ORIE 5259: Market Microstructure & Algorithmic Trading

High-Frequency Execution Strategy Based on Market Microstructure Features

Team Members: Kexin Deng(kd537), Yichen Gao(yg635), Dora Xu(dx33), Michael Cao(yc849)

Cornell Financial Engineering

Introduction

Objective: Develop a high-frequency trading algorithm that reacts to real-time order book dynamics and improve trade execution timing using short-term microstructure signals

Strategy (executes 1 buy & 1 sell per minute) uses 6 key features:

Bid-Ask Spread	Short-term Momentum
Order Flow (10s)	Volatility-to-Spread Ratio
Order Book Imbalance	Time Pressure

Score-based strategy: computed via weighted linear combination to guide buy/sell decisions

- Parameters tuned to minimize average price difference (Buy Price–Sell Price) per minute
- Results compared against TWAP benchmark: Average spread at last event of each minute

Variables

Variable Construction

1. **Momentum Factors:** Capture mid-price changes over different short-term horizons (every 5-10 updates)

$$\text{Momentum\_1} = \text{Mid\_price}_t - \text{Mid\_price}_{t-1}$$

2. **Volatility-to-Spread Ratio:** Evaluates short-term price volatility relative to the market spread

$$\text{Vol\_Spread\_Ratio} = \frac{\sigma(\text{Mid Price})}{\text{Ask Price} - \text{Bid Price}}$$

3. **Order Book Imbalance:** Represents the direction of price movement based on bid and ask volume

$$\text{OBI} = \frac{\text{Bid Volume - Ask Volume}}{\text{Bid Volume + Ask Volume}}$$

4. **Order Flow:** Measures net aggressive trading at the best bid and ask prices over a time window of 10

$$\text{OFI}_{10}(t) = \sum_{\tau=t-9}^{t} (\text{Bid Volume} - \text{Ask Volume})$$

5. **Bid-Ask Spread:** Represents the difference between the best ask price and the best bid price

$$\text{Spread} = \text{Ask Price - Bid Price}$$

6. **Time Pressure:** Represents the fraction of the current minute that has elapsed

