AI

MDPI

Review

Reinforcement Learning in Quantitative Trading: A Review of Current Approaches and Breakthroughs

Matloob Khushi [1,2,*], Rakesh Kumar Sanodiya [3], Basem Suleiman [4], Sachit Desa [5], Shahadat Uddin [6] and Xiaobin Lu [2]

[1] Department of Computer Science, Brunel University of London, Kingston Lane, Uxbridge UB8 3PH, UK
[2] School of Computer Science, The University of Sydney, Camperdown, Sydney, NSW 2050, Australia; xilu5047@uni.sydney.edu.au
[3] School of Artificial Intelligence and Data Engineering, Indian Institute of Technology Ropar, Rupnagar 140111, Punjab, India
[4] School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2033, Australia
[5] School of Environment, Education, and Development, The University of Manchester, Oxford Rd., Manchester M13 9PL, UK
[6] School of Electrical and Computer Engineering, The University of Sydney, Darlington, NSW 2006, Australia
* Correspondence: matloob.khushi@brunel.ac.uk

Abstract

Reinforcement learning (RL) provides an autonomous approach to quantitative trading by allowing an agent to learn a trading policy through repeated interaction with a market environment rather than through hand-crafted rules. The aim of an RL trading agent is to select actions that maximize long-term rewards while meeting a chosen level of risk. This review examines RL applications in quantitative trading, with particular attention to value-based, policy-based, and actor–critic methods. It summarizes the reported returns, risk-adjusted performance, evaluation settings, and practical limitations of the reviewed studies. Although many RL methods report promising results in backtests and simulated environments, their outcomes are highly dependent on data selection, feature construction, transaction cost assumptions, training procedures, reward design, and validation protocols. Consequently, reported returns, Sharpe ratios, and drawdowns across studies are not directly comparable and should not be interpreted as evidence that one RL method is universally superior to another. This review shows that relevant state information, suitable reward functions, realistic risk management, and deep neural feature extraction can improve RL trading performance. However, reliable progress requires transparent and consistent reporting of data handling, time-respecting validation, execution assumptions, transaction costs, slippage, and risk-adjusted trading outcomes. We therefore argue that future RL trading research should adopt Quantitative Finance Reporting Standard (QFRS), to make trading claims data leakage free, reproducible, back-tested, economically meaningful, and comparable across studies.

Keywords: machine learning in trading; reinforcement learning; deep learning; trading

Check for updates

Academic Editor: Joaquín Torres-Sospedra

Received: 21 May 2026
Revised: 7 September 2026
Accepted: 14 September 2026
Published: 10 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Predicting the price fluctuation of financial instruments may be profitable because exploitable short-term price inefficiencies can translate into positive risk-adjusted returns. However, profit margins are curtailed by commissions, large profit drawdowns, and excessive switching behavior. The complexities of equity markets arise from various uncertainties driven by speculators, momentum traders, institutions, and influences from

macroeconomic and social factors. The dynamics of trading within this landscape changed with the advent of the Internet in the early 1990s. This technological revolution facilitated the rise of electronic trading, leading to increased efficiency, tighter bid/ask spreads, and reduced transaction costs. The widespread availability of affordable mobile devices in the mid-2000s, coupled with Internet connectivity, further empowered traders to operate anytime and anywhere. While these advancements enhanced market participation, liquidity, and volume, they also introduced vast amounts of data, adding noise to market signals and underscoring the need for improved risk management and prediction tools.

Initially, risk management and algorithmic trading relied heavily on rule-based systems crafted by domain experts, which programmers then translated into decision-making algorithms. These systems, however, had notable limitations. They were often subjective, heavily biased by pre-existing knowledge, and lacked the ability to adapt to changing market conditions. Additionally, developing, testing, and deploying such algorithms demanded considerable time and effort. In this context, reinforcement learning (RL) has emerged as a powerful tool, offering a framework that enables agents to learn and adapt their trading strategies to maximize returns in this multifaceted environment.

While existing review papers have explored different aspects of RL in trading [1,2], this survey aims to provide a more comprehensive examination of RL applications in quantitative trading. It offers a detailed taxonomy of existing works, along with the key data, findings, and results of RL algorithms. Additionally, it presents a thorough discussion of RL algorithms across different data types, evaluating their merits and performance against key metrics.

The review concludes that the published superiority of RL methods and claims about making profits are not comparable. Hence we recommendation of adoption of the Quantitative Finance Reporting Standard (QFRS), an established seven-part framework for transparent reporting of data handling, information availability, reward or label construction, preprocessing, validation, performance evaluation, and cost- and slippage-aware backtesting. Applying QFRS would enable reviewers, researchers, and practitioners to assess whether reported RL gains represent robust and economically meaningful trading evidence, rather than artefacts of dataset selection, backtest design, or incomplete reporting.

The structure of this review is as follows. Section 2 covers this review's background and motivation, introduces common trading performance metrics, and provides an overview of machine learning (ML) and RL concepts. Section 3 describes the review approach. Section 4 details the RL algorithms discussed, including their performance and metric evaluations. In Section 5, a discussion of the current state of the art in RL and potential future research opportunities is presented. Finally, Section 6 presents this review's conclusions. The goal of this review is not to be exhaustive but to highlight the novel methods, approaches, and techniques that support the application of RL in quantitative trading.

2. Background

2.1. Motivation

The efficient market hypothesis describes market price as a reflection of the fair value of a financial asset [3]. The basic premise of the hypothesis implies that predictive analysis is unlikely to yield a competitive advantage, as the true valuation of assets incorporates relevant information. However, this has not dissuaded the academic and commercial sectors from exploring the use of fundamental assessments, technical analysis, and ML to challenge the efficient market hypothesis [4].

Initially, simple moving averages were used as a form of technical analysis. The study in [5] demonstrated that using moving averages could lead to above-normal returns compared to the buy-and-hold (BH) strategy. Similarly, using a 31-year sample of U.S. stock data from

1962 to 1996, ref. [6] found that several technical indicators added value to the investment process. Technical analysis is also often paired with Fibonacci sequences and ratios to determine entry and exit points. Ref. [7] further showed that security prices consistently exhibited retracement patterns that aligned with Fibonacci sequences, demonstrating their predictive value in trading systems.

Building on the foundations of technical analysis, the Elliott Wave Principle was introduced in the 1970s in [8], based on R.N. Elliott's 75-year study of the Dow Jones Industrial Average (DJIA). Elliott observed that market trends and reversals were influenced by investor psychology and followed repetitive, predictable patterns. According to this theory, coined the Elliott Wave (EW), markets move upward in a series of five waves and downward in a series of three waves. Applying this theory, ref. [9] successfully predicted EURUSD patterns between 2009 and 2015. Similarly, the study in [10] on U.S. stocks (1916–1976) found that historical peaks and valleys aligned with both the EW and Fibonacci sequences. These findings further support the notion that stock markets exhibit discernible trends and patterns, reinforcing the potential of using technical analysis in quantitative trading.

2.2. Financial Performance Metrics

The most common performance metric for evaluating a trading strategy is realized profit and loss (P&L), which can be expressed in absolute terms or as a percentage return. The absolute return at any given time, t, is calculated as $z_t = p_t - p_{t-1}$, where p_t represents the price at time t. Given a trading decision $\delta_t \in \{-1, 0, 1\} = \{long, neutral, short\}$, the trading return is calculated as $R_t = \delta_{t-1} z_t - c|\delta_t - \delta_{t-1}|$, where c is the transaction cost. Alternatively, unrealized P&L represents the net profit yet to be settled, and conservative traders may apply a dampening factor, η, to adjust their P&L. However, relying solely on P&L can be insufficient, as it does not account for the variance in returns, which is crucial for risk-averse investors.

To address risk, the Sharpe ratio is frequently used to measure risk-adjusted returns. It is represented by $(r - r_f)/\sigma$, where r is the expected portfolio return, σ is the standard deviation of the asset's returns, and r_f is the risk-free rate (e.g., Treasury Bill). A Sharpe ratio of 1 or higher is considered good, indicating higher returns relative to risk. However, the Sharpe ratio assumes returns are normally distributed and is insensitive to the direction of price movements, potentially leading to overestimated results in skewed distributions or highly volatile markets like hedge funds or cryptocurrencies. To address this, the Sortino ratio focuses on negative volatility, providing a more accurate measure of downside risk.

The choice of performance metric depends on the investor's objectives and risk tolerance. Selected metrics also play a significant role in guiding the reinforcement learning process in quantitative trading strategies.

2.3. Non-Reinforcement Learning for Quantitative Trading

Prior to the popularity of RL, methods for quantitative trading included statistical, econometric, and non-RL ML approaches. These methods relied on financial and probabilistic principles to support quantitative trading.

2.3.1. Statistical Methods

Basic statistical methods for trading encompass linear regression, multivariate analysis, and time-series models. Linear regression and multivariate analysis analyze associations between multiple variables to identify patterns and relationships, while likelihood inference models estimate parameters for probabilistic predictions about future price movements. Time-series models, such as ARIMA (autoregressive integrated moving average), explore the relationships among variables over time. ARIMA combines three components: an autoregressive (AR) component, which uses past values to predict future ones; an integrated (I)

component, which makes data stationary through differencing; and a moving average (MA) component, which models the relationship between observations and residual errors.

2.3.2. Machine Learning

Artificial intelligence (AI), with an ability to solve cognitive problems like pattern recognition and decision-making, has evolved significantly, incorporating principles from various fields such as mathematics, game theory, and decision theory. Within AI, a subfield of ML, deep learning (DL) has emerged as a powerful tool that automates learning using artificial neural networks (ANNs) with many layers. Classical ML methods can be categorized into supervised, unsupervised, semi-supervised, self-supervised, and RL approaches, each suited to different prediction tasks, including forecasting financial trends.

Artificial neural networks (ANNs), the backbone of DL algorithms, have demonstrated strong potential in financial trading. The authors of [11] undertook a survey of studies that applied ANNs to foreign exchange trading. They found that 60.0% of studies reported that ANNs outperformed other models in foreign exchange trading, although outcomes varied based on data types and forecasting periods. ANNs have demonstrated their effectiveness in handling high-dimensional data and avoiding the "curse of dimensionality," making them adaptable to complex financial datasets [12–14].

The rise of DL has been supported by advancements in technologies like graphics processing units (GPUs) and improvements in ANN techniques such as gradient descent and weight initialization [15,16]. These advancements have allowed ANNs to converge more efficiently and have led to breakthroughs in financial applications. For example, convolutional neural networks (CNNs) have shown promise in quantitative trading, outperforming traditional strategies like buy and hold (BH). Studies such as [17] demonstrated that CNNs trained on technical indicators of the S&P 500 data were more effective in predicting stock prices than models based on ANNs and support vector machines (SVMs).

These advancements in DL, particularly neural networks like CNNs, provide a strong foundation for more complex approaches like deep reinforcement learning (DRL), which further enhances decision-making in quantitative trading by learning optimal strategies from market data. This progression sets the stage for DRL's application in developing more adaptive and profitable trading strategies.

2.3.3. Probabilistic Models and Optimization Functions

Probabilistic models play a crucial role in managing uncertainty and informing decision-making in quantitative trading by estimating the likelihood of various market outcomes. Bayesian networks, for instance, represent dependencies among variables through directed acyclic graphs, capturing relationships between market indicators and asset prices. Similarly, hidden Markov models (HMMs) identify unobserved regimes, such as bull or bear markets, and predict market movements by estimating probabilities of state transitions, providing insights into evolving market conditions.

In the context of multi-asset trading, optimization functions further refine trading strategies by determining optimal asset allocations or strategy parameters that maximize returns. These functions, though computationally intensive, ensure that model parameters are fine-tuned for enhanced performance. Techniques like Bayesian optimization are particularly effective, as they leverage historical data to balance exploration and historical performance, proving valuable in scenarios where objective functions are noisy and challenging to predict. This interconnected approach of combining probabilistic modeling with optimization highlights the adaptability of non-reinforcement methods in addressing the complexities of quantitative trading.

Although non-RL methods have shown strong potential for quantitative trading, their models are generally trained on fixed historical datasets and often require retraining when market conditions change substantially. RL methods, by contrast, can in principle update their policies through ongoing interaction with the market, giving them greater flexibility in dynamic environments. However, RL systems typically demand extensive data preparation, reward shaping, and hyperparameter tuning, and their training can be unstable, so the practical effort required to maintain an RL-based strategy is not necessarily lower than that required to retrain a non-RL model.

2.4. Reinforcement Learning

Reinforcement learning (RL) is a learning approach that automates the decision-making process through repetitive self-learning and evaluation [18]. Figure 1 shows the components of RL, namely the agent, action, reward, and system, also referred to as the environment. Unlike supervised and unsupervised learning, RL does not require pre-existing training data. Instead, the RL agent generates training data simply by interacting with its environment or system. RL derives abstract representations from each simulation to make better decisions over time via the Markov decision process (MDP).

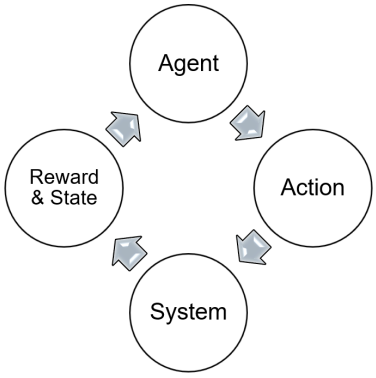


Figure 1. Markov decision process.

An MDP is a mathematical representation of a sequential state, action, reward, and environment or system. The approach is typified by a decision-maker, or an agent, interacting with its environment. At each time step, the agent observes the representation of the environment's state and decides on an action. The action taken at the beginning of an episode is purely random and thus equivalent to learning from scratch (exploration). As the environment transitions to a new state, the agent is either rewarded or penalized. As a consequence of its action, the agent learns to alter its behavior in the next episode by choosing an action that achieves the maximum reward (exploitation). The repetitive nature of such a decision-making process can support an agent in developing a policy related to states, actions, and rewards.

In trading, *exploitation* describes the actions of an investor who executes strategies that are known to be profitable. In contrast, *exploration* describes the adoption of a strategy that has not been used before. *Exploitation* is also termed 'acting greedily' because the trader chooses to exploit their current knowledge to pick the best strategy. However, this is done at the expense of possibly missing out on other unexplored strategies that may yield better results. A balance between an *exploration* and *exploitation* policy can be struck if the agent is allowed to behave greedily most of the time but also explores with a small probability, ε, to randomly uncover other competitive strategies.

One way of categorizing RL methods is into model-based and model-free approaches. The former requires state probabilities and reward functions to be known, allowing the agent to plan ahead. The latter hinges on trial and error to learn patterns, relationships,

and policies derived from data or through the environment. In lieu of trying to understand system dynamics and explicitly model them, model-free techniques aim to find optimal actions and make predictions based on observed input-output data or experiences. Within the context of trading, reliance on domain experts to carefully craft trading strategies can be replaced with RL. Through the learning process, an agent eventually discovers which action (buy/sell) will better maximize the reward within a state.

The aim of reinforcement learning is to train an agent to gradually refine its decision-making process and maximize the expected future reward, or achieve a goal. We define the goal as $G_t = r_{t+1} + \gamma G_{t+1}$, where t is the time, r is the reward, and $\gamma \in [0, 1)$ is the discount rate. The use of the discount rate γ is analogous to the concept of the time value of money in finance, whereby an immediate reward is considered more desirable than future rewards. A good reward function encourages an agent toward an intended outcome, while a seemingly harmless reward function can easily have the opposite of the intended effect [19]. For example, a sparse reward is a type of infrequent feedback given only when a trade is closed. Without constant feedback, an agent may end up holding a negative trade indefinitely until it appreciates in value. One way to avoid such a situation is to use unrealized P&L, which provides immediate feedback at each time step [20].

RL agents balance exploration and exploitation to autonomously learn and adapt to complex and dynamic market environments. Unlike traditional models, RL provides a framework for sequential decision-making that, in principle, can improve trading policies through continuous interaction with the market. However, in financial applications, RL often suffers from sample inefficiency and instability, and its performance can degrade when market regimes change, for example, through catastrophic forgetting of past patterns. As a result, current RL trading systems should be viewed as promising research prototypes rather than fully mature solutions for volatile real-world markets.

3. Review Methodology

The primary literature search was conducted using dblp with the keyword string "reinforcement learning trading". Because dblp is a computer science bibliography that primarily indexes computer science journals, conference proceedings, and related publications, this review is deliberately framed from a computer science methodological perspective. It focuses on how reinforcement learning algorithms are designed, trained, evaluated, and applied to trading- and portfolio-related tasks, rather than attempting to provide a complete survey of all financial, economic, and econometric research on trading. Studies were included if they applied an RL algorithm to a trading- or portfolio-related task using real or realistic market data and reported quantitative performance against at least one baseline. Studies were excluded if they did not use finance-related data, did not provide a clear evaluation setup, or referred to RL only at a conceptual level. Of the 440 records identified through the dblp search, 150 studies met these criteria and were reviewed. Therefore, the review aims to provide a representative and methodologically focused account of RL-based quantitative trading research from the computer science literature, rather than an exhaustive systematic review across all finance and economics journals.

Citation networks were constructed and visualized using Gephi version 0.11.1 to examine relationships among studies within the reviewed literature. Separate networks were generated for value-based, policy-based, and actor–critic methods, alongside a combined network representing the complete set of reviewed studies. Publications were represented as nodes and citation relationships as edges. Network layouts were generated using the ForceAtlas force-directed algorithm, which positions more interconnected publications closer together while separating less closely related nodes. Layout parameters were adjusted according to network size and density to produce stable and legible visualizations;

repulsion strength ranged from 350 to 2000, while the remaining principal layout parameters were held constant. These parameters control the spatial arrangement of the networks and were not used as analytical thresholds for determining the presence or absence of citation relationships. Networks were interpreted primarily according to connectivity, clustering, and the relative position of publications, allowing highly interconnected studies and methodological relationships to be identified.

4. Review of RL Articles on Quantitative Trading

This review addresses the different types of RL algorithms applied in trading and compares their accuracy and efficiency. The objective of an RL algorithm is to maximize the cumulative reward. As the cumulative reward depends on the action selected by the agent at each state, the agent learns an optimal policy through repeated interaction with the environment. An optimal policy, π^*, is defined as the policy that maximizes the expected cumulative reward. Reinforcement learning algorithms can be categorized in different ways. In this review, they are classified according to how the optimal policy is learned into three main classes: value-based, policy-based, and actor–critic methods. In value-based methods (critic-only), the optimal policy is implicitly derived by approximating the optimal state-value function $V^*(s)$ or action-value function $Q^*(s,a)$. In policy-based methods (actor-only), the policy is learned directly through a parameterized function $\pi_\theta(a \mid s)$. Actor–critic methods combine these two approaches by simultaneously learning a policy (actor) and a value function (critic).

4.1. Value-Based Methods

Value-based methods stem from the idea that an agent learns a value function and evaluates the outcome of an action. A value function, as illustrated in Figure 2, consists of an agent observing the current state of the market and informing a choice of action, such as buying, selling, or holding an asset. This action is executed in the environment, leading to a new state and generating a reward, typically measured in profit or loss. The value function then updates based on the received reward, helping the agent refine its strategy by evaluating the long-term benefits of its actions in various states. This feedback loop enhances the agent's ability to make optimal trading decisions over time.

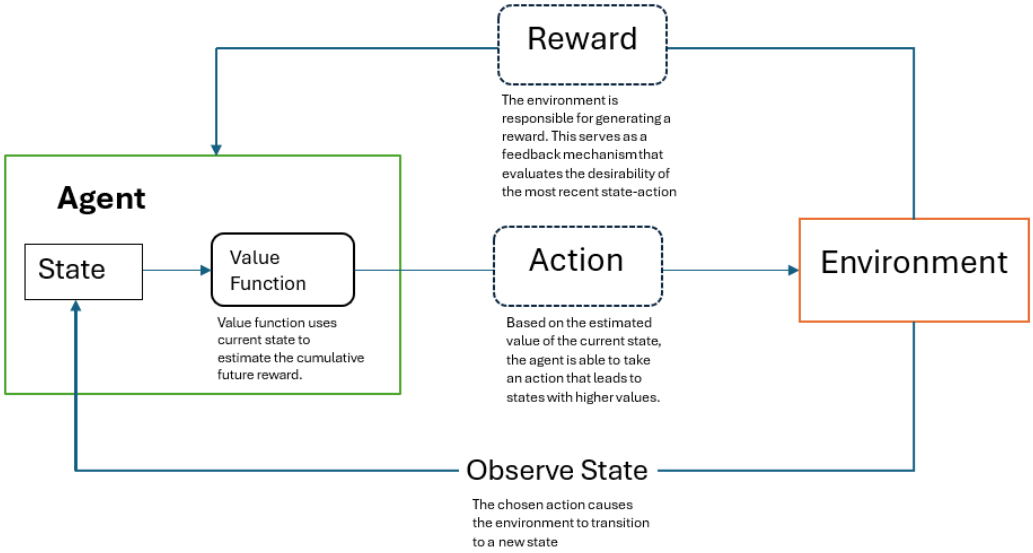


Figure 2. Feedback loop for a value function.

In a value function, the agent seeks to optimize its decisions by identifying the best action in any given state to maximize cumulative future rewards. To do this, it evaluates

either the state-value function, V, or action-value function, Q. The state-value function, defined as $V_\pi(s) = E_\pi[G_t \mid S_t = s]$, represents the expected return of each state, s, when following a policy, π. For actions, the function becomes $Q_\pi(s,a) = E_\pi[G_t \mid S_t = s, A_t = a]$, also known as the Q-value, which represents the expected return of each state–action pair. To determine the optimal policy, π_*, we seek to maximize these values:

$$V_*(s) = max_\pi V_\pi(s); Q_*(s,a) = max_\pi Q_\pi(s,a) \quad (1)$$

The value function can be estimated using methods like dynamic programming (DP), Monte Carlo (MC) sampling, or temporal difference (TD) learning. DP is model-based, requiring knowledge of state probabilities and rewards. When this information is unavailable, MC and TD allow learning directly from samples. In MC, updates are made toward the return G_t:

$$Q(s,a) \leftarrow Q(s,a) + \gamma(G_t - Q(s,a)) \quad (2)$$

MC methods can be used to estimate value functions by sampling complete episodes or trajectories from an environment. MC sampling provides an unbiased estimate of the true value function by averaging the observed returns obtained from sampled episodes.

The core principle of TD is learning to estimate the value function by bootstrapping from the current state to the next state using observed rewards and value estimates. Value-function updates using TD are made toward $R_{t+1} + \gamma Q(s', a')$ in the expression:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(R_{t+1} + \gamma Q(s', a') - Q(s,a)) \quad (3)$$

TD methods, such as Q-learning and SARSA, face the challenge of bias from initial values, which can lead to instability. Q-learning is off-policy, learning from past experiences, while SARSA is on-policy and requires new samples for each policy update, making it less efficient.

In environments with finite states, the value function can be directly calculated. However, with larger action spaces, we may need to use custom discretization or leverage prior knowledge. In real-world scenarios with infinite state–action spaces, we approximate the value function or policy using weights, θ, often through deep neural networks for better generalization.

Earlier reinforcement learning algorithms struggled with stability, particularly when using nonlinear function approximators. Ref. [21] addressed this by introducing experience replay buffers for uncorrelated data sampling and a fixed target network. This led to the development of the Deep Q-Network (DQN), which approximates $Q(s,a,\theta_i)$ using convolutional neural networks. The loss function is defined as

$$L_i(\theta_i) = E_{(s,a,r,s') \backsim U(D)} \left(r + \gamma max_\alpha Q(s', a', \theta_i^-) - Q(s,a,\theta_i) \right)^2 \quad (4)$$

Here, γ indicates the time horizon, θ_i represents the current network parameters, and θ_i^- represents the parameters for the target network. This separation enhances stability and convergence. Variants of DQN, such as Double DQN and Deep Recurrent Q-Network (DRQN), further improve performance. Table 1 presents a summary of the value-based functions that demonstrate the best results in returns and profit margins. The results are reported as in the original studies and are not strictly comparable across different datasets, time periods, and evaluation setups (see the discussion in Section 5).

Table 1. Summary of value-based methods (critic-only).

References	Data	Summary and Key Findings	Results
[3,20]	Forex	Q-Learning algorithms are capable of finding relationships within financial data. However, conventional models are labor-intensive and not "out-of-the-box" solutions. AI principles such as SHAP can provide explanations of agent actions and decisions for simple DQN algorithms.	Annual returns of 16.3 ± 2.8%
[22]	U.S. stocks + private and market variables	Q-learning is trained to optimize limit orders and reposition remaining stock inventory. The choice of state variables and the use of efficient algorithms results in better performance than simple optimization.	Improvement in execution of over 50% more than the benchmark
[23]	Korean stocks + indicators	Multiple agents are trained to work cooperatively to execute trades. These multi-agent networks outperform single agents due to a form of collective intelligence that allows a broader set of indicators to be considered.	Achieved 2.5–4 times asset growth
[24,25]	Italian stocks	Compares Sharpe ratio calculated over L {5,22} days and runs 1000 randomly initialized tests with Q-learning and KbRL.	Sharpe ratio: Q-learning fared better than KbRL
[26]	U.S. stocks and T-Notes	High learning frequency or adaptive learning with continuous actions consistently perform well. Annual trading frequency produces better results than semi-annual or quarterly trading.	TD consistently outperformed Q and SARSA by a significant margin
[27]	Forex, indicators	DRQN is trained using smaller replay memory, and longer training sampling shows good returns. A similar framework with a feature extraction module and a decision-making module outperforms existing SOTA algorithms.	Achieved annual returns of 23.8% and an improvement of 12.4% over SOTA models
[5,28,29]	Bitcoin, stocks	DDQN using the Sharpe ratio returns an average return of 8.0% with a standard deviation of 2.77 in the out-of-sample test. Reward functions that adopt a long-term approach and multi-objective strategies show promise in improving profits.	The long-term reward function outperformed baseline algorithms by a significant margin in terms of TR, SR, CR, and SoR
[30]	U.S. stocks, indicators	Q-learning determines trading strategy based on the state of the boundary boxes.	Yielded an average return of 38.8%
[31]	Korean stocks	MP uses the recommendations and stock fund ratio over assets to dynamically allocate assets.	Generated a profit of 258.0%
[32–34]	Stocks	Variants of DQN are applied to the problem of pair trading by predicting and exploiting structural breaks and reversion patterns.	Newer DQN variations outperformed other state-of-the-art pair trading methods
[35–40]	Stocks, virtual currencies	Critic-only applications overcome noisy data and address issues related to micro and macro temporal patterns in order to make profitable long-term decisions.	Yielded promising results and profit margins over selected baselines for each type of problem
[7,41–44]	Forex, stocks + indicators	Hybrid models combine RL with other methods to establish long-term trading strategies. These studies leverage signals from technical indicators to inform trading and investment decisions.	Yielded promising results and profit margins over selected baselines for each type of problem

Table 1. *Cont.*

References	Data	Summary and Key Findings	Results
[45–47]	Sentiment analysis data	These studies explore RL algorithms to make trading decisions based on sentiment scores. The findings highlight how positive and negative market sentiment, as well as company-specific outlook, can drive share prices and inform trading strategies.	The method yielded promising results and profit margins over selected baselines for each type of problem
[48]	Stocks	Deep reinforcement learning incorporates momentum investment strategies from behavioral finance.	The method achieved a higher profit ratio than existing models
[49]	Cryptocurrency	Introduces multimodal financial embedding.	The method significantly improved profitability
[50]	Limit-order market	While informed trading improves information efficiency, excessive fast trading can reduce market liquidity and degrade overall market quality.	The method demonstrated the effectiveness of RL in reaching market equilibrium
[51]	Stock	A combination of deep reinforcement learning (DRL), knowledge graphs, and sentiment analysis from social media data	Boosted prediction performance
[52]	P2P electricity	Value-based QRL using a VQC and $V(s)$. State represents generation, demand, price, and trading balance; action is energy bought/sold; reward is $X_{ijt}p_t$. QRL is compared with classical RL and linear optimization. Limitation: random action selection and no reported out-of-sample evaluation.	Simulation only. QRL converged faster than classical RL and yielded higher energy-profit percentages for most prosumers; exact aggregate return, Sharpe ratio, maximum drawdown, and statistical significance values were not reported. No live/paper trading was performed; the blockchain security layer was omitted from the experiments.
[53]	Six U.S. assets: DJI, IXIC, S&P 500, AAPL, GE, and GOOGL	CSI-DDQN partitions each single-asset series across sequential clients. Double DQN learns Q-values for short/hold/long actions from OHLCV windows; reward is one-period action-weighted close return. A KL self-imitation loss aligns adjacent clients, and volatility-weighted aggregation forms a global model.	CSI-DDQN yielded a mean cumulative return of 56.85%, a Sharpe ratio of 1.42, and a maximum drawdown of 17.14%, versus DQN-Vanilla's results of 3.95%, 0.25, and not reported, respectively, and DRL-Ensemble's results of 19.53% and 0.54. The results included a stated 0.1% transaction cost
[54]	Eight DJIA stocks (AAPL, BA, GS, INTC, JNJ, MCD, PG, TRV)	RPADiRL extends IQN/DDQN to learn return-quantile distributions and applies probability distortion for risk-sensitive long/flat/short trading. State: 20-day OHLC history plus position; reward: P&L less 0.1% commission.	RPADiRL achieved the highest test cumulative return across eight stocks compared with BH, DMA, MACD, GDQN, and RLPT; e.g., AAPL, 95.99%; SR, 1.34; and MDD, 34.06%. Wilcoxon tests versus GDQN/RLPT yielded <0.01 $p < 0.01$.
[55]	Energy-trading simulation	DQN jointly selects discrete block size, block interval, cluster count, and clustering-feature weights for blockchain-enabled V2V energy trading. State contains EV location, direction, speed, and battery level; reward combines buyer/seller utility, blockchain throughput, and within-cluster energy variance under latency and security constraints.	Simulation only. The DQN scheme generated 295 simulated energy trades, versus 157 for mobility-only K-means clustering, and improved economic utility.

