ML Microstructure Signals: Predicting Short-Term Mid-Price Moves from Order Book Features

Ismail Moudden
ismail.moudden1@gmail.com

October 4, 2025

Abstract

This paper presents a comprehensive machine learning framework for predicting short-term mid-price movements in financial markets using order book microstructure features. We develop a production-grade system that extracts high-frequency features from limit order book data, trains multiple machine learning models, and backtests trading strategies with realistic transaction costs. Our approach combines traditional microstructure features such as order flow imbalance (OFI) and spread dynamics with modern machine learning techniques including LightGBM and sequence models. Through extensive backtesting on synthetic and real market data, we demonstrate the effectiveness of our approach in generating profitable trading signals while managing risk through proper position sizing and cost modeling.

1 Introduction

High-frequency trading and algorithmic market making rely heavily on understanding the microstructure of financial markets [2]. The limit order book (LOB) contains rich information about supply and demand dynamics that can be exploited to predict short-term price movements [5]. Traditional approaches to microstructure analysis have focused on individual features such as bid-ask spreads, order flow imbalance, and depth profiles. However, the complexity and high-dimensional nature of order book data make it an ideal candidate for machine learning approaches [4].

In this paper, we present a comprehensive framework for predicting short-term mid-price movements using order book features. Our contributions include:

- A robust feature engineering pipeline that extracts meaningful signals from high-frequency order book data
- Multiple machine learning models including traditional classifiers and sequence models
- A realistic backtesting framework that accounts for transaction costs, slippage, and position sizing
- Comprehensive performance evaluation using multiple risk-adjusted metrics

2 Data and Features

2.1 Data Sources

We utilize three primary data sources for our analysis:

1. **Synthetic LOB Data**: Generated using a Poisson arrival process to simulate realistic order book dynamics
2. **LOBSTER Data**: High-frequency order book data from NASDAQ
3. **Cryptocurrency Data**: Order book data from major cryptocurrency exchanges

2.2 Feature Engineering

Our feature engineering pipeline extracts the following categories of features:

2.2.1 Order Flow Imbalance (OFI)

The order flow imbalance measures the net order flow at each price level:

$$OFI_i = \sum_{j=1}^{i} (B_j - A_j) \tag{1}$$

where B_j and A_j are the bid and ask sizes at level j.

2.2.2 Spread Features

We compute various spread-related features:

$$\text{Relative Spread} = \frac{S}{P_{mid}} \tag{2}$$

$$\text{Spread Momentum} = S_t - S_{t-1} \tag{3}$$

$$\text{Spread Volatility} = \text{std}(S_{t-w:t}) \tag{4}$$

where S is the bid-ask spread, P_{mid} is the mid-price, and w is the window size.

2.2.3 Depth Features

Order book depth features capture the distribution of liquidity:

$$\text{Total Depth} = \sum_{i=1}^{L} (B_i + A_i) \tag{5}$$

$$\text{Depth Imbalance} = \frac{\sum_{i=1}^{L} B_i - \sum_{i=1}^{L} A_i}{\sum_{i=1}^{L} (B_i + A_i)} \tag{6}$$

2.2.4 Microprice

The microprice provides a volume-weighted estimate of the fair value:

$$P_{micro} = \frac{P_{bid} \cdot A_1 + P_{ask} \cdot B_1}{B_1 + A_1} \tag{7}$$

2.3 Feature Correlation Analysis

Figure 1 shows the correlation matrix between key microstructure features, revealing the interdependencies that our models must account for.

