Mixture Models for Bitcoin Volatility Forecasting

Auguste Darracq Paries
Department of Mathematics
ETH Zurich
Rämistrasse 101, 8092 Zürich
adarracq@ethz.ch

Yann Adorno
Department of Mathematics
ETH Zurich
Rämistrasse 101, 8092 Zürich
yadorno@ethz.ch

Laurent Martini–Bonnet
Department of Mathematics
ETH Zurich
Rämistrasse 101, 8092 Zürich
martinil@ethz.ch

Antoine Allain
Department of Mathematics
ETH Zurich
Rämistrasse 101, 8092 Zürich
aallain@ethz.ch

Abstract

Forecasting financial market volatility is a fundamental objective in quantitative finance. It plays a critical role in risk management, portfolio construction, derivative pricing, and regulatory compliance. In this work, we focus on the prediction of realized volatility for Bitcoin. The motivation for predicting volatility stems from several important financial applications. In risk management, volatility estimates are used to quantify exposure and rebalance portfolios under stress conditions (Fischer et al. [2021]). For example, portfolio managers may use predicted volatility to adjust leverage or to structure hedging strategies using derivatives. In algorithmic trading (Her-Jiun Sheu [2011]), volatility forecasts are central to strategies that seek to exploit periods of calm or turbulence—for instance, deploying high-frequency arbitrage only in low-volatility regimes, or engaging in momentum trades during highly volatile periods. Furthermore, in a regulatory context, accurate volatility projection supports the calculation of Value-at-Risk and Expected Shortfall (Stavros Degiannakis [2013]) , both of which are required by frameworks such as Basel III for banks and financial institutions.

Among the many assets where volatility modeling is useful, Bitcoin presents an especially challenging and informative case: it exhibits extraordinarily high volatility compared to traditional assets (Dirk G. Baur [2021]). For example, in 2021, Bitcoin experienced a decline of over 50% in a matter of weeks, falling from above 60,000 USD to below 30,000 USD. Such important fluctuations make it a difficult asset to model and an ideal candidate for developing robust volatility forecasting tools. Beyond its price behavior, Bitcoin is notable for being an emerging and speculative asset whose returns are influenced by a wide range of factors: investor sentiment, social media activity, regulatory news, market indicators (Corbet et al. [2019])... These elements introduce complex, nonlinear dependencies in its return and volatility dynamics: dependencies that often escape the reach of classical econometric models.

1 Introduction

This study investigates whether dynamically combining past returns with limit-order-book microstructure and investor sentiment signals—within a single mixture framework-can yield more accurate hourly forecasts of Bitcoin's realized volatility than traditional univariate, econometric, and deep

learning models. To answer this question, we develop and evaluate a broad set of forecasting models, comparing baseline econometric methods with more complex machine learning architectures. We assess the added value of order book dynamics and sentiment indicators through statistical testing, feature selection, and rigorous cross-validated performance benchmarks, aiming to pinpoint which signals and modeling strategies offer the most reliable predictive edge in Bitcoin's volatile market.

Volatility, Order Book and Sentiment

Let $(S_h)_h$ denote the minute-level time series of Bitcoin prices. Our objective is to predict the hourly realized volatility, denoted by $(v_t)_t$. It is defined as:

$$v_t := \sqrt{\sum_{h \in \text{hour } t} r_h^2},$$

where $r_h := \log S_t - \log S_{t-1}$ represents the log return computed at the minute frequency.

Order book features

To evaluate the predictive potential of order book data, we extract a set of features designed to capture key market microstructure signals. Following the methodology proposed by Guo et al. [2019] we consider the following set of order book features:

- Spread: The difference between the lowest ask and the highest bid:
- Ask/bid depth: Number of orders on the bid or ask side.
- Ask/bid volume: Number of Bitcoins on the bid or ask side.
- Depth/Volume difference: Difference between ask and bid depth/volume.
- Weighted spread: Difference between cumulative price over 10% of bid depth and the cumulative price over 10% of ask depth.
- Ask/bid slope: The cumulative volume over 10% of bid/ask depth.

2 Models

This section goes over the models we conceived/implemented, starting with those using alternative data sources on top of historical returns.

2.1 Temporal Mixture Models

The first models we consider are temporal mixture models introduced in Guo et al. [2019]. Intuitively, they are ensemble models tailored to time series prediction, as they can dynamically shift their ensemble weight at each time step prediction. The formal definition for such types of models is given in the appendix, we focus on concrete examples of those.

