đòi hỏi không gian lưu trữ dữ liệu states (các trạng thái) cũng như tính toán là rất lớn, chính vì vậy, ta phải sử dụng deep reinforcement learning để có thể xử lý được bài toán.

1.2 Các hướng tiếp cận

Những ứng dụng của deep reinforcement learning vào thị trường tài chính trong những năm gần đây nhìn bài toán theo hướng xem trạng thái và không gian hành động là liên tục hoặc rời rạc, từ đó sẽ có ba hướng tiếp cận chính: critic-only, actor-only, và actor-critic.

1.2.1 Hướng tiếp cận Critic-only

Đây là hướng tiếp cận thường gặp nhất, giải quyết bài toán với không gian trạng thái hành động là rời rạc. Ý tưởng cốt lõi của phương pháp này là sử dụng hàm giá trị Q-value để tìm ra chính sách tối ưu giá phần thưởng mong đợi trong tương lai.

Điển hình của phương pháp này là thuật toán Q-learning. Q-learning là một loại thuật toán học tăng cường với mục tiêu là học một chiến lược, cho biết máy sẽ thực hiện hành động nào trong hoàn cảnh nào. Q ở đây là Quality (chất lương), là chỉ action quality, tức là mức độ tốt của reward mà việc thực hiện hành động đó mang lại.

Hạn chế lớn nhất của hướng tiếp cận này là chỉ sử dụng được trên các bài toán có không gian trạng thái hành động rời rạc và hữu hạn, nên rõ ràng ta không thể áp dụng nó vào giải quyết bài toán thị trường chứng khoán mà có giá trị chứng khoán là liên tục.

1.2.2 Hướng tiếp cận Actor-only

Hướng tiếp cận này giải quyết bài toán với không gian trạng thái hành động là liên tục. Ý tưởng cốt lõi của phương pháp này là agent tự học chiến lược trực tiếp. Khuyết điểm của phương pháp này là agent sẽ phải thực hiện một chuỗi hành động và rất khó để ấn định giá trị đúng cho hành động, đồng nghĩa khi cập nhật hàm gradient sẽ có phương sai rất lớn. Hơn nữa, phương pháp này chỉ hoạt động với những bài toán có phong cách chơi theo màn (episode), nếu agent không hoàn thành màn chơi thì sẽ không có việc cập nhật nào diễn ra cả.

1.2.3 Hướng tiếp cận Actor-Critic

Hướng tiếp cận cuối cùng gần đây đã được áp dụng vào tài chính. Ý tưởng cốt lõi của cách tiếp cận này là nó kết hợp cả hai hướng Actor-only và Critic-only, nghĩa là nó cập nhật hàm trạng thái và phân phối xác suất của chiến chiến lược một cách đồng thời. Theo thời gian mô

hình sẽ học được chiến lược với những hành động tốt hơn và hàm đánh giá cải thiện khả năng đánh giá những hành động đó. Phương pháp này hứa hẹn khả năng xử lý được những bài toán với môi trường phức tạp, từ đó người ta đưa nó vào bài toán giao dịch chứng khoán.

Chương 2

Ý tưởng

2.1 Mô hình MDP cho bài toán giao dịch chứng khoán

Ta mô hình hóa bài toán theo hướng Markov Decision Process (MDP) để có thể áp dụng các thuật toán DRL:

- Trạng thái (State) $s = [p, h, b]$: vector mang giá cổ phiếu (stock price) $p \in \mathbb{R}_+^D$, lượng cổ phiếu đang nắm giữ (hold) $h \in \mathbb{Z}_+^D$ và số tiền còn lại trong tài khoản (remaining balance) $b \in \mathbb{R}_+$ với D biểu thị cho số lượng chứng khoán ta giao dịch và $\mathbb{Z}_+$ biểu thị cho số nguyên không âm.

- Hành động (Action) a : vector mang các hành động mà ta có thể tác động lên các mã chứng khoán, bao gồm *mua, bán và giữ* tương ứng dẫn đến kết quả tăng, giảm hoặc không thay đổi lượng cổ phiếu nắm giữ h

- Phần thưởng (Reward) $r(s, a, s')$: phần thưởng tức thì, nhận được khi thực hiện hành động a trong trạng thái s và trạng thái sau khi hành động sẽ là s'.

- Chiến lược (Policy) π : chiến lược giao dịch tại trạng thái s, cụ thể chính là phân phối xác suất của các hành động tại trạng thái s.

- Hàm giá trị Q-Value $Q_\pi(s, a)$: giá trị phần thưởng kỳ vọng khi thực hiện hành động a tại trạng thái s theo chiến lược π

Khi thực hiện một hành động a lên chứng khoán $d(d = 1, ..., D)$ tại thời điểm t, trạng thái s sẽ thay đổi theo công thức sau:

- Bán một lượng $\mathbf{k}[d] \in [1, \mathbf{h}[d]]$ chứng khoán sẽ dẫn đến kết quả $\mathbf{h}_{t+1}[d] = \mathbf{h}_t[d] - \mathbf{k}[d]$ với $\mathbf{k}[d] \in \mathbb{Z}_+$

- Giữ $\mathbf{h}_{t+1}[d] = \mathbf{h}_t[d]$

- Mua một lượng $\mathbf{k}[d]$ chứng khoán sẽ dẫn đến kết quả $\mathbf{h}_{t+1}[d] = \mathbf{h}_t[d] + \mathbf{k}[d]$

2.2 Các ràng buộc

- Tính thanh khoản: mặc định cho rằng thị trường chứng khoán nói chung cũng như các mã chứng khoán giao dịch trong D nói riêng sẽ không bị ảnh hưởng nhiều bởi những giao dịch thực hiện trong quá trình học.