Table 1. *Cont.*

References	Data	Summary and Key Findings	Results
[56]	Cryptocurrency (BNB/USDT, BTC/USDT, DOT/USDT, and ETH/USDT)	FineFT selectively updates an ensemble of DQNs using ensemble TD errors, filters experts by market-dynamic profitability, and uses VAE reconstruction losses for out-of-distribution routing. Discrete actions select target long/short positions and leverage; rewards are margin-balance changes net of order loss and funding fees.	Backtesting was performed with 5x leverage and a 0.02% transaction cost. FineFT achieved a BTC TR of 76.90%, an ASR of 2.13, and an MDD of 23.51%; for BNB, the corresponding values were 83.70%, 2.55, and 15.49%. It reported lower risk than all 12 baselines.
[57]	Bitcoin and Ethereum	XGBoost selects features using one-day-ahead returns, and then DDQN with either LSTM, BiLSTM, or GRU outputs buy/hold/sell signals. States include selected market, technical, macroeconomic, and blockchain variables; rewards maximize discounted returns.	BTC: annualized return, 22.04%; Sharpe ratio, 0.393; Sortino ratio, 0.564; and MDD, 45.36%. Ethereum GRU + XG: P&L, 56.72, and Sharpe ratio, 1.22; 0.1% transaction fee implemented.
[58]	BTC/ETH options	Multi-agent DQN framework for volatility trading. OP-Agent selects long/short/neutral option positions using n-step TD learning; HR-Agent selects among risk-sensitive hedgers using TD learning. Rewards use portfolio-value change and hedger-relative value. Limitation: evaluation is restricted to BTC/ETH cryptocurrency options and a single temporal split.	Historical backtesting was performed with transaction costs. OPHR achieved 33.10% BTC and 44.89% ETH test returns, with Sharpe ratios of 1.87/1.76 and MDDs of 9.41%/27.25%, outperforming all stated baselines. Costs represented 9.36%/5.75% of PnL. Statistical significance tests were not reported.
[59]	Bitcoin	Rainbow/DD-DQN Bitcoin trading bot with open-ended learning. Actions switch between USD and Bitcoin; reward tracks price changes and penalizes trading fees. Hybrid replay combines recent and random experiences and outperforms recent sampling. Limitation: only Bitcoin and two validation periods; performance deteriorates over longer horizons.	Historical backtesting was performed with 0.01% transaction fees. The best hybrid configuration produced a median return of 41.46% versus a 40.69% market return in 2024, and 196.86% versus −64.39% in 2017–2018. Sharpe ratio and maximum drawdown were not reported. Performance showed substantial run-to-run dispersion.
[60]	Three-microgrid electricity market; 24 h simulation; industrial, commercial, residential MGs	Improved multi-agent Nash-Q for P2P power trading. States include renewable output, SOC, load, and prices; discretized actions control storage and grid/MG trades; reward penalizes operating cost and constraint violations. Experience replay and frozen target networks improve convergence. Limitation: simulation-only case study.	Simulation, not financial-market backtesting/live trading. Total operating cost was 12,065.55, 18.4% below no-P2P trading and 7.2% below traditional Nash-Q. Profit, return, Sharpe ratio, drawdown, and transaction-cost metrics were not reported.
[61]	U.S. equities (AAPL, AMZN, GOOGL, META, MSFT, TSLA)	DDQN augmented with monthly GPT-4o Mini guidance encoded as an uncertainty-weighted observation feature; discrete LONG/SHORT actions. The LLM improves state information without modifying DDQN or reward shaping. Limitations include fixed precomputed LLM signals, restricted equity universe, and no neutral/hold guidance.	Historical backtests: mean Sharpe improved from 0.64 (DDQN) to 1.10 (LLM + DDQN), with significant SR improvements across 25 runs. Mean MDD decreased from 0.36 to 0.31, although inconsistently per stock. Absolute/cumulative returns and transaction-cost effects were not reported.

Table 1. *Cont.*

References	Data	Summary and Key Findings	Results
[62]	P2P energy	Multi-agent DQN integrates probabilistic KTU forecasts and uncertainty into states and rewards for P2P energy/battery trading. Discrete actions cover buying, selling, charging, discharging, and self-consumption. Agents train independently. Limitation: simulated environment with no real-world deployment or theoretical convergence analysis.	Simulation: versus standard DQN, forecasting-DQN reduced purchase costs by 5.7% without P2P and 3.2% with P2P; sales revenue increased 6.4% and 44.7%, respectively. Peak demand fell 38.8%/45.6%. Sharpe ratio, drawdown, transaction costs, and statistical significance: Not reported.
[63]	Simulated EV energy trading	Multi-agent tabular Q-learning under centralized learning–decentralized execution. EVs choose idle, charge, or V2G discharge actions; stations adapt prices. Rewards minimize EV energy/degradation costs and maximize station net profit. Lightning Network enables micropayments. Limitation: proof-of-concept simulation with strong simplifying assumptions and no real energy-market validation.	Simulation: charging-station agents converged to positive net profits, while EV profits approached near-zero; exact returns and risk-adjusted metrics were not reported. Lightning network transactions used 0.0029% fees, versus 2.3% baseline fees, and achieved 98.2% payment success. Sharpe ratio, drawdown, and statistical tests were not reported.
[64]	Simulated peer-to-peer microgrid energy trading	SA-DDQN maximizes microgrid energy-trading benefit under photovoltaic uncertainty and energy-storage degradation. The state includes time, storage, photovoltaic generation, load, and grid price; actions set energy quantity and selling price. Reward combines trading revenue with unmet-demand penalties.	Simulation only. SA-DDQN achieved system benefits of \$4527.36 k, versus \$4004.37 k for DDQN and \$3779.98 k for DQN; after 90 days, benefits were 13.0% and 19.7% higher, respectively.
[65]	S&P 500 futures, with DIX, GEX, and VIX indicators	Level-2 wavelet denoising and attention–BiLSTM features support binary long-only buy or hold/sell decisions. DQN learns $Q(s, a)$ under \$0.25 trade costs, \$0.25 bid–ask spreads, and \$50 point value. Coiflet-4 produces the best DQN configuration.	DQN (coif4) achieved a total return of 112.5%, a Sharpe ratio of 0.960, an alpha of 0.301, and a maximum drawdown of 0.283, versus the buy-and-hold strategy's total return of 43.5% and Sharpe ratio of 0.711. With proportional slippage, Sharpe ratio fell to 0.918 and drawdown rose to 0.312.
[66]	Two agricultural-related equities: Deere & Company and BAYN.DE.	LSTM-DQL combines denoised multi-factor features, PCA, LSTM state encoding, and DQN trading. The agent selects buy-all, sell-all, or hold; reward is portfolio-value change incorporating transaction tax and position costs. Phased rolling training targets seasonal, policy-sensitive agricultural-finance markets.	The reported model achieved an annualized return of 45.12% and a Sharpe ratio of 1.51, with a claimed 35% maximum-drawdown reduction and 62% Sharpe improvement versus the benchmark. Final assets were RMB 2,631,700.
[67]	Dow Jones Industrial	LORSTransformerDRL is a value-based DQN with an LSTM–Transformer–LORS–Q-head architecture. It uses historical OHLCV, holdings, and cash; actions buy, hold, or sell one unit. Reward is portfolio-value change net of 10 bps transaction costs; replay, target networks, and ϵ-greedy exploration are used.	Results: test CR, 214.9%; Sharpe ratio, 0.216; MDD, 42.3%; and win rate, 58.3%. The proposed network outperformed CNN_DQN (CR, 197.1%) and DQN_MLP (Sharpe ratio, 0.186). Costs were 10 bps, without slippage.

4.1.1. Q-Learning to Execute Simple Trades

Ref. [20] developed an automated trading system using the Q-learning algorithm. Initially, the reward function was based on the profit realized per transaction. In order to turn a profit, the agent quickly learned to hold positions until negative unrealized profits turned positive; however, this resulted in system inefficiencies. To address this, the reward function was adjusted to distribute risk and reward throughout the episode, which not only resolved the credit assignment problem but also accelerated the learning process. Testing the system on EURUSD data from 2010 to 2017 resulted in an average yearly profit of 16.3 ± 2.8%. Despite these positive outcomes, the system was far from an ideal solution, requiring significant data preparation and model tuning, such as reward normalization, batch size configuration, update frequency, and buffer size adjustments. This example illustrates that, although Q-learning provides a flexible policy-learning framework, achieving robust performance in practice depends heavily on careful reward and hyperparameter design, and thus RL does not eliminate the need for substantial model engineering.

Building on the idea of understanding agent behavior and optimizing reward functions, explainable AI principles have been introduced to enhance transparency in automated trading systems. Ref. [68] proposed applying the SHapley Additive exPlanation (SHAP) technique to Deep Q-Networks (DQNs) to explain an agent's decision-making during trade execution. SHAP provides explanations based on the agent's pursuit of maximum cumulative rewards, which are visualized through waterfall plots, offering a clear justification of trading actions. The approach significantly increases transparency, enhancing user confidence and providing deeper insights into the agent's behavior in quantitative trading.

Further extending the capabilities of Q-learning for trade execution, ref. [69] introduced an advanced model named Extended Trading DQN (ETDQN), tailored to handle trades across varying market conditions with feedback received after trade liquidation. ETDQN improves decision-making by focusing on sub-goals, using distributional learning, and simplifying the reward structure. When tested across multiple assets, ETDQN demonstrated superior performance compared to traditional models like TDQN, excelling in cumulative returns, Sharpe ratio, and drawdown management, particularly in volatile market conditions.

Building on the advancements in Q-learning applied to trading systems, subsequent studies have explored the application of reinforcement learning (RL) to more complex trading environments, specifically using limit-order data. Ref. [22] applied Q-learning to optimize trade execution using high-frequency limit-order data. In the approach, the trading model observed the states of the order book and placed a limit order at every time step, t, adjusting the remaining stock inventory based on the current bid and ask prices. Testing the model on three NASDAQ stocks resulted in an execution performance improvement of over 50.0% compared to the submit-and-leave strategy. The study emphasized that incorporating a combination of "private" state variables (such as elapsed time and remaining stock inventory) and "market" state variables (like market order cost, bid–ask volume imbalance, and transaction volume) enhanced the model's efficiency over simpler optimization strategies.

Further expanding on the role of reinforcement learning in limit-order markets, ref. [50] explored how RL could be employed to model and optimize trader decision-making processes. In such markets, traders are categorized as either informed or uninformed. RL enables traders to make more informed decisions based on market information, such as order book data and price deviations. Informed traders leverage their knowledge of an asset's true value to place more aggressive market orders when they detect significant discrepancies between the market price and the asset's fundamental value. In contrast, uninformed traders tend to mimic informed traders, a phenomenon known as "order chasing," which contributes to herding behavior and affects market dynamics like liquidity. The study highlighted that while informed trading enhanced information efficiency, excessive rapid trading could diminish

market liquidity and overall market quality. Through simulations and analysis, the study demonstrated the effectiveness of RL in achieving market equilibrium and enhancing our understanding of trading behaviors, ultimately providing valuable insights for market design and policy implications.

Other RL-based approaches for trading have shown the effectiveness of critic-only models. Ref. [35] employed RL to train an agent that buys Japanese government bonds when yields are high and sells when they are low. Similarly, ref. [36] used DDQN for executing trades based on price trailing, outperforming a benchmark trading RL model with a profit-to-loss ratio of 0.27 compared to 0.02. Meanwhile, ref. [70] applied a risk-averse RL model to NASDAQ data from the 2010 flash crash, outperforming risk-neutral counterparts while maintaining low trading costs.

Additional studies include [37], in which RL was combined with a cointegration approach, dynamically maximizing the Sortino ratio with superior performance over constant parameter models. Ref. [38] focused on maximizing immediate rewards by incorporating seasonality, achieving substantial improvements with limited order options. Lastly, ref. [71] compared DQN, DDQN, and Dueling DDQN on U.S. stocks, finding that DQN excelled in executing stock market strategies.

These studies underscore the versatility and potential of reinforcement learning, not only in optimizing trade execution but also in capturing the complexities of trader behavior and market dynamics within limit-order environments.

The 2024–2026 studies in Table 1 extend critic-only reinforcement learning beyond equities and foreign exchange. In peer-to-peer electricity and microgrid trading, refs. [52,55,60,62–64] applied Q-learning, DQN, or Double DQN variants, sometimes combined with quantum circuits, uncertainty forecasts, or blockchain-based settlement. The studies reported lower operating costs, higher simulated benefits, or improved trading utility, but nearly all were simulation-based and did not report the Sharpe ratio, maximum drawdown, or live-market performance. For cryptocurrency and multi-asset trading, refs. [53,54,56–59] reported higher headline returns through Double DQN, distributional Q-learning, ensemble learning, and recurrent state representations. Several of these studies included transaction costs, but their market periods, asset universes, leverage assumptions, reward functions, and statistical validation procedures differed considerably. For equity and futures markets, refs. [65–67] showed that wavelet denoising, LSTM-based feature learning, and Transformer-based state encoders can improve DQN performance against the buy-and-hold strategy and simpler RL baselines. Finally, ref. [61] incorporated GPT-4o Mini signals into the state of a DDQN agent and reported an improved mean Sharpe ratio, showing the emerging role of large language models as information sources for value-based RL trading. Collectively, these studies demonstrate continued methodological innovation, but their reported performance cannot be compared directly without a common framework for data selection, trading costs, risk measurement, validation, and reproducibility.

4.1.2. Q-Learning to Train Multiple Cooperative Agents

The application of Q-learning has been extended to train multiple cooperative agents for more sophisticated trading strategies. Ref. [23] proposed a system called MQ-Trader that employs Q-learning to train multiple agents, each with a specialized role but working together cooperatively. The system consists of two sets of agents: signal agents and order agents. Signal agents utilize long-term price history to predict future stock movements, while order agents focus on intraday stock characteristics to find the optimal bid and ask prices. Both sets of agents operate with different state representations that include various technical indicators. Using data from the Korean Stock Exchange (KOSPI200) from January 1999 to November 2005, the authors accounted for transaction costs, price slippage, and purchase price caps. The results demonstrated that MQ-Trader outperformed

five other variants in both returns and risk management, achieving asset growth that was 2.5 to 4.0 times higher.

Building on the concept of cooperative agents, ref. [72] introduced a framework in which agents specialize in different timeframes (short, medium, and long), emphasizing the temporal aspects of trading. This system employs a hierarchical information-sharing mechanism, in which insights from agents operating in longer timeframes are shared with those working in shorter ones, effectively enhancing the overall intelligence of the trading agents. The approach, inspired by the fractal nature of financial markets, demonstrated superior performance compared to single-agent models and benchmark trading strategies, highlighting the benefits of leveraging multiple cooperating agents in dynamic environments.

Further advancing the idea of multi-agent cooperation, ref. [73] presented the Efficient hierArchical Reinforcement learNing method for High-Frequency Trading (EarnHFT), tailored to cryptocurrency markets. EarnHFT uses a three-stage hierarchical RL framework to handle the complexities of long trading sequences and market fluctuations. The method enhances training efficiency by calculating an optimal action value (Q-teacher), diversifying RL agents to handle different market trends based on profitability, and training a router to select the most suitable agent for each market condition. The results showed that EarnHFT achieved a 30.0% improvement in profitability compared to other top baselines in cryptocurrency trading experiments.

Lastly, ref. [74] introduced the collaborative multi-agent reinforcement learning-based portfolio management system (CMPS), which employs deep reinforcement learning (DRL) to improve decision-making using financial indices and stock quotes. CMPS integrates traditional financial theories with a three-agent DRL model and a self-attention mechanism, enhancing the system's ability to predict market trends. Additionally, the authors developed a risk-free asset strategy (CMPS-RF) to further minimize risks. Testing on the China Shanghai Stock Exchange 50 Index revealed that CMPS outperformed other models in terms of returns, while CMPS-RF was most effective in risk management, achieving the highest Sharpe and Calmar ratios.

Ref. [58] proposed a multi-agent DQN framework for cryptocurrency option volatility trading, in which an OP-Agent determines long, short, or neutral option positions while an HR-Agent selects among risk-sensitive hedging strategies. Both agents are trained through temporal-difference learning using portfolio-value-based objectives. Historical BTC and ETH option backtests showed higher returns and risk-adjusted performance than the reported baselines, illustrating how specialized value-based agents can jointly address position selection and dynamic hedging.

The studies discussed above highlight the potential of Q-learning in training cooperative agents capable of adapting to diverse trading aspects, including market trends, different timeframes, high-frequency trading, and portfolio management. These examples demonstrate the effectiveness and adaptability of multi-agent reinforcement learning in addressing the complexities of real-world financial markets.

4.1.3. DRQNs and DQNs to Solve MDPs

A subset of studies has specifically explored the application of Deep Recurrent Q-Networks (DRQNs) and Deep Q-Networks (DQNs) to solve Markov decision processes (MDPs) in financial trading environments. Ref. [27] focused on enhancing the DRQN model by employing long short-term memory (LSTM) to address the sequential nature of trading data. The approach used a smaller replay memory and a longer training sample sequence, allowing the model to better capture market trends over extended periods. By incorporating an action augmentation technique to reduce random exploration and maintaining the agent's position for longer periods, the authors achieved a shorter learning cycle. The

approach led to an impressive 23.8% annualized return on 12 currency pairs from 2012 to 2017, showcasing the effectiveness of DRQNs with LSTM in adapting to financial markets.

Building on the theme of using deep reinforcement learning for MDPs, ref. [75] introduced a DQN-based framework with two key modules: feature extraction and decision-making. The feature extraction module processed stock data (open, high, low, and close prices) to generate a feature vector, which was then fed into the DQN-based decision-making module. The module employed a deep neural network (DNN) to estimate the Q-function, enabling the model to generate effective trading signals, even for less popular market trends. The proposed DQN framework outperformed existing state-of-the-art models, achieving a 12.4% increase in profits, further demonstrating the potential of DQN in financial decision-making.

In a comparative study, refs. [24,25] examined the performance of Q-learning against kernel-based reinforcement learning (KbRL) for automated trading. Their model introduced a third signal—the option to "stay out of the market"—alongside the standard buy and sell signals. Using the classical Sharpe ratio as a performance metric, the authors compared the Sharpe ratio calculated over $L \in \{5,22\}$ days for Q-learning and KbRL. Each test was repeated 1000 times with random initialization. Although transaction costs and other frictions were not considered, the results showed that Q-learning consistently outperformed KbRL, with trading-day variations having a significant impact on performance.

DRQN and DQN models are able to effectively solve MDP problems within the context of financial trading. Through the incorporation of advanced techniques such as LSTM, feature extraction modules, and additional trading signals, these models have demonstrated their capability to outperform traditional and state-of-the-art algorithms, achieve superior returns, and improve decision-making in financial markets.

4.1.4. Reward Functions Trained on DQNs, DDQNs, and DDDQNs

Reward functions have a significant impact on profitability and returns; the effectiveness of these functions varies across Deep Q-Network (DQN), Double DQN (DDQN), and Dueling Double DQN (DDDQN) models. Ref. [29] conducted a comparative analysis using the Sharpe ratio and profit as reward functions across DQN, DDQN, and DDDQN models. Using Bitcoin price data from December 2014 to June 2018 at 1 min intervals, they found that DDQN combined with the Sharpe ratio as the objective function delivered the highest average return during in-sample testing. The out-of-sample test yielded an average return of 8.0% with a standard deviation of 2.77, highlighting the potential of using risk-adjusted metrics as reward functions to enhance model performance.

Ref. [76] introduced DeepScalper, a framework designed for intraday trading that mirrors human decision-making by incorporating both micro- and macro-level market information, trend prediction, risk profiling, and trade execution. DeepScalper utilizes a Dueling Q-Network with action branching to train RL agents. It also features a novel reward function tailored for long-term horizons, alongside an encoder–decoder architecture for temporal embeddings and a volatility prediction module to assess market risk. The reward function includes a hindsight bonus, which favors holding assets over longer periods by overlooking short-term fluctuations. DeepScalper significantly outperformed the best baseline results, with improvements of 30.8%, 33.3%, 21.4%, and 7.5% for TR, SR, CR, and SoR, respectively.

In another study, ref. [77] examined the benefits of using a multi-objective (MO) deep reinforcement learning (DRL) approach compared to a single-objective (SO) reward strategy for single-asset trading. Their research demonstrated that MO algorithms were generally more stable than SO strategies, especially when dealing with sparse reward mechanisms.

Additionally, MO algorithms exhibited a superior ability to generalize across different reward structures, showcasing their adaptability and robustness in diverse trading scenarios.

Ref. [54] proposed RPADiRL, which extends IQN/DDQN by learning return-quantile distributions and applying probability distortion to support risk-sensitive long, flat, and short trading decisions. Using OHLC histories and current positions as state information, the framework optimizes trading performance after commission costs and achieved higher cumulative returns across the evaluated DJIA stocks than the reported technical-trading and reinforcement-learning baselines, with statistical comparisons also indicating significant improvements over selected DRL methods.

Reward function design in training DQN-based models demonstrates that risk-adjusted metrics, long-term reward considerations, and multi-objective strategies can significantly enhance trading outcomes.

4.1.5. RL for Dynamic Asset Allocation and Pair Trading

Reinforcement learning (RL) serves as a powerful tool for both dynamic asset allocation and pair trading strategies, with both approaches relying on RL's ability to adapt to evolving market conditions. While dynamic asset allocation employs RL to adjust portfolio weights in response to market shifts, DQN-based pair trading leverages RL to identify and capitalize on the price relationships between asset pairs.

Ref. [31] proposed a MetaPolicy (MP), which integrates stock selection and dynamic asset management to optimize portfolio returns. The framework combines pattern-based supervised learning (SL) for stock selection with dynamic asset allocation using stock fund ratios, resulting in more effective investment strategies. In tests conducted on the KOSPI index, the MP approach significantly outperformed five other asset allocation strategies, achieving a 258.0% profit, highlighting the potential of RL methods in enhancing portfolio performance.

Ref. [26] examined the performance of two RL methods, on-policy SARSA(λ) and off-policy Q(λ), for dynamic asset allocation. Using data from the S&P 500 and AGG Bond Index (1976–2016), the study compared agents designed to maximize portfolio returns or differential Sharpe ratios. The findings showed that a high-frequency learning approach using adaptive learning TD(λ) in a continuous action setting consistently outperformed both Q(λ) and SARSA(λ). This underscores the importance of adaptive learning in portfolio management, emphasizing the need for regular fine-tuning and reallocation. However, the increased complexity of continuous state–action models poses challenges, requiring larger datasets for better generalization.

In the context of pair trading, ref. [32] developed a two-phase optimization framework called Structural Break-Aware Pairs Trading (SAPT) to identify and trade co-integrated pairs. The first phase employs SWANet (Spread Wavelet-Aware hybrid Network) using a CNN and an LSTM to detect the probability of a structural break between stock pairs. In the second phase, a Structural Break-Aware Deep Q-Network (SADQN) determines the trading amount and stop-loss boundaries. The advanced approach outperformed existing pairs trading strategies by 456.0% in profit and 934.0% in the Sortino ratio, demonstrating its superior risk management capabilities. Similarly, ref. [33] applied a Double Deep Q-Network (DDQN) to learn pair trading strategies. The DDQN effectively captured reversion patterns in co-integrated stocks, enabling successful trading decisions. A negative reward multiplier was incorporated to promote more conservative trades, which enhanced the model's robustness in volatile markets.

Addressing the temporal aspects of pair trading, ref. [78] introduced the reCurrent Reinforcement lEarning methoD for paIrs Trading (CREDIT). CREDIT uses RL to dynamically learn a risk-aware agent capable of exploiting long-term trades, incorporating a Bi-GRU (bidirectional gated recurrent unit) to capture long-term price movement patterns between

correlated assets. By integrating a risk-aware reward system, CREDIT effectively balances profit potential with risk minimization. The dynamic approach achieved profitability when applied to real-world data, demonstrating the advantages of considering temporal dependencies and risk in pair trading strategies.

4.1.6. Hybrid Value-Based RL with Evolutionary and Fuzzy Methods

Hybrid reinforcement learning (RL) models have shown promise in combining traditional techniques with advanced machine learning methods to enhance trading strategies. Ref. [30] introduced an adaptive box-normalization (ABN) trading strategy based on box theory, using support vector regression (SVR) to detect patterns within boundary boxes and a genetic algorithm (GA) for feature optimization. The model dynamically adjusts boundary boxes according to stock price volatility, and Q-learning is employed to construct trading strategies based on these states. When tested on two U.S. stock indices (1999–2018), the ABN strategy achieved an average return of 38.8%.

Other studies, such as [41], also incorporated GAs to select optimal state representations from eight technical indicators, demonstrating that evolutionary RL consistently outperforms standard RL across different trading frequencies and transaction costs. Notably, GA-enabled models were profitable up to 4 basis points (bp) of slippage, while standard RL models were unprofitable beyond 1 bp. This suggests that GAs enhance the robustness of RL in dynamic trading environments.

Ref. [42] evaluated a hierarchical neuro-fuzzy system combined with RL called RL-HNFP to trade EURYEN (1999–2006), finding that RL-HNFP could identify long-term trading strategies and perform more profitably with fewer trades than other models. Similarly, ref. [43] proposed an adaptive network fuzzy inference system (ANFIS) with RL capable of detecting trend changes and adjusting trading decisions, outperforming market returns by 50 percentage points across five U.S. stocks.

Ref. [44] developed the Parallel Multi-Module Deep Reinforcement Learning (PMMRL) algorithm, which combines fully connected (FC) layers and long short-term memory (LSTM) to extract environment features before passing them to a DQN. The approach resulted in higher profits and lower drawdowns compared to other state-of-the-art methods.

Lastly, ref. [45] proposed an end-to-end model named DRLUTrans, which integrates DRL with Transformer layers and a U-Net architecture for single-stock trading strategies. The Transformer captures complex dynamic patterns, while the U-Net's skip connections blend long- and short-term features. The model can output trading actions and action weights, allowing it to manage the proportion of trades and reduce investment risk. When tested on ten stocks, DRLUTrans outperformed baseline models, generating higher profits with reduced risk.

In parallel with RL-based hybrid trading systems, recent work has explored hybrid metaheuristic optimization algorithms for stock trend prediction on the Indonesian Stock Exchange, combining improved bacterial foraging optimization with an elitist whale optimization scheme [79]. In the study, the metaheuristics were used to optimize the parameters of a supervised prediction model, yielding enhanced trend-forecasting accuracy and improved error metrics, rather than directly learning a sequential trading policy. This highlights that optimization-driven hybrids and RL-driven hybrids address related but distinct aspects of the trading problem: the former focus on offline optimization of forecasting models, while the latter focus on online optimization of trading decisions through interaction with a market environment.

Ref. [53] introduced CSI-DDQN, combining Double DQN with sequential-client learning, self-imitation through KL regularization, and volatility-weighted model aggregation. Ref. [56] developed FineFT, an ensemble-DQN framework that selectively updates experts using

temporal-difference errors and employs VAE reconstruction loss for out-of-distribution routing in cryptocurrency markets, while ref. [57] combined XGBoost feature selection with recurrent DDQN architectures for Bitcoin and Ethereum trading. Other hybrid designs integrate specialized temporal and feature representations: ref. [65] combined wavelet denoising and attention–BiLSTM representations with DQN for S&P 500 futures; ref. [66] integrated denoising, PCA, LSTM state encoding, and DQN for agricultural-related equities; and ref. [67] combined LSTM, Transformer, and LORS components with a DQN value-learning head. Ref. [59] further explored open-ended Bitcoin trading using a Rainbow/DD-DQN agent with hybrid recent-and-random experience replay. Collectively, these studies indicate a continuing shift from standalone DQN architectures toward richer representation learning and adaptive training mechanisms.

4.1.7. RL with Different Data Sources

Reinforcement learning (RL) models have been enhanced by incorporating various data sources such as sentiment analysis, news articles, and technical indicators, demonstrating improved trading strategies.

Ref. [80] explored the impact of investor sentiment on market direction, developing a sentiment reward-based system using the Thomson Reuters News Analytics (TRNA) database to filter signals that trigger positive or negative market responses. The model outperformed benchmark strategies, proving particularly effective in volatile markets, although it was sensitive to transaction costs. Similarly, ref. [46] utilized tweets translated into sentiment scores via Stanford CoreNLP, combining these with portfolio information and technical indicators in a Q-learning algorithm. The Q-learning model outperformed both machine learning and logistic regression alternatives.

Expanding on sentiment analysis, ref. [47] introduced a multimodal DRNN (MDRNN) model that integrates RL, sentiment analysis, and multimodal learning by incorporating price information and news sentiment. The approach yielded improved profit margins, with multimodal agents outperforming standard agents by 44.7% for IT companies and by 38.5% across other sectors.

Incorporating more diverse data sources, ref. [81] developed a multi-type data framework that integrates stock data, technical indicators, and candlestick charts. An LSTM network extracts temporal features from stock data, while a CNN and a BiLSTM network process candlestick chart features. The fused features are then used in an RL module to make trading decisions. The approach demonstrated superior trading performance, achieving greater profits and a higher Sharpe ratio compared to singular methods, as the combination of features offered a deeper representation of market trends.

