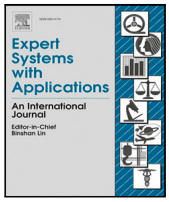
Expert Systems With Applications 238 (2024) 121897

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Deep reinforcement learning applied to a sparse-reward trading environment with intraday data

Lucas de Azevedo Takara [a,*], André Alves Portela Santos [b], Viviana Cocco Mariani [a,c], Leandro dos Santos Coelho [a,d]

[a] Department of Electrical Engineering, Federal University of Parana, UFPR, Avenida Coronel Francisco Herácito dos Santos, 100, 81530–000, Curitiba, Parana, Brazil
[b] CUNEF Universidad, Calle Almansa, 101, 28040, Madrid, Spain
[c] Department of Mechanical Engineering, Pontifical Catholic University of Parana, PUCPR, Rua Imaculada Conceição, 1155, 80215–901, Curitiba, Parana, Brazil
[d] Industrial and Systems Engineering Graduate Program, Pontifical Catholic University of Parana, PUCPR, Rua Imaculada Conceição, 1155, 80215–901, Curitiba, Parana, Brazil

ARTICLE INFO

Keywords:
Quantitative trading
Deep reinforcement learning
Deep Q-learning
Sparse-reward trading environment
Financial signal processing

ABSTRACT

Deep reinforcement learning (DRL) has made remarkable strides in empowering computational models to tackle intricate decision-making tasks. In quantitative trading, DRL trading agents have emerged as a means to optimize decisions across diverse market scenarios, culminating in developing profitable trading strategies by assimilating knowledge from past experiences. This study introduces an innovative trading system centered around the Deep Q-Network (DQN) algorithm called Extended Trading DQN (ETDQN). ETDQN stands out by its ability to adapt its learning process to trade effectively across varying market conditions, with feedback received exclusively upon trade liquidation. This contrasts with models that inundate agents with continuous feedback signals. ETDQN leverages distributional learning and several other independent extensions to enhance its DRL capabilities, streamlining its decision-making process. The model accomplishes this by prioritizing experiences encompassing diverse sub-objectives, facilitating the accumulation of maximum profit while obviating the need for intricate reward fine-tuning. Through extensive training on three distinct financial time series signals, ETDQN demonstrates its proficiency in identifying trading opportunities, particularly during periods of heightened price volatility. Notably, the model exhibits a more assertive approach towards managing annual returns volatility compared to the conventional DQN model, outperforming it by a factor of 1.46 and 7.13 concerning average daily cumulative returns, as evidenced in the historical data of Western Digital Corporation and the Cosmos cryptocurrency, respectively.

1. Introduction

Deep reinforcement learning (DRL) represents a combination of reinforcement learning (RL) and deep learning, propelling advancements in diverse fields such as gaming (Mnih et al., 2015) and advanced control systems (Levine et al., 2016). DRL empowers computers to undertake intricate decision-making tasks by simulating the interactive learning process of agents in dynamic environments. These algorithms equip agents with the capacity to adapt through trial and error, learning from the feedback generated by their actions and experiences. Leveraging this paradigm, researchers have introduced various trading systems based in DRL, aiming to maximize profits by identifying and capitalizing on lucrative opportunities.

In the realm of trading, DRL-based systems have gained prominence, addressing distinct trading tasks such as portfolio management, market making, and stock trading (Jang & Seong, 2023; Kumar, 2023; Sun et al., 2022). Value-based techniques, exemplified by the Deep Q-Network (DQN) and its derivatives, have been extensively employed. These methods guide trading agents in making optimal decisions from a limited set of actions: Buy, sell, or hold. For instance, Shin et al. (2019) propose a DQN-based system that employs stock trading data to generate charts, utilizing them as inputs for an artificial neural network. Carta et al. (2021) introduce a multi-layer, multi-ensemble DQN stock trader, optimizing profits and generating stock signals through

* Corresponding author.
E-mail addresses: lucastakara@ufpr.br (L.A. Takara), andre.santos@cunef.edu (A.A.P. Santos), viviana.mariani@pucpr.br (V.C. Mariani), leandro.coelho@pucpr.br (L.S. Coelho).

https://doi.org/10.1016/j.eswa.2023.121897
Received 20 March 2023; Received in revised form 13 September 2023; Accepted 27 September 2023
Available online 6 October 2023

several trading agents. Sagiraju and Mogalla (2022) explore intelligent DQN agents, market sentiment, and historical data to enhance investment decisions.

Despite the proliferation of DRL-based trading systems in literature, some still face certain limitations. These restrictions often manifest as constant feedback signals provided to agents. This stems from the intricate exploration and complex reward adjustment procedures required to steer the algorithm towards a profitable policy, rendering it reliant on continuous information updates (Li et al., 2019; Shavandi & Khedmati, 2022; Théate & Ernst, 2021). Furthermore, these systems often rely on fixed, regular time intervals for sampling (Hao et al., 2023; Huang et al., 2023; Singh et al., 2022), and value-based methods typically adhere to standard expectation calculations instead of distributional learning (Brim, 2020; Brim et al., 2022; Suliman et al., 2022; Wu et al., 2020).

Constant feedback can inhibit environment exploration, impeding the agent's ability to generalize its decision-making process effectively. It is important to acknowledge that financial markets do not generate meaningful information continuously (de Prado, 2018). Therefore, agents should receive feedback primarily at the conclusion of trading events instead of continuously throughout. Additionally, learning through expectation can hinder convergence due to the influence of extreme values.

This study introduces a DRL algorithm variant, Extended Trading DQN (ETDQN), aimed at addressing the limitations mentioned in existing literature while enhancing decision-making capabilities. ETDQN introduces an alternative exploration method and an event-driven sampling approach, allowing it to generalize trading decisions across diverse market scenarios while receiving a straightforward exponential profit-and-loss reward at the end of each trade.

ETDQN tackles the limited exploration issue by incorporating prioritized (Schaul et al., 2016) and hindsight (Andrychowicz et al., 2017) experience replays to select meaningful experiences based on prioritization. In addition, it adopts the dueling neural network architecture (Wang et al., 2016) combined with noisy linear layers (Fortunato et al., 2018) to promote exploration, avoiding the conventional ϵ-greedy heuristic approach.

In ETDQN, each experience tuple is enriched with various subgoals, aiding the model in optimizing its decisions to maximize profits, thus eliminating the need for intricate reward tuning. Furthermore, noisy linear layers introduce parametric noise into the artificial neural network, enabling learned perturbations to drive exploration. Meanwhile, the dueling architecture distinguishes valuable states without necessitating an in-depth understanding of the impact of each action in every state. In contrast to existing DRL-based trading systems that adopt regular time sampling, our approach adheres to a market activity-driven sampling method proposed by de Prado (2018). This approach aligns the agent's information updates with the pace at which the market processes information, avoiding oversampling during periods of low activity and undersampling during high activity.

Finally, ETDQN extends traditional value-based DRL approaches adopted in trading, such as Théate and Ernst (2021), by embracing distributional learning rather than relying solely on the standard expectation. This modification aligns with recommendations from Bellemare et al. (2017), highlighting the benefits of distributional learning in optimizing various statistics beyond just the mean.

ETDQN is evaluated against two classical benchmarks: *Buy-and-Hold* (B&H) and *Sell-and-Hold* (S&H). B&H represents a passive investment strategy where an investor purchases an asset and holds it for the long term, expecting appreciation in its value. Conversely, S&H involves borrowing an asset, selling it to profit from its subsequent depreciation, and repurchasing it at a lower price.

To assess ETDQN's performance, the Trading DQN (TDQN) algorithm, following the instructions outlined in the original paper (Théate & Ernst, 2021) was implemented. TDQN represents existing value-based DRL systems, offering constant feedback to the agent and relying on standard time-based sampling. To gauge the efficacy of ETDQN, we also compare it to a strategy termed Random Action, which selects actions arbitrarily based on a discrete uniform distribution. Performance evaluation encompasses cumulative returns (equity curve), the 6-month rolling Sharpe ratio, maximum drawdown, and annual average return (AAR) metrics.

In summary, the following contributions have been provided by this paper:

- Propose an intraday trading model for financial markets, showcasing its profitability in terms of cumulative and annual returns.
- Explore a pre-processing approach aligned with market activity, which contrasts with conventional time sampling methodologies.
- Evaluate an alternative approach to agent exploration, incorporating noisy linear layers within dueling neural networks instead of relying on the standard ϵ-greedy approach.
- Enhance learning in DRL trading systems by adopting distributional learning, extending beyond traditional expected value calculations.
- Present a model that prioritizes experiences and incorporates subgoals, thereby aiding learning through delayed feedback using a simplified reward function.

The remainder of this paper is organized as follows. This introduction, Section 1, presents the main contributions of this study. Section 2 describes the literature review. Section 3 depicts the experimental and iShares Core S&P 500 ETF, Western Digital Corporation, and Cosmos/United States Dollar Tether (ATOM/USDT) cryptocurrency data sets. Section 4 presents the proposed model. Section 5 contains the presentation and discussion of the results achieved by the ETDQN model compared to other models. Finally, Section 6 concludes the article by summarizing the strengths and shortcomings of the proposed approach and looks forward to future research directions.

2. Related work

In this section, we conduct a systematic literature review (SLR) focused on the application of DRL within trading tasks. Our aim is to provide a comprehensive overview of the existing literature in this domain by gathering, analyzing, and discussing research papers related to RL, deep learning, and DRL as they pertain to trading. Additionally, this literature review encompasses an extensive search across academic research databases such as Scopus, Web of Science, and Google Scholar. Our search scope spans from 2017 to 2022 to ensure the inclusion of the most current and relevant information on the subject matter.

Following the guidelines proposed by Kitchenham and Brereton (2013), this systematic literature review (SLR) follows a well-structured three-stage process: Planning, conduction, and data synthesis. The planning stage involves defining the research questions that guide our investigation. In the conduction phase, we establish inclusion and exclusion criteria, design a comprehensive search strategy for academic research databases, validate the selection of papers, and perform data extraction. Lastly, we present a synthesized overview of the extracted data results. This systematic literature review addresses the following research questions: (i) What are the main RL methods applied to quantitative trading? (ii) What are the main performance metrics used on these algorithms? (iii) What are the main markets to which the models have been applied?

The following inclusion criteria are adopted: (i) Primary studies publications; (ii) Published within the last five years; (iii) English language; (iv) Journal and conference published papers; (v) Relevant publications to deep learning, RL, and DRL topics related to quantitative trading tasks. The exclusion criteria are set to identify publications that are eventually removed from the research study. The following criteria are adopted: (i) Secondary studies publications; (ii) Technical reports and pre-print papers; (iii) Non-English language paper publications; (iv) Publications that do not approach RL or DRL methods related

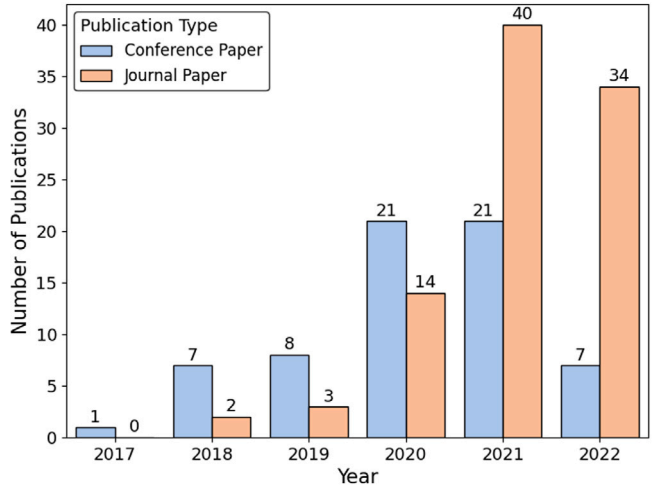


Fig. 1. Number of publications per year of DRL algorithms applied to quantitative trading tasks between 2017 and 2022.

to quantitative trading tasks; (v) Non-downloadable publications on their respective research database.

To construct precise search queries, we employ OR and AND operators strategically. The OR operator links search groups, while synonyms are connected using the AND operator. The resulting keywords are as follows: ("Finance Trading" OR "Quantitative Trading" OR "Algorithmic Trading" OR "Trading" OR "Strategy Trading") AND ("Reinforcement Learning" OR "Deep Learning" OR "Deep Reinforcement Learning").

In the initial stage, no restrictions are placed on fields, date ranges, or languages. This yields a total of 808 publications. Subsequently, papers are meticulously chosen according to inclusion criteria, and the search scope is narrowed down to article titles, abstracts, and keywords. This refinement results in 376 selected publications. Next, reviewing these selected papers leads to the exclusion of 1 article not written in English, 47 not published between 2017–2022, and the identification and removal of 42 duplicates. A closer analysis of their respective titles eliminates 44 papers that fall outside the scope of our review, as they mentioned DRL methods but did not apply them to trading tasks. In the fourth stage, papers are further refined based on a thorough examination of their titles, abstracts, and conclusions, resulting in a final selection of 188 papers. Finally, the ultimate exclusion criteria are applied through comprehensive analysis, leading to the retention of 158 publications.

Fig. 1 illustrates the yearly distribution of publications related to DRL applied to trading tasks from 2017 to 2022. Our analysis comprises a selection of 90 journal articles and 65 conference publications, with the highest concentration occurring in 2021. In that specific year, a notable total of 40 journal papers and 21 conference articles were published. The bar chart depicts a consistently increasing trend over the years, signifying that each subsequent year witnessed a greater number of publications than the preceding one, with the exception of 2022.