$$\text{Time\_Pressure} = \frac{\text{Seconds\_Elapsed\_In\_Minute}}{60}$$

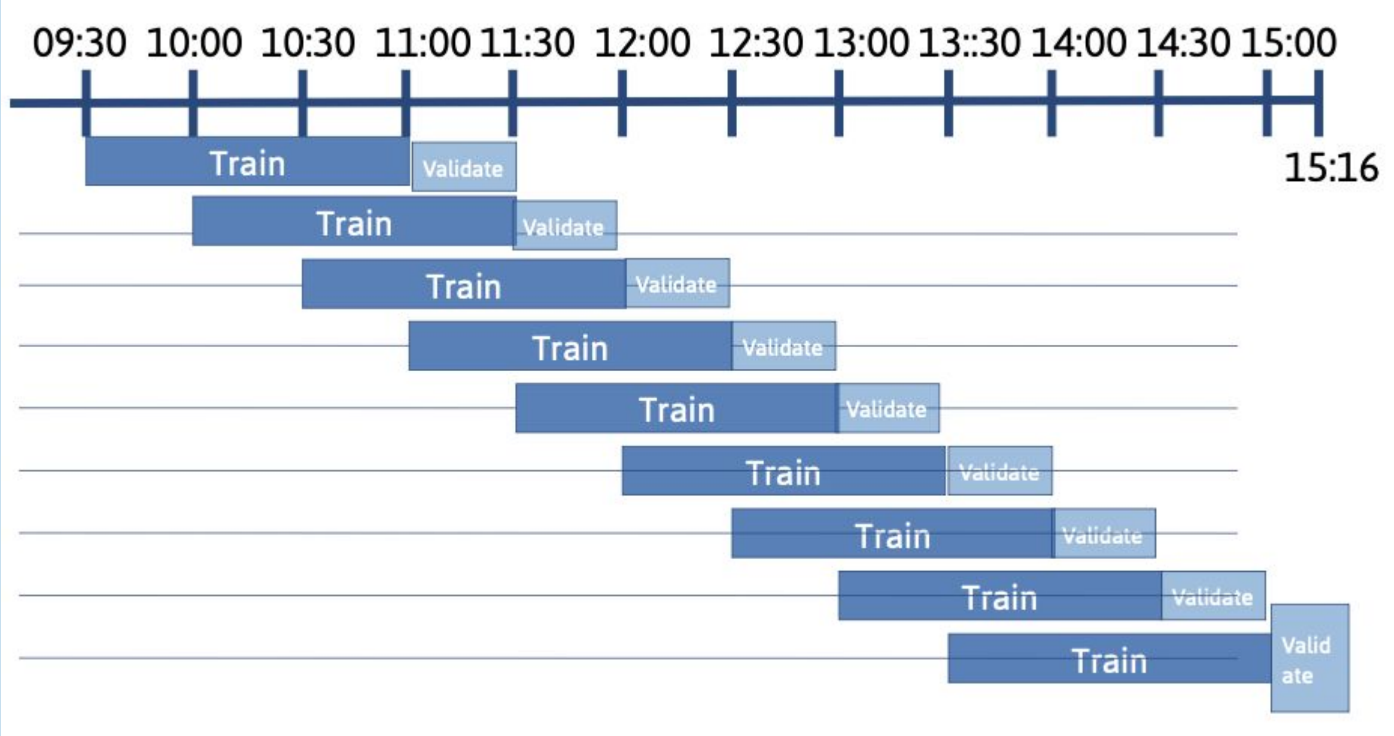
Strategic Logic - Pipeline

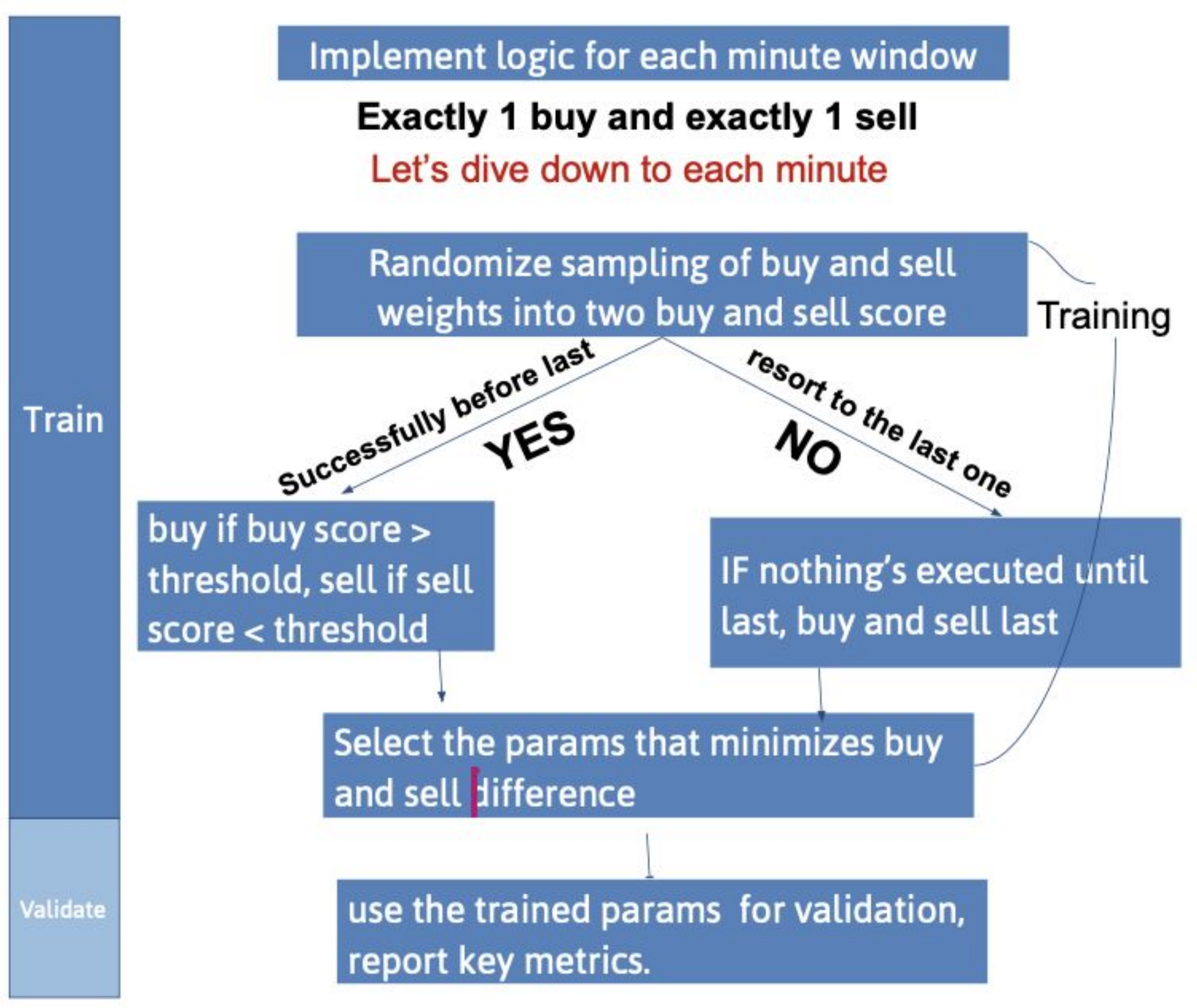
Pipeline:

