High-Frequency Volatility Forecasting and Risk-Managed Trading Strategies in Cryptocurrency Markets: A Gradient Boosting Approach

Victor Hsu, Mackenzie Qu, Maria Kravchenko

Abstract

We present a comprehensive machine learning framework for predicting realized volatility in cryptocurrency markets using high-frequency data. Our approach combines microstructure features from order book dynamics, trade flow imbalances, and price movements to forecast 60-minute ahead realized volatility of Bitcoin futures. Using a LightGBM regressor optimized through walk-forward cross-validation, we achieve a strong out-of-sample Information Coefficient (IC) of 0.80. Furthermore, we implement a volatility-targeting strategy based on these forecasts. Backtesting results demonstrate that the volatility-managed portfolio yields superior risk-adjusted returns (Sharpe Ratio: 1.78) and significantly reduced maximum drawdown (-17.09%) compared to a passive Buy & Hold benchmark (-29.55%).

1 Introduction

Volatility forecasting is a cornerstone of quantitative finance, essential for risk management, option pricing, and portfolio construction. In cryptocurrency markets, characterized by regime shifts and extreme kurtosis, accurate volatility prediction is particularly challenging. While traditional econometric models (e.g., GARCH) capture autoregressive properties, they often fail to incorporate the rich information embedded in market microstructure.

This study proposes a supervised learning approach using Gradient Boosting Decision Trees (LightGBM) to forecast 1-hour ahead realized volatility for BTCUSDT perpetual futures. We hypothesize that incorporating Order Book Imbalance (OBI), Order Flow Imbalance (OFI), and trade flow dynamics enhances predictive power beyond simple autoregressive returns. Finally, we leverage these predictions to construct a dynamic volatility-targeting strategy that adjusts position sizing to stabilize portfolio variance.

2 Methodology

2.1 Data

The dataset comprises high-frequency market data for Binance BTCUSDT Futures, spanning the period from January 1, 2023, to December 31, 2024. The raw data consists of three distinct streams:

- **Level 1 Data**: Best bid/ask prices, spreads, and tick-level volatility aggregated to 1-minute bars
- **Order Book Data**: Limit order book depth at various basis point levels from the mid-price (1, 3, 10, 30, 100 bps)
- **Trade Data**: Executed trades including volume, direction (buy/sell, taker/maker), and trade counts

All data streams were resampled to 1-minute intervals. Time series alignment was performed by taking the intersection of timestamps across all dataframes to ensure data integrity, resulting in a continuous timeline indexed by minute.

For missing observation due to various reasons, we implement a weighted training approach where carried-over observations receive reduced weight

(0.5) compared to original data (1.0), while completely missing observations are removed. This preserves temporal continuity while appropriately downweighting less reliable data points.

2.2 Feature Engineering

We engineered a diverse set of features capturing price dynamics, liquidity, and market sentiment over multiple lookback windows $\tau \in \{1, 5, 15, 30, 60, 120\}$ minutes, detailed in Table 1. The target variable is the annualized 1-hour forward realized volatility.

2.3 Feature Selection

To mitigate the curse of dimensionality and remove noise, we employed a sequential, two-step feature selection pipeline on the training data.

A. Train-Validation-Test Split

We utilized a Walk-Forward Validation approach with three folds for model training and validation to respect the temporal structure of financial data. A final Out-of-Sample (OOS) test period from July 1, 2024, to December 31, 2024, was reserved for final performance evaluation.

B. Filtering & Pruning

The selection process involves two main stages: a predictive filter and a redundancy pruning step.

i) 1. Predictive Filtering (Information Coefficient) Features are initially filtered based on their predictive power, measured by the Spearman Rank Correlation, or Information Coefficient (IC), between the feature X and the target variable Y:

$$\rho_S(X, Y) = \frac{\mathrm{cov}(R_X, R_Y)}{\sigma_{R_X}\sigma_{R_Y}} \tag{1}$$

where R_X and R_Y are the rank variables of X and Y. We retain only features X_i that satisfy an absolute IC threshold, $|\rho_S(X_i, Y)| \geq 0.10$. This step ensures all remaining features possess a statistically and economically significant monotonic relationship with the target.

