SỬ DỤNG HỌC TĂNG CƯỜNG ĐỂ ĐƯA RA QUYẾT ĐỊNH TỐI ƯU: THỬ NGHIỆM ĐẦU TƯ BITCOIN

TS. Phạm Văn Khánh[*], Phan Khôi Nguyên[**], Bùi Minh Đức[***]

Tóm tắt

Trong bài báo này chúng tôi nghiên cứu các thuật toán học tăng cường (Reinforcement Learning: RL) và áp dụng nó trong việc tối ưu hóa lợi nhuận của một khoản đầu tư. Chúng tôi phải xây dựng một tác nhân (thực thể đưa ra quyết định mua hay bán Bitcoin trên thị trường) cùng với môi trường mà tác nhân hoạt động (giá cả, các chỉ báo, các thông tin về chính sách, chính trị và kinh tế thế giới). Mỗi tác nhân sẽ có một tập hợp các hành động có thể được thực hiện (mua, mua thêm, bán, bán thêm hoặc nắm giữ và không làm gì cả) cùng với các phần thưởng tương ứng với mỗi hành động đó (lãi bao nhiêu, lỗ bao nhiêu?). Chúng tôi sử dụng dữ liệu trong quá khứ để huấn luyện tác nhân sau đó sử dụng dữ liệu mới để thử nghiệm. Dùng dữ liệu theo khung giờ và thử nghiệm 1000 lần chúng tôi thu được lợi nhuận trung bình trên 2%/giờ.

1. Giới thiệu

Trong đầu tư chứng khoán cũng như trong đầu tư tiền điện tử (Crypto) người ta coi trọng "time in the market" (thời gian đã từng ở trong thị trường) nhưng chúng tôi ở đây quan tâm đến bài toán "timing the market" (chọn thời điểm vào và thời điểm ra tối ưu) bởi vì với sự giúp đỡ của thuật toán thì dữ liệu trong quá khứ thông qua quá trình huấn luyện của tác nhân (một mạng nơ ron hay một Bot giao dịch tự động) đã là "time in the market" rồi.

Vấn đề ra quyết định dưới điều kiện không chắc chắn có thể được chia thành hai phần. Đầu tiên, làm thế nào để chúng ta tìm hiểu về thế giới? Điều này liên quan đến cả vấn đề *mô hình hóa sự không chắc chắn ban đầu* của chúng ta về thế giới và *rút ra kết luận từ bằng chứng và niềm tin ban đầu* của chúng ta. Thứ hai, với những gì chúng ta hiện đang biết về thế giới, chúng ta nên quyết định phải làm gì, có tính đến các sự kiện và quan sát trong tương lai có thể thay đổi kết luận của chúng ta? Đưa ra một số lựa chọn thay thế, đâu sẽ là lựa chọn hợp lý trong một tình huống cụ thể tùy thuộc vào một mục tiêu và mong muốn của bạn? Để trả lời câu hỏi này, chúng ta cần phát triển một khái niệm tốt về hành vi hợp lý. Điều này sẽ phục vụ hai mục đích: Thứ nhất, điều này có thể phục vụ như một lời giải thích cho những gì động vật và con người (nên) làm. Thứ hai, nó sẽ hữu ích cho việc phát triển các mô hình và thuật

* *Viện Toán và các khoa học ứng dụng, Đại học Thăng Long Hà Nội*

** *Lớp 11 D1- THPT Tạ Quang Bửu – Bách Khoa – Hai Bà Trưng – Hà Nội*

*** *11A2, chuyên Toán, Trường PTTH chuyên Khoa học tự nhiên - ĐH Khoa học Tự nhiên - ĐHQGHN*

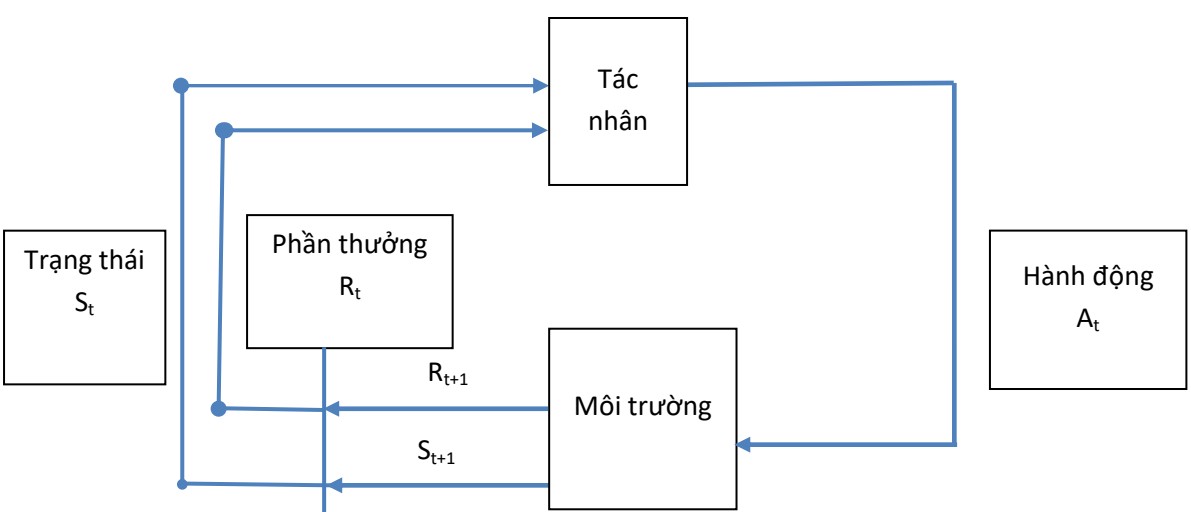
toán để ra quyết định tự động trong các nhiệm vụ phức tạp.

Một vấn đề đặc biệt thú vị trong thiết lập này là học tăng cường. Vấn đề này phát sinh khi môi trường không xác định và người học phải đưa ra quyết định chỉ thông qua tương tác, điều này chỉ đưa ra phản hồi hạn chế. Do đó, tác nhân học tập không có quyền truy cập vào các hướng dẫn chi tiết về nhiệm vụ nào sẽ thực hiện, cũng như về cách thực hiện. Thay vào đó, nó thực hiện các hành động, ảnh hưởng đến môi trường và thu được một số quan sát (tức là đầu vào cảm giác) và phản hồi, thường là dưới dạng phần thưởng tương ứng với mong muốn của tác nhân. Vấn đề học tập sau đó được hình thành là vấn đề học cách hành động để tối đa hóa tổng số phần thưởng. Trong các hệ thống sinh học, phần thưởng thực chất được gắn kết với các tín hiệu liên quan đến nhu cầu cơ bản. Trong các hệ thống nhân tạo, chúng ta có thể chọn các tín hiệu phần thưởng để củng cố hành vi đạt được các mục tiêu của nhà thiết kế.

Học tăng cường (tiếng Anh: *reinforcement learning* là một lĩnh vực con của học máy, nghiên cứu cách thức một *agent* trong một *môi trường* nên chọn thực hiện các *hành động* nào để cực đại hóa một khoản *thưởng* (*reward)* nào đó về lâu dài. Các thuật toán học tăng cường cố gắng tìm một *chiến lược* ánh xạ các *trạng thái* của thế giới tới các hành động mà agent nên chọn trong các trạng thái đó [1].