Ref. [61] augmented a DDQN equity-trading agent with monthly GPT-4o Mini guidance encoded as an uncertainty-weighted observation feature, allowing language model-derived information to influence state representation without modifying the underlying DDQN optimization or reward function. Across six U.S. equities, the augmented agent reported higher mean Sharpe ratios and lower average drawdowns than the standard DDQN baseline, although the language model signals were precomputed and the evaluation was restricted to a relatively small equity universe.

4.1.8. RL for Energy Trading

Refs. [82,83] applied RL to the role of a retail energy broker in deciding the optimal price tariff to encourage participation in the energy market. Ref. [84] designed an autonomous broker that maximizes profit by setting the electricity tariff based on supply and demand. The proposed model successfully mitigates transmission capacity fees, procures energy at lower prices, estimates parameters of the MDP, and optimizes tariffs. The model

was the runner-up in the 2018 power trading agent competition. Refs. [85–87] proposed an energy-trading framework that applies Nash equilibrium (NE) in order to maximize profit under limited information. The authors demonstrated how NE could accommodate a variety of conditions under which microgrids can operate, improve utility, and benefit a power plant schedule.

Ref. [52] investigated quantum reinforcement learning based on a variational quantum circuit and state-value estimation for peer-to-peer electricity trading, reporting faster convergence than classical RL in simulation. Ref. [55] employed DQN to jointly optimize blockchain and clustering parameters for vehicle-to-vehicle energy trading, while ref. [60] proposed an improved multi-agent Nash-Q approach for energy exchange among interconnected microgrids. Forecast uncertainty has also been incorporated explicitly: ref. [62] combined probabilistic forecasts with multi-agent DQN for peer-to-peer energy and battery trading, and ref. [64] proposed SA-DDQN to account for photovoltaic uncertainty and storage degradation when optimizing microgrid trading quantities and prices. Ref. [63] further used multi-agent tabular Q-learning under centralized learning and decentralized execution for EV charging, discharging, and dynamic station pricing. Although these studies generally reported improvements in simulated economic or operational outcomes, most were simulation-based and therefore provided limited evidence of robustness under real-world energy-market conditions.

4.1.9. Graph-Based Methods in RL

Graph-based methods have shown promise in enhancing reinforcement learning (RL) for trading by capturing complex relationships within financial data. Ref. [88] introduced an end-to-end network named Optimal Action Space Search (OASS), which marks a shift from conventional trading approaches by exploring decision-making across an entire graph via probabilistic dynamic programming. The method separates the state into two components: the environment state and the position state, with the latter modeled as a directed acyclic graph. This structure allows OASS to operate with a simple reward mechanism, making it more stable and reliable compared to other deep reinforcement learning (DRL) models. The model demonstrated superior performance over benchmark algorithms for both high- and low-frequency trading.

Ref. [51] explored a novel approach for stock price prediction using a combination of deep reinforcement learning (DRL), knowledge graphs, and sentiment analysis from social media data. The authors developed a system that integrates the sentiments of influential users (influencers) with financial data to forecast stock price movements. The model incorporates a knowledge graph to identify relationships between companies, key figures, and other relevant entities, which is used to enhance the accuracy of sentiment analysis. The DRL methodology utilizes deep Q-learning to guide investment decisions by modeling the stock market as a Markov decision process. The system was tested using Turkish stocks from Borsa Istanbul, and the experimental results demonstrated improved prediction accuracy and profitability compared to traditional methods. Specifically, the model's incorporation of social media sentiment and community awareness via influencers significantly boosted prediction performance, making it a robust tool for financial forecasting.

Graph-based methods in reinforcement learning for trading leverage the ability to capture complex relationships and interactions within financial data, enhancing decision-making processes. The primary takeaway from both studies is that incorporating graph structures, such as directed acyclic graphs in OASS and knowledge graphs with sentiment analysis, significantly improves trading performance by offering more stable, reliable, and informed trading strategies compared to traditional models.

4.1.10. RL with Sentiment Analysis and Behavioral Finance

Ref. [89] introduced the Momentum Investment Twin Delay Deep Deterministic (MITD3) algorithm by combining the momentum investment strategy from behavioral finance with deep reinforcement learning. The approach uses a momentum investing critic (MI-critic) to compute Q-values that encourage or penalize past winners or losers, respectively. The MITD3 algorithm, incorporating a cross-time module for evaluating historical performance, showed significant improvement over other state-of-the-art deep reinforcement learning models using real-world stock market data. It effectively combines finance theory with deep learning to create sophisticated and intuitively explainable stock trading strategies.

Similarly, ref. [49] proposed a novel deep reinforcement learning framework that integrates price and sentiment data for cryptocurrency trading. By utilizing multimodal financial embeddings, the model efficiently integrates various data sources, thereby enhancing performance and explainability. Both studies demonstrated how the combination of various modalities of information in deep reinforcement learning can optimize financial trading strategies.

4.1.11. Network Analysis of Value-Based Methods

Figure 3 presents a network analysis of the articles referenced by the reviewed studies. The reviewed articles heavily cite core research in the area of reinforcement learning as a basis for value-based methods. Central nodes include foundational RL works like "Playing Atari with Deep RL" [90] and "Human-Level Control through Deep Reinforcement Learning" [21], indicating strong methodological influence from early deep Q-learning approaches. References to finance-specific studies like "Deep Direct RL for financial signal representation and trading" [48] and "Performance functions and RL for trading systems and portfolios" [91] indicate an emphasis on Q-learning adaptations to different trading environments.

4.2. Policy-Based Methods

Policy-based methods involve agents making decisions or taking an action, a, based on a given state, s. These approaches are particularly useful in high-dimensional or continuous action spaces. In policy-based methods, actions are modeled as a probability distribution, making them stochastic; a stochastic policy provides probabilities for each action, while a deterministic policy results in a single action. This probabilistic approach adds randomness, promoting exploration and allowing flexibility in action selection. The agent's goal is to optimize decisions by identifying actions that maximize cumulative future rewards. After an action is executed, the environment responds with a new state and a reward. The reward is evaluated based on the action and state, leading to an update of the policy function in the next iteration to improve future rewards. This feedback cycle is illustrated in Figure 4.

Policy-based methods encourage actions with better rewards by adjusting the probability distribution. Policy gradient (PG) directly learns the policy through a parameterized function $\pi_\theta(a|s)$ and updates θ using gradient-based or gradient-free optimization to minimize loss.

$$L^{PG}(\theta) = \mathbb{E}_t[log\pi_\theta(a_t|s_t)A_t] \quad (5)$$

where π_θ is a stochastic policy and $A_t = Q(s,a) - V(s)$ is the advantage function that estimates the advantage of taking a particular action in a given state compared to the expected value of taking an action according to the current policy. Table 2 presents a summary of the reviewed publications that explore policy-based functions for quantitative trading.

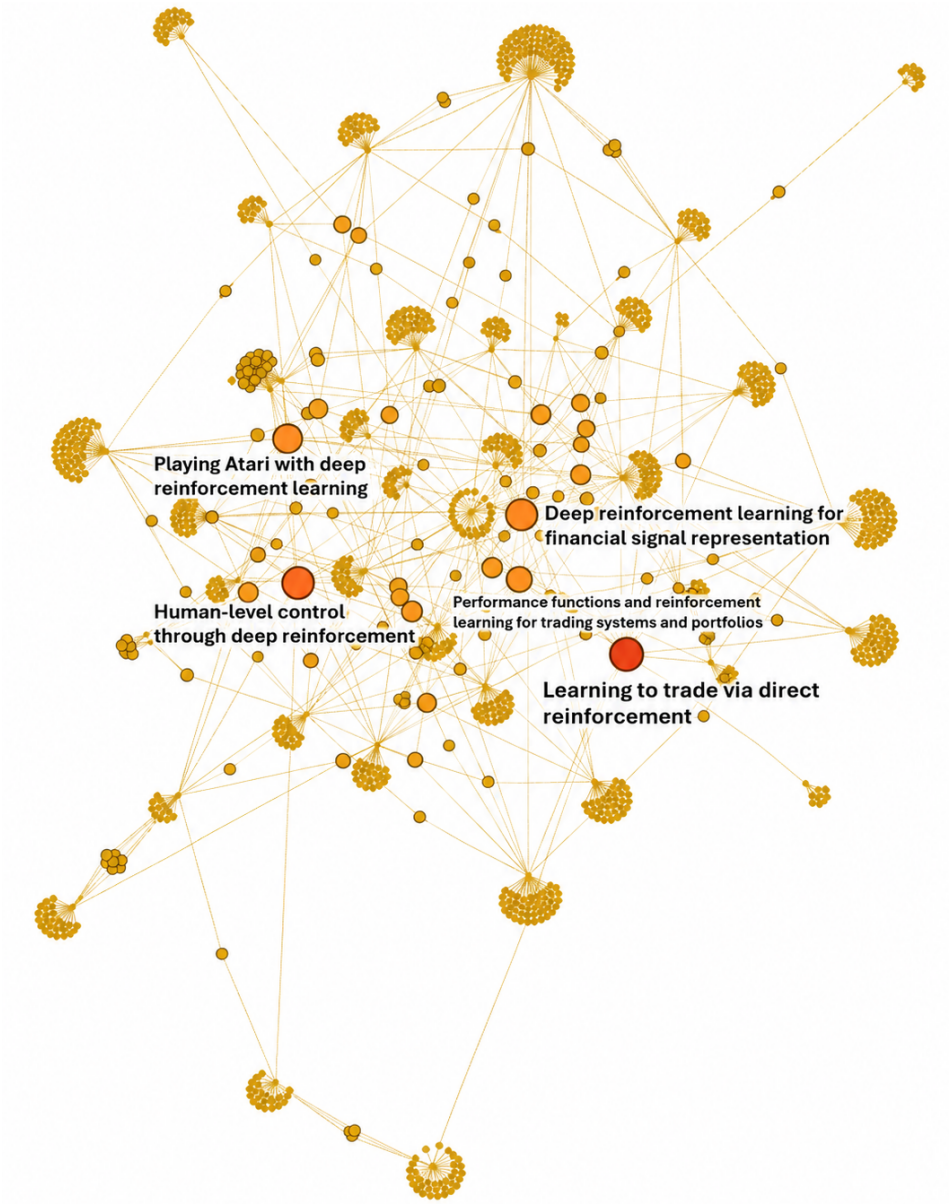


Figure 3. Network analysis of value-based studies.

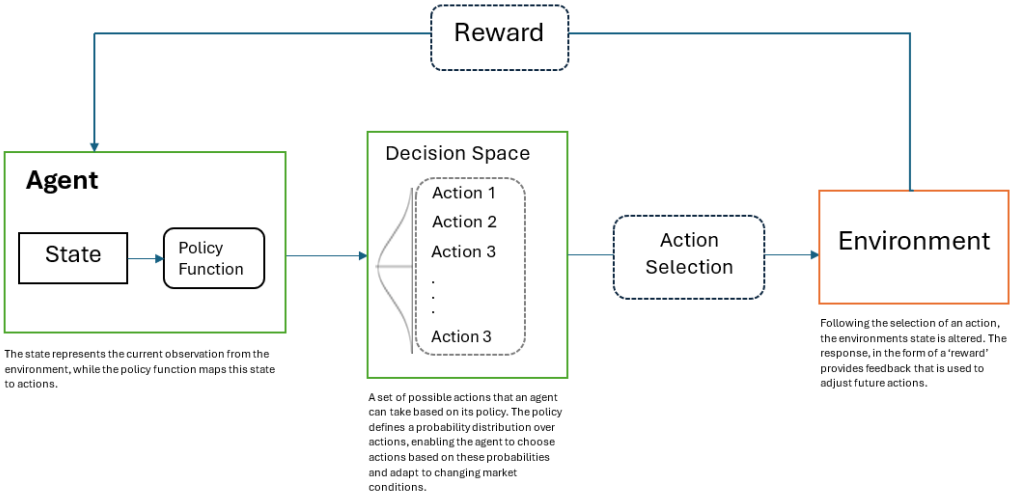


Figure 4. Feedback loop for a policy function.

Table 2. Summary of policy-based methods.

Author	Data	Summary and Key Findings	Results
[91–94]	Simulated data, Forex, stocks, and T-Bills	RRL that uses the differential Sharpe ratio as a reward function outperforms RRL that maximizes profit and supervised models that minimize MSE	RRL outperformed Q-learning with a Sharpe ratio of 0.83 vs. 0.63
[95]	International stocks	Robust performance from a model that also mitigates large losses (large drawdowns)	Cumulative risk-free return was 3.75
[96]	International stocks	Augmented RRL model with a regime-switching property such as volatility	TRRL outperformed RRL in terms of Sharpe ratio
[97]	U.S. stocks + transition variables	RSRRL with various transition variables yields a higher Sharpe ratio in the in-sample test	RRL performed better in the out-of-sample set
[98]	Swiss stocks + transition variables	Use of P/E as a transition variable produces a higher Sharpe ratio	Sharpe ratio was greater at the 5% significance level
[99]	Chinese futures contracts	MODRL with two objective functions maximizes average profits and reduces risk	Outperformed existing benchmarks in terms of Sharpe ratio and average profits
[100]	Cryptocurrencies	Deterministic policy gradient (DPG) using Ensemble of Identical Independent Evaluators (EIIE) outperformed benchmarks in accumulated portfolio values, Sharpe ratio, and maximum drawdown	Recorded a 4-fold return in 50 days
[101]	Stocks	PG model successfully reproduces around 80% of trading decisions. Dynamic ϵ-greedy yields better results	Successfully cloned 80% of trading decisions
[102,103]	U.S. stocks	Use of a hybrid model combining RL with DL shows substantial returns and improved Sharpe ratios	The combination of RL with DL showed promising results
[104]	Forex	A single-layer network achieves overall better results than a 2-layer network	Achieved 28.3% $\pm$ 1.6 in profit and 3.0 $\pm$ 0.16 in Sharpe ratio
[48]	Chinese stock index futures, silver, and gold	Use of an RNN for automatic feature selection and fuzzy learning for reducing uncertainty in time series generates good profits	Average Sharpe ratio above 10.0
[105,106]	Cryptocurrencies	Modified DPG optimizes the reward function directly instead of using a Q-function, helping prevent high variance in the data and allowing for a more stable model. Dynamic parameter optimization has also been shown to have a competitive performance	Outperformed most benchmark models and achieved high Sharpe ratios
[107]	U.S. stocks	RRL with stop-loss, assessed via the Calmar ratio, performs well and is able to make a profit even when the transaction cost is 25 bps	Achieved annual returns of 29.65 at 0 bps and 3.71 at 25 bps
[108]	U.S. stocks	Particle swarm optimization (PSO)-based RRL with stop-loss exhibits superior performance over the benchmark models	Achieved an annual return of 30.42 at 20 bps
[109]	Forex	Fully automated trading system with a layered structure, consisting of an ML algorithm, a risk management method, and an optimization layer	The model achieved an annual profit of 26.0%
[110]	Forex	Explores a radial basis function network, comprising hidden processing units modeled as Gaussian mixture models, shown to learn a risk position in an online context via transfer learning	The model achieved a net compound return of 9.3% over 7 years
[112,113]	Stocks	Optimizes both returns and risk by integrating a 'dependency factor' into deep reinforcement learning (DRL) for portfolio management	Achieved higher portfolio returns than existing models

Table 2. *Cont.*

Author	Data	Summary and Key Findings	Results
[111]	Stocks + indicators	A stochastic adaptation of RRL model is applied to daily, weekly, and monthly intervals of data to show that RRL can outperform the BH strategy for high-frequency data	RRL outperformed the BH strategy 98.5% of the time on daily data
[114]	Stock index	Employs BiLSTM–Attention networks to represent complex and volatile stock markets	Outperformed baseline strategies
[115]	Stock index	Employs a multi-agent system with a continuous action space and replaces gradient descent with a genetic algorithm for parameter optimization	Achieved annual returns of $27.9 \pm 9.5\%$
[73]	Stock index	RL-based NQT framework	Demonstrated superiority over state-of-the-art NQT methods
[116]	Cryptocurrencies: BTC/USDT, ETH/USDT, DOT/USDT, and BNB/USDT	Hierarchical policy-based framework: a VQ encoder–decoder distills single-trade DP demonstrations into ten discrete archetypes; a horizon-level policy selects an archetype, and a step-level adapter makes at most one regret-aware override. Actions set short, flat, or long target positions; rewards are net-value changes after LOB execution costs.	In backtesting with 0.02% commission, the model reported the following results for BTC: TR, 76.25%; ASR, 2.66; and MDD, 23.75%; the corresponding values for DOT were 147.48%, 3.19, and 16.26%.
[117]	Daily data for GOOGL, META, MSFT, NVDA, and TSLA	AlphaQuanter uses GRPO-trained LLM policies to iteratively choose financial-data tools and issue buy/sell/hold decisions. Rewards combine directional outcome, response-format and tool-use scores.	Backtests applied a 0.1% transaction fee and 0.9 capital utilization. AlphaQuanter-7B produced an average ARR of 34.94%, a Sharpe ratio of 0.65, and an MDD of 24.93% over five stocks; the corresponding values for the buy-and-hold strategy were 12.90%, 0.57, and 31.13%.
[118]	CSI 500, S&P 500, STOXX Europe 600, and Nikkei 225	MAML-based MetaPolicy adapts to regime-specific long/short/hold trading tasks. Cognitive level-k and hierarchy models augment beliefs about opponent actions; GPT-3 generates candidate strategies, while an opponent pool simulates heterogeneous participants.	Reported annual returns were 46.5–53.7%, Sharpe ratios were 2.08–2.45, and MDDs ranged from -10.2% to -12.1% across four indices, exceeding EIIE and ADRL. Transaction costs were 0.1% plus 0.05% of slippage per trade.
[119]	TXF and E-mini S&P 500 futures	Hierarchical deterministic policy gradient manager allocates continuous capital weights across five rule-based local traders: Bollinger Bands, SAR, RSI, dual thrust, and MACD. Fuzzy price/performance states and CNN features guide allocation; reward is weighted strategy profit less reallocation cost.	In TXF period 1, the system reported an average ARR of 69.66%, a Sharpe ratio of 1.63, a Sortino ratio of 1.64, and an MDD of 32.94%, outperforming the baselines. The transaction cost was four futures points.

4.2.1. Reward Functions for RRL

Early work by [91,92] sought to automate trading systems, motivated by the limitations of supervised learning (SL), which aims to minimize mean squared errors (MSEs). While minimizing MSE is logical, it fails to align with the primary goal of trading systems: maximizing profits, economic utility, or risk-adjusted returns. To address this, ref. [91] proposed a single-layer recurrent neural network (RNN) known as recurrent reinforcement learning (RRL). The model employs an adaptive policy search algorithm that allows it to derive policies without relying on value-function learning, thus avoiding Bellman's curse of dimensionality. In simulations with a 0.5% transaction cost, the RRL model maximized

the differential Sharpe ratio, produced more consistent results, and achieved better mean returns compared to an RRL model focused solely on profit maximization. Both RRL models significantly outperformed the SL model that aimed to minimize MSE.

Building on this foundation, ref. [93] applied the RRL algorithm to maximize the differential downside deviation ratio using USDGBP data from 1996. The model achieved an annualized return of 15.0% and a Sharpe ratio of 2.3. However, the authors noted that their simulation assumptions—such as continuous trading and not accounting for low-liquidity periods—might not reflect real market conditions. In a final test involving S&P 500 and Treasury Bills data from January 1950 to December 1994, the authors compared RRL to Advantage Updating Q-learning, finding that both models exhibited good predictability and outperformed the buy-and-hold (BH) strategy. Analysis of 45 years of data revealed that both models adjusted their trading strategies to favor Treasury Bills during economic downturns. Although RRL executed fewer trades than Advantage Updating Q-learning, it achieved a higher Sharpe ratio of 0.83 compared to 0.63 for Q-learning and demonstrated computational efficiency and interpretability.

Ref. [94] employed a Gradient Monte Carlo (GMC) algorithm to evaluate stocks in the DOW30 index, testing RL algorithms that held purchased stocks for 24 h before selling. Despite the simplicity of this strategy, the GMC algorithm outperformed growth funds between April and June 2021, achieving a Sharpe ratio comparable to RNN algorithms. Ref. [95] developed a similar model aimed at minimizing large drawdowns. Using major stock indices from 1992 to 2007, the results indicated that eight out of nine stocks generated positive cumulative risk-free returns, with a total portfolio risk-free return of 3.752.

These studies generally found that RRL models effectively enhance trading strategies by aligning reward functions with profit maximization and risk management, outperforming traditional supervised learning approaches and demonstrating adaptability to varying market conditions. It is also noteworthy that the models showed strong predictive performance and computational efficiency, highlighting their potential for real-world financial applications.

4.2.2. Extensions to RRL

Ref. [120] argued that the original RRL model is too simplistic and fails to capture the nonlinearities present in financial data. To address this limitation, they developed Threshold RRL (TRRL/RSRRL), which incorporates regime-switching properties such as volatility, inspired by economic cycles that cause structural changes in data. TRRL operates with two independent RRL systems, generating two sets of weights for different regimes. When tested on four European stock indices over a period reflecting various financial crises (December 1999–October 2008), TRRL consistently outperformed RRL in terms of the Sharpe ratio. The advantage stems from TRRL's ability to adapt to changing market cycles more swiftly due to its dual weight sets. It is worth noting that these tests assumed a frictionless environment with zero transaction costs.

In a follow-up study, ref. [120] investigated the impact of different transition variables, such as volatility, volume, and the rate of information arrival, on various RSRRL models. Using 15 stock indices from the DJIA (April 2000–September 2009), they found that RSRRL achieved a higher Sharpe ratio than RRL for in-sample data but underperformed in out-of-sample scenarios. This discrepancy suggests that while RSRRL is effective when market regimes are correctly identified, RRL proves more robust in entirely new market conditions due to its broader adaptability.

Expanding on automated transition function models, ref. [98] introduced RSRRL 2.0, which utilizes utility maximization to dynamically drive the transition function, thus eliminating the need for manual updates. Evaluating various transition variables like

volume, relative strength index, price-to-earnings (P/E), conditional volatility, and moving average price using data from 20 Swiss stocks (April 2009–September 2013), RSRRL 2.0 delivered statistically significant improvements in the Sharpe ratio compared to the original RSRRL. Notably, using P/E as the transition variable resulted in the highest Sharpe ratio within the sampled data, although this may vary across different datasets.

Ref. [116] proposed a hierarchical policy framework in which trade demonstrations are distilled into discrete archetypes, with a horizon-level policy selecting an archetype and a step-level adapter allowing regret-aware overrides for short, flat, or long positions. Ref. [118] further introduced a MAML-based meta-policy for regime-specific trading tasks, combining adaptive policy learning with cognitive opponent modeling and GPT-generated candidate strategies. These approaches illustrate a broader shift toward hierarchical and meta-adaptive policies that can respond to changing market regimes rather than relying on a single fixed trading policy.

4.2.3. RL Combined with DL

Several studies demonstrate the effective integration of RL with various deep learning architectures; these works yield promising results across different financial contexts. For instance, ref. [99] introduced a multi-objective RRL (MODRL) method enhanced by an LSTM network, focusing on maximizing profits while minimizing risk. Using data from three highly liquid Chinese futures contracts, MODRL outperformed both RRL and the buy-and-hold strategy in terms of the Sharpe ratio and average profits. The proposed architecture demonstrated the advantages of combining RL with a DL algorithm such as LSTM for handling multiple objectives. In a similar study, ref. [114] introduced a deep State–Action–Reward–State–Action (SARSA) model combined with BiLSTM–Attention networks. The model performed exceptionally well across datasets like DJI and S&P 500. The advantage of combining LSTM with RL is that LSTM is able to capture long-term dependencies and patterns within sequential data, allowing the model to make informed decisions on future buy/sell options.

Ref. [100] developed the Ensemble of Identical Independent Evaluators (EIIE) framework, tested with CNN, RNN, and LSTM architectures, to evaluate securities based on historical data. Despite assuming zero market impact and slippage, the model achieved significant returns even with a 0.25% commission rate, emphasizing the robustness of ensemble methods in leveraging deep learning for RL trading strategies. In another example, ref. [101] utilized a six-layer CNN trained with a policy gradient (PG) algorithm to replicate professional trading strategies, achieving around 80.0% accuracy in reproducing trading decisions on the Taiwan stock index. The study highlights how RL can emulate complex human trading behaviors when combined with sophisticated deep learning models. In contrast, ref. [104] explored network depth in RL models, finding that a single-layer neural network outperformed a two-layer network in high-frequency currency trading, suggesting that simpler models can sometimes be more effective in certain contexts.

Ref. [102] introduced a time-driven feature-aware joint deep reinforcement learning (TFJ-DRL) model, integrating a GRU-based deep learning perception module with an RL decision-making module. The model demonstrated higher profits and Sharpe ratios on U.S. stocks, highlighting the potential of combining GRU with RL to process historical trading data and technical indicators effectively. In a similar study, ref. [103] applied DRL techniques to portfolio management by combining deep learning for asset weight prediction with RL for decision-making. The model outperformed traditional portfolio management methods, particularly in virtual currency markets.

Ref. [73] focused on addressing challenges in news-driven quantitative trading (NQT) through reinforcement learning (RL). Traditional NQT methods typically operate in a

two-step approach, involving market analysis followed by trading decisions based on financial predictions. However, the limitations of these predictions often hinder trading success. To overcome these limitations, the authors proposed a novel RL-based NQT framework, SpotlightTrader, which optimizes trading decisions by constructing a decision trajectory that maximizes profits. Key contributions include modeling a state trajectory driven by spotlight news (news with significant market impact) and implementing an offline pretraining and online fine-tuning training pipeline to adapt to non-stationary markets. The proposed approach demonstrated superior performance compared to state-of-the-art NQT methods, with improved profitability and risk management. The work highlights the potential of trajectory optimization and spotlight-driven state modeling to enhance trading decision-making in complex financial environments.

To address markets that are characterized by swift dynamics and uncertainty, ref. [121] proposed the fusion of two established financial trading strategies—constant proportion portfolio insurance (CPPI) and time-invariant portfolio protection (TIPP)—with the multi-agent deep deterministic policy gradient (MADDPG) framework. Tested on real-market data, both methods showed better performance than traditional strategies such as DQN-based approaches.

These studies highlight the effectiveness of integrating RL with deep learning techniques, such as LSTM, CNNs, GRUs, and BiLSTM, to enhance trading strategies and improve profitability and risk management. Deep learning algorithms adapt well to dynamic market conditions; they are especially effective within ensemble model architectures, which leverage the strengths of each base learner.

Ref. [117] introduced AlphaQuanter, in which a GRPO-trained LLM policy iteratively selects financial-data tools and produces buy, sell, or hold decisions. Its reward combines trading-direction outcomes with response-format and tool-use objectives, enabling the language model to learn both decision-making and information-acquisition behavior. Backtesting across several large-cap U.S. equities showed improved average annual return and risk-adjusted performance relative to the buy-and-hold strategy.

4.2.4. RL with Feature Selection

The original RRL [91,92] consists of a single-layer RNN and is dependent on a robust set of features for good performance. Rather than pre-selecting features, ref. [48] proposed the use of a recurrent deep neural network to automate feature selection. Using China-based stock-IF contracts, silver, and gold, they compared the performance of RDNN to RRL, SCOT (a model proposed by the same authors in a previous paper), and FDDR (fuzzy logic-based RDNN). The results showed that FDDR generated good results and was able to achieve an average Sharpe ratio above 10.

4.2.5. Optimization for Reward Functions and Parameters

Reward functions and parameters play a crucial role in shaping an agent's decision-making and learning processes, significantly impacting its stability and adaptability to market changes. For instance, ref. [105] introduced a framework utilizing a slightly modified deterministic policy gradient (DPG) to manage a multi-asset portfolio of 12 high-volume cryptocurrencies from 14 March 2016 to 3 July 2016. By opting for DPG to directly optimize the reward function instead of relying on the Q-function, the authors effectively mitigated high data variance and enhanced model stability. They employed convolutional neural networks (CNNs) to generate portfolio weights, thus defining the agent's optimal actions. Their model outperformed several benchmark strategies, including Uniform Buy and Hold (BH), Uniform Constant Rebalanced Portfolio (UCRP), Universal Portfolio, Best Stock, Online Newton Step, Passive Aggressive Mean Reversion (PAMR),

and others. Based on these findings, the authors recommended reducing the time interval between the training and backtesting sets to ensure the validity of learned patterns.

Similarly, ref. [106] developed a high-frequency trading framework that addresses limitations in existing price trend prediction and adaptive trading strategies. Their approach incorporates a rolling model training method for accurate price trend predictions, coupled with a dynamic parameter optimization algorithm based on reinforcement learning. The framework takes trading constraints, such as limited short positions, into account while optimizing parameters in real time using a reward-enhanced upper confidence bound (UCB) selection algorithm. The experimental results showcased competitive performance and promising profitability in the Chinese stock market, with the framework's design allowing for adaptation to changing market conditions, potentially extending its application to other secondary markets.

Additionally, ref. [107] applied recurrent reinforcement learning (RRL) with various objective functions across five portfolios from 1 January 2011 to 31 December 2015, aiming to create equally weighted and variable-weighted portfolios. They found that the Calmar ratio consistently outperformed both the Sharpe and Sterling ratios when transaction costs were considered. Notably, RRL using the Calmar ratio in conjunction with stop-loss strategies consistently outperformed hedge funds.