Fig. 2 presents a breakdown of publications related to DRL sourced from various research databases, including Elsevier, Institute of Electrical and Electronics Engineers Xplore digital library (IEEE Xplore), Wiley, Association for Computing Machinery (ACM), Springer, Hindawi, and others. The data highlights that most of these publications, accounting for 56.96%, originate from journals, while the remaining 43.04% are sourced from conferences. Among the research databases, Elsevier stands out with the most substantial contribution, comprising 39 papers. Following closely are IEEE Xplore and the Multidisciplinary Digital Publishing Institute (MDPI), with 30 and 26 publications, respectively.

Jiang and Liang (2017) evaluate a Deep Deterministic Policy Gradient (DDPG) agent combined with a Convolutional Neural Network (CNN) for cryptocurrency portfolio management. The study aims to maximize returns while mitigating risk, assessing performance using metrics such as the Sharpe ratio, maximum drawdown, final portfolio value, and standard deviation. The proposed model is compared to various benchmarks, including B&H and constant rebalanced portfolios.

Si et al. (2017) combine recurrent Reinforcement Learning (RRL) and Long Short-Term Memory (LSTM) in a multi-objective framework for trading index-based futures contracts. They measure profit and risk separately, using metrics such as Sharpe ratio, annual profit, and total trading number. The system is tested on futures contracts with input data consisting of 1-minute closing prices.

Li et al. (2019) extend both value-based DQN and actor-critic Asynchronous Advantage Actor-Critic (A3C) algorithms to trading markets. They employ a LSTM module to capture temporal patterns from market observations. The study uses various technical indicators and introduces a novel positions-embedded action space. The reward function is based on the Sharpe ratio.

Shin et al. (2019) propose a deep multi-modal RL policy that combines CNN and LSTM neural networks in a DQN-based agent. The model is trained on 256 stocks and uses various indicators to generate input images for the CNN. Performance is evaluated based on metrics including average returns, standard deviation, maximum profits, minimum returns, and Sharpe ratio.

Wu et al. (2020) introduce adaptive stock trading strategies that incorporate gated recurrent units into DQN and DDPG algorithms. They use indicators such as moving averages (MA), exponential moving averages (EMA), moving average convergence/divergence (MACD), volatility rank, and on-balance volume. The Sortino ratio and rate of return are used as performance metrics.

Conegundes and Pereira (2020) investigate the potential of using DDPG for asset allocation in the Brazilian stock market. The study uses open and close prices as input data and compares the model's performance to stock portfolios suggested by Brazilian banks and brokers. Performance metrics include annual returns, cumulative returns, and maximum drawdown.

Yang et al. (2020) explore ensemble methods, training an agent with an ensemble trading strategy composed of three actor-critic algorithms: Proximal Policy Optimization (PPO), Advantage Actor Critic (A2C), and DDPG. The study applies the strategy to Dow Jones 30 constituent stocks and evaluates performance using metrics such as cumulative return, annualized return, volatility, Sharpe ratio, and maximum drawdown.

Théate and Ernst (2021) introduce Trading Deep Q-Network (TDQN), a DQN-inspired algorithm tailored for trading. TDQN incorporates Double DQN, a feed-forward network architecture, the Adam optimizer, and Huber loss. It is trained on historical data from various indexes and stocks, and its performance is compared to benchmark strategies using metrics like the Sharpe ratio, profit & loss, annualized returns, and annualized volatility.

Carta et al. (2021) present a multi-DQN algorithm applied to real-world trading scenarios, including the S&P 500 futures market and J.P. Morgan. The study employs meta-feature generation based on Gramian angular field images from price time series data. Various performance metrics, such as Sharpe ratio, profit & loss, annualized return, volatility, profitability, Sortino ratios, and maximum drawdown, are used to evaluate the proposed model, which is compared to a B&H benchmark.

Koratamaddi et al. (2021) suggest a sentiment-aware extension to the adaptive DDPG algorithm. The model adjusts the Q-value change rate based on sentiment prediction errors and incorporates a sentiment confidence score into the state. Performance is assessed using reward functions involving portfolio value changes and market sentiment confidence scores.

Sagiraju and Mogalla (2022) propose a framework that includes four DRL algorithms: A2C, PPO, DDPG, and DQN, applied to historical stock

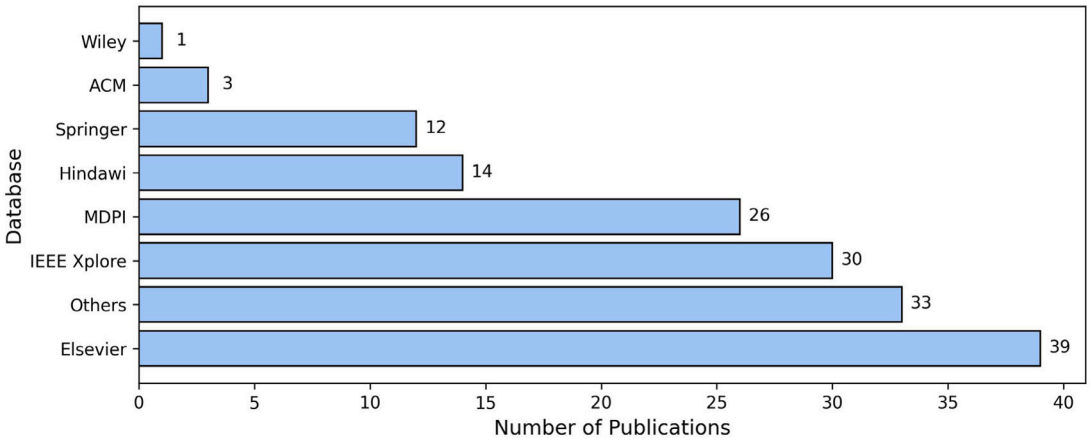


Fig. 2. Number of publications by research databases of DRL methods applied to quantitative trading tasks between 2017 and 2022.

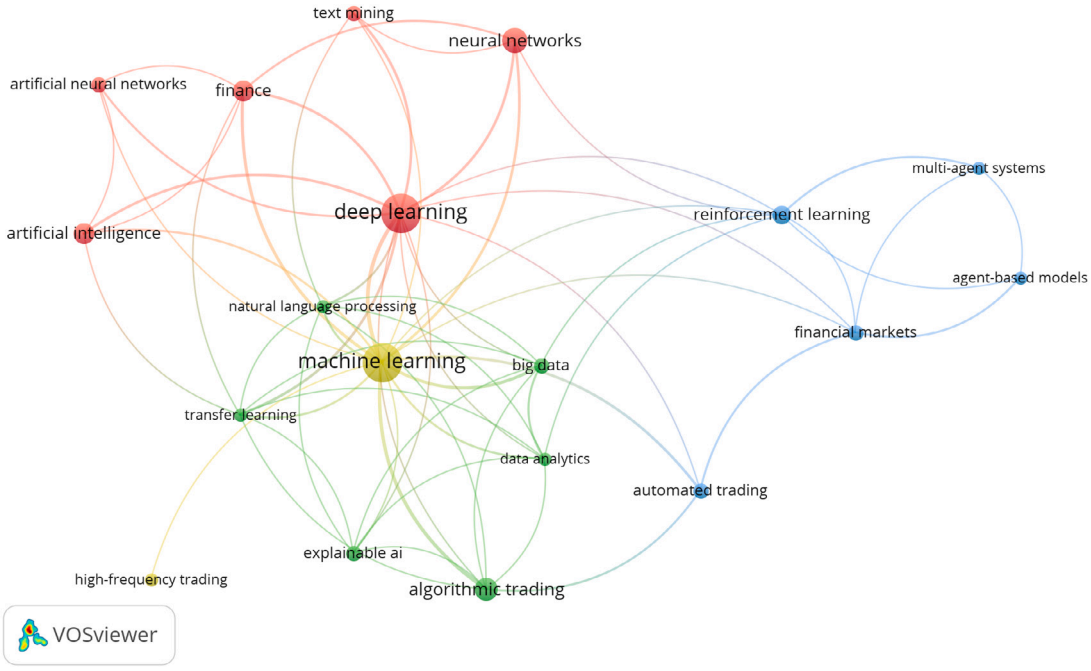


Fig. 3. Co-occurrence network of DRL-related publications applied to quantitative trading tasks between 2017 and 2022.

and Twitter market sentiment data. The study evaluates their performance using various financial metrics, including Sharpe ratio, Sortino ratio, maximum drawdown, cumulative returns, annual volatility, and annualized investment return.

2.1. Data synthesis

In this research, we employ the VOSviewer (van Eck & Waltman, 2010) software to conduct bibliometric network visualization and analysis. By exporting data from selected papers into ".ris" format, we construct the co-occurrence network. The resulting visualization highlights the most significant keyword clusters, with a focus on machine learning and deep learning, and RL as prominent themes. The VOSviewer tool aids in identifying key research directions and connections within the literature.

Fig. 3 presents the co-occurrence networks of keywords extracted from the selected research papers. Of the 444 keywords considered, only 19 meet the threshold for inclusion. These keywords are grouped into four clusters, with machine learning and deep learning emerging as the most prominent themes, each containing 25 occurrences.

Fig. 4 reveals that the actor-critic DDPG stands out as the predominant DRL algorithm employed in quantitative trading tasks, commanding 42.1% of the total citations in DRL-related papers. Following closely, DQN and PPO hold the second and third positions with 28.9% and 18.4%, respectively. A2C (5.3%), A3C (2.6%), and RRL (2.6%) also find mention in a subset of the reviewed studies.

Table 1 displays the predominant performance metrics employed for assessing algorithm performance in the aforementioned studies. Specifically, five papers (6.5%) incorporate annualized returns, eight (10.4%) employ maximum drawdown, ten (13%) utilize the Sharpe ratio, and three (3.9%) make use of standard deviation and total profits as performance measures. Notably, the Sharpe ratio emerges as the most prevalent metric for evaluating performance, followed by maximum drawdown and annualized return.

Table 2 provides an overview of the markets to which the models under review are predominantly applied. The S&P500 market emerges as the foremost choice, featured in 7 papers, accounting for 14.3% of all RL-related publications in this review. Following closely is the Stock-IF market, with 4 works attributed to it. Cryptocurrencies, the Dow Jones, and National Association of Securities Dealers Automated

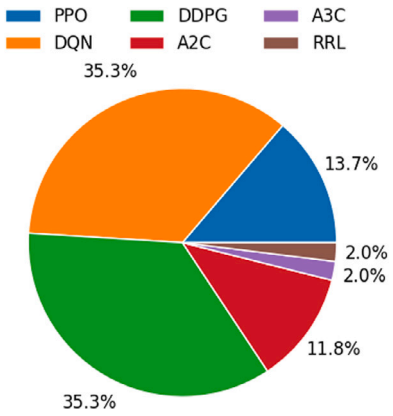


Fig. 4. Appearance amount in the percentage of each DRL algorithm applied to quantitative trading tasks.

Table 1
Main performance metrics applied to the reviewed works.

Performance metrics	Absolute frequency	Percentage frequency
Sharpe Ratio	10	13%
Maximum Drawdown	8	10.4%
Annualized Returns	5	6.5%
Standard Deviation	3	3.9%
Total Profits	3	3.9%
Others	48	62.3%
Total	**77**	**100%**

Table 2
Main markets applied to reviewed works.

Markets	Absolute frequency	Percentage frequency
S&P500	7	14.3%
Stock-IF	4	8.2%
Cryptocurrency	3	6.1%
Dow Jones	3	6.1%
NASDAQ	3	6.1%
FOREX	2	4.1%
Stock-IC	2	4.1%
Others	25	51%
Total	**49**	**100%**

Quotations (NASDAQ) markets each comprise 6.1% of the reviewed publications. Lastly, the foreign exchange market (FOREX) and Stock-IC markets make up 4.1% of the reviewed works.

3. Materials

In this study, we retrieve transaction data for the iShares Core S&P 500 ETF and Western Digital Corporation from the Kibot website (Oricsoft, 2017), which provides free intraday data with tick prices and share amounts. We exclude bid/ask features from the data, focusing solely on tick prices and share quantities. The original dataset for the iShares Core S&P 500 ETF spans from September 28, 2009, at 09:30:00 to December 2, 2022, at 16:00:00, containing a total of 10,460,165 transactions. Similarly, the dataset for Western Digital Corporation covers the period from September 28, 2009, at 09:41:53 to September 21, 2022, at 16:00:00, with a total of 69,242,555 transactions. Additionally, we collect data for the ATOM/USDT cryptocurrency pair through the Binance spot API, encompassing the period from June 5, 2019, at 16:35:28 to November 26, 2021, at 12:11:26, with a total of 30,535,634 transactions.

By locating and eliminating outliers in the data gathered, the interquartile range (IQR) approach was used to assess the quality of the data. The data are divided into quartiles using the IQR method, with Q_1 denoting the first quartile, Q_2 the median, and Q_3 the third quartile. Following the calculation of the IQR, the following lower and upper boundaries were established:

$$Q_1 = (N+1) \times \frac{1}{4}, \tag{1}$$

$$Q_2 = (N+1) \times \frac{1}{2}, \tag{2}$$

$$Q_3 = (N+1) \times \frac{3}{4}, \tag{3}$$

$$IQR = Q_3 - Q_1, \tag{4}$$

$$Lower = Q_1 - (1.5 \times IQR), \tag{5}$$

$$Upper = Q_3 + (1.5 \times IQR). \tag{6}$$

Subsequently, IQR is employed to detect and remove outliers using a sliding window approach. As each state is built, the data from the most recent seven days is retained in relation to the current timestamp in a matrix. This is followed by pre-processing to ensure data integrity and reliability. Volatility serves as a statistical indicator, offering insights into the extent of fluctuations in returns for a specific asset over time. This metric quantifies the percentage by which the asset's price varies around its mean value. The calculation of volatility involves utilizing the standard deviation, represented by σ, which measures the dispersion of returns. It is mathematically defined as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N-1}}, \tag{7}$$

where N means the population size, x_i represents the values within the population, and μ denotes the population's mean. A higher value of σ indicates that the data points deviate significantly from the mean, reflecting higher dispersion. Conversely, a lower standard deviation signifies that the data points are closely clustered around the mean.

