Effectiveness Evaluation of High-Frequency Trading Machine Learning Prediction Models under Transaction Cost Constraints

Hongtao Chen

Jinan University – University of Birmingham Joint Institute, Jinan University, Guangzhou, China

midnight65412@gmail.com

Abstract. This study investigates the difference between the statistical effectiveness and economic effectiveness of short-term mid-price direction prediction models under transaction cost constraints, based on the preprocessed FI-2010 limit order book (LOB) dataset. Four three-class prediction models, including Logistic Regression, eXtreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM), were evaluated. Their predictions were mapped to long, neutral, and short trading signals, followed by simplified backtesting under different fixed transaction cost coefficients. Within a representative prediction window, CNN-LSTM achieved the highest Accuracy, Macro-F1, and Weighted-F1 scores, at 0.799, 0.751, and 0.793, respectively. However, with a 20-tick prediction window and a transaction cost coefficient of 0.001, the cumulative net return proxy of LSTM is 6.090, higher than that of CNN-LSTM (-4.606). Further analysis revealed that with a 100-tick prediction window, the cumulative net return proxy of all models is negative, and the correlation coefficient between Macro-F1 and the cumulative net return proxy is only about 0.21. The results show that stronger classification performance does not necessarily correspond to greater economic effectiveness. High-frequency trading prediction models should therefore be evaluated jointly in terms of classification performance, cumulative net return proxy, and turnover.

Keywords: High-frequency trading, Limit order book, Mid-price prediction, Machine learning, Transaction costs

1. Introduction

High-frequency trading is an important form of trading in modern financial markets. Unlike traditional investment analysis, which mainly relies on macroeconomic variables, company fundamentals, or daily market data, high-frequency trading focuses more on market microstructure information on a very short time scale, such as bid and ask prices, order book volume, order depth, changes in order flow, and bid-ask spreads. As a core data structure recording pending buy and sell orders in the market, LOB can

© The Author(s) 2026
V. Erokhin et al. (eds.), *Proceedings of the 6th International Conference on Internet Finance and Digital Economy (ICIFDE 2026)*, Advances in Economics, Business and Management Research 412,
https://doi.org/10.2991/978-94-6239-830-6_25

directly reflect short-term supply and demand, liquidity status, and trading pressure. Therefore, how to extract effective features from LOB data and use them to predict the direction of the future short-term midpoint has become an important issue in high-frequency trading and financial machine learning research.

Existing research has advanced the task of predicting LOB mid-prices by focusing on dataset construction, machine learning modeling, and deep learning prediction. Ntakaris et al. constructed the FI-2010 public benchmark dataset, which is derived from the order book time series of 5 stocks in the NASDAQ Nordic market for 10 consecutive trading days. It contains approximately 4 million samples and provides an anchored cross-validation experimental protocol based on the trading day order, providing a basis for reproducible experimental comparisons between different models [1]. In terms of traditional machine learning methods, Kercheval and Zhang used a multi-class SVM to predict mid-price movements and price-spread crossings from LOB features, showing that price, volume, and depth information can support short-term market-state prediction [2]. In terms of deep learning methods, Tsantekidis et al. built a CNN model based on more than 4 million order book events and compared it with methods such as MLP and SVM. The results showed that CNN is more suitable for extracting price change features from large-scale high-frequency order book sequences [3]. Furthermore, Zhang et al. proposed the DeepLOB model, which captures the spatial structure of the order book through convolutional structures and uses LSTM to model time dependencies. Under the 7-day training and 3-day testing settings of the FI-2010 dataset, DeepLOB achieved F1 scores of 83.40%, 72.82%, and 80.35% in prediction windows of k=10, k=20, and k=50, respectively, which are better than many existing models [4]. In addition, Nousi et al. systematically compared the performance of various machine learning methods in the mid-price direction prediction task based on the FI-2010 dataset, providing a supplementary reference for the subsequent evaluation of LOB prediction models [5]. From the perspective of market microstructure, Cont et al. studied the impact of order book events on price changes, indicating that pending orders, cancellations, and transactions in the order book are closely related to the short-term price formation process [6].