These studies emphasize the importance of optimizing reward functions and parameters to enhance trading performance, with [105] focusing on stability through modified DPG and [106] emphasizing adaptability and profitability through dynamic optimization.

4.2.6. Combining RL with Algorithms (PSO and ARL)

Ref. [108] integrated the particle swarm optimization (PSO) algorithm with recurrent reinforcement learning (RRL) to enhance asset selection and weight optimization. The particle swarm optimization technique mimics the social behavior of birds and fish, enabling a swarm of particles to iteratively adjust their positions in a search space to identify optimal solutions. By analyzing S&P 100 data from January 2011 to December 2015, the study demonstrated that incorporating a stop-loss mechanism significantly outperformed benchmark strategies. The researchers reaffirmed that the Calmar ratio is more effective than the Sharpe ratio and portfolios based on mean-variance principles, even in high-transaction-cost scenarios.

In a different approach, ref. [109] developed a fully automated trading system that layers RRL with risk management and dynamic optimization, termed adaptive reinforcement learning (ARL). The first layer focuses on risk management by limiting high-uncertainty trades, while the optimization layer fine-tunes hyperparameters to align with the trader's risk profile. Utilizing EUR/USD data, ARL achieved an impressive annual profit of 26.0%. Combining RL with PSO and ARL algorithms allows for improvements in asset selection, risk management, and overall trading performance.

Ref. [119] employed a deterministic policy gradient manager to allocate continuous portfolio weights across five rule-based trading strategies, including Bollinger Bands, SAR, RSI, Dual Thrust, and MACD. Fuzzy market and strategy-performance states together with CNN-derived features guide the allocation policy, while the reward balances strategy profits against reallocation costs. The resulting framework reported stronger risk-adjusted performance than the evaluated baselines in futures-market experiments.

4.2.7. Transfer Learning and Actor-Only Models in RL

Actor-only models have shown promising results for RL in trading. Ref. [111] conducted a comparative study of recurrent reinforcement learning (RRL), genetic programming (GP), and the buy-and-hold (BH) strategy across various data intervals. The findings revealed that RRL consistently outperformed BH 98.5% of the time in daily data, while

GP surpassed BH 90.0% of the time in monthly data. In a similar vein, ref. [122] examined the performance of a supervised learning (SL) approach based on a ResNet architecture against standard RL algorithms, including DQN, A2C, and Bootstrapping Thompson Sampling, within the cryptocurrency market. Although the SL model outperformed these RL algorithms, it is important to note that the authors did not modify the architectures of the available RL algorithms.

Ref. [110] explored the application of transfer learning for RL trading agents. The model utilizes a radial basis function network to enable online transfer learning for a direct recurrent RL agent, tested on Forex data from 36 currency pairs between 2010 and 2021. The initial one-third of the dataset was used for training, while the remaining data served as the test set, allowing the model to learn continuously during the testing phase. The results demonstrated that online transfer learning significantly improved risk-adjusted returns, with the agent achieving a compound return of 9.3% over the 7-year test period.

4.2.8. RL for Risk and Portfolio Management

Recent advancements in risk and portfolio management highlight the integration of reinforcement learning (RL) techniques to optimize both returns and risk. Ref. [112] introduced the Risk-balanced Deep Portfolio Constructor (RDPC), an end-to-end Transformer-based model that emphasizes the balance between risk and returns. By incorporating a hard attention mechanism and a price-volatility-regulated reward function, the RDPC was designed to learn and optimize the relationship between asset pairs, effectively mitigating risks within trading strategies.

Building on the theme of enhancing reward systems, ref. [113] developed a deep reinforcement learning (DRL)-based portfolio management system that integrates a novel feature called the "dependency factor." This factor considers the performance of holding companies and their shareholdings in other firms, improving the DRL model's reward system. Utilizing the deterministic policy gradient (DPG) algorithm, the framework aims to optimize portfolio allocation within a continuous action space. The experimental results indicated that the inclusion of the dependency factor significantly enhances portfolio management and optimization, demonstrating the effectiveness of integrating additional market-related features into reinforcement learning strategies.

Further extending the application of RL in portfolio management, ref. [115] proposed an innovative approach that incorporates the Sharpe ratio and environmental, social, and governance (ESG) scores into the utility function for portfolio optimization. Employing a multi-agent system with a continuous action space, the study replaced traditional gradient descent methods with a genetic algorithm for parameter optimization. The results revealed that the approach not only balanced financial returns with risk reduction and sustainable investing but also matched state-of-the-art solutions in financial returns while promoting a sustainable portfolio.

Overall, these studies collectively demonstrate the potential of integrating advanced reinforcement learning techniques and additional market-related features to enhance portfolio management, balancing risk and returns while promoting sustainable investment practices.

4.2.9. Network Analysis of Policy-Based Methods

Policy-based studies heavily cite the core RL literature, including "Learning to trade via direct reinforcement learning" [93] and "Performance functions and RL for trading systems and portfolios" [91], highlighting the importance and influence of early recurrent reinforcement learning (RRL) approaches. The reference to "long short-term memory" [123] indicates the reliance on this modeling architecture by policy-based methods. The network analysis also reflects a strong link between traditional RRL and deep policy-based methods,

emphasizing risk-adjusted reward functions. Figure 5 presents the network analysis of studies referenced by policy-based methods.

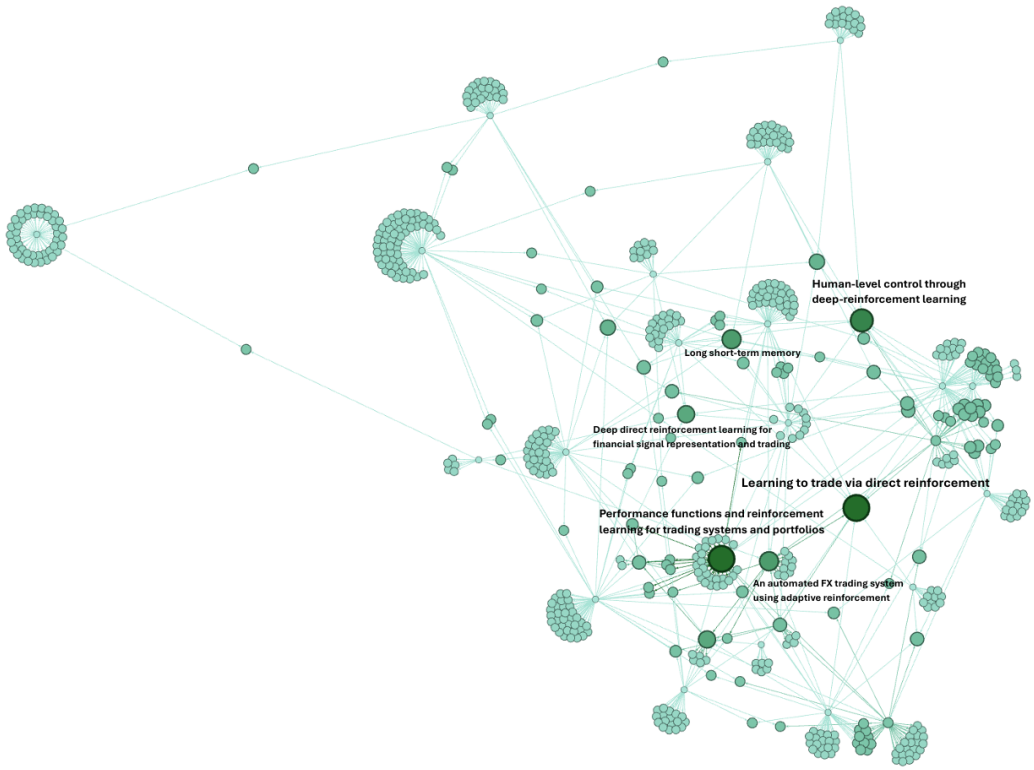


Figure 5. Network analysis of policy-based studies.

4.3. Actor–Critic-Based Methods

Actor–critic models leverage two independent networks: the actor, which selects actions based on the current policy, and the critic, which evaluates these actions by estimating the value function or expected return. During training, the actor proposes actions that the critic assesses, providing feedback in the form of value estimates or advantage values. This feedback helps the actor refine its policy for future action selection. The process involves iterative updates where the actor learns to maximize expected rewards while the critic enhances its value estimation, thus improving overall learning and the agent's performance in the environment. The feedback cycle for actor–critic-based methods is provided in Figure 6.

Various deterministic policy methods have been developed to enhance learning efficiency and stability for actor–critic methods. Deterministic policy gradient (DPG) employs a deterministic policy that focuses solely on the state space, requiring fewer samples for training [124]. Building on this concept, Deep DPG (DDPG) merges DPG with Deep Q-Networks (DQNs), resulting in a model-free, off-policy actor–critic algorithm capable of learning policies in high-dimensional and continuous action spaces [125]. To further improve training stability, proximal policy optimization (PPO) constitutes a simpler approach by clipping the probability ratios, which helps create a pessimistic estimate of the policy's performance, thereby mitigating issues associated with training instability [126]. Table 3 provides a summary of the actor–critic studies designed for quantitative trading.

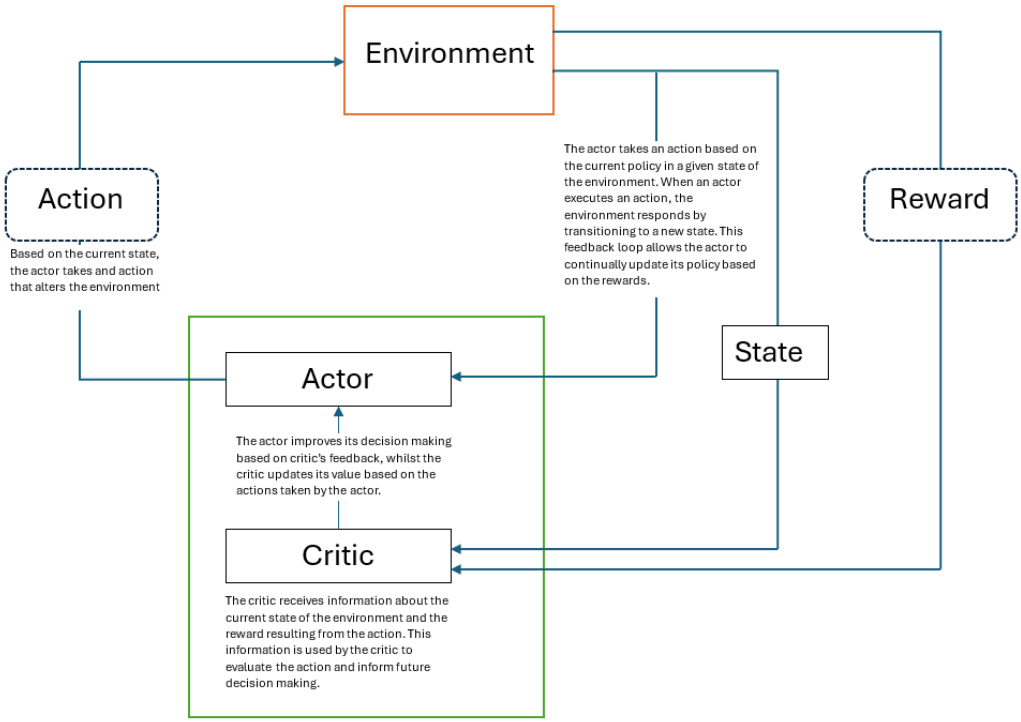


Figure 6. Feedback loop for actor–critic methods.

Table 3. Summary of actor–critic methods.

References	Data	Summary and Key Findings	Results
[127–130]	Stocks	Actor–critic models can be used to efficiently trade stocks and have been demonstrated to return significant profits.	Results indicated strong potential for A–C models in trading.
[131]	Chinese stocks	Explores adversarial training methods that add random noise and a wide range of training parameters. The study compares policy gradient (PG) algorithms to Uniform Constant Rebalanced Portfolio (UCRP).	PG outperformed UCRP in terms of average return and Sharpe ratio at 95% confidence.
[132]	U.S. stocks + indicators	DDPG achieves a high annual return and a high Sharpe ratio compared to the minimum variance strategy and DJIA.	DDPG recorded an annual return of 22.24%.
[133]	Forex	Applies knowledge distillation techniques to a PPO model to address the optimization problem in RL agents and thus improve their trading performance.	Knowledge distillation was found to improve profit and loss margins.
[134,135]	Stocks	Demonstrates that RL, in combination with a gated recurrent unit (GRU) and adaptive trading, can achieve stable returns. The second study proposes a multi-stock trading model that leverages multi-agent DRL for online learning.	Actor–critic models were found to be effective in trading stocks and maximizing profits.
[136]	Stocks	Employs candlestick charts as features in order to minimize the noise of raw data.	The model outperformed other competitive DL models.
[137]	S&P 500 index	Uses offline and online state representation learning before applying actor–critic reinforcement learning to reduce noise and increase the likelihood of learning good policies. The method yields higher profits in stocks with varying trends.	The model exhibited superior generalization ability compared with SOTA methods.

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[138]	U.S. and Chinese stocks	Actor–critic architecture that incorporates NLP models to trade stocks with time-aware text-based data from Twitter and the news.	The model outperformed the baseline models in profitability and risk.
[130,139,140]	U.S. stocks	Hybrid actor–critic models outperform non-hybrid models.	A–C predicted market direction 63.64%, 64.47%, and 58.47% of the time when applied to the S&P 500, NASDAQ, and IBM (1984–2004).
[141–143]	Energy data	A–C architecture is used to increase participation in peer-to-peer energy trading and interaction with microgrids.	A–C algorithms effectively reduced prosumer costs and optimized power sharing.
[144]	Stock	Hybrid models outperform current models.	The model achieved lower prediction error.
[145]	Stock	TD3 algorithm is used to embed investors' risk aversion.	The model achieved superior performance against the S&P 500.
[146]	Futures	DRL based on pairwise cointegration (CA-DRL).	The model outperformed traditional methods in both return and risk management.
[147]	Stock	Uses a fuzzy ensemble deep reinforcement learning (RL) algorithm.	The model outperformed the benchmark models in terms of return rates and volatility.
[148,149]	Stock	Integrates an LLM with reinforcement learning.	The model boosted cumulative investment returns.
[150]	Stock	Integrates an LLM with reinforcement learning.	The method achieved an average improvement of 36% in financial metrics and a notable return of 92.27% on one dataset.
[73]	Cryptocurrency	Hierarchical reinforcement learning (RL) framework.	The method significantly outperformed existing state-of-the-art methods, especially in terms of profitability.
[151]	Stock	Dynamic financial knowledge graph to assist in decision-making and monitoring.	The method demonstrated superiority over SOTA models.
[152]	Energy data	PPO-based actor–critic framework for joint carbon and electricity trading under uncertainty-aware GHIES scheduling.	The method achieved a 30.44% reduction in operational costs and a 20.32% reduction in CO_2 emissions compared with the baseline scenario.
[153]	Energy data	Multi-agent actor–critic reinforcement learning framework integrating decentralized bid/ask price learning and two-tier microgrid trading mechanisms.	The method reduced annual surplus energy sold to the grid by 43.9%, lowered utility grid reliance by 7.1%, and decreased electricity purchase costs by 7.5%.
[57]	Cryptocurrency	Hybrid deep reinforcement learning framework integrating LSTM networks and XGBoost-based feature selection for cryptocurrency trading.	The framework improved trading decision-making and enhanced trading performance in volatile cryptocurrency markets.
[154]	Stock	Fusion LLM–agent framework integrating gradient-based reinforcement learning for adaptive financial trading.	The framework achieved the highest cumulative returns for TSLA and BTC, reaching 50.394% and 45.511%, respectively, with the highest Sharpe ratio of 1.734 for BTC.

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[155]	Cryptocurrency	Reinforcement learning-based portfolio optimization framework with deterministic state transition modeling and patch-wise correlation learning for dynamic asset allocation.	The framework achieved APV values of 238.99, 10.39, and 263.19 on Crypto-A/B/C datasets, significantly outperforming existing DRL portfolio optimization methods.
[156]	Stock	Reinforcement learning framework analyzing hidden-layer configurations for portfolio optimization and investment decision-making.	PPO with one hidden layer achieved the highest cumulative return of 55.27% and an annualized return rate of 52.94%, while DDPG without hidden layers obtained the highest Sharpe ratio of 0.94.
[157]	Stock	Reinforcement learning-based ensemble framework integrating LSTM, XGBoost, and Deep RankNet for stock ranking and portfolio selection.	The framework achieved a 15% increase in cumulative returns compared with traditional ensemble approaches, while LSTM-based portfolios produced average daily returns of 0.47% with a Sortino ratio of 2.617.
[158]	U.S. equities	MMR augments PPO and SAC actor–critic portfolio traders with macroeconomic state variables and momentum-guided action reversal. State includes OHLCV, technical indicators, holdings, cash, and macro data.	PPO-MMR yielded a 10% cumulative return versus 3% for the DJIA and a maximum Sharpe ratio of 0.73; the high-volatility subset yielded a 32% cumulative return and a Sharpe ratio of 0.85. Transaction costs and statistical significance were not reported.
[159]	21 U.S. stocks + S&P 500 ETF	DDPG actor–critic for intraday trading with continuous buy/sell actions. States include prices, technical indicators, holdings, Sharpe ratio, and market correlation; correlation-adjusted Sharpe reward penalizes S&P 500 dependence. Includes 0.01% slippage. Limitations include restricted asset universe, no short selling, and potential look-ahead/data-snooping bias.	Simulation/backtesting: the best agent achieved $114,868 final net worth versus $100,937 for the S&P 500 (+13.8%), a Sharpe ratio of 1.11 versus 0.98 (+13%), and a correlation of 0.15.
[160]	S&P 500	Quantum A3C with VQCs in actor and critic, compared with classical A3C. States combine S&P 500 and macroeconomic features; actions are hold/buy/sell; rewards realize profit on selling with trading costs. LSTM forecasts improve Quantum A3C. Limitations: single asset, short period, single-trajectory evaluation, and no statistical validation.	Historical backtest: Quantum A3C + LSTM achieved a cumulative return of 30.13%, a Sharpe ratio of 4.01, and a maximum drawdown of −3.78%; the corresponding values for classical A3C were 14.9%, 1.74, and −6.79%. The profit factor was 3.05.
[161]	Chinese stocks	Adaptive TwinTRPO actor–critic learns long-only single-stock buy/hold/sell policies under T + 1 constraints. RF–SHAP/XGBoost selects technical-indicator states; a Sortino-based five-day ex post reward guides policy learning. Volatility scales dual trust-region bounds.	Adaptive TwinTRPO achieved a cumulative return of 88.80%, 20.76 percentage points above DQN-Vanilla.

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[162]	Taiwanese stocks	PPO actor–critic for sector rotation, comparing LSTM/Transformer with QNN, QRWKV, and QASA policy/value networks. The agent selects top-10 sectors; the proxy reward rewards next-period capital-share leaders. Quantum models achieve higher training rewards but poorer investment performance, indicating reward–return misalignment and possible overfitting.	Out-of-sample backtest: LSTM achieved a cumulative return of 125.33%, an annualized return of 16.90%, a Sharpe ratio of 0.88, and an MDD of −31.98%. Quantum models underperformed. QNN: 105.93%/0.78; QRWKV: 103.91%/0.77; and QASA: 93.14%/0.71. Transaction costs and statistical significance were not reported.
[163]	BTC, SOL, XRP, BNB, ETH	PPO actor–critic with LSTM features and buy/hold/sell actions. A hierarchical 5 min agent uses signals from independently trained 1 h and 1-day experts. Reward ultimately uses realized PnL. Limitation: evaluation covers only five cryptocurrencies and one fixed temporal split.	In backtesting with 0.1% transaction fees, the 5 min multi-agent achieved a Sharpe ratio of 0.840 and an MDD of 0.200; the corresponding values for the single agent were −0.444 and 0.327. The 1-day agent achieved an annualized return of 52.85%, a Sharpe ratio of 3.408, and an MDD of 0.045, outperforming FinAgent (with corresponding values of 43.08%, 1.18, and 0.1272). Statistical significance was not reported.
[164]	European carbon instruments GRN, CFI2Z, and KEUA	Carbon and energy variables include crude oil, natural gas, coal, and electricity. RSAC-LSTM combines GBM–LSTM volatility modeling with soft actor–critic trading. State includes price, volatility, log return, VaR, CVaR, illiquidity, and position; continuous exposure actions; reward is log return penalized by VaR/CVaR and illiquidity.	RSAC-LSTM yielded ROIs of 5.422% (CFI2Z), 7.528% (KEUA), and 16.893% (GRN), and Sharpe ratios of 0.402, 0.662, and 1.467. GRN produced an MDD of 7.835%.
[165]	Four indices from Yahoo Finance, four futures, and AMZN, INTC, MSFT, and NVD	Multi-agent Dueling DQN with CNN, BiLSTM, GRU, and attention feature extraction executes buy/sell/hold decisions. Inputs include 60-day OHLCV and moving average indicators; rewards, commissions, position constraints, and ATR-based dynamic stop-loss are incorporated.	Cumulative returns were 128.89% (N100: Sharpe, 1.93; MDD, 9.44%) and 472.74% (crude futures: Sharpe, 0.75; MDD, 53.03%).
[166]	CSI 300 (IF) and CSI 500 (IC) index futures from JoinQuant	tIRDPG-LID combines a GRU deterministic actor, twin GRU critics, and delayed actor updates. Feature enrichment, prioritized replay, and dynamically weighted imperfect demonstrations address partial observability and overestimation.	The method achieved the highest total return and Sharpe ratio. In ablation, the LT-IC return was 99.70%, the Sharpe ratio was 1.71, and the MDD was 22.11%. Costs were modeled.
[167]	BTC/USDT 15-min OHLCV from Binance	A Sortino-weighted ensemble combines PPO, DQN, and Dueling DQN for discrete long, short, hold, and close decisions. States include OHLCV, technical indicators, position, cash, and VADER Twitter sentiment. Reward is portfolio P&L or log return.	With a cumulative gain of USD 1257.53 and a profit factor of 1.8, the method outperformed PPO (1.08), DQN (1.04), and Dueling DQN (1.104).

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[168]	AAPL, GOOG, MSFT, AMZN, and INTC	3WD-HRL routes high-certainty states to momentum or order-flow heuristics and boundary states to a TD3-style hierarchical actor–critic. CNN features feed high-level bias/risk controllers and a low-level continuous order actor. Reward combines shaped log return, trading costs, risk penalties, and closure bonuses.	For AAPL, 3WD-HRL returned 1.56% with an MDD of 7.76%; the corresponding results for Rainbow were −7.67% and 24.25%. Five-seed Welch tests reported $p < 0.05$.
[169]	CSI 800 China A-share, DJIA/Hang Seng China 50, and top-10 cryptocurrencies	Logic-Q applies neuro-symbolic trend analysis to tune actor–critic backbone policies. A program sketch and MLP infer market-regime parameters that rescale single-policy action logits or weight time series and LLM subpolicies. Tasks are discrete order execution and multi-asset portfolio rebalancing with transaction-cost-aware rewards.	Logic-Q reported 311.8% cumulative return, 1.280 Sharpe ratio, and −21.8% maximum drawdown, outperforming AlphaMix (CR, 169.7%; SR, 0.816). With 0.13% costs, CR remained 295.2%.
[170]	Energy trading: rooftop-PV data	MAASCPO is a multi-agent safe actor–critic for peer-to-peer electricity trading. Each prosumer selects continuous ESS charge/discharge power; reward is social welfare, while personalized dynamic-operating-envelope deviations define safety costs. TRPO-style policy updates use reward and state-wise cost critics.	Simulation only. No financial profitability metrics were reported. P2P energy trading raised simulated prosumer rewards by USD 8.18 (five agents) and USD 96.89 (ten agents) versus no-P2P trading. MAASCPO achieved zero evaluation violations on IEEE-33.
[171]	Daily gold (XAU/USD) OHLC and New York Times news	Comparative FinRL study of A2C, PPO, and SAC with continuous buy/sell actions. States include cash, holdings, price, six technical indicators, and optionally LLM-filtered FinBERT sentiment/news count; reward is portfolio-value change. LLM voting filters relevant news before sentiment extraction.	PPO with sentiment achieved an average return of 10.05%, a Sharpe ratio of 0.74, and an MDD of −12.42%; the corresponding results for the BH strategy were 9.58%, 0.71, and −12.42%. A 0.1% transaction cost was included.
[172]	Simulation of peer-to-peer (P2P) energy trading	MMAPPO-LSTM models self-interested microgrids in a DEC-POMDP. Decentralized actors bid continuous electricity price, quantity, and storage-control parameters; a shared centralized critic estimates value. Rewards equal grid and P2P operating profit under a multi-round double-auction mechanism.	Across 960 simulated test days, MMAPPO with MRDAC reported total profit of USD −123.81 versus −185.53 for VDA and −221.02 for Greedy.
[173]	Cryptocurrency (BTC-USD, ETH-USD) and TSLA stock	Federated PPO trains local edge-based actor–critic trading agents under FedAvg. States contain log returns, volatility, and momentum features; actions are buy, sell, or hold. Risk-aware rewards penalize drawdown and volatility.	Reported return, 16.8% ± 1.2; Sharpe ratio, 1.32 ± 0.05; and MDD, 11.6% ± 0.8; the corresponding results for FPPO were 13.6%, 1.12, and 16.1%.
[174]	Forex pairs (EUR/GBP, EUR/CAD, AUD/USD, AUD/NZD, GBP/USD, and GBP/JPY)	PPO actor–critic trading agent with long, short, and exit actions. Deep Gaussian filtering learns bandwidths and supplies smoothed price and derivative views to MLP/LSTM models. Reward uses future-price averaging during training, actual next price in backtesting, and position-change commissions.	Ensemble PnL results were 0.977 (EUR/GBP), 1.699 (AUD/NZD), and 0.771 (AUD/USD). Commission was modeled.

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[175]	Chinese stocks	Federated PPO trains encrypted local actor–critic execution policies and aggregates them with FedAvg. State includes LOB, time, and remaining inventory; integer actions specify market-order liquidation quantity. Rewards combine execution quantity and a quadratic market-impact/ risk-preference penalty.	FPPO produced average profit margins above 3% for selected stocks, compared with values below 1.5% produced by AC, DQN, and PPO; a small-cap case reported 24% maximum profit with retracement below 5%.
[176]	Gold, oil, and Bitcoin	Comparative study of DQN, DDPG, and PPO for separate single-asset buy/sell/hold trading environments. State combines price, volume, and STFT features; rewards use next-period price movements and holdings. DDPG/PPO use actor–critic optimization; PPO is reported as the preferred model.	PPO reportedly led DQN/DDPG in total profit, Sharpe ratio, RTSS, and MAI across assets, but the reported absolute STFT model results are not extractable.
[177]	Dow Jones 30 constituents from Yahoo Finance, Alpha Vantage earnings reports, FRED consumer sentiment, and Federal Reserve news sentiment.	A2C_RMS, DDPG_RMS, and PPO_RMS perform multi-stock buy/sell/hold portfolio management. States combine holdings, cash, technical indicators, sentiment, and earnings data; actions trade up to 100 shares. A maximum-drawdown-augmented portfolio-value reward aims to balance return and risk.	A2C_RMS achieved an annual return of 10%, a cumulative return of 27%, a Sharpe ratio of 0.66, and a Sortino ratio of 0.96.
[178]	Gold futures, SPX, USO, SLV, EUR/USD, and geopolitical-risk indices	Hybrid RF–DRL framework compares DQN, PPO, and SAC for discrete gold-futures actions: hold, buy/sell one or two units. State combines market, technical, geopolitical, RF, and portfolio features.	RF+SAC achieved a cumulative profit of 297.14%, a Sharpe ratio of 0.94, an MDD of 1.56%, and an annualized volatility of 1.07%. SAC superiority was reported as significant at $p < 0.05$.
[179]	14 cryptocurrency/ USDT pairs	LSTM-based actor–critic agent using long/short/neutral actions and commission-adjusted PnL reward. Synthetic expert forecasts, sentiment, and asset identity augment price features. Expert guidance improves PnL when sufficiently accurate. Limitation: experts use future prices to construct synthetic signals, creating an oracle-like experimental assumption despite injected errors.	Historical backtesting: price-only test PnL was 14.15 ± 6.18; a 70% accurate expert on three coins increased it to 36.53 ± 6.18, while 65% achieved 23.80 ± 3.93. Commission penalty was included. Sharpe ratio, maximum drawdown, return percentages, and statistical significance tests were not reported.
[180]	Cryptocurrency/ USDT; 1 h; training during August 2017–March 2021; testing during March 2021–February 2022.	PPO trading agents with LSTM features and long/short/neutral positions. TIES-MERGING combines general and asset-specific agents by trimming, resolving sign conflicts, and selectively merging parameters. PnL includes commissions. Limitation: merging hyperparameters experimentally selected, with limited robustness and statistical validation.	Historical backtesting: TIES-MERGING improved PnL over the general baseline on 5/7 known and all 3 unseen cryptocurrencies, e.g., BTC 0.48 vs. 0.43 and LTC 2.21 vs. 2.06. BCH and UNI remained unprofitable. Sharpe ratio, MDD, return percentages, and statistical significance were not reported.

