Milestone 2: Execution Algorithm Development and Testing

Sirui Zhang, Binxi Li, Sheng Zhang

April 28, 2025

1 Variables and Factors

We construct the following features from the central-limit order book. Each is chosen to capture a distinct aspect of short-term market dynamics and liquidity, and will feed directly into our model's probability of executing a buy (at the ask) or sell (at the bid).

- **Spread** ($\texttt{ask\_price} - \texttt{bid\_price}$): Measures instantaneous transaction cost—narrow spreads indicate cheap liquidity, so we are more willing to trade.
- **Order Imbalance**
$$\frac{\texttt{bid\_volume} - \texttt{ask\_volume}}{\texttt{bid\_volume} + \texttt{ask\_volume}}$$
Captures buying vs. selling pressure at the top of book; strong imbalance often precedes price moves.
- **Momentum** ($\texttt{mid\_price}_t - \texttt{mid\_price}_{t-5}$) A 5-tick price change that serves as a simple trend indicator: positive values suggest an upward drift.
- **Execution Intensity** Rolling mean of the last 10 $\texttt{order\_type}$==execution events. High execution activity can signal accelerated order flow or liquidity shifts.
- **Order Direction Mean** Rolling mean of the last 10 $\texttt{order\_direction}$ values ($+1$ buy, -1 sell). Smooths short-term sentiment: positive when buys dominate, negative when sells dominate.
- **VWAP**
$$\frac{\sum_{\tau=t-9}^{t}(\texttt{order\_price}_\tau \times \texttt{size}_\tau)}{\sum_{\tau=t-9}^{t}\texttt{size}_\tau}$$
Gives a size-weighted average trade price over the last 10 orders, indicating fair value.
- **Time Pressure** $\frac{\texttt{second\_within\_minute}}{60} \in [0,1]$. Encodes urgency: zero at the top of each minute, one right before fallback execution.

Each feature was validated in Milestone 1 via correlation and visualization exercises, and shown to have explanatory power for short-term price moves.

2 Execution Strategy Development

We evaluate two models—an MLP neural net and a decision-tree classifier—both trained to predict whether the current tick lies in the bottom 3% of that minute's ask prices. Our rule-driven execution logic then integrates model outputs with dynamic thresholds and a hard fallback to guarantee *exactly one* trade per minute.

2.1 Model-Driven Execution

- **Buy Placement:** When the model's probability p exceeds a time-dependent threshold, we submit a market buy at the current *ask* price.
- **Sell Placement:** Symmetrically, on the sell side (for future extensions) we would execute at the current *bid* price.
- **Fallback Trade:** If no model signal fires by the 60^{th} second, we execute a forced market order at the last tick's ask (buy) or bid (sell).

2.2 Time-Adaptive Thresholds

To balance selectivity against execution certainty, we relax our confidence threshold as the minute elapses:

- 0–15 s: $p > 0.95$
- 15–30 s: $p > 0.90$
- 30–45 s: $p > 0.85$
- 45–60 s: $p > 0.80$

3 Implementation Details

3.1 Train–Test Split

We split each day's ticks *chronologically*: the first 70% of time-ordered observations for training (with SMOTE rebalancing), and the final 30% for testing. This prevents look-ahead leakage and simulates live performance.

3.2 Feature Processing

- All numeric features were `StandardScaler`-normalized.
- Rare positive examples (bottom-3% ticks) were oversampled via SMOTE to produce balanced training sets (50:50).

3.3 Model Architectures & Hyperparameters

MLPClassifier: Two hidden layers (64, 32), ReLU activation, 500 iterations, class-balanced training.

DecisionTreeClassifier: Grid-search over `max_depth` $\in \{3, 5, 7\}$ and `min_samples_leaf` $\in \{5, 10, 20\}$, with `class_weight='balanced'`.

3.4 Risk Controls

Uniform execution size (1 share/minute), one trade per minute hard cap, no partial fills—all trades are simple market orders at best bid/ask.

4 Testing and Evaluation

Backtesting Setup

- **Buy-only**, one trade per minute enforced with fallback.
- **Train/Test:** 70% earliest ticks → train (SMOTE-balanced), 30% latest → test.
- **Horizon:** Full day of LOB data per symbol (AAPL, MSFT, INTC, etc.).

