Hawkes Process for Realized Volatility and Price Movement Predictions

Isaac Mcallister

November 29, 2025

1 Introduction

1.1 Motivation and the High-Frequency Problem:

When I started this project, my first question was why stock prices look so erratic at short timescales. The fundamental value of a company like AAPL doesn't change from minute to minute. What does change is the balance of supply and demand: hedge funds adjusting positions, HFT firms exploiting small inefficiencies, and retail orders hitting the book. These small actions accumulate into constant micro-movements that create a sea of noise.

One approach for capturing this behavior is the Hawkes process. Its core idea is that events cluster: a price move increases the likelihood of more moves shortly after. This makes it a natural way to describe microstructure dynamics. There's plenty of discussion about using Hawkes processes to model market behavior, but far less on actually turning those models into a workable trading strategy.

That became the goal here: combine Hawkes modeling with machine-learning magic to test whether a practical strategy could be built.

1.2 Project Outputs:

On the other screen you will see several plots and numbers, a quick explanation:

- **Plot of Market Price:** This shows the mid, bid, and ask price. Data is taken from the bb0-1s top-of-order-book dataset.
- **Hawkes Intensities:** Full explanation for the math below. These are event intensities for the following market events:

  - MU: mid price up
  - MD: mid price down
  - ADNM: ask size down, no mid price change
  - AUNM: ask size up, no mid price change
  - BDNM: bid size down, no mid price change
  - BUNM: bid size up, no mid price change
- **Predicted Mid Price Direction:** ML model outputs, predicted mid-price direction. This model combines two XGB classifiers. One uses the Hawkes intensities to predict price jumps, the other predicts the direction of the price jump. A prediction is only made if the direction confidence is above 60%. Both models are about 70% accurate; see below for all model CV scores and results.
- **Real Mid Price Delta (5s):** What the average mid-price change will be in the next 5 seconds (current price jump not included). Compare to the predicted mid-price direction to see if the model is accurate or not.
- **RV Forecast:** Realized volatility forecast for the next 5 seconds using Hawkes intensities, R^2 score of 0.43. See below for further discussion.
- **Secondary Features:** Common market features used to analyze order books, used as additional features in the ML models.

References

[1] Timoshenko, N. (2024). *Modelling High-Frequency Market Microstructure with Hawkes Processes*. arXiv:2405.10527. Available at: https://arxiv.org/html/2405.10527v1.

[2] McLean, D. (2022). *Modelling Microstructure Noise Using Hawkes Processes*. Available at: https://dm13450.github.io/2022/05/11/modelling-microstructure-noise-using-hawkes-processes.html.

[3] Hicks, M. (2017). *A Deep Dive into Ogata's Algorithm*. Simply Statistics. Available at: https://simplystatistics.org/posts/2017-09-04-deep-dive-ogata/.

[4] Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., & Frank, L. M. (2002). *The Time-Rescaling Theorem and Its Application to Neural Spike Train Data*. Available at: https://sites.stat.columbia.edu/liam/teaching/neurostat-fall17/papers/brown-et-al/time-rescaling.pdf.

2 Project Resources

Github link:
For the EM algorithm I used the Hawkes lib python package: https://hawkeslib.readthedocs.io/en/latest/
In its current form this package is broken, I rewrote the cypthon EM algorithm in python and modified it use fix θ parameters (see below for explanation.)

3 Hawkes Process Theory

3.1 Introduction

Hawkes processes rely on the idea of self-excitation, that an event might trigger a series of future events. A basic Hawkes process is a stochastic process $N(t)$ that counts the number of events up to time t, and the rate of arrivals is a function of the past state. Mutually exciting Hawkes processes model multiple processes that work together: one process may excite another.

3.2 Classical Hawkes Process

Definition: A stochastic process $N(t)$ defined for $t \geq 0$ is a counting process starting at $N(0) = 0$ if $N(t)$ only takes values in $\{0, 1, 2, \cdots\}$ and increases in jump sizes of $+1$. The random jump times $T_1, T_2, \cdots$ form a point process with $0 < T_1 < T_2 < \cdots$. The counting process is then defined as

$$N(t) = \sum_{i=1}^{\infty} I\{T_i \leq t\} = \sum_{T_i \leq t} 1. \tag{1}$$

The point process is simple: there cannot be more than one jump at any given time.

- T_i are also called arrival times.
- $T_0 = 0$ is not counted by N_t.
- $E_i = T_i - T_{i-1}$ are called the inter-arrival times.

The simplest counting process is the Poisson process.

3.2.1 Conditional Intensity and Compensator

Definition: The conditional intensity process of N_t is the instantaneous expected rate of events given the previous history. It is defined for $t \geq 0$ by

$$\lambda^*(t) = \lim_{\Delta \to 0} \frac{E(N_{t+\Delta} - N_t \mid H_t)}{\Delta}. \tag{2}$$

We have defined H_t as the filtration or history of the process N_t. The process $\lambda^*(t)$ is itself a random process, representing the instantaneous rate of arrivals at time t, given the arrival times up to, but not including, t. However, if we consider the special case where $\lambda^*(t) = \lambda(t)$, so the conditional intensity does not depend on the previous history, then N_t is a (possibly inhomogeneous) Poisson process with rate function $\lambda(t)$.

$N(t)$ **is right-continuous with left limits:** N_t only jumps at event times T_i. Between events it is constant, and just before an event it is constant. Mathematically, $N_t = N_{t^-} + 1$ at each jump time t. At any time t the function equals the limit to its right, but it has a left limit that is always defined, N_{t^-}.

$\lambda^*(t)$ **is left-continuous with right limits:** H_t is defined with all the information right before t but not including t. Immediately after the event the history changes and there is an instantaneous jump. The function is well defined up to t, then after t its right limit differs.

Definition: The compensator of the point process N_t is defined for $t \geq 0$ by

$$\Lambda_t = \int_0^t \lambda^*_s \, ds, \tag{3}$$

which describes the expected cumulative intensity up to time t and satisfies $E[N(t)] = E[\Lambda_t]$.

3.2.2 Self-Exciting Property

Definition: A Hawkes process is a counting process N_t whose conditional intensity process for $t \geq 0$ is

$$\lambda^*_t = \mu + \sum_{T_i \leq t} \gamma(t - T_i), \tag{4}$$

where $\mu > 0$ is the background arrival rate, and γ is the excitation function. A simple and common choice for the excitation function is $\gamma(t) = \alpha e^{-\theta t}$.

3.2.3 Excitation Functions and Markov Structure

The choice of excitation kernel γ is crucial, as it can encode domain-specific structure. For the exponential kernel, the pair (N_t, λ_t^*) is a Markov process: the future evolution depends on the past only through the current intensity. Let T_n denote the n-th jump time. At a jump, the intensity has a fixed upward jump:

$$\lambda_{T_n}^* = \lambda_{T_n^-}^* + \alpha, \tag{5}$$

where $\lambda_{T_n^-}^*$ is the left limit (intensity just before the event). Between jumps, $T_n < t < T_{n+1}$, the intensity decays exponentially toward μ:

$$\lambda_t^* = \mu + \left(\lambda_{T_n}^* - \mu\right) e^{-\theta(t-T_n)}. \tag{6}$$

This piecewise-deterministic Markov structure is what makes inference and simulation for exponential Hawkes processes relatively tractable.

Definition: An exponentially decaying Hawkes process is a counting process N_t defined by $\lambda_0, \mu, \alpha, \theta > 0$ whose conditional intensity starts at λ_0 and for $t \geq 0$ follows

$$\lambda_t^* = \mu + (\lambda_0 - \mu)e^{-\theta t} + \sum_{T_i < t} \alpha e^{-\theta(t-T_i)} \quad \text{hence} \tag{7}$$

$$\Lambda_t = \mu t + \frac{(\lambda_0 - \mu)}{\theta}(1 - e^{-\theta t}) + \sum_{T_i < t} \frac{\alpha}{\theta}(1 - e^{-\theta(t-T_i)}). \tag{8}$$

Here we introduce λ_0, which is the intensity at the start of our process.

3.2.4 Likelihood function

Given as

$$\ell = \sum_{i=1}^{n} \log \lambda_{T_i}^* - \Lambda_T. \tag{9}$$

See [1] for a derivation.

