Journal of Applied And Basic Machine Intelligence Research (ABMIR)

پژوهش‌های نظری و کاربردی هوش ماشینی

سال دوم، شماره دوم، پاییز و زمستان ۱۴۰۳، صفحات ۶۵ – ۴۹

استفاده از الگوریتم یادگیری عمیق کیو جهت طراحی عاملی خودمختار برای معامله در بازار رمزارزها با تمرکز بر رفتار معامله‌گران

سید مهرداد اسلامی[۱]، مهدی آقاصرام[۲*]، محمدعلی زارع چاهوکی[۲]

[۱] دانشجوی دکتری، دانشگاه یزد، دانشکده مهندسی کامپیوتر، یزد، ایران

[۲] دانشیار، دانشگاه یزد، دانشکده مهندسی کامپیوتر، یزد، ایران

مقاله پژوهشی

تاریخ دریافت:
۱۴۰۳/۰۷/۰۸

تاریخ پذیرش:
۱۴۰۳/۱۰/۱۲

کلیدواژه‌ها:
عامل معامله‌گر هوشمند، رفتار معامله‌گران، بازار رمزارزها، یادگیری تقویتی عمیق، الگوریتم DQN

نویسنده‌مسئول:
mehdi.sarram@yazd.ac.ir

چکیده

بازار رمزارزها، محیطی پیچیده، غیرقطعی و همراه با نوسان‌های زیادی است. ایجاد استراتژی معاملاتی در این بازار بسیار چالش‌برانگیز است. در این مقاله تأثیر رفتار معامله‌گران (معاملاتی که انجام می‌دهند) بر تغییر شرایط بازار بررسی‌شده است. عوامل بسیاری در تغییر شرایط بازار تأثیر دارند، اما درنهایت این تأثیرات، از طریق رفتار معامله‌گران به فعلیت می‌رسد. در این مقاله عاملی خودمختار جهت انجام معامله در بازار رمزارز طراحی‌شده است. عاملی که تنها با بررسی معاملات انجام‌شده تصمیم می‌گیرد. طراحی عامل مبتنی بر الگوریتم DDQN از یادگیری تقویتی است. برای آموزش عامل تمام معامله‌های انجام‌شده در صرافی رمزارز HitBTC در طول نزدیک به ۳ ماه برای ۳ جفت رمزارز، گردآوری‌شده است. نتایج پیاده‌سازی نشان می‌دهد مدل همگرا شده و در شرایط محیطی با ریسک بالا پایداری خوبی از خود نشان داده است. درنتیجه معامله‌های انجام‌شده منبع مهمی برای تصمیم‌گیری است. ترکیب این روش با روش‌های پیش‌بینی قیمت می‌تواند رویکردی جدید در طراحی عامل‌های معامله‌گر باشد.

: 10.22034/ABMIR.2025.22203.1062

E-ISSN: 2821-2037

۱- مقدمه

استفاده از روش‌های ریاضیاتی و آماری در محاسبات کامپیوتری جهت ایجاد استراتژی‌های معاملاتی در بازارهای مالی، معاملات الگوریتمی نامیده می‌شود [۱]. هدف اصلی در معاملات الگوریتمی بیشینه‌سازی سود بلندمدت هم‌زمان با نگهداری ریسک معامله در محدوده‌ای قابل‌قبول است [۲]. طراحی استراتژی‌های معاملاتی در بازارهای مالی با استفاده از یادگیری تقویتی عمیق از موضوعات موردتوجه در چند سال اخیر بوده است و محققان و فعالین بازار به دنبال آموزش عاملی خودمختار هستند که بتواند سود بیشتری به دست آورد [۱-۱۳].

معاملات الگوریتمی به دلیل نداشتن احساسات، استفاده از توان محاسباتی بالا و همچنین واکنش سریع به تغییرات بازار موردتوجه بسیاری از معامله‌گران قرارگرفته است به‌گونه‌ای که بسیاری از معامله‌گران به استفاده از معاملات الگوریتمی سوق پیدا کرده‌اند [۸].

معاملات الگوریتمی مسئله تصمیم‌گیری متوالی در محیطی غیرقطعی است. در بازارهای مالی یک معامله‌گر انسانی شرایط بازار را رصد می‌کند و از تکنیک‌های مختلف مانند تحلیل تکنیکال، تحلیل بنیادی و یا امواج الیوت برای تحلیل بازار استفاده کرده و توالی از تصمیم‌ها را در زمانی مناسب اخذ و باهدف افزایش سود، معاملاتی را در بازار انجام می‌دهد، لذا یادگیری تقویتی روشی مناسب جهت طراحی معاملات الگوریتمی است [۹].

هدف عامل معامله‌گر، افزایش سود بلندمدت و کاهش ریسک معامله است [۱۰]. برای این منظور باید به سیاستی دست پیدا کند که کارائی عامل معامله‌گر بهینه شود. کارائی می‌تواند به‌صورت سود کسب شده بعد از هر معامله، میزان موجودی سبد سهام و یا ریسک معامله تعریف گردد [۱۱]. عامل باید بتواند توالی از تصمیم‌ها را در زمان‌هایی مناسب اخذ کند که منجر به بهینگی کارایی شود [۱۲، ۱۳].

در یادگیری تقویتی، عامل می‌تواند به‌طور خودکار مهارت‌های رفتاری، نزدیک به رفتار بهینه به دست آورد. رفتار عامل بیانگر واکنش عامل در هر حالت از محیط است و این واکنش با انجام عمل عینیت پیدا می‌کند. عامل، حالت محیط را مشاهده و سپس با استفاده از تابع سیاست، تصمیمی را اخذ و عملی را روی محیط انجام می‌دهد. محیط در پاسخ تغییر حالت داده و پاداشی را برمی‌گرداند. پاداش، میزان مفیدبودن هر عمل در هر حالت از محیط را مشخص می‌کند. سیاست، نگاشتی از فضای حالت به فضای عمل است و عامل با سعی و خطا یاد می‌گیرد که سیاست بهینه را به دست آورد و در شرایط مختلف محیطی به‌صورت خودمختار عمل بهینه را انجام دهد. عامل در طول فرایند یادگیری تلاش می‌کند احتمال رخداد عملی که پاداش بالاتری دریافت کرده است را افزایش دهد [۱۴].

تحلیل‌های بسیاری در رابطه با قابل پیش‌بینی بودن قیمت در بازارهای مالی انجام شده است. یکی از نظریات مهم در این زمینه، نظریه بازار کارآمد فاما است [۱۵]. بازار کارآمد، بازاری است که در آن تمام معامله‌گران به‌کل اخبار و اطلاعات دسترسی دارند. طبق نظریه بازار کارآمد فاما، قیمت‌های بازار به‌طور کامل همه اطلاعات و اخبار موجود را منعکس می‌کنند. و ازآنجاکه اخبار و اطلاعات آینده قابل‌پیش‌بینی نیست، تغییرات قیمت به‌صورت تصادفی و غیرقابل‌پیش‌بینی است. طبق این نظریه، قیمت، بازتابی از تمام اطلاعات موجود و ناشی از رفتار معامله‌گران است؛ بنابراین تمرکز بر معامله‌های انجام شده و رفتار معامله‌گران به‌جای داده‌های قیمتی و تکنیکال، می‌تواند روشی مناسب برای طراحی عامل‌های معامله‌گر در بازارهای مالی باشد.

در این مقاله عاملی خودمختار برای انجام معامله در بازار رمزارزها مبتنی بر الگوریتم شبکه کیوی عمیق دوگانه[1] یا به‌اختصار DDQN طراحی‌شده است [۱۶]. عامل معامله‌گر خودمختار، با تحلیل معاملات انجام شده، یاد می‌گیرد که چگونه در بازار، معامله انجام دهد. در بسیاری از تحقیقات انجام شده درزمینه طراحی معاملات الگوریتمی با استفاده از مدل‌های عمیق [۱۹-۱۷] و مدل‌های یادگیری تقویتی عمیق[۱۳-۱]، ورودی مدل شامل سری‌های زمانی مالی (ازجمله قیمت پایانی)، مقادیر ابزارهای تحلیل تکنیکال و میزان موجودی سبد سهام هستند. در این مدل‌ها، دادگان ورودی

[1] Double Deep Q Network

به بازه‌های زمانی ثابتی تقسیم می‌شوند. درنتیجه مدل در بازه‌های زمانی مشخصی تصمیم را اخذ می‌کند. این بازه زمانی معمولاً روزانه در نظر گرفته شده است، لذا عامل در هر روز در زمانی مشخص عملی را انجام می‌دهد. محدودیت اصلی در این روش‌ها، عدم وابستگی مدل به شرایط بازار است. به عبارتی مستقل از شرایط بازار، عامل در بازه‌ای مشخص عملی را انجام می‌دهد. برخلاف این تحقیقات در این مقاله، زمان عمل عامل به بازه زمانی ثابتی وابسته نیست بلکه به تعداد معامله‌ها بستگی دارد. به‌عبارت‌دیگر زمان واکنش عامل وابسته به شرایط بازار متغیر است. زمانی که تعداد معاملات در بازار افزایش پیدا می‌کند، عامل سریع‌تر واکنش نشان می‌دهد و زمانی که تعداد معاملات در بازار کاهشی می‌شود، عامل دیرتر واکنش نشان می‌دهد. این ویژگی باعث می‌شود عامل در شرایط نوسانی سریع‌تر واکنش نشان دهد.

در این مقاله همچنین تغییری در تابع کیوی بهینه داده شده است. تابع کیو بهینه در واقع چشم‌اندازی است که عامل تلاش می‌کند به آن برسد [۱۵]. طبق تعریف، تابع کیوی بهینه، بالاترین پاداش تجمعی است که عامل می‌تواند در آینده به دست آورد. در این مقاله هر حالت محیط مبتنی بر معاملات انجام شده تعریف شده است؛ لذا می‌توان بالاترین پاداش ممکن در هر حالت را به دست آورد که از این مقدار برای تخمین تابع کیوی بهینه استفاده شده است.