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[181]	Simulated carbon-trading supply chain	DOPPOL extends PPO-Lagrangian for joint inventory ordering and carbon-quota trading under emission constraints. Continuous actions specify goods orders and carbon trades; states include inventory, demand, emissions, quotas, and historical orders. Limitation: entirely simulated environment with stylized assumptions and no real-market validation.	Simulation only. DOPPOL generally improved total return versus PPO: 183.95 vs. 177.06, 320.18 vs. 304.04, 603.51 vs. 569.85, and 425.78 vs. 385.02 across four settings. It also outperformed (s, S) under fluctuating demand. Sharpe ratio, MDD, transaction costs, and statistical significance were not reported.
[182]	NASDAQ 100 stocks	Adaptive framework combining stock-specific PPO/CPPO agents with DeepSeek-enhanced variants. SMA50/SMA200 and NASDAQ trends identify market regimes; Adaptive Portfolio switches from PPO in bull markets to CPPO-DeepSeek in bear markets. Limitation: state/action/reward specifications, temporal split, and statistical significance tests were not reported.	Historical backtesting: PPO achieved the highest IR (0.0080) and tied for the highest weekly performance (0.5134); CPPO achieved the highest Rachev ratio (1.0754). Adaptive Portfolio obtained an IR of 0.0036, a CVaR of −0.0375, and a Rachev ratio of 0.9909. Exact cumulative returns, Sharpe ratios, MDDs, and transaction costs were not reported.
[183]	AMZN LOBSTER MBO data	IPPO with GRU networks jointly trains heterogeneous market-making and execution agents. Discrete LOB actions and task-specific PnL/slippage rewards are used. Learned policies beat standard baselines. Limitation: simplified action/observation spaces; market-making agents fail to achieve net profitability despite reward convergence.	Historical LOB simulation/backtesting: Market makers lost approximately 0.2 ticks on average but outperformed Avellaneda–Stoikov on portfolio value. The execution agent reduced execution cost by up to 0.2 ticks compared with TWAP. Sharpe ratio, MDD, aggregate return, transaction costs, and statistical significance were not reported.
[184]	Five equity datasets (U.S., Germany, and China)	MAT forecasts multi-stock returns; forecasts augment an offline actor–critic portfolio allocator using a Transformer actor, DDPG critic, and TD3 + BC regularization. Actions are long-only allocation weights; reward is portfolio return. Limitation: entirely offline evaluation with no online/live validation.	Historical backtesting with 0.31% transaction fee. Forecast-based strategies outperformed MV/EW in portfolio value; Sharpe ratio peaked at 1.44 on HighTech versus 0.75/0.12 for MV/EW. Sharpe ratio was low or negative on most other datasets. MDD and statistical significance were not reported.
[162]	USD/TWD	QLSTM forecasting features feed a QA3C actor–critic agent with buy/sell/hold actions. State combines forecast probabilities, portfolio variables, and technical indicators; reward encourages trend-following and penalizes losses. Limitation: single FX pair, long-only strategy, classical quantum simulation, and no transaction-cost modeling.	Out-of-sample backtest: QA3C returned 11.87% versus 11.42% for A3C, with 0.92% MDD and 56.71% win rate. It outperformed several currency ETFs but trailed FXB (13.81%) and FXF (12.21%). Sharpe ratio was not reported. Transaction costs were omitted.
[185]	Five U.S. technology stocks (AAPL, NFLX, MSFT, CRM, AMZN) and Finnhub news/ press releases	PrimoRL integrates Llama-3-derived news features with technical indicators in a FinRL/Gymnasium environment. A2C, PPO, and SAC actor–critic agents select continuous buy/sell positions; reward changes from portfolio return to Sharpe ratio after 30 observations.	PrimoRL (PPO) produced a portfolio return of 27.15%, a Sharpe ratio of 1.70, and a maximum drawdown of −11.44%; the corresponding values for FinRL (SAC) were 24.50%, 1.28, and −16.50%.

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[186]	Six equity indices: DJI and S&P 500 (US), CAC 40 (France), HSI (Hong Kong), KOSPI (Korea), and Nikkei 225 (Japan)	IFF-DRL combines AutoConNet future-OHLCV forecasts, online EWC, and A2C/PPO actor–critic agents for single-index long/short trading. States concatenate observed daily and forecast-fused weekly OHLCV.	Backtesting was performed with 0.3% transaction costs and no slippage. IFF-PPO achieved a cumulative return of 1462.42% and an annualized return of 103.19% on HSI, a Sharpe ratio of 4.59, and a maximum drawdown of 4.83%.
[187]	BTC, ETH, BCH, ETC, and USD	A2C manages continuous portfolio weights across four cryptocurrencies and USD. States contain current/prior 10 prices, current allocations, and next-period predicted prices. The actor outputs softmax weights; the critic estimates value. Reward incorporates returns, a loss penalty, market movement, and 0.1% fees.	At 12-h granularity, the final portfolio value was $808,846 and the Sharpe ratio was 1.17; at 5 min intervals, the corresponding results were $300,420 and 0.20. Predictions produced corresponding values of $930,198 and 1.11.
[188]	Electricity trading data from Kaggle	OPRTDPG applies a TD3-style actor–critic architecture to continuous dispatch, storage, TCL-control, and market-bidding actions. State variables include generation, demand, storage, TCL, and price information. Dual critics use clipped Double-Q targets; the actor maximizes critic-estimated return.	Custom-simulator evaluation only. Compared with DDPG, OPRTDPG reduced operating cost from $100,000 to $88,000/day (12%) and price fluctuation from $50 to $45/MWh (10%); energy efficiency rose from 65% to 74.75%.
[189]	30 stocks from Dow (US), Sensex (India), TWSE (Taiwan), and FTSE (UK)	PPO performs multi-stock trading with discrete buy/sell quantities. States include OHLC, holdings, cash, and eight technical indicators per stock. Profit-and-loss reward is augmented with expert feedback derived from SMA, RSI, Bollinger Bands, ROC, and volume signals to reward trend-consistent actions.	The expert-shaped PPO achieved cumulative returns/Sharpe ratios of 45.869%/1.642 (Dow), 69.938%/2.036 (Sensex), 151.665%/2.328 (TWSE), and 17.381%/0.602 (FTSE), exceeding the reported baselines. Maximum drawdowns remained material, including 18.417% for TWSE. The transaction cost was 0.05%.
[190]	Stocks from CSI 300 (88 assets), S&P 100 (59), and NASDAQ 100 (76)	RL-TVDT uses short- and long-horizon temporal–variable Transformers, asset-relation learning, and dual adaptive attention to form portfolio states. SAC actor–critic optimizes portfolio actions using entropy-regularized Q-values.	RL-TVDT achieved portfolio returns/Sharpe ratios of 2.65/1.48 (CSI 300), 1.79/1.01 (S&P 100), and 1.98/1.09 (NASDAQ 100), exceeding the reported benchmarks. Maximum drawdowns were 0.29, 0.34, and 0.32.
[191]	Crude oil, gold, and Euro markets	Multi-agent actor–critic trading with separate fully connected actor and critic networks. The 19-variable state includes holdings, capital, and recent prices; 27 buy/sell/hold actions; ROC-based rewards. Agents are specialized by temporal data windows to reduce loss and improve learning efficiency.	Mean test profits were 2.40% (Euro), 4.61% (gold), and 5.01% (crude oil) using semiannual windows; average losses were 0.04%, 0.37%, and 0.21%, respectively. Reported commissions were below 1% of initial capital.
[192]	U.S. stocks from the S&P 500	A2C and PPO actor–critic agents optimize continuous upper/lower thresholds for market-neutral pair trading. The state includes cash, prices, holdings, daily changes, cumulative fees, and a 30-day signal average; reward is the net portfolio-value change, incorporating realized/unrealized P&L and trading/short-selling fees.	A2C on ACN–NDAQ achieved a mean annual return of 34.47%, a Sharpe ratio of 0.70, and a maximum drawdown of −15.08%; the corresponding results for PPO were 29.41%, 0.60, and −18.48%.

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[193]	Indian equities listed on the NSE	K-means selects mid-cap equities; an LSTM-assisted actor–critic agent assigns non-negative portfolio weights to maximize cumulative wealth. State comprises holdings, cash, and closing prices; reward is portfolio wealth net of initial cash.	The portfolio reportedly increased by approximately 35% over one year, compared with 13% for Nifty50, and outperformed a fixed-deposit comparator.
[194]	Forty USDT-quoted cryptoassets	PPO, A2C, and DQN agents with attention-processed OHLCV, PCA-reduced technical indicators, and Korean Venue Share Indicator (KVSI). Discrete buy/sell/hold actions execute at the next-day open; reward is portfolio log return. KVSI measures Korean exchange volume share and is evaluated through a controlled ablation.	The KVSI model achieved a cumulative return of 18.66, a Sharpe ratio of 0.24, and an MDD of 28.10%; the corresponding values for the model without KVSI were 17.26%, 0.21, and 28.71%; all differences were significant at the 5% level. Fee-adjusted return was 17.64%.
[195]	Industrial-park energy trading	SMA-PDPPO is a centrally trained, decentralized-execution, multi-agent primal–dual PPO method. Each agent selects continuous energy quantity-price bids/offers; a policy actor and reward/penalty critics maximize discounted utility subject to bid/offer safety constraints. States include generation, thermal/electric loads, PV output, and battery state.	SMA-PDPPO produced the highest combined cost savings and additional revenue versus stochastic policy, PPO, DDPG, and MAPPO while increasing locally traded energy and participation.
[196]	Simulated electricity and carbon-permit trading	NE-TRPO multi-agent actor–critic framework for prosumer dispatch and continuous double-auction bidding. Actors choose microturbine output, demand response, and electricity/carbon bid prices; a central critic estimates joint returns. Reward equals negative aggregate operating cost.	NE-TRPO converged in 2600 episodes, 36.39% faster than MADDPG, and reduced aggregate cost by 32.48% versus MADDPG. Joint trading saved \$8/day and reduced emissions by 0.51 tCO2 versus the stated comparator.
[197]	Simulated P2P electricity trading	MASAL is a decentralized multi-agent actor–critic algorithm using trust-region constrained policy optimization. Actors schedule load, microturbine, storage, and bid prices; global reward and individual safety-cost critics guide updates.	Under normal testing, MASAL incurred a daily energy cost of \$221 versus \$218 for unconstrained MAPPO while attaining zero training violations. It achieved the largest P2P volume of 6043 kWh.
[198]	Residential PV, energy storage, and peer-to-peer electricity trading.	Leader–follower framework: SAC, DDPG, or TD3 actor–critic leader schedules energy consumption, storage, and trade quantity; a gradient-ascent follower optimizes bid price. The weighted reward combines consumption utility, expected trading gain, self-sufficiency, and carbon-emission preference.	The proposed SAC, DDPG, and TD3 frameworks achieved higher ten-day mean rewards than standard DRL, L-BFGS-B optimization, and a hybrid DRL baseline.
[199]	CSI 800 constituent stocks	Logic-Q uses Bayesian optimization to parameterize a market-trend program sketch that adjusts pretrained DRL action probabilities without changing network weights. States include volatility, downside risk, and growth; rewards target execution advantage or portfolio balance changes.	For order execution, Logic-Q achieved an additional annualized return of 11.53% versus 9.22% for its strongest baseline. In U.S. cash trading, it achieved an annualized return of 18.1%, a Sharpe ratio of 1.097, and a maximum drawdown of −22.3%.

Table 3. *Cont.*

References	Data	Summary and Key Findings	Results
[200]	Stock data from Compustat via WRDS	An ensemble of A2C, PPO, and DDPG learns stock buy/sell decisions. State combines capital, portfolio holdings, prices, and technical indicators; actions buy or sell stocks.	In backtesting, capital increased from $1,000,000 to $1,457,596.67. Reported annual and cumulative returns were 8.99% and 45.48%, respectively.
[154]	Five stocks (MSFT, JNJ, UVV, HON, and TSLA) and one cryptocurrency (BTC)	FLAG-TRADER fine-tunes SmolLM2-135M using PPO actor–critic learning. Prompt-based states contain price history, portfolio status, and decision history; discrete actions liquidate, hold, or invest all cash. Reward is the daily change in historical Sharpe ratio. Shared LLM layers feed separate policy and value heads.	FLAG-TRADER achieved CR/SR values of 20.106/1.373 (MSFT), 33.724/3.344 (JNJ), 50.394/1.362 (TSLA), and 45.511/1.734 (BTC).
[201]	BTC, ETH, and DOGE	PPO actor–critic cryptocurrency trader integrating multi-LLM news/social-media sentiment. States combine holding status, OHLCV, and decayed sentiment; actions are all-in buy, sell, or hold. Reward uses next-hour close-price change; 0.1% fees apply to buy/sell actions.	Backtesting yielded AR/SR values of 72.34%/2.53 (BTC), 19.91%/2.10 (ETH), and 27.22%/2.69 (DOGE). A half-month live BTC trial returned 3% profit.

4.3.1. Basic Applications of the Actor–Critic Method

The actor–critic method has demonstrated its versatility in trading applications, with studies illustrating its effectiveness in generating actionable signals and optimizing investment strategies. For instance, ref. [127] demonstrated that the deep deterministic policy gradient (DDPG) model can produce buy, sell, or hold signals, enabling traders to maximize asset value. Similarly, ref. [128] showed that actor–critic frameworks contribute to stable investment and income growth.

Building on this, ref. [129] adopted a continuous action space to enhance the trading agent's ability to adjust portfolio positions at each timestep, thereby facilitating better interaction with the trading environment and accelerating convergence in the learning process. A key innovation of the study was the formulation of the trading environment as a partially observed Markov decision process (POMDP), which incorporates stock market constraints such as liquidity and transaction costs. Additionally, the authors enriched the state representation with technical indicators, like the relative strength index and simple moving average, alongside sentiment analysis. The POMDP problem was effectively addressed using the twin delayed deep deterministic policy gradient (TD3) algorithm, which excels in high-dimensional and continuous action spaces typical of stock markets.

Echoing this approach, ref. [202] validated the application of TD3 within a continuous action space for developing a trading strategy based on daily close prices. Their findings indicated that TD3 enhanced performance through improved positioning and share trading, reaffirming the model's efficacy in dynamic market environments.

In contrast, ref. [203] proposed a reinforcement learning framework tailored to agents operating in noisy environments. The model utilizes discrete features to mitigate input complexity and market noise. By incorporating an adaptive scalar to minimize dependence on historical trends and applying a low-pass filter to immediate rewards, the authors demonstrated that their framework successfully trained day trading agents to achieve consistent excess returns.

Expanding the application of actor–critic methods, ref. [204] introduced a novel approach to portfolio management. Their findings indicated that actively managed portfolios can surpass market performance, with a five-stock portfolio achieving an impressive annu-

alized return of 46.58%, a Sharpe ratio of 1.37, and a cumulative return of 115.18%. The study challenges traditional perspectives, suggesting that a smaller number of holdings can yield higher returns with reduced risk, particularly in volatile markets. The model further demonstrates that dynamic strategies outperform static ones, reinforcing the adaptability of actor–critic methodologies in financial contexts.

Approaches like DDPG and TD3 enhance agent performance through continuous action spaces, enabling dynamic adjustments to portfolios while considering market constraints. Additionally, integrating various features, such as technical indicators and sentiment analysis, further improves decision-making. A key takeaway is that employing adaptive and innovative RL techniques can lead to superior financial outcomes, challenging conventional strategies that favor larger portfolios and emphasizing the potential for smaller, actively managed portfolios to yield higher returns with reduced risk.

Ref. [155] proposed a reinforcement learning-based portfolio optimization framework with deterministic state transition modeling to improve the stability and effectiveness of sequential investment decision-making. The framework reduces environmental uncertainty by incorporating deterministic state transition mechanisms into the portfolio management process, enabling reinforcement learning agents to learn more reliable allocation strategies across financial assets. The study demonstrates the potential of reinforcement learning for dynamic portfolio optimization and financial asset allocation under complex market conditions.

Ref. [157] proposed a reinforcement learning-based framework for portfolio selection through stock ranking and matching mechanisms. The study integrates stock ranking strategies with reinforcement learning to dynamically optimize portfolio construction and asset allocation decisions under changing market conditions. By combining ranking-based stock selection with sequential decision-making, the framework enhances portfolio management efficiency and improves investment performance in stock trading environments.

Ref. [159] applied DDPG to continuous intraday equity trading with a correlation-adjusted Sharpe reward, while ref. [187] used A2C to allocate continuous portfolio weights across multiple cryptocurrencies and cash. Multi-agent actor–critic trading was explored in [191] for crude oil, gold, and Euro markets, whereas ref. [192] employed A2C and PPO to optimize trading thresholds for market-neutral statistical pair trading. Ref. [193] further combined equity clustering and an LSTM-assisted actor–critic portfolio allocator for Indian equities. Together, these studies illustrate the continued use of actor–critic methods for continuous trading, portfolio allocation, and relative-value strategies across heterogeneous asset classes.

4.3.2. Combining DL and Deep Reinforcement Learning (DRL)

The integration of DL with deep reinforcement learning (DRL) has been demonstrated to improve trading strategies and portfolio management by optimizing financial decision making.

Motivated by maximizing returns, ref. [132] compared the DDPG algorithm against the DJIA and a traditional minimum variance portfolio allocation strategy, revealing that DDPG achieved the highest Sharpe ratio and an annualized return of 25.9%, outperforming the other benchmarks. Extending this field of work, ref. [131] explored various training parameters, including learning rates, optimizers, feature space, and reward functions for PG, PPO and DDPG. Motivated by [100], the authors reused the EIIE concept but instead opted to build on a deep residual network, which uses identity mapping to resolve the vanishing gradient problem. The authors also proposed an adversarial training method that added random noise to the price data. Using five random Chinese stocks, the results showed that PG outperformed UCRP in asset allocation at 95.0% confidence, while PPO and DDPG exhibited unsatisfactory results. Despite this, the authors acknowledged that the PG model is highly sensitive and may result in unstable performance.

Ref. [133] addressed DRL agent optimization through neural network distillation, particularly with PPO RL agents. By using teacher and student model structures, they demonstrated that distillation significantly enhanced the final performance of the PPO agent. Complementing this, ref. [134] showed that integrating GRU with RL led to more stable returns compared to conventional DRL methods.

Expanding on DRL applications, ref. [205] proposed a DRL-based framework that employs a cascaded LSTM structure. The framework consists of an LSTM that extracts time-series features from daily stock data; the extracted features are used to train an agent, while another LSTM is used to train a strategy function. The framework outperformed previous baseline models in terms of cumulative returns and the Sharpe ratio.

Applying the actor–critic methodology to energy trading and management, ref. [206] designed a network of microgrids where each microgrid bases its energy-trading decisions on local observations. Value decomposition was introduced into the training process with the aim of generating feasible cooperative policies. To address the uncertainty and price fluctuation associated with renewable energy and demand, an RNN with 'burn-in' initialization was combined with the critic network to achieve implicit predictions. The authors demonstrated that their framework is more efficient, practical, and computationally less expensive than conventional RL methods.

Ref. [145] proposed a novel model-free deep reinforcement learning (DRL) framework for constructing optimal portfolio strategies in dynamic and high-dimensional financial markets. The authors leveraged a twin delayed deep deterministic policy gradient (TD3) algorithm to embed investors' risk aversion and transaction cost constraints within a modified Markowitz mean-variance reward function. This DRL-TD3-based approach incorporates advanced exploration strategies and dynamic policy updates to overcome the complexities of high-dimensional state and action spaces typical of financial markets. The paper evaluated the model's performance through extensive empirical testing using data from the Dow Jones Industrial Average and the S&P 100 index, comparing its effectiveness against traditional portfolio optimization methods like minimum variance and maximum Sharpe ratio. The results consistently demonstrated the superior performance of the proposed DRL approach in terms of profitability, risk trade-offs, and resilience in the presence of transaction costs and varying degrees of risk aversion.

Ref. [146] explored the application of deep reinforcement learning (DRL) in pairs trading within the Chinese futures market, specifically focusing on iron ore, coke, and deformed bar futures. Pairs trading is a statistical arbitrage strategy that capitalizes on price discrepancies between related financial assets. The authors integrated the DRL method with cointegration techniques to form an autonomous trading model capable of adjusting trading rules over different cycles. The paper compared several models: a simple threshold (ST) method, a pairwise cointegration-based simple threshold (CA-ST) method, a tripartite cointegration threshold method (CA-ST-ALL), DRL, and DRL based on pairwise cointegration (CA-DRL). The models were tested across three different trading periods (one, two, and three years), with performance evaluated through key financial indicators. The results showed that DRL models, particularly those incorporating cointegration (CA-DRL), outperformed traditional methods in both return and risk management, demonstrating the effectiveness of DRL in China's volatile futures markets.

In a similar study that focused on a larger number of stocks, ref. [147] explored the use of a fuzzy ensemble deep reinforcement learning (RL) algorithm for stock portfolio management. The authors trained a machine learning agent to learn trading strategies by interacting with historical market data without relying on market predictions. By incorporating fuzzy logic, the model captures market trends more flexibly, and an ensemble of RL algorithms, such as advantage actor–critic, trust-region policy optimization, and

deep deterministic policy gradient, enhances decision-making. Using S&P 100 stocks, the proposed method outperformed benchmark models in terms of return rates and volatility, achieving higher returns while maintaining a Sharpe ratio comparable to market indices. The work demonstrated the potential for automated trading systems to achieve superior performance compared to traditional approaches, although future research may address limitations such as scaling issues and intraday data processing.

Ref. [207] discussed the application of reinforcement learning (RL) to optimize trade execution in electronic markets, focusing on minimizing trading costs. Traditional methods for trade execution often rely on static assumptions about market dynamics, limiting their adaptability. Recent advancements in RL have shown potential to develop more adaptive strategies; however, these methods tend to suffer from overfitting, particularly when trained on limited market data. To address this, the authors introduced the Offline Reinforcement Learning with Dynamic Context (ORDC) framework and proposed two algorithms—CASH (Context Aggregate with Hand-crafted Statistics) and CATE (Context Aggregate with End-to-end Training)—to combat overfitting by learning compact representations of market contexts. The authors also developed a high-fidelity simulator based on historical limit-order-book (LOB) data to evaluate their methods, demonstrating improved generalization and performance compared to existing models. The research contributes to both the theoretical understanding and practical implementation of RL in financial markets, emphasizing the importance of context aggregation for better policy generalization.

These studies collectively demonstrate that combining deep learning with deep reinforcement learning (DRL) enhances trading strategies and portfolio management by optimizing asset selection, balancing risk and returns, and improving adaptability to market changes. Approaches using DDPG, PPO, and cascaded LSTM structures consistently achieve higher Sharpe ratios and more stable returns and outperform traditional benchmarks, highlighting the potential of DRL-based frameworks for more effective trading and investment decision-making in dynamic and high-dimensional financial markets.

Ref. [163] combined PPO with LSTM representations and hierarchical multi-timescale agents for cryptocurrency trading, while ref. [164] integrated LSTM-based volatility modeling with soft actor–critic learning for risk-sensitive carbon trading. Ref. [166] employed GRU-based deterministic actors and twin critics with feature enrichment and prioritized demonstrations, and ref. [174] combined PPO with deep Gaussian filtering and MLP/LSTM representations for foreign-exchange trading. Forecast-assisted architectures were further investigated by [186], which fused future OHLCV predictions with A2C/PPO agents, and [190], which combined temporal–variable Transformers and relation learning with SAC for portfolio allocation. These studies indicate that actor–critic performance is increasingly coupled with richer temporal, predictive, and cross-asset representation learning.

4.3.3. Use of State Representation Learning in RL

State representation learning in RL helps extract relevant features from complex financial data; this enables RL agents to make informed and effective trading decisions. Ref. [136] examined the application of candlestick patterns as input data for reinforcement learning (RL) agents. Using K-line theory, raw financial data is processed into candlestick representations before being fed into a deep reinforcement learning (DRL) model. The approach enables the model to capture essential price movements and trends, resulting in improved performance. In this case, the candlestick-based RL model outperformed other DRL models and predictive models like RNNs, demonstrating the effectiveness of using financial patterns in state representation for RL tasks.

Building on the theme of state representation, ref. [137] introduced an innovative approach called SIRL-Trader, which integrates state representation learning (SRL) with an

imitative RL model. The method employs an offline SRL module for dimensionality reduction and feature extraction, helping to condense the state space and improve data quality. Concurrently, an online SRL process co-trains an imitative twin delayed deep deterministic policy gradient (TD3) model, enhancing the accuracy of state information for decision-making. As a result, SIRL-Trader achieved substantially higher profits compared to state-of-the-art trading models. By incorporating SRL in RL, both studies improved the profitability of RL trading agents by improving feature extraction and state information accuracy.

Ref. [158] augmented PPO and SAC portfolio agents with macroeconomic variables and momentum-guided information, while ref. [194] incorporated attention-processed OHLCV, reduced technical indicators, and a Korean Venue Share Indicator into PPO and A2C trading states. Ref. [179] augmented cryptocurrency actor–critic agents with synthetic expert forecasts, sentiment, and asset-identity features, and ref. [189] incorporated technical expert signals into PPO learning for international equity markets. These approaches suggest that improvements in actor–critic trading increasingly depend on informative state construction and auxiliary domain signals rather than changes to the core policy optimizer alone.

4.3.4. NLP for Actor–Critic Models

Ref. [138] combined NLP with an actor–critic RL model to generate trade actions for quantitative day trading based on news and tweets. The NLP model leverages a time-aware LSTM and an intraday attention mechanism to better capture the impact of news or tweets. The RL model employed was DDPG, which outperformed all other contemporary methods, with higher risk-adjusted returns and lower losses.

Ref. [185] integrated Llama-3-derived news representations with technical indicators for A2C, PPO, and SAC equity-trading agents, while ref. [171] combined LLM-filtered financial news with FinBERT sentiment in PPO, A2C, and SAC gold-trading models. Ref. [201] similarly incorporated sentiment extracted from multiple large language models into a PPO cryptocurrency trader, combining textual signals with OHLCV and portfolio information. Across these studies, language-derived information generally improved the informational content available to the policy, although performance remains dependent on the quality, timing, and relevance of the extracted textual signals.

4.3.5. Hybrid Actor–Critic and Deep RL Architectures

Studies on hybrid models using reinforcement learning (RL) reveal that combining RL with other techniques enhances prediction accuracy and profitability in trading. Ref. [139] developed two hybrid actor-only and actor–critic models to forecast short-term stock price movements, comparing them with supervised learning (SL) and classical random walk methods. Their models use multiple performance measures, including DA2, which assesses the agreement between predicted and actual values. The actor–critic models showed robust short-term predictive ability, outperforming actor-only models across various metrics. Specifically, the actor–critic models predicted market direction 63.6%, 64.5%, and 58.5% of the time for the S&P 500, NASDAQ, and IBM (1984–2004), respectively, indicating their consistency and accuracy in diverse market conditions.

Ref. [140] introduced a risk-return algorithm called R3L to craft a portfolio trading strategy, which incorporates a portfolio optimization model, short selling via linear transformation, and the Ape-X algorithm to accelerate training. The use of a linear function for short selling enables profitability even in bear markets, with the critic network learning cumulative return distributions through quantile regression, and the actor network determining optimal portfolio weights. The approach outperformed traditional benchmark strategies, offering higher profitability with lower risk.

In a different approach, ref. [130] combined three agents—DD-DQN, A2C, and PPO—to develop an efficient Bitcoin trading strategy. While DD-DQN individually outperformed other learning agents, the combination of all three proved more effective for automatic Bitcoin trading, as indicated by profit reward functions and Sharpe ratio results.

Ref. [138] integrated natural language processing (NLP) with an actor–critic RL model to generate trading actions based on news and tweets for quantitative day trading. Using a time-aware LSTM with an intraday attention mechanism enhanced the DDPG model's ability to capture market sentiment, leading to higher risk-adjusted returns and fewer losses compared to other methods and demonstrating the effectiveness of incorporating textual data.

Finally, ref. [144] evaluated two hybrid methods for stock market prediction: one integrating deep learning with RL and another with technical analysis indicators. Utilizing advanced models like Bi-LSTM and Bi-GRU on 10 years of Google stock data, both methods proved effective, with the RL strategy slightly outperforming the technical analysis-based approach in profitability. The findings highlight the potential for further optimization across various assets, including stocks and cryptocurrencies.

Ref. [57] developed a hybrid deep reinforcement learning framework for cryptocurrency trading by integrating LSTM networks with XGBoost-based feature selection to improve market representation and trading decision-making. The framework leverages sequential modeling capabilities for financial time-series prediction while employing feature selection to reduce noise and enhance informative trading signals. By combining deep learning architectures with reinforcement learning-based trading strategies, the proposed system improved trading performance and demonstrated the effectiveness of hybrid intelligent trading frameworks for volatile cryptocurrency markets.

It is generally observed that hybrid models, when combined with other ML algorithms or domain-specific features, can significantly enhance trading strategy performance and predictive accuracy.

Ref. [168] proposed a three-way hierarchical framework that routes high-confidence market states to heuristic strategies and uncertain states to a TD3-style actor–critic controller. Ref. [178] combined random forest information with PPO and SAC for gold-futures trading, while ref. [167] constructed a Sortino-weighted ensemble of PPO, DQN, and Dueling DQN using technical and sentiment features for cryptocurrency trading. Ref. [200] similarly combined A2C, PPO, and DDPG within an ensemble stock-trading framework. These studies illustrate a broader tendency to embed actor–critic learning within hybrid decision systems rather than relying on a single standalone reinforcement learning model.