Dollar bars, in contrast to time bars, do not adhere to a fixed, predefined sampling frequency. Instead, they are generated each time a predefined amount of market value has been exchanged. These bars are instrumental in creating Open-High-Low-Close-Volume (OHLCV) data. Specifically, the open price corresponds to the first transaction price within that interval, the high is the highest transaction price recorded, the low mirrors the lowest price observed, and the close price aligns with the last transaction price. Furthermore, the volume is calculated as the sum of shares exchanged during this interval.

Fig. 5 offers a visual comparison between time bars and dollar bars. Transaction data from the iShares S&P500 ETF, covering the interval from 2009-09-28 09:41 to 2009-09-28 10:32, serves as the basis. Time bars are sampled at a 1-minute frequency, resulting in 52 candlesticks during this period. In contrast, dollar bars are generated whenever the market reaches 500,000 exchanged dollars, leading to 21 candlesticks during the same time frame.

Between 9:41 and 9:51, time bars provide a limited sampling of information during an ascending price movement, failing to capture its strength adequately. In contrast, dollar bars offer a more insightful perspective, as they generate a series of robust green candles. Moreover, from 9:51 to 10:00, time bars tend to oversample information, suggesting a potential price decline. Dollar bars, on the other hand, reveal a weaker retreat based on traded value, resulting in a sideways movement with relatively feeble bars. These dollar bars contribute to a clearer understanding of the intricate relationship between market value and price fluctuations.

Fig. 6(a) illustrates the evolution of the closing price feature of the iShares S&P500 ETF from 2009 to 2022. This asset demonstrates a predominantly favorable trend, marked by a few notable declines in 2016, 2019, and 2020.

The returns, as depicted in Fig. 6(b), exhibit a stationary pattern

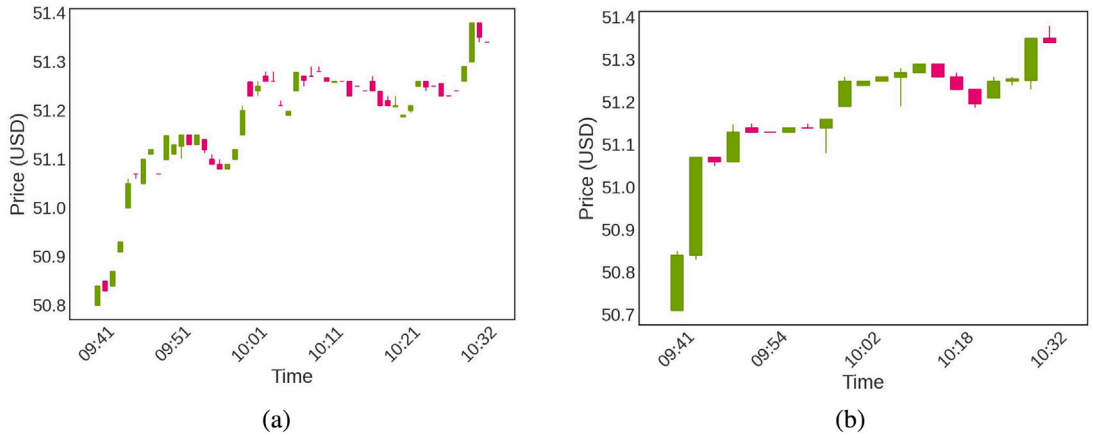


Fig. 5. Comparing the bar sampling approach using data from the iShares S&P500 ETF from 2009-09-28 09:41 to 2009-09-28 10:32: (a) Time bars are sampled every minute, while (b) dollar bars are sampled every 500K market-value exchanges.

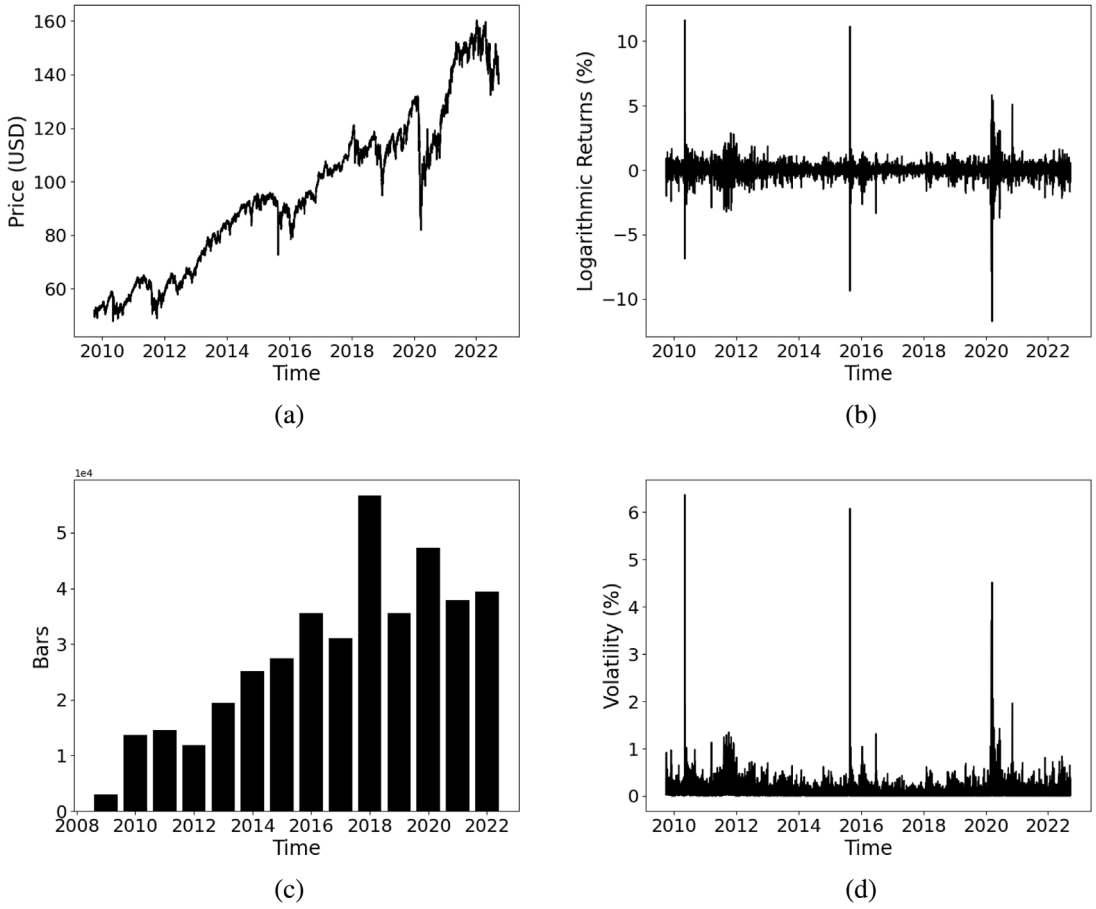


Fig. 6. Time series attributes for the iShares Core S&P 500 ETF from 2009 through 2022: (a) Price series in US dollars, (b) simple logarithmic returns, (c) the number of weekly dollar bars produced annually, and (d) daily volatility.

with occasional spikes observed on both sides of the data frame. Notably, the ETF achieved its highest return of 12.25% in 2010 and its lowest return of −11.13% in 2020, a consequence of the COVID-19 pandemic. Additionally, a significant swing in returns occurred in 2016, ranging from 11.2% to −8.2%. The returns are characterized by a mean of 0.00025%, a variance of 0.012%, and a standard deviation of 0.11%.

Fig. 6(c) provides insight into the total number of weekly sampled bars for each year. Over the course of the study, a total of 398,362 dollar bars and 1,081,690 one-minute time bars were generated. Notably, 2018 witnessed the highest volume of swapped market value, resulting in 56,691 dollar bars. Conversely, 2009 had the lowest volume, with only 2,902 bars. This discrepancy might be attributed to missing data, as the data frame began on 2009-09-28 09:30:00.

Fig. 6(d) illustrates the daily volatility of the iShares S&P500 ETF, represented by rolling standard deviations in a 7-bar window. This volatility metric highlights periods of increased trading opportunities during times of market turbulence. The most significant spikes in volatility occurred in 2010, 2016, and 2020, registering increases of 6.32%, 5.44%, and 4.3%, respectively. However, the ETF also experienced phases of moderate volatility, as observed in 2012 and 2020. It

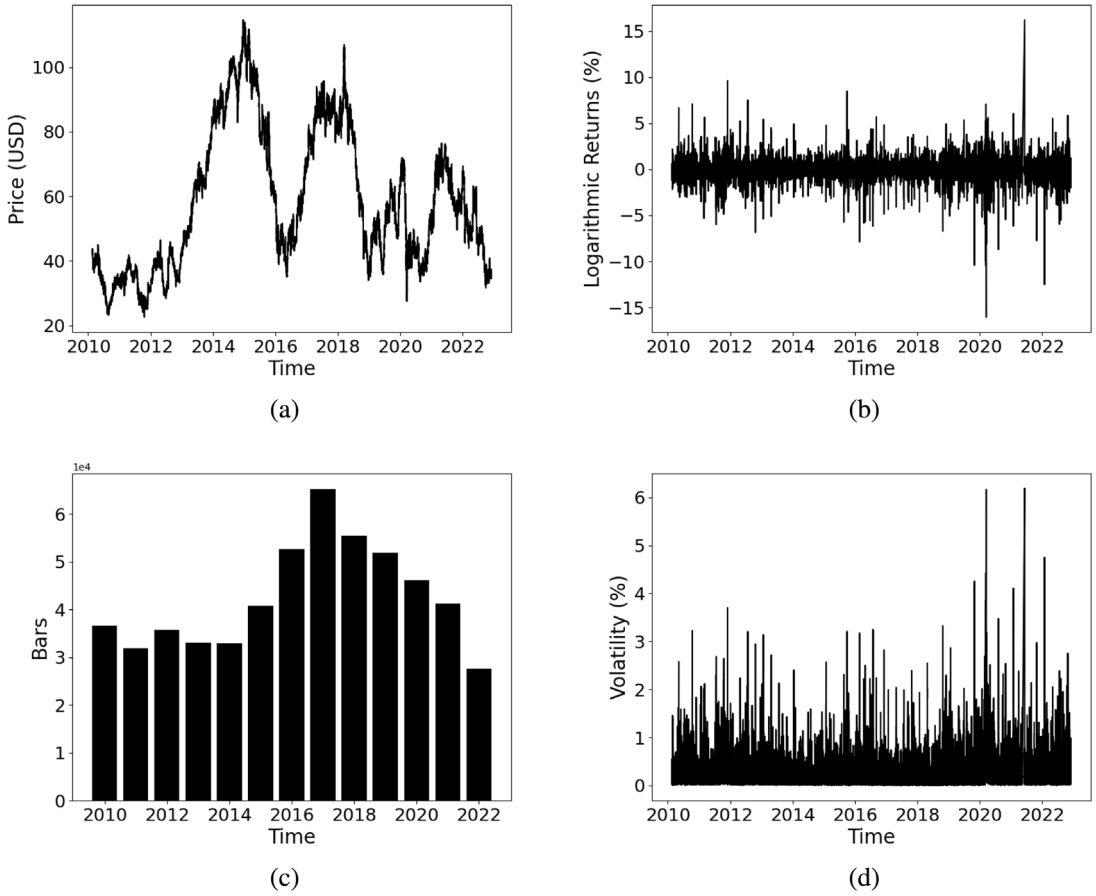


Fig. 7. Time series attributes for Western Digital Corporation from 2010 through 2022: (a) Price series in US dollars, (b) simple logarithmic returns, (c) the number of weekly dollar bars produced annually, and (d) daily volatility.

is worth noting that the overall volatility of the iShares Core S&P 500 ETF is not exceptionally high.

This subsection provides an analysis of the time series attributes derived from Western Digital Corporation data. Fig. 7 visually represents these attributes.

The evolution of the closing price feature from 2010 to 2022 is depicted in Fig. 7(a). Notably, this asset does not exhibit a clear trend but rather behaves akin to an oscillator.

Western Digital's data displays more significant amplitude compared to the preceding data frame. This observation is further corroborated by Fig. 7(b), which illustrates the logarithmic returns. In 2020 and 2021, the asset experiences both-sided returns of 15%, along with fluctuations of −6.2% and −5.9% in 2022 and 2016, respectively. On average, the logarithmic returns have a mean of −0.000035 percent, a standard deviation of 0.21 percent, and a variance of 0.0441 percent.

Fig. 7(c) presents the generated dollar bars per week from 2010 to 2022. The highest number of bars, amounting to 110,186, was recorded in 2020, while the lowest count was 88,450 in 2010. Notably, the Western Digital Corporation data set defines the market for sampling a dollar bar as one million dollars. As evidenced by the increased number of transactions, this asset exhibits better liquidity compared to the previous one, generating 1,305,281 1-minute time bars in contrast to 550,598 bars.

To assess daily volatility, refer to Fig. 7(d). The most volatile periods, with variations nearing 6%, occurred from 2020 to 2022. These surges can be attributed to significant events such as the COVID-19 pandemic and the rise in inflation within the United States (USA).