ii) 2. Collinearity Pruning (Sequential Selection) The remaining features often exhibit high multicollinearity, which can destabilize tree-based models and obfuscate feature importance. We use the Pearson Correlation (ρ_P) to measure linear redundancy between features X_i and X_j:

$$\rho_P(X_i, X_j) = \frac{\sum(X_i - \mu_i)(X_j - \mu_j)}{\sqrt{\sum(X_i - \mu_i)^2 \sum(X_j - \mu_j)^2}} \tag{2}$$

We apply a greedy, sequential selection algorithm [**featureselect**]:

1. Rank all filtered features X in descending order based on their absolute Spearman IC 2.
2. Initialize the set of selected features, $S = \emptyset$.
3. Iterate through the ranked features X_i. If X_i is highly correlated ($|\rho_P(X_i, X_j)| \geq 0.70$) with any feature $X_j \in S$, it is discarded. Otherwise, X_i is added to S.

This process ensures that the final feature set, S, is diverse while prioritizing those features with the strongest individual predictive capacity (highest IC).

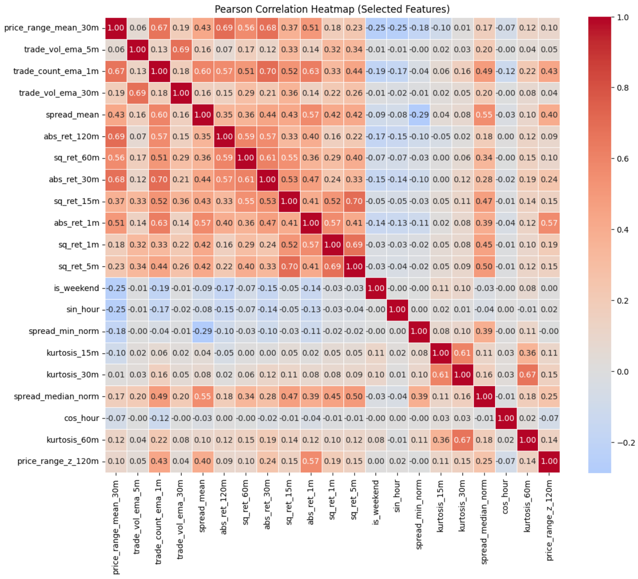

Figure 1: *Feature Correlation Matrix (Post-Pruning)*

2.4 Modeling Framework

We utilized the LightGBM regressor [**ke2017**], a highly efficient gradient boosting framework capable of handling large-scale datasets. The model is built as an additive expansion of K decision trees, $h_k(X)$, where X is the input feature vector:

$$F_K(X) = \sum_{k=1}^{K} \gamma_k h_k(X) \tag{3}$$

Table 1: *Feature Construction Definitions*

Feature Group	Feature Name(s)	Mathematical Formulation / Description	Time Scales (τ)
Target (y_t)	Target_RV$_{60m}$	$\sigma_{t,\text{fwd}} = \sqrt{\frac{A}{h}\sum_{i=1}^{h} r_{t+i}^2}$, where $A = 525,600$ (min/year) and $h = 60$.	$60m$
Return-Based	log_ret$_\tau$, abs_ret$_\tau$, sq_ret$_\tau$	log_ret$_\tau$ = $\log(P_t/P_{t-\tau})$	$\forall\tau \in \{1, 5, 15, 30, 60, 120\}$
Realized Volatility	RV$_\tau$	$\text{RV}_\tau = \text{std}_\tau(r_t)\cdot\sqrt{525600}$	$\forall\tau \in \{1, 5, 15, 30, 60, 120\}$
Distribution Shape	skew$_\tau$, kurtosis$_\tau$	Third and fourth standardized moments of returns.	$\tau \geq 3$
Order Book Imbalance (OBI)	OBI$_b$	$\frac{V_b^{\text{bid}} - V_b^{\text{ask}}}{V_b^{\text{bid}} + V_b^{\text{ask}}}$	$b \in \{1, 3, 10, 30, 100\}$ bps
Order Flow Imbalance (OFI)	OFI$_{b,t}$	$\frac{\Delta V_{b,t}^{\text{bid}} - \Delta V_{b,t}^{\text{ask}}}{\Delta V_{b,t}^{\text{bid}} + \Delta V_{b,t}^{\text{ask}}}$	$b \in \{1, 3, 10, 30, 100\}$ bps
Trade Flow	volume_imb, count_imb, EMA	Imbalances based on traded volume/count (buy vs sell), plus Exponential Moving Averages.	$1m$ base, various EMA scales
Microstructure Noise	spread_norm, price_range	Normalized bid-ask spread and normalized high-low price range.	$1m$
Seasonality	Time-based indicators	Cyclical encodings of hour/minute, DOW indicators.	N/A