In recent years, LOB forecasting research has focused more on model robustness, feature representation, and trading operability. Lucchese et al. pointed out that the order book representation and forecast window affect short-term return forecasting performance [7]. Prata et al. incorporated profit analysis into the LOB deep learning model benchmark, indicating that model evaluation should focus on trading performance [8]. Briola et al. further pointed out that higher predictive ability does not necessarily correspond to actionable trading signals [9]. In addition, studies such as signature transform, TLOB and LiT have expanded the short-term trend prediction method of LOB from the perspective of feature representation and Transformer architecture [10–12]. Existing studies have shown that machine learning and deep learning models can achieve good statistical performance in predicting LOB midpoint movements. However, high prediction accuracy does not necessarily mean that the resulting signals are still useful in actual trading, especially when transaction costs are considered. To examine this issue, the present study uses the preprocessed FI-2010 dataset and evaluates the models from both prediction and backtesting perspectives. Logistic Regression,

XGBoost, LSTM, and CNN-LSTM are compared under different prediction horizons. Their performance in identifying upward, unchanged, and downward price movements is measured using Accuracy, Macro-F1, and Weighted-F1. Each predicted class is assigned a trading position: long for an upward movement, neutral for no movement, and short for a downward movement. The resulting signals are then tested under a simplified cost setting, and the models are compared in terms of a cumulative net return proxy and turnover.

In addition to classification performance, cumulative net return proxies and turnover are considered to assess whether stronger predictive performance translates into practical economic value after transaction costs.

2. Experimental Design

2.1 Data Sources and Forecasting Task

The experiments use the preprocessed FI-2010 dataset [13], containing LOB data for five NASDAQ Nordic stocks over ten consecutive trading days. The data used in this study has been organized into a "sample × variable" format, where the first 144 dimensions of variables serve as the model input features, and the last 5 dimensions are the mid-price labels under different prediction horizons. During the experiments, this study only uses the first 144 dimensions of features as the model input, and does not use the label column or any future window information as input variables to avoid the leakage of future information.

It should be noted that this study uses preprocessed FI-2010 feature data, rather than unstandardized raw order book price data. Therefore, price-related variables are not directly equivalent to actual tradable prices. This study does not calculate the actual bid-ask spread cost based on these variables, but instead uses a fixed transaction cost coefficient as a proxy variable for transaction frictions such as fees, spreads, and slippage.

The existing FI-2010 training file was split contiguously in its original row order into a training subset of 310,628 samples and a validation subset of 51,772 samples, while the separate test file containing 31,937 samples was reserved for final evaluation. For LSTM and CNN-LSTM, the checkpoint with the highest validation Macro-F1 was selected, and the independent test set was evaluated once after model selection was completed. Logistic Regression and XGBoost used fixed hyperparameters and were fitted on the combined training and validation subsets before evaluation on the same independent test set. The evaluated prediction horizons were h∈ {10,20,30,50,100}.

The core task of this study is to predict the direction of the midpoint price change several ticks in the future, based on the characteristics of the current and historical order books. The midpoint price is typically defined as the average of the optimal bid price and the optimal ask price.

$$m_t = \frac{a_t^{(1)} + b_t^{(1)}}{2} \tag{1}$$

where m_t denotes the mid-price at time t, $a_t^{(1)}$ denotes the best ask price at time t, and $b_t^{(1)}$ denotes the best bid price at time t.

The directional labels provided by FI-2010 are used directly without reconstruction and consist of three classes: upward, stationary, and downward movements. Given the limit order book feature input X_t, the objective of the model is to predict the directional label $\hat{y}_{t,h}$ over a future prediction horizon h.

2.2 Model Settings

This study compares Logistic Regression, XGBoost, LSTM, and CNN-LSTM. Logistic Regression serves as a linear baseline, whereas XGBoost captures nonlinear feature interactions. LSTM models temporal dependencies in LOB sequences, while CNN-LSTM further extracts local sequential patterns through one-dimensional convolution. The main model configurations and hyperparameters are summarized in Table 1.