Fig. 8 illustrates the time series attributes derived from the Binance spot ATOM/USDT cryptocurrency pair. In contrast to the two previously examined assets, the ATOM/USDT price showcases sideways fluctuations in October 2020 and trending movements in July 2021. The asset, as shown in Fig. 8(a), exhibits erratic price changes. Logarithmic returns are depicted in Fig. 8(b), with the most significant spike occurring in 2021 during a -32% dip, followed by a subsequent recovery. Notably, there are a couple of spikes of approximately -17% in April 2020 and some fluctuations of around 10% in the final quarter of 2019.

In the ATOM/USDT data frame, a defined market value quantity of 7.5 K USDT is used to sample a bar. It is evident that the number of time bars exceeds the number of dollar bars by 232.74%. Despite the continuous operation of the cryptocurrency asset, it offers a similar volume of data compared to the previously analyzed data frames. Fig. 8(c) clearly shows an exponential increase in the number of created bars, with the highest number of time bars recorded in 2021 and the lowest in 2019, producing 475,388 and 259,249 bars, respectively.

The asset's volatility plot is presented in Fig. 8(d), highlighting its status as the most volatile, with a peak of 25% in May 2021. Additional instances of volatility between 6% and 12% are observed in April 2020. In terms of the overall trend, volatility rises in the second half of 2019 and gradually declines starting in 2021. Subsequently, the asset's volatility remains relatively stable, with occasional spikes ranging from 2% to 4%.

To compare the distributions of the three time series, we utilize the standard deviation, denoted as σ. For the iShares Core S&P 500 ETF, the standard deviations of sampled dollar and time bar counts are 413.39 and 218.97, respectively. In the case of the second time series, these values are 368.24 and 271.97, respectively, indicating that the counts of time bars exhibit greater consistency and are closer to the mean compared to dollar bars. Lastly, for the cryptocurrency ATOM/USDT, the standard deviations are 6826.89 and 1375.11, respectively. Since

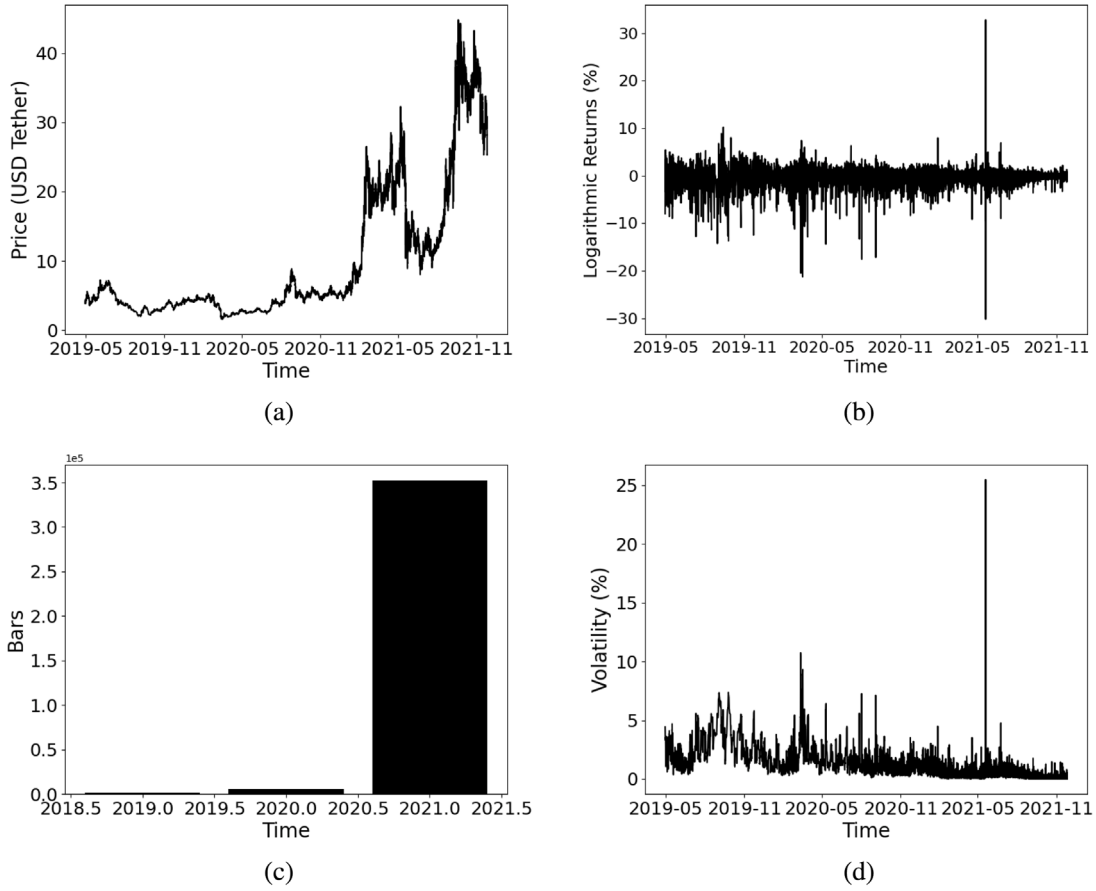


Fig. 8. Time series attributes for ATOM/USDT from 2019 through 2021: (a) Price series in US dollars, (b) simple logarithmic returns, (c) the number of weekly dollar bars produced annually, and (d) daily volatility.

the dollar type is sampled at a constant frequency, time bars are expected to be more stable than the dollar type.

4. Methods

This section describes the fundamental concepts of the Markov decision process (MDP) and RL. We then dive into an extensive exploration of distributional learning variation within the realm of RL. These variants offer a crucial perspective by not only accounting for the expected value of future rewards but also embracing the entirety of potential outcome distributions. Furthermore, we introduce the ETDQN method, a novel approach proposed in this study, and provide an overview of the evaluation metrics crafted to gauge the models' performance.

4.1. RL and distributional learning

In RL, an agent interacts with the environment to make decisions within a formal framework. The foundation of RL lies in the Markov Decision Process (MDP) (Bellman, 1957), which serves as a discrete-time stochastic control framework for decision-making problems. This framework comprises a 5-tuple $(S, A, P(\cdot,\cdot), R(\cdot,\cdot), \gamma)$, with each element holding a distinct role.

Firstly, S represents the set of states, often referred to as the state space. Correspondingly, A stands for the set of available actions denoted as the action space. The transition probabilities are captured by $P(\cdot,\cdot)$, signifying the probability that taking action a in state s at time t will lead to state s' at time $t+1$ ($P(s_{t+1} = s' \mid s_t = s, a_t = a)$).

Additionally, $R(\cdot,\cdot)$ represents the immediate reward when transitioning from state s to state s' due to action a. The discount factor, denoted as γ and falling within the interval $[0,1]$, defines the agent's importance on future rewards. Finally, π is a function mapping from the state space (S) to the action space (A).

In this framework, the primary objective for the agent is to maximize the expected return, denoted as R_t. This return is calculated through the state–action value function $Q^{\pi}(s,a) = \mathrm{E}[R_t|(s_t,a_t)]$, where R_t signifies the expected return at time step t, considering the state s_t and action a_t.

DQN approximates the optimal value function of the action by applying time difference methods. This approximation is optimized recursively at each time step, denoted as t, according to the Bellman equation described by:

$$Q_{t+1}(s,a) = \mathbb{E}[R_t + \gamma \underset{a' \in A}{max} Q_t(s',a')|s,a], \tag{8}$$

where $Q(s,a)$ at time-step $t+1$ is the expected value of the reward, R, t is time-step, a is the result of taking action, s is the state, $Q_t(s',a')$ is state–action value, and γ is discount factor.

Bellemare et al. (2017) suggests approximating the distribution of returns rather than focusing solely on expectations. This involves modeling these distributions as probability masses positioned on a discrete support vector $\boldsymbol{z}$. This vector is parameterized by a positive integer, $N_{atoms} \in \mathbb{N}^+$, defined as:

$$z_i = V_{min} + i\Delta z, \tag{9}$$

$$\Delta z = \frac{V_{max} - V_{min}}{N_{atoms} - 1}, \tag{10}$$

where V_{min} and V_{max} are maximum and minimum values, respectively, within the support vector $\boldsymbol{z}$, and N_{atoms} denotes the count of atoms that constitute the canonical returns of the distribution.

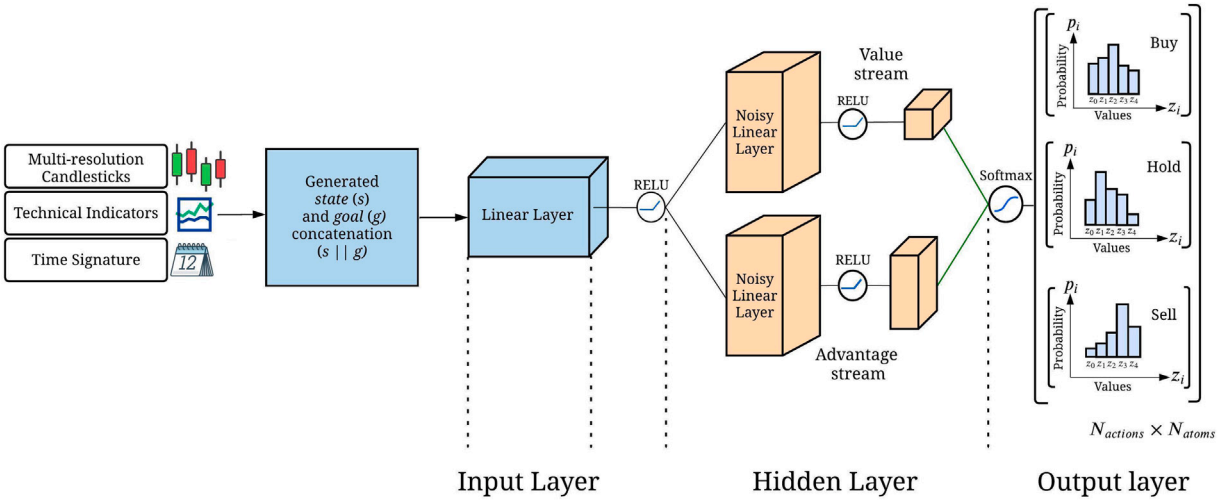


Fig. 9. Deep Dueling Feed-Forward Q-Neural Network used to solve the trading problem.

In this context, an approximated distribution, d_t, observed at time-step t, is defined on the support vector $\boldsymbol{z}$ and contains the probability mass $p^i_\theta(s,a)$ associated with each atom i. The primary objective is to update the weights denoted as θ to align d_t with the actual distribution of returns. A key insight is that the return distributions should adhere to a modified form of Bellman's equation. Specifically, for a given state s_t and action a_t, the distribution of returns, when following the optimal policy π^*, should align with a target distribution. This target distribution is established by taking the distribution corresponding to the subsequent state s_{t+1} and action $a_{t+1} = \pi(s_{t+1})$, scaling it down towards zero through multiplication by the discount factor γ, and shifting it by the immediate reward r_t. A distributional variant of the DQN can be derived by constructing a support vector $\boldsymbol{z}$ at each time-step denoted as t and then minimizing the Kullback–Leibler divergence D_{KL} between the distribution d_t and a target distribution. This process is expressed as:

$$d'_t = (R_{t+1} + \gamma \boldsymbol{z}, \boldsymbol{p}^-_\theta(S_{t+1}, \overline{a}^*_{t+1})), D_{KL}(\Phi_{\boldsymbol{z}} d'_t \parallel d_t), \tag{11}$$

where Φ is an L2-projection of the target distribution onto the fixed support vector $\boldsymbol{z}$, R_{t+1} is the reward received at time-step $t+1$, γ is the discount factor defined in the MDP, and $\overline{a}^*_{t+1}$ is defined as $\underset{a \in A}{\text{argmax}} q_{\overline{\theta}}(s_{t+1}, a)$, is the action that maximizes the mean action values $q_{\overline{\theta}}(s_{t+1}, a) = \boldsymbol{z}^\top \boldsymbol{p}_\theta(s_{t+1}, a)$ in state s_{t+1}. The parameterized distribution can also employ a frozen copy denoted as $\overline{\theta}$ of the neural network parameters, θ, to construct the target distribution.

4.2. ETDQN

This study employs a neural network architecture that consists of two deep feed-forward artificial neural networks: the target Q-network and the online Q-network. The agent makes decisions with the goal of maximizing its overall payoff, guided by a system of incentives and penalties.

The target Q-network is essentially a duplicate of the online Q-network. However, it differs in that its weights are not updated after every time step but rather at the conclusion of a trade. In contrast to the original DQN implementation, which relies on image inputs (Mnih et al., 2015), our approach utilizes non-image inputs. Consequently, we replace convolutional and pooling layers with linear layers in our neural network architecture. Fig. 9 illustrates a visual representation of the inputs and outputs within this artificial neural network structure.

Both neural network architectures consist of three main layers: Input, hidden, and output. The dimensions and node count in the input layer are tailored to accommodate the state-goal pair $s \parallel g$, with $\parallel$ representing concatenation. The input layer involves a standard linear layer, followed by a Rectified Linear Unit (ReLU) activation function defined as:

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.}, \end{cases} \tag{12}$$

where $f(x)$ represents the function's output, and x denotes the input signal.

The hidden layer of our neural network adopts the dueling network architecture, as proposed by Wang et al. (2016). This architecture comprises two noisy linear stream layers, as introduced by Fortunato et al. (2018). The dueling network is designed to enable separate estimations of the value and advantage functions. Specifically, the value stream outputs the state-goal value function $V_\pi(s,g)$ with a dimension of N_{atoms}, while the advantage stream generates the advantage of actions within the state-goal pair $s||g$ with dimensions of $N_{actions} \times N_{atoms}$. Notably, we replace the traditional exploration heuristic, the ϵ-greedy approach, by incorporating noisy linear layers that introduce parametric noise to the weights of the network. These learned perturbations in network weights enhance exploration by injecting factorized Gaussian noise σ_0, resulting in improved exploration strategies.