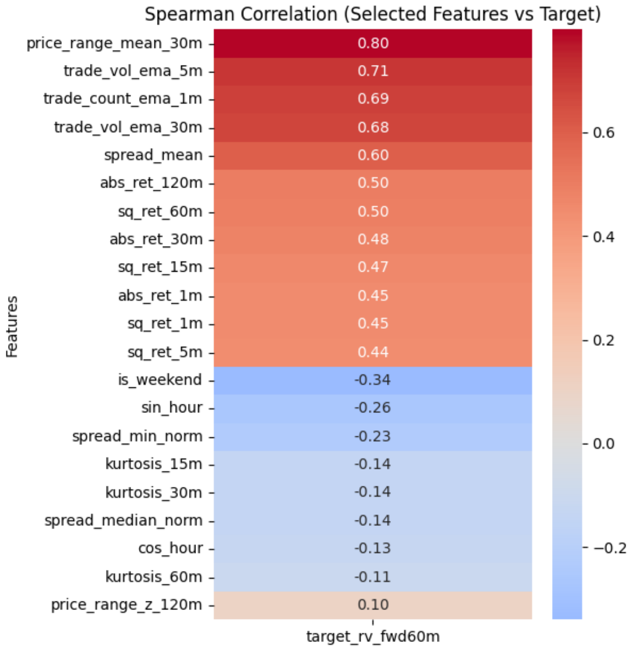

Figure 2: *Spearman Correlation Matrix (Post-Pruning)*

The model minimizes a differentiable loss function $L(y, F(X))$ by iteratively fitting each new tree $h_k(X)$ to the negative gradient (pseudo-residuals) of the loss function with respect to the current model $F_{k-1}(X)$.

A. Loss Function

The loss function minimized during training was the Mean Squared Error (MSE), defined as:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (4)$$

where y_i is the observed realized volatility and $\hat{y}_i = F(X_i)$ is the predicted volatility for the i-th observation, and N is the total number of samples.

B. Evaluation Metric

While MSE guides the gradient optimization, the model selection and performance assessment relied on the Spearman Information Coefficient (IC). The IC, calculated as the rank correlation between predicted ($\hat{y}$) and actual (y) volatility, quantifies the model's ability to correctly rank the relative magnitude of future volatility regimes, which is critical for a volatility-targeting strategy:

$$\text{IC} = \rho_S(\hat{y}, y) = \frac{\text{cov}(\text{Rank}(\hat{y}), \text{Rank}(y))}{\sigma_{\text{Rank}(\hat{y})}\sigma_{\text{Rank}(y)}} \qquad (5)$$

Hyperparameters (number of leaves, learning rate, and $n\_estimators$) were tuned using the average IC across the three validation folds.

3 Results

The final model was trained on data prior to July 2024 and evaluated on the subsequent test set. The model demonstrated robust predictive capability with the following metrics:

- **RMSE:** 0.1996
- **MAE:** 0.1145
- **Spearman IC:** 0.8031

The high IC suggests the model is exceptionally good at ranking volatility regimes (i.e., distinguishing high volatility periods from low volatility periods), which is crucial for the trading strategy.