RL là một loại máy học cho phép chúng tôi tạo ra các tác nhân AI học hỏi từ môi trường bằng cách tương tác với nó để tối đa hóa phần thưởng tích lũy của nó. Giống như cách chúng ta học cách đi xe đạp, AI học nó bằng cách thử và sai, các tác nhân trong thuật toán RL được khuyến khích bằng các hình phạt cho hành động xấu và phần thưởng cho hành động tốt.

Sau mỗi hành động, tác nhân nhận được thông tin phản hồi. Phản hồi bao gồm phần thưởng và trạng thái tiếp theo của môi trường. Phần thưởng thường được xác định bởi một con người. Nếu chúng ta sử dụng phép tương tự với chiếc xe đạp, chúng ta có thể định nghĩa phần thưởng là khoảng cách từ điểm xuất phát ban đầu.



Hình 1. Sơ đồ học tăng cường của một tác nhân

Sau đó, đối với mỗi lần lặp, một tác nhân lấy trạng thái hiện tại (S_t), chọn hành động tốt nhất (dựa trên dự đoán mô hình) (A_t) và thực hiện nó trên một môi trường. Sau đó, môi trường trả về phần thưởng (R_{t+1}) cho một hành động nhất định và thiết lập trạng thái mới (S_{t+1}). Quá trình lặp lại cho đến khi kết thúc.

2. Cơ sở toán học của học tăng cường

2.1. Quá trình quyết định Markov (Markov Decision Process: MDP)

Quá trình quyết định Markov là một bộ gồm 4 thành phần (S, A, P, R), trong đó:

+ S là tập các trạng thái,

+ A là tập các hành động,

+ $P : S \times A \rightarrow P(S)$ là ma trận xác suất chuyển trạng thái dưới tác động của các hành động,

+ $R : S \times A \rightarrow \square$ là phân phối phần thưởng.

Vì vậy, thực hiện bất kỳ hành động nào a ∈ A tại bất kỳ trạng thái nào $s \in S$, $P(\cdot \mid s,a)$ xác định xác suất của trạng thái tiếp theo và $R(\cdot \mid s,a)$ là phân phối phần thưởng. Một chính sách $\pi : S \rightarrow P(A)$ ánh xạ mọi trạng thái s ∈ S thành phân phối xác suất π (· | s) trên A.

Học tăng cường hợp nhất tính toán xấp xỉ hàm và tối ưu hóa mục tiêu, ánh xạ các cặp trạng thái-hành động với kì vọng phần thưởng. Các thuật toán này xem xét hành vi của tác nhân tại thời điểm học tập với cấu trúc phần thưởng hành động của nó, thưởng cho tác nhân khi hành động đã chọn là tốt, và phạt trong trường hợp ngược lại.

2.2. Thuật toán Q-Learning

Thuật toán Q-Learning [1] tạo ra một ma trận chính xác để tác nhân có thể tối đa hóa phần thưởng của mình về lâu dài. Cách tiếp cận này chỉ thực tế cho môi trường hạn chế, với không gian hạn chế để quan sát, do sự gia tăng số lượng trạng thái hoặc hành động gây ra một thuật toán sai hành vi. Q-Learning là một RL không có chính sách, không có mô hình dựa trên Phương trình Bellman, trong đó v đề cập đến giá trị tối ưu của nó:

$$v(s) = E\left[R_{t+1} + \lambda v(S_{t+1}) \mid S_t = s\right]$$

E đề cập đến kỳ vọng, trong khi γ đề cập đến hệ số chiết khấu cho các phần thưởng phía trước, và viết lại nó dưới dạng Q-value:

$$Q^{\pi}(s,a) = E[r_{t+1} + \lambda r_{t+2} + \lambda^2 r_{t+3} + \ldots \mid s,a] = E_{s'}[r + \lambda Q^{\pi}(s',a') \mid s,a]$$

Trong đó giá trị tối ưu Q^* có thể được diễn đạt như sau:

$$Q^*(s,a) = E_{s'}[r + \lambda \max_{a'} Q^*(s',a') \mid s,a]$$

Mục tiêu của Q-Learning là tối đa hóa chính sách lặp lại đáy giá trị Q, chính sách này điều chỉnh vòng lặp giữa đánh giá chính sách và cải tiến chính sách. Đánh giá chính sách ước tính giá trị của hàm V với chính sách tham lam, đã nhận được từ chính sách cuối cùng sự cải tiến. Mặt khác, cải tiến chính sách cập nhật chính sách với hành động mà tối đa hóa hàm V cho mỗi trạng thái. Phép lặp giá trị cập nhật hàm V dựa trên Phương trình Bellman tối ưu như sau:

$$V^*(s) = max_a E[R_{t+1} + \gamma V^**(S_{t+1}) \mid S_t = s, A_t = a] = max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma V^**(s')]$$

Khi phép lặp hội tụ, chính sách tối ưu nhận được bằng cách áp dụng đối số là max chức năng cho tất cả các trạng thái.

$$\pi(s) = max \sum_{s',r} p(s', r \mid s, a)[r + \gamma v*(s')]$$

Do đó, phương trình cập nhật được thay thế bằng công thức sau, trong đó α là tốc độ học:

$$Q(s_t, a_t) = Q(s_t,\ a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a_t) - Q(s_t,\ a_t)] \qquad (2)$$

2.3. Bài toán dừng tối ưu

Thuật toán Q-Learning sử dụng quá trình học tập trên dữ liệu huấn luyện để đưa ra quyết định tối ưu. Trong đầu tư Bitcoin hay chứng khoán thì quyết định tối ưu là mua được giá rẻ nhất hay bán được giá đắt nhất. Thời điểm mua được rẻ nhất hay bán lúc đắt nhất được gọi là thời điểm dừng tối ưu.

Chúng ta xem xét các bài toán dừng tối ưu có dạng $\sup_\tau Eg(\tau; X_\tau)$, trong đó $X = (X_n)_{n=0}^N$ là một quá trình Markov thời gian rời rạc nhận giá trị trong R^d và giá trị tối ưu trên tất cả các thời gian điểm dừng τ dựa trên quan sát của X.

Thông thường trong thực tế $X = (X_n)_{n=0}^N$ là một chuỗi giá cả (Giá cổ phiếu, giá vàng, giá ngoại tệ, giá café, giá kim loại…), còn $g(t; X_t)$ là một hàm trên giá cả: có thể là lợi nhuận hay chi phí….

Về mặt hình thức, điều này chỉ bao gồm các tình huống mà quyết định dừng chỉ có thể được thực hiện tại một số thời điểm nhất định. Nhưng trên thực tế, tất cả các vấn đề thời điểm dừng liên tục có liên quan có thể được tính gần đúng với các phiên bản tùy ý về thời gian. Giả định Markov có nghĩa là không mất tính tổng quát. Chúng tôi tạo ra nó vì nó đơn giản hóa việc trình bày và nhiều vấn đề quan trọng đã có ở dạng Markovian. Nhưng mọi bài toán dừng tối ưu đều có thể được thực hiện Markov bằng cách đưa tất cả thông tin liên quan từ quá khứ vào trạng thái hiện tại của X (mặc dù với cái giá phải trả là tăng số chiều của bài toán).