Gaussian TM

A specific instance of the temporal mixture model, referred to as the *Gaussian TM*, assumes that the conditional distribution of v_h is Gaussian, with the latent variable $z_h \in \{0, 1\}$ selecting the appropriate component:

$$\begin{aligned} v_h \mid \mathbf{v}_{[h,-l_v]}, z_h = 0 &\sim \mathcal{N}(\mu_{h,0}, \sigma_{h,0}^2) \\ v_h \mid \mathbf{X}_{[i(h),-l_b]}, z_h = 1 &\sim \mathcal{N}(\mu_{h,1}, \sigma_{h,1}^2) \end{aligned} \tag{1}$$

The component means $\mu_{h,0}$ and $\mu_{h,1}$ are parameterized as follows:

$$\begin{aligned} \mu_{h,0} &= \sum_{i=1}^{l_v} \phi_i v_{h-i} \\ \mu_{h,1} &= U^\top \mathbf{X}_{[i(h),-l_b]} V \end{aligned} \tag{2}$$

where $\phi_i \in \mathbb{R}$, $U \in \mathbb{R}^n$, and $V \in \mathbb{R}^{l_b}$, with n denoting the number of order book features used. The variance σ_h^2 is held constant across time steps and is selected empirically, following the recommendation in Guo et al. [2019].

The gating function g_h, which determines the mixture weights, is defined via a softmax function applied to smoothed activations for numerical stability:

$$g_h := \frac{\text{softplus}\left(\sum_i \theta_i v_{h-i}\right)}{\text{softplus}\left(\sum_i \theta_i v_{h-i}\right) + \text{softplus}\left(A^\top \mathbf{X}_{[i(h),-l_b]} B\right)} \tag{3}$$

where $\text{softplus}(x) := \log(1 + \exp(x))$.

The full set of model parameters is $\Theta = \{\phi_i, U, V, \theta_i, A, B\}$. These are optimized by minimizing the following loss:

$$L_G = \sum_h \log\left[g_h \cdot \mathcal{N}(v_h \mid \mu_{h,0}, \sigma_{h,0}^2) + (1 - g_h) \cdot \mathcal{N}(v_h \mid \mu_{h,1}, \sigma_{h,1}^2)\right] + \alpha\left[(\delta - \mu_{h,0})^+ + (\delta - \mu_{h,1})^+\right] + \lambda\|\Theta\|_2^2 \tag{4}$$

Sentiment TM

We extend the temporal mixture model to incorporate a second source of alternative data: a sentiment time series denoted by $(s_t)_t$. The model introduces a third latent component, $z_h = 2$, indicating that the volatility at time h is driven by sentiment signals. The parameterization of the conditional distributions for $z_h \in \{0, 1, 2\}$ follows the general formulation provided in Appendix A.

The sentiment-driven component is modeled analogously to the historical volatility component in (2), with the conditional distribution given by:

$$v_h \mid \mathbf{s}_{[h,-l_s]}, z_h = 2 \sim \mathcal{N}(\mu_{h,2}, \sigma_{h,2}^2) \tag{5}$$

The mean of this component is specified as:

$$\mu_{h,2} = \sum_{i=1}^{l_s} \phi_i^s s_{h-i} \tag{6}$$

where $\phi_i^s \in \mathbb{R}$ are learnable weights, and l_s is the lookback length for the sentiment series.

Model parameters are optimized by minimizing the following loss:

$$L_S = \sum_h \log\left[\sum_{i=0}^{2} g_h^i \cdot \mathcal{N}(v_h \mid \mu_{h,i}, \sigma_{h,i}^2)\right] + \lambda\|\Theta\|_2^2 + \alpha \sum_{i=0}^{2} (\delta - \mu_{h,i})^+ \tag{7}$$

where g_h^i denotes the gating probability associated with component $z_h = i$, derived similarly to the formulation in Equation (3).

Inverse Gamma TM

To better capture the heavy right skew observed in realized volatility, we introduce an Inverse Gamma Temporal Mixture Model (IG-TM), which replaces the Gaussian conditional distributions with Inverse Gamma distributions. This model intentionally does not incorporate sentiment data, as we aim to isolate and evaluate the effect of modifying the conditional distribution on model performance.

