Global Economic Perspectives Vol 3 Issue 1 2025

Original Research Article

Open Access

Omniscient Pte.Ltd.

Research on Spread Arbitrage Strategy and Optimization Analysis Based on Deep Reinforcement Learning in High-Frequency Trading

Zu-He Xu*

Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, 215000, China

***Correspondence to:** Zu-He Xu, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, 215000, China, E-mail: 17803705856@163.com

Abstract: This paper explores the implementation of spread arbitrage strategies in high-frequency trading (HFT) environments by constructing an adaptive trading system based on deep reinforcement learning (DRL). The study focuses on four core components: multi-dimensional data integration and preprocessing, reinforcement learning algorithm framework design, trading signal generation and execution mechanism, and strategy optimization with risk control. A comprehensive high-frequency trading solution is proposed, which incorporates multi-level feature engineering, optimized state space design, refined execution decision-making, and dynamic parameter adjustment. This approach enhances the robustness and profitability of arbitrage strategies, providing both theoretical foundation and practical guidance for the field of quantitative trading.

Keywords: Deep Reinforcement Learning; High-Frequency Trading; Spread Arbitrage

Introduction

With the increasing digitalization of financial markets, high-frequency trading has become an indispensable component of modern financial systems. As one of the primary strategies in HFT, spread arbitrage offers advantages such as short risk exposure duration and theoretical market neutrality[1]. However, traditional spread arbitrage methods suffer from inefficiencies in information processing and significant execution slippage. This paper integrates deep reinforcement learning with high-frequency trading by constructing an adaptive learning-based trading system to improve the accuracy of spread identification and execution efficiency. The discussion is structured around four dimensions: data processing, model design, signal generation, and strategy optimization, with the aim of developing a comprehensive HFT solution.

1. Data Integration and Preprocessing Strategies

1.1 Data Acquisition Methods

The successful implementation of a high-frequency spread arbitrage strategy fundamentally depends on establishing a comprehensive, stable, and real-time data acquisition system. This paper designs a multi-source heterogeneous data acquisition framework by deploying low-latency data interfaces to establish direct connections to exchange feeds. Specialized hardware is used to accelerate the data transmission process, ensuring millisecond-level market data capture capability. Specifically, the acquisition system adopts a distributed architecture with geographically redundant server clusters, while a high-precision time synchronization mechanism ensures cross-market data alignment[2]. Adapter modules tailored to the specific APIs of different exchanges are developed for unified data formatting and storage, thereby constructing a standardized microstructure data lake.

To enhance data quality and stability, an adaptive sampling technique is introduced, dynamically adjusting the sampling frequency based on market volatility. This ensures data completeness at critical moments while optimizing system resource allocation. A multi-level caching mechanism and breakpoint-resume functionality effectively handle network jitter or exchange service interruptions, maximizing data continuity. Combined with a full-link monitoring system, key metrics such as latency and packet loss rates are tracked in real-time to ensure stable and efficient system operation, providing a solid data foundation for subsequent strategy execution.

1.2 Data Cleaning Scheme

Raw high-frequency market data often contains significant noise and anomalies; therefore, a scientifically rigorous data cleaning process is crucial for enhancing strategy robustness[3]. This paper proposes a multi-stage data cleaning scheme. First, anomaly detection algorithms are employed to identify and handle phenomena such as price spikes and volume anomalies. Statistical methods are used to set dynamic thresholds and validate against historical data distributions. Second, for the common issue of timestamp disorder in high-frequency data, an event reordering mechanism ensures that data is processed in its actual chronological order, avoiding erroneous trading signals caused by sequence errors.

Regarding quote data quality, multidimensional validation rules are designed, including bid-ask spread reasonableness checks and order book depth continuity validation. A rule engine automatically flags suspicious data for correction or removal. In cases of exchange downtime or missing data, intelligent imputation techniques are applied. These generate plausible substitute values by constructing a microstructure similarity model based on market conditions before and after the missing interval.

1.3 Feature Engineering Design

Feature engineering, as a key bridge between raw data and model training, directly impacts the quality of reinforcement learning decisions. This paper constructs a multi-level feature system. First, from the perspective of microstructure, order book shape features are extracted, including bid-ask imbalance indicators, order book elasticity coefficients, and slope of the market depth curve—capturing short-term price movement signals. Next, a trade volume analysis framework is introduced, designing indicators such as volume-weighted average price (VWAP) and trade volume entropy to quantify implied market intent and liquidity conditions[4].

To enhance feature representation, the framework applies nonlinear transformations and time series decomposition techniques to project raw features into a more discriminative feature space. Specifically, wavelet transforms are introduced to capture multi-scale market volatility patterns, and empirical mode decomposition (EMD) is used to separate trend and noise components. These are combined with autoencoders to achieve unsupervised feature dimensionality reduction and reconstruction.

