Final Submission: Outline of Modeling Pipeline

Aryanil Kundu

Overview

This report summarizes the end–to–end pipeline used in the final submission. The notebook learns the features by applying models to the train-test split of the whole training data and cross-validate it. Then it applies the best learned model to the actual training and testing data.

Data and Preprocessing

- **Targets & Features:** The target variable (`label`) is predicted from market microstructure and volatility features (order-book levels, bandwidth measures, GARCH/Kalman volatilities, PCA factors, and a time-based spline feature).
- **Cleaning:** Missing values filled (e.g., zeros for microstructure gaps); sanity checks for NaN/∞; optional clipping at extreme quantiles to curb outliers.
- **Scaling:** Robust scaling (for heavy tails) and/or MinMax scaling (for sequence models) applied consistently using training statistics only.

Feature Engineering (Leakage-Safe)

Features are recomputed *inside each fold* using only the fold's training window to prevent look-ahead.

- **Volatility Proxies:**
  - GARCH(1,1) conditional variance on log-returns; out-of-sample forecasts mapped to validation horizon.
  - Kalman-filtered latent variance from squared return observations; filtered state propagated online into validation.
  - Classical realized-volatility variants (e.g., Parkinson, Garman–Klass, Yang–Zhang) as inputs.
- **Dimensionality Reduction:** PCA on (i) ETH vol features to form `eth_vol_pca_1`, (ii) cross-asset vol blocks to form `cross_asset_vol_pca_1`, and (iii) cross-asset Kalman forecasts to form `cross_asset_kalman_forecast_pca_1`.
- **Time Structure:** A cubic spline on elapsed time provides a smooth trend component (`spline_feature`).
- **Bandwidth:** Bollinger bandwidth as defined by (Upper Bollinger Band - Lower Bollinger Band)/Middle Band was used.

These features have been finalized by considering the correlation matrix with 'label' and the feature importance plot in the XGBoost model.

Modeling

- **Tree Model (XGBoost):** Regressor with `reg:squarederror`, large tree budget with small learning rate, subsampling and column subsampling for generalization. Trained per fold on the final feature set.
- **Sequence Models.**

- *LSTM:* Two-layer LSTM with `tanh`, dropout regularization, gradient clipping and low learning rate to avoid exploding gradients; sequences built with a fixed lookback window.
- *Transformer:* Stacked multi-head self-attention blocks with feed-forward layers, layer normalization, dropout, and global average pooling for regression output.

Out of all the three models xgboost has shown comparatively better results and also for high frequency market data running transformer or LSTM may be time inefficient.

Evaluation: Time-Series Cross-Validation

A simple train/test split can be unstable. We employ K-fold chronological splits (e.g., $K{=}3$) via `TimeSeriesSplit`: each fold trains on the past and validates on the immediately following segment. For each fold k, we record RMSE_k and R^2_k on the validation horizon; the final estimate reports mean (and optionally standard deviation) across folds:

$$\overline{\text{RMSE}} = \frac{1}{K}\sum_{k=1}^{K}\text{RMSE}_k, \qquad \overline{R^2} = \frac{1}{K}\sum_{k=1}^{K}R^2_k.$$

Fold-level plots (predicted vs. actual) provide qualitative diagnostics.

Trading Strategy

A trading strategy using 10 second ahead predicted IV and Order Flow Imbalance (measures buying/ selling pressure) has been described at the last of the notebook.

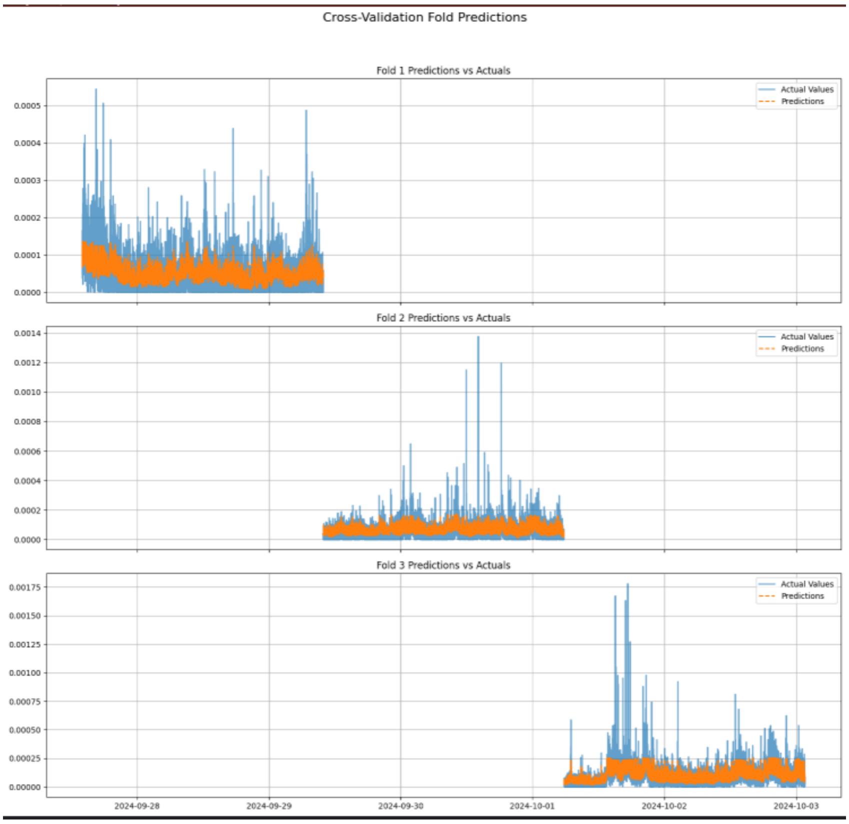

Figure 1: Cross Validation Actual vs Predicted IV plots