The conditional generative process is defined as:

$$\begin{aligned} v_h \mid \mathbf{v}_{[h,-l_v]}, z_h = 0 &\sim \text{IG}(\alpha_{h,0}, \beta_{h,0}) \\ v_h \mid \mathbf{X}_{[i(h),-l_b]}, z_h = 1 &\sim \text{IG}(\alpha_{h,1}, \beta_{h,1}) \end{aligned} \tag{8}$$

The shape and scale parameters are modeled as:

$$\begin{aligned}
\alpha_{h,0} &= 1 + \log\left[1 + \exp\left(\sum_{i=1}^{l_v} \phi_i^{\alpha} v_{h-i}\right)\right] \\
\beta_{h,0} &= \log\left[1 + \exp\left(\sum_{i=1}^{l_v} \phi_i^{\beta} v_{h-i}\right)\right] \\
\alpha_{h,1} &= 1 + \log\left[1 + \exp\left(U^{\top}\mathbf{X}_{[i(h),-l_b]}V\right)\right] \\
\beta_{h,1} &= \log\left[1 + \exp\left(U^{\top}\mathbf{X}_{[i(h),-l_b]}V\right)\right]
\end{aligned} \tag{9}$$

The constraints $\alpha > 1$ and $\beta > 0$ are enforced to ensure a finite mean and valid Inverse Gamma distribution. The gating function g_h remains as defined in Equation (3), and the model parameters are optimized by minimizing the following loss:

$$L_{\text{IG}} = \sum_h \log\left[g_h \cdot \text{IG}(v_h \mid \alpha_{h,0}, \beta_{h,0}) + (1 - g_h) \cdot \text{IG}(v_h \mid \alpha_{h,1}, \beta_{h,1})\right] + \lambda\|\Theta\|_2^2 \tag{10}$$

2.2 Benchmark models

HAR

As a baseline, we implement the HAR model introduced by Corsi [2009], which is specifically designed for volatility forecasting. The HAR model captures the behavior of market participants operating on different time horizons by incorporating realized volatility over multiple temporal aggregates. While the original formulation targets daily volatility and includes daily, weekly, and monthly components, we adapt this structure to our high-frequency setting by using 1-hour, 6-hour, and 24-hour realized volatility as explanatory variables.

Formally, the HAR model is a simple linear regression and can be written as:

$$v_{t+1} = \beta_0 + \beta_1 \cdot v_t^{(1)} + \beta_2 \cdot v_t^{(6)} + \beta_3 \cdot v_t^{(24)} + \varepsilon_{t+1}$$

with

$$v_t^{(1)} = v_t, \quad v_t^{(6)} = \frac{1}{6}\sum_{i=0}^{5} v_{t-i}, \quad v_t^{(24)} = \frac{1}{24}\sum_{i=0}^{23} v_{t-i}$$

and where ε_{t+1} is a mean-zero error term capturing unpredictable components of future volatility not explained by the model.

For more details on the theoretical construction and motivation of the HAR model, we refer the reader to Corsi [2009].

Flexible HAR

We also consider a *Flexible HAR* model, which extends the HAR framework introduced by Corsi [2009]. While the original model aggregates realized volatility over fixed horizons (day, week, month), it does not provide a method to determine the optimal number or length of these windows.

To address this, we precompute 30 rolling averages of realized volatility, corresponding to window lengths from 1 to 30 periods. These serve as candidate regressors, assembled into a feature vector:

$$\mathbf{x}_t = \left(v_t^{(1)}, v_t^{(2)}, \ldots, v_t^{(30)}\right)$$

The target variable is defined as $y_t = v_{t+1}$, the realized volatility at the next time point. To select the most informative regressors, we apply Lasso regression, which estimates the coefficient vector $\hat{\boldsymbol{\beta}}_{\text{Lasso}}$ by solving:

$$\hat{\boldsymbol{\beta}}_{\text{Lasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{t \in \mathcal{T}_{\text{train}}} (y_t - \mathbf{x}_t^{\top}\boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{30} |\beta_j| \right\}$$

This procedure shrinks less informative coefficients to zero, effectively performing variable selection. Let $\tilde{\mathbf{x}}_t \subset \mathbf{x}_t$ denote the reduced feature vector containing only the selected windows. We then refit a final OLS model—the Flexible HAR—on the full training set using $\tilde{\mathbf{x}}_t$, by solving:

$$\hat{\boldsymbol{\beta}}_{\text{HAR}}^{\text{flex}} = \arg\min_{\boldsymbol{\beta}} \sum_t (y_t - \tilde{\mathbf{x}}_t^\top \boldsymbol{\beta})^2$$

Note: In the case of a *Flexible HARX* specification, we extend this framework by including lagged order book variables as additional candidate features in the Lasso selection procedure. The notation *Flexible HARX(ℓ)* indicates that we consider all lagged order book variables up to lag ℓ.