3.3 Multivariate Hawkes

The multivariate form involves multiple event types (marks), and these event types can excite each other. Let Ω be all possible events; then we are interested in a set of k events

$\{w_1, w_2, \cdots, w_k\} \subset \Omega$. Take $i, j \in \{w_1, w_2, \cdots, w_k\}$. We then have several counting processes defined by $N_i(t)$. Each counting process will look like $T^i = \{T_1^i, T_2^i, T_3^i, \cdots, T_n^i\}$; we index events in process i by $l \in [1, n]$. Then we can write the individual excitation intensity function as

$$\lambda_i^*(t) = \mu_i + \sum_{j=1}^{k} \sum_{T_l^j \leq t} \gamma_{ij}(t - T_l^j). \tag{10}$$

As discussed, it is most common to take the kernel function γ_{ij} as an exponential; we can write it as

$$\gamma_{ij}(t) = \eta_{ij} e^{-\theta_{ij} t}. \tag{11}$$

The problem of inference then boils down to identifying the base rates $\mu_1, \mu_2, \cdots, \mu_k$, the joint excitation strength η_{ij}, the individual excitation strength η_{ii}, and the joint decay rate θ_{ij} and individual decay rate θ_{ii}.

Number of offspring: It is sometimes more helpful to parameterize not by the excitation strength but instead by the expected number of offspring, as in the number of type i events directly triggered by one type j event, defined as α_{ij}. We can write this as

$$\int_0^\infty \gamma_{ij}(t)\, dt = \int_0^\infty \eta_{ij} e^{-\theta_{ij} t} dt = \frac{\eta_{ij}}{\theta_{ij}} = \alpha_{ij}. \tag{12}$$

All these α_{ij} can be put together to form the **infectivity matrix.** It is equally valid to parameterize by α_{ij}. Continuing, we are going to write our kernel instead as

$$\gamma_{ij}(t) = \alpha_{ij} \theta_{ij} e^{-\theta_{ij} t}. \tag{13}$$

3.3.1 Stability Conditions

For the multivariate exponential Hawkes process to be stable (i.e., non-explosive), the matrix $A = (\alpha_{ij})$ must satisfy the standard branching condition

$$\rho(A) < 1, \tag{14}$$

where $\rho(A)$ denotes the spectral radius. This ensures that the total expected number of offspring events remains finite and that the intensity process admits a stationary limit.

3.3.2 Markov Property

Note first that we can write

$$\lambda_i^*(t) = \mu_i + \sum_{j=1}^{k} \sum_{T_l^j \le t} \alpha_{ij}\theta_{ij} e^{-\theta_{ij}(t-T_l^j)} \tag{15}$$

$$= \mu_i + \sum_{j=1}^{k} \alpha_{ij}\theta_{ij} \sum_{T_l^j \le t} e^{-\theta_{ij}(t-T_l^j)}. \tag{16}$$

Define the state process

$$S_{ij}(t) = \sum_{l:T_l^j \le t} e^{-\theta_{ij}(t-T_l^j)}. \tag{17}$$

We now show that $S_{ij}(t)$ is a Markov process. Consider the m-th event of type j, occurring at time t_m^j. Let

$$S_{ij}^{(1)} = S_{ij}(t_m^j-)$$

be the state just before t_m^j, and let $S_{ij}^{(2)}(t)$ denote the state after this event. For $t \ge t_m^j$ we have

$$S_{ij}^{(2)}(t) = \sum_{l \le m-1} e^{-\theta_{ij}(t-T_l^j)} + e^{-\theta_{ij}(t-t_m^j)} \tag{18}$$

$$= \sum_{l \le m-1} \left(e^{-\theta_{ij}(t-t_m^j)} e^{-\theta_{ij}(t_m^j-T_l^j)} \right) + e^{-\theta_{ij}(t-t_m^j)} \tag{19}$$

$$= e^{-\theta_{ij}(t-t_m^j)} \left(\sum_{l \le m-1} e^{-\theta_{ij}(t_m^j-T_l^j)} + 1 \right). \tag{20}$$

The inner sum no longer depends on t, so we can write

$$S_{ij}^{(2)}(t_m^j) = S_{ij}^{(1)} + 1, \tag{21}$$

$$S_{ij}^{(2)}(t) = (S_{ij}^{(1)} + 1)e^{-\theta_{ij}(t-t_m^j)}, \qquad t > t_m^j. \tag{22}$$

Thus, given the value $S_{ij}(t_m^j-)$ immediately before the jump, the post-jump value at t_m^j and its subsequent evolution for $t > t_m^j$ depend only on this current state and the elapsed time. In particular, $S_{ij}(t)$ evolves via deterministic exponential decay between jumps and a unit upward jump at each event of type j, so S_{ij} is a Markov process.

Finally, the intensity can be written as

$$\lambda_i^*(t) = \mu_i + \sum_{j=1}^{k} \alpha_{ij}\theta_{ij} S_{ij}(t). \tag{23}$$

Compensator Explicit Form: We can derive

$$\Lambda_i(t) = \int_0^t \lambda_i^*(s)\, ds = \int_0^t \left(\mu_i + \sum_{j=1}^{k} \alpha_{ij}\theta_{ij} S_{ij}(s) \right) ds = \mu_i t + \sum_{j=1}^{k} \alpha_{ij}\theta_{ij} \int_0^t S_{ij}(s)\, ds. \tag{24}$$

Then consider the state integration,

$$\int_0^t S_{ij}(s)\, ds = \frac{1}{\theta_{ij}} \sum_{l:\, T_l^j < t} \big(1 - e^{-\theta_{ij}(t - T_l^j)}\big). \tag{25}$$

Hence the compensator becomes

$$\Lambda_i(t) = \mu_i t + \sum_{j=1}^{k} \alpha_{ij} \sum_{l:\, T_l^j < t} \big(1 - e^{-\theta_{ij}(t - T_l^j)}\big). \tag{26}$$

Now, to highlight the Markov property, define the most recent event time of type j as T_n^j. We can decompose the sum as

$$\sum_{l:\, T_l^j < t} \big(1 - e^{-\theta_{ij}(t - T_l^j)}\big) = N_j(t) - \sum_{l:\, T_l^j < t} e^{-\theta_{ij}(t - T_l^j)} = N_j(t) - S_{ij}(t), \tag{27}$$

so that

$$\Lambda_i(t) = \mu_i t + \sum_{j=1}^{k} \alpha_{ij} \big(N_j(t) - S_{ij}(t)\big). \tag{28}$$

Looking at the compensator increment between arrival times T_n and T_{n+1}, we have

$$\Lambda_i(T_{n+1}) - \Lambda_i(T_n) = \mu_i(T_{n+1} - T_n) + \sum_{j=1}^{k} \alpha_{ij} \Big(N_j(T_{n+1}) - N_j(T_n) + S_{ij}(T_n) - S_{ij}(T_{n+1}) \Big) \tag{29}$$

$$= \mu_i(T_{n+1} - T_n) + \sum_{j=1}^{k} \alpha_{ij} \Big(1 + S_{ij}(T_n) - S_{ij}(T_{n+1}) \Big), \tag{30}$$

where in the last line we used that between two successive jumps of type j we have $N_j(T_{n+1}) - N_j(T_n) = 1$. Define

$$\Delta\Lambda_i(T_{n+1}, T_n) = \int_{T_n}^{T_{n+1}} \lambda_i^*(t)\,dt = \mu_i(T_{n+1} - T_n) + \sum_{j=1}^{k} \alpha_{ij}\Big(1 + S_{ij}(T_n) - S_{ij}(T_{n+1})\Big). \tag{31}$$

By the time-rescaling theorem we have $\Delta\Lambda_i(T_{n+1}, T_n) \sim \text{Exp}(1)$ (see [4], equation (2.16)).

3.3.3 Likelihood

For an event at (t_m, c_m) with $c_m \in \{1, \dots, k\}$,

$$\log L(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \sum_{m=1}^{M} \log\left[\mu_{c_m} + \sum_{j=1}^{k} \alpha_{c_m j}\,\theta_{c_m j}\,S_{c_m j}(t_m)\right] - \sum_{i=1}^{k}\left[\mu_i\,T + \sum_{j=1}^{k} \alpha_{ij} \sum_{T_l^j < T}\Big(1 - e^{-\theta_{ij}(T - T_l^j)}\Big)\right], \tag{32}$$

$$S_{ij}(t) = \sum_{T_l^j < t} e^{-\theta_{ij}(t - T_l^j)}, \quad \lambda_i^*(t) = \mu_i + \sum_{j=1}^{k} \alpha_{ij}\,\theta_{ij}\,S_{ij}(t). \tag{33}$$

To actually solve for the best parameters, computer programs use EM solvers.