3.1 Trade strategy: Volatility Targeting

We implemented a Volatility-Managed Portfolio strategy [**moreira2017**]. The core premise is that asset returns inversely scaled by their volatility produce better risk-adjusted performance due to the "leverage effect" and clustering of variance. The strategy aims to maintain a constant level of

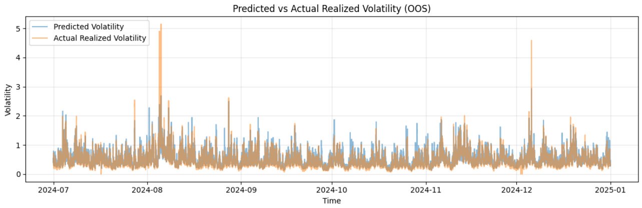

Figure 3: *Out-of-Sample Volatility Forecasts*

risk exposure (σ_{target}) by dynamically adjusting the position size (weight w_t) inversely proportional to the predicted volatility ($\hat{\sigma}_t$).

The weight of the asset at time t, w_t, is determined by:

$$w_t = \text{clip}\left(\frac{\sigma_{\text{target}}}{\hat{\sigma}_t}, w_{\min}, w_{\max}\right) \tag{6}$$

Where:

- σ_{target} is the median realized volatility of the pre-test period (approx. 0.3191).
- $\hat{\sigma}_t$ is the predicted volatility for the next period.
- $w_{\min} = 0.5$ and $w_{\max} = 2.0$ are leverage constraints to prevent excessive risk-taking.

3.2 Strategy backtest results

The strategy was backtested on the OOS period using 1-minute rebalancing logic (applied to the 1-hour forecast horizon).

Table 2: *Strategy Performance Comparison (OOS)*

Metric	**Buy & Hold**	**Vol-Managed Strategy**
Total PnL	0.4425	0.3078
Sharpe Ratio	1.6310	**1.7765**
Sortino Ratio	2.1726	**2.5935**
Max Drawdown	-29.55%	**-17.09%**
Annualized Volatility	54.02%	34.50%

While the raw PnL of the Buy & Hold strategy was higher due to the strong directional trend of BTC during the period, the Volatility-Managed strategy significantly improved the quality of returns. The Max Drawdown was reduced by over 12 percentage points, and the Sharpe Ratio increased from 1.63 to 1.78.

4 Limitations

While the volatility-managed portfolio strategy demonstrates superior risk-adjusted returns in backtesting, the following real-world implementation constraints and model limitations must be rigorously addressed:

1. **Transaction Costs and Market Slippage:** The current backtest assumes zero transaction costs and instant execution. In a high-frequency trading environment, the frequent rebalancing mandated by the volatility-managed strategy (w_t) would incur significant exchange fees and market slippage, particularly when adjusting large positions during periods of predicted high volatility (low liquidity). This could potentially erode the observed alpha.
2. **Execution Latency and Data Staleness:** The model uses 1-minute bar data for feature calculation and assumes that the prediction $\hat{\sigma}_t$ is immediately actionable for the next period's returns. Real-world systems face inevitable execution latency. The delay means that the predicted volatility $\hat{\sigma}_t$ might be based on stale market microstructure data by the time the position adjustment is executed, degrading predictive value.
3. **Model Stability and Feature Robustness:** The high out-of-sample IC suggests strong predictive power, but continuous monitoring of the model's stability is crucial. The model's reliance on highly granular microstructure features (OBI, OFI) may lead to performance degradation (concept drift) if market structures or trading behaviors shift.

5 Future Development

To transition this research into a robust, deployable trading strategy, the following enhancements are proposed:

1. **Cost-Agnostic Optimization:** Implement an optimization step to incorporate realistic transaction cost and slippage models. This involves solving for the optimal position size w_t^* by maximizing the utility function:

$$\begin{aligned} w_t^* = \arg\max_{w_t} [\mathbb{E}_t[r_{t+1} \cdot w_t] - \\ \frac{1}{2}\gamma \cdot \text{Var}_t[r_{t+1} \cdot w_t] - \\ C(w_t, w_{t-1})] \end{aligned} \tag{7}$$