Table 1. Main Configurations and Hyperparameters of the Compared Models

Model	Model Configuration	Main Hyperparameters
Logistic Regression	StandardScaler + Logistic Regression	L2 regularization; C=1.0; LBFGS solver; balanced class weights; maximum iterations = 1000
XGBoost	Gradient-boosted decision trees	200 estimators; maximum depth = 5; learning rate = 0.05; subsample = 0.9; column sampling = 0.9
LSTM	One-layer LSTM + fully connected layer	Input window = 10; hidden size = 64; dropout = 0.2;
CNN-LSTM	1D-CNN + ReLU + one-layer LSTM + fully connected output layer	64 convolution channels; kernel size = 3; padding = 1; input window = 10; hidden size = 64; dropout = 0.2

Note: Both LSTM and CNN-LSTM used Adam with cross-entropy loss, a learning rate of 0.001, a batch size of 256, and 20 training epochs. A fixed random seed of 42 was used for all models. XGBoost used the multi:softprob objective, multiclass logarithmic loss, and the histogram-based tree method.

2.3 Classification Evaluation Indicators

Classification performance is evaluated using Accuracy, Macro-F1, and Weighted-F1, with Macro-F1 emphasized because of potential class imbalance.

2.4 Trading Signal Construction and Cost-Constrained Backtesting

After completing the classification predictions, this study further transforms the model prediction results into trading signals to test the relative economic effectiveness of the prediction results under transaction cost constraints. Specifically, an upward prediction is mapped to a long position, a downward prediction is mapped to a short position, and

an unchanged prediction is mapped to a neutral position. Trading positions are defined as follows:

$$p_t = \begin{cases} 1, & \hat{y}_{t,h} = Up, \\ 0, & \hat{y}_{t,h} = Stationary, \\ -1, & \hat{y}_{t,h} = Down. \end{cases} \tag{2}$$

where p_t denotes the trading position at time t, and $\hat{y}_{t,h}$ denotes the model's prediction of the price direction h ticks ahead. The labels Up, Stationary, and Down indicate upward movement, no movement, and downward movement, respectively.

Since this study uses preprocessed price features, it is not possible to directly calculate the rate of return using real market prices. Therefore, this study constructs a proxy variable for the mid-price based on price-related features:

$$\widetilde{m}_t = \frac{x_{t,0} + x_{t,2}}{2} \tag{3}$$

where $x_{t,0}$ and $x_{t,2}$ are treated as normalized ask-side and bid-side price proxies in the processed FI-2010 representation. Their average is used to construct the mid-price proxy $\widetilde{m}_t$.

Based on this proxy variable, the gross return proxy is defined as:

$$R_t^{gross} = p_t \cdot \frac{\widetilde{m}_{t+h} - \widetilde{m}_t}{\widetilde{m}_t} \tag{4}$$

where R_t^{gross} denotes the return proxy before accounting for transaction costs, and $\widetilde{m}_{t+h}$ denotes the mid-price proxy h ticks ahead.

Furthermore, a fixed transaction cost coefficient c is introduced to account for trading frictions, including transaction fees, bid–ask spreads, and slippage. The magnitude of the position change is defined as:

$$Turnover_t = \mid p_t - p_{t-1} \mid \tag{5}$$

where $Turnover_t$ measures the absolute position change between consecutive signals.

To facilitate comparisons across models with slightly different numbers of valid trading signals, the average turnover is further defined as:

$$\overline{\text{Turnover}} = \frac{1}{N_h} \sum_{t=1}^{N_h} \text{Turnover}_t \tag{6}$$

where $\overline{\text{Turnover}}$ denotes the average turnover over the test period, and N_h denotes the number of valid backtesting observations for a given model at prediction horizon h.

The cost-adjusted net return proxy is defined as:

$$R_t^{net} = R_t^{gross} - c \cdot Turnover_t \tag{7}$$

where R_t^{net} denotes the net return proxy after deducting transaction costs, and c denotes the fixed transaction cost coefficient.

The cumulative net return proxy is defined as:

$$CR = \sum_{t=1}^{N_h} R_t^{net} \tag{8}$$

where CR denotes the cumulative net return proxy over the test period, and N_h denotes the effective number of backtesting observations after model alignment and horizon truncation.

This study evaluates the trading signals under several fixed transaction cost coefficients. Specifically, the transaction cost coefficient was set to c∈ {0,0.0001,0.001}. The comparison focuses on the cumulative net return proxy and average turnover, which are used to examine the relative economic performance and trading activity of different model-generated signals. It is important to emphasize that because the backtesting in this study is based on the mid-price proxy variable rather than the original market price, the backtesting results are not interpreted as the actual market executable return rate, but rather as a comparative indicator of the relative economic efficiency of different model prediction signals under cost constraints.