The advantage and value streams are then combined, followed by a Softmax activation function that transforms values into probabilities. The output layer consists of three probability mass functions, each corresponding to one of the available actions: Buy, sell, and hold. This output layer exhibits a dimension of $N_{action} \times N_{atoms}$.

In the context of state construction, we initiate the process by aggregating tick data, which records asset transactions and prices, into groups until they collectively reach a predefined market value. Subsequently, we generate what we refer to as a dollar bar. For detailed insights into this procedure, please refer to Section 3 and consult de Prado (2018).

Each dollar bar encapsulates OHLCV price data. Moving forward, we extract the current date and time from the generated bar. We then employ the Interquartile Range (IQR) method, taking into account the last seven days based on the date, to identify and remove outliers. Following data cleaning, we proceed to assemble the state.

The state, determined by the current date and time, seeks the previous LOOKBACK amount of data samples for each feature and appends them accordingly. To ensure that no single high-valued variable dominates the neural network, we normalize all the features. In addition to OHLCV price dollar bars, we derive various features, including technical indicators and time signature attributes.

In terms of technical indicators, we calculate the Relative Strength Index (RSI) over a 14-day period, which is based on closing prices and

measures asset strength. The Balance of Power indicator assesses the influence of buyers and sellers by evaluating significant price changes. Furthermore, the Aroon oscillator helps identify trends over the last 25 periods. To complement this, we utilize the Moving Average Convergence/Divergence (MACD) technical indicator, which incorporates 12-day EMA (Exponential Moving Average), 26-day EMA, and 9-day EMA data.

Time signature features encompass the current time of day and week. These attributes can have a substantial impact on profitability, providing additional information beyond conventional numerical timestamps, particularly for events recurring at regular weekly intervals. The algorithm captures the current index position, extracts the present day, hour, and minute, and maps time into a floating-point number, normalizing it to represent the time of day. Similarly, the day of the week is mapped to an integer value and also subjected to normalization for inclusion in the state.

In this study, we employ the exponential profit and loss function as our primary reward mechanism. This function is computed at each time step, denoted as t, by following a specific formula. It takes into account various factors, such as the current asset price p_t, transaction costs t_c incurred during both trade entry and exit, and the entry price p_0. All of these variables are normalized by dividing them by the entry price p_0 and then further adjusted for transaction costs. Additionally, the result is multiplied by the position taken in the trade, with a value of 1 for long positions and −1 for short positions. Finally, the natural exponent of this value is calculated to yield the reward. Eq. (13) represents this reward function:

$$R_t = e^{\frac{p_t(1-t_c)-p_0(1+t_c)}{p_0(1+t_c)} \times position}. \tag{13}$$

The selection of the exponential profit and loss as the reward function is grounded in the principle that the agent should receive a higher reward for achieving greater returns. At each time step, the agent assesses whether the trade has concluded. If it has, the agent receives the corresponding reward. Conversely, if the trade remains ongoing, the agent proceeds with its action without receiving any reward.

The backtesting process for the proposed ETDQN model comprises two primary phases: Sampling and learning. During the sampling phase, the agent interacts with the environment, gathers experience tuples, and then samples these tuples in batches for subsequent processing in the learning phase, with priority considerations. The learning phase involves taking the batch of experience tuples derived from the sampling phase, calculating the loss function, and conducting a backward pass to update the neural network weights based on what the agent has learned from the loss function. Fig. 10 provides an overview of the sampling process within this framework.

The algorithm follows a structured process where it samples state-goal pairs from a distribution $p(s, g)$, where states are denoted as $s \in S$ and goals as $g \in G$. Subsequently, it selects actions $a \in A$ based on the policy $\pi : S \times G \rightarrow A$, computes the associated reward r, and stores these experiences as tuples. Each experience is represented as a state-goal pair $s||g$, an action a, a reward r, and the next-state goal pair $s||g$ $(s||g, a, r, s'||g)$.

Through empirical experimentation, we introduce additional goals, as per Andrychowicz et al. (2017), by randomly selecting a state to concatenate with the original goal. When an extra goal g' is introduced, a new experience tuple $(s||g', a, r, s'||g')$ is stored, sharing identical values except for the goal g'.

During each iteration i, priority values p_i are recorded to prioritize the most critical samples based on the loss function $L(\theta)$. Following the principles outlined in Schaul et al. (2016), the compensation factor β undergoes a linear increase to correct the bias introduced by the sampling proportion linked to the temporal difference error.

Furthermore, the algorithm updates the weights of the target network at the trade liquidation stage. Fig. 11 provides a visual representation of the comprehensive learning process.

During the learning phase, exploration takes place by introducing noise into the weights denoted as θ of the online Q-network, as proposed by Fortunato et al. (2018). This noise injection is initiated with an initial standard deviation value σ_0 within the range of (0,1]. The online Q-network processes input in the form of the next state-goal pair $s' \parallel g$ and subsequently generates action-value distributions $\boldsymbol{p}_\theta((s' \parallel g), a)$ for each of the available actions (buy, hold, and sell).

Subsequently, this same network calculates the Q-values associated with each action by computing the sum of each element i in the vector $\boldsymbol{z}$ multiplied by its corresponding probability value $p^i_\theta((s' \parallel g), a)$. The network then stores the action associated with the highest Q-value, which is the greedy action denoted as a^*.

Following this, the target Q-network, whose weights remain frozen until the trade is liquidated, assesses the greedy action a^* (Hasselt et al., 2016). This network, equipped with weights denoted as $\bar{\theta}$, also takes the next state-goal pair $s' \parallel g$ as input. However, instead of selecting an action, it retrieves the probability mass distribution of the greedy action a^* generated by the online Q-network and evaluates it. In equation terms, this is denoted as $\boldsymbol{p}_{\bar{\theta}}((s' \parallel g), a^*)$.

Following the approach detailed in Bellemare et al. (2017), a discrete-valued support vector $\boldsymbol{z}$ is created. This vector contains evenly spaced elements within the defined boundaries (V_{min} and V_{max}) to facilitate the application of the Bellman update, as described in the paper. For each value j in the range of 0 to $N_{atoms} - 1$, the update $\hat{\mathcal{T}} z_j$ is applied to each element z_j by displacing the support through the multiplication by the discount factor γ and the addition of the reward r. Finally, the projected distribution is clipped to ensure it remains within the established boundaries (V_{min} and V_{max}).

The $\hat{\mathcal{T}} z_j$ update is used to compute the nearest probability mass neighbors of $\boldsymbol{p}_{\bar{\theta}}((s' \parallel g), a^*)$, where m_l represents the lower distribution probability neighbor, and m_u represents the upper distribution probability neighbor.

In conclusion, the cross-entropy loss $L(\theta)$ is calculated between the value distribution generated by the online Q-Network, considering the state-goal pair $s \parallel g$ and the action a, denoted as $\boldsymbol{p}_\theta((s \parallel g), a)$, and the shifted distribution m_i at value i using the following equation:

$$L(\theta) = -\sum_i m_i \log p^i_\theta((s \parallel g), a). \tag{14}$$

This equation entails the summation over each iteration i of the projected distribution m_i, which is then multiplied by the natural logarithm of the predicted probability mass distribution $p^i_\theta((s \parallel g), a)$. This prediction is generated by the online Q-network with weights θ and obtained through the state-goal pair $s||g$ and action a.

4.3. Evaluation metrics

The evaluation metrics employed in this paper encompass the Sharpe ratio, simple returns, cumulative returns, and maximum drawdown.

Simple returns, which quantify the gains or losses of an investment irrespective of time, are defined by Arratia (2014). The τ-period simple return at time t, denoted as $R_t(\tau)$, represents the rate of change in the asset's price when held from time $t-\tau$ to time t. The formula for simple returns is as follows:

$$R_t(\tau) = \frac{P_t - P_{t-\tau}}{P_{t-\tau}} = \frac{P_t}{P_{t-\tau}} - 1, \tag{15}$$

where P_t signifies the asset's price at time step t within a time scale of τ.

Cumulative Returns, also known as multi-period returns, illustrate the investment's historical evolution. Arratia (2014) defines the τ-period simple return $R_t(\tau)$ at time t as the product of τ one-period

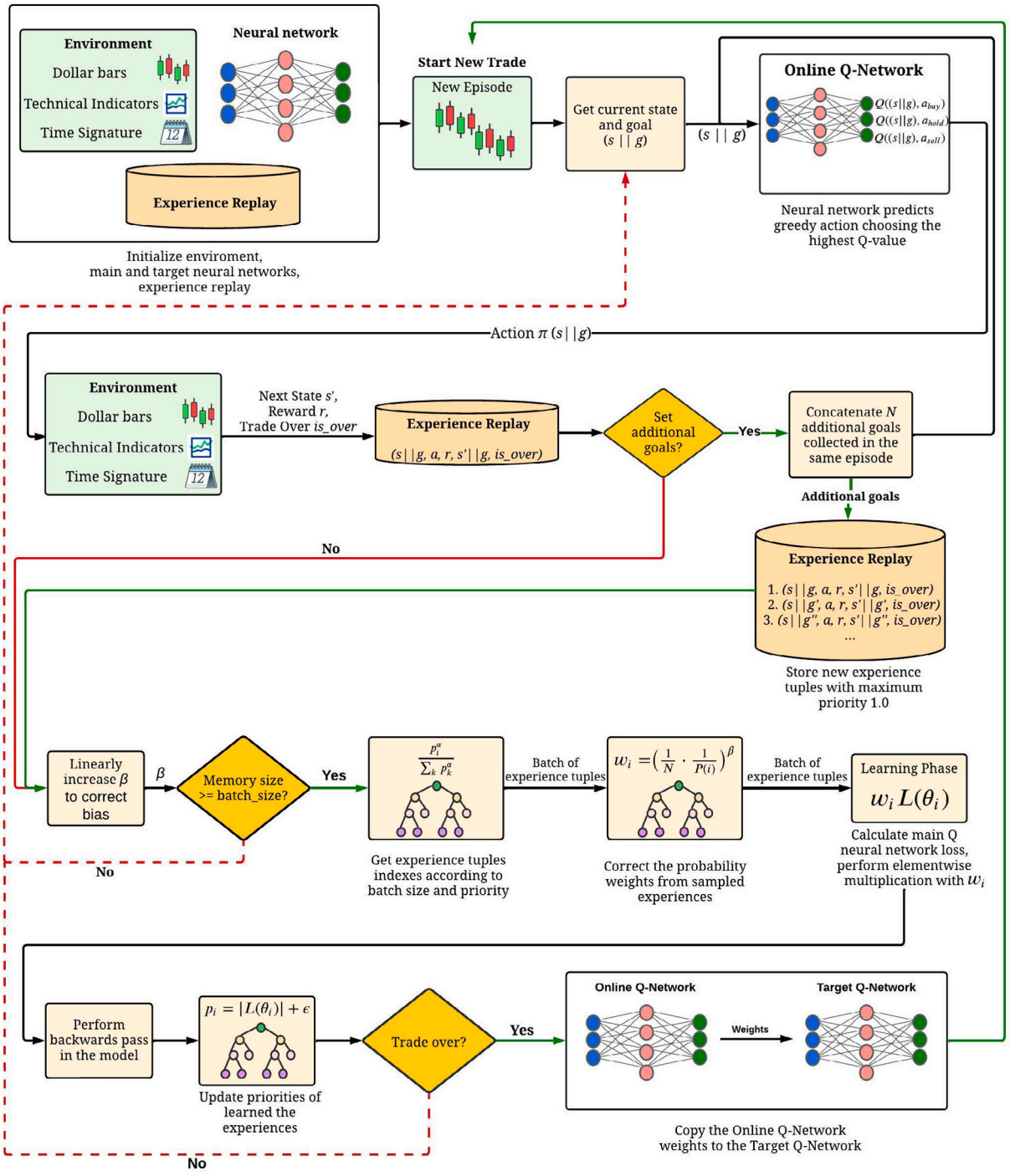


Fig. 10. Backtesting overview of the ETDQN strategy.

simple gross returns spanning from times $t-\theta+1$ to t. The proof of this proposition is provided in Eq. (16):

$$R_t(\tau)+1 = \frac{P_t}{P_{t-\tau}} = \frac{P_t}{P_{t-1}} \cdot \frac{P_{t-1}}{P_{t-2}} \cdots \frac{P_{t-\tau+1}}{P_{t-\tau}} = (1+R_t)\cdot(1+R_{t-1}) \cdots (1+R_{t-\tau+1}). \tag{16}$$

The Sharpe ratio assesses the risk-adjusted return of a financial portfolio. Proposed by Sharpe (1994) in 1966, this ratio condenses risk and reward into a single number. Eq. (17) defines the Sharpe ratio as:

$$S_a = \frac{\mathbb{E}[R_a - R_b]}{\sigma_a}, \tag{17}$$

where it measures the quality of an investment as the expected value ($\mathbb{E}$) of the asset return R_a minus the risk-free return R_b, divided by the standard deviation of the asset σ_a. A lower standard deviation signifies less risk and a higher Sharpe ratio, with the opposite indicating higher risk.