نکته دیگر در روش پیشنهادی، عدم وابستگی عامل به جفت‌ارز مشخصی است. عامل با استفاده از معاملات انجام شده، عملی را انجام می‌دهد و پاداشی را به دست می‌آورد و از این پاداش برای یادگیری استفاده می‌کند. ازآنجاکه معامله و پاداش به‌دست‌آمده نرمال شده است، امکان آموزش عامل روی جفت‌ارزهای متفاوت فراهم شده است.

نوآوری‌هایی که در این مقاله ارائه شده است عبارت‌اند از: ۱- آموزش عامل مبتنی بر معاملات انجام شده، ۲- محاسبه بالاترین پاداش دریافتی در هر حالت از روی معاملات انجام شده و استفاده از آن در تخمین تابع کیوی بهینه، ۳- بازه‌های عمل عامل بر خلاف روش رایج مبتنی بر تعداد معاملات است و این منجر به واکنش سریع‌تر عامل در شرایط نوسانی شده است و ٤- عامل به جفت‌ارز مشخصی وابسته نیست.

ساختار مقاله در ادامه به این صورت است، در بخش ۲ مروری بر تحقیق‌های گذشته انجام شده است و چالش‌های موجود درزمینه معاملات الگوریتمی بررسی شده‌اند. بخش۳ به تعاریف پایه در یادگیری تقویتی و الگوریتم DQN پرداخته است. در بخش ٤ روش پیشنهادی معرفی گردیده است. نتایج در بخش ۵ تحلیل شده‌اند. درنهایت بخش ٦ نتیجه‌گیری و کارهای آتی را بیان می‌کند.

۲- مروری بر تحقیقات و بررسی چالش‌ها

ایجاد مدلی برای پیش‌بینی روند بازار، قدمتی به‌اندازه خود بازار دارد. مدل‌های ریاضیاتی و آماری بسیاری جهت پیش‌بینی سری‌های زمانی معرفی شده‌اند. در ادامه روش‌های یادگیری ماشین مانند بردار پشتیبان، برای پیش‌بینی قیمت در بازارهای مالی مورداستفاده قرار گرفتند. این تحقیقات با ظهور یادگیری عمیق وارد دوره جدیدی شد و نتایج بهتری نسبت به روش‌های کلاسیک به دست آمد.

در [۱۷]، از شبکه عمیق کانولوشن CNN[1]، برای انتخاب عمل مناسب استفاده شده است. دادگان مورداستفاده شامل سری زمانی قیمت در بازه روزانه به همراه ۱۵ اندیکاتور تحلیل تکنیکال هست. ورودی مدل تنسوری دوبعدی به‌اندازه ۱۵×۱۵ است که شامل مقادیر ۱۵ اندیکاتور برای ۱۵ روز است. خروجی مدل شامل سه کلاس خرید، نگهداری و فروش است.

مدل‌های توالی‌دار عمیق مانند شبکه عصبی بازگشتی RNN[2] و مدل‌های عمیق GRU[3] و LSTM[4] در بسیاری از تحقیقات مربوط به معاملات الگوریتمی مورد استفاده و بررسی قرار گرفته است. در [۱۸]مدل عمیق LSTM برای پیش‌بینی روند حرکتی بازار مورد استفاده قرار گرفته است. ورودی مدل سری‌زمانی قیمت در بازه روزانه و خروجی مدل دو کلاس صعودی و نزولی هست. در مقاله[۱۹]از مدل مولد عمیقِ شبکه متخاصم مولد GAN[5] برای

[1] Convolutional Neural Network
[2] Recurent Neural Network
[3] Gated Recurrent Unit
[4] Long Short Term Memory
[5] Generative Adversarial Network

پیش‌بینی قیمت در بازارهای مالی استفاده شده است. در این کار از سری زمانی قیمت در بازه روزانه استفاده شده است.

به‌موازات این تحقیق‌ها، سوی دیگری از تحقیق‌ها در یادگیری تقویتی در حال انجام بود. یادگیری تقویتی ترکیبی از سه‌شاخه علمی است. **شاخه اول** در علم روان‌شناسی گسترش پیدا کرد. مطالعه‌هایی که روی آموزش حیوان‌ها با روش سعی و خطا تمرکز دارد. **شاخه دوم** به بهینه‌سازی در ریاضی مربوط می‌شود. روش‌های بهینه‌سازی با استفاده از تابع ارزش و برنامه‌نویسی پویا در شاخه ریاضی گسترش پیدا کردند. **شاخه سوم**، هوش مصنوعی است. با ترکیب دو روش فوق و معرفی روش تفاضل زمانی، یادگیری تقویتی شکل گرفت [۱۴].

از اولین تحقیق‌ها درزمینه استفاده از یادگیری تقویتی در معاملات الگوریتمی می‌توان به کار مؤدی [۲۰] اشاره کرد. در این کار، عاملی خودمختار طراحی‌شده است که با تحلیل قیمت‌های گذشته در بازه‌های زمانی ثابت، تصمیم معاملاتی اخذ می‌کند. در این کار فضای حالت با استفاده از قیمت پایانی و موجودی سبد سهام و فضای عمل به‌صورت سه‌گانه خرید، فروش و نگهداری تعریف شده است. برای تخمین مقدار تابع کیو از رابطه بازگشتی خطی استفاده شده است.

استفاده از ظرفیت یادگیری عمیق به همراه توانایی کنترل تصمیم در یادگیری تقویتی منجر به ایجاد شاخه یادگیری تقویتی عمیق شده است. این شاخه در طراحی معاملات الگوریتمی برای ایجاد عامل معامله‌گر خودمختار، بسیار موردتوجه قرار گرفته است. بسیاری از تحقیقات در این زمینه از الگوریتم‌های یادگیری تقویتی مبتنی بر تابع ارزش به همراه مدل‌های عمیق برای تخمین تابع کیو استفاده کرده‌اند.

از اولین و مهم‌ترین تحقیق‌ها در استفاده از شبکه عصبی عمیق در یادگیری تقویتی می‌توان به DQN[1] اشاره کرد [۲۱]. این مدل در سال ۲۰۱۳ جهت طراحی عاملی خودمختار برای انجام بازی‌های آتاری ارائه شد و عامل توانست بازی‌های آتاری را با دقتی نزدیک به انسان انجام دهد. دو تکنیک مهم در این مقاله معرفی و در تمام تحقیقات آتی مورد استفاده مجدد قرار گرفته است. اولین تکنیک، تعریف بافر تکرار[2] است که حافظه‌ای برای ذخیره تجربیات[3] تعامل عامل با محیط هست. هدف از تعریف بافرتکرار ایجاد استقلال بین تجربیات عامل و بهبود فرایند یادگیری است. تکنیک دوم استفاده از شبکه‌ای مجزا از شبکه کیو برای تخمین مقدار تابع کیوی بهینه هست.

در سال ۲۰۱۷ با استفاده از مدل DQN، عاملی خودمختار، توانست بدون دانش قبلی، بازی گو را به‌صورت حرفه‌ای یاد بگیرد و قهرمان این بازی را شکست دهد [۲۲]. بعدازاین نتایج، الگوریتم‌های یادگیری تقویتی عمیق به‌طور گسترده در زمینه‌های دیگر ازجمله در طراحی عامل معامله‌گر خودمختار مورداستفاده قرار گرفتند.

استفاده از روش‌های یادگیری تقویتی عمیق در محیط‌های پیچیده و غیرقطعی چالش‌برانگیز است. برای رسیدن به مدلی همگرا، حجم زیادی داده، پردازش بالا و تنظیم پارامترهای زیادی موردنیاز است. از طرفی دیگر، مدل‌های یادگیری تقویتی عمیق مبتنی بر الگوریتم یادگیری کیو بسیار ناپایدار هستند و تغییر کوچکی در تخمین‌گر تابع کیو می‌تواند تأثیر بسیاری در عملکرد عامل داشته باشد. رسیدن به مدلی همگرا در این نوع مسائل کار پیچیده‌ای به‌حساب می‌آید. برای بهبود همگرایی در الگوریتم یادگیری کیو، ایده استفاده از دو تخمین‌گر مجزا در [۲۳]. پیشنهاد شده است. در این کار تخمین‌گر اول، تابع کیو و تخمین‌گر دوم مقدار تابع کیوی بهینه را تخمین می‌زند.

در [۱۶] ایده استفاده از دو تخمین‌گر برای مدل DQN ارائه شده است. در این مدل از دو شبکه پرسپترون مجزا برای تخمین تابع کیو و تابع کیوی بهینه استفاده شده است. این روش، تحت عنوان شبکه عمیق کیو دوتایی یا DDQN شناخته می‌شود.

از نخستین مقاله‌ها، درزمینه استفاده از یادگیری تقویتی عمیق در معاملات الگوریتمی می‌توان به مقاله [۲۴]. اشاره کرد که در آن عامل معامله‌گری مبتنی بر مدل DQN و شبکه RNN طراحی‌شده است. فضای حالت با استفاده از داده‌های قیمتی در بازه روزانه و فضای عمل به‌صورت گسسته شامل سه عمل خرید، فروش و نگهداری تعریف شده است.