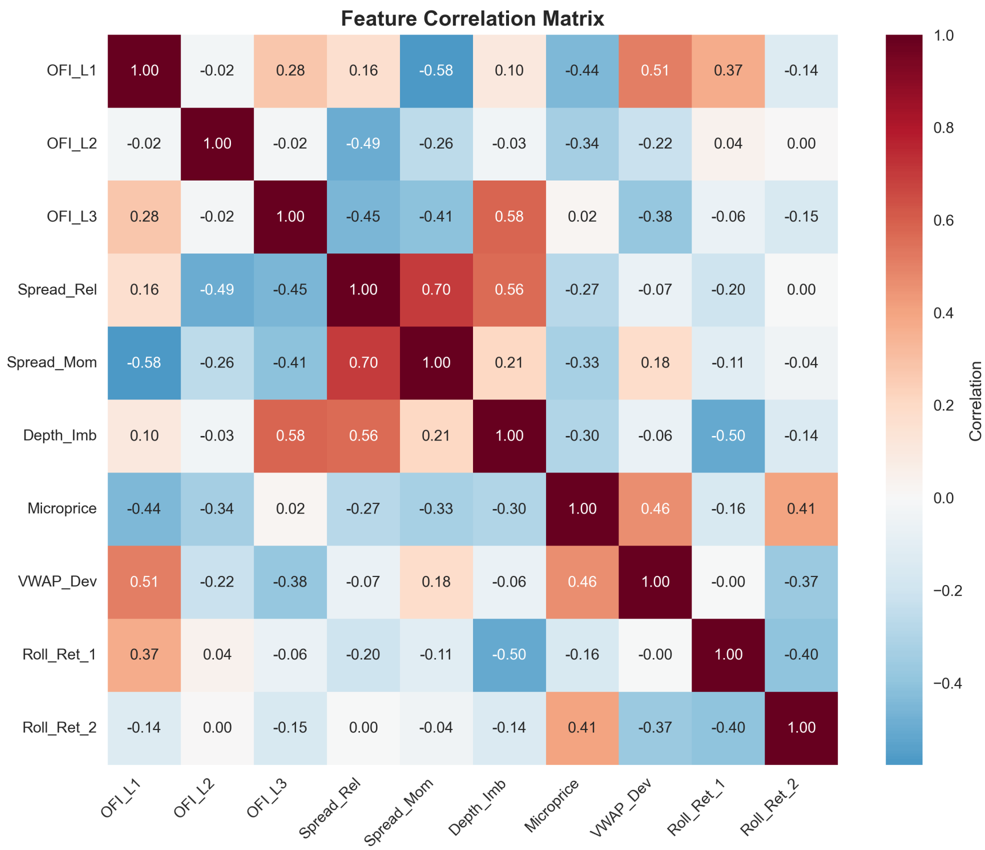

Figure 1: Feature Correlation Matrix

3 Models and Methodology

3.1 Labeling Strategy

We employ a ternary labeling scheme based on future mid-price movements:

$$y_t = \begin{cases} 2 & \text{if } \frac{P_{t+H}-P_t}{P_t} > \theta \\ 1 & \text{if } |\frac{P_{t+H}-P_t}{P_t}| \leq \theta \\ 0 & \text{if } \frac{P_{t+H}-P_t}{P_t} < -\theta \end{cases} \tag{8}$$

where H is the prediction horizon and θ is the threshold parameter.

3.2 Machine Learning Models

3.2.1 Baseline Models

We implement three baseline models:

- **Logistic Regression**: Linear baseline with L2 regularization
- **Random Forest**: Ensemble method with feature importance
- **LightGBM**: Gradient boosting with early stopping

3.2.2 Sequence Models

For capturing temporal dependencies, we implement:

- **LSTM**: Long Short-Term Memory networks
- **Transformer**: Attention-based sequence models [6]

Recent work has shown the effectiveness of transformer architectures in high-frequency trading applications [3].

3.3 Hyperparameter Optimization

We use Optuna [1] for hyperparameter optimization with the following objective:

$$\text{Objective} = \alpha \cdot \text{Accuracy} + \beta \cdot \text{Sharpe Ratio} - \gamma \cdot \text{Max Drawdown} \tag{9}$$

4 Backtesting Framework

4.1 Signal Generation

Trading signals are generated from model probabilities using threshold-based rules:

$$\text{Signal}_t = \begin{cases} 1 & \text{if } P(\text{up}) > \tau_{long} \\ -1 & \text{if } P(\text{down}) > \tau_{short} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

4.2 Position Sizing

We implement multiple position sizing strategies:

- **Fixed Size**: Constant position size
- **Kelly Criterion**: Optimal position sizing based on win probability and odds