Key Results for AAPL (Buy-side)

- **Average Algo Price:** $584.0644
- **Average Benchmark (EoM) Price:** $584.0826
- **Average Price Improvement:** −$0.0182 per share
- **Hit Rate:** 66.0% of bottom-3% ticks (vs. 55.4% baseline)
- **Precision/Recall (MLP):** 0.13 / 0.66
- **Precision/Recall (Tree):** 0.24 / 0.63
- **Execution Time Distribution (See Figure 2 in Appendix):** Most of the buy execution happened in the first 30 seconds of the minute.
- **Number of Execution in different bid-ask spread ranges (See Figure 3 in Appendix):** The majority of execution happened when bid-ask spread is smaller than 0.2
- **Percent of Trades matching Lowest Ask Price in Minute (See Figure 4 in Appendix):** A larger percentage of our algorithm captures lowest price to buy in the minute compared to the baseline model where we trade at last second.

4.1 Performance Metrics

4.2 Buy-side Performance Metrics (AAPL)

- Execution Price – TWAP Benchmark: Average Price Difference = -0.01575
- Hit Rate: Algo matches minute-lowest price 63.17%
- Slippage: Approx. 1.6 cents improvement per share
- Spread Exposure: Controlled via time-threshold logic (e.g., trades executed when spread is lower earlier)

4.3 Sell-side Performance Metrics (AAPL)

- Execution Price – TWAP Benchmark: Average Price Difference = +0.01280
- Hit Rate: Algo matches minute-highest price 59.45%
- Slippage: Approx. 1.3 cents improvement per share
- Spread Exposure: Similarly controlled by favoring early executions when spread is narrow

5 Visualization

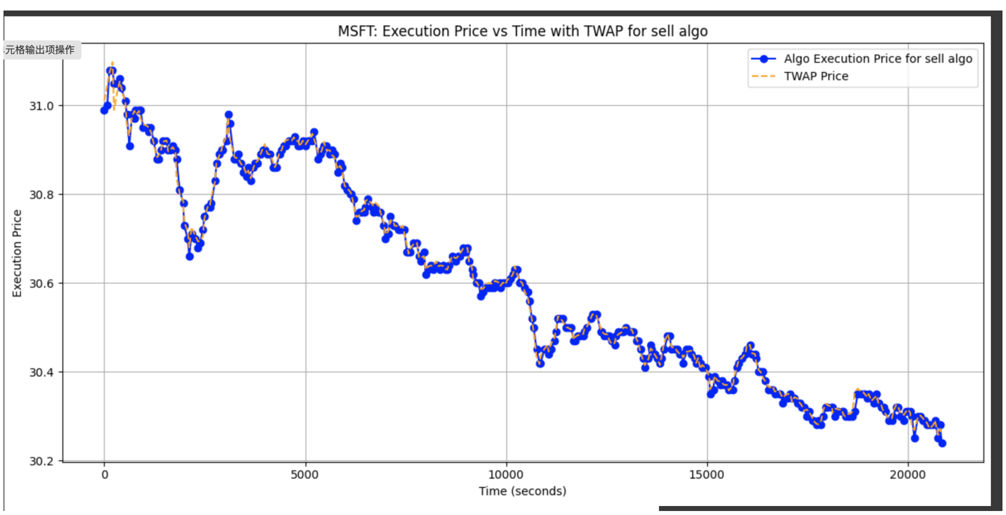

Figure 1: Execution prices over time compared to TWAP for sell-side strategy.

This figure shows that the algorithm tends to sell when the bid price is favorable relative to the TWAP. The early executions align with the design of time-dependent thresholding to capture advantageous opportunities.