4.3.6. Extensions to Actor–Critic

Extensions to the actor–critic method demonstrate how integrating additional features into RL models can significantly enhance trading strategies. Ref. [208] introduced a novel strategy that incorporates an RSRS indicator for assessing stock support and resistance into a model-based RL method for trading optimization. The indicator allows the model to determine optimal buy and sell times. The approach proved especially effective during highly volatile periods, such as the COVID-19 pandemic, where it showed improved performance in maximum drawdown and annualized returns. Overall, embedding the RSRS tool within the RL algorithm enhanced model stability, adaptivity, and convergence speed, outperforming benchmark algorithms in terms of stability, risk management, and profit gains.

Building on the need for more adaptable RL frameworks, ref. [209] developed SafeFinRL, a high-frequency trading framework designed to reduce bias and variance in training off-policy agents. By separating long-term financial periods into four independent segments, they implemented a single-step soft actor–critic (SAC) on each segment, followed by the application of non-conservative and conservative trace algorithms to estimate the value

function. The approach effectively addressed the challenges of a non-stationary financial environment, reducing bias and variance when estimating value functions in volatile, heterogeneous conditions and achieving positive returns across various tested environments.

Addressing the complexities of identifying and acting on market signals in large volumes of data, Asodekar et al. proposed modifications to DRL agents, including short selling, a new thresholding framework, and the use of turbulence as a safety switch. Their thresholding mechanism determines whether to take a short position or maintain a long position based on a k value, preventing the agent from acting on weak signals. Additionally, the turbulence safety switch ensures that all positions are liquidated when market turbulence reaches a predefined threshold. These modifications led to higher returns, and the resulting approach outperformed benchmark DRL approaches in terms of absolute returns, risk-adjusted returns, and maximum drawdowns.

Collectively, these studies demonstrate that enhancing the actor–critic framework with specialized indicators, off-policy learning adaptations, and safety mechanisms significantly improves trading strategy performance, especially in volatile and unpredictable market conditions.

Ref. [160] incorporated variational quantum circuits into both the actor and critic of an A3C trading model, while refs. [173,175] explored federated PPO to train distributed trading policies without centralizing local financial data. Ref. [180] extended PPO through TIES-MERGING of general and asset-specific cryptocurrency agents, and ref. [184] combined an offline actor–critic portfolio allocator with Transformer-based return forecasting and TD3 + BC regularization. These extensions broaden actor–critic learning beyond conventional centralized online optimization and address issues such as distributed learning, model specialization, and offline policy training.

4.3.7. Actor–Critic Methods for Energy Trading

Actor–critic methods have emerged as effective tools for energy-trading applications, offering solutions for dynamic and decentralized trading environments. Ref. [141] developed a two-step framework that combines a soft actor–critic (SAC) model with an optimization model to facilitate dynamic trading prices within a local microgrid network. The SAC model manages continuous state and action spaces, enhancing the learning process while avoiding local optima through entropy-regularized reinforcement learning (RL). This hierarchical structure includes multiple control levels to manage both internal microgrid trading and interactions with the utility grid. The proposed framework demonstrated a significant reduction in external power trading (up to 21.2%) and power selling (27.4%), showcasing its effectiveness in optimizing local energy trading.

In a similar vein, ref. [142] explored a hierarchical approach to energy sharing and management at the community level using a hierarchical deep reinforcement learning (HDRL) structure. Their two-stage learning process addresses household appliance scheduling and internal energy-trading pricing, with each problem formulated as a Markov decision process (MDP). Initially, a multi-agent deep reinforcement learning (MADRL) model is employed for appliance scheduling policy, followed by a DRL-based model to determine electricity prices for internal trading. The framework significantly reduced prosumer costs, demonstrating the potential of HDRL in efficiently managing energy resources at the community level.

Building on these approaches, ref. [143] focused on improving participation in peer-to-peer markets by developing a three-module framework. The first module collects environmental data, such as agent information, actions, and energy costs. The second module classifies agents based on trading behavior using data from the first module, while the third module predicts trading prices and quantities. A key feature of the study is the

application of Dynamic Time Warping (DTW) to cluster agents according to their trading behaviors, enabling the RL algorithm to support multiple households effectively. This multi-agent RL framework resulted in a 17.0% reduction in energy bills, highlighting its potential for cost savings in peer-to-peer energy trading.

Ref. [153] proposed a multi-agent reinforcement learning framework for decentralized energy trading in networked microgrids, integrating microgrid clustering, dynamic bid/ask price learning, and two-tier trading mechanisms for intra- and inter-cluster energy transactions. The framework combines local optimization with strategic pricing mechanisms to improve market coordination, reduce grid dependency, and enhance economic fairness among distributed microgrids. Experimental results demonstrated substantial reductions in surplus energy sold to the utility grid and lower electricity transaction costs, highlighting the effectiveness of multi-agent reinforcement learning for scalable and market-driven energy-trading systems.

Ref. [188] employed a TD3-style actor–critic model for dispatch, storage, and market bidding, while ref. [170] developed a safe multi-agent actor–critic framework for peer-to-peer electricity trading under personalized operating-envelope constraints. Ref. [172] used MAPPO with recurrent policies for multi-round microgrid electricity auctions, and ref. [195] introduced a primal–dual PPO approach for constrained industrial-park energy trading. Related multi-agent formulations include [196] for joint electricity and carbon-permit trading and [197] for network-constrained peer-to-peer markets. Ref. [198] compared SAC, DDPG, and TD3 within a leader–follower residential energy-trading framework, while ref. [181] extended PPO-Lagrangian optimization to joint inventory and carbon-quota decisions. Collectively, these studies emphasize continuous control, decentralized coordination, and explicit operational constraints, although most evaluations were simulation-based.

4.3.8. Learning Approaches for Actor–Critic

Learning approaches for actor–critic methods are evolving to enhance trading strategies through sophisticated frameworks and signal generation techniques. Ref. [210] presented a novel framework for pair trading utilizing a hierarchical reinforcement learning (HRL) model that jointly learns and optimizes two sub-tasks. In the approach, a high-level policy selects pairs of assets from a pool of possible combinations, while a low-level policy executes the trading actions. The innovative use of a 'feudal framework'—a specific implementation of HRL—allows the action space of the high-level policy to encompass sub-objectives aligned with the sub-tasks, enabling the low-level policy to focus on achieving these sub-objectives. This structure significantly enhances the effectiveness of existing pair trading strategies, demonstrating the potential for improved decision-making in asset management.

In a complementary study, ref. [211] introduced a model designed to generate consecutive trading signals by extracting temporal features from complex observations to maximize profit. The incorporation of a hybrid learning loss allows reinforcement learning sequences to preserve the characteristics of the original state, ensuring that the learning process remains grounded in real market conditions. Additionally, a self-attention mechanism is integrated into the model to refine the decision-making process by learning the temporal importance of hidden representations. The approach not only enhances signal generation but also contributes to a more informed trading strategy. Together, both studies demonstrate the advancements in learning approaches for actor–critic methods, showcasing their capacity to refine trading strategies through hierarchical task management and the extraction of significant temporal features.

Ref. [152] proposed a PPO-based actor–critic deep reinforcement learning framework for optimizing carbon and electricity trading in a green hydrogen-based integrated energy system (GHIES). The study addressed the challenge of uncertainty in energy system optimization by

adopting DRL without requiring explicit uncertainty modeling. Four operational scenarios were investigated, including combinations of electricity trading and carbon trading mechanisms. The experimental results demonstrated that the joint consideration of both electricity and carbon trading achieved the best performance, reducing daily operational costs by 30.44% and CO_2 emissions by 20.32% compared with the baseline scenario.

Ref. [156] developed a reinforcement learning framework to investigate the impact of hidden-layer configurations on stock portfolio optimization performance. The study evaluated different neural network architectures within reinforcement learning models to identify how hidden-layer depth and structure influence portfolio allocation quality and investment returns. The work highlights the importance of network architecture design in improving reinforcement learning-based financial decision-making and portfolio management performance.

Ref. [183] used independent PPO with recurrent networks to jointly train heterogeneous market-making and execution agents in a limit-order-book environment, illustrating the use of specialized policies for distinct market roles. Ref. [177] compared risk-sensitive A2C, DDPG, and PPO portfolio agents using a maximum-drawdown-augmented reward and heterogeneous market, sentiment, and earnings information. Ref. [176] further compared DDPG and PPO with value-based alternatives across gold, oil, and Bitcoin, reporting PPO as the preferred model. These studies highlight continuing interest in multi-agent coordination, risk-sensitive learning, and comparative policy optimization within the actor–critic family.

4.3.9. Reinforcement Learning with LLMs

Ref. [148] discussed FINMEM, a novel autonomous trading agent built using a large language model (LLM) framework tailored for financial decision-making. The framework incorporates three key modules: profiling, memory, and decision-making. The profiling module customizes the agent's risk tolerance, while the memory module, inspired by human cognitive architecture, processes multi-layered financial data to improve decision-making. FINMEM's memory system, divided into short- and long-term layers, mirrors human memory by retaining critical investment data over varying timescales. The approach allows FINMEM to adapt dynamically to financial markets, refining its trading decisions based on both immediate and historical data. The framework outperformed existing models by enabling continuous learning, thus improving trading accuracy even in volatile conditions.

Ref. [150] introduced FinAgent, a multimodal financial trading agent that integrates tool augmentation, diverse data sources, and advanced generalist capabilities for enhanced financial trading performance. In financial markets, traditional AI methods such as reinforcement learning and deep learning face challenges when processing multimodal data (numerical, textual, and visual) and lack generalization across tasks. To address these issues, FinAgent incorporates a market intelligence module to analyze diverse data inputs, a dual-level reflection module to learn from market dynamics, and memory retrieval systems to support decision-making. FinAgent combines reasoning for financial actions with expert insights and trading strategies, ensuring its decisions are both data-driven and well-grounded. In evaluations across six financial datasets, FinAgent significantly outperformed 12 state-of-the-art baselines with a 36% average improvement in financial metrics and a notable 92.27% return on one dataset, highlighting its superiority in multimodal financial trading.

Ref. [149] proposed a novel reinforcement learning (RL) framework called TrendTrader, designed to enhance stock portfolio management by incorporating news sentiment alongside traditional price data. The framework uses a large language model to assess sentiment from financial news, fusing this with historical price information to create heterogeneous contexts, which stabilize the model's ability to handle multimodal data. TrendTrader applies a spatial-temporal backbone network to capture asset correlations and sequential patterns, utilizing supervised learning and incremental RL for training. By integrating trend

heuristics into both its architecture and reward function, TrendTrader aims to better react to stock price movements, thereby improving profitability and robustness. In extensive experiments across U.S. and Chinese markets, TrendTrader significantly outperformed traditional and RL-based portfolio management strategies, validating its effectiveness in capturing financial trends and news impacts.

Ref. [154] proposed a fusion LLM–agent framework combined with gradient-based reinforcement learning for financial trading, enabling large language models to interact with market environments and optimize trading decisions through reinforcement learning. The framework integrates the reasoning and contextual understanding capabilities of LLMs with adaptive policy optimization, allowing the agent to dynamically refine trading strategies in response to financial market conditions. The work highlights the potential of combining language-driven financial reasoning with reinforcement learning for intelligent trading systems.

Ref. [154] fine-tuned a compact language model using PPO, with prompt-based market and portfolio states feeding shared language model layers and separate policy and value heads. Ref. [182] combined PPO-based portfolio agents with DeepSeek-enhanced variants and regime-dependent switching between policies. Ref. [169] further introduced a neuro-symbolic framework that adjusts actor–critic policies using learned market-regime programs and can combine time-series and LLM-based subpolicies. These approaches indicate an emerging direction in which language models become components of policy representation, adaptation, or strategy selection within reinforcement learning systems.

4.3.10. Knowledge Graph-Based Reinforcement Learning

Ref. [151] focused on constructing a dynamic financial knowledge graph to assist in decision-making and monitoring in the financial domain. The authors introduced a framework that leverages structured and semi-structured data related to A-share companies, incorporating machine learning and deep learning models like BERT, BiLSTM, and CRF for financial entity recognition. Reinforcement learning was utilized to manage noise in the data, particularly for relation classification tasks. The dynamic aspect of the graph captures temporal data changes, offering insights into trends over time. This dynamic knowledge graph enables regulatory bodies, financial professionals, and the public to monitor financial data effectively through an interactive visualization system that displays the evolution of financial relationships. The paper contributes to the fields of dynamic knowledge graph construction, financial data analysis, and the application of transfer learning to specific domains, offering a scalable and practical approach for future developments in financial knowledge representation.

Some recent articles from the period 2024–2026 shown in Table 3 extend actor–critic reinforcement learning into cryptocurrency, energy, and large language model-augmented trading. In equity and multi-asset trading, refs. [158–161,165–167,176,177,184] applied PPO, DDPG, A3C, or ensemble actor–critic methods. Several combined these methods with macroeconomic features, quantum circuits, or sentiment inputs, and most reported back-tested returns and Sharpe ratios above simple benchmarks, although few include statistical significance testing. Order-book and market-making approaches such as refs. [175,183] applied PPO-based actor–critic agents to high-frequency execution. Ref. [183] reported that its market-making agents did not reach net profitability despite reward convergence, a useful counterexample to the generally positive results reported elsewhere. For cryptocurrency, foreign exchange, and gold trading, the methods proposed in refs. [162,171,201] combine actor–critic learning with quantum forecasting, sentiment from multiple large language models, or gold-specific technical filters, demonstrating strong headline returns from backtesting only. Large language model-based actor–critic agents are a growing subgroup. The authors of [154] fine-tuned a small language model with PPO; ref. [200] combined A2C, PPO, and DDPG in an ensemble on equities; and [185] integrated Llama-3-derived

news features into PPO and SAC agents. Energy-market applications, including [170], are simulation-only and do not report profitability metrics in contrast to the financial market studies. Symbolic and hierarchical extensions such as [118,119,169,186] combine actor–critic backbones with neuro-symbolic program sketches, meta-learning, or hierarchical policy managers. They report some of the highest cumulative returns in the table, which again highlights how differences in asset universe, cost assumptions, and backtesting window make direct ranking across these studies unreliable. Portfolio-focused studies such as [191–193] further confirm that actor–critic methods remain competitive for pair trading and mid-cap equity selection, although the reported superior performance again depends heavily on the specific asset sample and time period used.

4.3.11. Network Analysis of Actor–Critic Methods

Similarly to value- and policy-based methods, actor–critic studies also cite key literature related to RL. However, additional emphasis is placed on stable policy updates; for example, in references to "Proximal policy optimization algorithms" [126] and "Continuous control with Deep RL" [125]. References to "Deep Direct RL for financial signal representation and trading" [48] appear across policy-based, value-based, and actor–critic methods, reinforcing the importance of the study as a hybrid method and as an alternative to Q-learning. Network analysis of actor–critic methods reveals the shift toward balancing stability (via critics) with expressiveness (via actors), especially in high-dimensional and continuous action spaces. Figure 7 presents the network analysis of actor–critic studies.



Figure 7. Network analysis of actor–critic studies.

5. Discussion

5.1. Discussion of Value-Based Methods

In this section, we outline the strengths and weaknesses of the various RL methods and highlight some notable observations. Critic-only approaches, or value-based methods, excel at handling high-dimensional observation spaces and are efficient in learning from past experiences using replay buffers. These methods focus on estimating the optimal value function for state–action pairs, selecting the best actions by maximizing expected future rewards. Techniques like the Deep Q-Network (DQN) and its variants (e.g., Double DQN) have gained popularity due to their demonstrated success in trading applications. The use of experience replay and target networks has also contributed to more stable training processes.

Despite their strengths, critic-only methods are typically confined to discrete and low-dimensional action spaces [125], limiting their applicability in more complex environments. Additionally, experience replay can be memory-intensive, and value-function approximations often lead to deterministic policies, favoring actions with the highest estimated value, which might restrict adaptability in dynamic markets.

In practical applications, value-based models such as Q-learning have demonstrated consistent profitability across various financial markets, often outperforming simpler benchmarks. For instance, Q-learning achieved an average annual return of 16.3% in Forex trading, while DQN variations applied to stock trading showed strong results, with one model yielding an 8.0% average return in out-of-sample tests.

5.1.1. Discussion of Policy-Based Methods

Actor-only approaches, particularly suited to high-dimensional or continuous action spaces, are adept at adapting to non-stationary market conditions. Policy gradient (PG) methods aim to directly optimize the agent's action distribution, making them more effective in environments with continuous action spaces. However, a key drawback of PG is its susceptibility to high variance due to Monte Carlo (MC) sampling, which can lead to suboptimal gradient estimation, potentially compromising model stability and robustness. These methods also exhibit sensitivity to hyperparameters, necessitating meticulous tuning to achieve optimal performance.

In financial trading, policy-based methods such as recurrent reinforcement learning (RRL) and policy gradient (PG) have demonstrated a strong ability to capture the nonlinearities inherent in market data. RRL, in particular, has been extended with regime-switching properties to better adapt to changing market conditions, often achieving higher Sharpe ratios and outperforming value-based methods. Despite their computational intensity and sensitivity to parameter settings, policy-based approaches have yielded impressive results. For instance, RRL using policy gradients achieved annual returns exceeding 25.0%, especially when combined with strategies like stop-loss and the Calmar ratio. Moreover, RRL models that integrated technical indicators showed enhanced Sharpe ratios and profitability, underscoring the potential of policy-based methods in trading applications.

5.1.2. Discussion of Actor–Critic Methods

Actor–critic methods effectively combine the strengths of both actor-only and critic-only approaches. While actor-only methods excel in continuous and stochastic environments, critic-only methods offer superior sample efficiency. However, despite this promising combination, actor–critic methods remain relatively underexplored in trading applications. This is partly due to the complexity of training two networks simultaneously, a challenge further intensified by the dynamic and continuous nature of financial markets. Additionally, the integration of deep learning as a function approximator introduces instability issues that are yet to be fully addressed.

Notably, actor–critic techniques like deep deterministic policy gradient (DDPG) and proximal policy optimization (PPO) have gained traction, with some studies incorporating LSTM networks to capture temporal dependencies. These models have shown remarkable adaptability and performance in complex, high-dimensional trading scenarios, often outperforming both value-based and policy-based methods, particularly in environments requiring continuous action spaces. For instance, DDPG achieved substantial annualized returns, with one study reporting a 22.24% return on U.S. stocks. Additionally, other research indicated that actor–critic methods consistently outperformed their value-based and policy-based counterparts, achieving market direction prediction accuracy rates exceeding 60.0%.

5.1.3. Comparison of Actor-Only and Critic-Only Approaches

Despite the dominance of critic-only research, actor-only methods have achieved impressive results, with some models reporting annual returns exceeding 25.0%. Notable examples include DPG with EIIE [100], RRL with stop-loss and the Calmar ratio [107], PSO-based RRL with the Calmar ratio [108], and Adaptive RRL with risk management [109]. Similarly, critic-only models like DRQN with smaller replay and longer training sequences [27], Q-learning based on boundary boxes [30], and MetaPolicy [31] have incorporated customization and risk management strategies to enhance model performance. These studies highlight that while RL models can adapt to market dynamics, they often operate under assumptions that may not always align with real-world scenarios, such as continuous trading periods and the exclusion of seasonality. In general, actor-only approaches demonstrate impressive adaptability and performance, achieving high annual returns, particularly when incorporating advanced techniques like stop-loss and risk management. Similarly, critic-only methods have shown effectiveness by integrating customization strategies, highlighting the adaptability of RL models in financial trading.

5.1.4. Influence of State Representation

The choice of state representations plays a critical role in the performance of RL models. The effectiveness of specific state features can vary significantly across different datasets. For example, ref. [96] demonstrated that RSRRL outperformed RRL in one dataset, while ref. [97] found that RRL performed better on another dataset, attributing this to the lack of correlation between the data. Various studies have adopted diverse approaches to state representation. Ref. [22] used private and market variables, whereas ref. [23] tailored state representations to each agent. Moreover, ref. [98] observed that using the P/E ratio as a transitional variable improved the Sharpe ratio but cautioned against assuming similar results across different datasets. The diversity in approaches highlights the importance of aligning state representations with the specific characteristics of the trading environment. This helps optimize RL model performance, as different datasets may exhibit unique patterns, correlations, and dynamics that influence the effectiveness of various features. Tailoring state representations to dataset characteristics enhances the model's ability to learn and adapt; it also improves its robustness and predictive accuracy in real-world trading scenarios.

5.1.5. Feature Selection and Reward Function Design

The choice of state representations and feature selection is critical to the performance of reinforcement learning (RL) models. The effectiveness of specific states can vary significantly across datasets; optimal performance on one dataset does not guarantee similar results on another. For instance, ref. [96] demonstrated that RSRRL outperformed RRL on one dataset, while ref. [97] found the opposite on a different dataset, attributing this variance to the lack of correlation between the data. Various studies have employed diverse approaches to state representation. Ref. [22] utilized private and market variables, while ref. [23] tailored state representations to each agent. Additionally, ref. [98] noted that

incorporating the P/E ratio as a transitional variable improved the Sharpe ratio but warned against assuming similar results across different datasets.

Feature selection also significantly influences model efficacy. For example, ref. [41] used genetic algorithms (GAs) to automate feature selection, whereas ref. [48] applied a deep neural network for this task. This diversity underscores the necessity of selecting features that align with the specific characteristics of the data. Moreover, the design of reward functions is essential, as an inappropriate function can lead to suboptimal agent behavior, as highlighted by [20]. Conflicting findings regarding reward function efficiency further emphasize the need for customization based on context; refs. [107,108] assert the superiority of the Calmar ratio over the Sharpe ratio, contrary to the perspectives of [91–93].

The application of RL algorithms to different datasets can yield contrasting outcomes. For instance, while DDPG demonstrated disappointing results [131], more favorable outcomes were reported by [127,132]. These discrepancies highlight the importance of the underlying data in RL model performance, reinforcing the need for careful selection of state and reward functions tailored to specific dataset characteristics. This aligns with [11], who noted that varying currency pairs, data types, and forecasting periods can lead to different conclusions, emphasizing that a successful RL model is highly dependent on capturing the nuanced aspects of the dataset [212]. By effectively integrating feature selection with well-designed reward functions, RL models can better adapt to the complexities of trading environments, ultimately enhancing performance and robustness.

5.1.6. Tailored vs. One-Size-Fits-All Approaches: Navigating Model Specificity

The notion of an "out-of-the-box" model that performs consistently across any dataset is impractical. Ref. [20] emphasized that, despite promising results, Q-learning cannot be regarded as a universal solution. Similarly, both the RSRRL model [97] and DDPG [131,132] demonstrated effectiveness only under specific conditions, underscoring the data-dependent nature of success in reinforcement learning (RL) trading.

The No Free Lunch theory, as articulated by [213], posits that an algorithm that excels on one dataset may not necessarily perform as well on another. This discrepancy arises from how well the data aligns with the algorithm, rather than indicating the inherent superiority of one algorithm over another. To gain an advantage, it becomes imperative to employ a specialized structure tailored to the dataset's characteristics. As a result, using deep neural networks for feature selection, selecting appropriate reward functions, and optimizing hyperparameters become essential for enhancing the model's performance. Overall, the reviewed studies emphasize that the success of RL models in trading is contingent on careful customization of state representations, feature selection, and reward functions tailored to the dataset's properties. Given the varying results across different datasets and the need for model-specific adjustments, there is no one-size-fits-all RL model.

5.2. Generalizability and Comparison of the Models

To address the generalization challenge in machine learning, particularly in the context of financial trading, it is crucial to underscore the importance of theoretical analysis alongside practical market execution [214,215]. Financial markets are highly dynamic, characterized by ever-changing variables and conditions, such as fluctuating liquidity, market sentiment, and macroeconomic events [216]. Models that perform well during backtesting or in simulated environments often struggle when exposed to the real-world volatility of live markets [217]. This discrepancy arises due to the gap between controlled academic research and actual trading decisions, which highlights the importance of theoretical insights to ensure models are robust enough to generalize well in live trading environments [218].

Reinforcement learning (RL), in particular, brings unique advantages to financial trading due to its focus on sequential decision-making and long-term reward optimization [219]. However, without careful consideration of market realities, RL models are prone to overfitting to specific datasets compared to some other machine learning models [220], failing to adapt when conditions shift. A solid theoretical foundation enables the development of models that not only perform well on historical data but also generalize to unseen market conditions [221]. This foundation includes incorporating economic principles, behavioral finance insights, and broader statistical theories, all of which provide the necessary scaffolding to make these models more adaptable to real-time financial decision-making [222].

Moreover, successful market execution depends on understanding the market microstructure and real-world trading constraints, such as transaction costs, liquidity slippage, and execution time delays [223]. Academic research can fall short if these factors are not adequately considered, making it difficult to translate promising model results into profitable trading strategies. Hence, linking academic insights with the practicalities of market execution is critical for bridging the gap between theoretical success and trading performance [224].

Incorporating research that specifically focuses on the generalization issue within financial machine learning models is therefore essential. Newer studies have begun to address this challenge by employing techniques such as domain adaptation, transfer learning, and adversarial training, which seek to improve a model's robustness and adaptability [225]. These approaches, combined with market-informed execution strategies, will be pivotal in ensuring that RL models in financial trading are not only theoretically sound but also practical for real-world applications [226].

The broader literature on financial asset forecasting now comprises hundreds of new studies each year using statistical, machine learning, deep learning, hybrid, and optimization-based approaches. Many existing reviews have already surveyed these diverse AI-driven prediction and optimization methods. In contrast, our aim in this paper is deliberately narrow: we focus on reinforcement learning (RL) strategies for quantitative trading, summarizing how different RL algorithms interact with market environments, define reward functions, and manage risk. As a result, this review does not provide an exhaustive comparison between RL and all other AI-driven trading strategies or metaheuristic optimization techniques, such as genetic algorithms, particle swarm optimization, bacterial foraging optimization, or whale optimization. These methods form an important complementary strand of research in financial forecasting and trading, but a full treatment of them lies outside the scope of this RL-centered review. We therefore acknowledge that the comparisons presented here are selective and that RL should be viewed as one part of a much larger landscape of AI-based and statistical trading approaches.

A more fundamental limitation is that we report performance metrics, for example, annual returns and Sharpe ratios, exactly as presented in the original RL trading papers, without re-deriving them under a single, unified backtesting framework. This makes it difficult to judge which of the reviewed results reflect a genuine trading edge and which reflect weaknesses in how the underlying study was designed and reported. Reading the reviewed papers against the seven standards proposed in QFRS [227] helps explain why. First, on data leakage (QFRS standard 3, anti-leakage feature engineering, and standard 5, chronological train–validation–test splitting), most RL trading papers do not state whether features, scaling parameters, or state representations were fitted using only information available at or before each decision point, and several use random or non-chronological splits that can let future price information leak into training. Second, on transaction costs (part of QFRS standard 7, economic backtesting), the studies we review differ widely in whether they model commissions, bid-to-ask spreads, slippage, and borrowing costs at all, and when they do, the assumed cost levels are rarely justified or compared with real

market conditions. Third, on backtesting design (also QFRS standard 7), few papers report the trading rule, position sizing, execution timing, and holding period distribution in enough detail for another researcher to reproduce the reported trades, and fewer still test performance across more than one market regime or time period. Fourth, on risk-adjusted economic profit (QFRS standards 6 and 7 together), many studies emphasize raw returns or prediction error metrics such as RMSE, while reporting the Sharpe ratio, Sortino ratio, maximum drawdown, and win rate inconsistently or not at all, which makes it hard to judge whether a reported gain reflects real skill or simply higher risk-taking. Because these four issues are rarely addressed together in a single paper, the reported results are not directly comparable across studies without all of them adopting a shared, reviewer-enforceable framework such as QFRS [227].

Our review therefore focuses on summarizing the returns and risk metrics as reported and on comparing RL approaches at a high level, rather than claiming precise, like-for-like comparability of performance across studies. In addition, deep RL trading results are typically sensitive to hyperparameters, environment design, reward scaling, and training budget, and the reviewed studies differ in how rigorously they report and validate their findings. Some papers provide confidence intervals or statistical tests and make code or data available, while others report single-run performance without formal uncertainty quantification or open resources. For this reason, the metrics we summarize should be viewed as indicative rather than definitive, and readers are encouraged to consult the original papers and, where possible, the QFRS checklist to assess the robustness of individual results in terms of data handling, cost assumptions, backtesting design, and risk-adjusted economic value.

5.2.1. Network Analysis of the Complete Set of RL Methods for Quantitative Trading

The general network analysis highlights a tightly interconnected core of foundational studies that underpin reinforcement learning in quantitative trading. Central nodes such as "Human-Level Control through Deep Reinforcement Learning" [21] and "Learning to trade via direct reinforcement learning" [93] serve as key methodological anchors, linking deep Q-learning advancements with financial applications. Papers like "Deep Direct Reinforcement learning for financial signal representation and trading" [48] appear across all sub-networks, reflecting their hybrid utility and broad influence. The analysis reveals a strong overlap between value-based, policy-based, and actor–critic methods, indicating that the field has evolved through a shared foundation of deep reinforcement learning principles, enhanced by domain-specific adaptations such as risk-aware reward functions and temporal modeling techniques. Figure 8 summarizes the network analysis of RL in quantitative trading.

5.2.2. Regulatory, Privacy, and Cybersecurity Considerations

A further limitation of many RL-based trading studies is that they evaluate performance mainly in terms of profitability and risk-adjusted return while giving limited attention to the regulatory and operational constraints that apply in real financial markets. In practice, algorithmic trading systems must satisfy requirements for governance, testing, monitoring, kill-switch controls, record-keeping, and market abuse surveillance under frameworks such as MiFID II in Europe while also meeting operational resilience and security expectations in other jurisdictions such as the United States. Privacy is also relevant when models are trained on proprietary order flow, client-level data, or other sensitive financial information, since data access, storage, and reuse may be restricted by institutional policy or law. In addition, AI-driven trading systems are exposed to cybersecurity risks such as data poisoning, adversarial manipulation of inputs, unauthorized access, and failures of automated execution pipelines. These issues do not negate the value of RL for quantitative trading, but they show

that real-world deployment depends not only on predictive or trading performance but also on compliance, privacy protection, and system security.