Gọi $X = (X_n)_{n=0}^N$ là quá trình Markov thời gian rời rạc có giá trị $\in R^d$ không gian xác suất (Ω; F; P), trong đó N và d là các số nguyên dương. Chúng tôi ký hiệu F_n là σ-đại số được tạo ra bởi X_0; X_1; ...; X_n (Ta có thể hiểu F_n là tất cả những thông tin về giá cả, các thông tin xấu và tốt liên quan tới X hay F_n là lý lịch của X, môi trường của X) và gọi một biến ngẫu nhiên $\tau: \Omega \to \{0, 1, 2, ..., N\}$ là X- thời điểm dừng X nếu biến cố $\{\tau = n\} \in F_n, n = 0, 1, 2, ..., N$

Mục đích của chúng ta là phát triển một phương pháp học sâu có thể tìm hiểu một

cách hiệu quả các chính sách cho bài toán dừng tối ưu $\sup_{\tau \in T} Eg(\tau; X_\tau)$,

trong đó $g: \{0,1,2,...,N\} \times R^d \to R$ là một hàm có thể đo được và T biểu thị tập hợp tất cả các X-thời điểm dừng. Để đảm bảo rằng bài toán (1) được xác định rõ ràng và tồn tại ít nhất một nghiệm ta giả sử rằng g thỏa mãn điều kiện tích phân: $E|g(n, X_n)| < \infty; \forall n \in \{0,1,2,...,N\}$

Để có thể tính được khoảng tin cậy cho giá trị tối ưu (1), chúng ta sẽ phải đưa ra giả thiết mạnh hơn một chút $E|g(n, X_n)|^2 < \infty; \forall n \in \{0,1,2,...,N\}$

Biểu diễn thời điểm dừng dưới dạng các quyết định dừng

Bất kỳ thời điểm dừng X nào cũng có thể được phân tách thành một chuỗi các quyết định dừng 0-1. Về nguyên tắc, quyết định có dừng quá trình tại thời điểm n nếu nó chưa bị dừng trước đó, có thể được thực hiện dựa trên toàn bộ diễn biến của X từ thời điểm 0 cho đến n. Nhưng để dừng tối ưu quá trình Markov X, ta cần đưa ra một hàm gọi là **hàm quyết định dừng**:

$$f_n(X_n): R^d \to \{0,1\}; n \in \{0,1,2,..N\} \quad (3)$$

Định lý 1 dưới đây là một định lý quen thuộc và đóng vai trò là cơ sở lý thuyết của phương pháp này.

Xét bài toán dừng tối ưu

$$V_n = \sup_{\tau \in T_n} Eg(\tau; X_\tau), \quad (4)$$

trong đó T_n là tập hợp tất cả các thời điểm dừng thỏa mãn $n \le \tau \le N$

Rõ ràng, T_N bao gồm phần tử duy nhất $\tau_N \equiv N$, và người ta có thể viết $\tau_N \equiv N f_N(X_N)$ trong đó $f_N = 1$.

Hơn nữa với $n \in \{0,1,2,..,N\}$ và một dãy các hàm đo được:

$f_n, f_{n+1}, ..., f_N: R^d \to \{0,1\}, f_N \equiv 1$, ta xác định thời điểm dừng:

$$\tau_n = \sum_{m=n}^{N} m f_m(X_m) \prod_{j=n}^{m-1} \left[1 - f_j(X_j)\right] \quad (5)$$

và đây là một thời điểm dừng thuộc T_n.

Nhận xét:

$$\tau_n = \sum_{m=n}^{N} m f_m(X_m) \prod_{j=n}^{m-1} \left[1 - f_j(X_j)\right]$$

$$= n f_n(X_n) + (n+1) f_{n+1}(X_{n+1}) \left[1 - f_n(X_n)\right] + (n+2) f_{n+2}(X_{n+2}) \left[1 - f_n(X_n)\right] \left[1 - f_{n+1}(X_{n+1})\right] +$$

$$+ N f_N(X_N) \left[1 - f_n(X_n)\right] \left[1 - f_{n+1}(X_{n+1})\right] .... \left[1 - f_{N-1}(X_{N-1})\right]$$

- Nếu quá trình dừng tại một thời điểm $N \ge n_0 \ge n$ khi đó:

$$\begin{aligned}\tau_{n_0} &= n.0+(n+1).0[1-0]+(n+2).0[1-0][1-0]+n_0 f_{n_0}(X_{n_0})[1-f_n(X_n)]...[1-f_{n_0}(X_{n_0})]\\ &+Nf_N(X_N)[1-f_n(X_n)][1-f_{n+1}(X_{n+1})]....[1-f_N(X_N)]\\ &=0+0+0+n_0.1[1-0]...[1-0]+...+\\ &+Nf_N(X_N)[1-f_n(X_n)][1-f_{n+1}(X_{n+1})]...[1-f_{n_0}(x_{n_0})]...[1-f_{N-1}(X_{N-1})]\\ &=0+n_0+...+\\ &+Nf_N(X_N)[1-f_n(X_n)][1-f_{n+1}(X_{n+1})]...[1-1]...[1-f_{N-1}(X_{N-1})]\\ &=n_0\end{aligned}$$

Mỗi một quyết định mua hay bán được xác định bởi hàm $f_n(X_n)$, mỗi hàm này tương ứng với 1 hành động trong tập hành động A_n và sigma trường F_n tương ứng với không gian trạng thái S_n. Quá trình huấn luyện trong thuật toán học tăng cường là quá trình xấp xỉ hàm $f_n(X_n)$ bằng 1 mạng nơ ron để hàm mục tiêu $Eg(\tau; X_\tau)$ đạt cực đại.

3. Ứng dụng học tăng cường trong đầu tư Bitcoin

Bây giờ chúng ta sẽ thiết kế một robot để tự động đầu tư trên sàn chứng khoán. Trước hết chúng ta cần thiết lập nền tảng từng bước cho môi trường giao dịch Bitcoin, nơi chúng ta có thể phát triển, kiểm tra và thử nghiệm. Khi có môi trường, chúng ta có thể tạo một tác nhân Học tăng cường để mô phỏng các giao dịch tiền điện tử.

3.1. Môi trường giao dịch Bitcoin

Môi trường chứa tất cả các hàm cần thiết để chạy một tác nhân và cho phép nó học hỏi. Môi trường trong giao dịch Bitcoin sẽ gồm **action_space,** sẽ chứa tất cả các hành động có thể cho một tác nhân thực hiện trong môi trường và **state_size:** chứa tất cả dữ liệu của môi trường mà tác nhân quan sát được.

Điều đầu tiên mà chúng ta cần xem xét cách con người quyết định giao dịch mà chúng ta muốn thực hiện. Chúng ta quan sát những gì trước khi quyết định giao dịch?

Thông thường, một nhà giao dịch chuyên nghiệp rất có thể sẽ xem xét một số biểu đồ về hành vi của giá cả bao gồm các chỉ báo kỹ thuật. Từ đó, họ sẽ kết hợp thông tin trực quan này với kiến thức trước đây của họ về các hành động giá tương tự để đưa ra quyết định sáng suốt về hướng giá có khả năng di chuyển.

Vì vậy, chúng tôi cần chuyển những hành động của con người này thành mã để tác nhân do chúng tôi tạo ra có thể hiểu hành động giá theo cách tương tự. Chúng tôi muốn **state_size** chứa tất cả các biến đầu vào mà chúng tôi cần tác nhân của mình xem xét trước khi thực hiện một hành động. Ở đây, chúng tôi muốn tác nhân của chúng tôi có thể "xem" các điểm dữ liệu chính của thị trường (giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa và khối lượng giao dịch hàng ngày) trong 50 ngày qua, cũng như một vài điểm dữ liệu khác như số dư tài khoản, các vị thế mở hiện tại và lợi nhuận hiện tại.