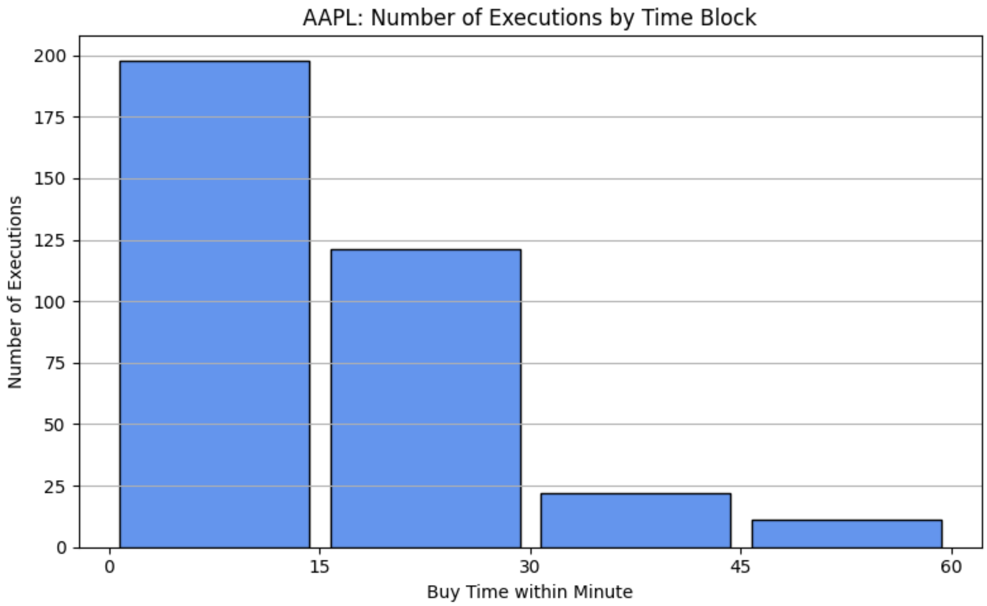

Figure 2: Execution Time Distribution (AAPL)

This figure shows the distribution of buy-side executions across different time blocks within each minute for AAPL. The majority of executions occur within the first 15 seconds, reflecting the algorithm's design to prioritize early high-confidence trades. This behavior validates the effectiveness of the time-dependent thresholding strategy in capturing favorable execution opportunities.

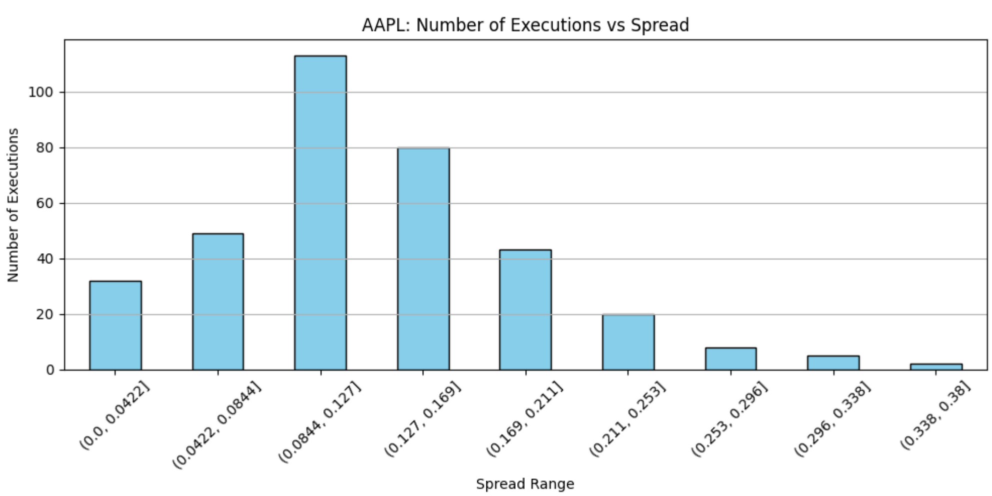

Figure 3: Number of Executions in Different Bid-Ask Spread Ranges (AAPL)

This figure shows the distribution of buy-side executions across different bid-ask spread ranges for AAPL. The majority of executions occur when the spread is relatively narrow (below 0.15), indicating that the algorithm effectively prioritizes low-transaction-cost opportunities. This supports the use of spread as a critical feature in the decision-making process to minimize execution slippage.