- Số tiền còn lại trong tài khoản không thể âm $b \geq 0$: Các hành động làm cho số tiền còn lại trong tài khoản xuống dưới 0 là không hợp lệ. Dựa vào hành động tại thời điểm t, các chứng khoán sẽ được chia thành các tập con không giao nhau bao gồm: tập chứng khoán bán đi $\mathcal{S}$, tập mua vào $\mathcal{B}$ và tập giữ $\mathcal{H}$, $\mathcal{S} \cup \mathcal{B} \cup \mathcal{H} = \{1, ..., D\}$. Gọi $\mathbf{p}_t^B = [p_t^i : i \in \mathcal{B}]$ và $\mathbf{k}_t^B = [k_t^i : i \in \mathcal{B}]$ lần lượt là các vector mang giá và số lượng cổ phiếu giao dịch của các chứng khoán mua vào thời điểm t. Tương tự chúng ta có $\mathbf{p}_t^S$ và $\mathbf{p}_t^S$ cho chứng khoán bán đi, và $\mathbf{p}_t^H \mathbf{p}_t^H$ cho chứng khoán giữ. Khi đó số tiền còn lại trong tài khoản tại thời điểm $t+1$ là: $b_{t+1} = b_t + (p_t^S)^T k_t^S - (p_t^B)^T k_t^B$ và quy định này được biểu diễn thành:

$$b_{t+1} = b_t + (p_t^S)^T k_t^S - (p_t^B)^T k_t^B \geq 0 \tag{2.1}$$

- Phí giao dịch: Mỗi giao dịch trong thị trường đều phải đóng phí. Có nhiều loại phí như phí trao đổi, phí thi hành, phí an ninh Mỗi sàn giao dịch sẽ có những quy định riêng về loại phí này. Để đơn giản hóa ta cho phí này sẽ bằng 0.1% giá trị của mỗi giao dịch:

$$c_t = \mathbf{p}^T \mathbf{k}_t \times 0.1\% \tag{2.2}$$

- Mức chấp nhận rủi ro sụp đổ thị trường: Thị trường chứng khoán tương đối nhạy cảm. Nó có thể bị sụp đổ do xảy ra chiến tranh, sự sụp đổ của bong bóng tài chính, các công ty vỡ nợ hay khủng hoảng tài chính. Để đảm bảo an toàn tránh trường hợp xấu nhất, ta sử dụng một tham số gọi là chỉ số nhiễu loạn đo những biến động cực đoan của giá chứng khoán:

$$turbulence_t = (\mathbf{y}_t - \mu)\Sigma^{-1}(\mathbf{y}_t - \mu)' \in \mathbb{R} \tag{2.3}$$

  với $\mathbf{y}_t \in \mathbb{R}^D$ là lợi nhuận cổ phiếu tại thời điểm t, $\mu \in \mathbb{R}^{\mathbb{D}}$ là lợi nhuận trung bình kể từ trước đó, và $\Sigma \in \mathbb{R}^{D \times D}$ là phương sai của lợi nhuận kể từ trước đó. Khi chỉ số nhiễu loạn $turbulence_t$ cao hơn một ngưỡng nhất định, toán tử sẽ dừng mua và bán hết toàn bộ chứng khoán đang giữ. Toán tử chỉ tiếp tục việc giao dịch khi chỉ số này quay về xuống dưới ngưỡng.

2.3 Mục tiêu bài toán

Ta định nghĩa hàm phần thưởng Reward (r) là sự thay đổi giá trị danh mục đầu tư khi ở trạng thái s thực hiện hành động a trở thành trạng thái s'. Mục tiêu của chúng ta là thiết kế một chiến lược giao dịch mà ở đó giá trị danh mục đầu tư của chúng ta là tối đa:

$$r(s_t, a_t, s_{t+1}) = (b_{t+1} + p_{t+1}^T h_{t+1}) - (b^t + p_t^T h_t) - c_t, \tag{2.4}$$

Để tiếp tục phân rã hàm giá trị, ta tiếp tục định nghĩa

$$h_{t+1} = h_t - k_t^S + k_t^B, \tag{2.5}$$

kết hợp với cả công thức 2.1. Công thức 2.4 giờ đây có thể được viết lại thành

$$r(s_t, a_t, s_{t+1}) = r_H - r_S + r_B - c_t, \tag{2.6}$$

với

$$r_H = (p_{t+1}^H - p_t^H)^T h_t^H, \tag{2.7}$$

$$r_S = (p_{t+1}^S - p_t^S)^T h_t^S, \tag{2.8}$$

$$r_B = (p_{t+1}^B - p_t^B)^T h_t^B, \tag{2.9}$$

Với r_H, r_B, r_S thể hiện cho sự thay đổi giá trị danh mục tương ứng với hành động giữ, mua và bán chứng khoán từ thời điểm t đến thời điểm $t+1$. Từ công thức 2.6 chỉ ra rằng chúng ta tối đa hàm phần thưởng r bằng cách mua và giữ những chứng khoán có xu hướng tăng và bán những chứng khoán có xu hướng giảm tại thời điểm tiếp theo.

Chỉ số nhiễu loạn $turbulence_t$ được liên kết với hàm phần thưởng để xác định mức chấp nhận rủi ro việc thị trường sẽ sụp đổ.Khi chỉ số ở công thức 2.3 vượt quá ngưỡng cho phép, 2.8 sẽ trở thành

$$r_{sell} = (p_{t+1} - p_t)^T k_t, \tag{2.10}$$

chỉ ra rằng chúng ta tối đa danh mục bằng cách tối thiểu r_{sell} bằng cách bán toàn bộ chứng khoán bởi vì khi đó mọi chứng khoán giá sẽ đều giảm.

Mô hình được khởi tạo như sau. p_0 được đặt vào thời điểm $t = 0$ và b_0 là lượng vốn khởi điểm. h và hàm chiến lược $Q_\pi(s, a)$ là 0, and $\pi(s)$ thì được khởi tạo cho mọi trạng thái. Sau đó sẽ tiến hành tương tác với môi trường, cập nhật $Q_\pi(s, a)$. Đây là chiến lược tối ưu theo thuật toán Bellman, như vậy phần thưởng kỳ vọng của việc thực hiện hành động a_t ở trạng thái s_t sẽ là tổng của phần thưởng trực tiếp $r(s_t, a_t, s_{t+1}$ và phần thưởng trong tương lai ở trạng thái s_{t+1}. Để hội tụ, ta cho phần thưởng tương lai này chiết khấu bởi $0 < \gamma < 1$, ta có

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r(s_t, a_t, st + 1) + \gamma \mathbb{E}_{a_{t+1} \sim \pi(st+1)} Q_\pi(s_{t+1}, a_{t+1})]]. \tag{2.11}$$

Với mục tiêu là thiết kế được một chiến lược có khả năng tối đa giá trị danh mục đầu tư $r(s_t, a_t, s_{t+1}$ trong môi trường động, chúng ta phải dùng đến deep learning để giải quyết được bài toán này.

Chương 3

Thuật toán áp dụng

Đầu tiên, chúng ta dùng 3 thuật toán dựa vào phương pháp actor-critic để thiết lập cho agent giao dịch của chúng ta. 3 thuật toán đó là: A2C, DDPG, và PPO. Chiến lược tổng hợp đề ra là kết hợp 3 phương pháp này để có được 1 agent đủ mạnh để có thể giao dịch trên thị trường chứng khoán.