2. Reinforcement Learning Model Design Strategies

2.1 Algorithm Framework Construction

The design of the deep reinforcement learning algorithm framework determines the core intelligence level of the high-frequency trading system. This paper adopts an improved algorithmic framework based on the Actor-Critic architecture, integrating the advantages of both DQN and DDPG to enable hybrid control over continuous and discrete action spaces, which is more aligned with the decision-making characteristics of high-frequency trading. The core model consists of a value network and a policy network: the value network evaluates the value of states, while the policy network directly outputs trading decisions. Asynchronous training is employed to improve learning efficiency.

Innovatively, this framework introduces a multi-agent cooperation mechanism, dividing the identification and execution of different market spread opportunities into multiple specialized agents. A central coordinator is used for resource allocation and task scheduling. At the algorithmic level, a Transformer-based attention mechanism is incorporated to enhance the modeling of temporal dependencies, and Graph Neural Networks (GNNs) are integrated to capture complex inter-market relationships, thereby improving the accuracy of cross-market spread identification.

2.2 State Definition Optimization

The design of the state space in reinforcement learning directly determines the model's perception capabilities and the upper bound of its decision quality. This paper goes beyond the limitations of traditional state definitions and constructs a multi-dimensional, multi-scale state representation system. Firstly, it introduces microstructural market state representations by encoding order book snapshots, trade records, and price movements into structured tensors, preserving detailed market depth information. Secondly, a macro-market state encoding mechanism is designed, incorporating features such as volatility indicators, trading activity levels, and liquidity assessments, allowing the model to perceive the overall operating condition of the market.

To address the challenges of processing high-dimensional states, a hierarchical state compression strategy is adopted. An adaptive feature selection mechanism dynamically filters the most informative state dimensions under current market conditions. Technically, a self-attention mechanism is introduced to assign weights to different state components, highlighting key information. Additionally, a memory-enhanced state representation is implemented using recurrent structures such as LSTMs, which helps retain historical state information and strengthens the model's ability to identify sequential patterns in the market.

3. Trade Signal Generation and Execution Strategy

3.1 Signal Extraction Methods

Trade signal extraction serves as the bridge between model outputs and actual trading behavior; its accuracy and timeliness directly impact the success rate of arbitrage strategies. This study constructs a multi-layered signal extraction framework. First, raw signals are generated by the reinforcement learning model, including predictions on spread direction, magnitude, and persistence. Second, a statistical arbitrage model is introduced for cross-validation by applying cointegration tests and mean-reversion assessments to confirm signal reliability. Third, a market microstructure analysis module is integrated to evaluate the feasibility of signal execution under current liquidity conditions, dynamically adjusting signal confidence levels. This multi-dimensional signal fusion significantly enhances signal quality and stability.

To counteract market noise, this framework incorporates adaptive filtering techniques, dynamically adjusting signal thresholds based on market volatility—raising activation thresholds during high-volatility periods to reduce false signals. On the technical side, a signal sequence stacking analysis module is designed to identify patterns in consecutive signals, capturing reinforced trends and turning points. A signal stratification mechanism is also embedded to categorize signals by confidence level, enabling priority-based execution and optimal resource allocation.

3.2 Execution Decision Mechanism

In high-frequency trading environments, the execution decision mechanism significantly affects the profitability of arbitrage strategies. This paper proposes an adaptive intelligent execution system that integrates market state awareness with cost-optimization frameworks. A dynamic liquidity evaluation model is first constructed to monitor real-time changes in market depth and forecast market impact costs, which are then used to adjust order sizes and order-splitting strategies. A trade timing optimization algorithm is developed to analyze microstructural price evolution and identify optimal execution windows, reducing the risk of adverse selection and front-running.

To address the common issue of slippage in high-frequency trading, an adaptive order type selection mechanism is developed. It dynamically selects between market orders, limit orders, and hidden orders based on current market conditions and signal strength. Additionally, a smart limit pricing strategy is implemented by analyzing order book structures to predict execution probabilities and determine the most favorable limit prices.

3.3 Trade Order Management

An efficient trade order management system is a critical component for ensuring the stable operation of trading strategies. This study develops a full-lifecycle trade order management framework, encompassing order generation, validation, distribution, execution monitoring, and result feedback. Firstly, a pre-trade risk control system is designed using a rules engine to perform real-time compliance checks on every trade, including volume limits and price deviation verification. Secondly, a dynamic order prioritization and scheduling mechanism is introduced, which evaluates expected returns, timing urgency, and risk exposure to optimally allocate resources.

To handle system failures and network latency issues, a multi-tier fault tolerance mechanism is implemented. This includes automatic cancellation of timed-out orders, intelligent reordering for partial fills, and emergency liquidation protocols to maintain risk control under extreme circumstances. Architecturally, the system is built on a high-availability distributed processing framework, utilizing redundancy and load-balancing technologies to ensure high concurrency and low latency in order processing.