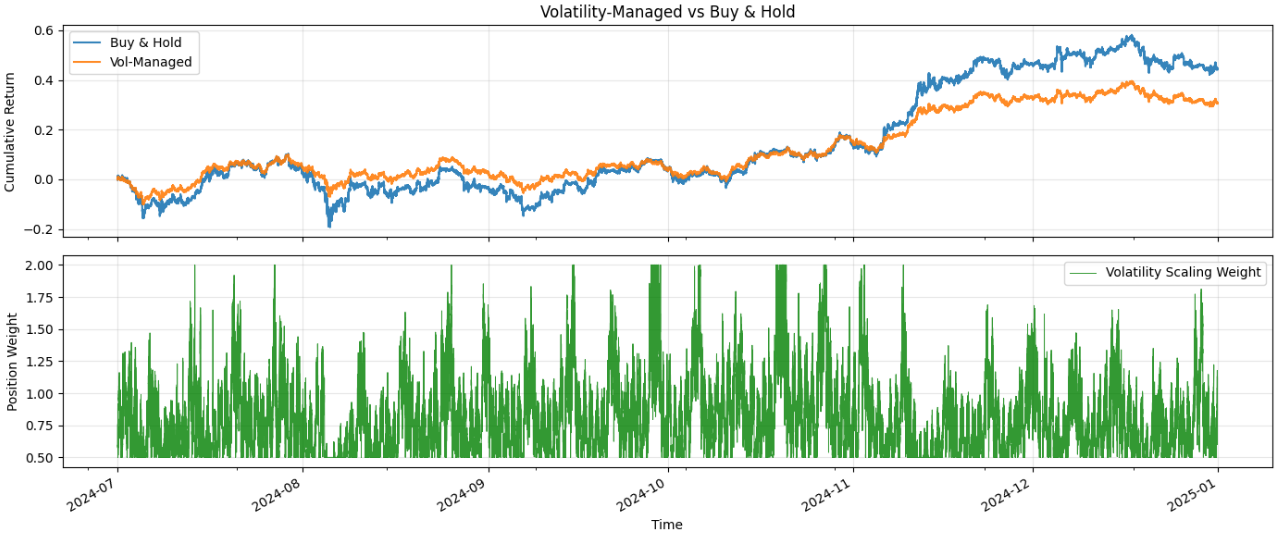

Figure 4: *Cumulative Returns of Volatility Targeting Strategy*

where $C(w_t, w_{t-1})$ is the change in transaction cost incurred by rebalancing from w_{t-1} to w_t, and γ is the investor's risk aversion coefficient. This will determine the cost-efficient rebalancing frequency. Future work should incorporate a cost function into the objective, optimizing the trading strategy's speed and magnitude to maximize the Sharpe Ratio net of transaction costs, following advanced dynamic portfolio principles [**GarleanuPedersen2013**]

2. **Advanced Feature Validation and Pruning:** Conduct more rigorous validation, including Out-of-Time (OOT) testing and utilizing **Permutation Feature Importance** to confirm feature causality and prevent reliance on spurious correlations. This process should also involve explicitly modeling the decay rate of feature importance.
3. **Dynamic Weight Management (Risk Parity):** Move beyond the fixed target volatility approach (σ_{target}). Instead, implement a Dynamic Risk Budgeting framework where the target volatility is itself a rolling, time-varying parameter calibrated to prevailing macroeconomic or sentiment conditions. This involves allocating capital based on the marginal risk contribution of the asset over time, potentially leveraging dynamic, time-consistent risk measures and deep learning techniques to adapt the risk profile based on changing market regimes [**Pesenti2025Risk**].
4. **Ensemble Modeling:** The standalone LightGBM model could be combined with traditional, theoretically grounded time-series models (e.g., GARCH or HAR-RV) to form a hybrid ensemble. Combining the non-linear predictive power of the Machine Learning model with the strong time-series properties of classic volatility models is often shown to yield a more stable and accurate forecast across different market conditions [**Kim2018Hybrid**].

$$\hat{\sigma}_{\text{final}} = \lambda \hat{\sigma}_{\text{LGBM}} + (1 - \lambda)\hat{\sigma}_{\text{HAR-RV}} \quad (8)$$

6 Summary

This research demonstrates that Gradient Boosting models, fed with granular order book and trade flow features, can forecast Bitcoin volatility with high accuracy (IC $\approx$ 0.80). By utilizing these forecasts to dynamically size positions, we constructed a Volatility-Managed strategy that successfully smooths equity curves. The strategy reduces downside risk (Max Drawdown -17.09%) and enhances risk-adjusted returns (Sharpe 1.78) compared to a passive holding strategy, validating the utility of machine learning in crypto-asset risk management.