3.1 Actor critic trong RL

Trước đó, Policy Gradient (theo phương pháp Monte Carlo) được cập nhật trên các mẫu ngẫu nhiên có phương sai lớn. Ngoài ra, vấn đề phần thưởng tích lũy trong Policy Gradient bằng 0 sẽ không giúp được việc tăng xác suất hành động tốt và giảm xác suất của hành động "xấu". Điều đó dẫn tới việc cập nhật Policy có thể đi theo hướng không tối ưu. Quay lại công thức của Vanilla Policy Gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_0,a_0,\ldots,s_t,a_t}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right)\right] \mathbb{E}_{r_{t+1},s_{t+1},\ldots,r_T,s_T}\left[G_t\right]$$

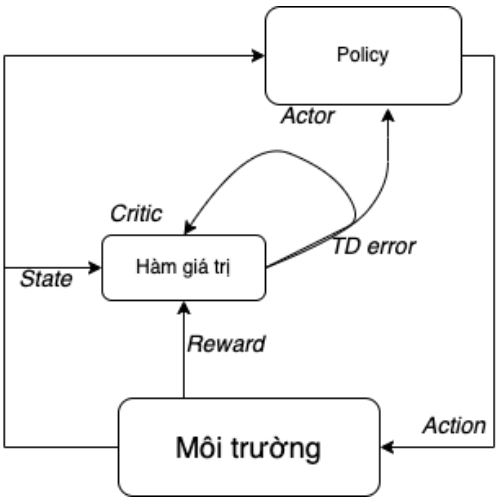
Mà trong đó vế thứ 2 lại bằng với các giá trị kì vọng hàm giá trị Q

$$\mathbb{E}_{r_{t+1},s_{t+1},\ldots,r_T,s_T}\left[G_t\right] = Q\left(s_t, a_t\right)$$

Vậy nên ta có thể viết lại công thức hàm mục tiêu J trong Policy Gradient là

$$\begin{aligned}\nabla_\theta J(\theta) &= \mathbb{E}_{s_0,a_0,\ldots,s_t,a_t}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right)\right] Q_w\left(s_t, a_t\right) \\ &= \mathbb{E}_\tau\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right) Q_w\left(s_t, a_t\right)\right]\end{aligned}$$

Như chúng ta đã biết, trọng số W hàm giá trị Q có thể học được qua 1 mạng nơron. Điều này dẫn tới phương pháp Actor - Critic, là 1 Temporal Difference Learning của Policy Gradient. Trong đó, "Critic" ước lượng hàm giá trị bằng bộ tham số W, có thể là giá trị-hành động (hàm giá trị Q) hoặc giá trị-trạng thái (hàm giá trị V). "Actor" sẽ giúp cập nhật lại Policy theo hướng mà "Critic" đưa ra. Hình thức Q Actor - Critic, cũng như là đơn giản nhất, dùng TD - error và hàm giá trị Q (Action-Value) để tìm được 1 Policy tối ưu.



3.2 Advantage Actor Critic (A2C)

A2C là 1 thuật toán actor-critic đặc trưng, chúng ta dùng nó như 1 thành phần của chiến lược tổng hợp. A2C được dùng để cải thiện cập nhật Policy Gradient. A2C có thêm 1 hàm lợi ích (advantage function), $A(s_t, a_t)$ được tính bằng cách lấy hàm giá trị $Q(s_t, a_t)$ trừ đi hàm giá trị $V(s_t)$. Điều này có nghĩa ta sẽ biết được hành động $Q(s_t, a_t)$ tốt hơn hay tệ hơn so với hành động trung bình tại trạng thái hiện tại $V(s_t)$. Do đó, việc đánh giá không chỉ ở việc hành động đó có tốt hay không mà còn là xem xét hành động ấy có thể trở nên tốt hơn (hay tệ hơn) như thế nào.

$$\nabla J_\theta(\theta) = \mathbb{E}\left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right) A\left(s_t, a_t\right)\right]$$

với

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t) = r(s_t, a_t, a_{t+1}) + \gamma V(s_{t+1}) - V(s_t)$$

A2C dùng các bản sao của cùng 1 agent để cập nhật gradients với các mẫu khác nhau. Mỗi agent sẽ chạy độc lập với nhau trên cùng 1 môi trường. Trong mỗi lần lặp, sau khi tất cả các agent tính xong gradients của bản thân, A2C dùng bộ điều phối để truyền trung bình gradient của các agent kia vào 1 mạng tổng quát. Mạng tổng quát này sẽ cập nhật lại mạng actor và mạng critic. Sự có mặt của 1 mạng tổng quát tăng độ đa dạng của bộ dữ liệu huấn luyện. Việc cập nhật gradient được đồng bộ hóa tiết kiệm chi phí hơn, nhanh hơn và hoạt động tốt hơn với 1 batch size lớn. A2C là một mô hình tốt để giao dịch chứng khoán vì tính ổn định của mình.

3.3 Deep Deterministic Policy Gradient (DDPG)

DDPG được dùng để tối ưu hoá lợi tức đầu tư. DDPG kết hợp cả Q-Learning và Policy Gradient, và dùng mạng nơron như 1 hàm xấp xỉ. Ngược với DeepQLearning học gián tiếp qua bảng giá trị Q và chịu phải giới hạn về mặt không gian, DDPG học trực tiếp từ sự quan sát qua Policy Gradient. Phương pháp này được đề xuất để kết nối 1 cách xác định từ các trạng thái đến các hành động tốt hơn trong 1 không gian liên tục các trạng thái và hành động. Điều mà

DQN bị hạn chế. Trong mỗi lần lặp, agent DDPG thực hiện 1 hành động a ở trạng thái s, và nhận được 1 phần thưởng r và đến được tráng thái s'. Sự chuyển dịch trạng thái (s_t, a_t, s_{t+1}, r_t) này được lưu trữ trong 1 buffer R. 1 tệp N các chuyển dịch này trong R được rút ra và giá trị Qy_i được cập nhật như sau: $y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1} \mid \theta^{\mu'}, \theta^{Q'}\right)\right), i = 1, \cdots, N$. Mạng critic sau đó được cập nhật bằng cách tối thiểu hoá hàm loss $L(\theta^Q)$, hay là hàm giá trị kì vọng giữa mạng critic mục tiêu Q' và mạng critic Q hiện tại:

$$L\left(\theta^Q\right) = \mathbb{E}_{s_t, a_t, r_t, s_{t+1} \sim \text{buffer}}\left[\left(y_i - Q\left(s_t, a_t \mid \theta^Q\right)\right)^2\right]$$

DDPG hiệu quả trong việc thực hiện trên không gian liên tục, vậy nên nó cũng phù hợp trong việc giao dịch chứng khoán.