- Factor Generation
- Score Calculation
- Trade Execution
- Benchmark & Evaluation

Approach:

- Initial Approach: Before 12 VS After 12
- Improved: Last 90 mins VS Next 30 mins
- Backtest: Last 30 mins VS Next 5 mins





- Theoretical motivation: Minimize price execution spread (buy-sell difference) – **proxy for trading efficiency.**

Strategic Logic - Score

Algorithm Implementation

Two independent scores

$$\begin{aligned}\text{Buy\_Score} = {} & w_{1,\text{buy}} \times \text{Momentum1} + w_{2,\text{buy}} \times \text{Momentum3} + w_{3,\text{buy}} \times \text{Momentum5} \\ & + w_{4,\text{buy}} \times \text{Volatility-to-Spread Ratio} + w_{5,\text{buy}} \times \text{Order Flow (10s)} \\ & + w_{6,\text{buy}} \times \text{Imbalance} + 0.5 \times \text{Time Pressure},\end{aligned}$$

$$\begin{aligned}\text{Sell\_Score} = {} & w_{1,\text{sell}} \times \text{Momentum1} + w_{2,\text{sell}} \times \text{Momentum3} + w_{3,\text{sell}} \times \text{Momentum5} \\ & + w_{4,\text{sell}} \times \text{Volatility-to-Spread Ratio} + w_{5,\text{sell}} \times \text{Order Flow (10s)} \\ & + w_{6,\text{sell}} \times \text{Imbalance} - 0.5 \times \text{Time Pressure}.\end{aligned}$$

- Separately trained weights for a linear combination
- Scoring threshold is tuned to be (0,0) in milestone 2
    - buy if buy score > 0, sell if sell score < 0
    - If no trades meet scoring thresholds, executes last trade each minute

Refinements and Optimization

Rolling Window Design

- **Change**: Implemented overlapping rolling windows for train/validate segmentation —— Forward-looking while frequently re-tune on recent data.
- **Result**: More robust performance across intraday sessions, mimic reality where can't keep all past information

Signal Weight Optimization

- **Change:** The weight used in scoring optimized over training data using 500 sampled parameter sets —— Re-learning weights per window account for signal strength
- **Result:** Higher score relevance, tighter buy-sell execution spreads

End-of-Minute Fallback Execution

- **Change**: Trade the last order of minute if no thresholds met —— Guaranteed execution and position flattening every minute
- **Result**: Reduced variance in returns, avoided holding risk

Performance Table From Backtest

	AAPL	AMZN	GOOG	INTC	MSFT
buy diff	-0.046316	0.006531	0.068372	0.005238	0.004286
sell diff	0.085263	0.044286	0.061163	0.007857	0.008333
benchmark diff	0.10135	0.10151	0.21476	0.01202	0.01253
algo diff	0.061679	0.052245	0.053953	-0.00309	-0.002619
percent improved	39.14%	48.53%	74.88%	125.71%	120.90%