3.4 Residuals

Now we come to the question of how we analyze the performance of our Hawkes parameters (besides log likelihood). Consider, for some event type i, arrival times T_n and T_{n-1}. The expected number of arrivals in this interval should be 1:

$$E[N(T_n) - N(T_{n-1})] = 1 = \int_{T_{n-1}}^{T_n} \lambda_i^*(t)\,dt. \tag{34}$$

Suppose our data actually has cumulative intensity increment $\Delta\Lambda_i(T_{n-1}, T_n)$, and define

$$z_i^n = \Delta\Lambda_i(T_{n-1}, T_n). \tag{35}$$

Now if $z_i^n > 1$ our model is generally too slow, and if $z_i^n < 1$ our model is too fast.

3.5 Residual Compensator (original work)

Let's apply a scaling factor δ over all our terms to compensate. We want to apply this compensator to λ_i^* directly:

$$\bar{\lambda}_i^*(t) = \delta\, \lambda_i^*(t). \tag{36}$$

Then our residual-adjusted increment looks like

$$\overline{\Delta\Lambda_i(T_{n-1}, T_n)} = \delta\, \Delta\Lambda_i(T_{n-1}, T_n), \tag{37}$$

$$\tag{38}$$

$$\bar{z}_i^n = \delta\, \Delta\Lambda_i(T_{n-1}, T_n) - 1. \tag{39}$$

Before continuing, note that it is not helpful to simply optimize for one residual. Instead, let's consider the previous k residuals:

$$\sum_{v=n-k+1}^{n} \bar{z}_i^v = \delta \sum_{v=n-k+1}^{n} \Delta\Lambda_i(T_{v-1}, T_v). \tag{40}$$

We want to force the mean around one, so let's denote carefully $\delta_{n,k}$ as the scaling factor at time step n, using the previous k terms:

$$\frac{1}{k} \sum_{v=n-k+1}^{n} \bar{z}_i^v = 1 = \frac{1}{k} \delta_{n,k} \sum_{v=n-k+1}^{n} \Delta\Lambda_i(T_{v-1}, T_v), \tag{41}$$

$$\delta_{n,k} = \frac{k}{\sum_{v=n-k+1}^{n} \Delta\Lambda_i(T_{v-1}, T_v)}. \tag{42}$$

Then define the one-step-forward residual. The idea is that we look at the step immediately after where the compensator was applied. Since the scaling factor is applied directly to λ_i^*, we have

$$\Delta\Lambda_i(T_n, T_{n+1}) = \int_{T_n}^{T_{n+1}} \lambda_i^*(t)\, dt, \tag{43}$$

$$\overline{\Delta\Lambda_i(T_n, T_{n+1})} = \delta_{n,k} \int_{T_n}^{T_{n+1}} \lambda_i^*(t)\, dt = \delta_{n,k} \Delta\Lambda_i(T_n, T_{n+1}). \tag{44}$$

The theory is then that these compensated residuals will average to one. Critically, we are using the previous k terms to adjust the $(n+1)$-th term—or more specifically to adjust the instantaneous intensity between arrivals n and $n+1$.

3.6 Analytical Hawkes Volatility Forecast

3.6.1 Volatility

Let's start by rigorously defining the thing we want to forecast. Realized volatility is a measure of how much price fluctuates, usually over a fixed time window.

Definition: Market Volatility Let p_t be the price of an asset at time t. We can define the return from $t-1$ to t as

$$r_t = \ln p_t - \ln p_{t-1}. \tag{45}$$

Then the volatility σ_T over a period T is the standard deviation of returns over that time period, or

$$\sigma_T^2 = E[r_t^2]. \tag{46}$$

A standard statistical estimator is

$$\hat{\sigma}_T = \sqrt{\frac{1}{T-1}\sum_{i=1}^{T} r_i^2}. \tag{47}$$

Definition: Realized Volatility The above definition is not, however, that useful for HFT applications. Instead we are interested in the continuous-time price model,

$$dX_t = \mu_t \, dt + \sigma_t \, dW_t, \tag{48}$$

which is used to estimate continuous-time volatility. We won't be getting into the stochastic calculus details here; however, the important definition is that the realized volatility (RV) can be estimated as

$$RV = \sum_{i=1}^{n} r_i^2. \tag{49}$$

The realized volatility converges to the continuous-time volatility:

$$RV \to \int_0^T \sigma_t^2 \, dt. \tag{50}$$

Connection to Hawkes Process: The realized volatility in an interval $[t, t+h]$ should be proportional to the expected Hawkes intensity on this interval.

3.6.2 Forecasting

Let $N_i(t)$ be the counting process of event type i, then the expected number of future events in the interval $[t, t+h]$ is

$$E(N_i(t+h) - N_i(t) \mid \mathcal{F}_t) = \int_t^{t+h} E[\lambda_i^*(s) \mid \mathcal{F}_t]\, ds. \tag{51}$$

Now the realized volatility over that time period can be defined as

$$RV_{t,t+h} = \sum_{i:\tau_i \in [t,t+h]} (\Delta X_{\tau_i})^2, \tag{52}$$

where τ_i corresponds to the time of a particular price jump, and ΔX_{τ_i} is the magnitude of the price change

$$\Delta X_{\tau_i} = \ln p_{\tau_i}^+ - \ln p_{\tau_i}^-. \tag{53}$$

For a simple example, let N_i model mid-price jumps, and p be the mid price. Then, if the price jumps are roughly equal in magnitude, we expect

$$RV_{t,t+h} \propto N_i(t+h) - N_i(t). \tag{54}$$

We can take expectations of both sides with respect to $\mathcal{F}_t$ and we get

$$E[RV_{t,t+h} \mid \mathcal{F}_t] \propto \int_t^{t+h} E[\lambda_i^*(s) \mid \mathcal{F}_t]\, ds. \tag{55}$$

Over short time horizons we get, for any $s > t$,

$$E[\lambda_i^*(s) \mid \mathcal{F}_t] = \mu_i + \sum_{j=1}^{k} \alpha_{ij}\theta_{ij} S_{ij}(t) e^{-\theta_{ij}(s-t)}. \tag{56}$$

Then the integral resolves to

$$\int_t^{t+h} E[\lambda_i^*(s) \mid \mathcal{F}_t]\, ds = \mu_i h + \sum_{j=1}^{k} \alpha_{ij}(1 - e^{-\theta_{ij} h}) S_{ij}(t). \tag{57}$$

The compensator could also be applied to this entire term if needed. Now what we really need to check is if this term is correlated with the real implied volatility.

4 Project Goals

We will study the following market events:

- Mid Price Up (MU): increase in the mid price - Event 0
- Mid Price Down (MD): decrease in the mid price - Event 1
- Increase in ask size, no mid price change (AUNM) - Event 2
- Increase in bid size, no mid price change (BUNM) - Event 3
- Decrease in ask size, no mid price change (ADNM) - Event 4
- Decrease in bid size, no mid price change (BDNM) - Event 5

We will fit a 6-dimensional Hawkes process to estimate all these events. We will be working with bb0-1s top-of-order-book data.

4.1 Estimate Branching Ratio and Parameters

The goal here is to study how parameters change throughout the day and start with some baseline estimates. Our theory is the compensator should allow the model to adjust these values (or specifically the outputted intensities) in live trading regimes.

- Fit the Hawkes process in the early morning, mid-day, and end of day; analyze parameters.
- Compare realized volatility in those regimes; see if Hawkes parameters are more unstable in more volatile hours.
- Implement the residual compensator and analyze residuals.

4.1.1 Design Decisions

BBO-1s dataset: For this project I used the BBO-1s dataset; this dataset contains top-of-order-book information aggregated every one second. I originally experimented with the MBO dataset, which contains the live top-of-order-book information (shows the top of the order book after every trade that changes it, occurring on millisecond scales). I found that MBO was incredibly noisy, and it was nearly impossible to fit a set of general parameters. MBO is also several times larger than BBO-1s, and my tiny laptop struggled to analyze even a single training day.