3.4 Proximal Policy Optimization (PPO)

PPO cũng là 1 thành phần trong chiến lược tổng hợp của chúng ta. PPO được dùng để kiểm soát việc cập nhật Policy Gradient và đảm bảo rằng policy mới sẽ không qua khác biệt so với policy cũ. PPO đơn giản hoá hàm mục tiệu của Trust Region Policy Optimization (TRPO):

$$\nabla J_\theta(\theta) = \max \theta = \hat{\mathbb{E}}[\frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{old}}(a_t \mid s_t)}\hat{A}_t] \text{ sao cho } \hat{\mathbb{E}}[KL[\pi_{old}(\cdot \mid s_t), \pi(\cdot \mid s_t)]] \leq \delta$$

Trong đó, tỉ lệ các xác suất giữa 2 policy cũ và mới là:

$$r_t(\theta) = \frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{old}}\left(a_t \mid s_t\right)}$$

Còn KL được xem đơn giản như 1 hàm phân kỳ để tính độ lệch của của 2 policies và giới hạn độ lệch này không vượt quá δ. PPO cải tiến từ TRPO bằng cách áp dụng một điều kiện "kẹp" lên hàm mục tiêu của TRPO. và hàm mục tiêu thay thế được kẹp của PPO là

$$J^{\text{CLIP}}\left(\theta\right) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}\left(s_t, a_t\right), \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}\left(s_t, a_t\right)\right)\right]$$

khi mà $(r_t(\theta)\hat{A}(s_t, a_t)$ là hàm mục tiêu bình thường của Policy Gradient, và $\hat{A}(s_t, a_t)$ là giá trị ước tính của hàm lợi ích trong actor-critic. Và hàm $\text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)$ kẹp cho tỉ lệ $r_t(\theta)$ chỉ trong đoạn $[1-\epsilon, 1+\epsilon]$ Hàm mục tiêu của PPO sẽ lấy tối thiểu giữa giá trị hàm bị cắt đi và giá trị bình thường của hàm mục tiêu. PPO là phương pháp không khuyến khích sự dịch chuyển lớn trong cập nhật policy ngoài vùng đã được định sẵn, hay "cắt sẵn". Vì thế PPO sẽ giúp cải thiện sự ổn định của huấn luyện mạng policy bằng cách giới hạn chặt chẽ phần cập nhật policy ở mỗi bước huấn luyện. PPO được chọn bởi vì tính ổn định, nhanh chóng và dễ dàng cài đặt và điều chỉnh tham số.

3.5 Chiến lược tổng hợp

Và mục đích của chúng ta là tạo nên 1 chiến lược thật là mạnh mẽ. Vậy nên chúng ta dùng chiến lược tổng hợp để tự động chọn agent nào thể hiện tốt nhất giữa PPO, A2C, DDPG để thực hiện giao dịch dựa trên tỉ lệ Sharpe. Các bước trong chiến lược được miêu tả như sau:

Bước 1. Chúng ta dùng 1 cửa sổ n tháng để huấn luyện lại đồng thời 3 agents. Trong báo cáo này, chúng ta huấn luyện lại 3 agents mỗi 3 tháng - tức 1 quý trong năm.

Bước 2. Chúng ta kiểm tra lại 3 agents bằng cách cứ sau mỗi quý giao dịch trên dữ liệu xác thực, kiểm tra lại khả năng giao dịch của các agents với tỉ lệ Sharpe. Tỉ lệ Sharpe được tính như sau:

$$\textit{Sharpe ratio } = \frac{\bar{r}_p - r_f}{\sigma_p}$$

với $\bar{r}_p$ là lợi nhuận đầu tư dự kiến, r_f là lãi suất phi rủi ro, và σ_p là là độ lệch chuẩn của lợi tức vượt quá của danh mục đầu tư. Chúng ta cũng điều chỉnh mức ngại rủi ro trong bằng cách sử dụng chỉ số nhiễu loạn trong giai đoạn kiểm thử

Bước 3. Sau khi chọn được agent tốt nhất, chúng ta sẽ chọn agent đó và tiếp tục giao dịch trong quý kế tiếp.

Đằng sau việc lựa chọn như thế là bởi mỗi agent nhanh nhạy đối với các xu hướng khác nhau trên thị trường. Agent này có thể làm tốt khi gặp xu hướng tăng, ngược lại agent này giao dịch không tốt khi gặp xu hướng giảm. Agent khác thì lại được điều chỉnh thích hợp khi thị trường biến động. Tỉ lệ Sharpe càng cao của 1 agent chứng tỏ lợi nhuận của nó tốt so với mức rủi ro đầu tư mà agent đó thực hiện. Do đó, chúng ta chọn agent nào mà giao dịch của nó có thể tối đa hoá lợi nhuận khi điều chỉnh theo tỉ lệ gia tăng rủi ro

Chương 4

Thực nghiệm

4.1 Thông tin dữ liệu đầu vào

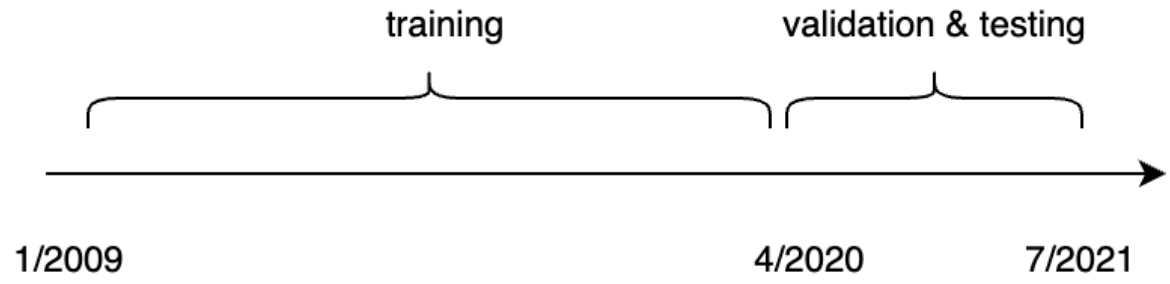
Để chuẩn bị dữ liệu, chúng em đã sử dụng tool (phamdinhkhanh/vnquant) dùng để crawl các dữ liệu thị trường chứng khoán của 30 công ty Việt Nam trong khoảng thời gian từ năm 2010 đến 2021. Các trường dữ liệu bao gồm:

- Datadate: dữ liệu của ngày khảo sát
- Tic: Symbol của công ty trên sàn giao dịch
- High, low: giá cổ phiếu cao nhất và thấp nhất của ngày khảo sát
- Open, close: giá cổ phiếu khởi điểm và kết thúc của ngày khảo sát
- Volume: số lượng cuộc giao dịch được thực hiện trong ngày.