Chúng tôi muốn tác nhân của chúng tôi cho mỗi bước thời gian xem xét hành động giá dẫn đến giá hiện tại, cũng như trạng thái danh mục đầu tư để đưa ra quyết định sáng suốt cho hành động tiếp theo. Nói về hành động, tác nhân của chúng tôi sẽ có **action_space** sẽ bao gồm ba khả năng: **mua, bán** hoặc **giữ** trong bước thời gian hiện tại.

Nhưng điều này không đủ để biết số lượng Bitcoin để mua hoặc bán mỗi lần. Vì vậy, chúng tôi sẽ cần tạo một không gian hành động có số lượng loại hành động (mua, bán và giữ) riêng biệt, cũng như một phổ liên tục các số tiền để mua / bán (0-100% số dư tài khoản / quy mô vị thế tương ứng).

Điều cuối cùng cần xem xét trước khi triển khai môi trường là phần thưởng. Chúng tôi muốn thúc đẩy lợi nhuận dài hạn, vì vậy đối với mỗi bước, chúng tôi sẽ tính toán chênh lệch số dư tài khoản giữa bước trước đó và bước hiện tại. Chúng tôi muốn rằng tác nhân của chúng tôi có thể duy trì số dư cao hơn trong thời gian lâu hơn, thay vì mục tiêu nhanh chóng kiếm được tiền bằng cách sử dụng các chiến lược không bền vững.

Ở mỗi bước, tác nhân của chúng tôi sẽ chọn và thực hiện hành động mua, bán hoặc giữ, tính toán phần thưởng và trả lại lần quan sát tiếp theo. Ngoài ra, chúng tôi có thể tính toán phần thưởng, bằng cách trừ đi giá trị ròng của bước trước và bước hiện tại.

3.2. Chuẩn hóa dữ liệu

Chúng tôi có dữ liệu của giá cả Bitcoin trên yahoofinace và tham khảo trên sàn giao dịch tiền điện tử Binance – sàn uy tín và lớn nhất thế giới về Crypto. Chúng tôi thu thập dữ liệu theo giờ, theo ngày, theo tuần và theo tháng. Những khung lớn giúp chúng tôi có thể theo dõi được chu kì và xu thế dài hạn của giá cả, những khung nhỏ như ngày và giờ giúp ta biết rõ những biến động ngắn hạn.

Date	Open	High	Low	Close	Adj Close	Volume
2016-01-01	430.721008	436.246002	427.515015	NaN	434.334015	36278900
2016-01-02	434.622009	436.062012	431.869995	0.670676	433.437988	30096600
2016-01-03	433.578003	433.743011	424.705994	0.662162	430.010986	39633800
2016-01-04	430.061005	434.516998	429.084015	0.684015	433.091003	38477500
2016-01-05	433.069000	434.182007	429.675995	0.669879	431.959991	34522600
...	...	...	...	...	...	...
2022-07-08	21637.154297	22314.941406	21257.453125	0.679922	21731.117188	49899834488
2022-07-09	21716.828125	21877.138672	21445.957031	0.664373	21592.207031	29641127858
2022-07-10	21591.080078	21591.080078	20727.123047	0.623691	20860.449219	28688807249
2022-07-11	20856.353516	20856.353516	19924.539062	0.610473	19970.556641	24150249025
2022-07-12	19970.474609	20043.445312	19308.531250	0.625955	19323.914062	25810220018

2385 rows × 7 columns

Hình 2. Dữ liệu về giá cả của Bitcoin từ ngày 1-1-2016 đến 12-7-2022 bao gồm giá mở cửa, giá cao nhất, giá thấp nhất và khối lượng giao dịch

Hình 3. Biểu đồ giá cả Bitcoin theo khung 1 giờ (nguồn: https://www.binance.me/)

Hình 4. Biểu đồ giá cả Bitcoin theo khung 1 ngày (nguồn: https://www.binance.me/)

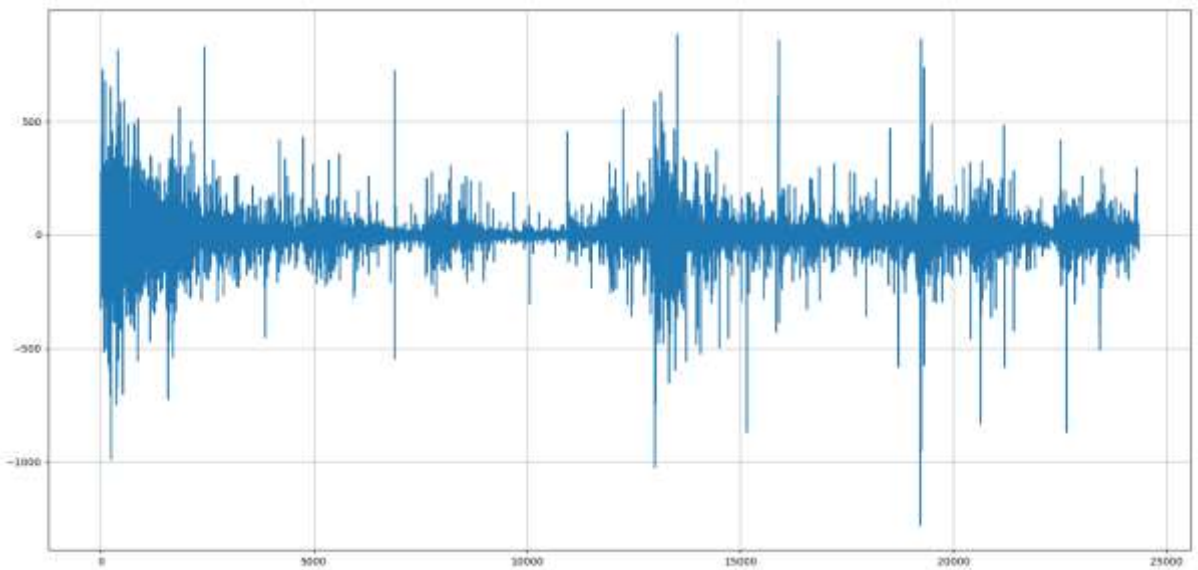
Hình 5. Biểu đồ giá cả Bitcoin theo khung 1 tuần (nguồn: https://www.binance.me/)

Hình 6. Biểu đồ giá cả Bitcoin theo khung 1 tháng (nguồn: https://www.binance.me/)

Biểu đồ giá cả từ khung tuần trở lên ta có thể thấy được tính chu kì của giá Bicoin cũng như xu thế và biến động dài hạn của nó. Ở các khung nhỏ hơn như khung ngày hay khung giờ đôi khi ta nhìn thấy giá tăng nhưng thực chất là đang giảm trong một chu kì lớn: đáy của ngày hôm nay có thể là đỉnh của ngày mai. Tuy nhiên nếu đầu tư ngắn hạn ta cần đến các khung thời gian ngắn hơn.