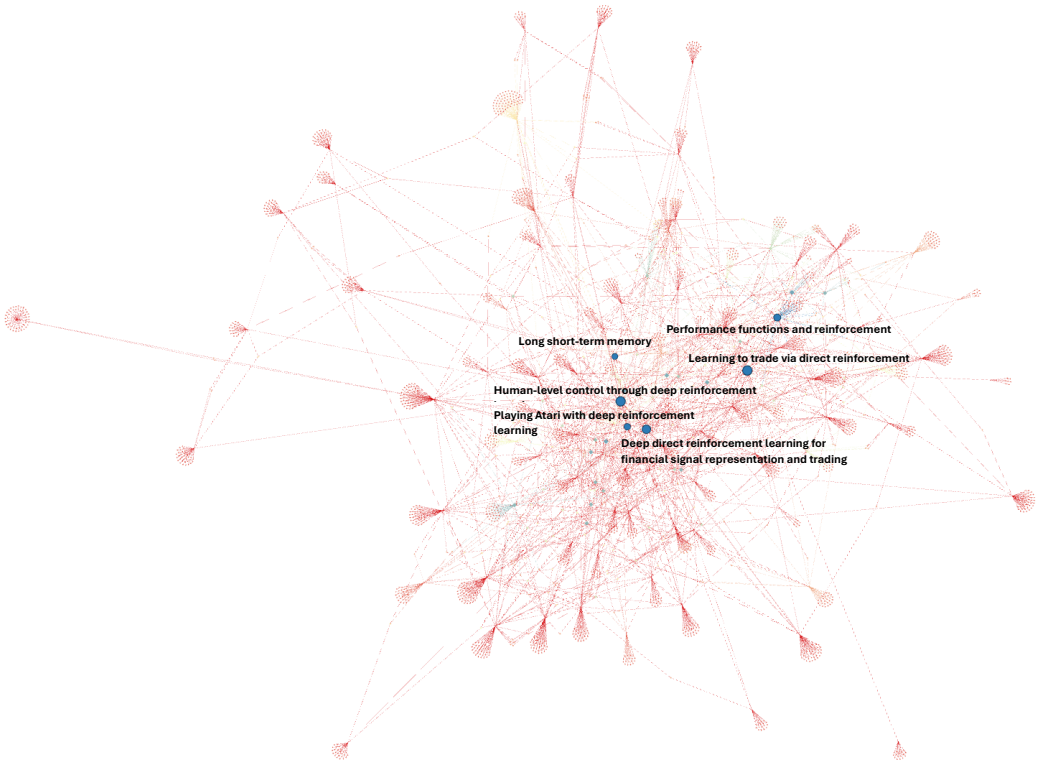


Figure 8. General network analysis of RL in quantitative trading.

6. Conclusions

This review examined reinforcement learning applications in quantitative trading from a computer science methodological perspective. The reviewed literature shows that value-based, policy-based, and actor–critic methods can learn trading and portfolio decisions from market data. Their performance can be improved through deep learning architectures, richer state representations, carefully designed reward functions, and explicit risk-management mechanisms. These approaches are especially useful when market conditions are complex, data are high-dimensional, and trading decisions must be made sequentially.

However, the main finding of this review is that the reported performance of RL trading methods cannot be compared directly across studies. Although many papers report promising returns, Sharpe ratios, or drawdown reductions, the studies use different datasets, asset universes, sample periods, state variables, reward functions, trading rules, transaction cost assumptions, training procedures, and validation protocols. Some studies evaluate agents only in simulated environments, while others use historical backtests with different assumptions about execution, liquidity, slippage, leverage, and market impact. Consequently, a strong result in one study does not demonstrate that a particular RL method is superior to another method evaluated under a different framework.

Prediction metrics such as MAE, MSE, and RMSE may be useful for measuring forecast error, but they do not by themselves establish economic value. A model can achieve a low prediction error while producing an unprofitable trading strategy after transaction costs, bid–ask spreads, slippage, financing costs, and drawdowns are considered. Likewise, a high reported Sharpe ratio is not sufficient evidence of a deployable strategy if the study does not clearly report the data construction process, timing of information, model selection procedure, execution rules, and robustness across market regimes. Reliable

evaluation requires a full trading pipeline that connects the learning objective to realistic and reproducible portfolio outcomes.

For this reason, future RL trading research should move beyond reporting isolated returns or prediction metrics and adopt a common reporting and evaluation framework. The Quantitative Finance Reporting Standard (QFRS) provides such a framework through seven connected standards: dataset specification and data handling; label or reward definition; point-in-time feature engineering; leakage-free scaling and normalization; time-respecting training, validation, and test splits; appropriate evaluation metrics; and cost- and slippage-aware backtesting with explicit execution and decision rules. Applying these standards would reduce the risk of data leakage, backtest overfitting, metric optimization, and unsupported trading claims. It would also enable returns, risk measures, and practical trading results to be assessed on a more consistent basis across studies [227].

This review is intentionally based on the computer science literature indexed by dblp and therefore does not claim to be an exhaustive survey of all RL, econometric, financial, or optimization-based trading studies published in finance and economics venues. Rather, it provides a computer science-focused account of how RL methods are designed and evaluated for quantitative trading. RL should be viewed as one component of a broader ecosystem that includes statistical models, econometric methods, supervised and deep learning, hybrid optimization, and metaheuristic approaches. Nevertheless, the need for transparent, reproducible, and economically meaningful evaluation applies to all of these approaches. Adopting QFRS would help the field distinguish genuinely robust progress from results that are specific to a particular dataset, backtest design, or set of assumptions.

Author Contributions: Conceptualization, M.K., B.S., S.D. and X.L.; methodology, M.K., B.S., X.L. and S.U.; validation, M.K., B.S., S.U., R.K.S. and X.L.; formal analysis, M.K., B.S., S.U., X.L., S.D. and R.K.S.; investigation, M.K., B.S., S.U., X.L., S.D. and R.K.S.; writing—original draft preparation, M.K., X.L. and S.D., writing—review and editing, M.K., B.S., S.U., X.L., S.D. and R.K.S.; visualization, S.D. and X.L.; supervision, M.K., B.S., S.U. and R.K.S.; project administration, M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (GPT-5.2, OpenAI, San Francisco, CA, USA) for language editing, including grammar and sentence flow and for the LaTeX environment. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Alameer, A.; Saleh, H.; Alshehri, K. Reinforcement Learning in Quantitative Trading: A Survey. *TechRxiv* **2022**. [CrossRef]
2. Sun, S.; Wang, R.; An, B. Reinforcement Learning for Quantitative Trading. *arXiv* **2021**, arXiv:2109.13851.
3. Fama, E.F. Efficient capital markets. *J. Finance* **1970**, *25*, 383–417. [CrossRef]
4. Hu, Z.; Zhao, Y.; Khushi, M. A Survey of Forex and Stock Price Prediction Using Deep Learning. *Appl. Syst. Innov.* **2021**, *4*, 9. [CrossRef]
5. Masry, M. The Impact of Technical Analysis on Stock Returns in an Emerging Capital Markets (ECM's) Country: Theoretical and Empirical Study. *Int. J. Econ. Finance* **2017**, *9*, 91–107. [CrossRef]
6. Lo, A.W.; Mamaysky, H.; Wang, J. Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *J. Finance* **2000**, *55*, 1705–1765. [CrossRef]

7. Bhattacharya, S.; Kumar, K. A Computational Exploration of the Efficacy of Fibonacci Sequences in Technical Analysis and Trading. *Ann. Econ. Finance* **2006**, *7*, 185–196.
8. Frost, A.J.; Prechter, R.R. *Elliott Wave Principle: Key to Market Behavior*; Elliott Wave International: Gainesville, GA, USA , 2005.
9. D'Angelo, E.; Grimaldi, G. The Effectiveness of the Elliott Waves Theory to Forecast Financial Markets: Evidence from the Currency Market. *Int. Bus. Res.* **2017**, *10*, 1–18. [CrossRef]
10. Chatterjee, A.; Ayadi, O.F.; Maniam, B. The applications of the fibonacci sequence and elliott wave theory in predicting the security price movements: A survey. *J. Commer. Bank. Finance* **2002**, *1*, 65–76.
11. Yu, L.; Wang, S.; Lai, K.K. *Foreign-Exchange-Rate Forecasting with Artificial Neural Networks*; Springer: New York, NY, USA, 2010.
12. Ni, L.; Li, Y.; Wang, X.; Zhang, J.; Yu, J.; Qi, C. Forecasting of Forex Time Series Data Based on Deep Learning. *Procedia Comput. Sci.* **2019**, *147*, 647–652. [CrossRef]
13. Lahmiri, S.; Bekiros, S. Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos Solitons* **2019**, *118*, 35–40. [CrossRef]
14. Pang, X.; Zhou, Y.; Wang, P.; Lin, W.; Chang, V. An innovative neural network approach for stock market prediction. *J. Supercomput.* **2020**, *76*, 2098–2118. [CrossRef]
15. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*; Teh, Y.W., Titterington, M., Eds.; ML Research Press: Maastricht, The Netherlands, 2010; Volume 9, pp. 249–256.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv* **2015**, arXiv:1502.01852.
17. Sim, H.S.; Kim, H.I.; Ahn, J.J. Is Deep Learning for Image Recognition Applicable to Stock Market Prediction? *Complexity* **2019**, *2019*, 4324878. [CrossRef]
18. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018.
19. Lehman, J.; Clune, J.; Misevic, D.; Adami, C.; Altenberg, L.; Beaulieu, J.; Bentley, P.J.; Bernard, S.; Beslon, G.; Bryson, D.M.; et al. The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities. *arXiv* **2018**, arXiv:1803.03453.
20. Carapuço, J.; Neves, R.; Horta, N. Reinforcement learning applied to Forex trading. *Appl. Soft Comput.* **2018**, *73*, 783–794. [CrossRef]
21. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.A.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
22. Nevmyvaka, Y.; Feng, Y.; Kearns, M. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd International Conference on Machine Learning*; Association for Computing Machinery: New York, NY, USA, 2006; pp. 673–680.
23. Lee, J.W.; Park, J.; Jangmin, O.; Lee, J.; Hong, E. A multiagent approach to q-learning for daily stock trading. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2007**, *37*, 864–877. [CrossRef]
24. Bertoluzzo, F.; Corazza, M. Testing Different Reinforcement Learning Configurations for Financial Trading: Introduction and Applications. *Procedia Econ. Finance* **2012**, *3*, 68–77. [CrossRef]
25. Bertoluzzo, F.; Corazza, M. Reinforcement Learning for Automated Financial Trading: Basics and Applications. In *Recent Advances of Neural Network Models and Applications*; Bassis, S., Esposito, A., Morabito, F.C., Eds.; Springer: Cham, Switzerland, 2014; pp. 197–213.
26. Pendharkar, P.C.; Cusatis, P. Trading financial indices with reinforcement learning agents. *Expert Syst. Appl.* **2018**, *103*, 1–13. [CrossRef]
27. Huang, C.Y. Financial Trading as a Game: A Deep Reinforcement Learning Approach. *arXiv* **2018**, arXiv:1807.02787.
28. Scott, B. *How Can Cryptocurrency and Blockchain Technology Play a Role in Building Social and Solidarity Finance?* Technical Report; UNRISD Working Paper; UNRISD: Geneva, Switzerland, 2016.
29. Lucarelli, G.; Borrotti, M. A Deep Reinforcement Learning Approach for Automated Cryptocurrency Trading. In *Artificial Intelligence Applications and Innovations*; MacIntyre, J., Maglogiannis, I., Iliadis, L., Pimenidis, E., Eds., Springer: Cham, Switzerland, 2019; pp. 247–258.
30. Zhu, Y.; Yang, H.; Jiang, J.; Huang, Q. An Adaptive Box-Normalization Stock Index Trading Strategy Based on Reinforcement Learning. In *Neural Information Processing*; Cheng, L., Leung, A.C.S., Ozawa, S., Eds.; Springer: Cham, Switzerland, 2018; pp. 335–346.
31. O, J.; Lee, J.; Lee, J.W.; Zhang, B.T. Adaptive stock trading with dynamic asset allocation using reinforcement learning. *Inf. Sci.* **2006**, *176*, 2121–2147. [CrossRef]
32. Lu, J.Y.; Lai, H.C.; Shih, W.Y.; Chen, Y.F.; Huang, S.H.; Chang, H.; Wang, J.Z.; Huang, J.L.; Dai, T.S. Structural break-aware pairs trading strategy using deep reinforcement learning. *J. Supercomput.* **2022**, *78*, 3843–3882. [CrossRef] [PubMed]
33. Brim, A. Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*; IEEE: Piscataway, NJ, USA, 2020; pp. 0222–0227. [CrossRef]

34. Jegadeesh, N. Long-term performance of seasoned equity offerings: Benchmark errors and biases in expectations. *Financ. Manag.* **2000**, *29*, 5–30. [CrossRef]
35. Matsui, T.; Goto, T.; Izumi, K. Acquiring a Government Bond Trading Strategy Using Reinforcement Learning. *J. Adv. Comput. Intell. Intell. Inform.* **2009**, *13*, 691–696. [CrossRef]
36. Zarkias, K.S.; Passalis, N.; Tsantekidis, A.; Tefas, A. Deep Reinforcement Learning for Financial Trading Using Price Trailing. In *ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2019; pp. 3067–3071. [CrossRef]
37. Fallahpour, S.; Hakimian, H.; Taheri, K.; Ramezanifar, E. Pairs trading strategy optimization using the reinforcement learning method: A cointegration approach. *Soft Comput.* **2016**, *20*, 5051–5066. [CrossRef]
38. Eilers, D.; Dunis, C.L.; von Mettenheim, H.J.; Breitner, M.H. Intelligent trading of seasonal effects: A decision support algorithm based on reinforcement learning. *Decis. Support Syst.* **2014**, *64*, 100–108. [CrossRef]
39. Théate, T.; Ernst, D. An application of deep reinforcement learning to algorithmic trading. *Expert Syst. Appl.* **2021**, *173*, 114632. [CrossRef]
40. Bisi, L.; Liotet, P.; Sabbioni, L.; Reho, G.; Montali, N.; Restelli, M.; Corno, C. Foreign exchange trading: A risk-averse batch reinforcement learning approach. In *Proceedings of the First ACM International Conference on AI in Finance*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–8.
41. Dempster, M.A.H.; Romahi, Y.S. Intraday FX Trading: An Evolutionary Reinforcement Learning Approach. In *Intelligent Data Engineering and Automated Learning—IDEAL 2002*; Yin, H., Allinson, N., Freeman, R., Keane, J., Hubbard, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 347–358.
42. Corrêa, M.F.; Vellasco, M.; Figueiredo, K.; Vellasco, P. Trading Strategy in Foreign Exchange Market Using Reinforcement Learning Hierarchical Neuro-Fuzzy Systems. In *Advances in Neuro-Information Processing*; Köppen, M., Kasabov, N., Coghill, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 461–468.
43. Tan, Z.; Quek, C.; Cheng, P.Y. Stock trading with cycles: A financial application of ANFIS and reinforcement learning. *Expert Syst. Appl.* **2011**, *38*, 4741–4755. [CrossRef]
44. Ma, C.; Zhang, J.; Liu, J.; Ji, L.; Gao, F. A parallel multi-module deep reinforcement learning algorithm for stock trading. *Neurocomputing* **2021**, *449*, 290–302. [CrossRef]
45. Yang, B.; Liang, T.; Xiong, J.; Zhong, C. Deep reinforcement learning based on transformer and U-Net framework for stock trading. *Knowl.-Based Syst.* **2023**, *262*, 110211. [CrossRef]
46. Xiao, C.; Chen, W. Trading the Twitter Sentiment with Reinforcement Learning. *arXiv* **2018**, arXiv:1801.02243.
47. Chen, Y.F.; Huang, S.H. Sentiment-influenced trading system based on multimodal deep reinforcement learning. *Appl. Soft Comput.* **2021**, *112*, 107788. [CrossRef]
48. Deng, Y.; Bao, F.; Kong, Y.; Ren, Z.; Dai, Q. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 653–664. [CrossRef] [PubMed]
49. Avramelou, L.; Nousi, P.; Passalis, N.; Tefas, A. Deep reinforcement learning for financial trading using multi-modal features. *Expert Syst. Appl.* **2024**, *238*, 121849. [CrossRef]
50. He, X.Z.; Lin, S. Reinforcement learning equilibrium in limit order markets. *J. Econ. Dyn. Control* **2022**, *144*, 104497. [CrossRef]
51. Altuner, A.B.; Kilimci, Z.H. A novel deep reinforcement learning based stock price prediction using knowledge graph and community aware sentiments. *Turk. J. Electr. Eng. Comput. Sci.* **2022**, *30*, 1506–1524. [CrossRef]
52. Moniruzzaman, M.; Sultana, A.; Kaddoum, G. Peer-to-peer energy trading in a local energy market using quantum reinforcement learning. In *2025 IEEE/ACIS 29th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*; IEEE: Piscataway, NJ, USA, 2025; pp. 796–801.
53. Long, L.; Zhang, C.; Ma, C.; Wang, H.; Ji, L.; Gao, F.; Zhang, J.; Qiu, K. Algorithmic trading by reinforcement learning in a collaborative manner. *Appl. Soft Comput.* **2026**, *197*, 115168. [CrossRef]
54. Ma, C.; Gao, F.; Ji, L.; Zhang, C.; Long, L.; Zhang, J. Toward personalized risk-sensitive decision-making: A novel risk preference adaptive distributional reinforcement learning algorithm for stock trading. *Appl. Soft Comput.* **2026**, *186*, 114269. [CrossRef]
55. Wu, J.; Yang, R.; Li, M.; Sun, E.; Yu, F.R.; Zhang, D.; Si, P. Efficiency Optimization for Blockchain-Enabled V2V Energy Trading with Dynamic Clustering Based on Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2026**, *75*, 405–418. [CrossRef]
56. Qin, M.; Cai, X.; Li, Y.; Xia, H.; Zong, C.; Sun, S.; Wang, X.; An, B. FineFT: Efficient and Risk-Aware Ensemble Reinforcement Learning for Futures Trading. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*; Association for Computing Machinery: New York, NY, USA, 2026; pp. 1194–1203. [CrossRef]
57. Ghadiri, H.; Hajizadeh, E. Designing a cryptocurrency trading system with deep reinforcement learning utilizing LSTM neural networks and XGBoost feature selection. *Appl. Soft Comput.* **2025**, *175*, 113029. [CrossRef]
58. Chen, Z.; Cai, X.; Qin, M.; An, B. Ophr: Mastering volatility trading with multi-agent deep reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2026**, *38*, 52889–52919.

59. Altermatt, A.; Dornberger, R.; Hanne, T. Open-ended Deep Reinforcement Learning for a Bitcoin Trading Bot. In *2025 7th International Symposium on Computational and Business Intelligence (ISCBI)*; IEEE: Piscataway, NJ, USA, 2025; pp. 40–45.
60. Tao, L.; Gong, D.; Du, K.; Zhang, T.; Chen, L.; Xia, S. Non-cooperative Game-Based Power Trading Strategy for Multi-Microgrid based on Multi-Agent Deep Reinforcement Learning. In *2025 IEEE Industry Applications Society Annual Meeting (IAS)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6.
61. Darmanin, A.; Vella, V. Language model guided reinforcement learning in quantitative trading. In *2025 3rd International Conference on Foundation and Large Language Models (FLLM)*; IEEE: Piscataway, NJ, USA, 2025; pp. 405–412.
62. Shah, M.I.A.; Barrett, E.; Mason, K. Uncertainty-aware knowledge transformers for peer-to-peer energy trading with multi-agent reinforcement learning. *arXiv* **2025**, arXiv:2507.16796.
63. Dasaklis, T.K.; Giannopoulos, P.G.; Malamas, V.; Tantis, G.; Patsakis, C. Multi-Agent Reinforcement Learning for EV energy management and trading using the Lightning Network. In *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*; IEEE: Piscataway, NJ, USA, 2025; Volume 1, pp. 1–6.
64. Xu, B.; Zhuang, Z.; Bian, H.; Wen, Q.; Li, C.; Qi, J. Energy trading optimisation of microgrids considering degradation: A self-attention-based deep reinforcement learning method. *Int. J. Control* **2025**, *98*, 2686–2697. [CrossRef]
65. Casares, A.J.M. Enhancing algorithmic trading with wavelet-based deep reinforcement learning: A multi-indicator approach. *Neural Comput. Appl.* **2025**, *37*, 25339–25385. [CrossRef]
66. Fan, D.; Ab Razak, N.H.; Soh, W.N. Financial trading decision model based on deep reinforcement learning for smart agricultural management. *PeerJ Comput. Sci.* **2025**, *11*, e3196. [CrossRef] [PubMed]
67. Ma, P.; Zeng, Z.; Lu, Z.; Lee, R.S.T. LORSTransformerDRL: A Novel Deep Reinforcement Learning Framework for Intelligent Stock Trading with Chaotic Oscillators and Attention Mechanisms. In *2025 IEEE International Conference on Big Data (BigData)*; IEEE: Piscataway, NJ, USA, 2025; pp. 6920–6929. [CrossRef]
68. Kumar, S.; Vishal, M.; Ravi, V. Explainable reinforcement learning on financial stock trading using SHAP. *arXiv* **2022**, arXiv:2208.08790.
69. Azevedo Takara, L.; Santos, A.A.P.; Mariani, V.C.; Coelho, L.d.S. Deep reinforcement learning applied to a sparse-reward trading environment with intraday data. *Expert Syst. Appl.* **2024**, *238*, 121897. [CrossRef]
70. Shen, Y.; Huang, R.; Yan, C.; Obermayer, K. Risk-averse reinforcement learning for algorithmic trading. In *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*; IEEE: Piscataway, NJ, USA, 2014; pp. 391–398. [CrossRef]
71. Li, Y.; Ni, P.; Chang, V. Application of deep reinforcement learning in stock trading strategies and stock forecasting. *Computing* **2020**, *102*, 1305–1322. [CrossRef]
72. Shavandi, A.; Khedmati, M. A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. *Expert Syst. Appl.* **2022**, *208*, 118124. [CrossRef]
73. Qin, M.; Sun, S.; Zhang, W.; Xia, H.; Wang, X.; An, B. Earnhft: Efficient hierarchical reinforcement learning for high frequency trading. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 14669–14676. [CrossRef]
74. Li, H.; Hai, M. Deep Reinforcement Learning Model for Stock Portfolio Management Based on Data Fusion. *Neural Process. Lett.* **2024**, *56*, 108. [CrossRef]
75. Taghian, M.; Asadi, A.; Safabakhsh, R. Learning financial asset-specific trading rules via deep reinforcement learning. *Expert Syst. Appl.* **2022**, *195*, 116523. [CrossRef]
76. Sun, S.; Xue, W.; Wang, R.; He, X.; Zhu, J.; Li, J.; An, B. DeepScalper: A Risk-Aware Reinforcement Learning Framework to Capture Fleeting Intraday Trading Opportunities. *arXiv* **2022**, arXiv:2201.09058.
77. Cornalba, F.; Disselkamp, C.; Scassola, D.; Helf, C. Multi-Objective reward generalization: Improving performance of Deep Reinforcement Learning for applications in single-asset trading. *arXiv* **2023**, arXiv:2203.04579.
78. Han, W.; Huang, J.; Xie, Q.; Zhang, B.; Lai, Y.; Peng, M. Mastering Pair Trading with Risk-Aware Recurrent Reinforcement Learning. *arXiv* **2023**, arXiv:2304.00364.
79. Kumar, D.; Pawar, P.P.; Meesala, M.K.; Pareek, P.K.; Addula, S.R.; Shwetha, K. Enhanced Stock Market Trend Prediction on the Indonesia Stock Exchange Using Improved Bacterial Foraging Optimization and Elitist Whale Optimization Algorithms. In *2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–8.
80. Yang, S.Y.; Yu, Y.; Almahdi, S. An investor sentiment reward-based trading system using Gaussian inverse reinforcement learning algorithm. *Expert Syst. Appl.* **2018**, *114*, 388–401. [CrossRef]
81. Liu, P.; Zhang, Y.; Bao, F.; Yao, X.; Zhang, C. Multi-type data fusion framework based on deep reinforcement learning for algorithmic trading. *Appl. Intell.* **2022**, *53*, 1683–1706. [CrossRef]
82. Chen, T.; Su, W. Indirect Customer-to-Customer Energy Trading with Reinforcement Learning. *IEEE Trans. Smart Grid* **2019**, *10*, 4338–4348. [CrossRef]
83. Chen, T.; Su, W. Local Energy Trading Behavior Modeling with Deep Reinforcement Learning. *IEEE Access* **2018**, *6*, 62806–62814. [CrossRef]

84. Ghosh, S.; Subramanian, E.; Bhat, S.P.; Gujar, S.; Paruchuri, P. VidyutVanika: A reinforcement learning based broker agent for a power trading competition. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 914–921. [CrossRef]
85. Wang, H.; Huang, T.; Liao, X.; Abu-Rub, H.; Chen, G. Reinforcement Learning in Energy Trading Game Among Smart Microgrids. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5109–5119. [CrossRef]
86. Wang, H.; Huang, T.; Liao, X.; Abu-Rub, H.; Chen, G. Reinforcement Learning for Constrained Energy Trading Games with Incomplete Information. *IEEE Trans. Cybern.* **2017**, *47*, 3404–3416. [CrossRef] [PubMed]
87. Xiao, L.; Xiao, X.; Dai, C.; Pengy, M.; Wang, L.; Poor, H.V. Reinforcement Learning-based Energy Trading for Microgrids. *arXiv* **2018**, arXiv:1801.06285.
88. Duan, Z.; Chen, C.; Cheng, D.; Liang, Y.; Qian, W. Optimal Action Space Search: An Effective Deep Reinforcement Learning Method for Algorithmic Trading. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 406–415. [CrossRef]
89. Deng, S.; Zheng, Z.; He, H.; Shao, J. Deep Reinforcement Learning for Stock Trading with Behavioral Finance Strategy. In *International Conference on Advanced Data Mining and Applications*; Springer: Cham, Switzerland, 2023; pp. 535–549.
90. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M.A. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
91. Moody, J.E.; Wu, L.; Liao, Y.; Saffell, M. Performance functions and reinforcement learning for trading systems and portfolios. *J. Forecast.* **1998**, *17*, 441–470. [CrossRef]
92. Moody, J.; Wu, L. Optimization of trading systems and portfolios. In *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*; IEEE: Piscataway, NJ, USA, 1997; pp. 300–307. [CrossRef]
93. Moody, J.; Saffell, M. Learning to trade via direct reinforcement. *IEEE Trans. Neural Netw.* **2001**, *12*, 875–889. [CrossRef] [PubMed]
94. Fiorini, P.M.; Fiorini, P.G. A Simple Reinforcement Learning Algorithm for Stock Trading. In *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*; IEEE: Piscataway, NJ, USA, 2021; Volume 2, pp. 824–830. [CrossRef]
95. Bertoluzzo, F.; Corazza, M. Making Financial Trading by Recurrent Reinforcement Learning. In *Knowledge-Based Intelligent Information and Engineering Systems*; Apolloni, B., Howlett, R.J., Jain, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 619–626.
96. Maringer, D.; Ramtohul, T. Threshold Recurrent Reinforcement Learning Model for Automated Trading. In *Applications of Evolutionary Computation*; Di Chio, C., Brabazon, A., Di Caro, G.A., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O'Neill, M., et al., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 212–221.
97. Maringer, D.; Ramtohul, T. Regime-switching recurrent reinforcement learning in automated trading. In *Natural Computing in Computational Finance: Volume 4*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 93–121.
98. Zhang, J. Automating Transition Functions: A Way to Improve Trading Profits with Recurrent Reinforcement Learning. In *Artificial Intelligence Applications and Innovations*; Iliadis, L., Maglogiannis, I., Papadopoulos, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 39–49.
99. Si, W.; Li, J.; Ding, P.; Rao, R. A Multi-objective Deep Reinforcement Learning Approach for Stock Index Future's Intraday Trading. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*; IEEE: Piscataway, NJ, USA, 2017; Volume 2, pp. 431–436. [CrossRef]
100. Jiang, Z.; Xu, D.; Liang, J. A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. *arXiv* **2017**, arXiv:1706.10059.
101. Chen, C.T.; Chen, A.P.; Huang, S.H. Cloning Strategies from Trading Records Using Agent-Based Reinforcement Learning Algorithm. In *2018 IEEE International Conference on Agents (ICA)*; IEEE: Piscataway, NJ, USA, 2018; pp. 34–37. [CrossRef]
102. Lei, K.; Zhang, B.; Li, Y.; Yang, M.; Shen, Y. Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. *Expert Syst. Appl.* **2020**, *140*, 112872. [CrossRef]
103. Weng, L.; Sun, X.; Xia, M.; Liu, J.; Xu, Y. Portfolio trading system of digital currencies: A deep reinforcement learning with multidimensional attention gating mechanism. *Neurocomputing* **2020**, *402*, 171–182. [CrossRef]
104. Gold, C. FX trading via recurrent reinforcement learning. In *2003 IEEE International Conference on Computational Intelligence for Financial Engineering*; IEEE: Piscataway, NJ, USA, 2003; pp. 363–370. [CrossRef]
105. Jiang, Z.; Liang, J. Cryptocurrency Portfolio Management with Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1612.01277.
106. Zhang, W.; Yin, T.; Zhao, Y.; Han, B.; Liu, H. Reinforcement Learning for Stock Prediction and High-Frequency Trading with T+1 Rules. *IEEE Access* **2023**, *11*, 14115–14127. [CrossRef]
107. Almahdi, S.; Yang, S.Y. An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Syst. Appl.* **2017**, *87*, 267–279. [CrossRef]
108. Almahdi, S.; Yang, S.Y. A constrained portfolio trading system using particle swarm algorithm and recurrent reinforcement learning. *Expert Syst. Appl.* **2019**, *130*, 145–156. [CrossRef]