[1] Deep Q Network
[2] Replay Buffer
[3] Trajectory

در سال ۲۰۲۱ روش TDQN[1] در مقاله [۱] ارائه شد. در این مقاله از الگوریتم DQN جهت آموزش عامل معامله‌گر استفاده شده است. فضای حالت به‌صورت گسسته و نامحدود تعریف شده است. هر حالت محیط با استفاده از داده‌های قیمتی بازار و مقادیر اندیکاتورهای تحلیل تکنیکال به همراه شرایط داخلی عامل مانند تعداد سهام و میزان نقدینگی، تعریف گردیده است. فضای عمل یک مجموعه گسسته از اعداد صحیح است که حجم معامله را نشان می‌دهد. اعداد مثبت برای عمل خرید، اعداد منفی برای عمل فروش و عدد صفر بیانگر انجام‌ندادن معامله است. پاداش هر عمل باتوجه‌به میزان تغییر در موجودی سبد سهام محاسبه می‌شود.

در [۲] با ترکیب دو مدل عمیق شامل مدل‌توجه و BiLSTM[2]، الگوریتم DQN پیاده‌سازی شده است. در این کار هر حالت محیط ابتدا به شبکه BiLSTM وارد شده و خروجی آن برای تخمین تابع کیو وارد یک شبکه ترنسفورمر شده است.

در مقاله [۷] هر حالت محیط ترکیبی از داده‌های قیمتی و خروجی سه ابزار تکنیکال هست. مدل ارائه شده مبتنی بر اینکدر–دیکدر است. قسمت اینکدر حالت محیط را دریافت کرده و در اختیار دیکدر قرار می‌دهد. شبکه اینکدر به شیوه‌های مختلف شامل شبکه عصبی پرسپترون، شبکه GRU و ترکیب CNN با GRU پیاده‌سازی شده است. شبکه دیکدر مبتنی بر DQN است.

مقاله [۸] چند عامل خودمختار، مستقل از یکدیگر و با معماری یکسان را ارائه داده است. تصمیم نهایی تلفیقی از تصمیم‌های دیگر عامل‌ها است. مدل ارائه شده در این کار دارای سه سطح است. در سطح اول داده‌های سری زمانی مالی با روش میدان زاویه‌ای گرامیان به تصویر تبدیل می‌شوند. سطح دوم به‌وسیله یادگیری عمیق و با شبکه CNN پیاده‌سازی شده است و در سطح سوم تصمیم نهایی با تلفیقی از تصمیم‌ها اخذ می‌شود.

در مقاله [۱۰]از مدل عمیق ترنسفورمر برای طراحی عامل معامله‌گر استفاده شده استفاده شده است. در این مقاله فضای حالت مبتنی بر قیمت پایانی و در بازه روزانه و فضای عمل به‌صورت گسسته و با سه عمل تعریف شده است. از مدل DQN استفاده شده است که در آن شبکه کیو و هدف با استفاده از مدل ترنسفورمر طراحی‌شده است. تابع پاداش میزان سود هر عمل در هر حالت در نظر گرفته شده است. در مقاله[۱۱]از CNN برای تخمین تابع هدف استفاده کرده است. فضای حالت از ۱۰ اندیکاتور تحلیل تکنیکال تشکیل‌شده است.

مقاله [۱۲]، روش DQN را برای انجام معامله در بازارهای مالی مورد استفاده قرار داده است. در این مقاله فضای حالت به‌صورت سری‌زمانی قیمت در بازه روزانه و فضای عمل به‌صورت یک مجموعه پیوسته بین ۱- تا ۱ در نظر گرفته شده است. تعریف فضای حالت به شکل پیوسته این امکان را ایجاد می‌کند که عامل علاوه بر نوع عمل (خرید یا فروش) حجم معامله را نیز بتواند تعیین کند. در مقاله [۱۳]مدل DQN برای معامله در بازارهای آتی طراحی‌شده است.

استفاده از الگوریتم یادگیری کیو برای طراحی عامل معامله‌گر خودمختار در بازار سهام ایران نیز موردتوجه و بررسی قرار گرفته است. در [۲۶]معاملات زوجی را با استفاده از یادگیری کیو پیاده‌سازی می‌کند. معاملات زوجی، معاملاتی است که در آن به دنبال ایجاد سبدی از دو دارایی با گرفتن موقعیت خرید در دارایی ارزان‌تر و گرفتن موقعیت فروش در آن سهمی که گران‌تر شده است، اقدام می‌کند. در نهایت مقاله[۲۷]داده‌های قیمت روزانه بازار سهام ایران به همراه ۳ اندیکاتور تکنیکال، فضای حالت را تشکیل می‌دهند و از یادگیری کیو برای تخمین بهترین معامله در هر روز استفاده می‌کند.

۳- الگوریتم یادگیری کیو و DDQN

در یادگیری تقویتی فرایند تصمیم‌گیری عامل، با کمک فرایند مارکوف تعریف می‌شود [۱۴]در رابطه (۱) شرط خاصیت مارکوف آورده شده است.

$$Pr(S_{n+1}|S_1 \cdots S_n) = Pr(S_{n+1} \vee S_n) \qquad (1)$$

طبق رابطه (۱)، فرایندی تصادفی دارای خاصیت مارکوف است، اگر هر حالت فقط به حالت قبل از خودش وابسته باشد.

[1] Trading Deep Q Network

[2] Bidirectional LSTM

اگر به فرایند مارکوف، عمل و پاداش دریافتی از آن عمل اضافه شود، فرایند تصمیم مارکوف به وجود می‌آید. رابطه (۲) فرایند تصمیم مارکوف را نشان می‌دهد.

$$Pr(S_{n+1},r_n|S_1 \cdots S_n,a_1 \cdots a_n) = Pr(S_{n+1},r_n|S_n,a_n) \quad (۲)$$

طبق رابطه (۲) احتمال‌رفتن به حالت S_{n+1} و دریافت پاداش r_n فقط به حالت و عمل قبل بستگی دارد.

چندگانه $M = (S, A, P, R, \gamma)$ فرایند تصمیم مارکوف است و در آن، S مجموعه تمام حالت‌های محیط است و فضای حالت نامیده می‌شود. A مجموعه تمام عمل‌هایی است که عامل می‌تواند در محیط انجام دهد، به این مجموعه فضای عمل گویند. P تابع توزیع تغییر حالت محیط را مشخص می‌کند. R تابعی است از $R:S\times A\rightarrow R$ و پاداش است که عامل بعد از انجام هر عمل دریافت می‌کند. γ فاکتور تخفیف، عددی است در بازه [۰,۱] که به منظور همگرا شدن سری پاداش تجمعی استفاده می‌شود.

زیربنای یادگیری تقویتی، مبتنی بر پاداش دریافتی از محیط است. پاداش یک عدد اسکالر هست و مشخص می‌کند که عمل عامل در حالت t چقدر خوب بوده است.

مجموع پاداش‌های دریافتی در آینده یعنی از حالت t به بعد با G_t نشان داده می‌شود. رابطه (۳) آن را نشان می‌دهد. به G_t بازده و یا پاداش تجمعی گویند. پاداش تجمعی یک دید بلند مدت به عامل می‌دهد. عامل باید به سمت سیاستی حرکت کند تا پاداش تجمعی خود را بیشینه سازد.

$$G_t = r_t + \gamma^1 r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^k r_{t+k} = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad (۳)$$

در رابطه (۳)، r_t، پاداش دریافتی از محیط بعد از انجام عمل a_t است.

تابع کیو (تابع ارزش حالت –عمل)، میانگین مجموع پاداش‌هایی که عامل با شروع از حالت s و با انجام عمل a می‌تواند در آینده به دست می‌آورد را برمی‌گرداند. به عبارتی این تابع مشخص می‌کند، عامل با انجام عمل a در حالت s انتظار دارد در آینده چه میزان پاداش به دست آورد. در رابطه (٤) تعریف تابع کیو آمده است.

$$Q(s_t,a_t) = \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,|s = s_t, a = a_t\right] \quad (٤)$$

با استفاده از رابطه بلمن می‌توان تابع کیو را به‌صورت بازگشتی مطابق رابطه (۵) بازنویسی کرد.

$$Q(s_t,a_t) = \mathrm{E}[r_t + \gamma Q(s_{t+1},a_{t+1})] \quad (۵)$$

در رابطه (۵)، r_t پاداش به‌دست‌آمده در مرحله t ام و $Q(s_{t+1},a_{t+1})$تابع کیو برای حالت و عمل مرحله بعد است.

در الگوریتم یادگیری کیو از رابطه (٦) برای بهبود و به‌روزرسانی تابع کیو استفاده می‌کند.

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha(r_t + \gamma max_a Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)) \quad (٦)$$

در رابطه (٦)، α نرخ یادگیری را مشخص می‌کند. خط دوم رابطه (٦) تخمین تابع کیوی را بیان می‌کند. همچنین $Q(s_{t+1},a_{t+1})$ مقدار تابع کیو برای حالت بعدی است. تصمیم عامل در الگوریتم یادگیری کیو مطابق رابطه (۷) به دست می‌آید. تصمیم عامل، عملی است که بالاترین مقدار تابع کیو را داشته باشد. البته در فرایند یادگیری برای اکتشاف بیشتر گاهی عملی به‌صورت تصادفی انتخاب می‌گردد.

$$\pi(s_t) = argmax_a Q(s_t,a) \quad (۷)$$

در رابطه (۷) $\pi(s_t)$، تابع سیاست عامل نامیده می‌شود. سیاست در هر حالت مشخص می‌کند کدام عمل انتخاب گردد. تصمیم عامل توسط این تابع اخذ می‌گردد. در الگوریتم DQN برای تخمین تابع کیو از شبکه عصبی عمیق استفاده می‌کند [۲۱]. علاوه بر این، برای استقلال نمونه‌های آموزشی، نمونه‌ها را در بافر تکرار قرار داده و برای هر مرحله آموزش، تعدادی از نمونه‌ها را به‌صورت تصادفی انتخاب می‌کند.