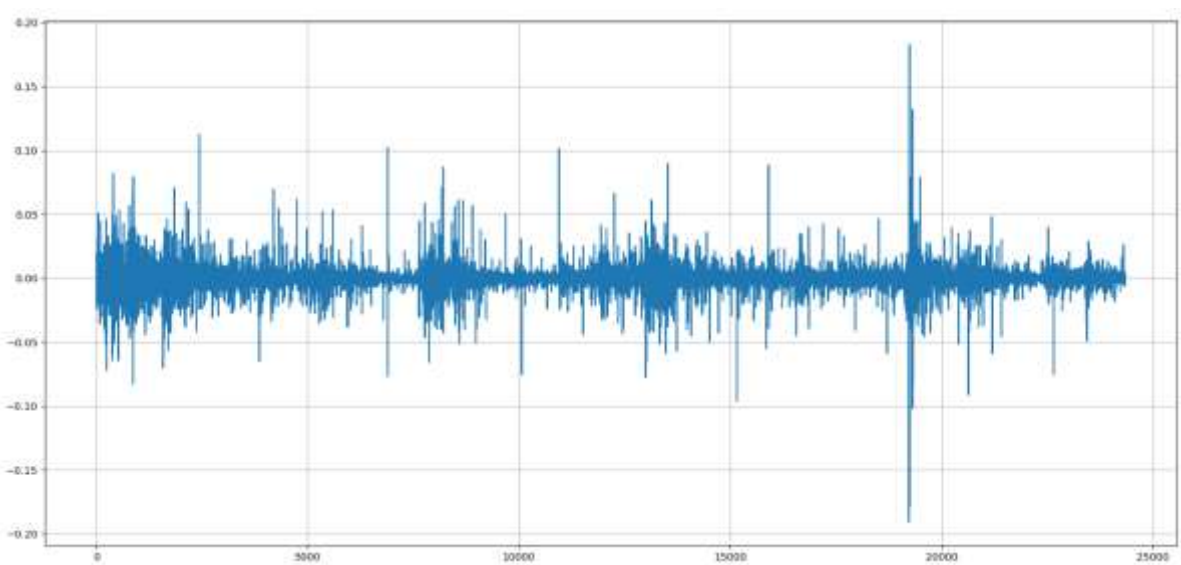
Chúng ta biết rằng, dữ liệu chuỗi thời gian không dừng, điều này có nghĩa là mô hình học máy khó có thể dự đoán xu hướng giảm nếu nó được học trên dữ liệu xu hướng tăng trong khi đào tạo.

Chúng ta có thể giải quyết vấn đề này bằng cách sử dụng các kỹ thuật biến đổi và sai phân để chuyển đổi dữ liệu của mình sang dạng phân phối chuẩn.



Hình 7. Sai phân của chuỗi giá đóng cửa theo khung ngày

Kết quả trông khá thú vị và có vẻ như xu hướng thị giác đã bị loại bỏ. Tuy nhiên, dữ liệu vẫn có tính thời vụ rõ ràng. Chúng ta có thể cố gắng loại bỏ điều đó bằng cách lấy logarit ở mỗi bước thời gian trước khi so sánh dữ liệu của chúng ta.



Hình 8. Logarit của chuỗi giá đóng cửa theo khung ngày

3.3. Chỉ báo kỹ thuật

Ta có 5 chỉ báo quan trọng: SMA, Bollinger Bands, Parabolic SAR, MACD và RSI. Thông thường, các chỉ báo kỹ thuật được sử dụng cho một số loại phân tích kỹ thuật, nhưng chúng tôi sẽ gọi đây là "Kỹ thuật tính năng" vì chúng tôi sẽ cố gắng chỉ trích xuất các chỉ báo tương quan ít nhất từ một lô, sau đó chúng tôi sẽ chuẩn hóa chúng với kỹ thuật và mọi thứ sẽ được cung cấp cho tác nhân học tăng cường của chúng tôi.

Để chọn các chỉ báo kỹ thuật mà chúng tôi sẽ sử dụng, chúng tôi sẽ so sánh mối tương quan của tất cả 42 chỉ báo kỹ thuật (tại thời điểm viết bài) trong thư viện **ta**. Cách đơn giản nhất là sử dụng thư viện **pandas** và **seaborn**[4] để tìm mối tương quan giữa từng chỉ báo cùng loại (xu hướng, biến động, khối lượng, động lượng, những chỉ báo khác). Sau đó, chúng tôi sẽ chỉ chọn các chỉ số ít tương quan nhất từ mỗi loại. Bằng cách này, chúng ta có thể nhận được nhiều lợi ích nhất có thể từ các chỉ báo kỹ thuật này mà không làm tăng quá nhiều nhiễu cho kích thước trạng thái của chúng ta.

3.3.1. Mối tương quan giữa các chỉ báo kĩ thuật

Một trong những cách nhanh nhất để nâng cao mô hình học máy là xác định và giảm bớt các tính năng của tập dữ liệu có tương quan cao. Các tính năng này tạo thêm nhiễu làm giảm độ chính xác cho mô hình của chúng tôi, do đó khiến việc đạt được kết quả mong muốn khó khăn hơn.

Khi hai đặc điểm độc lập có mối quan hệ chặt chẽ, chúng được coi là tương quan dương hoặc âm. Chúng tôi khuyến nghị rằng nên tránh các biến có tương quan cao khi phát triển mô hình vì chúng có thể làm sai lệch kết quả đầu ra. Nếu có hai biến độc lập đại diện cho cùng một sự kiện, nó có thể gây ra "nhiễu" hoặc không chính xác trong mô hình. Các mô hình chỉ dựa vào thông tin bên ngoài để tạo ra đầu ra hữu ích và có các biến cộng tuyến (tương quan) có thể làm tăng sự thay đổi ở ít nhất một trong các đầu ra hồi quy. Điều này gây khó khăn cho việc xác định biến nào thực sự ảnh hưởng đến biến phụ thuộc, gây khó khăn cho việc đánh giá mức độ hữu ích của mô hình.