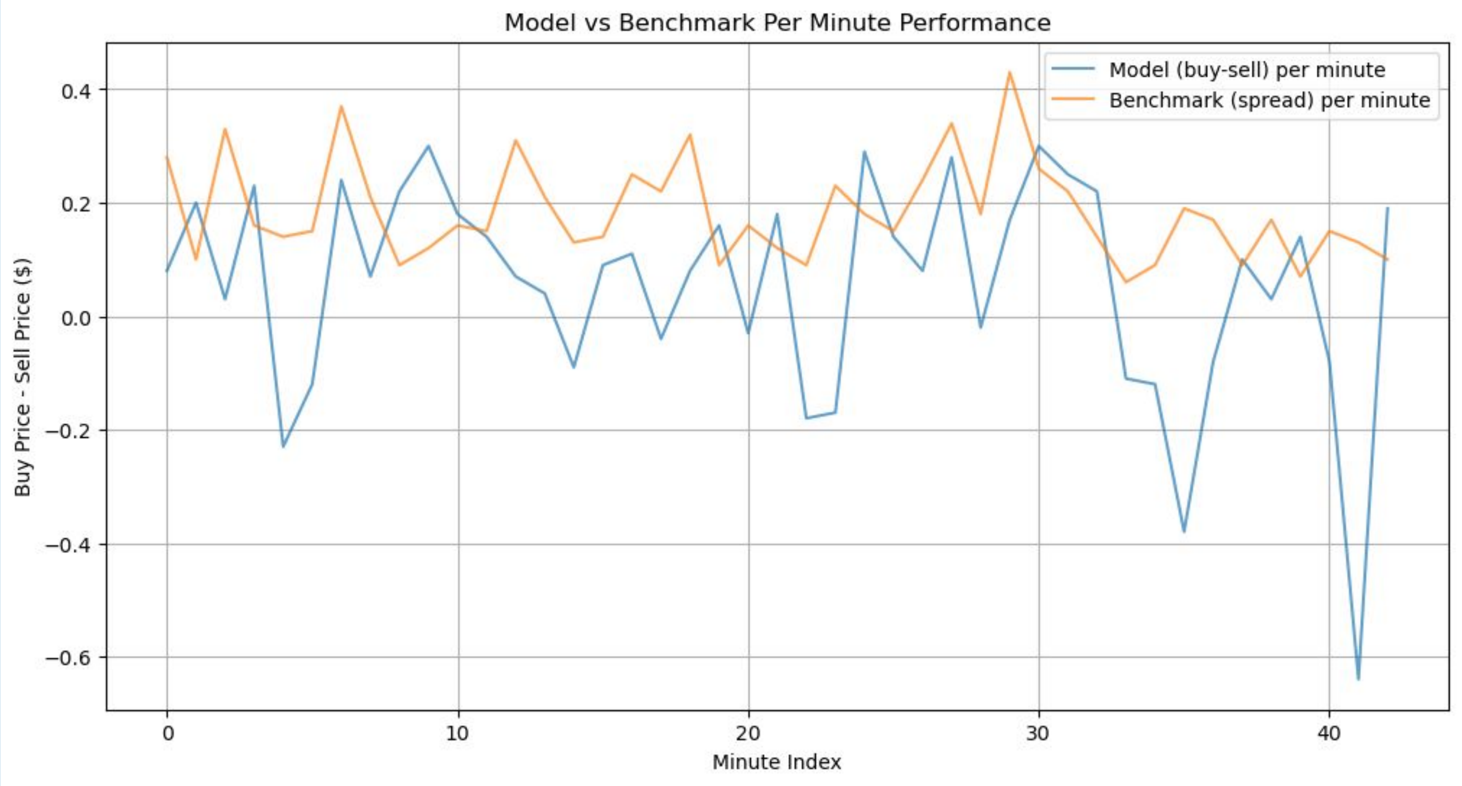
$$\text{Percentage Improvement} = \left(\frac{\text{Benchmark Performance} - \text{Our Algo Performance}}{\text{Benchmark Performance}}\right) \times 100$$

1. Best Real Performers
   a. GOOG: benchmark difference (0.21) reduced to (0.05) => 70% improvement
   b. AAPL and AMZN had solid absolute improvement with better buy/sell differences
2. INTC and MSFT struggled in the algorithm; produced extremely high improvement percentages due to low benchmark baseline (around 0.012)
   a. Requires further tuning as model may be overreacting to weak signals