Tuy nhiên, dữ liệu như vậy chưa đủ thông tin để huấn luyện mô hình, chúng ta cần chỉnh sửa dữ liệu gốc thêm nhiều loại dữ liệu sinh ra bởi dữ liệu gốc:

- MACD (Moving Average Convergence Divergence): chỉ số độ lớn, hướng, động lượng và thời gian (momentum) của một xu hướng trong giá cổ phiếu.
- RSI (Relative Strength Index): chỉ số biểu hiện mức độ thay đổi của giá cổ phiếu gần đây. Nếu như giá cổ phiếu thay đổi chạm đến ngưỡng báo động (support line của cổ phiếu), chúng ta sẽ thực hiện thao tác bán cổ phiếu. Hoặc khi giá cổ phiếu đạt đến ngưỡng resistance level (overbought), chúng ta thực hiện thao tác mua cổ phiếu. RSI được tính bởi close price.
- CCI (Commodity Channel Index) được tính bởi low, high và closed price. CCI thể hiện mức độ tương quan giữa giá cổ phiếu hiện tại với giá cổ phiếu trung bình trên một đơn vị thời gian. Từ đó ta có thể quyết định bán hay mua cổ phiếu hay không.
- ADX (Average Directional Index): cũng được tính từ low, high và closed priced. ADX thể hiện xu hướng của sự thay đổi trong giá cổ phiếu.
- Turbulence: chỉ số thể hiện sự rủi ro của cổ phiếu nếu như thị trường gặp khủng hoảng

Sau đó, chúng em có chuẩn bị 2 bộ dữ liệu, một bộ dữ liệu trên thị trường Mỹ và thị trường Việt Nam. Trong suốt quá trình training, bọn em chỉ dùng duy nhất chiến thuật Ensemble trên cả hai bộ dữ liệu. Bộ dữ liệu được phân bố để training và testing như sau:



Hình 4.1: Phân tách bộ dữ liệu

4.2 Thông tin dữ liệu đầu ra

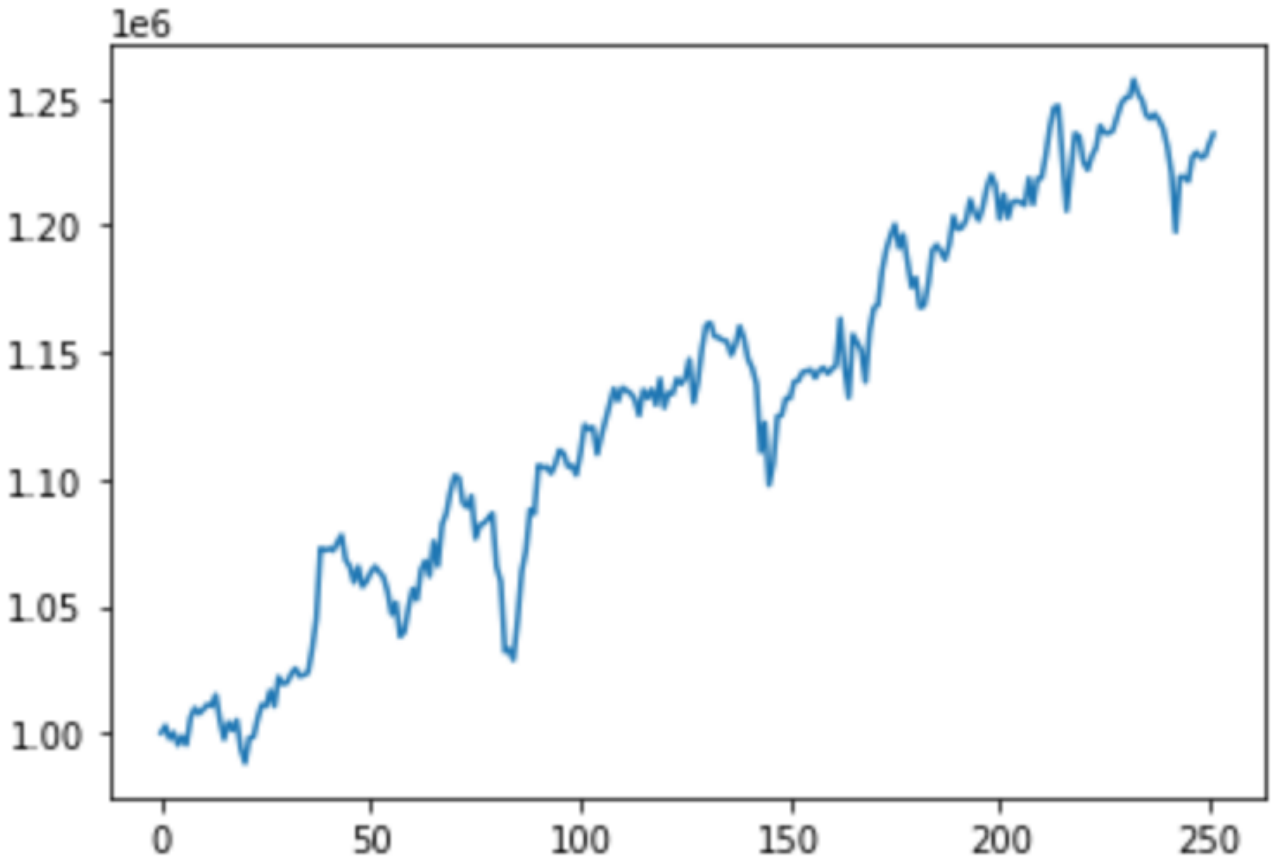
Chúng ta sẽ quan tâm một số metrics trả về như sau:

- Cumulative return: lợi nhuận tích lũy, trong suốt quá trình trade trên bộ dữ liệu test, lợi nhuận càng cao chứng tỏ mô hình trade càng tốt.
- Annualized return: lợi nhuận tích lũy hằng năm được kiếm bởi các agents.
- Annualized volatility: là thống kê về sự phân tán của lợi nhuận đối với một chỉ số thị trường hằng năm.
- Sharpe ratio: là tỷ lệ lợi nhuận thu được là bao nhiêu trên một đơn vị rủi ro khi đầu tư cổ phiếu được theo một chiến lược nào đó.
- Max drawdown: là mức sụt giảm vốn sâu nhất trong suốt quá trình đầu tư. Max drawdown càng thấp càng thể hiện sự an toàn và ổn định khi quyết định đầu tư cổ phiếu.