در مدل DQN از دو شبکه با معماری یکسان استفاده می‌شود. شبکه نخست تحت عنوان شبکه کیو، حالت محیط را به‌عنوان ورودی دریافت و مقدار تابع کیو را برای هر عمل برمی‌گرداند. شبکه دوم تحت عنوان شبکه هدف وظیفه تخمین تابع کیوی بهینه را برای حالت بعدی محیط دارد. این شبکه، حالت بعدی محیط را دریافت و مقدار تابع کیوی بهینه برای هر عمل را برمی‌گرداند. این دو شبکه دارای معماری یکسان هستند. شبکه هدف آموزش

نمی‌بیند؛ بلکه در بازه‌های زمانی مشخص مقدار پارامترهای شبکه سیاست در آن کپی می‌گردد به عبارتی مدل دارای دو شبکه ولی یک تخمین‌گر است.

در DDQN برای افزایش همگرایی مدل، از دو شبکه عمیق با معماری یکسان استفاده شده است؛ ولی برخلاف مدل DQN وزن‌های شبکه سیاست در شبکه هدف کپی نمی‌شوند. به عبارتی مدل DDQN از دو تخمین‌گر مجزا استفاده می‌کند و هر دو شبکه سیاست و هدف به‌صورت مجزا آموزش می‌بینند [۱۶]. شبکه اصلی مقدار Q(s, a) را تخمین می‌زند و شبکه هدف برای تخمین مقدار Q*(s', a') تعریف شده است.

٤- روش پیشنهادی

در این مقاله، عاملی خودمختار جهت انجام معامله در بازار رمزارزها طراحی‌شده است. عاملی خودمختار که تنها با استفاده از تحلیل معاملات انجام شده یا همان رفتار معامله‌گران بازار، تصمیم می‌گیرد چه عملی انجام دهد. معاملات انجام شده تنها مشاهده عامل از محیط است که هر معامله شامل چهار ویژگی است. زمان انجام معامله، قیمت، حجم معامله و سمت معامله که می‌تواند خرید یا فروش باشد.

در شکل (۱) معماری روش پیشنهادی آمده است. این معماری از دو لایه تشکیل شده است. لایه اول مربوط به گردآوری دادگان و لایه دوم اجزاء عامل معامله‌گر را نشان می‌دهد.

لایه گردآوری دادگان، به‌صورت لحظه‌ای معاملات انجام شده را دریافت و در اختیار لایه دوم می‌گذارد. لایه دوم معماری از دو ماژول عامل و محیط تشکیل شده است. ماژول عامل، شامل دو شبکه کانولوشن برای تخمین تابع کیو و هدف است و هر دو دارای ساختاری یکسان هستند. ماژول محیط از ۳ بخش اصلی تشکیل شده است. بخش مشاهده‌کننده[1] که حالت محیط را شکل می‌دهد. این بخش، معامله‌های انجام شده را از پایگاه داده، دریافت، نرمال و با ساختار مناسبی در اختیار عامل می‌گذارد. بخش عمل[2] که تصمیم عامل را روی محیط اجرا می‌کند و بخش پاداش[3] که پاداش عمل را محاسبه می‌کند کرده و به عامل برمی‌گرداند. فرایند تصمیم‌گیری در عامل به این صورت است که ابتدا با استفاده از بخش مشاهده‌کننده، حالت محیط را دریافت کرده و به‌عنوان ورودی وارد شبکه سیاست می‌شود و عملی که بالاترین مقدار کیو را داشته باشد به‌عنوان تصمیم عامل در اختیار بخش عمل محیط قرار می‌گیرد. این بخش تصمیم عامل را روی محیط اجرا می‌کند. بعد از تغییر حالت محیط و ورود به حالت جدید بخش پاداش مقدار پاداش عمل را برمی‌گرداند. چهارگانه (حالت فعلی، عمل عامل، پاداش عمل و حالت بعدی) به‌عنوان یک مسیر در بافر تکرار ذخیره شده و از آن برای آموزش عامل استفاده می‌شود.

٤-١ گردآوری دادگان

دادگان مورداستفاده در این مقاله شامل دو مجموعه‌دادگان است. مجموعه اول که آموزش عامل مبتنی بر آن انجام شده است، شامل تمام معاملات انجام شده در یک بازار رمزارز است. هر معامله شامل چهار ویژگی زمان، قیمت، مقدار و سمت معامله هست. مجموعه‌دادگان دوم شامل اطلاعات قیمتی (بالاترین قیمت، پایین‌ترین قیمت، قیمت شروع و قیمت پایان) هر جفت رمزارز است که در بازه ۳۰ دقیقه‌ای گردآوری‌شده است. از مجموعه‌دادگان قیمت برای محاسبه پاداش استفاده می‌شود. برای گردآوری دادگان معاملات انجام شده از صرافی آنلاین HitBTC استفاده شده است و تمام معاملات صورت‌گرفته در این صرافی در طول چهار ماه از تاریخ دهم مرداد لغایت دهم آذر ۱۴۰۳ گردآوری شده است. مطابق شکل (۱) از چارچوب نرم‌افزاری آپاچی کافکا برای این منظور استفاده شده است. آپاچی کافکا شامل تو بخش اصلی تولیدکننده و مصرف‌کننده هست. تولیدکننده با فراخوانی واسط نرم‌افزاری، معاملات انجام شده اخیر در این صرافی را دریافت و به سمت مصرف‌کننده ارسال می‌کند. مصرف‌کننده به‌صورت بلادرنگ آن دادگان را دریافت و در پایگاه داده سری زمانی influx ذخیره می‌کند. بخش مشاهده‌کننده ماژول محیط داده‌های موردنیاز را از پایگاه داده فراخوانی و هر حالت محیط را تشکیل می‌دهد.

[1] Observer
[2] Action
[3] Reward

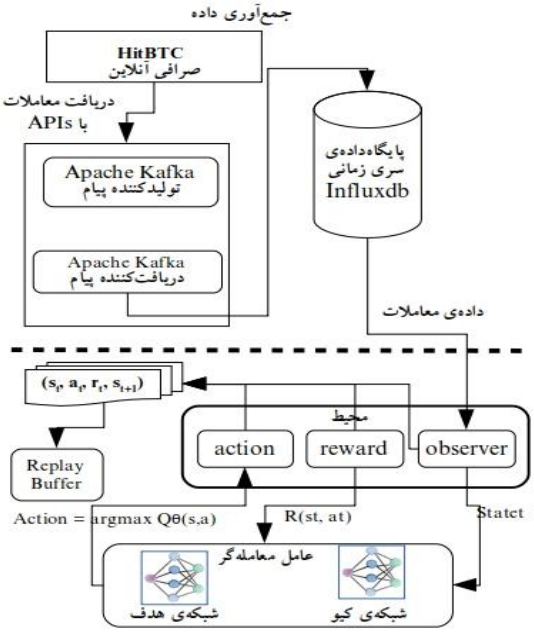
٤-٢ فضای حالت

در روش پیشنهادی، هر حالت از محیط، مجموعه‌ای با تعدادی مشخص از معاملات انجام شده است. هر معامله دارای چهار ویژگی است. در رابطه (۸) معامله i ام تعریف شده است. مجموعه‌ای از معاملات انجام شده حالت محیط را تشکیل می‌دهند. رابطه (۹) حالت t ام محیط را تعریف کرده است.

$$trade_i = \{time, price, quantity, side\} \quad (۸)$$

$$state_t = \{trade_i, \cdots, trade_{i+d\times d}\} \quad (۹)$$

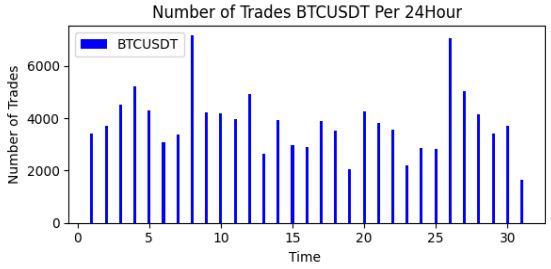
در رابطه (۸)، time، زمان انجام معامله، quantity، مقدار معامله انجام شده، price، قیمت ارز در انجام معامله و side، سمت معامله را مشخص می‌کند.



شکل (۱): معماری روش پیشنهادی

ساختار داده هر حالت محیط، یک تنسور مربعی به عرض d هست؛ لذا هر حالت دارای $d\times d$ معامله انجام شده است. هر معامله ٤ ویژگی دارد؛ بنابراین تنسور ورودی دارای ٤ کانال است. کانال اول مربوط به مقدار معامله انجام شده برای سمت فروش است. کانال دوم مربوط به مقدار معامله انجام شده برای سمت خرید است. کانال سوم قیمتی که معامله در آن انجام شده است را نشان می‌دهد. کانال چهارم زمان انجام معامله را ثبت می‌کند. ابعاد تنسور ورودی شبکه برابر با $4 \times d \times d$ هست. تمام مقادیر با روش نرمال‌سازی کمینه، بیشینه نرمال شده‌اند و در بازه [۱] قرار گرفته‌اند.

شکل (۲) نمودار تعداد معاملات انجام شده در هر روز برای جفت رمزارز بیت‌کوین–تتر برای بازه یک ماه را نشان می‌دهد.



شکل (۲): نمودار تعداد معاملات روزانه در بیت‌کوین – تتر از صرافی HitBTC

به‌صورت میانگین در هر روز ۲۷۰۰ معامله در این صرافی برای جفت‌ارز بیت‌کوین–تتر انجام می‌شود. اگر مقدار ۱٦= d تعیین گردید آنگاه تعداد معاملات هر حالت از محیط ۵۱۲ عدد است. در این صورت به‌صورت میانگین عامل در هر ۵ ساعت عملی را انجام می‌دهد. البته شایان‌ذکر است اگر بازار نوسان زیادی داشته باشد و تعداد معاملات افزایش پیدا می‌کند، عامل می‌تواند در بازهٔ زمانی کوتاه‌تری واکنش نشان دهد.