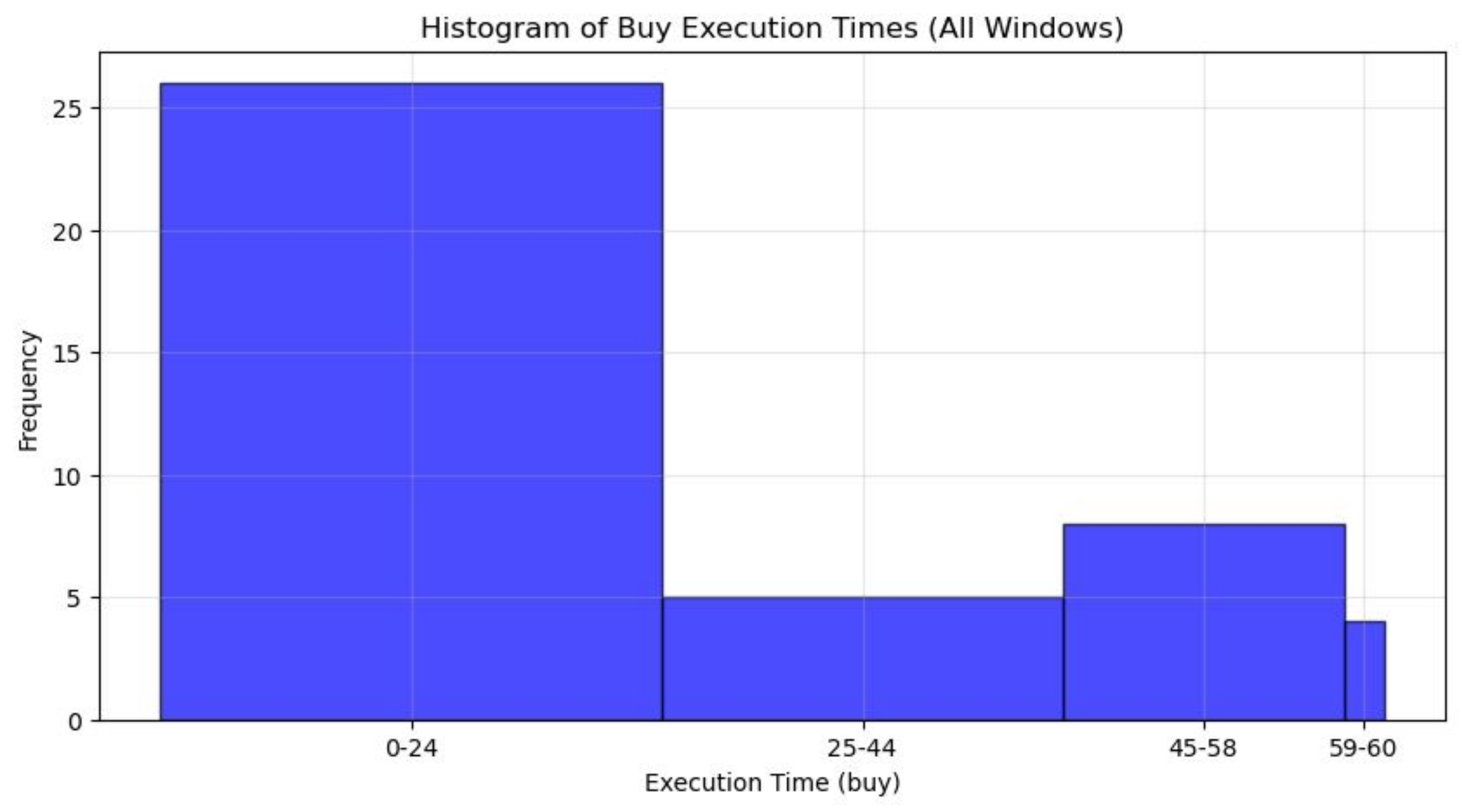
Visualizations in example of GOOG

Results & Visualizations

