Predicting Ultra-High-Frequency Volatility from Limit Order Book Data

Group 07
University of Sydney

1 Project Overview

Market makers such as Optiver rely on accurate forecasts of future price variability to quote options, allocate risk capital and manage inventory. We are provided with one–second limit-order-book (LOB) snapshots for 127 equities, grouped into ten-minute buckets identified by `time_id`. Each snapshot records the two best bid and ask prices together with their sizes. Our goal is to predict the *future realised volatility*—measured over the next 30-second window—using information that is available up to the current window. Initial experiments carried out in the practice lab employed classical econometric baselines; the full project extends both the data scope (all stocks, the complete time span) and the modelling arsenal (modern deep-learning architectures).

2 Data Description

Every CSV file `stock_`$\langle id \rangle$`.csv` contains roughly six hundred rows per ten-minute bucket, one for each second. The essential numerical fields are the best two bid/ask prices (`bid_price1`, `ask_price1`, ...) and their corresponding sizes; an additional column `seconds_in_bucket` counts the seconds within the bucket. After parsing, we augment the data with engineered features (spread, micro-price, imbalance, entropy, weighted-average price) and with the realised-volatility label σ^2_{RV} shifted forward so that each 30-second window is paired with the volatility of the *next* window.

Because the full corpus contains approximately 167 million rows, all transformations must be vectorised and streamed out to efficient storage (Parquet) before model fitting.

3 Feature-Engineering Pipeline

We proceed in six steps.

1. **Micro-price and spread:** $\mathrm{mid}_t = (\mathrm{bid\_price1}_t + \mathrm{ask\_price1}_t)/2$ and $\mathrm{spread}_t = \mathrm{ask\_price1}_t - \mathrm{bid\_price1}_t$.

2. **Liquidity and pressure metrics:** order-book imbalance, book pressure, depth entropy and a two-level weighted-average price capture queue lengths and price pressure.

3. **Return statistics:** log-returns on mid-price and on WAP, as well as bi-power variation [1] to proxy jump-robust volatility.

4. **Window aggregation:** the per-second series are collapsed into non-overlapping 30-second windows within each `time_id`, computing means, counts and other summary statistics.

5. **Realised volatility computation:** inside each window we sum squared log-returns, $RV_t = \sum r^2_{t,i}$, and then shift the series so that feature vector x_t is paired with label RV_{t+1}.

6. **Lag and rolling features:** for example a 30-window rolling mean of volatility (`rolling_vol_30`) provides long-memory context.

4 Modelling Framework

Three classical baselines have already been validated on a subset of `stock_1`.

a) **Linear weighted least squares (WLS):** volatility is regressed on lagged window averages of price, order count and spread, with weights proportional to the inverse conditional variance.

b) **HAR/HAV-RV:** the heterogeneous autoregressive family models daily, weekly and monthly realised volatilities to capture long-range dependence.

c) **ARMA(1,1)–GARCH(1,1):** an ARMA process models returns while GARCH handles time-varying variance; scenario simulation (1,000 paths) delivers the volatility forecast.

Planned extensions move beyond linearity and single-asset modelling. Sequence models such as LSTM, Temporal Fusion Transformers and N-Beats will learn non-linear temporal dependencies, while graph neural networks will exploit cross-asset relations derived from sector membership or historical return correlations.

5 Evaluation Methodology

Forecasts are judged primarily by the quadratic likelihood error QLIKE and the mean squared error MSE:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \qquad \mathrm{QLIKE} = \frac{y_i}{\hat{y}_i} - \ln\frac{y_i}{\hat{y}_i} - 1. \tag{1}$$

Directional-accuracy and the Pearson correlation between realised and predicted volatilities serve as secondary diagnostics. Validation uses an expanding-window back-test that preserves chronological order and stratifies folds so that every split retains all 127 equities.

6 Deliverables

Three artefacts will be submitted.

1. *Reproducible codebase:* modular Python scripts and notebooks, a command-line interface for batch inference and evaluation, and continuous-integration checks for data-leakage.

2. *Analytic report:* a written document (this LaTeX) detailing data preparation, methodology, results with interpretation and a frank discussion of limitations.

3. *Communication product:* either a two-minute animated explainer or an interactive dashboard that demonstrates how volatility is computed, how the best model learns, and where the predictions might fail in practice.

7 Stretch Goals and Research Directions

Beyond the core deliverables we will explore: sensitivity of HAR-RV coefficients and GARCH parameters to forecast error; unsupervised clustering of stocks into volatility regimes and subsequent transfer-learning; and an application study where predicted volatilities feed into Black–Scholes option pricing, benchmarked against market quotes.

References

[1] O. E. Barndorff-Nielsen and N. Shephard. Power and bipower variation with stochastic volatility and jumps. *Journal of Financial Econometrics*, 2004.