I still wanted to keep this project focused on market microstructure, and going to the BBO-1 minute dataset was no longer in this spirit. After some experimenting I found that BBO-1s was stable enough to detect some structure and signal. However, this dataset comes with some key caveats:

- Many market microstructure events that affect the mid price are washed out in the BBO-1s aggregate. We only get a single observation per second, when in reality many market events occur within that second.
- BBO-1s is not available live: this is the core limitation that already stops us from turning this into a real trading strategy. After some research I found you can get BBO-1s after a 1–2 second delay, but at that point any edge we have predicted is already overtaken by the market. If you wanted to turn this into a real trading strategy you would need to get access to the live trading feed from NASDAQ and NYSE (as is required for most HFT strategies).

If I were to revisit this project and had a bigger budget (more than the 20 dollars I allocated) I would look for a way to use the MBO dataset. I think it is possible to get the Hawkes process working using some of the Bayesian methods discussed later to make the parameters more reactive.

Marked, but no Magnitude: The Hawkes events above show relative direction (MU = mid increase) but not by how much. The magnitude of the price jump (or fall) should be clearly relevant. We could have, say, two large price jumps (MU, MU) and two small price decreases (MD, MD), and with our setup the Hawkes process now computes similar intensities for MU and MD. There are a few ways to address this issue, but I decided to ignore it for now for a few reasons:

- Most price jumps or decreases at the 1-second scale are of similar magnitude.
- We start deviating from the point-process architecture and breaking some core assumptions. This isn't the worst thing, but I wanted to keep this project's scope reasonable.
- The solutions I came up with (jitter for multiple arrivals, adding different events for different magnitudes, etc.) introduced more complexity and challenges.

This is another area I would return to if I had put more time into this project. Right now we are going to try and stick to the beaten path.

Fixed Theta: As discussed, the problem of inference resolves to estimating the below parameters for every state by maximizing log likelihood:

$$\lambda_i^*(t) = \mu_i + \sum_{j=1}^{k} \sum_{T_l^j \leq t} \alpha_{ij}\theta_{ij} e^{-\theta_{ij}(t-T_l^j)}. \tag{58}$$

That exponential term is nasty; it dramatically changes the geometry of the optimization problem. Originally I ran the EM algorithm to fit this parameter; you can see the experimental results under GitHub, in the file `hawkes_market_results_variable_theta.csv`. I found that fixing $\theta = \theta_{ij}$ did not affect the log likelihood significantly, and it dramatically sped up the EM algorithm. The value of θ really determines how long an event continues to affect intensity. Fixing $\theta = 0.1$ gave a decent log likelihood and let intensities react quickly to market events. You can go further and argue an interesting approach would be to use two simulation processes, a fast and slow process controlled by θ_f and θ_s; these processes could understand both short-term changes and long-term trends. I didn't do this because the residual compensator would rescale them to be very similar; however, this is another valid direction. Fixing θ was also necessary to let me fit parameters in a reasonable amount of time. It also made inter-day comparison of the other parameters more relevant.

4.2 Forecast Short-Term Volatility

For multiple trading days:

- Compute the 5-second realized volatility.
- Compare against Hawkes-forecasted intensities.
- Compute a 5-second volatility forecast using a regression model, with forecasted Hawkes intensities as inputs (intensities at the second before the 5-second prediction window).
- Analyze scores and results; see if Hawkes parameters are relevant predictors of realized volatility.

4.2.1 Design Decisions:

Regression vs Classification: When I first approached this problem I computed the 5-second realized volatility, then split it by quartiles into different regimes (slow, medium slow, medium fast, fast). The idea was we would use the Hawkes intensities to predict if we were entering more volatile regimes. This approach was very unstable and I struggled to get a decent model. Realized volatility values are themselves super noisy, and even though forecasted intensities do carry some information about future volatility (as argued above), there just wasn't enough predictive power to reliably forecast the regime.

Instead I decided on an easier problem: I decided to predict the log of the realized 5-second volatility

$$Y = \log(RV_5 + 1e - 9). \tag{59}$$

This smoothed out much of the short-term noise. My final model had an $R^2 = 0.43$ and MSE = 2 with 6-fold cross validation. I also decided to predict the 20% and 80% quantile bands as well to give a range to the model predictions.

XGB boost: Generally I personally prefer to use linear regression for these types of problems. Our data here, however, was very noisy and had many nonlinear relationships; we needed to use a more flexible model. XGB Boost had the most reliable scores and was an easy plug-and-play model. There are probably some performance gains in different GAM models, but XGB worked well enough for this application.

Why 5 seconds: A 5-second interval is somewhat arbitrary, but shorter windows like 1–2 seconds are too noisy to forecast reliably. Over 5 seconds, short-term trends are more visible and occasionally predictable, and the extra buffer also helps bridge the gap between receiving information and executing a trade.

4.3 ML Using Hawkes Intensities to Predict Mid-Price Changes

We construct a model that forecasts the direction of 5-second price changes.

- Use the Hawkes intensities as the input to an ML model to predict if there will be a large swing in price in the next 5 seconds.
- Use a secondary model to predict the direction of the price swing.

5 Branching Ratio and Parameters Results

The first task was to create a "Hawkes parameter survey." The idea was to refit the Hawkes parameters at different times of day and then examine how they evolve across many trading days. For this survey I used 14 days sampled randomly from 2024 and 2025. A larger sample would have been ideal, but even this small set required several hours of computation on my laptop. All of these parameters are estimated with the fixed $\theta = 0.1$. Parameters were fit three times per day, once between the hours of 9–11, then 11–2, then 2–5, the goal being to characterize how the best parameters change throughout the day. My goal is to also use this dataset for additional data exploration and model training; in addition to the Hawkes parameters I computed:

- **liq_ratio**: Bid size divided by ask size (capped), a basic liquidity-pressure indicator.
- **OFI_inc**: Instantaneous order-flow imbalance from changes in best bid/ask prices and sizes.
- **OFI_cum_short**: Cumulative short-horizon OFI (last few seconds), measuring rapid flow imbalance.
- **OFI_cum_long**: Cumulative long-horizon OFI (longer window), capturing sustained buying/selling pressure.
- **QI**: Queue imbalance at the top of book, bid size divided by (bid size + ask size).
- **price_mov_derivative_s**: Slope of a linear fit to mid-price over the short window (5 seconds).
- **vol_s**: Short-window mid-price volatility (standard deviation of first differences).
- **momentum_s**: Short-window momentum, last mid-price minus mid-price a few seconds earlier.
- **price_mov_derivative_l**: Slope of a linear fit to mid-price over the long window (10 seconds).
- **vol_l**: Long-window mid-price volatility.
- **momentum_l**: Long-window momentum, last mid-price minus mid-price at the start of the window.
- **rv_5_ticks**: Realized volatility computed over the most recent 5 ticks.
- **spread**: Best ask price minus best bid price.
- **Next-horizon realized volatility**: 5-second and 10-second realized volatility used as volatility-model targets.
- **Future average mid-price targets**: Average mid-price over the next 5 and 10 seconds minus the current mid-price, used for the price-direction model.

5.1 Hawkes Results

Below shows the baseline intensity estimates.

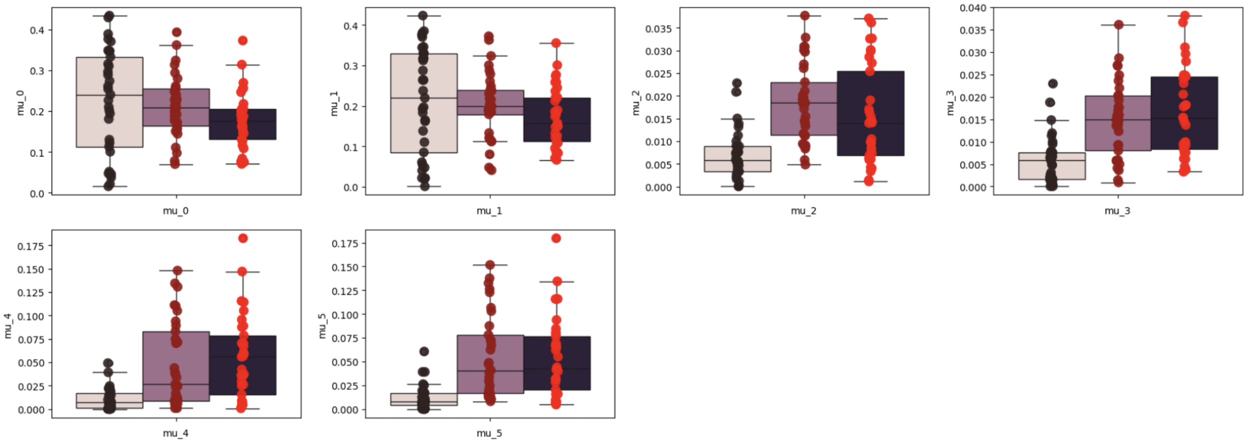

Figure 1: Survey results baseline intensities.