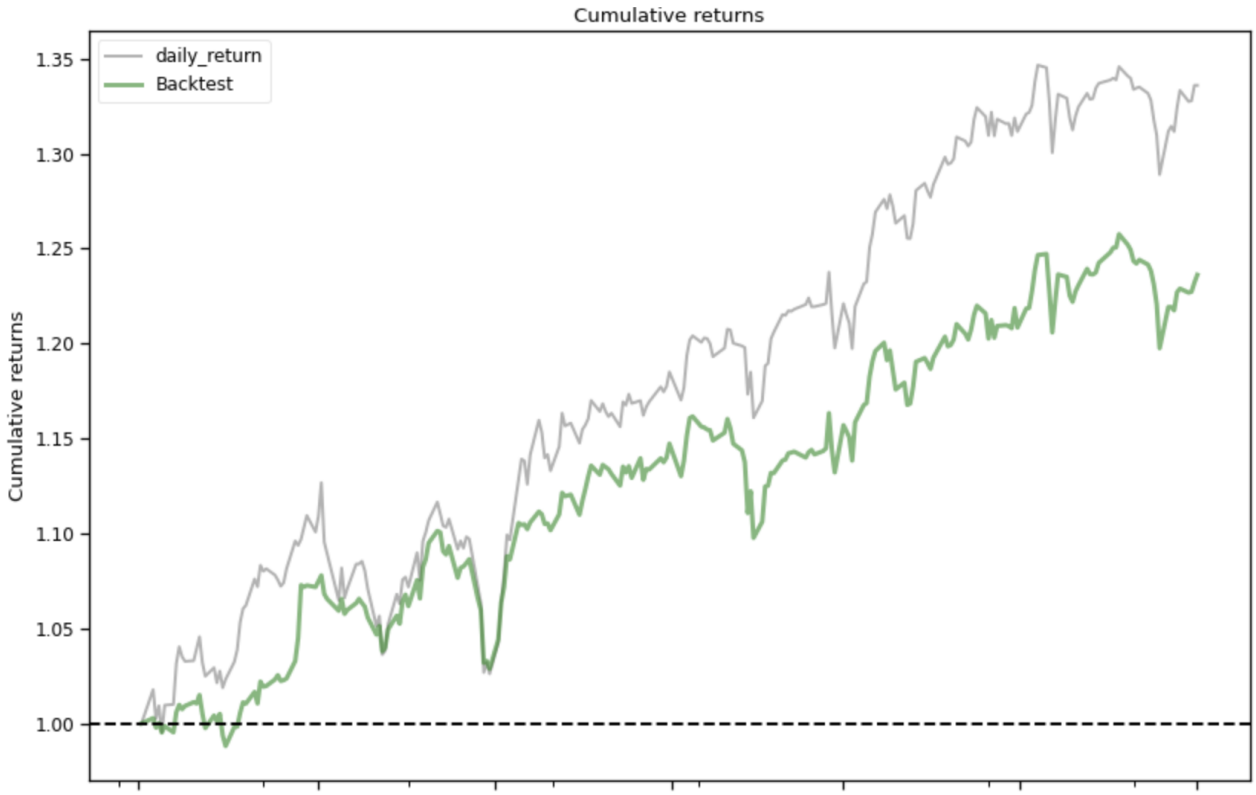
4.3 Kết quả thực nghiệm

4.3.1 Bộ dữ liệu thị trường chứng khoán Mỹ

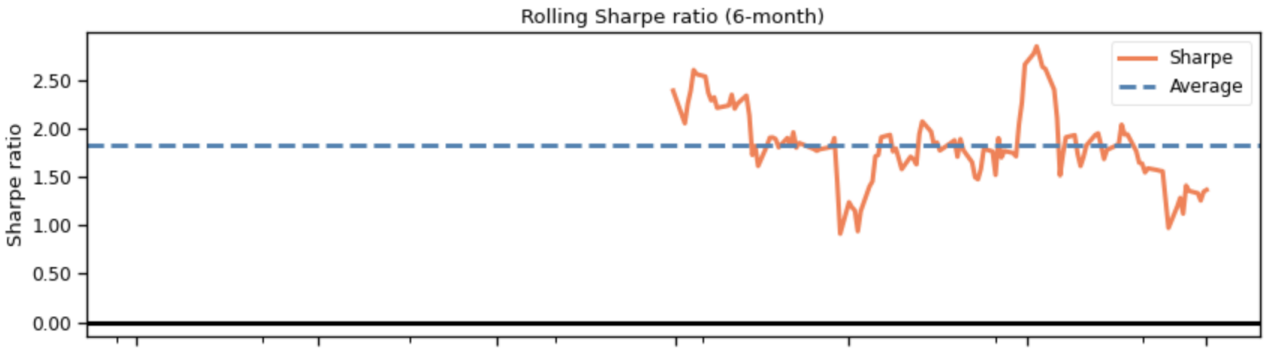
Iter	Val Start	Val End	Model	A2C Sharpe	PPO Sharpe	DDPG Sharpe
126	2020-04-02	2020-07-02	PPO	0.215905	0.231723	0.201333
189	2020-07-02	2020-10-01	PPO	0.0809334	0.282006	0.105878
252	2020-10-01	2021-12-31	PPO	0.16838	0.202466	0.153827
315	2020-12-31	2021-04-05	PPO	0.149787	0.277354	0.262803