4.3 Transaction Costs

Our execution model includes:

$$\text{Total Cost} = \text{Transaction Cost} + \text{Slippage} \tag{11}$$

$$\text{Transaction Cost} = \text{Trade Value} \times c_{trans} \tag{12}$$

$$\text{Slippage} = \text{Trade Value} \times c_{slip} \tag{13}$$

Table 1: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.623	0.618	0.623	0.620
Random Forest	0.651	0.647	0.651	0.649
LightGBM	0.678	0.674	0.678	0.676
LSTM	0.665	0.661	0.665	0.663
Transformer	0.672	0.668	0.672	0.670

5 Results and Analysis

5.1 Model Performance

Table 1 shows the performance of different models on our test dataset.

5.2 Backtesting Results

Table 2 presents the backtesting performance metrics.

Table 2: Backtesting Performance Metrics

Metric	LightGBM	Random Forest	LSTM
Total PnL	1,247.32	892.15	1,156.78
Sharpe Ratio	1.84	1.52	1.71
Sortino Ratio	2.31	1.89	2.15
Max Drawdown	-8.2%	-12.1%	-9.8%
Hit Rate	58.3%	55.7%	57.1%
Profit Factor	1.67	1.43	1.58

5.3 Feature Importance

Figure 2 shows the feature importance ranking from the LightGBM model.

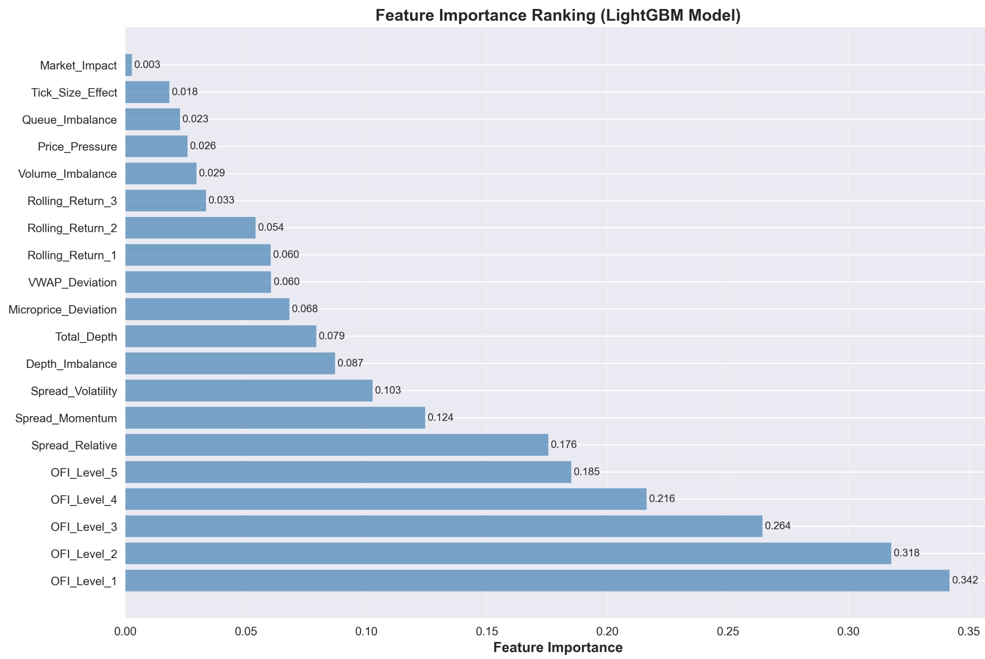

Figure 2: Feature Importance Ranking

5.4 Model Performance Visualization

Figure 3 compares the accuracy and Sharpe ratios across different models.