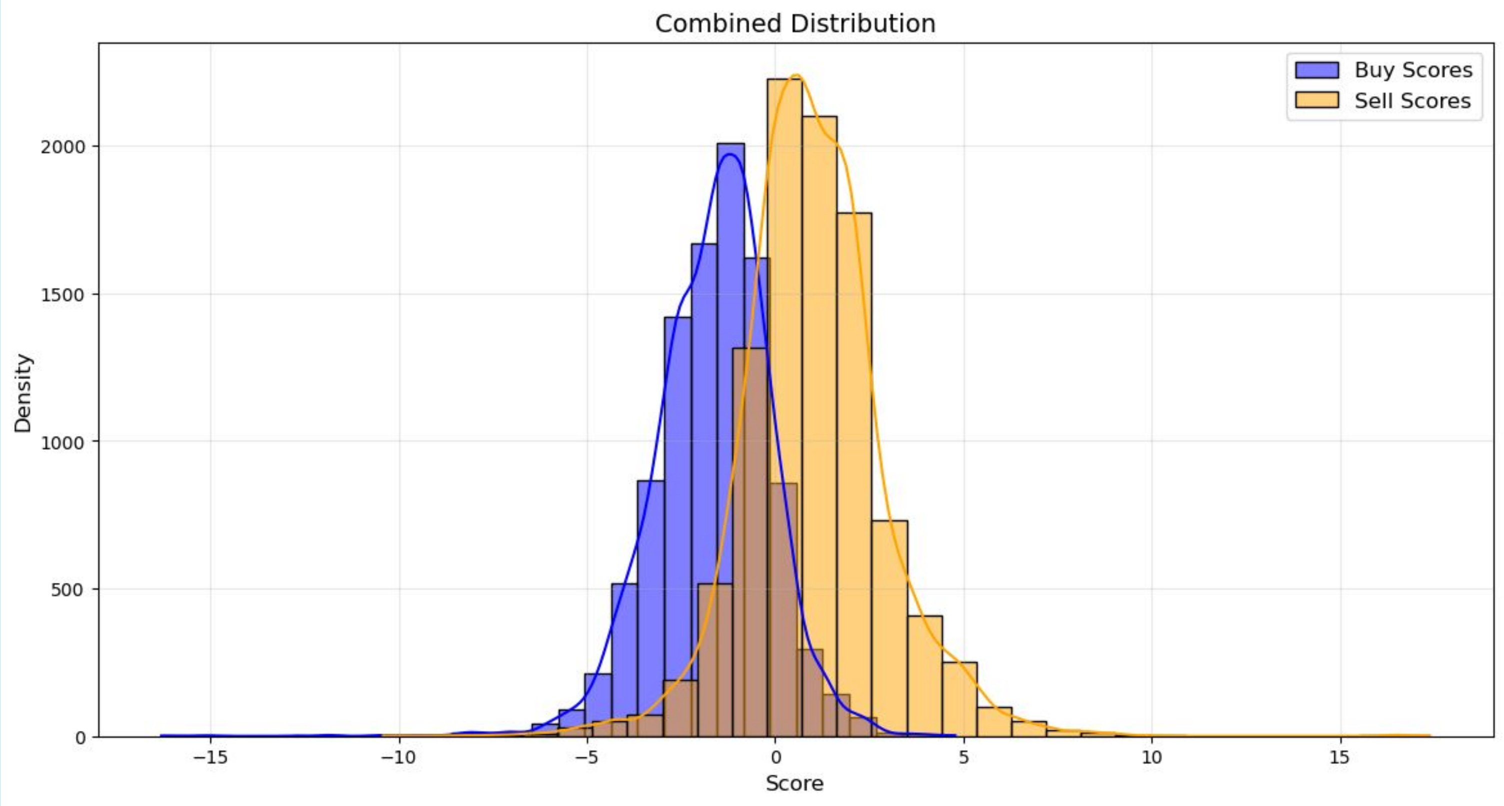
Model vs Benchmark per minute performance