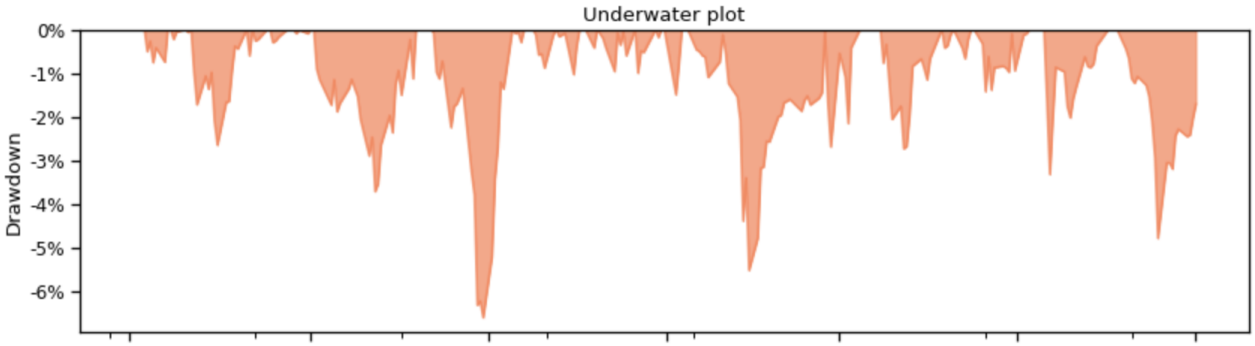
Hình 4.2: Portfolio



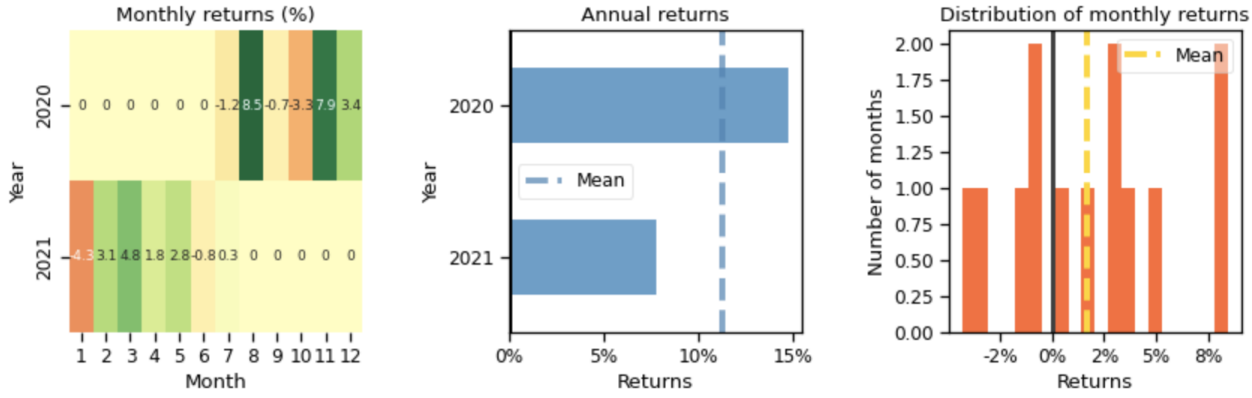
Hình 4.3: Lợi nhuận tích luỹ



Hình 4.4: Sharpe Ratio



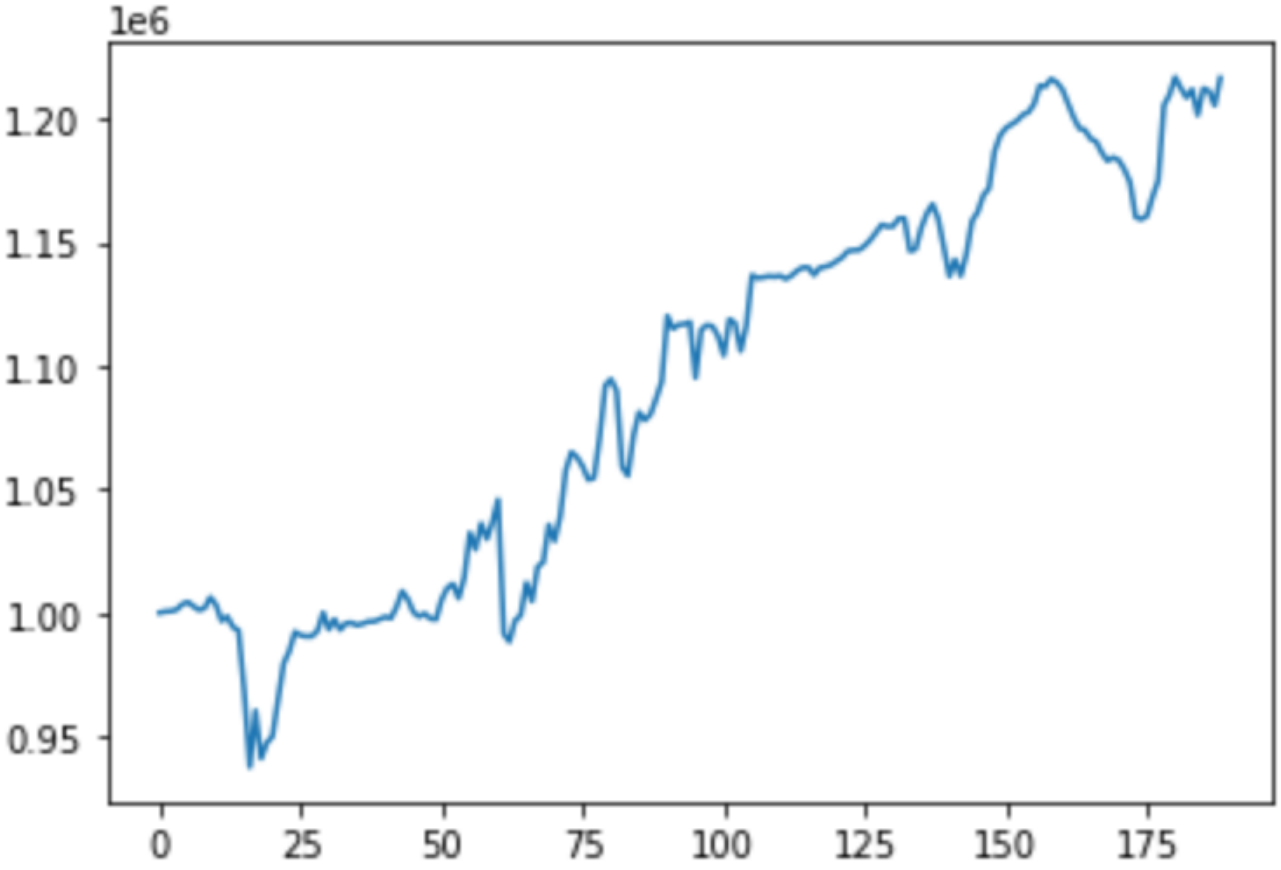
Hình 4.5: Drawdown



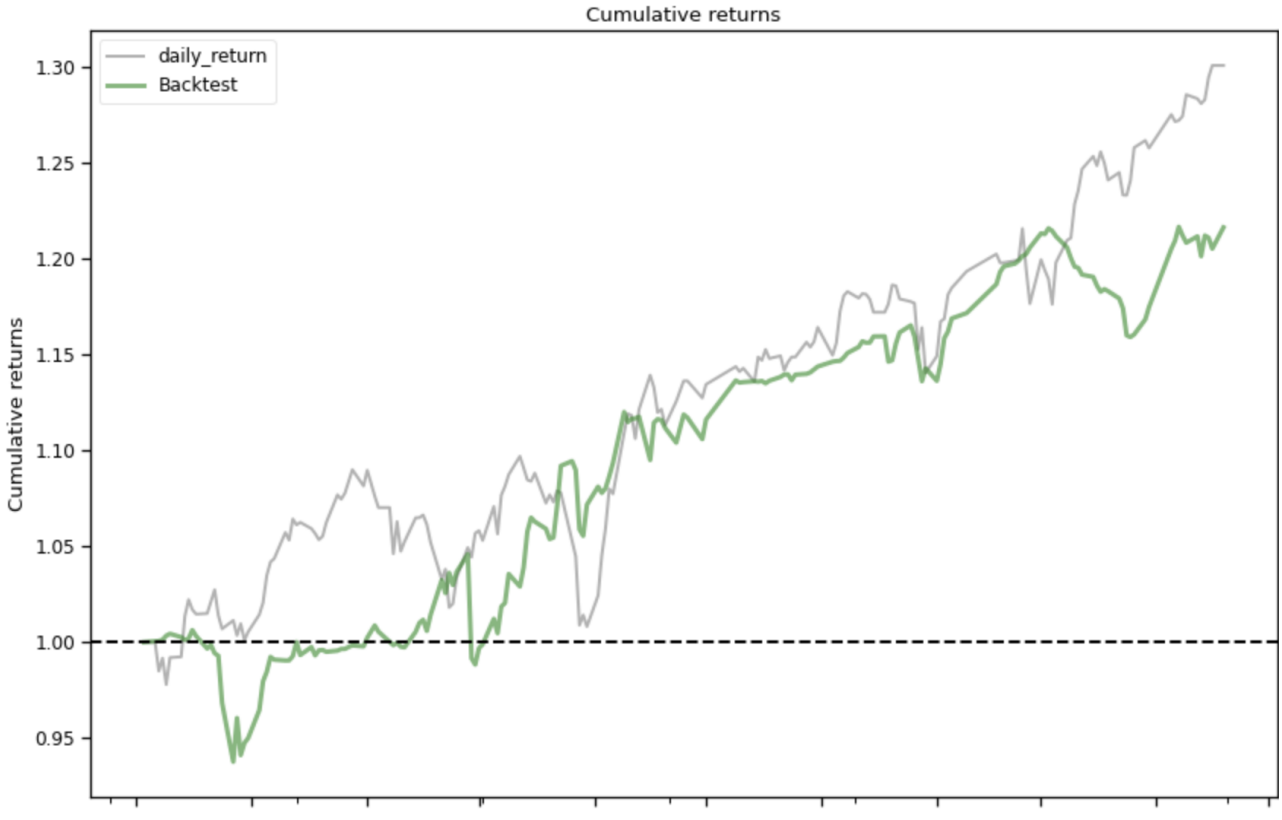
Hình 4.6: Lợi nhuận theo các chu kỳ

4.3.2 Bộ dữ liệu thị trường chứng khoán Việt Nam

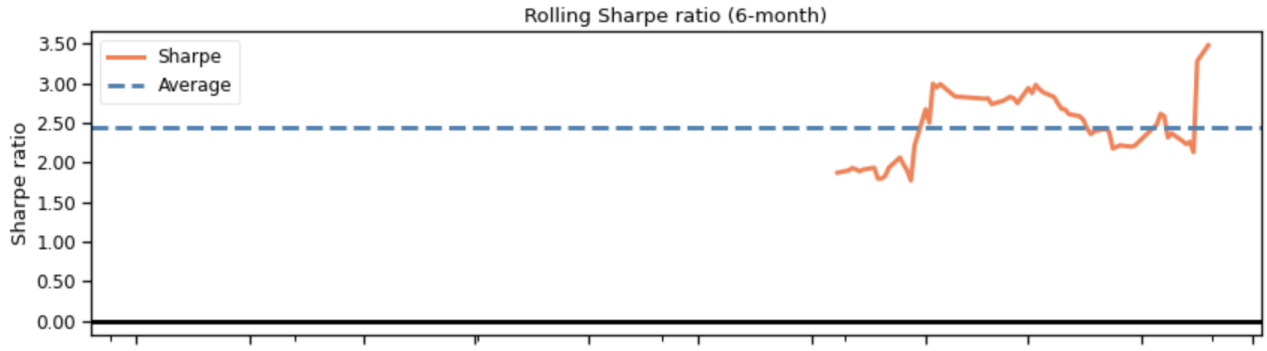
Iter	Val Start	Val End	Model	A2C Sharpe	PPO Sharpe	DDPG Sharpe
126	2020-04-03	2020-07-03	DDPG	0.108128	-0.0336746	0.151153
189	2020-07-03	2020-10-01	PPO	-0.109512	0.076417	0.0102378
252	2020-10-01	2021-01-08	PPO	0.487705	0.699703	0.404334



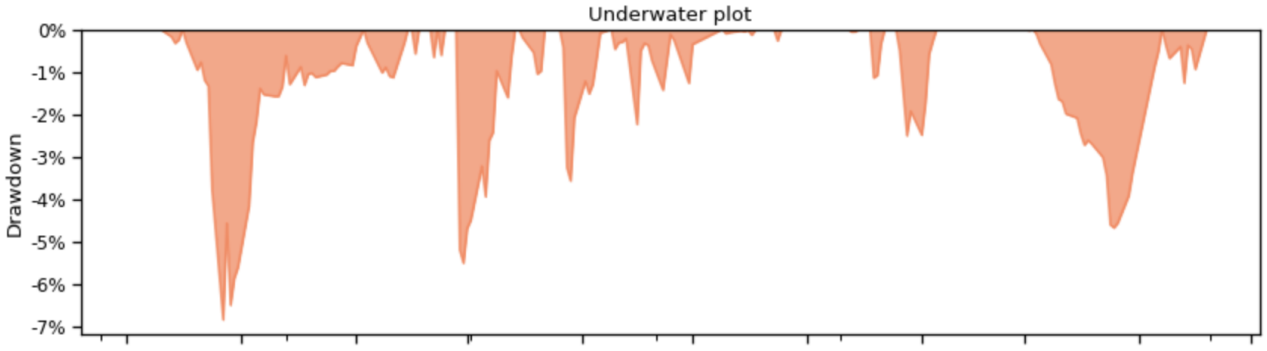
Hình 4.7: Portfolio



Hình 4.8: Lợi nhuận tích luỹ



Hình 4.9: Sharpe Ratio



Hình 4.10: Drawdown

Hình 4.11: Lợi nhuận theo các chu kỳ

4.4 Kết luận

Market	US	VN
Cumulative return	23.614%	21.651%
Annualized return	23.614%	29.863%
Annualized volatility	12.08 %	14.027%
Sharpe ratio	1.82	1.94
Max Drawdown	-6.589%	-6.83%

Từ các báo cáo số liệu trên, ta nhận thấy mô hình PPO thường được chọn bởi Ensemble nhất để trade trong đa số các đợt giao dịch. Nhìn chung, ở cả hai thị trường, các chỉ số chênh lệch không quá nhiều, tuy nhiên về lợi nhuận tích luỹ của thị trường Mỹ cao hơn ở thị trường VN xấp xỉ 2% cùng với đó là mức độ an toàn hơn thị trường VN do Max Drawdown có khoảng thời gian sụt giảm sâu hơn.

Tài liệu tham khảo

[1] Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy - Hongyang Yang et el.
`Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy`

[2] Code for Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy - Hongyang Yang et el.
`Github`