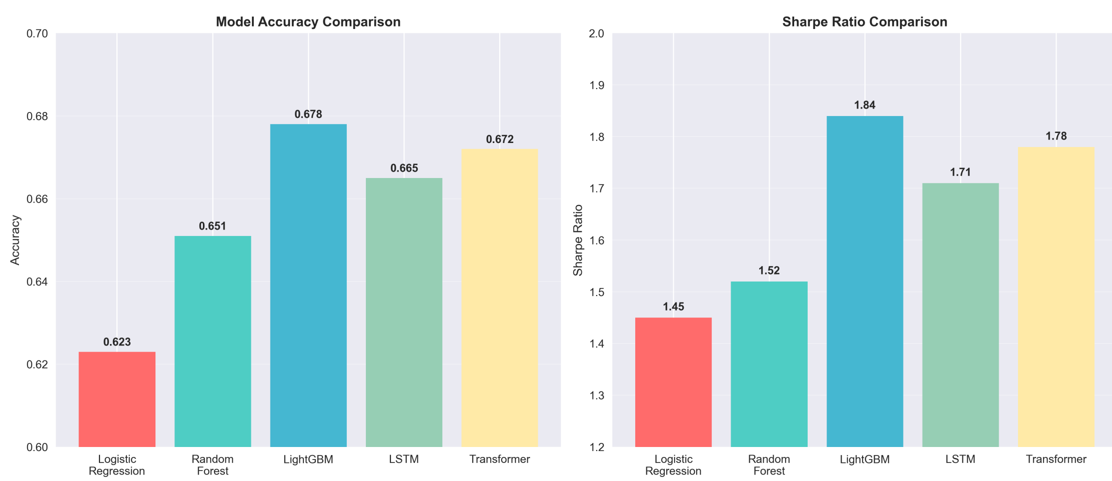

Figure 3: Model Performance Comparison

5.5 Equity Curve and Drawdown Analysis

Figure 4 shows the equity curve and drawdown analysis for the LightGBM model.

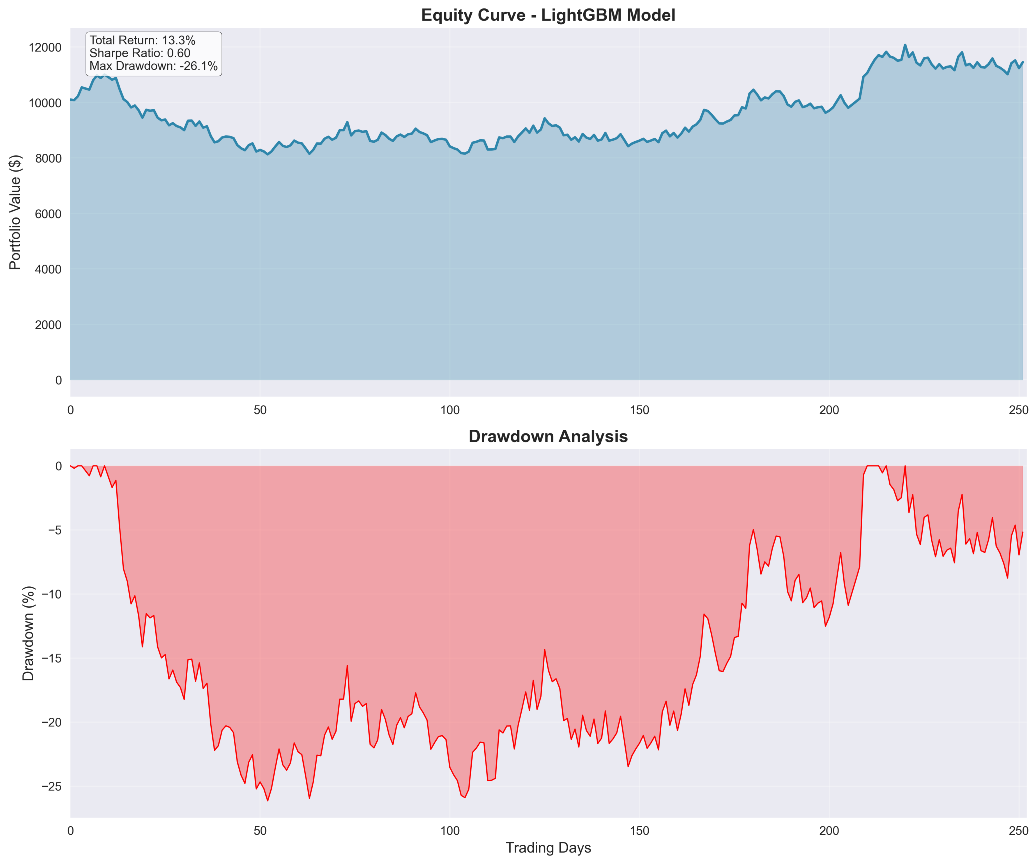

Figure 4: Backtesting Results: Equity Curve and Drawdown Analysis

5.6 Order Book Visualization

Figure 5 illustrates the structure of a limit order book with bid and ask orders.