٤-٣ فضای عمل

عامل معامله‌گر باید تصمیم بگیرد که در چه زمانی، با چه قیمتی، به چه میزان و در کدام سمت (خرید یا فروش)، معامله را ثبت کند.

در روش پیشنهادی، زمان انجام معامله بلافاصله بعد از دریافت حالت از محیط است. زمانی که تعداد معاملات به عدد مشخصی برسند آنگاه حالت محیط تشکیل و به عامل داده می‌شود. به‌عبارت‌دیگر زمان انجام عمل عامل در بازه‌های زمانی ثابتی نیست؛ بلکه وابسته به تعداد معاملات بازار متغیر است. قیمت بازار در زمان انجام معامله، قیمت را مشخص می‌کند؛ بنابراین، عامل تنها باید سمت و مقدار معامله را مشخص کند.

برای این منظور، فضای عمل به‌صورت مجموعه‌ای گسسته با سه عضو $\{۱, ۰, -۱\}$ تعریف شده است. ۱- به معنی فروش کل دارایی موجود. ۰ به معنی عدم انجام معامله و ۱ به معنی خرید با کل میزان

نقدینگی است. تصمیم نهایی عامل مبتنی بر عمل مرحله قبل و عملی که در حال حاضر با توجه به خروجی شبکه کیو، پیشنهاد شده است تعیین می‌گردد. در جدول (۱) تصمیم نهایی عامل مشخص‌شده است.

جدول (۱): تصمیم عامل مبتنی بر عمل قبلی و فعلی

تصمیم	عمل پیشنهادی	عمل قبلی
۰	۰	{-۱،۰،۱}
۱	۱	۰
-۱	-۱	۰
۰	۱	۱
-۱	-۱	۱
۱	۱	-۱
۰	-۱	-۱

٤-٤ تابع پاداش

خروجی تابع پاداش نشان‌دهنده میزان کارائی عامل معامله‌گر است و می‌توان آن را مهم‌ترین بخش در فرایند یادگیری عامل در نظر گرفت. در روش‌های یادگیری تقویتی مستقل از محیط، عامل در ابتدا هیچ اطلاعی از ساختار و دینامیک محیط ندارد. تنها با بررسی پاداش دریافتی بعد از هر تصمیم و تغییر وضعیت محیط، یادگیری را انجام می‌دهد.

در این مقاله، تابع پاداش با استفاده از سود حاصل از معامله‌ها تعریف شده است. اگر عامل در حالت s_t عمل a_t را انجام دهد و هزینه انجام تراکنش مالی با C نمایش داده شود، سود دریافتی از رابطه زیر محاسبه خواهد شد:

$$R_t(s_t,a_t) = \begin{cases} (P_{t+1}-P_t)a_t - P_t \times a_t \times C & if\, a_t \neq 0 \\ 0 & if\, a_t = 0 \end{cases} \quad (۱۰)$$

در رابطه (۱۰)، P_t قیمت بازار در زمان انجام معامله t است. a_t عمل عامل را در حالت t نشان می‌دهد و C هزینه تراکنش است. تابع پاداش به نحوی طراحی‌شده است که هزینه‌های انجام معامله در محاسبه سود و زیان لحاظ شود. عامل با استفاده از این تابع، به‌گونه‌ای آموزش می‌بیند که در هر زمان بهترین تصمیم را برای حداکثرسازی سود خود اتخاذ کند.

٤-٥ تابع هزینه

تابع هزینه در الگوریتم DDQN طبق رابطه (۱۱) محاسبه می‌شود [۲۰].

$$L = \|Q^*(s,a) - Q(s,a)\| \quad (۱۱)$$

هدف کمینه‌کردن رابطهٔ (۱۱) است. به‌عبارت‌دیگر آموزش عامل، با کاهش فاصله تابع کیو با مقدار بهینه آن صورت می‌گیرد. مقدار بهینه تابع کیو با $Q^*(s, a)$ نشان داده می‌شود. طبق رابطه (۵) می‌توان تابع بهینه کیو را به‌صورت رابطه (۱۲) بازنویسی کرد.

$$Q^*(s_t, a_t) = \mathrm{E}\left[r_{t+1} + \gamma \max_a Q^*(s_{t+1}, a_{t+1})\right] \quad (۱۲)$$

در رابطه (۱۲)، r_{t+1} مقدار پاداش دریافتی از محیط بعد از انجام عمل a_t است. این مقدار، به‌وسیله تابع پاداش بعد از انجام هر عمل محاسبه می‌شود. $\max_a Q(s_{t+1},a_{t+1})$ ارزش عملی است، که بالاترین مقدار تابع کیوی بهینه را دارد. با استفاده از رابطه‌های (۱۱) و (۱۲) می‌توان تابع هزینه را به‌صورت رابطه (۱۳) تعریف کرد.

$$L = \left\| r_t + \gamma \max_a Q^*(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right\| \quad (۱۳)$$

$Q^*(s_{t+1}, a_{t+1})$ بیانگر مقدار بیشینه برای میانگین مجموع پاداش دریافتی عامل در آینده با شروع از حالت s و انجام عمل a است. این مقدار، خروجی شبکه هدف هست. مجموع $\max_a Q^*(s_{t+1},a_{t+1})$ با پاداش دریافتی، r_t تخمینی از مقدار بهینه تابع کیو می‌دهند. برای بهینه‌سازی تابع هزینه از گرادیان نزولی استفاده می‌شود.

در این مقاله برای افزایش هم‌گرائی، تغییری در تخمین تابع کیوی بهینه داده شده است. با توجه به اینکه معاملات انجام شده در اختیار است، در هر حالت می‌توان بیشترین پاداش یک معامله را محاسبه کرد. از این مقدار در تخمین تابع کیوی بهینه استفاده گردیده است. رابطه (۱٤) تخمین جدید تابع کیوی بهینه را نشان می‌دهد و تابع هزینه جدید در رابطه (۱۵) نوشته شده است.

$$Q^*(s_t,a_t) = \beta_{max} R + (1-\beta)\max_a Q^*(s_{t+1},a_{t+1}) \quad (۱٤)$$

$$L = \left\| r_{t+1} + \gamma[\beta_{max} R + (1-\beta) \max_a Q^*(s_{t+1}, a_{t+1})] - Q(s_t, a_t) \right\| \quad (۱۵)$$

در رابطه (۱۵)، $\max R$ بیشترین پاداش از بین معامله‌های انجام شده در حالت s و β مقداری ثابت در بازه [۰،۱] است به‌کارگیری بیشترین پاداش در رابطه (۱۵) منجر به افزایش هم‌گرائی شده است. به عبارتی این مقدار مانند یک تنظیم‌کننده عمل کرده است.

۵- پیاده‌سازی و ارزیابی نتایج

در این مقاله، عامل معامله‌گر خودمختار مبتنی بر یادگیری تقویتی عمیق آموزش‌داده‌شده است. فضای حالت به‌صورتی متفاوت از تحقیق‌های پیشین و مبتنی بر معاملات انجام شده تعریف شده و همچنین تغییری در تخمین تابع بهینه کیو داده شده است. در این بخش از مقاله جزئیات پیاده‌سازی و نتایج ارزیابی روش پیشنهادی آمده است. برای ارزیابی روش پیشنهادی سؤال‌هایی مطرح شده و برای پاسخ به آن‌ها، آزمایش‌هایی انجام‌گرفته و نتایج در ادامه توضیح داده شده است.

سؤال ۱: آیا تخمین جدید تابع بهینه کیو، تأثیری بر همگرایی و پاداش عامل دارد؟

سؤال ۲: آیا روش پیشنهادی می‌تواند حداکثر کاهش سرمایه را نسبت به دیگر استراتژی‌ها کاهش دهد؟

سؤال ۳: آیا مدل پیشنهادی برای دیگر ارزها نیز کار می‌کند؟

سؤال ٤: آیا مدل پیشنهادی می‌تواند عملکرد بهتری نسبت به استراتژی‌های معاملاتی دیگر داشته باشد؟

پیاده‌سازی مدل با استفاده از زبان برنامه‌نویسی python 3.10 انجام شده است. از شبکه عمیق کانولوشن vgg16 برای تخمین تابع کیو استفاده شده است. شبکه کانولوشن با کتابخانه keras ۲,۱۵ پیاده‌سازی شده است. برای پیاده‌سازی الگوریتم‌های یادگیری تقویتی کلاس‌های reward, action, observer و environment مبتنی بر کتابخانه gym 0.26.2 نوشته شده است. عامل پیشنهادی در این مقاله MY_DQN نامیده شده است. پروژه پیاده‌سازی شده در آدرس گیت‌هاب[1] قابل‌مشاهده و دسترسی است.

در این آزمایش از دو مجموعه‌دادگان معاملات انجام شده و سری زمانی قیمت پایانی مربوط به سه جفت‌ارز BTCUSDT (بیت‌کوین-تتر)، ETHUSDT (اتریوم-تتر) و ETHBTC (اتریوم-بیت‌کوین) استفاده شده است. معاملات انجام شده در بازه ٤ ماه از ۱۰ مرداد لغایت ۱۰ آذر ۱٤۰۳ گردآوری شده‌اند. مجموعه‌دادگان به دو مجموعه آموزش و آزمون تقسیم شده‌اند. از تاریخ ۱۰ مرداد لغایت ۲۰ آبان برای آموزش مدل و از تاریخ ۲۱ آبان لغایت ۱۰ آذر برای آزمون مدل استفاده شده است. به عبارتی ۸۰ درصد دادگان برای آموزش و ۲۰ درصد برای آزمون در نظر گرفته شده است. مدل ابتدا روی جفت‌ارز بیت‌کوین - تتر به تعداد ۱۰۰۰ تکرار آموزش‌دیده است و سپس مدل روی دو جفت‌ارز دیگر با ۱۰۰ تکرار مورد آموزش مجدد قرار گرفته است. ارزیابی مدل روی هر سه جفت‌ارز و با دادگان آزمون صورت‌گرفته و نتایج در ادامه تحلیل شده است.