4. Strategy Optimization and Risk Control Framework

4.1 Parameter Optimization Methods

Parameter optimization is a core component of reinforcement learning-based trading strategies, directly impacting model performance and profitability. This paper constructs a multi-layered parameter optimization framework. First, for model architecture parameters, a Bayesian optimization scheme is proposed. By building a surrogate performance model, the framework guides efficient exploration of the parameter space, enabling rapid convergence to a global optimum. Second, for execution-related parameters, an online learning mechanism is introduced to dynamically adjust execution thresholds and order sizes based on real-time market conditions. Third, for risk control parameters, a scenario-based simulation optimization method is developed. It evaluates strategy performance under extreme market conditions to determine safe boundaries for risk parameters.

To address the computational complexity of parameter optimization, a distributed parameter search framework is introduced. By decomposing tasks and leveraging parallel computing, the optimization efficiency is significantly enhanced. A parameter sensitivity analysis module is also implemented to identify the subset of parameters with the greatest impact on strategy performance, enabling rational allocation of optimization resources. On the technical side, a hybrid optimization approach that combines evolutionary algorithms and gradient descent is employed, balancing global exploration and local fine-tuning.

4.2 Model Performance Evaluation

A comprehensive model evaluation system is essential for ongoing strategy optimization. This study constructs a multi-dimensional evaluation framework that goes beyond traditional return-based metrics. From a financial perspective, composite performance indicators are designed, including risk-adjusted return, maximum drawdown, Sharpe ratio, and Sortino ratio, to comprehensively quantify the risk-return profile of the strategy. From a trading perspective, execution quality metrics are established, such as slippage statistics, fill rate, and signal accuracy, to assess operational effectiveness.

Given the characteristics of high-frequency trading, the framework incorporates microstructural impact assessment to evaluate the influence of strategy activity on market liquidity and price discovery, avoiding overly aggressive trading behaviors. A signal decay monitoring mechanism is also designed to track changes in signal effectiveness and detect signs of strategy degradation in advance. In terms of evaluation workflow, a layered assessment approach is adopted—from backtesting and paper trading to live trading—ensuring strategy robustness across all stages. A controlled experiment design is also introduced, enabling attribution analysis by comparing the proposed strategy with baseline benchmarks to isolate market factors from strategy-specific contributions.

4.3 Iterative Optimization Mechanism

An iterative strategy optimization mechanism is critical to ensure that trading systems continuously adapt to evolving market conditions. This paper constructs a closed-loop optimization framework for full-lifecycle strategy management. A periodic evaluation mechanism is designed to analyze strategy performance over fixed time windows and identify components requiring refinement. A trigger-based evaluation mechanism is also introduced, which initiates in-depth analysis automatically when key indicators deviate from predefined thresholds. A tiered optimization scheme is built, employing different levels of intervention—including parameter fine-tuning, model retraining, or architectural redesign—depending on the severity of performance degradation. Through a systematic optimization process, the strategy is ensured to remain in its optimal state at all times.

To enable continuous evolution, an online learning framework is implemented. It incrementally updates model parameters using real-time data streams to adapt to changes in market microstructure. Simultaneously, a knowledge retention mechanism is established through an experience replay buffer that stores historical successful cases to inform decision-making in new scenarios. On the technical side, an A/B testing architecture is adopted, running multiple strategy versions in parallel to scientifically validate the effectiveness of optimization measures.

Conclusion

This paper proposes a comprehensive HFT solution for spread arbitrage strategies based on deep reinforcement learning, forming a closed-loop framework encompassing data preprocessing, model design, trade execution, and strategy optimization. Through the construction of a multi-level feature engineering framework, optimization of reinforcement learning state space design, refinement of execution decision mechanisms, and dynamic parameter adjustment strategies, the system significantly improves the accuracy of spread identification and execution efficiency. Experimental results confirm that the proposed method demonstrates strong adaptability and robust profitability across various market conditions. Furthermore, the study emphasizes the importance of risk control and continuous optimization by establishing a multi-dimensional evaluation system and an iterative mechanism to ensure the long-term sustainability of the strategy.

References

[1] Rao Rui, Pan Zhisong, Li Wei, et al. Optimization Algorithm for High-Frequency Trading Based on Deep Reinforcement Learning [J]. Journal of Nanjing University of Science and Technology (Natural Science Edition), 2022, 46(3): 304–312.

[2] Wen Xinxian. Research on High-Frequency Quantitative Trading Strategy Based on Deep Reinforcement Learning [J]. Modern Electronic Technology, 2023, 46(2): 125–131.

[3] Sun Dachang, Bi Xiuchun. High-Frequency Trading Strategy Based on Deep Learning Algorithms and Its Profitability [J]. Journal of University of Science and Technology of China, 2018, 48(11): 923–932.

[4] Zhu Feng, Guo Wenjing, Yan Xiping. Volatility Prediction of High-Frequency Financial Data Based on Deep Learning [J]. Intelligent Computer and Applications, 2024, 14(9): 82–87.