The full box plots are shown in the GitHub file `survey_parameters.png`. For μ_0 and μ_1 (MU and MD baseline intensities) we can see generally mornings are more active, then a decrease throughout the day. This is expected: early mornings have a flurry of trading activity. We can also start to see a core problem emerge: the "best" parameter has very high variance between days. We see similar instability in all parameters.

We will take the average of all these parameters as our baseline moving forward. When we talk about Hawkes intensities it is using the average values derived from this survey. Baseline values:

$$\mu = \begin{bmatrix} 0.162854 \\ 0.167923 \\ 0.010609 \\ 0.010018 \\ 0.029403 \\ 0.030933 \end{bmatrix} \quad A = \begin{bmatrix} 0.232029 & 0.196443 & 0.003183 & 0.002989 & 0.010576 & 0.014131 \\ 0.246073 & 0.238171 & 0.007875 & 0.005970 & 0.008947 & 0.007436 \\ 0.076636 & 0.050834 & 0.123685 & 0.237787 & 0.063663 & 0.099735 \\ 0.044117 & 0.055406 & 0.261568 & 0.169705 & 0.050023 & 0.095065 \\ 0.046572 & 0.043708 & 0.033750 & 0.022529 & 0.162810 & 0.315138 \\ 0.047246 & 0.033829 & 0.029265 & 0.023071 & 0.408157 & 0.229749 \end{bmatrix} \tag{60}$$

5.2 Residuals

As mentioned in the theory section, the residuals are the integrated intensities between arrival times, and they should follow an Exp(1) distribution by the time-rescaling theorem. Looking just at the λ_0 intensities for our survey:

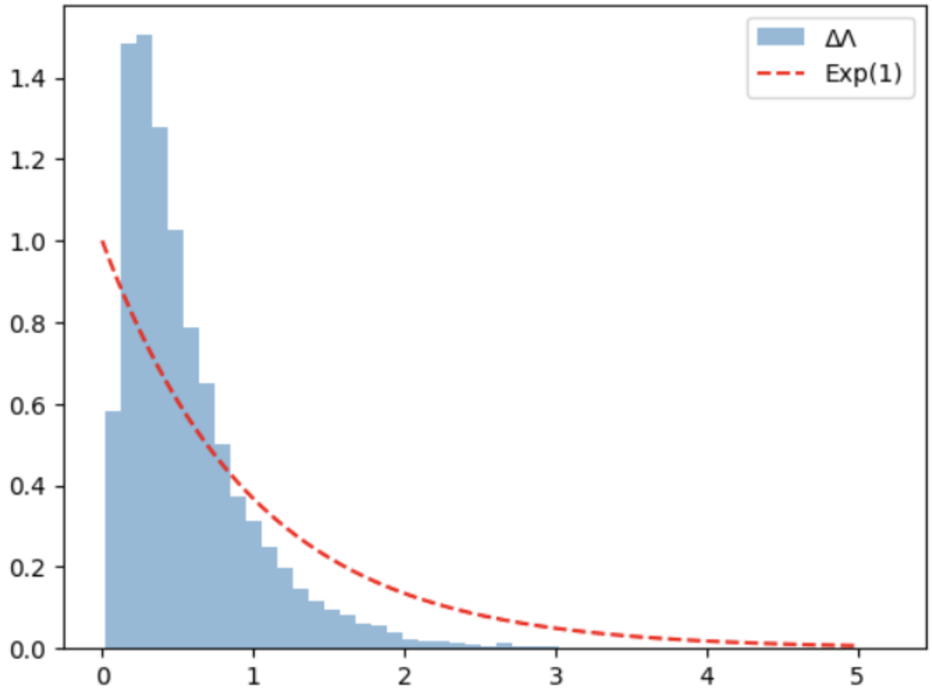

Figure 2: Residuals for λ_0.

We can see it's okay but not great; we could use a K-test to quantify it a bit more if we wanted. The average value is around 0.6, indicating we are systematically slow. Let's now look at the residuals with the residual compensator applied:

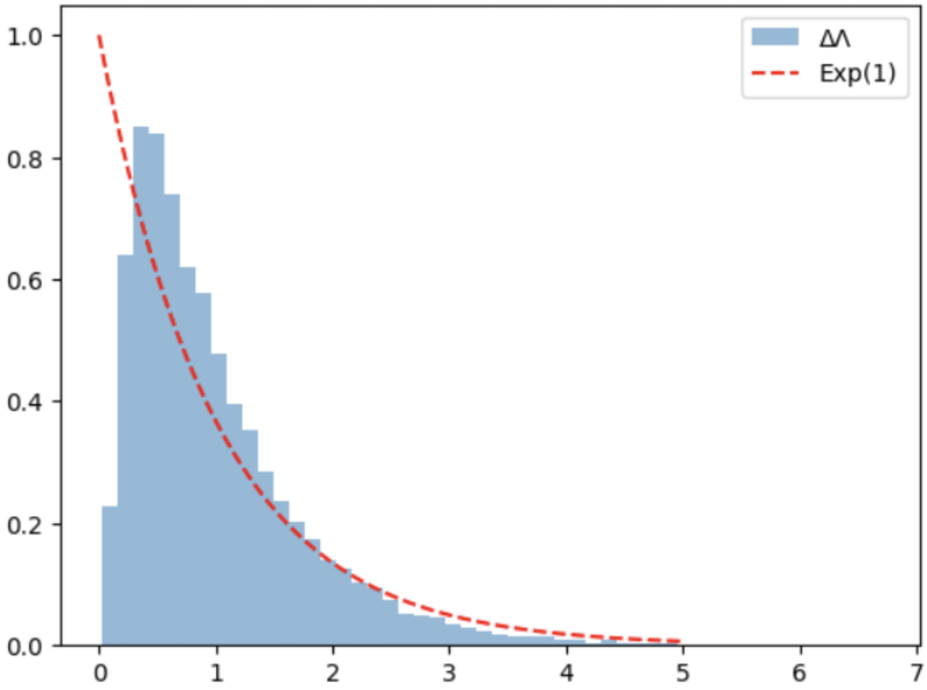

Figure 3: One-step residuals for λ_0 with compensator.

Not perfect but much better: the average is now at 1. These results are consistent across intensities.

5.3 Discussion

The key hurdle with practically using Hawkes processes in noisy systems is there is no single set of good parameters. These results are seen in our survey, where the best baseline rates and infectivity matrix can change wildly within days or between days. There are a number of solutions to this:

- Regime characterization: Split parameters by time of day and volatility; switch between regimes.
- More advanced kernels: Use a more adaptive and flexible kernel and set of parameters.
- Adjust parameters based on recent market conditions.

I have effectively chosen the last route: we aren't changing the parameters directly, but we rescale the intensities based on the previous observed error, and this is somewhat effective at making our Hawkes process more adaptive.
Going forward, however, I would recommend looking into Bayesian methods like Kalman filters or more advanced Bayesian filtering methods to update the parameters themselves in

real time. This is a more well-explored route and more interpretable; you also gain more control over the gain and rate of adaptation. A system where the residuals are used to feed a Kalman filter gain would be really powerful.
Moving forward we are going to use the compensated values as inputs into the ML models.