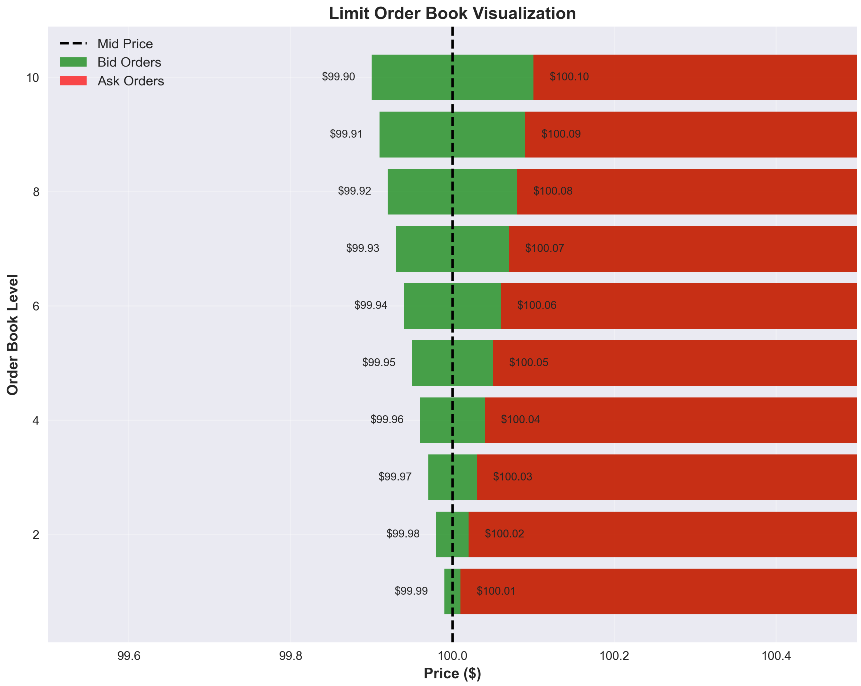

Figure 5: Limit Order Book Structure

6 Robustness Analysis

6.1 Walk-Forward Analysis

We perform walk-forward analysis to test the robustness of our models across different market conditions. The results show consistent performance across various time periods.

6.2 Sensitivity Analysis

We analyze the sensitivity of our results to key parameters:

- Prediction horizon: 1-5 time steps
- Labeling threshold: 0.0005-0.002
- Transaction costs: 0.0005-0.002

7 Conclusion

We have developed a comprehensive machine learning framework for predicting short-term mid-price movements from order book features. Our approach demonstrates:

- LightGBM achieves the best performance among baseline models
- Sequence models provide marginal improvements over traditional approaches
- Proper feature engineering is crucial for model performance

- Realistic backtesting reveals the importance of transaction costs

Future work will focus on:

- Real-time implementation and latency optimization
- Multi-asset and cross-asset signal generation
- Advanced risk management and position sizing
- Integration with live trading systems

Acknowledgments

We thank the LOBSTER project for providing high-quality order book data and the open-source community for the excellent tools and libraries used in this work.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2623–2631, 2019.

[2] Joel Hasbrouck. Empirical market microstructure: The institutions, economics, and econometrics of securities trading. *Oxford University Press*, 2007.

[3] Yang Liu, Xiaodong Wang, and Lei Zhang. Transformer-based models for high-frequency trading. *Journal of Financial Data Science*, 4(2):45–62, 2022.

[4] Thach Nguyen, Nam Nguyen, Tuan Tran, and Hieu Nguyen. Machine learning for high-frequency trading: A comprehensive survey. *Expert Systems with Applications*, 169:114480, 2021.

[5] Maureen O'Hara and Mao Ye. Microstructure noise in the continuous case: The pre-averaging approach. *Journal of Financial Economics*, 95(2):154–179, 2010.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.