3. Experimental Results

First, from a classification prediction perspective, the paper compares the performance of Logistic Regression, XGBoost, LSTM, and CNN-LSTM under different prediction windows. Evaluation metrics include Accuracy, Macro-F1, and Weighted-F1 (Table 2). Since the FI-2010 mid-price labels may exhibit class imbalance, using only Accuracy may overestimate the model's predictive ability for most classes. Therefore, this study focuses on combining Macro-F1 to analyze the overall classification performance of the models.

Table 2. Classification Performance at h=20

Model	Accuracy	Macro-F1	Weighted-F1
LR	0.551	0.442	0.535
XGBoost	0.732	0.631	0.704
LSTM	0.771	0.722	0.767
CNN-LSTM	0.799	0.751	0.793

Note: LR stands for Logistic Regression. This study conducted experiments with multiple prediction horizons; Table 2 shows the classification results using h=20 as the representative prediction window.

Among the four models, CNN-LSTM achieves the strongest classification performance. Its Accuracy, Macro-F1, and Weighted-F1 scores are 0.799, 0.751, and 0.793, respectively. The results show that temporal information plays an important role in describing changes in limit order book features. By comparison, Logistic Regression obtains a Macro-F1 score of only 0.442 and performs less effectively across the three directional classes. This may be because a linear decision boundary cannot adequately represent the complex relationships between LOB features and midpoint movements. XGBoost performs better than Logistic Regression, but its scores are still lower than

those of LSTM and CNN-LSTM. Although XGBoost can capture nonlinear feature interactions, the model does not explicitly account for the sequential dependence of order book observations, which may limit its performance in this task.

The economic effectiveness of the predicted signals is further examined through cost-constrained backtesting.

Table 3 reports the cumulative net return proxies under three transaction cost settings and the average turnover of the four models at h=20 and h=100.

Table 3. Cumulative Net Return Proxies and Average Turnover

Horizon	Model	R(c=0)	R(c=0.0001)	R(c=0.001)	Average Turnover
20	LR	-17.137	-18.211	-27.873	0.336
20	XGBoost	-5.809	-7.061	-18.331	0.392
20	LSTM	17.648	16.492	6.090	0.362
20	CNN-LSTM	7.616	6.394	-4.606	0.383
100	LR	-34.908	-35.725	-43.076	0.256
100	XGBoost	-168.579	-170.043	-183.225	0.459
100	LSTM	-46.063	-46.859	-54.022	0.249
100	CNN-LSTM	-94.667	-95.463	-102.622	0.249

Note: R(c) denotes the cumulative net return proxy under transaction cost coefficient c. Average Turnover denotes the mean absolute position change over the effective backtesting observations.

As shown in Table 3, the cumulative net return proxy of all models decreased as the transaction cost coefficient increased from c=0 to c=0.001, indicating that transaction costs weaken the economic effectiveness of high-frequency prediction signals. Average turnover also varies across models and prediction horizons. It should be noted that the backtesting in this study is based on the mid-price proxy variable constructed from preprocessed features; therefore, the values in the table are mainly used for relative comparisons between models and do not represent the actual market executable return rate.

At h = 20 and c = 0.001, LSTM is the only model that retains a positive cumulative net return proxy, with a value of 6.090. CNN-LSTM, although achieving the strongest classification performance, records a negative value of -4.606 after the cost coefficient is applied. The average turnover values of LSTM and CNN-LSTM are 0.362 and 0.383, respectively. Logistic Regression and XGBoost also remain below zero under the same setting. This contrast shows that the model ranking based on classification metrics is not consistent with the ranking obtained from cost-adjusted trading performance.

The pattern becomes more pronounced when the prediction horizon is extended to h = 100. At c = 0.001, all four models produce negative cumulative net return proxies. XGBoost records the lowest cumulative net return proxy of -183.225 and the highest average turnover of 0.459. LSTM and CNN-LSTM have nearly identical average turnover values, both rounded to 0.249, while their cumulative net return proxies are -54.022 and -102.622, respectively. The longer-horizon results therefore provide additional evidence that acceptable classification performance alone is not sufficient to ensure positive trading outcomes.