۵-۱ پارامترهای محیط

یکی از دلایل پیچیدگی در الگوریتم‌های یادگیری تقویتی عمیق تعداد زیاد پارامترهایی است که مقادیر آن‌ها باید تعیین شود. در روش پیشنهادی پارامترهای مدل شامل d، α، β و γ هستند. در ابتدا روش به‌دست‌آوردن مقادیر هر پارامتر شرح داده شده است و سپس در جدول (۲) این مقادیر نشان داده شده‌اند.

پارامتر d، اندازه تنسور ورودی به شبکه کیو را مشخص می‌کند. هر حالت محیط با تنسوری مربعی با ابعاد $4\times d\times d$ تعریف می‌شود؛ لذا مقدار d تعداد معاملات در هر حالت و زمان پاسخ عامل را تعیین می‌کند و در افزایش دقت مدل تأثیر بسزایی دارد. برای تعیین مقدار d، مقادیر مختلف آن در بازه [۳۲, ۱۰] بررسی شده و میانگین دقت[2] مدل در ارزیابی هر سه جفت رمزارز محاسبه گردیده است. شکل (۳) نمودار میانگین دقت مدل برای مقادیر مختلف d را نشان داده است.

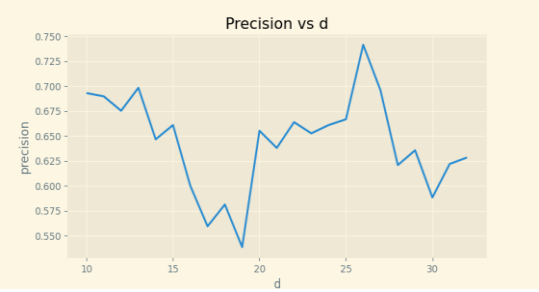
مطابق شکل (۳)، بالاترین دقت مدل برای $d=$ ۲٦ ثبت شده است؛ بنابراین در ارزیابی مدل پارامتر d برابر با مقدار ۲٦ تنظیم می‌گردد.

پارامتر α، مقداری است در بازه [۰،۱] و نرخ یادگیری مدل نامیده می‌شود. این پارامتر میزان تغییر پارامترهای شبکه کیو در هر

[1] https://github.com/MehrdadEslami/cryptoRL

[2] Precision

به‌روزرسانی را مشخص می‌کند. برای تعیین مقدار بهینه، در شروع فرایند آموزش مقدار اولیه آن ۰,۹۹ لحاظ گردیده و در هر دوره آموزش به میزان ۰,۰۱ از آن کم شده است. مقداری به‌عنوان مقدار بهینه لحاظ شده است که نوسان مجموع پاداش مدل در آن دوره آموزش کمترین مقدار را داشته باشد.



شکل (۳): نمودار میانگین دقت مدل برای مقادیر مختلف d

پارامتر γ، فاکتور تخفیف، عددی است در بازه [۰,۱] که به‌منظور همگرا شدن سری پاداش تجمعی استفاده می‌شود. این پارامتر هر چه به ۰ نزدیک‌تر باشد تأثیر پاداش‌های آتی را کاهش می‌دهد و هر چه به ۱ نزدیک‌تر باشد، پاداش آتی را هم‌وزن با آخرین پاداش به‌دست‌آمده لحاظ می‌کند. برای تعیین مقدار بهینه برای این پارامتر از الگوریتم تبرید شبیه‌سازی‌شده[1] استفاده شده است که در آن هدف پیداکردن مقداری برای این پارامتر است که تابع هزینه مدل را کمینه کند.

پارامتر β پارامتری است که در مدل پیشنهادی برای تخمین تابع کیوی بهینه استفاده شده است. این پارامتر برای محاسبه میانگین بالاترین پاداش معاملات انجام در حالت‌های مختلف استفاده شده است. طبق رابطه (۱٤)، از این میانگین برای تخمین تابع کیوی بهینه استفاده شده است.

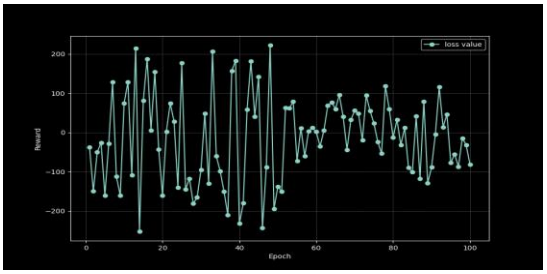
در جدول (۲) مقادیر به‌دست‌آمده برای پارامترهای مختلف محیط آورده شده است.

جدول (۲): مقادیر پارامترهای مدل

epoch = ۱۰۰۰	α = ۰/۰۹۲
batch_size = ۱٦	γ = ۰/۰٥٤
d = ۲٦	β =۰/٦۳۲

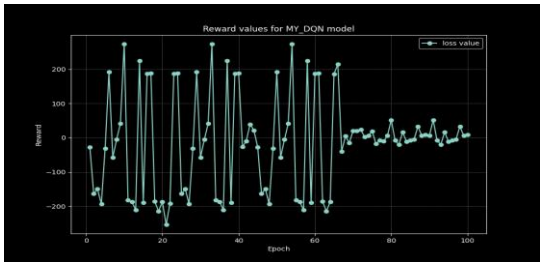
۵-۲ بررسی همگرایی (سؤال ۱)

در یادگیری ماشین جهت بررسی همگرایی مدل خروجی تابع هزینه در فرایند یادگیری، مورد بررسی قرار می‌گیرند. درصورتی‌که مقدار تابع هزینه به صفر نزدیک شود، آنگاه تفسیر به همگرایی مدل می‌شود. اما در یادگیری تقویتی رسیدن تابع هزینه به مقادیر نزدیک به صفر برای محیطی پیچیده؛ مانند بازار رمزارزها قابل‌دسترسی نیست [۷]. اثبات نشده است که الگوریتم یادگیری کیو در فضای حالت نامحدود، همگرا است؛ لذا برای بررسی همگرایی مدل، پاداش‌های دریافتی در طول فرایند یادگیری مورد بررسی قرار خواهد گرفت. اگر پاداش‌های دریافتی در طول فرایند یادگیری افزایشی و در بازه محدودی نوسان کند آنگاه تعبیر به همگرایی مدل می‌شود. نمودار میانگین پاداش دریافتی در طول فرایند یادگیری برای مدل vanilla_DDQN در شکل (٤) و برای MY_DDQN در شکل (۵) آمده است. برای آموزش، مدل هزار بار روی مجموعه‌دادگان آموزش‌دیده است؛ ولی نتیجه ۱۰۰ دوره آموزش آخر در شکل‌های (۳) و (٤) نشان‌داده‌شده است.



شکل (٤): مجموع پاداش دریافتی در مدل vanilla_DQN

فضای حالت در هر دو مدل یکسان است مقادیر پارامترها مشابه یکدیگر مقداردهی شده‌اند و روی یک سیستم یکسان اجرا گرفته شده است.



شکل (۵): میانگین پاداش دریافتی در مدل MY_DDQN

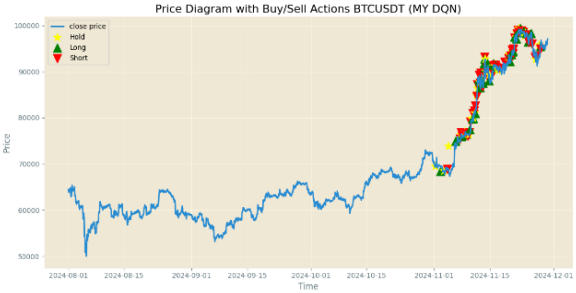
[1] Simulated annealing

در *vanilla-DQN* از تابع هزینه رابطه (۱۳) استفاده شده است و در *MY_DQN* تابع هزینه مبتنی بر رابطه (۱۵) تعریف شده است. در مدل *MY_DQN* بعد از ۹۰۰ دوره میانگین پاداش دریافتی در یک بازه مشخص نوسان می‌کند و این نشانه این است که مدل به بازه‌ای مشخص از پاداش همگرا شده است.

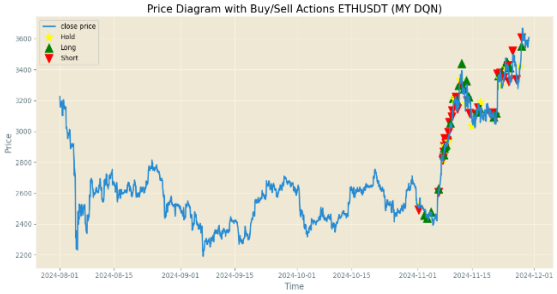
۵-۳ آیا تصمیم‌های مدل دارای دقت کافی است؟ آیا مدل برای دیگر ارزها نیز کار می‌کند؟

یکی از ویژگی‌های بارز مدل ارائه شده، عدم وابستگی مدل به جفت رمزارز خاصی است. دلیل این ویژگی یادگیری مدل از روی معاملات انجام شده است. از طرفی قیمت، میزان و زمان معامله و پاداش دریافتی، نرمال شده‌اند؛ لذا انتظار می‌رود مدل بتواند روی جفت‌ارزهایی که تاکنون ندیده است به‌خوبی عمل کند. برای آموزش مدل، دادگان به دو مجموعه آموزش و آزمون به نسبت ۸۰ به ۲۰ تقسیم شده‌اند. مدل ابتدا روی دادگان آموزشی جفت‌ارز بیت‌کوین-تتر آموزش‌دیده است و سپس همان مدل روی دادگان آموزشی اتریوم-تتر به تعداد ۱۰۰ دوره آموزش مجدد دیده است. برای بررسی مدل، تصمیم عامل روی دادگان آزمون در دو جفت رمزارز بیت‌کوین-تتر و اتریوم-تتر و روی کل دادگان اتریوم-بیت‌کوین محاسبه شده است.