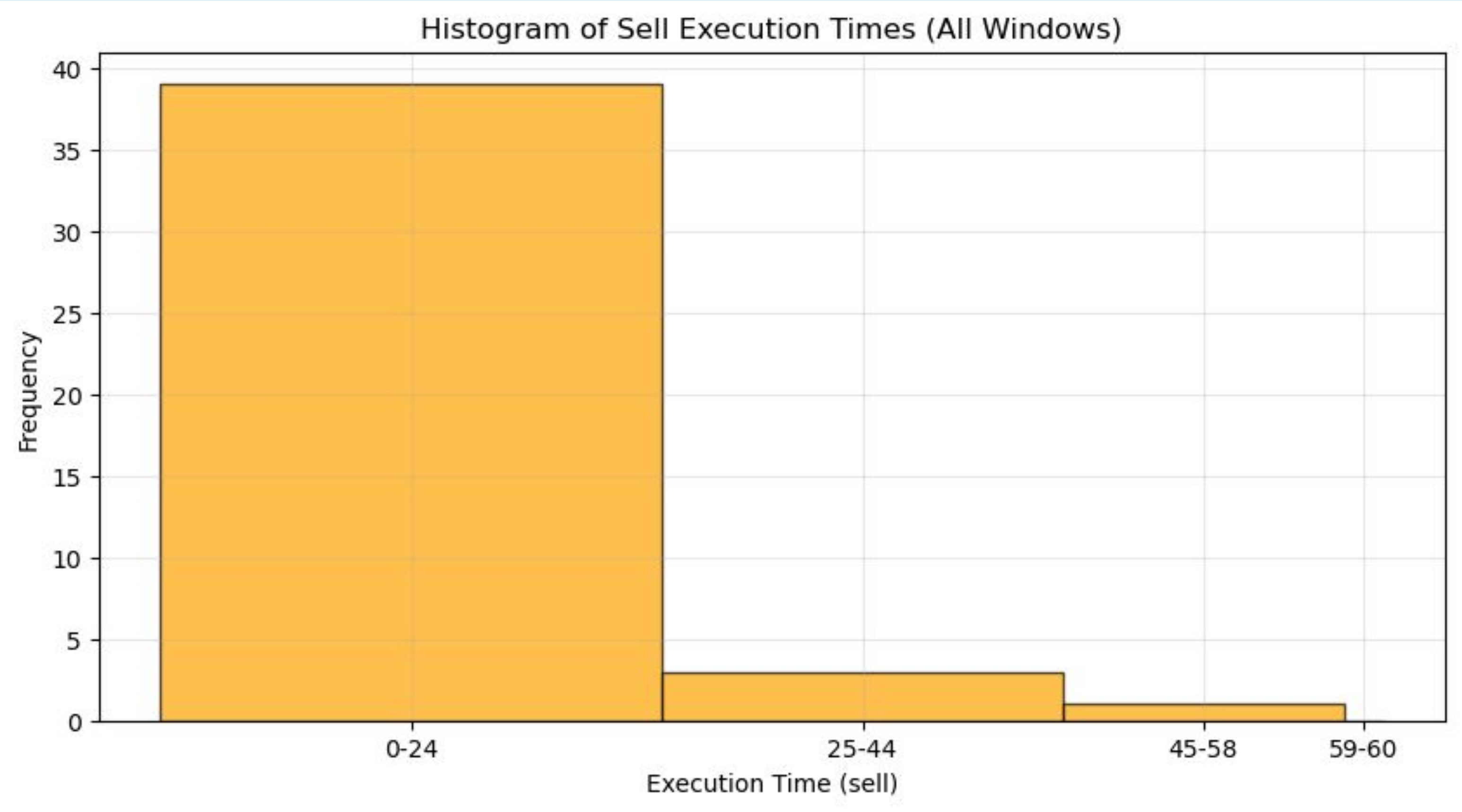


Execution Time Histograms



Distribution of Buy and Sell Score





Critical Evaluation

😮 What Surprised Us

- **End-of-minute trades worked better than expected**: Fallback trades consistently reduced variance and ensured execution continuity.
- **Score sensitivity**: Minor changes in weights led to disproportionate changes in execution timing, showing the fragility of scoring in volatile periods.

🧪 What We'd Try Next

- **Smarter parameter search**: Move beyond random sampling to smarter methods for weight and threshold tuning.
- **Dynamic thresholds**: Instead of static cutoffs, learn threshold values as a function of volatility or market regime.

Conclusion

📈 Summary

- Buy-sell spread improvement over the TWAP benchmark varies from 40% to 125 %.
- The strategy adapts to momentum, volatility, imbalance, and order flow, and executed trades earlier, capturing more favorable pricing

✅ Key Takeaways

- Adaptive scoring improves microstructure execution when tuned with rolling validation.
- Fallback logic ensures stability and realism under weak signals.
- Rolling windows align the model with live strategy needs, reducing overfitting.