5.3.1 Correlation

First let's look at the correlation between parameters:

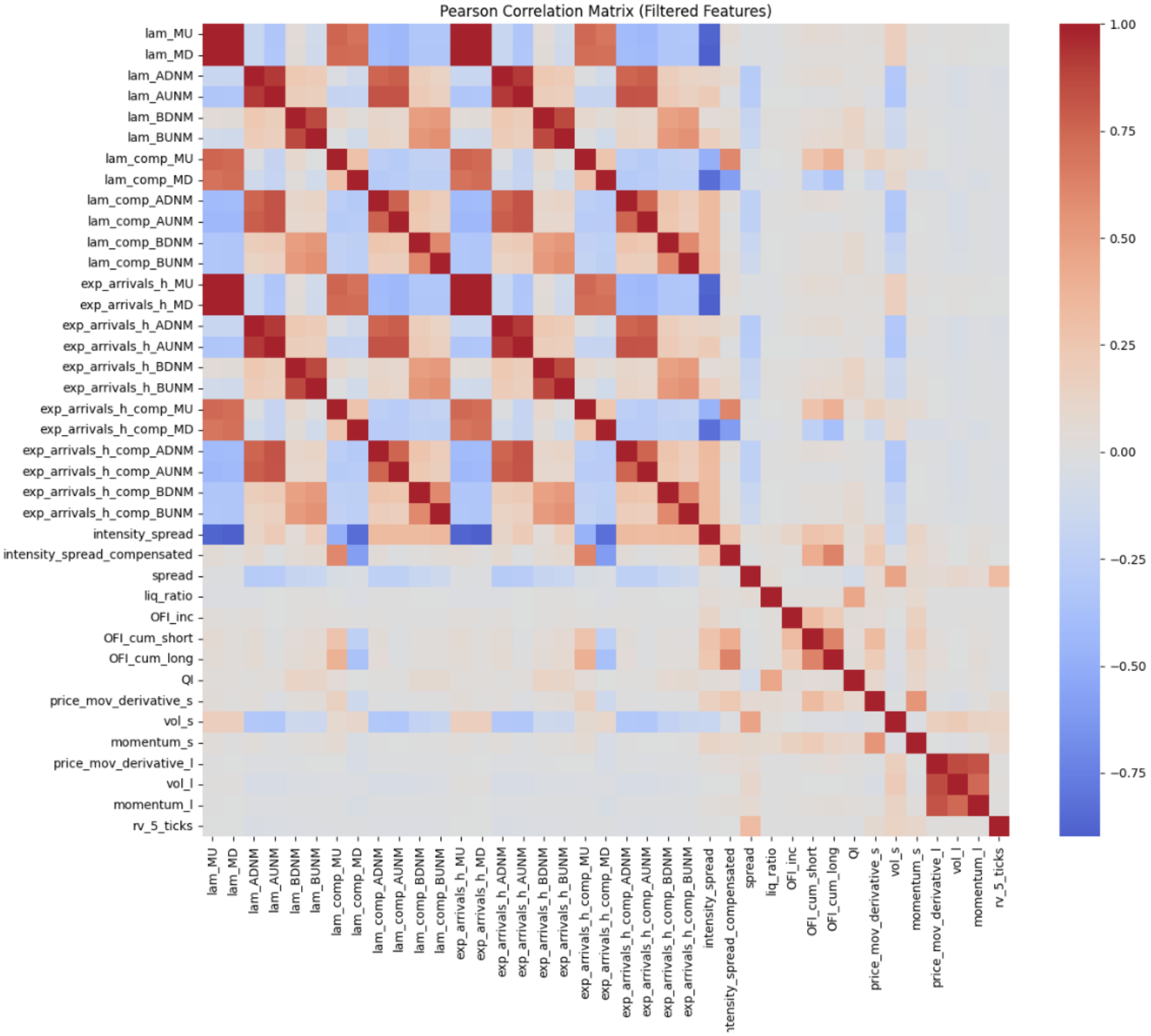

Figure 4: Pearson correlation for Hawkes parameters.

We can see the Hawkes parameters have significant correlation with each other. In general this isn't surprising: the raw value, compensated value, and forecast value all differ by a constant. In general I will use the compensated value when picking between different correlated variables. However, it is interesting to note that MU is highly correlated with MD. This brings us to a core, but unfortunate, point: since MU and MD happen frequently and dominate most market movements, the Hawkes intensities have limited predictive power for price direction; instead they track more closely with overall volatility and market activity. Hawkes intensities are useful for describing current market conditions; however, by themselves they have limited forecasting power.

Perhaps reweighting the events by magnitude, or looking at market activity at a finer scale (not the 1-second aggregate) could yield some more predictive power.

Another idea is that the secondary events ADNM, AUNM, BDNM, BUNM could signal price jumps. This is probably true, but the 1-second aggregate seems to erase this effect; the market simply moves too quickly.

Below shows a trading day with the MU compensated intensity vs the price off open; we can see it follows the price swings instead of predicting them:

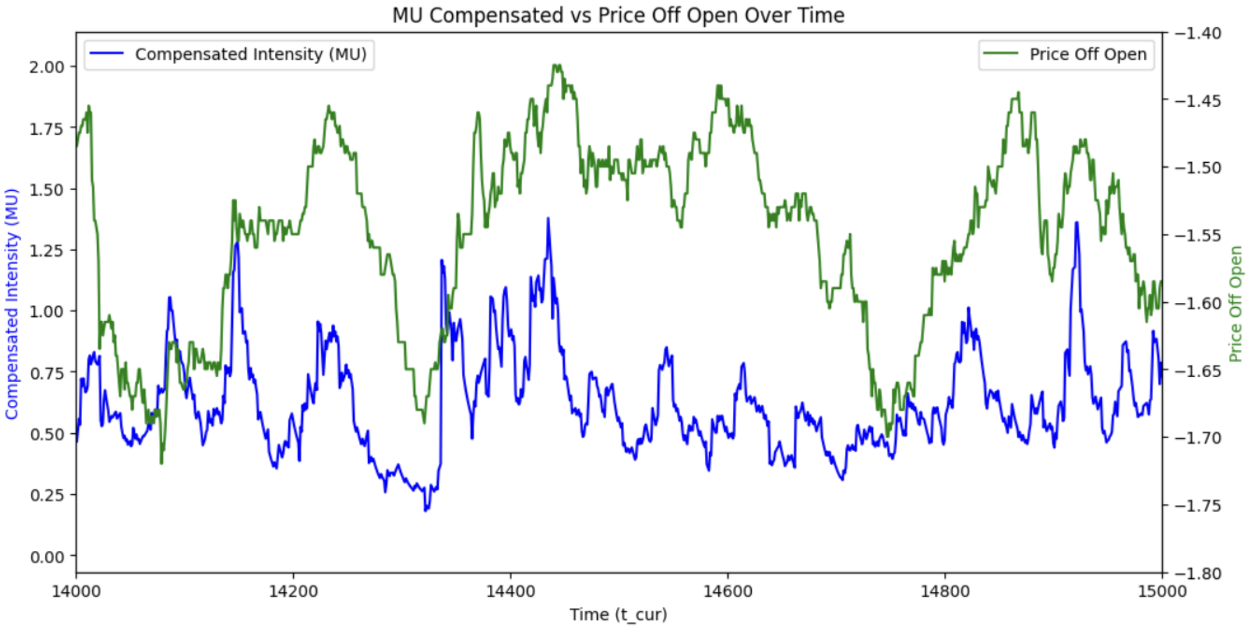

Figure 5: MU intensity vs price off open.

6 Short-Term Volatility Results

6.1 Expected Events vs Realized Volatility

Our core argument is that forecasted event count could be used to predict short-term (5s) realized volatility. So let's look at a plot of forecasting MU events (price jumps) in the next 5 seconds vs what the realized volatility actually was. The data is very noisy, so I chose to use binning and compare the expected arrivals vs the mean of RV_5:

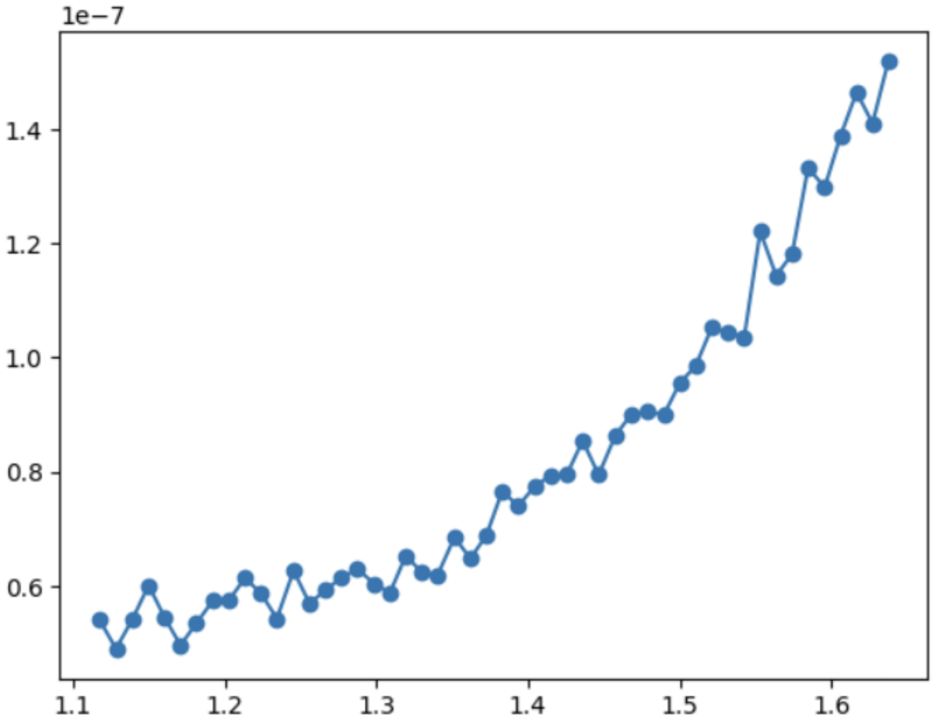

Figure 6: Mean RV_5 vs expected arrivals (MU).

This is a very encouraging result: we can see that on the macro level higher expected arrival correlates strongly with average realized volatility. Interestingly, ADNM volatility has the opposite correlation:

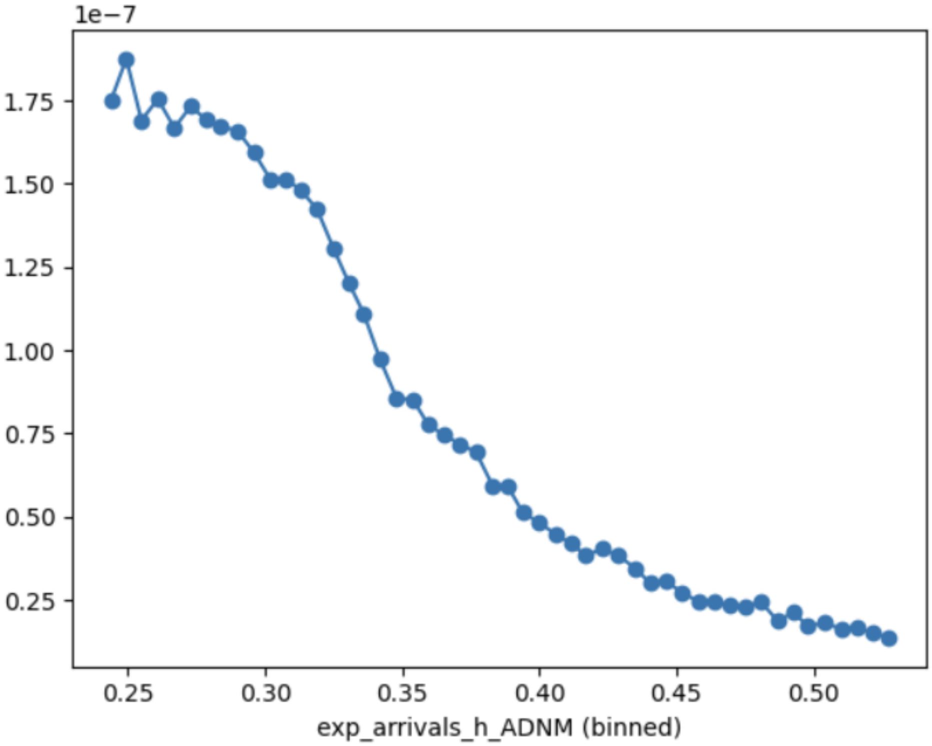

Figure 7: Mean RV_5 vs expected arrivals (ADNM).

We can see that the forecasted Hawkes intensities do carry some predictive power for realized volatility.

6.2 Forecasting Model

With this in mind, let's build a model to forecast the realized volatility. To smooth out some of the noise we will be predicting Y, which is

$$Y = \log(RV_5 + 1e - 9). \tag{61}$$

I started with this set of features (none of them correlated with another feature with Pearson's > 0.85):

Hawkes-derived features	**Non-Hawkes features**
exp_arrivals_h_MU	spread
exp_arrivals_h_ADNM	liq_ratio
exp_arrivals_h_BDNM	OFI_inc
exp_arrivals_h_comp_MU	OFI_cum_short
exp_arrivals_h_comp_MD	OFI_cum_long
exp_arrivals_h_comp_BDNM	QI
exp_arrivals_h_comp_BUNM	price_mov_derivative_s
intensity_spread_compensated	vol_s
	momentum_s
	price_mov_derivative_l

Table 1: Hawkes-derived and non-Hawkes feature groups.

I trained two models, one with the combined feature set, and one with no Hawkes features. Models were scored on R^2 and MSE, using 6-fold CV. CV was based on the day (no data leaking with the averaging). All models used an XGB Boost regressor.

Model	**Average R^2 (Std)**	**Average MSE (Std)**
Combined model	0.3152 (0.1033)	1.9584 (0.9151)
Non-Hawkes model	0.2904 (0.0882)	2.0080 (0.8754)

Table 2: Performance comparison between combined and non-Hawkes models.

And for the combined model the feature importance through permutation testing:

Feature	Importance Mean	Importance Std
vol_s	0.295099	0.001665
spread	0.095633	0.000473
exp_arrivals_h_comp_BDNM	0.032058	0.000462
exp_arrivals_h_comp_BUNM	0.009509	0.000228
price_mov_derivative_l	0.008906	0.000200
exp_arrivals_h_ADNM	0.007516	0.000162
exp_arrivals_h_BDNM	0.006629	0.000156
exp_arrivals_h_MU	0.004986	0.000122
momentum_s	0.003108	0.000165
exp_arrivals_h_comp_MD	0.002971	0.000098
exp_arrivals_h_comp_MU	0.001931	0.000041

Table 3: Permutation importance scores for the combined model, top 10.

We can see the combined model does score better in both metrics, but the results are inconclusive with such a high standard deviation. We can see from the feature importance, however, that the Hawkes features do add some predictive power.

As mentioned, we also trained a model that predicts the 20% and 80% confidence bounds. Below shows the realized volatility and the model predictions, and confidence bounds, on an out-of-sample day:

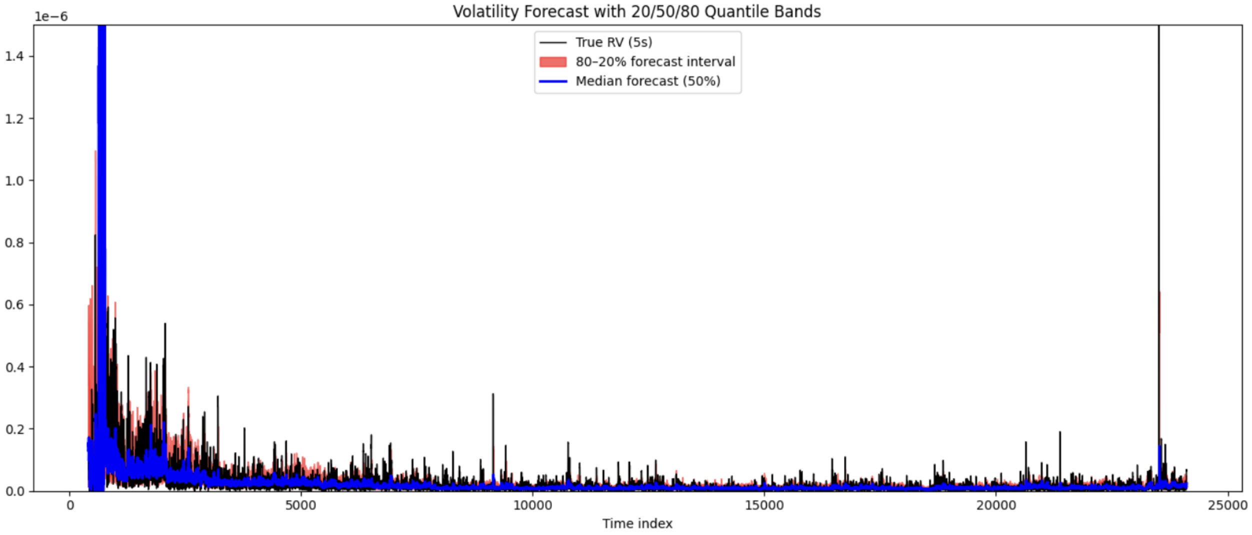

Figure 8: Volatility forecasting, combined model.