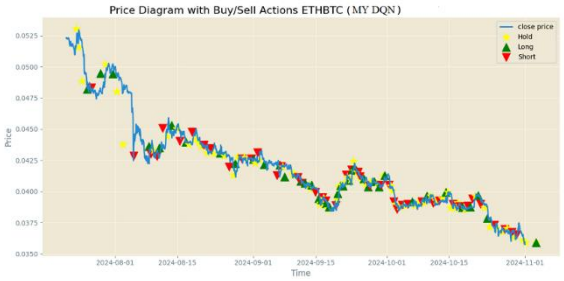
تصمیم‌های هر عامل مستقل از میزان موجودی هر عامل است و فقط مبتنی بر حالت محیط اخذ شده است. خروجی تصمیم‌های مدل برای دادگان آزمون بیت‌کوین-تتر در شکل (٦)، روی دادگان آزمون اتریوم-تتر در شکل (۷) و برای کل دادگان اتریوم-بیت‌کوین در شکل (۸) آمده است.



شکل (٦): تصمیم‌های عامل روی دادگان آزمون بیت‌کوین-تتر



شکل (۷): تصمیم‌های عامل روی دادگان آزمون، اتریوم-تتر



شکل (۸): تصمیم‌های عامل روی دادگان آزمون اتریوم-تتر

هر تصمیم می‌تواند تصمیم خرید (مثلث سبز)، تصمیم فروش (مثلث قرمز) و تصمیم حفظ وضعیت موجود (ضربدر زرد) باشد. برای بررسی کیفیت تصمیم‌های پیش‌بینی شده از دو معیار دقت و بازخوانی[1] استفاده شده است. رابطه (١٦) این دو معیار را تعریف می‌کند:

$$precision = \frac{TruePositive}{TruePositive + FalsePositive} \qquad (١٦)$$
$$recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

در رابطه (١٦)، TP[2] بیانگر تعداد تصمیم‌های درستی است که منجر به سود یا جلوگیری از ضرر می‌شود. FP[3] بیانگر تعداد تصمیماتی است که منجر به ضرر یا ازدست‌دادن سود می‌شوند. FN[4] بیانگر تعداد فرصت‌های ازدست‌رفته است و برابر با تعداد تصمیم‌های نگهداری (عدم انجام معامله) است، در شرایطی که اخذ تصمیم دیگری می‌توانست منجر به سود یا کاهش ضرر گردد.

[1] Precision and Recall
[2] True Positive
[3] False Positive
[4] False Negative

جدول (۳): دقت و بازخوانی تصمیمات مدل

	BTCUSDT	ETHUSDT	ETHBTC
TP	۸۷	٤٦	١٤٢
FP	۳۰	١٦	٦١
FN	٤	١١	١٥
دقت	۰/۷۳	۰/۷۳	۰/٦٩
بازخوانی	۰/۹۵	۰/۸۱	۰/۹۰

دقت به این معناست که از بین تمامی تصمیمات مثبت مدل (سیگنال‌های خرید/فروش)، چه تعداد از آن‌ها به‌درستی پیش‌بینی شده‌اند. دقت نشان می‌دهد حدود ۷۰ درصد از سیگنال‌های خرید یا فروش صحیح بوده‌اند و تنها ۳۰ درصد از سیگنال‌های مثبت مدل اشتباه بوده‌اند.

این دقت نسبتاً بالاست و نشان می‌دهد که مدل توانسته تعداد زیادی از سیگنال‌های خریدوفروش درست را شناسایی کند و خطای زیادی در تشخیص سیگنال‌ها نداشته است.

بازخوانی به معنای نسبت پیش‌بینی‌های درست مثبت به‌تمامی موارد مثبت واقعی است. مقدار بازخوانی ۰,۹٦۷ نشان می‌دهد که مدل توانسته است ۹٦ درصد از تمام سیگنال‌های خریدوفروش واقعی را به‌درستی شناسایی کند. این نشان‌دهنده حساسیت بالای مدل است و اینکه تقریباً همه مواردی که باید خرید یا فروش صورت می‌گرفت، توسط مدل تشخیص داده شده‌اند.

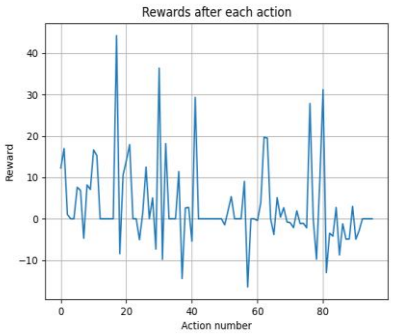
٤-۵ آیا مدل پیشنهادی می‌تواند عملکرد بهتری نسبت به مدل‌های پایه داشته باشد

در این بخش عملکرد مدل پیشنهادی با دو مدل پایه (*B&H* و *S&H*) و مدل *TDQN* ارائه شده در [۱] مقایسه می‌شود. برای بررسی عملکرد روش پیشنهادی سه معیار پایداری سود، حداکثر افت و سود به‌دست‌آمده در نظر گرفته می‌شوند.

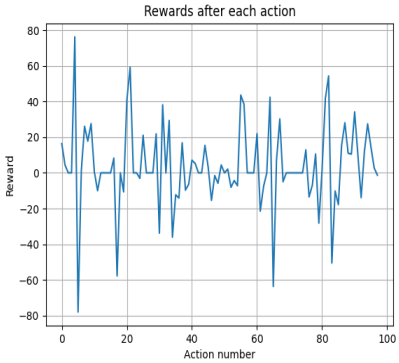
پایداری سود نشان‌دهنده میزان سود به‌دست‌آمده در مقایسه با میزان ریسک آن است. برای این معیار، نسبت شارپ محاسبه شده است. این نسبت مشخص می‌کند به‌ازای هر واحد ریسک، چه مقدار سود کسب شده است.

حداکثر افت به بیشترین کاهش از اوج تا کف سرمایه، در طول دوره معاملاتی اشاره دارد. این معیار نشان می‌دهد که در بدترین شرایط، بیشترین ضرری که در طول دوره ممکن است رخ دهد، چه میزان بوده است. به‌صورت ریاضی، برای هر نقطه در زمان، حداکثر افت به شکل زیر تعریف می‌شود:

سود به‌دست‌آمده، عبارت از میزان کل سرمایه در پایان دوره نسبت به مقدار اولیه آن است.



شکل (۹): پاداش دریافتی بعد از هر تصمیم برای جفت رمزارز، اتریوم-بیت‌کوین



شکل (۱۰): پاداش دریافتی در اتریوم-تتر

شکل (۹) و (۱۰) نشان می‌دهد که بیشتر تصمیم‌های روش پیشنهادی منجر به سود شده است در بازه‌هایی نیز سود منفی یا سود صفر داشته است؛ اما میانگین پاداش به‌دست‌آمده مثبت است.

در جدول (۳) روش پیشنهادی با دو روش پایه و همچنین روش TDQN مقایسه شده است. سه معیار برای یک دوره زمانی مشخص محاسبه شده است.

طبق جدول (س) روش پیشنهادی در بازه زمانی مشخصی سود بیشتری کسب کرده است و حداکثر افت سرمایه کمتری به دست آورده است. اما نسبت به روش TDQN نسبت شارپ کمتری را کسب کرده است.

جدول (۳): معیارهای ارزیابی برای مقایسه مدل پیشنهادی

معیار ارزیابی	میانگین نسبت شارپ			حداکثر افت سرمایه			مجموع پاداش به‌دست‌آمده		
مدل	بیت‌کوی ن – تتر	اتریوم – تتر	اتریوم – بیت‌کوین	بیت‌کوین – تتر	اتریوم – تتر	اتریوم – بیت‌کوین	بیت‌کوین – تتر	اتریوم – تتر	اتریوم – بیت‌کوین
روش پیشنهادی	۰/۸	۰/۱۸	۰/۱۷	۹%	۱۰%	۲۱%	۳۸۶	۳۶۱	۳۰۰
مدل پایه B&H	–	-	-	۵۵%	۳۳%	۲۳%	٤٨٦	-۳۳۳	-۲۳۳
مدل پایه S&H	–	-	-	۵۵%	۳۳%	۲۳%	-٤٨٦	۳۳۳	۲۳۳
مدل TDQN	۰/۷۹	۰/۵۷	۱/۳٦		۱٦%	۱۵%	۲۵٤	۲۹۹	۲۱٤

٦- نتیجه‌گیری و کارهای آتی

در این مقاله رویکردی متفاوت در آموزش عامل معامله‌گر مبتنی بر یادگیری تقویتی عمیق ارائه شده است. در این رویکرد عامل توجهی به روند تغییرات قیمت و یا خروجی‌های ابزارهای تکنیکال و بنیادی ندارد. فقط با تحلیل رفتار دیگر معامله‌گران بازار عمل می‌کند.

به دلیل فقدان مجموعه‌دادگان معاملات انجام شده، تمام معاملات صورت‌گرفته در صرافی رمزارز HitBTC برای سه جفت رمزارز (بیت‌کوین – تتر، اتریوم – تتر، اتریوم – بیت‌کوین) از تاریخ ۱۰ مرداد لغایت ۱۰ آذر ۱٤۰۳ به مدت ۱۲۰ روز جمع‌آوری شده است. تعداد معاملات، برای جفت‌ارز بیت‌کوین – تتر نزدیک به ۱۲۳ هزار معامله، اتریوم – تتر نزدیک به ۸۵ هزار معامله و اتریوم – بیت‌کوین نزدیک به ۸۹ هزار معامله شده است. باتوجه‌به حجم بالای معامله‌ها از دو شبکه عمیق کانولوشن با معماری یکسان برای تخمین تابع کیو و تابع کیوی بهینه، استفاده گردیده است.