Maximum drawdown (MDD) represents the largest peak-to-valley decline (often expressed as a percentage) of an investment during a specific holding period. Drawdowns aid in assessing an investment's financial loss risk based on historical data. Choi (2021) defines the maximum drawdown as:

$$MDD(T) = \max_{\tau \in (0,T)} \left[\max_{t \in (0,\tau)} (P(t) - P(\tau)) \right], \tag{18}$$

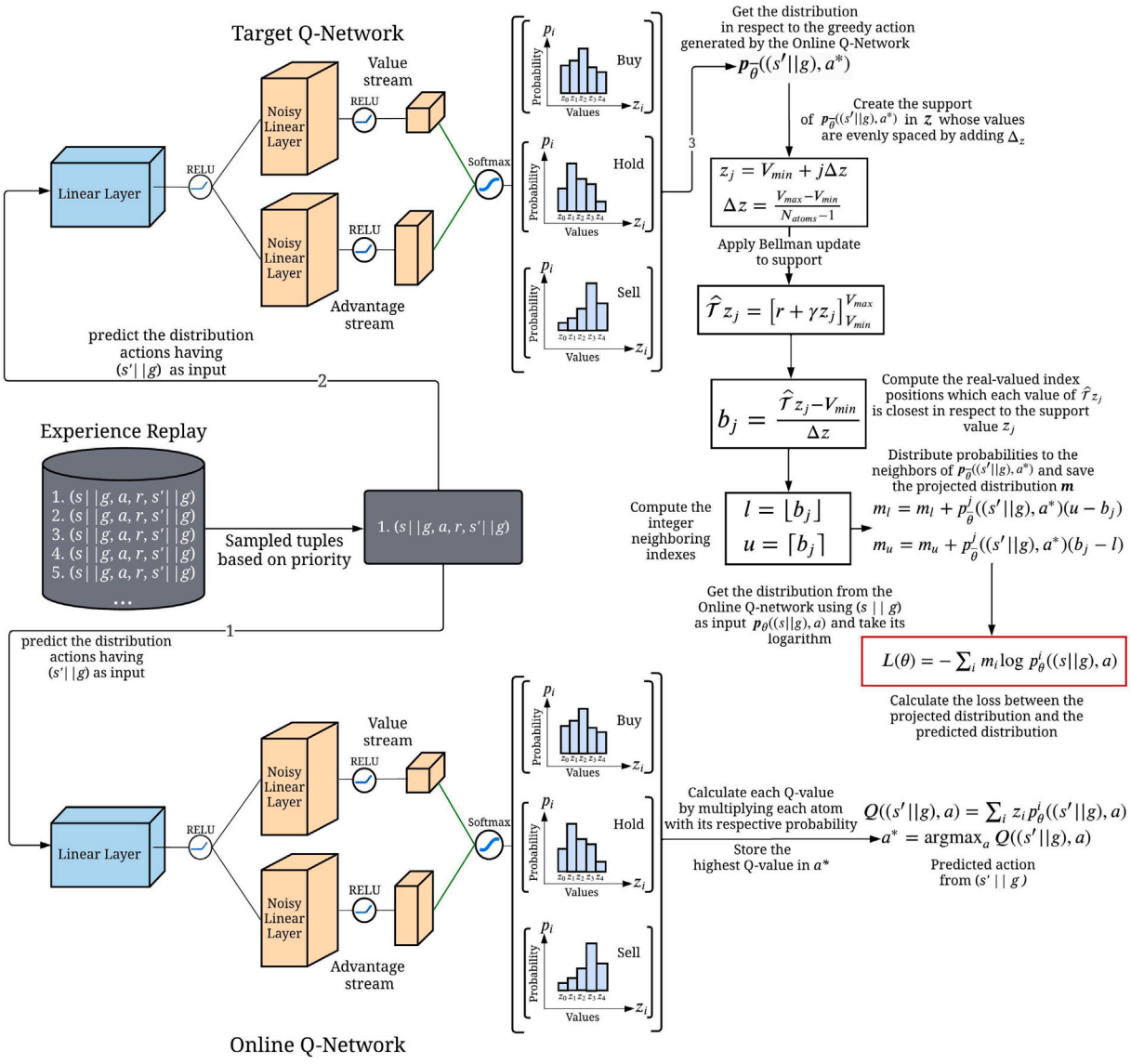


Fig. 11. ETDQN adopted learning process.

where the inner part calculates drawdowns by taking the maximum difference between the logarithmic-price $P(t)$ at time t over the range from 0 to τ, and $P(\tau)$, the logarithmic-price at period τ. The outer part selects the maximum drawdown among these values.

5. Results and discussion

To evaluate and compare performance, metrics for the benchmark and ETDQN were analyzed, and daily cumulative returns, Sharpe ratios over six months, maximum drawdowns, and monthly and annual returns were calculated. Transaction charges, represented as t_c, were taken into account, with rates of 0.03% for cryptocurrencies based on the ATOM/USDT protocol and 0.18% for other assets.

In this study, version 3.10 of the Python programming language was used for the experiments. It was decided to use the deep learning library (Paszke et al., 2019). The DRL models were trained and tested using an Nvidia GeForce RTX-2080 Super graphics processing unit with 32 GB of random access memory and an Intel Core™ i9-10900X central processor unit. The trials were conducted using the Linux operating system Ubuntu 22.04.3.

5.1. Hyperparameters

The optimized set of hyperparameters required for tuning the ETDQN model are presented in Table 3 for each time series employed in this study.

Due to the vast number of ETDQN hyperparameters, only a small number of modifications were made based on earlier methods that used DQN variants such as Andrychowicz et al. (2017), Bellemare et al. (2017), Fortunato et al. (2018), Hessel et al. (2018), Schaul et al. (2016). Using manual coordinate descent, the hyperparameters that were the most sensitive were changed. The learning rate optimizer was Adaptive Moment Estimation (Adam) (Hessel et al., 2018). Compared to stochastic gradient descent and root mean square propagation, Adam is less sensitive. This study evaluated variations of the original DQN, which employed a learning rate of $\alpha' = 0.00025$, such as $\{\alpha'/2, \alpha'/4, \alpha'/6\}$. The second variation was chosen since it resulted in bigger cumulative rewards, which gave Adam a learning rate of 0.000625. The ReLUs were employed as activation functions, except for the final layer, which was incorporated the Softmax function (Bellemare et al., 2017). The loss function for the optimizer of the artificial neural network was the cross-entropy.

Table 3
Hyperparameters values used in ETDQN for each time series.

Hyperparameters	Assets		
	iShares Cores S&P500 ETF	Western Digital Corporation	ATOMUSDT cryptocurrency
Learning Rate	0.0000625	0.0000625	0.00000425
Loss Optimizer	Adam	Adam	Adam
Loss Function	Cross-entropy	Cross-entropy	Cross-entropy
Activation Function	Rectified Linear Unit	Rectified Linear Unit	Rectified Linear Unit
Lookback	20	20	20
Batch Size	8	16	8
Discount Factor (γ)	0.99	0.99	0.99
Memory Size	500	500	600
Additional Goals	5	3	4
Initial Standard Deviation (σ_0)	0.5	0.8	0.3
Experiences prioritization (α)	0.5	0.5	0.5
Prioritization bias compensation (β)	0.4 → 1.0	0.4 → 1.0	0.4 → 1.0
Distributional atoms (N_{atoms})	51	51	51
Distributional min/max values (V_{min}/V_{max})	[−20, 100]	[−20, 100]	[−20, 100]
Market value sampling threshold	500K USD	1M USD	7.5K USD

In terms of exploration, the standard deviation for perturbing the noise of the artificial neural networks is 0.5, 0.8, and 0.3, respectively, with lower values for more volatile assets, as suggested by Fortunato et al. (2018). Furthermore, the number of atoms in the distribution was 51 (Bellemare et al., 2017). However, the distribution's allowable V_{min} and V_{max} values range from −20 to 100. These values are determined through an evaluation of various setups, including $[-10, 10]$, $[-20, 100]$, $[0, 100]$, $[-100, 100]$, $[-20, 20]$.

The Lookback hyperparameter undergoes evaluation among {10, 20, 30, 60}. According to the study, 20 prior observations produced the highest overall benefits. Notably, the input to the model is greatly increased by this parameter, which is also very sensitive. The algorithm has trouble spotting patterns when given 60 past numbers as input, a behavior also seen for the lower value. As a result, the Lookback option is directly impacted by the lot size, with higher values increasing the data input. Candidates for the lot size are {8, 16, 32, 64}, but after analyzing the all-time series, the lot size was determined to be no greater than 16. Similar to the Lookback parameter, a larger lot size prevented the model from converging to a desirable policy. The γ discount factor for the MDP is set at 0.99. This decision was made with the intention of encouraging the algorithm to place more value on long-term learning than on short-term incentives. Alternative values, such as 0.91, 0.93, and 0.96, were also successfully used. However, the greatest overall results are typically obtained when $\gamma = 0.99$.

Although various values were considered, 500 samples consistently produced the best cumulative rewards in conjunction with the aforementioned lot size, with the exception of the ATOM/USDT time series. Regarding the repetition prioritization, the proportional variant presented in Schaul et al. (2016) was chosen. Involving a linear increase in the importance sampling exponent β from 0.4 to 1 during training, together with an experience prioritization exponent $\alpha = 0.5$. Finally, a maximum of 5 additional goals were added concatenated with each experience tuple, improving exploration and learning (Andrychowicz et al., 2017), but others were tested. The optimal amount of additional goals varied for each time series, as determined by experiments.

5.2. Cumulative returns

This section presents the portfolio's historical development through measurements of cumulative returns, providing insight beyond its most recent assessment. Fig. 12(a) illustrates the daily cumulative returns for the five models employed in analyzing data from the iShares S&P500 ETF, Western Digital Corporation, and ATOM/USDT data sets.

In Fig. 12(a), ETDQN consistently outperforms other models in terms of cumulative returns for the iShares S&P500 ETF. With a mean of 3.86, a peak of 8.61, and a minimum of 1, the model exhibits consistent performance from 2010 to 2022. It notably achieves a remarkable 164% profit in March 2020. TDQN, on the other hand, demonstrates a mean of 2.64, reaching a maximum of 4.92 but struggling to capitalize on market downturns. The B&H model maintains a mean of 2.06, with a maximum of 3.01, but experiencing occasional setbacks. The S&H model delivers poor performance with a mean of 0.23, while the Random Action model achieves a mean of 0.96, maintaining relative stability.

Fig. 12(b) showcases cumulative returns for various strategies applied to Western Digital Corporation from 2010 to 2022. ETDQN consistently outperforms all benchmarks, except for a dip in 2012 when it reached its lowest point of 0.93. However, it rapidly recovers and surpasses other benchmarks, eventually doubling its initial investment. TDQN records a minimum value of 0.87, a maximum of 6.12, and a mean of 2.40, securing the second-best performance. The B&H model achieves a minimum of 0.78 in 2012, a maximum of 3.86 in 2015, and a mean of 2.13, securing the third position. The S&H model remains in fifth place, with a minimum evaluation of 0.05 in 2022, a maximum of 1.06 in 2010, and a mean of 0.23. Lastly, the Random Action strategy claims the fourth position, with a minimum value of 0.45.

Fig. 12(c) illustrates cumulative returns for various strategies applied to ATOM/USDT from 2019 to 2022. ETDQN consistently outperforms all benchmarks, achieving a minimum value of 0.95 in August 2019 and a maximum value of 37.62 in November 2021. With a mean portfolio performance of 12.08, the strategy gradually increases its returns over time. TDQN exhibits a minimum cumulative return of 0.26, a maximum of 6.12, and a mean of 2.40, initially facing challenges but learning from experiences to outperform both the B&H and S&H models. The B&H model secures the second position, with a minimum of 0.29 in August 2019 and a maximum of 7.67 in August 2021, generating profits during the cryptocurrency market's Bullrun. The S&H model ranks fourth, while the Random Action strategy demonstrates improved performance, particularly between January and April 2021.

5.3. Sharpe ratio

It is worth mentioning that this section analyzes this metric over a continuous 6-month period using a window value of 126 days, accounting for working days, which results in an average of 21 days per month. To summarize the information regarding the 6-month rolling Sharpe ratio, Table 4 presents the minimum, mean, and maximum values for the ETDQN, TDQN, B&H, S&H, and Random Action models in relation to the iShares S&P 500 ETF, Western Digital Corporation, and Binance spot ATOM/USDT cryptocurrency data.