109. Dempster, M.; Leemans, V. An automated FX trading system using adaptive reinforcement learning. *Expert Syst. Appl.* **2006**, *30*, 543–552. [CrossRef]
110. Borrageiro, G.; Firoozye, N.; Barucca, P. Reinforcement Learning for Systematic FX Trading. *IEEE Access* **2022**, *10*, 5024–5036. [CrossRef]
111. Gorse, D. Application of stochastic recurrent reinforcement learning to index trading. In *Proceedings of the ESANN 2011 Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*; ESANN: Bruges, Belgium, 2011; pp. 981–993.
112. Luo, D.; Liao, W.; Yan, R. Lower Risks, Better Choices: Stock Correlation Based Portfolio Selection in Stock Markets. In *Proceedings of the Companion Proceedings of the ACM Web Conference 2023*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1–4.
113. Kanakaian, S.; Martin, J.P. Deep Reinforcement Learning for Portfolio Management with Dependency Factor. In *2023 9th International Conference on Smart Computing and Communications (ICSCC)*; IEEE: Piscataway, NJ, USA, 2023; pp. 443–448.
114. Huang, Y.; Wan, X.; Zhang, L.; Lu, X. A novel deep reinforcement learning framework with BiLSTM-Attention networks for algorithmic trading. *Expert Syst. Appl.* **2024**, *240*, 122581. [CrossRef]
115. Maree, C.; Omlin, C.W. Balancing Profit, Risk, and Sustainability for Portfolio Management. In *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–8.
116. Zong, C.; Qin, M.; Xia, H.; An, B. ArchetypeTrader: Reinforcement Learning for Selecting and Refining Learnable Strategic Archetypes in Quantitative Trading. *Proc. AAAI Conf. Artif. Intell.* **2026**, *40*, 29269–29276. [CrossRef]
117. Deng, Z.; Yan, W.; Yu, C.; Wang, J. AlphaQuanter: An End-to-End Tool-Augmented Agentic Reinforcement Learning Framework for Stock Trading. In *Findings of the Association for Computational Linguistics: ACL 2026*; Liakata, M., Moreira, V.P., Zhang, J., Jurgens, D., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2026; pp. 9373–9394. [CrossRef]
118. Shen, Z.; Huang, H. An adaptive quantitative trading strategy optimization framework based on meta reinforcement learning and cognitive game theory. *Appl. Intell.* **2025**, *55*, 689. [CrossRef]
119. Huang, J.C.; Chen, C.T.; Chang, C.C.; Huang, S.H. Strategy allocation for financial trading using competitive reinforcement learning and fuzzy logic. *Appl. Soft Comput.* **2025**, *185*, 113927. [CrossRef]
120. Zhang, J.; Maringer, D. Using a genetic algorithm to improve recurrent reinforcement learning for equity trading. *Comput. Econ.* **2016**, *47*, 551–567. [CrossRef]
121. Zhang, H.; Shi, Z.; Hu, Y.; Ding, W.; Kuruoğlu, E.E.; Zhang, X.P. Optimizing Trading Strategies in Quantitative Markets Using Multi-Agent Reinforcement Learning. In *ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2024; pp. 136–140.
122. Felizardo, L.K.; Lima Paiva, F.C.; de Vita Graves, C.; Matsumoto, E.Y.; Costa, A.H.R.; Del-Moral-Hernandez, E.; Brandimarte, P. Outperforming algorithmic trading reinforcement learning systems: A supervised approach to the cryptocurrency market. *Expert Syst. Appl.* **2022**, *202*, 117259. [CrossRef]
123. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
124. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
125. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2019**, arXiv:1509.02971.
126. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
127. Azhikodan, A.R.; Bhat, A.G.K.; Jadhav, M.V. Stock Trading Bot Using Deep Reinforcement Learning. In *Innovations in Computer Science and Engineering*; Saini, H.S., Sayal, R., Govardhan, A., Buyya, R., Eds.; Springer: Singapore, 2019; pp. 41–49.
128. Tang, L. An actor-critic-based portfolio investment method inspired by benefit-risk optimization. *J. Algorithms Comput. Technol.* **2018**, *12*, 351–360. [CrossRef]
129. Kabbani, T.; Duman, E. Deep Reinforcement Learning Approach for Trading Automation in the Stock Market. *IEEE Access* **2022**, *10*, 93564–93574. [CrossRef]
130. El Akraoui, B.; Daoui, C. Deep Reinforcement Learning for Bitcoin Trading. In *Business Intelligence*; Fakir, M., Baslam, M., El Ayachi, R., Eds.; Springer: Cham, Switzerland, 2022; pp. 82–93.
131. Liang, Z.; Chen, H.; Zhu, J.; Jiang, K.; Li, Y. Adversarial Deep Reinforcement Learning in Portfolio Management. *arXiv* **2018**, arXiv:1808.09940.
132. Liu, X.Y.; Xiong, Z.; Zhong, S.; Yang, H.; Walid, A. Practical Deep Reinforcement Learning Approach for Stock Trading. *arXiv* **2022**, arXiv:1811.07522.
133. Tsantekidis, A.; Passalis, N.; Tefas, A. Diversity-driven knowledge distillation for financial trading using Deep Reinforcement Learning. *Neural Netw.* **2021**, *140*, 193–202. [CrossRef] [PubMed]

134. Wu, X.; Chen, H.; Wang, J.; Troiano, L.; Loia, V.; Fujita, H. Adaptive stock trading strategies with deep reinforcement learning methods. *Inf. Sci.* **2020**, *538*, 142–158. [CrossRef]
135. Abdelraouf Shokery, R.; Abdelmoez, W.; Shoukry, A. A synchronous deep reinforcement learning model for automated multi-stock trading. *Prog. Artif. Intell.* **2021**, *10*, 83–97. [CrossRef]
136. Fengqian, D.; Chao, L. An Adaptive Financial Trading System Using Deep Reinforcement Learning with Candlestick Decomposing Features. *IEEE Access* **2020**, *8*, 63666–63678. [CrossRef]
137. Park, D.Y.; Lee, K.H. Practical Algorithmic Trading Using State Representation Learning and Imitative Reinforcement Learning. *IEEE Access* **2021**, *9*, 152310–152321. [CrossRef]
138. Sawhney, R.; Wadhwa, A.; Agarwal, S.; Shah, R. Quantitative day trading from natural language using reinforcement learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 4018–4030.
139. Li, H.; Dagli, C.H.; Enke, D. Short-term Stock Market Timing Prediction under Reinforcement Learning Schemes. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*; IEEE: Piscataway, NJ, USA, 2007; pp. 233–240. [CrossRef]
140. Jin, B. An intelligent algorithmic trading based on a risk-return reinforcement learning algorithm. *arXiv* **2022**, arXiv:2208.10707.
141. Bui, V.H.; Hussain, A.; Su, W. A Dynamic Internal Trading Price Strategy for Networked Microgrids: A Deep Reinforcement Learning-Based Game-Theoretic Approach. *IEEE Trans. Smart Grid* **2022**, *13*, 3408–3421. [CrossRef]
142. Yan, L.; Chen, X.; Chen, Y.; Wen, J. A Hierarchical Deep Reinforcement Learning-Based Community Energy Trading Scheme for a Neighborhood of Smart Households. *IEEE Trans. Smart Grid* **2022**, *13*, 4747–4758. [CrossRef]
143. Pereira, H.; Gomes, L.; Vale, Z. Peer-to-peer energy trading optimization in energy communities using multi-agent deep reinforcement learning. *Energy Inform.* **2022**, *5*, 44. [CrossRef]
144. Tran Van, Q.; Nguyen Bao, T.; Pham Minh, T. Integrated Hybrid Approaches for Stock Market Prediction with Deep Learning, Technical Analysis, and Reinforcement Learning. In *Proceedings of the 12th International Symposium on Information and Communication Technology*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 213–220.
145. Jiang, Y.; Olmo, J.; Atwi, M. Deep reinforcement learning for portfolio selection. *Glob. Finance J.* **2024**, *62*, 101016. [CrossRef]
146. Guo, M.; Liu, J.; Luo, Z.; Han, X. Deep reinforcement learning for pairs trading: Evidence from China black series futures. *Int. Rev. Econ. Finance* **2024**, *93*, 981–993. [CrossRef]
147. Hao, Z.; Zhang, H.; Zhang, Y. Stock portfolio management by using fuzzy ensemble deep reinforcement learning algorithm. *J. Risk Finance Manag.* **2023**, *16*, 201. [CrossRef]
148. Yu, Y.; Li, H.; Chen, Z.; Jiang, Y.; Li, Y.; Zhang, D.; Liu, R.; Suchow, J.W.; Khashanah, K. FinMem: A performance-enhanced LLM trading agent with layered memory and character design. In *Proceedings of the AAAI Symposium Series*; Association for the Advancement of Artificial Intelligence: Washington DC, USA, 2024; Volume 3, pp. 595–597.
149. Ding, W.; Chen, Z.; Jiang, H.; Lin, Y.; Lin, F. Trend-Heuristic Reinforcement Learning Framework for News-Oriented Stock Portfolio Management. In *ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2024; pp. 5120–5124.
150. Zhang, W.; Zhao, L.; Xia, H.; Sun, S.; Sun, J.; Qin, M.; Li, X.; Zhao, Y.; Zhao, Y.; Cai, X.; et al. A Multimodal Foundation Agent for Financial Trading: Tool-Augmented, Diversified, and Generalist. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 4314–4325.
151. Miao, R.; Zhang, X.; Yan, H.; Chen, C. A dynamic financial knowledge graph based on reinforcement learning and transfer learning. In *2019 IEEE International Conference on Big Data (Big Data)*; IEEE: Piscataway, NJ, USA, 2019; pp. 5370–5378.
152. Lan, P.; Chen, S.; Wang, F. Carbon and electricity trading for the green hydrogen-based integrated energy system: A deep reinforcement learning-based scheduling optimization. *Renew. Energy* **2025**, *256*, 124176. [CrossRef]
153. Ahsan, S.M.; Gholizadeh, N.; Musilek, P. Multi-agent systems in networked microgrids: Reinforcement learning and strategic pricing mechanisms. *Renew. Energy* **2025**, *254*, 123678. [CrossRef]
154. Xiong, G.; Deng, Z.; Wang, K.; Cao, Y.; Li, H.; Yu, Y.; Peng, X.; Lin, M.; Smith, K.E.; Liu, X.Y.; et al. Flag-trader: Fusion llm-agent with gradient-based reinforcement learning for financial trading. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 13921–13934.
155. Song, G.; Zhao, T.; Ma, X.; Lin, P.; Cui, C. Reinforcement learning-based portfolio optimization with deterministic state transition. *Inf. Sci.* **2025**, *690*, 121538. [CrossRef]
156. Aritonang, P.K.; Wiryono, S.K.; Faturohman, T. Hidden-layer configurations in reinforcement learning models for stock portfolio optimization. *Intell. Syst. Appl.* **2025**, *25*, 200467. [CrossRef]
157. Alzaman, C. Optimizing portfolio selection through stock ranking and matching: A reinforcement learning approach. *Expert Syst. Appl.* **2025**, *269*, 126430. [CrossRef]
158. Sattar, A.; Ansari, Y.; Bukhari, F.; Bukhari, M.; Park, S.; Rho, S. Driving Investor Success: Novel Macro Momentum Accelerated Reinforcement Learning Framework for Stock Trading. *IEEE Access* **2026**, *14*, 56461–56482. [CrossRef]

159. Cornejo, T.; Baladron, J. Reinforcement Learning for the Development of Trading Strategies with Market-Uncorrelated Returns. In *2025 44th International Conference of the Chilean Computer Science Society (SCCC)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–4.
160. Liu, Y.K.; Pan, Y.H.; Lu, P.F.; Tsai, Y.C.; Chen, S.Y.C. Quantum-enhanced reinforcement learning with LSTM forecasting signals for optimizing fintech trading decisions. In *2025 IEEE International Conference on Quantum Computing and Engineering (QCE)*; IEEE: Piscataway, NJ, USA, 2025; Volume 2, pp. 235–240.
161. Huang, Y.; Yan, J.; Lu, X.; Zhao, L. Explainable reinforcement learning with adaptive feature selection for stock trading. *Appl. Soft Comput.* **2025**, *189*, 114543. [CrossRef]
162. Chen, C.S.; Zhang, X.; Chen, Y.C. Quantum reinforcement learning trading agent for sector rotation in the taiwan stock market. In *2025 IEEE International Conference on Quantum Computing and Engineering (QCE)*; IEEE: Piscataway, NJ, USA, 2025; Volume 2, pp. 217–222.
163. Sevgili, S.O.; Gündüz Öğüdücü, Ş. Multi-agent Reinforcement Learning for Financial Market Trading: An Expert-System Approach. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Cham, Switzerland, 2025; pp. 509–521.
164. Pradana, Y.A.; Irawan, M.I.; Mukhlash, I.; Putri, E.R.M. Carbon trading strategy in the European market based on risk analysis using deep stochastic reinforcement learning. *Appl. Soft Comput.* **2026**, *201*, 115431. [CrossRef]
165. Yuan, X.; Wang, J.; Gu, S.; Guo, Y.; Qi, A.; Li, S.; Zhao, L. Adaptive multi-agent stock trading decision support system based on deep reinforcement learning. *Eng. Appl. Artif. Intell.* **2026**, *169*, 114130. [CrossRef]
166. Han, F.; Ling, Q.; Lu, S.; Han, H. Feature enrichment imitative reinforcement learning for high-frequency trading. *Expert Syst. Appl.* **2026**, *296*, 129043. [CrossRef]
167. Nouira, A.Y.; Amara, M. A multimodal data-driven ensemble deep reinforcement learning model for automated cryptocurrency trading. *Int. J. Data Sci. Anal.* **2026**, *22*, 193. [CrossRef]
168. Cao, J.; Zhang, X.; Sun, Y.; Zhou, X.; Li, H. Three-way decision-guided hierarchical reinforcement learning for high-frequency trading. *Inf. Sci.* **2026**, *757*, 123880. [CrossRef]
169. Jiang, J.; Li, Z.; Cui, A.; Zhou, X.; Li, B.; Sun, D. Towards robust deep reinforcement learning-based quantitative trading with neuro-symbolic trend analysis. *Neural Netw.* **2026**, *201*, 108924. [CrossRef] [PubMed]
170. Zhao, C.; Song, M.; Wang, W.; Gao, C. Personalized Dynamic Operating Envelope-Based P2P Energy Trading: A Novel Multiagent Safe Reinforcement Learning. *IEEE Trans. Smart Grid* **2026**, *17*, 3248–3261. [CrossRef]
171. Thanasarn, T.; Anuntachai, A.; Netisopakul, P. A Comparative Study of Deep Reinforcement Learning Agents for Gold Trading with Technical Indicators and LLM-Filtered News Sentiment. In *Proceedings of the 14th International Conference on Advances in Information Technology*; Association for Computing Machinery: New York, NY, USA, 2026. [CrossRef]
172. Ren, J.; Gao, H.; Zhao, L.; Kang, Q.; Xiao, G.; Sun, Y. Multi-agent Reinforcement Learning for Low-Carbon P2P Energy Trading among Self-Interested Microgrids. In *ICC 2026—IEEE International Conference on Communications*; IEEE: Piscataway, NJ, USA, 2026; pp. 1–6. [CrossRef]
173. Wali, A.; Lera, I.; Belguith, L.H. Explainable Federated Deep Reinforcement Learning for Distributed Trading Agents over 6G-Enabled Wireless Edge Networks. In *2026 International Wireless Communications and Mobile Computing (IWCMC)*; IEEE: Piscataway, NJ, USA, 2026; pp. 213–218. [CrossRef]
174. Nalmpantis, A.; Passalis, N.; Tefas, A. Explicit Bandwidth Learning for FOREX Trading Using Deep Reinforcement Learning. *IEEE Signal Process. Lett.* **2025**, *32*, 686–690. [CrossRef]
175. Feng, H.; Wang, Y.; Zhong, S.; Yuan, T.; Quan, Z. Federated Reinforcement Learning in Stock Trading Execution: The FPPO Algorithm for Information Security. *IEEE Access* **2025**, *13*, 25074–25086. [CrossRef]
176. Roshanpour, R.; Parsanejad, M.; Saman Pishvaee, M. Reinforcement Learning in Multi-Asset Trading: Integrating Temporal-Frequency Features and Novel Risk-Adaptability Indices. *IEEE Access* **2025**, *13*, 195016–195030. [CrossRef]
177. Sattar, A.; Sarwar, A.; Gillani, S.; Bukhari, M.; Rho, S.; Faseeh, M. A Novel RMS-Driven Deep Reinforcement Learning for Optimized Portfolio Management in Stock Trading. *IEEE Access* **2025**, *13*, 42813–42835. [CrossRef]
178. Kaur, B.; Sidhu, B.K.; Bhathal, G.S. A Hybrid Deep Reinforcement Learning Approach for Algorithmic Trading in Commodity Futures Markets. *Appl. Comput. Intell. Soft Comput.* **2025**, *2025*, 5993683. [CrossRef]
179. Drossas, I.; Avramelou, L.; Passalis, N.; Tefas, A. Expert-Guided Deep Reinforcement Learning for Autonomous Financial Asset Trading. In *2025 16th International Conference on Information, Intelligence, Systems & Applications (IISA)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6.
180. Avramelou, L.; Litsidis, D.; Passalis, N.; Tefas, A. Model Merging for Deep Reinforcement Learning in Autonomous Financial Trading Systems. In *2025 16th International Conference on Information, Intelligence, Systems & Applications (IISA)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–5.
181. Wang, Q.; Yang, Y. Carbon trading supply chain management based on constrained deep reinforcement learning. *Auton. Agents Multi-Agent Syst.* **2024**, *38*, 38. [CrossRef]

182. Kosidphokin, V.; Loedtrakunchai, P.; Sinamnuaiphon, N.; Kuptanon, S. FinRL: Adaptive Model Selection for Reinforcement Learning in Stock Trading. In *2025 IEEE 11th International Conference on Intelligent Data and Security (IDS)*; IEEE: Piscataway, NJ, USA, 2025; pp. 71–72.
183. Mohl, V.; Frey, S.; Leyland, R.; Li, K.; Nigmatulin, G.; Cucuringu, M.; Zohren, S.; Foerster, J.; Calinescu, A. JaxMARL-HFT: GPU-accelerated large-scale multi-agent reinforcement learning for high-frequency trading. In *Proceedings of the 6th ACM International Conference on AI in Finance*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 18–26.
184. Bi, Y.; Wang, Z.; Zhao, L.; Ventre, C. Long-term financial forecasting and trading via multi-agent reinforcement learning. In *Proceedings of the 6th ACM International Conference on AI in Finance*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 788–796.
185. Botunac, I.; Petković, T.; Bosna, J. Automated Trading Framework Using LLM-Driven Features and Deep Reinforcement Learning. *Big Data Cogn. Comput.* **2025**, *9*, 317. [CrossRef]
186. Zhou, C.; Huang, Y.; Kong, Y.; Lu, X. Enhancing trading strategies by combining incremental reinforcement learning and self-supervised prediction. *Expert Syst. Appl.* **2025**, *289*, 128297. [CrossRef]
187. Kang, C.; Woo, J.; Hong, J.K. A2C Reinforcement Learning for Cryptocurrency Trading: Analyzing the Impact of Price Prediction Accuracy and Data Granularity. *Int. J. Netw. Manag.* **2025**, *35*, e70024. [CrossRef]
188. Jing, Z.; Yuan, S.Q.; Lv, X.F.; Kang, H.W. Optimizing Power Dispatch and Market Trading Under Renewable Energy Uncertainty Using OPRTDPG Deep Reinforcement Learning. *Informatica* **2025**, *49*, 371–382. [CrossRef]
189. Orra, A.; Choudhary, H.; Sharma, A.; Thakur, M. Enhancing deep reinforcement learning for stock trading: A reward shaping approach via expert feedback. *Knowl. Inf. Syst.* **2025**, *67*, 11075–11094. [CrossRef]
190. Li, Y.; Dong, X.; Wu, Z.; Gao, J.; Zhang, T.; Yu, L. Reinforcement learning with temporal and variable dependency-aware transformer for stock trading optimization. *Neural Netw.* **2025**, *192*, 107905. [CrossRef] [PubMed]
191. Guevara, C. Stock market trading via actor-critic reinforcement learning and adaptable data structure. *PeerJ Comput. Sci.* **2025**, *11*, e2690. [CrossRef] [PubMed]
192. Quintero, C.; Leon, D.; Sandoval, J.; Hernandez, G. Deep reinforcement learning in continuous action spaces for pair trading: A comparative study of A2 c and PPO. *SN Comput. Sci.* **2025**, *6*, 407. [CrossRef]
193. Ramya, D.; Suresha. Reinforcement learning driven trading algorithm with optimized stock portfolio management scheme to control financial risk. *SN Comput. Sci.* **2024**, *6*, 19. [CrossRef]
194. Han, D.; Kim, Y. Enhancing Reinforcement Learning-Based Crypto Asset Trading: Focusing on the Korean Venue Share Indicator. *Systems* **2026**, *14*, 111. [CrossRef]
195. Lu, R.; Wu, N.; Yang, T.; Chen, Y.; Sun, M.; Wang, D.; Peng, X. SMA-PDPPO: Safe Multiagent Primal-Dual Deep Reinforcement Learning for Industrial Parks Energy Trading. *IEEE Trans. Ind. Inform.* **2025**, *21*, 2640–2649. [CrossRef]
196. Liu, X.; Ye, Y.; Li, S.; Zhang, C.; Ma, Q.; Zhu, J. Carbon-Aware Peer-to-Peer Energy Trading in an Unbalanced Distribution Network via a Nash Equilibrium Discovery Deep Reinforcement Learning Approach. *IEEE Trans. Smart Grid* **2025**, *16*, 3392–3407. [CrossRef]
197. Ma, Q.; Liu, Z.; Ye, Y.; Liu, X. Network-Constrained P2P Trading: A Safety-Aware Decentralized Multi-Agent Reinforcement Learning Approach. *IEEE Trans. Smart Grid* **2025**, *16*, 5573–5588. [CrossRef]
198. Zhang, M.; Eliassen, F.; Taherkordi, A.; Jacobsen, H.A.; Li, Y.; Zhang, Y. Self-Determination Theory and Deep Reinforcement Learning for Personalized Energy Trading in Smart Grid. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 4216–4229. [CrossRef]
199. Li, Z.; Jiang, J.; Cao, Y.; Cui, A.; Wu, B.; Li, B.; Liu, Y.; Sun, D.D. Logic-q: Improving deep reinforcement learning-based quantitative trading via program sketch-based tuning. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 18584–18592. [CrossRef]
200. Hromyk, O.; Panasiuk, Y.; Matviichuk, L.; Dibrivna, E. Decision Support System for Stock Trading based on Reinforcement Learning Methods. In *2025 15th International Conference on Advanced Computer Information Technologies (ACIT)*; IEEE: Piscataway, NJ, USA, 2025; pp. 888–891. [CrossRef]
201. Du, N.; Zhao, Y.; Wang, J.; Zhu, Y.; Xie, S.; Yang, L.; Tong, Y.; Xu, S.; Zhang, W.; Tang, Z.; et al. Enhancing Cryptocurrency Trading Strategies: A Deep Reinforcement Learning Approach Integrating Multi-Source LLM Sentiment Analysis. In *2025 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CiFer)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–7. [CrossRef]
202. Majidi, N.; Shamsi, M.; Marvasti, F. Algorithmic Trading Using Continuous Action Space Deep Reinforcement Learning. *arXiv* **2022**, arXiv:2210.03469.
203. Shen, Z.; Mao, X.; Wang, C.; Zhao, D.; Zhao, S. Smart Trading: A Novel Reinforcement Learning Framework for Quantitative Trading in Noisy Markets. In *Advanced Intelligent Computing Technology and Applications*; Huang, D.S., Zhang, X., Zhang, Q., Eds.; Springer: Singapore, 2024; pp. 158–170.
204. Day, M.Y.; Yang, C.Y.; Ni, Y. Portfolio dynamic trading strategies using deep reinforcement learning. *Soft Comput.* **2023**, *28*, 8715–8730. [CrossRef]

205. Zou, J.; Lou, J.; Wang, B.; Liu, S. A Novel Deep Reinforcement Learning Based Automated Stock Trading System Using Cascaded LSTM Networks. *arXiv* **2023**, arXiv:2212.02721.
206. Liu, S.; Han, S.; Zhu, S. Reinforcement Learning-Based Energy Trading and Management of Regional Interconnected Microgrids. *IEEE Trans. Smart Grid* **2023**, *14*, 2047–2059. [CrossRef]
207. Zhang, C.; Duan, Y.; Chen, X.; Chen, J.; Li, J.; Zhao, L. Towards generalizable reinforcement learning for trade execution. *arXiv* **2023**, arXiv:2307.11685.
208. Huang, H.; Gao, T.; Gui, Y.; Guo, J.; Zhang, P. Stock Trading Optimization Through Model-Based Reinforcement Learning with Resistance Support Relative Strength. *arXiv* **2022**, arXiv:2205.15056.
209. Song, Z.; Jin, X.; Li, C. Safe-FinRL: A Low Bias and Variance Deep Reinforcement Learning Implementation for High-Freq Stock Trading. *arXiv* **2022**, arXiv:2206.05910.
210. Han, W.; Zhang, B.; Xie, Q.; Peng, M.; Lai, Y.; Huang, J. Select and Trade: Towards Unified Pair Trading with Hierarchical Reinforcement Learning. *arXiv* **2023**, arXiv:2301.10724.
211. Kwak, D.; Choi, S.; Chang, W. Self-attention based deep direct recurrent reinforcement learning with hybrid loss for trading signal generation. *Inf. Sci.* **2023**, *623*, 592–606. [CrossRef]
212. Mosavi, A.; Faghan, Y.; Ghamisi, P.; Duan, P.; Ardabili, S.F.; Salwana, E.; Band, S.S. Comprehensive review of deep reinforcement learning methods and applications in economics. *Mathematics* **2020**, *8*, 1640. [CrossRef]
213. Wolpert, D.; Macready, W. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
214. Berner, J.; Grohs, P.; Jentzen, A. Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations. *SIAM J. Math. Data Sci.* **2020**, *2*, 631–657. [CrossRef]
215. Advani, M.S.; Saxe, A.M.; Sompolinsky, H. High-dimensional dynamics of generalization error in neural networks. *Neural Netw.* **2020**, *132*, 428–446. [CrossRef] [PubMed]
216. Cao, L. Ai in finance: Challenges, techniques, and opportunities. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 64. [CrossRef]
217. Veldkamp, L. *Information Choice in Macroeconomics and Finance*; Princeton University Press: Princeton, NJ, USA, 2023.
218. Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; Philip, S.Y. Generalizing to unseen domains: A survey on domain generalization. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 8052–8072. [CrossRef]
219. Kuo, C.H.; Chen, C.T.; Lin, S.J.; Huang, S.H. Improving generalization in reinforcement learning–based trading by using a generative adversarial market model. *IEEE Access* **2021**, *9*, 50738–50754. [CrossRef]
220. Hu, Y.; Ma, X.; Sui, J.; Pun, M.O. PPMamba: A Pyramid Pooling Local Auxiliary SSM-Based Model for Remote Sensing Image Semantic Segmentation. *arXiv* **2024**, arXiv:2409.06309.
221. Wang, K.; Kang, B.; Shao, J.; Feng, J. Improving generalization in reinforcement learning with mixture regularization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7968–7978.
222. Hirchoua, B.; Ouhbi, B.; Frikh, B. Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy. *Expert Syst. Appl.* **2021**, *170*, 114553. [CrossRef]
223. Ngo, V.M.; Nguyen, H.H.; Van Nguyen, P. Does reinforcement learning outperform deep learning and traditional portfolio optimization models in frontier and developed financial markets? *Res. Int. Bus. Finance* **2023**, *65*, 101936. [CrossRef]
224. Liu, X.Y.; Xia, Z.; Yang, H.; Gao, J.; Zha, D.; Zhu, M.; Wang, C.D.; Wang, Z.; Guo, J. Dynamic datasets and market environments for financial reinforcement learning. *Mach. Learn.* **2024**, *113*, 2795–2839. [CrossRef]
225. Shuford, J. Deep reinforcement learning unleashing the power of AI in decision-making. *J. Artif. Intell. Gen. Sci. (JAIGS)* **2024**, *1*. [CrossRef]
226. Yang, P.; Zhang, L.; Liu, H.; Li, G. Reducing idleness in financial cloud services via multi-objective evolutionary reinforcement learning based load balancer. *Sci. China Inf. Sci.* **2024**, *67*, 120102. [CrossRef]
227. Khushi, M. QFRS: Quantitative Finance Reporting Standards for Forecasting, Evaluation and Trading Claims. *Artif. Intell. Rev.* **2026**, *in press*. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.