نتایج پیاده‌سازی روش پیشنهادی نشان از همگرایی مدل دارد. برای اندازه‌گیری کیفیت تصمیم‌های عامل، دقت و بازخوانی مدل اندازه‌گیری شده است. دقت ۷۰ درصدی و بازخوانی ۹۳ درصدی نشان از موفقیت روش دارد. همچنین روش پیشنهادی با دو استراتژی سنتی و روش TDQN[1] از مقاله [۱] مقایسه شده است. روش پیشنهادی در مدت‌زمان مشخصی، توانسته است، سود بیشتری کسب کند.

نتایج پیاده‌سازی روش نشان می‌دهد که مدل همگرا شده است. همچنین مدل مجموع پاداش بیشتری نسبت به روش‌های پایه و مدل TDQN به دست آورده است. میزان حداکثر افت سرمایه بسیار بهتر از دیگر روش‌ها بوده است و آن نیز به دلیل واکنش سریع‌تر عامل در شرایط نوساناتی است. اما معیار نسبت شارپ روش پیشنهادی عملکرد ضعیف‌تری نسبت به روش TDQN داشته است. این نشان می‌دهد مدل نیاز به بهبود دارد. این مقاله نشان داده است تحلیل رفتار معامله‌گران بازار می‌تواند منجر به اخذ تصمیم‌های بهتری گردد. ترکیب این روش با روش‌های پیش‌بینی قیمت می‌تواند رویکرد جدیدی در طراحی عامل‌های معامله‌گر ایجاد کند.

به‌عنوان یک کار تحقیقاتی، پیشنهاد می‌شود تحلیل رفتار معامله‌گران با استفاده از مدل‌های actor-ciritc انجام شود. همچنین به نظر می‌آید استفاده از روش تحلیل رفتار معامله‌گران همراه با تحلیل تغییرات قیمتی و خروجی ابزارهای تکنیکال می‌تواند مدل را بهبود بخشد.

[1] Trading Deep Q Network

References

[1] T. Theate, and D. Ernst, “An application of deep reinforcement learning to algorithmic trading,” *Expert Systems with Applications*, vol. 173, p. 114632, Jul. 2021.

[2] Y. Huang, X. Wan, L. Zhang, and X. Lu, “A novel deep reinforcement learning framework with BiLSTM-Attention networks for algorithmic trading,” *Expert Systems with*

[3] X. Cheng, J. Zhang, Y. Zeng, and W. Xue, “MOT: A Mixture of Actors Reinforcement Learning Method by Optimal Transport for Algorithmic Trading,” *Lecture Notes in Computer Science*, Springer, Singapore, 2024, vol. 14648, pp. 30-42.

[4] Z. Huang, N. Li, W. Mei, and W. Gong, “Algorithmic trading using combinational rule vector and deep reinforcement learning,” *Applied Soft Computing*, vol. 147, pp. 110802–110802, No. 2023.

[5] Shavandi and M. Khedmati, “A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets,” *Expert Systems with Applications*, p. 118124, Jul. 2022.

[6] B. Hirchoua, B. Ouhbi, and B. Frikh, “Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy,” Expert Systems with Applications, vol. 170, p. 114553, May 2021, doi: https://doi.org/10.1016/j.eswa.2020.114553.

[7] M. Taghian, A. Asadi, and R. Safabakhsh, “A Reinforcement Learning Based Encoder-Decoder Framework for Learning Stock Trading Rules,” pp. 1–39, 2021, [Online]. Available: http://arxiv.org/abs/2101.03867.

[8] S. Carta, A. Ferreira, A. S. Podda, D. Reforgiato Recupero, and A. Sanna, “Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting,” Expert Systems with Applications, vol. 164, p. 113820, Feb. 2021, doi: https://doi.org/10.1016/j.eswa.2020.113820.

[9] D. Kanzari and Y. Ridha Ben Said, “A complex adaptive agent modeling to predict the stock market prices,” Expert Systems with Applications, vol. 222, p. 119783, Jul. 2023, doi: https://doi.org/10.1016/j.eswa.2023.119783.

[10] B. Yang, T. Liang, J. Xiong, and C. Zhong, “Deep reinforcement learning based on transformer and U-Net framework for stock trading,” Knowledge-Based Systems, vol. 262, p. 110211, Feb. 2023.

[11] Z. Zhang, S. Zohren, and S. Roberts, “Deep Reinforcement Learning for Trading,” The Journal of Financial Data Science, vol. 2, no. 2, pp. 25–40, Mar. 2020, doi: https://doi.org/10.3905/jfds.2020.1.030.

[12] N. majidi, M. Shamsi, and F. Marvasti, “Algorithmic Trading Using Continuous Action Space Deep Reinforcement Learning,” SSRN Electronic Journal, 2022, doi: https://doi.org/10.2139/ssrn.4276310.

[13] Mahdi Massahi, and Masoud Mahootchi, “A deep Q-learning based algorithmic trading system for commodity futures markets,” Expert systems with applications, vol. 237, pp. 121711–121711, Mar. 2024. doi: https://doi.org/10.1016/j.eswa.2023.121711.

[14] R. S. Sutton, Reinforcement Learning: An Introduction, Second Edition, Cambridge, Massachusetts: The Mit Press, 2018.

[15] E. F. Fama, “The Behavior of Stock-Market Prices,” The Journal of Business, vol. 38, no. 1, pp. 34–105, Jan. 1965, doi: https://doi.org/10.1086/294743.

[16] H. Van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-Learning,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1, Mar. 2016, doi: https://doi.org/10.1609/aaai.v30i1.10295.

[17] O. B. Sezer and A. M. Ozbayoglu, “Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach,” Applied Soft Computing, vol. 70, pp. 525–538, Sep. 2018, doi: https://doi.org/10.1016/j.asoc.2018.04.024.

[18] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, Oct. 2018.

[19] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, “Quant GANs: deep generation of financial time series,” Quantitative Finance, vol. 20, no. 9, pp. 1419–1440, Apr. 2020.

[20] J. Moody, L. Wu, Y. Liao, and M. Saffell, “Performance functions and reinforcement learning for trading systems and portfolios,” Journal of Forecasting, vol. 17, no. 56, pp. 441–470, Sep. 1998.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, et al., “Playing Atari with Deep Reinforcement Learning,” pp. 1–9, 2013, [Online]. Available: http://arxiv.org/abs/1312.5602.

[22] D. Silver, et al., “Mastering the game of Go without human knowledge,” Nature, vol. 550, no. 7676, pp. 354–359, Oct. 2017, doi: https://doi.org/10.1038/nature24270.

[23] H. Van Hasselt, “Double Q-learning,” in Conf. Neural Inf. Process Syst, NIPS, 2010, pp. 1–9.
[24] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, “Deep Direct Reinforcement Learning for Financial Signal Representation and Trading,” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 3, pp. 653–664, Mar. 2017. doi:http://doi.org/10.1109/tnnls.2016.252240
[25] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, “Machine learning techniques and data for stock market forecasting: A literature review,” Expert Systems with Applications, vol. 197, p. 116659, Jul. 2022, doi: https://doi.org/10.1016/j.eswa.2022.116659.
[26] S. Fallahpour, H. Hakimian, “Paired Trading Strategy Optimization Using the Reinforcement Learning Method: Intraday Data of Tehran Stock Exchange” Financial Research Journal, vol. 21, no. 1, pp. 19-34, May 2019. [In Persian]
[27] B. Sabahi, “Designing Trading Strategies Based on Deep Reinforcement Leaerning Methodes” M.S thesis, Department of Financial Management, Tehran Univ. Technol., Tehran, 2022. [In Persian]

Journal of Applied And Basic Machine Intelligence Research (ABMIR)

Applied and Basic Machine Intelligence Research

Vol. 2, No. 2, Autumn & Winter 2024, pp. 49 - 65

Using Deep Q Network to develop an autonomous agent for trading in the cryptocurrency market, Focusing on traders' behavior

Seyed Mehrdad Eslami[1], Mehdi Agha Sarram[2*], Mohamad Ali Zare Chahoki[2]

[1] Phd candidate, Computer Engineering Department, Yazd University, Yazd, IRAN

[2] Associate Professor, Computer Engineering Department, Yazd University, Yazd, IRAN

Article Information

Original Research Paper

Received:
2024 October 29

Accepted:
2025 January 1

Keywords:
Autonomous Trading Agent, Deep Reinforcement Learning, Trades Behaviors, Deep Q Network

Corresponding Author*:
mehdi.sarram@yazd.ac.ir

Abstract

Crypto currency market is a complex, uncertain and dynamic environment with significant volatility. Developing a trading strategy in this market is highly challenging and a key area of academic research. In this article, An autonomous trading agent has been designed to analyze the effects of traders' behavior (transactions they do) on changing market conditions. Although many factors influence the market, these effects impact ultimately through traders behaviors. In this article, The agent makes decisions only by reviewing and analyzing the transactions which has been done by traders. The agent is built using DDQN reinforcement learning algorithm. To train the agent, all HitBTC`s transactions during nearly 3 months for 3 cryptocurrency pairs have been gathered. The results show that the model converges and is stable. As a result, The transactions data are important source for decision making. Combining this method with price prediction methods can be a new approach in designing trader agents.

: 10.22034/ABMIR.2025.22203.1062

E-ISSN: 2821-2037