As observed in Table 4, ETDQN consistently outperforms the TDQN, B&H, S&H, and Random Action benchmarks in terms of mean Sharpe ratios for the iShares S&P500 ETF, with improvements of 1.36, 1.01, 2.74, and 1.4, respectively. The policy implemented by ETDQN secures the top position in this performance metric. Furthermore, when analyzing the risk-return relationship for Western Digital Corporation, the

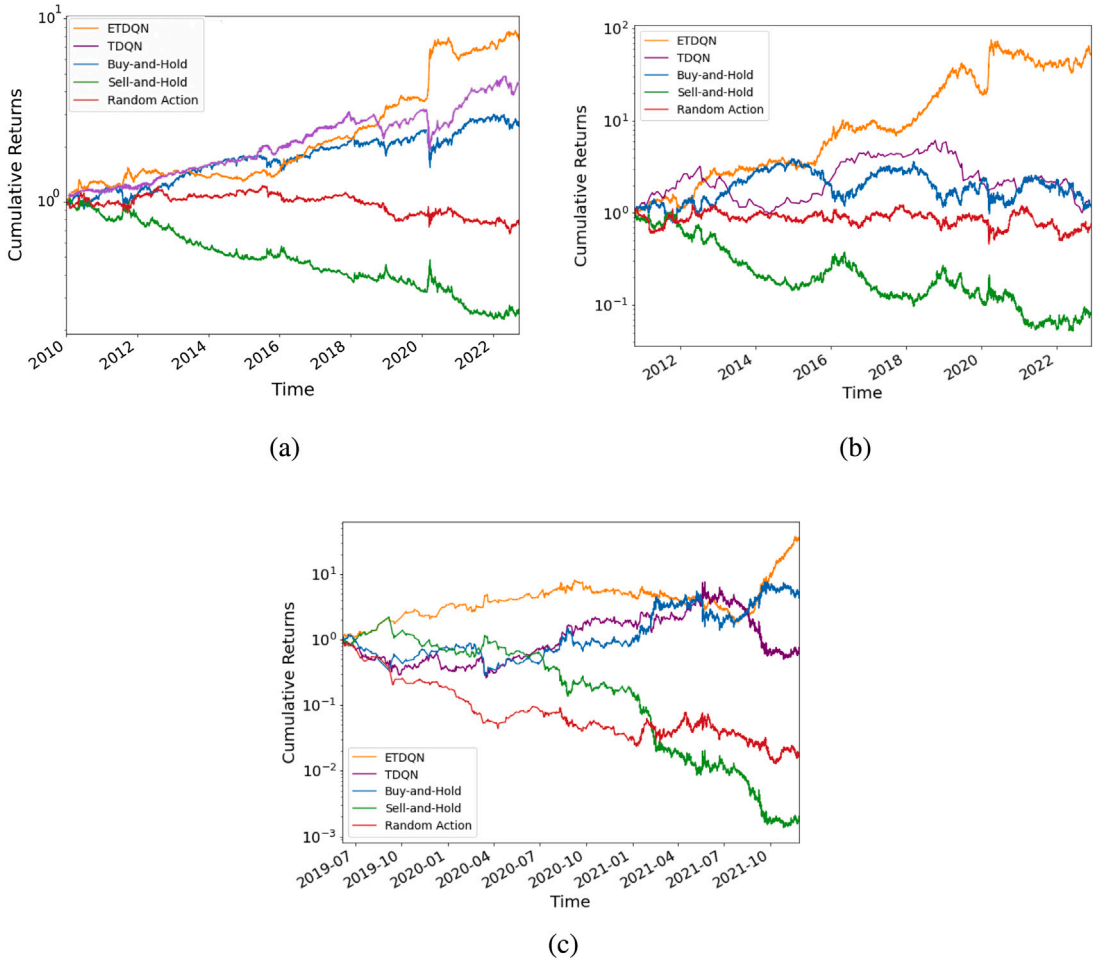


Fig. 12. Daily cumulative returns to the data set from (a) iShares S&P 500 ETF, (b) Western Digital Corporation, and (c) ATOM/USDT cryptocurrency comparing 5 models.

Table 4
Statistical metrics of the 6-month rolling Sharpe ratio applied to assets.

Assets	Models	Minimum	Mean	Maximum
iShares S&P500 ETF	ETDQN	−3.26	**1.04**	**6.07**
	TDQN	**−2.66**	1.03	4.56
	B&H	−2.97	0.76	4.6
	S&H	−4.6	−0.76	2.97
	Random	−2.95	0.0	2.81
Western Digital Corporation	ETDQN	−4.64	**1.00**	**6.57**
	TDQN	**−3.43**	0.3	2.76
	B&H	−4.55	0.3	3.51
	S&H	−3.51	−0.3	4.55
	Random	−3.62	0.13	4.02
ATOM/USDT cryptocurrency	ETDQN	−2.24	1.02	**7.3**
	TDQN	−3.18	0.86	3.88
	B&H	**−1.55**	**1.19**	3.65
	S&H	−3.65	−1.19	1.55
	Random	−4.66	−0.82	2.44

same policy exhibits significantly better results, with improvements of 3.3, 3.3, 6.6, and 7.7. However, in the context of the ATOM/USDT market, the strategy generated by the agent does not rank first in terms of the mean Sharpe ratio, despite achieving a higher maximum value than any of the benchmarks. This discrepancy could be attributed to insufficient data generated at the beginning of the data frame and the asset's volatility.

5.4. Maximum drawdown

Table 5 provides the description of the maximum drawdown (%) metric for all models applied in this study across the data sets. Among the models analyzed in the iShares S&P500 ETF, ETDQN exhibits the lowest maximum drawdown, indicating less risk compared to TDQN, B&H, S&H, and Random Action. In 2021, ETDQN experiences a maximum drawdown of −24.40%, while TDQN and B&H face their maximum drawdowns of −35.1% and −56.60% in March 2020, respectively. ETDQN also demonstrates the lowest maximum drawdown for Western Digital Corporation. During the COVID-19 event, both ETDQN and B&H encounter their maximum drawdowns. In the case of the ATOM/USDT cryptocurrency pair, B&H exhibits the smallest maximum drawdown among the strategies, closely followed by ETDQN. The high volatility of ATOM/USDT results in significant drawdowns, with ETDQN experiencing a substantial −76.70% decline in July 2021, and B&H facing a notable -74% maximum decline in April 2020.

5.5. Monthly returns

Fig. 13 depicts monthly returns heatmaps generated by the proposed ETDQN model applied to the iShares S&P500 ETF, Western Digital Corporation, and ATOM/USDT data sets.

As shown in Fig. 13(a), the ETDQN model did not yield significant profits in 2009 due to the limited data available. However, in 2010, the model recorded eight months of positive returns, with the best performance observed in March. August 2011 saw a notable return of

Table 5
Maximum drawdown (%) applying five models for assets between 2009 to 2022.

Models	iShares S&P500 ETF	Western Digital Corporation	ATOM/USDT cryptocurrency
ETDQN	**−24.4**	**−56.6**	−76.7
TDQN	−35.1	−90.9	−83.5
B&H	−56.6	−74.5	**−74.0**
S&H	−78.0	−95.0	−99.9
Random	−47.5	−65.4	−94.0

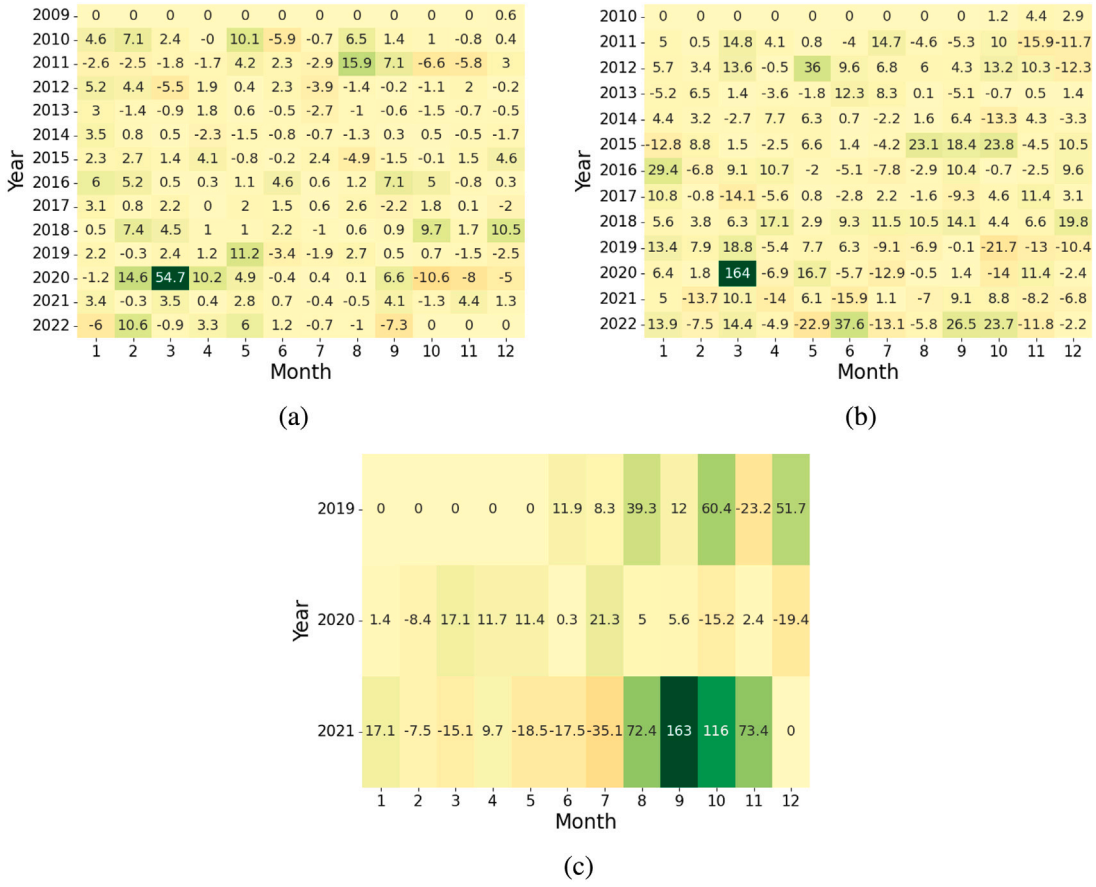


Fig. 13. ETDQN monthly returns (%) between 2009 to 2022 (a) iShares S&P 500 ETF, (b) Western Digital Corporation, and (c) ATOM/USDT cryptocurrency pair.

15.9%, followed by another positive return of 7.1% in the subsequent month. Unfortunately, the model experienced negative returns in the following months. From 2012 to 2017, the profits generated by the model displayed a relatively narrow range, fluctuating between −2.5% and 5%, indicating a slight positive skew. In 2018, the ETDQN model outperformed other models in February, October, and December. 2019 also witnessed the ETDQN model outperforming all others. In 2020, the proposed model achieved its highest performance, capitalizing on falling stock prices and generating returns of 14.6%, 54.7%, 10.2%, and 4.9%. Throughout 2021, the model maintained profits ranging between 3.5% and 4.4%. However, in 2022, the model exhibited increased volatility, with returns ranging from −7.3% to 10.6%.

In comparison, Fig. 13(b) illustrates the monthly returns generated by each model for Western Digital Corporation. The ETDQN model's heatmap displays more uniform colors, indicating relatively consistent returns. In 2010, the proposed model recorded exclusively positive returns. In 2011, positive returns were observed in January, March, April, August, and October, but the year also saw significant negative returns in August, September, November, and December. Notably, the backtest was in its early stages, and model weights required further adjustment, improving as more data was incorporated into training. In 2012, December was the only month with negative returns, while March, May, June, October, and November yielded significant positive returns. The following year, 2013, displayed positive returns with occasional declines, but overall maintaining a positive balance. 2014 experienced a substantial negative return in October, balanced by positive returns in April and March, resulting in a positive annual balance. 2015 featured a negative return in January but balanced by six months of substantial positive returns. 2016 saw the model generating positive returns, whereas 2017 was characterized by predominant negative returns. Outperforming the previous year, 2018 produced substantial positive returns. In 2019, the positive-to-negative return ratio was balanced, with remarkable positive returns. In the pandemic-hit year of 2020, as the global economy faced turmoil, the ETDQN model stood out by capitalizing on the downturn, delivering an exceptional 164% return in a single month, while other models recorded returns of −25.3%, 12.6%, and −18.8% in March. In 2021, the model experienced negative returns overall, with intermittent positive returns offsetting the performance. Finally, in 2022, the model generated significant positive and some negative returns.

In contrast to previous assets, a comprehensive analysis of ATOM/USDT is challenging due to the limited three-year dataset, as

Table 6
Annual returns (%) applying five models for assets between 2009 to 2022.

Year	iShares S&P500 ETF					Western Digital Corporation					ATOM/USDT cryptocurrency				
	B&H	S&H	Random	ETDQN	TDQN	B&H	S&H	Random	ETDQN	TDQN	B&H	S&H	Random	ETDQN	TDQN
2009	−0.56	0.56	0.31	0.56	**3.04**	X	X	X	X	X	X	X	X	X	X
2010	13.37	−14.6	−0.31	**28.15**	22.77	13.84	−14.09	−9.42	8.72	**29.71**	X	X	X	X	X
2011	−2.74	−2.28	−8.25	6.75	**13.20**	−9.35	−14.73	−7.41	3.38	**67.36**	X	X	X	X	X
2012	14.59	−14.31	6.91	3.45	**10.47**	35.89	−38.16	27.78	**138.02**	6.53	X	X	X	X	X
2013	28.47	−23.07	−3.55	−4.36	**29.05**	**99.52**	−53.81	−10.16	13.35	−47.08	X	X	X	X	X
2014	9.94	−9.99	10.58	−3.22	**12.18**	**33.48**	−29.34	−13.26	11.7	6.23	X	X	X	X	X
2015	−5.34	3.19	5.58	11.85	**14.24**	−45.52	63.67	−1.03	**84.94**	61.74	X	X	X	X	X
2016	13.94	−13.9	0.54	**35.49**	15.03	10.61	−29.82	−8.13	41.73	**120.43**	X	X	X	X	X
2017	12.99	−12.00	−1.67	10.95	**19.22**	17.62	−23.50	**20.03**	−4.21	−12.26	X	X	X	X	X
2018	−11.66	10.61	2.11	**45.87**	−17.20	−53.58	88.03	−23.73	**187.24**	39.17	X	X	X	X	X
2019	28.59	−23.51	0.56	11.11	**29.42**	**71.93**	−54.00	18.59	−18.66	−58.77	−15.97	−33.33	−81.8	**253.45**	−63.81
2020	−1.16	−10.27	−40.8	**68.75**	1.05	−13.55	−26.93	1.20	**140.85**	−22.57	30.28	−79.65	−82.57	27.73	**348.30**
2021	22.37	−19.65	−4.85	19.55	**20.51**	**19.67**	−31.63	−25.94	−26.8	10.25	462.68	−98.14	−46.3	**646.56**	−53.57
2022	−11.12	**9.70**	7.54	4.14	−0.27	−43.9	**44.65**	6.60	34.29	−30.63	X	X	X	X	X

seen in Fig. 13(c). Upon reviewing the heatmap generated by the ETDQN model in Fig. 13(c), a clear trend of positive returns is evident. In 2019, TDQN outperformed all models. During the cryptocurrency rally in 2020, B&H and TDQN outperformed the proposed ETDQN model. In 2021, the ETDQN model reasserted its dominance, generating returns of 72.4%, 163%, 116%, and 73.4%, with most of the significant returns occurring towards the year's end. Compared to the S&H and Random Action models, the proposed ETDQN model consistently outperformed them each year.