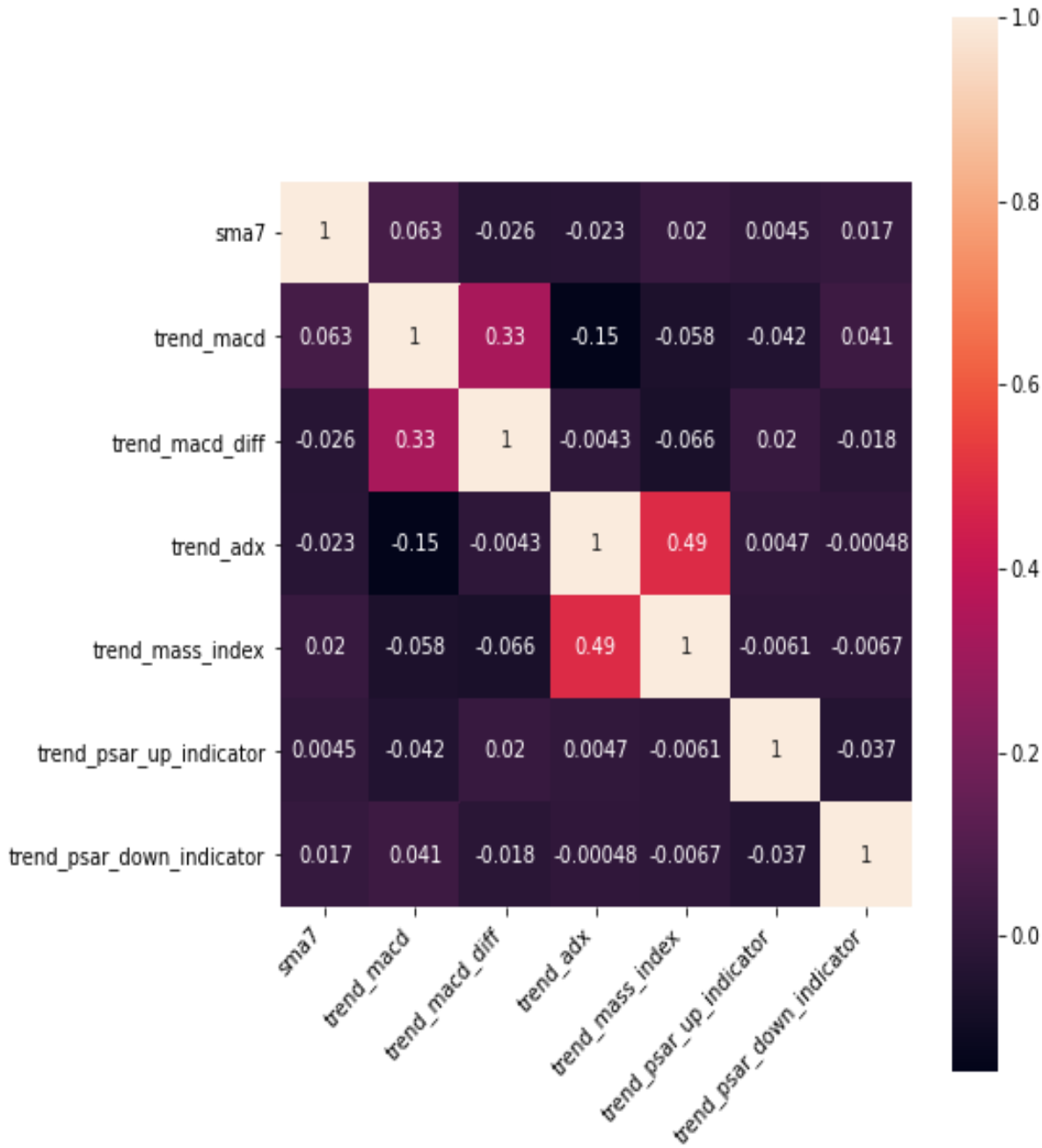
3.3.2. Chỉ báo xu hướng

Phần lớn nhất của toàn bộ các chỉ số trong thư viện **ta** là các chỉ báo xu hướng, tổng cộng có 14 chỉ số được đưa ra. Tất nhiên, các chỉ báo xu hướng cho chúng ta biết thị trường đang di chuyển theo hướng nào, nếu có một xu hướng nào đó. *(xem hình 9)*

3.3.3. Các chỉ số biến động

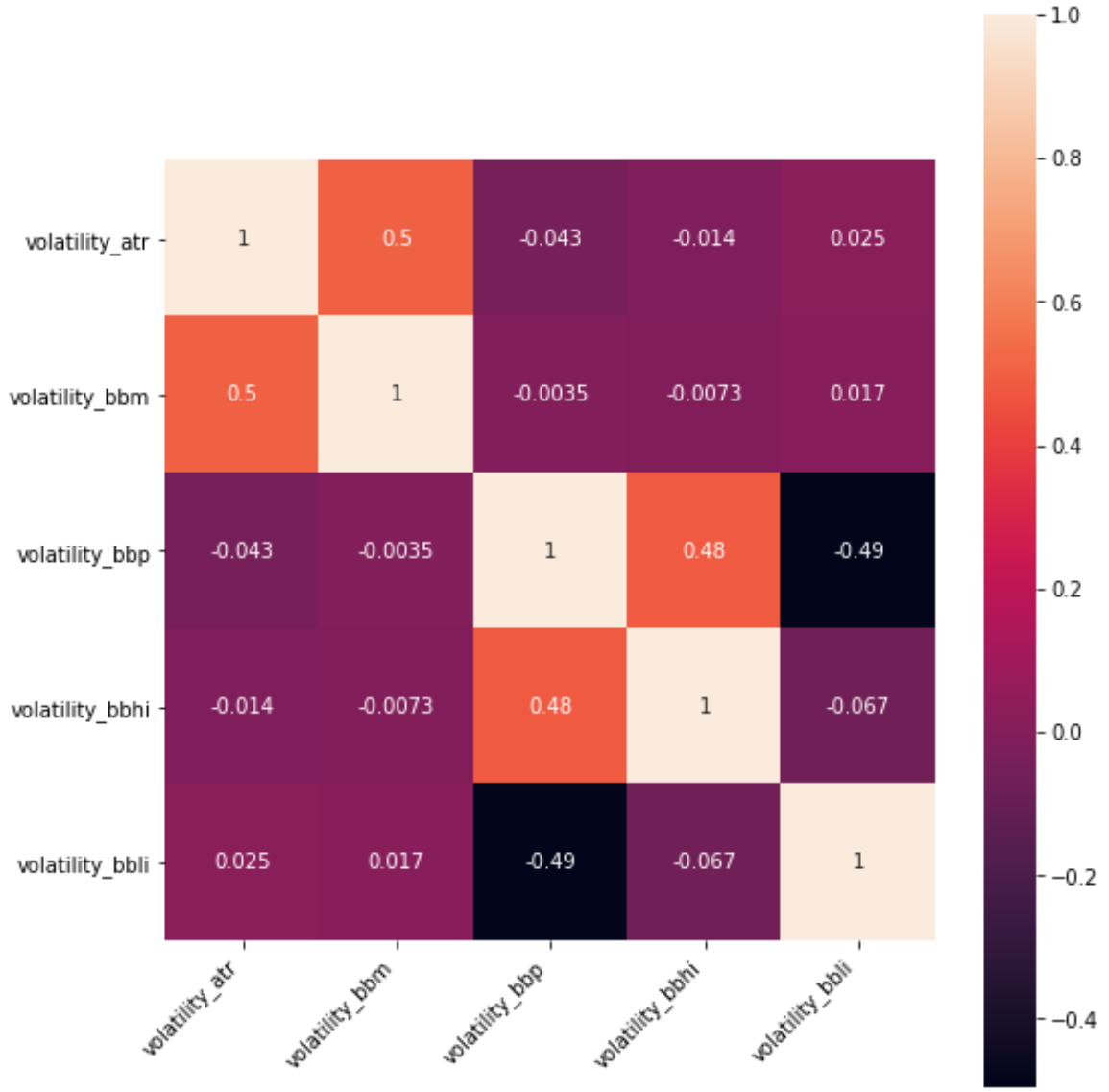
Nhóm chỉ số thứ hai là 5 chỉ số biến động. Đó là một dạng chỉ báo kỹ thuật đặc biệt, đo lường mức độ mà một tài sản đi lệch khỏi giá trị định hướng trung bình của nó. Điều này nghe có vẻ phức tạp nhưng nó khá đơn giản: Khi một tài sản có độ biến động cao, nó sẽ đi lệch xa so với hướng trung bình của nó. Ví dụ, một trận động đất có độ biến động cao so với điều kiện thời tiết bình thường. Rất giống với các chỉ báo xu hướng, tôi đã tạo một chức năng mà chúng tôi sẽ sử dụng để nhận Bản đồ nhiệt của các chỉ số biến động: *(xem hình 10)*

[4] *Thư viện dữ liệu trong ngôn ngữ lập trình Python.*



Hình 9. Tương quan giữa các chỉ báo xu hướng

Trước khi tính toán bản đồ nhiệt độ biến động, có 5 chỉ số, kết quả là chúng ta thấy số lượng các chỉ số giống nhau. Đây là những kết quả tuyệt vời, có nghĩa là tất cả các chỉ số của chúng tôi được tính toán bằng cách sử dụng các kỹ thuật khác nhau cung cấp cho chúng tôi các đặc điểm không tương quan. Điều này có nghĩa là không có chỉ báo biến động nào tương quan với nhau.