For such a noisy target, it's honestly not bad. However, it's unclear if the Hawkes intensities are adding a meaningful signal here; the model for the most part is just predicting off the previously observed volatility, with maybe some adjustments from the Hawkes features.

7 Price Direction Prediction Results

Our goal is now to predict the direction of the mid price over the next 5 seconds. To be explicit our goal is to classify Y such that

$$Z = \text{average mid price over next 5 seconds } - \text{current mid price}, \tag{62}$$

$$Y = \begin{cases} 1 & Z > \epsilon \\ -1 & Z < -\epsilon \\ 0 & \text{else.} \end{cases} \tag{63}$$

I broke this up into two separate classification problems:

- **Question 1:** Will there be a price jump, above some value ϵ? Is $Z \notin [-\epsilon, \epsilon]$? This problem is related to volatility forecasting, and where I expected the Hawkes parameters to be useful.
- **Question 2:** Given we know there will be a price jump, which direction will it be in? As previously discussed, Hawkes parameters (at least as used in this problem) don't carry much price-direction information; I expect the other features to carry this model.

The parameter ϵ also lets us control the "difficulty" of these problems, but also the usefulness of the model. A smaller ϵ means there is less extreme information available to predict the price direction (the features between an increase and decrease signal are closer together)—making Question 2 harder, and it makes the training set of Question 1 smaller. However, the smaller the price-direction jump we can predict, the more useful the model is (and hypothetically the more trades we can execute). After some experimenting I found any $\epsilon < 0.01\$$ impossible, with $\epsilon > 0.02\$$ having very good overall scores but few actionable trades. A middle value of $\epsilon = 0.015\$$ offered a good tradeoff.

All models are XGB Boost classifiers, we used 12-fold CV, and feature importance is the permutation importance.

7.1 Price Jump Model

This model is aiming to just answer Question 1. We used "no jump" meaning state 0, and "jump" as state 1. Similar to the volatility model I created three feature sets: one with the Hawkes features, one with no Hawkes features, and one with both Hawkes and non-Hawkes features.

Hawkes features	**Non-Hawkes features**
exp_arrivals_h_MU	spread
exp_arrivals_h_ADNM	liq_ratio
exp_arrivals_h_BDNM	OFI_inc
exp_arrivals_h_comp_MU	OFI_cum_short
exp_arrivals_h_comp_MD	OFI_cum_long
exp_arrivals_h_comp_BDNM	QI
exp_arrivals_h_comp_BUNM	price_mov_derivative_s
intensity_spread_compensated	vol_s
	momentum_s
	price_mov_derivative_l

Table 4: Hawkes features, non-Hawkes features, and prediction target.

For trading models, we have an option to simply "do nothing," as in we are uncertain of the price direction or the volatility and simply do not trade. We can control this with the conf_threshold parameter—if our model has a confidence above this threshold, then make a decision. For this model we will set it as 60%, or we need at least 60% confidence in jump / no jump to get a result; this is what the "average confident accuracy" refers to. "Average confident coverage" tells us what percent of predictions reach that threshold. Below shows our final scores with their standard deviation attached.

Metric	**Combined features**	**Hawkes-only**	**Non-Hawkes only**
Average accuracy	0.6864 (0.0388)	0.6463 (0.0347)	0.6834 (0.0411)
Average F1 (macro)	0.6816 (0.0345)	0.6405 (0.0313)	0.6790 (0.0374)
Average F1 (weighted)	0.6849 (0.0375)	0.6437 (0.0344)	0.6822 (0.0401)
Avg confident accuracy	0.7371 (0.0358)	0.6985 (0.0456)	0.7284 (0.0382)
Avg confident coverage	0.7265	0.6413	0.7354

Table 5: Classification performance for combined, Hawkes-only, and non-Hawkes feature subsets. Values shown as mean (standard deviation).

Note that accuracy between classes is basically equal. These results imply that the Hawkes model does have a signal on the price direction; however, this same signal can be derived from other less complicated features.

This is one of the main reasons these more complex models aren't used in practice: you can get similar information from simpler features that don't require all this fancy math. Moving on I will use the Hawkes-only model, since this project is specifically about the Hawkes process. Checking the permutation feature importance:

Feature	**Importance Mean**	**Importance Std**
exp_arrivals_h_ADNM	0.066148	0.000666
exp_arrivals_h_comp_BDNM	0.028866	0.000738
exp_arrivals_h_MU	0.015291	0.000452
exp_arrivals_h_comp_BUNM	0.007858	0.000507
exp_arrivals_h_BDNM	0.006272	0.000501
exp_arrivals_h_comp_MD	0.002330	0.000270
intensity_spread_compensated	0.001425	0.000241
exp_arrivals_h_comp_MU	0.001232	0.000241

Table 6: Permutation importance of Hawkes-only features.

This backs up an earlier point: the secondary events do have predictive power (e.g., ADNM means the ask size is decreasing, making a price change more likely). I expect that at a smaller time scale this phenomenon would be more pronounced and easier to capitalize on.

7.2 Price Direction Model

Identical setup as above, except we are aiming to predict price increase or decrease $(1, -1)$, filtering out all price jumps $< \epsilon$.

Metric	**Combined features**	**Non-Hawkes only**	**Hawkes only**
Average accuracy	0.6226 (0.0446)	0.6224 (0.0435)	0.5127 (0.0066)
Average F1 (macro)	0.6225 (0.0447)	0.6222 (0.0435)	0.5102 (0.0084)
Average F1 (weighted)	0.6224 (0.0448)	0.6221 (0.0436)	0.5099 (0.0092)
Avg confident accuracy	0.7417 (0.0634)	0.7392 (0.0725)	0.5471 (0.0174)
Avg confident coverage	0.3979	0.3900	0.0463

Table 7: Classification results comparing combined, non-Hawkes, and Hawkes-only feature subsets. Values shown as mean (standard deviation).

This aligns with expectations. As predicted, the Hawkes features have little power in predicting the direction of the move (the pure Hawkes model is basically just guessing). Looking at the top 10 features for the combined model:

Feature	Importance Mean	Importance Std
momentum_s	0.036780	0.001315
spread	0.036424	0.000979
liq_ratio	0.021897	0.000648
OFI_inc	0.017031	0.000978
exp_arrivals_h_comp_BUNM	0.008617	0.000683
vol_s	0.007532	0.000692
exp_arrivals_h_comp_BDNM	0.007252	0.000469
exp_arrivals_h_BDNM	0.006058	0.000627
price_mov_derivative_l	0.005144	0.000370
exp_arrivals_h_comp_MU	0.005016	0.000387

Table 8: Permutation importance for the combined feature model.

This model is basically just momentum trading, with some extra information from the current spread, liquidity ratio, order-flow imbalance, and Hawkes features.

8 Conclusion

A few clear conclusions

- Hawkes forcasted intensities correlated with average realized volatility
- Hawkes parameters can be used to predict price jumps
- The derived compensator does make the process more reactive to market conditions
- Parameter estimation is the key issue in Hawkes parameter utilit

However Hawkes parameters (as derived) do not carry significant predictive power on price movement direction, or if they do the 1 second aggregation erases it. A number of improvements could significantly improve this project

- Use MBO-1 dataset, couple this with a bayesian compensator to create accurate hawkes intensities and real time parameter adjustments
- Scale Hawkes parameters by magnitude of price jumps
- Experiment with forecasting different time horizons, price jumps on intervals ¡ 5 seconds could be easier to predict (if you can act fast enough to capitalize on them)

So do we have a tradable method? No. BBO-1s information is released in a 1-2 second delay, so any edge calculated by the price direction model is washed away before you can actionably trade. However I think there is the good bones here of a HFT trading method (especially if the above improvements are implemented.) I personally lack the funds or fiber optic cable running to the NASDAQ to finalize it. This model is also based around predicting the mid price, but you buy at the bid and sell at the ask - so just knowing the mid price direction is not sufficient to making a profitable trade.