To examine this discrepancy more directly, Fig. 1 compares Macro-F1 with the corresponding cumulative net return proxy at c = 0.001 across different models and prediction horizons. The comparison is intended to assess whether stronger classification performance is accompanied by better cost-adjusted economic performance.

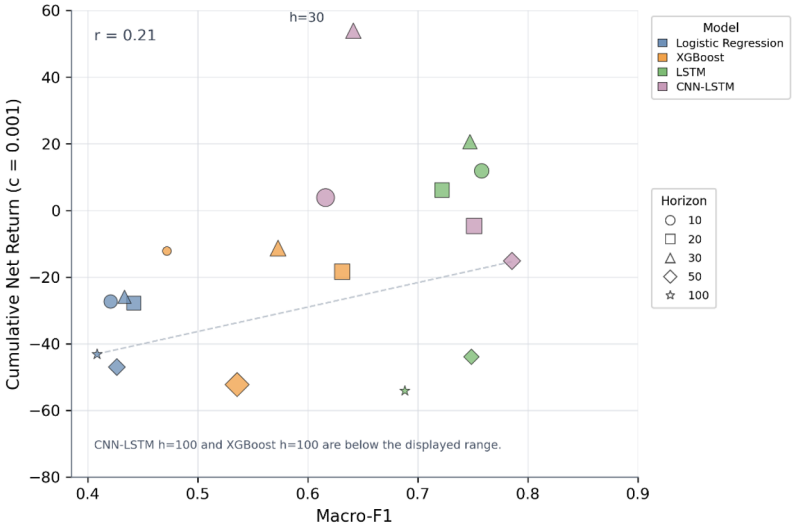


Fig. 1. Relationship between Macro-F1 and the Cumulative Net Return Proxy (Photo/Picture credit: Original).

Note: The horizontal axis represents Macro-F1, and the vertical axis represents the cumulative net return proxy at $c = 0.001$. Color identifies the model, marker shape represents the prediction horizon, and marker size indicates turnover. The dashed line denotes the fitted linear trend.

Fig. 1 compares Macro-F1 with CR across the different model–horizon combinations. Since CR is calculated from a mid-price proxy rather than actual execution prices, it should be interpreted only as a relative measure of economic performance, not as a realized monetary return.

The estimated correlation between Macro-F1 and CR is approximately 0.21, indicating only a weak positive association in the current experiments. Models with stronger classification results do not necessarily retain higher CR values once trading costs are taken into account.

4. Analysis

At the representative setting, the deep learning models outperform the two traditional baselines in classification. Logistic Regression is constrained by its linear decision structure, while XGBoost captures nonlinear feature interactions but does not explicitly model temporal dependence. LSTM and CNN-LSTM exploit sequential information in LOB observations, consistent with DeepLOB [4].

However, classification prediction performance is not directly equivalent to economic efficiency under transaction cost constraints. Indicators such as Accuracy and Macro-F1 mainly measure the model's ability to identify directional labels, but trading performance is also affected by trading frequency, turnover, and transaction costs. At h=20, LSTM has a slightly lower average turnover than CNN-LSTM, at 0.362 and 0.383, respectively. However, LSTM also achieves a higher cumulative return proxy before transaction costs are deducted, at 17.648 compared with 7.616 for CNN-LSTM, indicating that the difference in cost-adjusted performance cannot be explained by turnover alone. At h=100, LSTM and CNN-LSTM have nearly identical average turnover values, both rounded to 0.249, but substantially different cumulative net return proxies, further showing that trading activity and return generation capture different aspects of economic performance. These results show that transaction costs can weaken the economic value of high-frequency prediction signals, particularly when frequent position changes increase turnover. Consistent with Briola et al. [9], classification performance should therefore be evaluated together with cumulative net return proxies and turnover.

The results should be interpreted in light of the preprocessed FI-2010 data and simplified backtesting setting. Since the price-related features are preprocessed rather than raw executable prices, the cumulative net return proxy is intended only for relative model comparison; its magnitude may depend on the preprocessing scale. In addition, adjacent -step evaluation intervals overlap, so CR should be interpreted as an aggregate signal-level performance proxy rather than the realized return of a self-financing trading strategy. The fixed transaction cost coefficient does not capture dynamic spreads, order-book depth, execution probability, latency, or market shocks [14,15]. Future work could incorporate original order-book prices and more realistic execution and cost mechanisms [16].