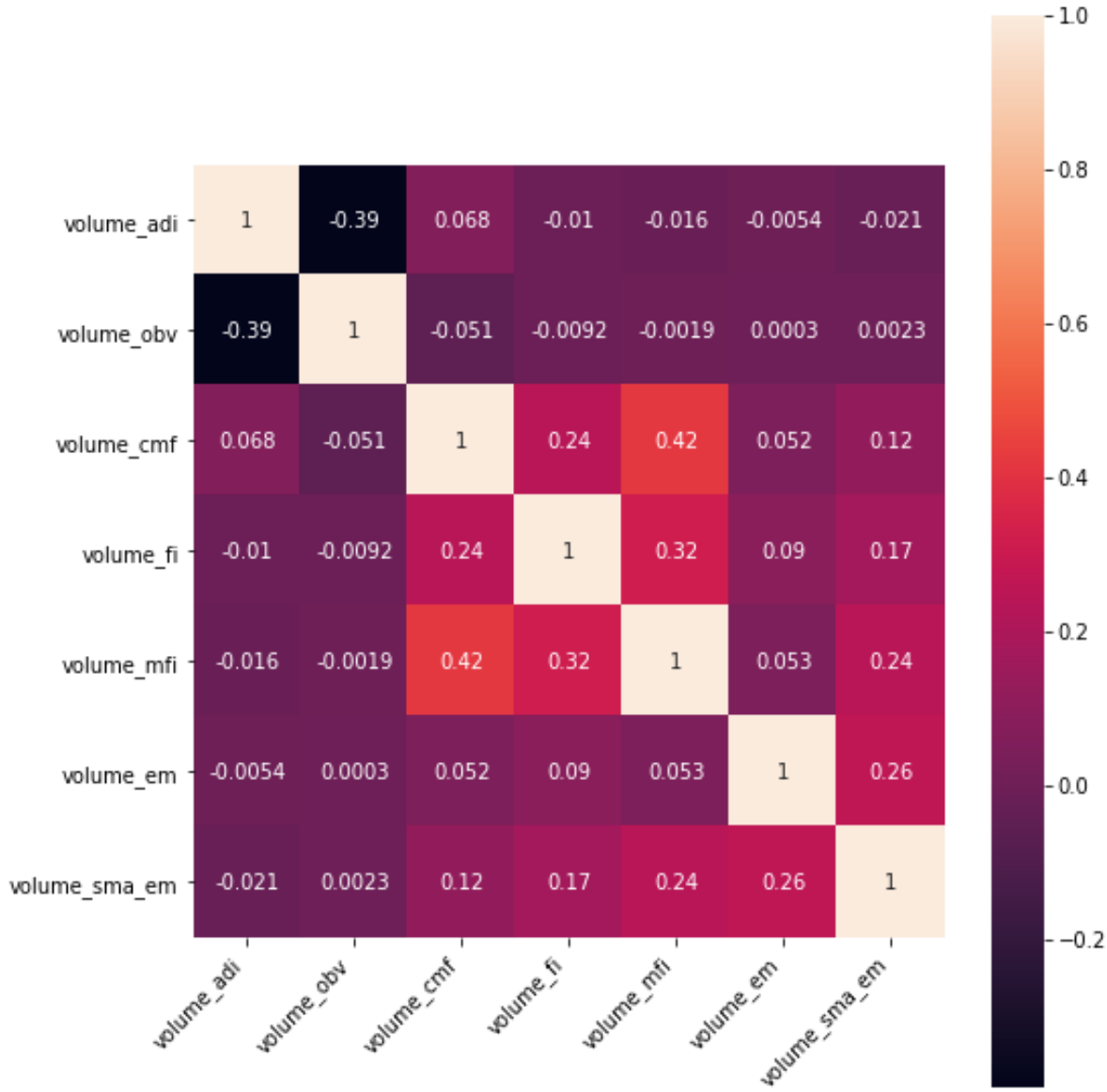


Hình 10. Tương quan giữa các chỉ báo biến động

3.3.4. Chỉ báo khối lượng giao dịch

Khối lượng cho chúng ta thấy số lượng cổ phiếu chứng khoán đã được giao dịch trong một khoảng thời gian nhất định. Chỉ báo khối lượng là các công thức toán học đơn giản được trình bày trực quan trong hầu hết các nền tảng giao dịch được sử dụng phổ biến.

Trước khi tính toán bản đồ nhiệt, còn lại 7 chỉ số không tương quan trên tổng số 9 chỉ số. Điều này có nghĩa là chỉ có 22% tất cả các chỉ báo khối lượng là tương quan. Đó là một kết quả ấn tượng, ngay cả bây giờ chúng ta có thể thấy rằng chúng ta có thể giảm ngưỡng nhưng số lượng các chỉ báo sẽ giữ nguyên vì chúng không có mối liên hệ chặt chẽ với nhau.

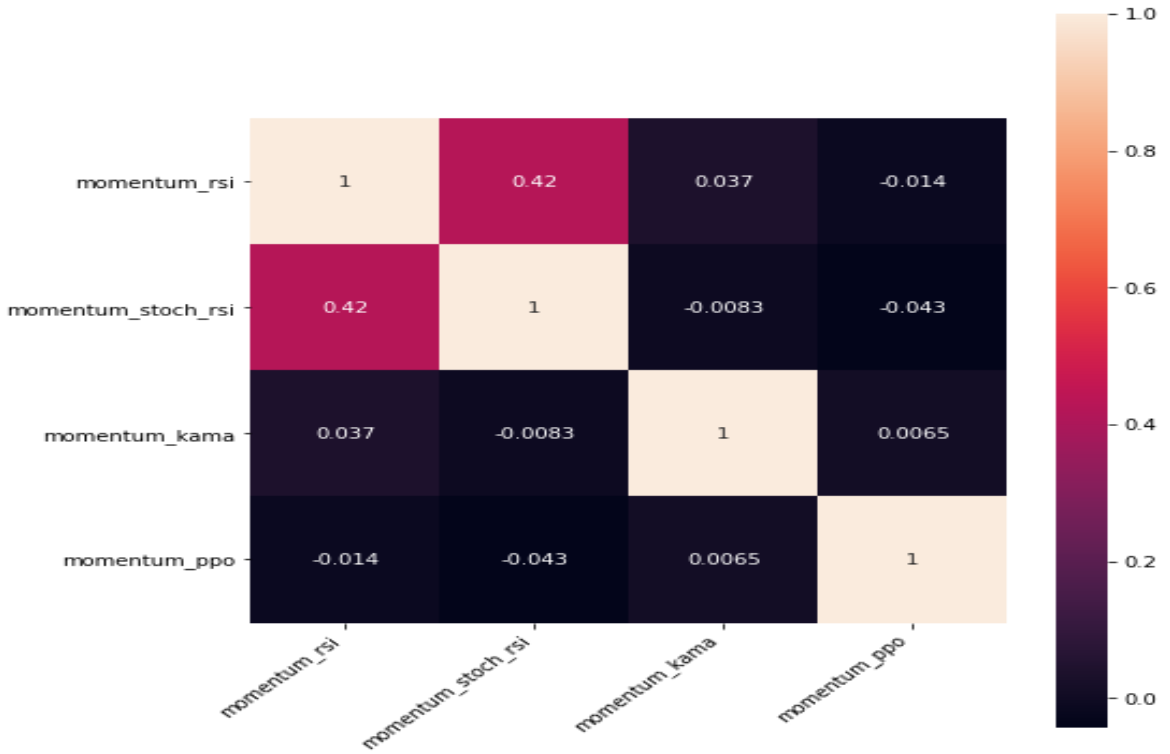


Hình 11. Tương quan giữa các chỉ báo khối lượng giao dịch

3.3.5. Chỉ báo động lượng

Các chỉ báo động lượng cho thấy sự chuyển động của giá theo thời gian và mức độ mạnh mẽ của những chuyển động đó, bất kể hướng giá đang di chuyển. Người ta nói rằng các chỉ báo động lượng cũng đặc biệt hữu ích vì chúng giúp các nhà giao dịch và nhà phân tích nhận ra các điểm mà thị trường có thể đảo chiều. Chúng tôi thêm các chỉ số này với chức năng sau: *(xem hình 12)*