5.6. Annual returns

This section provides an overview of annual returns generated by each model, showcasing their performance. Table 6 presents these returns applied to the iShares S&P500 ETF, Western Digital Corporation, and ATOM/USDT data sets. Additionally, the average annual return (AAR) is analyzed to represent the models' average performance.

In the context of the iShares S&P500 ETF, the TDQN model demonstrates the best performance in 2009, followed closely by ETDQN. Conversely, ETDQN emerges as the most profitable in 2010, surpassing TDQN. Over the next five years, TDQN consistently outperforms all models. From 2016 to 2021, TDQN and ETDQN alternate as top performers. In 2022, the S&H model claims the top position. Despite TDQN holding the top position for nine years and ETDQN for only four years, the latter model outperforms in terms of the historical AAR from 2009 to 2022. ETDQN achieves an AAR of 17.07%, surpassing TDQN's 12.34%.

In the Western Digital Corporation data set, the ETDQN model initially ranks third in terms of performance compared to TDQN and B&H in 2010. However, in the subsequent year, the proposed model secures the second position, trailing TDQN by a 63.68% return. In 2012, ETDQN outperforms both. In 2016, TDQN takes the lead, generating nearly triple the returns compared to ETDQN. Nonetheless, in 2017, ETDQN retakes the second position, trailing behind B&H, which yields an impressive 99.52% return. In 2014, the proposed model achieves approximately half the returns of B&H but bounces back in 2015, generating an 84.94% return. Starting from 2017, TDQN's performance weakens, resulting in only a few significant returns, such as in 2018 and 2021. During 2018 and 2020, ETDQN capitalizes on market fluctuations, delivering its best returns of 187.24% and 140.85%, respectively. Finally, the B&H model assumes the top position in annual returns in 2019 and 2021, yielding returns of 71.93% and 19.67%, respectively. In 2022, S&H outperforms ETDQN with a 10.36% difference.

Even though both the B&H and ETDQN models have the same number of times that achieve the first position concerning annual returns, ETDQN has an AAR from 2010 to 2022 of 47.27%, against 10.52%. In addition, the TDQN presents the AAR from 2010 to 2022 of 13.09%, being positioned in the second position regarding annual returns. The ETDQN shows aggressive behavior with the highest standard deviation of 65.93, compared to 48.44 and 43.14 from TDQN and B&H, respectively. According to the analysis, the model ETDQN tends to produce positive returns with high volatility, which is the reason that it achieves the best performance compared to all models.

Although both the B&H and ETDQN models achieve the top position in annual returns an equal number of times, ETDQN exhibits an AAR from 2010 to 2022 of 47.27%, outperforming B&H's 10.52%. Additionally, TDQN presents an AAR from 2010 to 2022 of 13.09%, securing the second position in terms of annual returns. ETDQN's aggressive approach is evident with the highest standard deviation of 65.93, compared to 48.44 for TDQN and 43.14 for B&H. The analysis indicates that ETDQN tends to generate positive returns with high volatility, contributing to its superior performance compared to all other models.

In the ATOM/USDT cryptocurrency data set, ETDQN outperforms all models in 2019 and 2021, delivering annual returns of 253.45% and 646.56%, respectively. However, it secures the second position in 2020, trailing TDQN's 348.30% return. ETDQN maintains the best performance in terms of AAR from 2019 to 2021, boasting a figure of 309.25%. The strategy consistently exhibits aggressive and volatile behavior, with a standard deviation 33.24% and 18.61% higher than TDQN and B&H, respectively. It is evident that ETDQN consistently secures either the first or second position, even as other models alternate between first, third, and last places.

6. Conclusion and future research

This study introduces a DRL-based trading system designed to process information related to stock market activities. At its core, the proposed model comprises a variant of the DQN known as ETDQN, which builds upon the Trading DQN benchmark. ETDQN incorporates enhancements tailored to identifying and capitalizing on trading opportunities. This model optimizes decision-making by adapting to infrequent environmental feedback and prioritizes experiences containing various sub-goals, all aimed at maximizing profits. Unlike the standard TDQN model, ETDQN demonstrates successful generalization, effectively identifying key events characterized by higher market volatility. Notably, it navigates events such as the March 2020 COVID-19 market downturn and the growth of ATOM/USDT in July 2021 while optimizing profit without requiring complex reward adjustments.

The reward function employed in ETDQN is based on exponential profit and loss, assuming that the model can perceive exponential growth within a given time frame. The trading environment is constructed to include candlestick dollar bars, technical indicators, and time-based features such as timestamps and day-of-the-week signatures. Historical intraday tick data for assets, including iShares Core S&P 500 ETF, Western Digital Corporation, and ATOM/USDT, are collected and pre-processed based on market value to generate the bars mentioned above.

In terms of daily average cumulative returns, ETDQN demonstrates significant outperformance. Specifically, it achieves approximately 1.46 and 7.13 times higher profitability than TDQN for iShares S&P 500

ETF and Western Digital Corporation, respectively. Moreover, ETDQN outperforms B&H by a factor of 2.14 for iShares S&P 500 ETF. When considering the 6-month average Sharpe ratio metric, the proposed algorithm exhibits a superior risk-return ratio, surpassing TDQN and B&H by factors of 1.01 and 3.33, respectively. Although ETDQN achieves a maximum Sharpe ratio of 7.3 for ATOM/USDT, it falls short of outperforming B&H on average. Regarding maximum drawdown, ETDQN presents 10.7% less risk of ruin for iShares S&P 500 ETF compared to TDQN and 8.8% less risk compared to the random action strategy for Western Digital Corporation. However, for ATOM/USDT, the model exhibits 2.7% more risk of ruin than B&H. The proposed model consistently outperforms other models in the context of the AAR metric.

For future research, the proposed model's applicability to trade various asset classes warrants exploration. Additionally, sentiment analysis can be leveraged to augment information processing, employing natural language processing techniques to analyze textual data and predict sentiment classes as inputs to the model's state. As suggested by Ribeiro et al. (2021), incorporating models to predict price return volatility could provide valuable insights into future trading opportunities. Moreover, computational intelligence approaches and linear models can be integrated to predict future prices, moving beyond reliance solely on past and current price data in the model's state. Researchers can further delve into reward decomposition techniques, which involve assigning rewards upon achieving sub-goals, contributing to enhanced generalization and performance.

CRediT authorship contribution statement

Lucas de Azevedo Takara: Conceptualization, Methodology, Software, Formal analysis, Validation, Writing – original draft, Writing – review & editing. **André Alves Portela Santos:** Writing – review & editing. **Viviana Cocco Mariani:** Supervision, Conceptualization, Writing – review & editing. **Leandro dos Santos Coelho:** Supervision, Conceptualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This research was funded by Conselho Nacional de Desenvolvimento Cientifico e Tecnologico – CNPq (Grants number: 307958/2019-1-PQ, 307966/2019-4-PQ, 404659/2016-0-Univ, and 408164/2021-2-Univ), and PRONEX 'Fundação Araucária' 042/2018, and Comunidad de Madrid Government through project 2022-T1/SOC-24167 and Ministerio de Ciencia y Inovacion through project PID2022-138289NB-I00.

References

Andrychowicz, M., Crow, D., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., & Zaremba, W. (2017). Hindsight experience replay. In *Advances in neural information processing systems 30: Annual conference on neural information processing systems* (pp. 5048–5058).

Arratia, A. (2014). Statistics of financial time series. In *Computational finance: An introductory course with R* (pp. 38–39). Paris, France: Atlantis Press.

Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. In *Proceedings of the 34th international conference on machine learning* (pp. 449–458). Sydney, Australia: PMLR.

Bellman, R. (1957). A Markovian decision process. *Indiana University Mathematics Journal*, *6*, 679–684.

Brim, A. (2020). Deep reinforcement learning pairs trading with a double deep Q-network. In *10th annual computing and communication workshop and conference* (pp. 0222–0227).

Brim, A., Flann, N., & S., N. (2022). Deep reinforcement learning stock market trading, utilizing a CNN with candlestick images. *PLoS ONE*, *17*(2), 1–25.

Carta, S., Corriga, A., Ferreira, A., Podda, A. S., & Recupero, D. R. (2021). A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning. *Applied Intelligence*, *51*(2), 889–905.

Choi, J. (2021). Maximum drawdown, recovery, and momentum. *Journal of Risk and Financial Management*, *14*(11), 542.

Conegundes, L., & Pereira, A. C. M. (2020). Beating the stock market with a deep reinforcement learning day trading system. In *International joint conference on neural networks* (pp. 1–8).

de Prado, M. L. (2018). *Advances in financial machine learning* (1st ed.). Hoboken, New Jersey, USA: Wiley Publishing.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Hessel, M., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., & Legg, S. (2018). Noisy networks for exploration. In *6th international conference on learning representations conference track proceedings*. Vancouver, Canada: OpenReview.net.

Hao, Z., Zhang, H., & Zhang, Y. (2023). Stock portfolio management by using fuzzy ensemble deep reinforcement learning algorithm. *Journal of Risk and Financial Management*, *16*(3).

Hasselt, H. v., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q-learning. In *Proceedings of the thirtieth AAAI conference on artificial intelligence* (pp. 2094–2100). AAAI Press.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G., & Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the 32nd AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18)* (pp. 3215–3222). New Orleans, USA: AAAI Press.

Huang, Z., Gong, W., & Duan, J. (2023). TBDQN: A novel two-branch deep Q-network for crude oil and natural gas futures trading. *Applied Energy*, *347*, Article 121321.

Jang, J., & Seong, N. (2023). Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory. *Expert Systems with Applications*, *218*, Article 119556.

Jiang, Z., & Liang, J. (2017). Cryptocurrency portfolio management with deep reinforcement learning. In *Intelligent systems conference* (pp. 905–913).

Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and Software Technology*, *55*(12), 2049–2075.

Koratamaddi, P., Wadhwani, K., Gupta, M., & Sanjeevi, S. G. (2021). Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation. *Engineering Science and Technology, An International Journal*, *24*(4), 848–859.

Kumar, P. (2023). Deep reinforcement learning for high-frequency market making. In *Proceedings of machine learning research: vol. 189, Proceedings of the 14th Asian conference on machine learning* (pp. 531–546). Hyderabad, India.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, *17*(1), 1334–1373.

Li, Y., Zheng, W., & Zheng, Z. (2019). Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access*, *7*, 108014–108022.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Oricsoft (2017). Kibot historical intraday data. URL: http://www.kibot.com/. (Accessed 13 September 2023).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd international conference on neural information processing systems*. Red Hook, NY, USA: Curran Associates Inc.

Ribeiro, G. T., Santos, A. A. P., Mariani, V. C., & Coelho, L. S. (2021). Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility. *Expert Systems with Applications*, *184*, Article 115490.

Sagiraju, K., & Mogalla, S. (2022). Deployment of deep reinforcement learning and market sentiment aware strategies in automated stock market prediction. *International Journal of Engineering Trends and Technology*, *70*(1), 43–53.

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2016). Prioritized experience replay. In *4th international conference on learning representations conference track proceedings*.

Sharpe, W. F. (1994). The sharpe ratio. *The Journal of Portfolio Management*, *21*(1), 49–58.

Shavandi, A., & Khedmati, M. (2022). A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. *Expert Systems with Applications*, *208*, Article 118124.

Shin, H. G., Ra, I., & Choi, Y. H. (2019). A deep multimodal reinforcement learning system combined with CNN and LSTM for stock trading. In *International conference on information and communication technology convergence* (pp. 7–11).

Si, W., Li, J., Ding, P., & Rao, R. (2017). A multi-objective deep reinforcement learning approach for stock index future's intraday trading. In *10th international symposium on computational intelligence and design, vol. 2* (pp. 431–436).

Singh, S., Goyal, V., Goel, S., & Taneja, H. (2022). Deep reinforcement learning models for automated stock trading. *Advances in Transdisciplinary Engineering*, *27*, 175–180.

Suliman, U., van Zyl, T. L., & Paskaramoorthy, A. (2022). Cryptocurrency trading agent using deep reinforcement learning. In *9th international conference on soft computing & machine intelligence* (pp. 6–10).

Sun, S., Xue, W., Wang, R., He, X., Zhu, J., Li, J., & An, B. (2022). DeepScalper: A risk-aware reinforcement learning framework to capture fleeting intraday trading opportunities. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 1858–1867). New York, USA: Association for Computing Machinery.

Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, *173*, Article 114632.

van Eck, N. J., & Waltman, L. (2010). Software survey: Vosviewer, a computer program for bibliometric mapping. *Scientometrics*, *84*(2), 523–538.

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., & de Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *JMLR workshop and conference proceedings*: *vol. 48*, *Proceedings of the 33nd international conference on machine learning* (pp. 1995–2003). New York, USA: JMLR.org.

Wu, X., Chen, H., Wang, J., Troiano, L., Loia, V., & Fujita, H. (2020). Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, *538*, 142–158.

Yang, H., Liu, X. Y., Zhong, S., & Walid, A. (2020). Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the 1st ACM international conference on AI in finance* (pp. 1–8). New York, USA: Association for Computing Machinery.