5. Conclusion

This study examines whether stronger classification performance in LOB mid-price prediction also leads to better trading outcomes when transaction costs are considered. Among the four models, CNN-LSTM achieves the strongest classification results, while the backtesting results give a different ranking. At h=20 and c=0.001, LSTM retains a positive cumulative net return proxy of 6.090, whereas CNN-LSTM falls to -4.606. When the prediction horizon increases to h=100, all four models produce negative cost-adjusted return proxies. These results show that improvements in directional prediction do not necessarily translate into stronger economic performance.

The weak correlation between Macro-F1 and the cumulative net return proxy ($r \approx 0.21$) further highlights this difference. Trading outcomes depend not only on prediction quality, but also on the return generated by the signals, their turnover, and the resulting transaction costs. Classification performance, cumulative net return proxy, and turnover therefore provide complementary views of model effectiveness and should be considered together when evaluating LOB prediction models. Since the backtest relies on preprocessed FI-2010 price features and a fixed transaction cost coefficient, the reported return values should be interpreted as relative economic indicators rather than

executable market returns. More realistic price and execution data would allow this relationship to be tested under practical trading conditions.

References

1. Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. Journal of Forecasting 37(8), 852–866 (2018)
2. Kercheval, A.N., Zhang, Y.: Modelling high-frequency limit order book dynamics with support vector machines. Quantitative Finance 15(8), 1315–1329 (2015)
3. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Forecasting stock prices from the limit order book using convolutional neural networks. In: 2017 IEEE 19th Conference on Business Informatics, pp. 7–12. IEEE (2017)
4. Zhang, Z., Zohren, S., Roberts, S.: DeepLOB: Deep convolutional neural networks for limit order books. IEEE Transactions on Signal Processing 67(11), 3001–3012 (2019)
5. Nousi, P., Tsantekidis, A., Passalis, N., Ntakaris, A., Kanniainen, J., Tefas, A., Gabbouj, M., Iosifidis, A.: Machine learning for forecasting mid-price movements using limit order book data. IEEE Access 7, 64722–64736 (2019)
6. Cont, R., Kukanov, A., Stoikov, S.: The price impact of order book events. Journal of Financial Econometrics 12(1), 47–88 (2014)
7. Lucchese, L., Pakkanen, M.S., Veraart, A.E.D.: The short-term predictability of returns in order book markets: A deep learning perspective. International Journal of Forecasting 40(4), 1587–1621 (2024)
8. Prata, M., Masi, G., Berti, L., Arrigoni, V., Coletta, A., Cannistraci, I., Vyetrenko, S., Velardi, P., Bartolini, N.: LOB-based deep learning models for stock price trend prediction: A benchmark study. Artificial Intelligence Review 57, 116 (2024)
9. Briola, A., Bartolucci, S., Aste, T.: Deep limit order book forecasting: A microstructural guide. Quantitative Finance 25(7), 1101–1131 (2025)
10. Sidogi, T., Mongwe, W.T., Mbuvha, R., Olukanmi, P., Marwala, T.: A signature transform of limit order book data for stock price prediction. IEEE Access 11, 70598–70609 (2023)
11. Berti, L., Kasneci, G.: TLOB: A novel transformer model with dual attention for price trend prediction with limit order book data. arXiv preprint arXiv:2502.15757 (2025)
12. Xiao, Y., Ventre, C., Wang, Y., Li, H., Huan, Y., Liu, B.: LiT: Limit order book transformer. Frontiers in Artificial Intelligence 8, 1616485 (2025)
13. Chen, F.: FI2010. Kaggle (2019), https://www.kaggle.com/datasets/freemanone/fi2010, last accessed 2026/06/26
14. Arroyo, Á., Cartea, Á., Moreno-Pino, F., Zohren, S.: Deep attentive survival analysis in limit order books: Estimating fill probabilities with convolutional-transformers. Quantitative Finance 24(1), 35–57 (2024)
15. Nagy, P., Calliess, J.-P., Zohren, S.: Asynchronous deep double dueling Q-learning for trading-signal execution in limit order book markets. Frontiers in Artificial Intelligence 6, 1151003 (2023)
16. 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. Machine Learning 113, 2795–2839 (2024)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