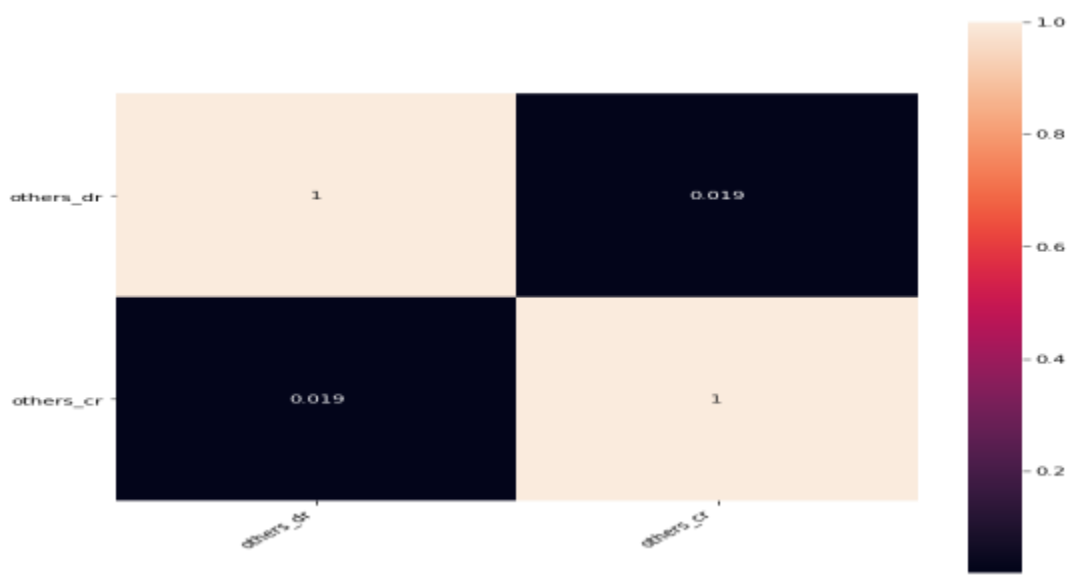
Trước khi tính toán bản đồ nhiệt, có 11 chỉ số, sau khi bỏ những chỉ số tương quan, chỉ còn lại 4 trong số đó. Điều này có nghĩa là 64% của tất cả các chỉ báo động lượng có tương quan, điều này có nghĩa là các chỉ báo động lượng có tương quan nhất.



Hình 12. Tương quan giữa các chỉ báo động lượng

3.3.6. Các chỉ số khác

Hàng loạt chỉ báo cuối cùng là Lợi nhuận hàng ngày (DR), Lợi nhuận nhật ký hàng ngày (DLR), Lợi nhuận tích lũy (CR).



Hình 13. Tương quan giữa các chỉ báo khác

Không có gì nhiều để nói về điều này, vì chỉ có 3 chỉ báo, một trong số đó có mối tương quan cao.

3.4. Đào tạo và kiểm tra

Chúng tôi đã đào tạo mô hình cho 400 nghìn bước đào tạo, mất khoảng 55 giờ. Bảng sau đây hiển thị phần thưởng của 1000 đầu tư trong dữ liệu kiểm tra:

Kinh nghiệm được rút ra từ thử nghiệm này: Nhiều chỉ báo kỹ thuật hơn không có nghĩa là chúng luôn mang lại cho chúng tôi hiệu suất tốt hơn của Bot giao dịch RL Bitcoin, điều này là do hầu hết các chỉ báo đều bị trễ và chúng gây thêm quá nhiều nhiễu cho quá trình đào tạo trên tập dữ liệu.

```
net_worth: 988 1081.948986726775 163
net_worth: 989 1080.2091108343238 155
net_worth: 990 1055.0594999440696 127
net_worth: 991 1098.573435680346 141
net_worth: 992 1040.6931840962757 133
net_worth: 993 1005.2077122011905 181
net_worth: 994 984.1921896204802 145
net_worth: 995 1040.7180469843356 140
net_worth: 996 1109.7548199859957 169
net_worth: 997 985.424493880857 156
net_worth: 998 1014.2064386018625 149
net_worth: 999 1036.363387283938 163
average 1000 episodes agent net_worth: 1041.1829741660276
```

4. Kết luận

Trong bài báo này chúng tôi nghiên cứu sử dụng thuật toán học tăng cường – một thuật toán đặc thù trong trí tuệ nhân tạo để huấn luyện một robot (tác nhân) đầu tư. Tác nhân sẽ học hỏi các chỉ báo và dữ liệu trong quá khứ để đưa ra các quyết định tối ưu (cực đại hóa lợi nhuận). Mỗi quyết định của tác nhân là việc xác định một thời điểm dừng để cực đại lợi nhuận. Dữ liệu chúng tôi thu được ở nhiều khung thời gian khác nhau và chúng tôi sử dụng khung giờ để huấn luyện và đầu tư ngắn hạn. Các kết quả thu được khá khả quan nhưng do thị trường mang tính biến động rất cao, việc thu được lợi nhuận trong thời kì này không có nghĩa là sẽ có lợi nhuận trong các thời kì khác.

Tài liệu tham khảo

[1] Wiering, M., Van Otterlo, M. (2012). Reinforcement Learning. Springer Berlin Heidelberg.

[2]. Fan J, Wang Z, Xie Y, Yang Z (2020) A theoretical analysis of deep Q-learning. In: Learning for Dynamics and Control. PMLR, pp 486–489

[3] Iftikhar Ahmad, Muhammad Ovais Ahmad, Mohammed A Alqarni, Abdulwahab Ali Almazroi, and Muhammad Imran Khan Khalil, Using algorithmic trading to analyze short term profitability of bitcoin**,** PeerJ Computer Science ***7*** (2021),e337.

[4] Dimitri P. Bertsekas. Dynamic Programming and Optimal Control. Volume 2. 4th Edition. (2012).

[5] Princeton, N.J., 1957. § D.P. Bertsekas and J. Tsitsiklis. Neuro-Dynamic Programming. Athena Scientific, [6] Belmont, MA, 1996. § W. Fleming and R. Rishel. Deterministic and stochastic optimal control.

[7] Applications of Mathematics, 1, Springer-Verlag, Berlin New York, 1975. § R. A. Howard. Dynamic

[8] Programming and Markov Processes. MIT Press, Cambridge, MA, 1960. § M.L. Puterman. Markov

[9] Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, Etats-Unis, 1994.