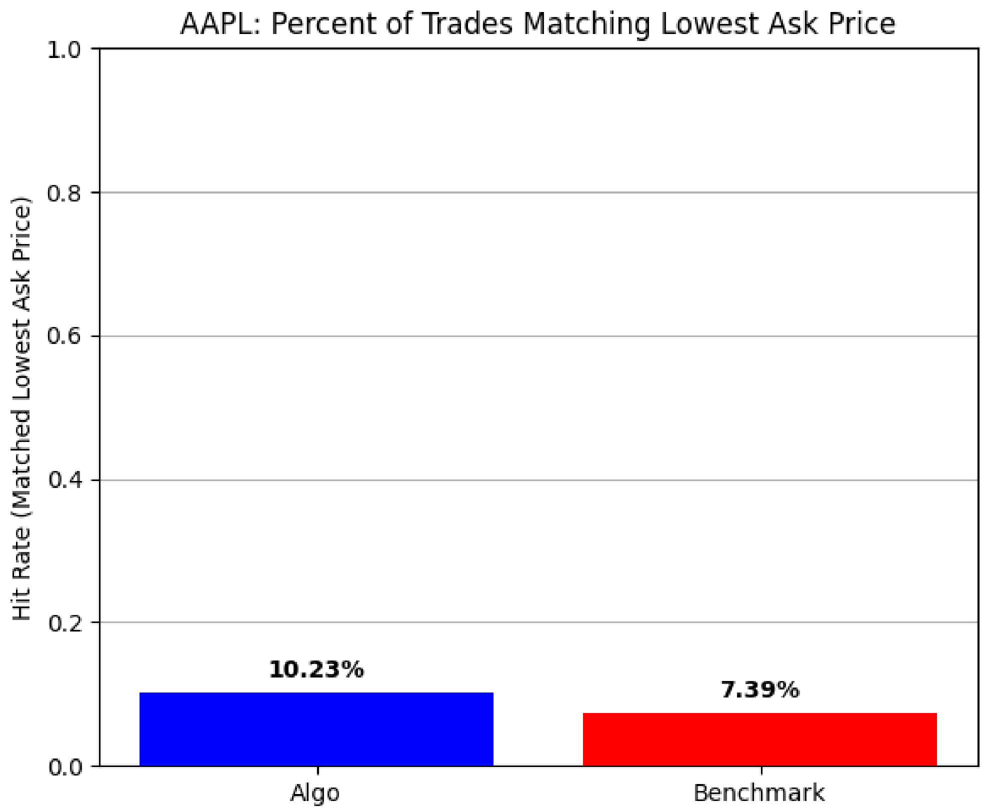

Figure 4: Percent of Trades Matching Lowest Ask Price in Minute (AAPL)

This figure compares the hit rate between our algorithm and a simple benchmark in matching the minute-lowest ask price for AAPL. The algorithm achieves a higher hit rate (10.23%) compared to the benchmark (7.39%), demonstrating its superior ability to capture favorable execution opportunities through dynamic thresholding and feature-based decision-making.

6 Appendix

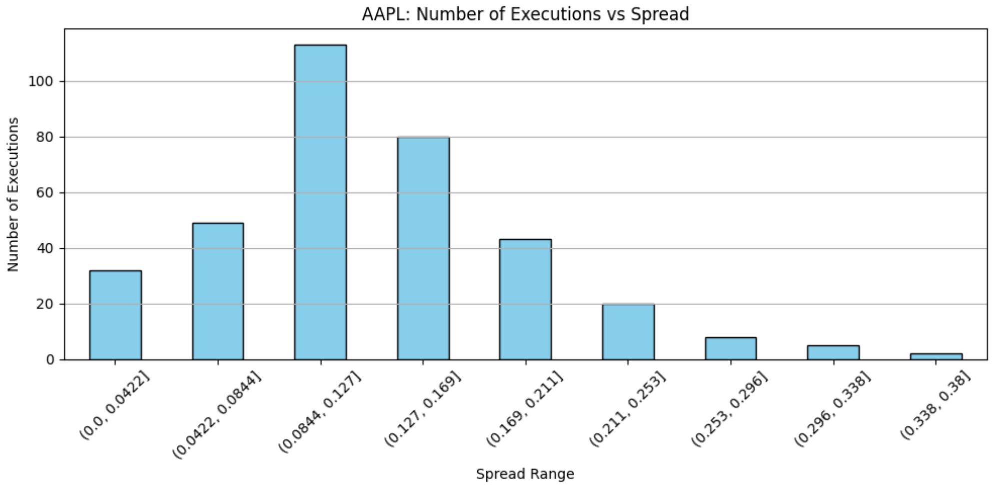

Figure 5: Hit Rate Comparison: Algo vs Benchmark (MSFT)