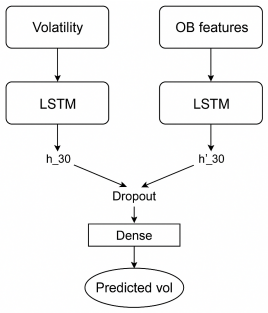
LSTM

LSTM networks were first introduced by Hochreiter and Schmidhuber in Sepp Hochreiter [1997] to address the vanishing gradient problem, a well-known limitation in standard RNNs. In recent years, LSTMs have been widely adopted in the context of cryptocurrency price forecasting (Alessandretti et al. [2019]), owing to their ability to learn long-term dependencies and capture temporal dynamics in sequential data. These properties make them particularly suitable for time series forecasting tasks such as volatility prediction.

In our model, we implement two separate LSTM networks:

- The first takes as input the past 30 timesteps of order book features.
- The second processes the past 30 timesteps of realized volatility values.

We extract the final hidden state from each LSTM, concatenate them into a single representation $\tilde{h}$, and apply a dropout mask to prevent overfitting. This combined vector is then passed through two fully connected (dense) layers. The first layer projects $\tilde{h}$ into an intermediate feature space, while the second layer outputs a scalar prediction in $\mathbb{R}$. A ReLU activation function is used after each layer to ensure that the predicted volatility remains non-negative. Here is a recapitulative scheme of our network:



Other implementations

In addition to the previously discussed models, we implement two additional predictors: GARCH and XGBoost. These serve as solid benchmarks and also form the basis for hybrid ensembles. Specifically, we construct XGB × GARCH and LSTM × GARCH models. The rationale is that GARCH captures volatility clustering effectively, while LSTM and XGBoost are better suited for modeling nonlinear patterns and long-range dependencies.

3 Experiments

Order Book Data

We use limit order book and trade data for the BTC/USD pair from the Bitstamp exchange, spanning June 4, 2018, 21:55 UTC to September 30, 2018, 21:59 UTC. The dataset consists of 318,616 snapshots recorded at 30-second intervals, each capturing bid and ask quotes. These are aggregated to a one-minute resolution, and hourly features are derived from the final minute of each hour.

To ensure data quality, we exclude the first three days of each month due to missing entries. Remaining gaps are filled using forward-fill imputation. For bid prices, we retain only those below the lowest ask and within 25% of the highest bid. A symmetric filter is applied to ask prices.

Sentiment Data

We use two sentiment time series from the dataset at `https://huggingface.co/datasets/danilocorsi/LLMs-Sentiment-Augmented-Bitcoin-Dataset`: the Fear and Greed Index (FNG) and the Crypto Bull & Bear Index (CBI), both normalized to the $[0, 1]$ interval. We define an *aggregate sentiment measure* (AggS_t) as the average of the normalized FNG and CBI. To isolate an orthogonal sentiment signal, we extract the residuals ϵ_t from the linear regression:

$$\text{FNG}_t = \beta_0 + \beta_1 \text{AggS}_t + \epsilon_t.$$

Statistical Tests

Prior to implementing predictive models, it is essential to examine the statistical properties of the data. To this end, we conduct a series of diagnostic tests on the datasets employed in our volatility modeling and sentiment analysis framework.

Diagnostics for the Volatility Time Series

We test for stationarity in the volatility time series because many modeling techniques assume or perform better with stationary data.

We use the Augmented Dickey-Fuller (ADF) test (Dickey and Fuller [1979]) to check for a unit root. The null hypothesis is rejected at the 5% significance level, indicating that the series is stationary. However, the Kwiatkowski-Phillips-Schmidt-Shin (Kwiatkowski et al. [1992]) test also rejects its null hypothesis of stationarity at the 5% level, suggesting non-stationarity. This conflicting outcome is common and points to structural complexity in the data, such as time-varying volatility or regime shifts, which simple tests may not capture.

Diagnostics for Order book features and news sentiment

We did not provide a theoretical justification for the order book features introduced in Section 1, but instead adopted the feature set proposed by Guo et al. Guo et al. [2019]. Prior to incorporating these features into our predictive models, we assess their relevance by examining redundancy and predictive power.

To this end, we apply the Granger causality test (Granger [1969]) with a maximum lag of 30 to each feature individually. The results indicate that only a subset of features Granger-cause volatility: ask depth (lags 1–30), bid depth (lags 1–9), bid volume (lags 1–30), spread (lags 1–8), and volume difference (lags 1–30). Based on this, we design two sets of experiments—one using only the Granger-significant features, and another using the full feature set from Section 1—to evaluate whether the excluded variables contribute additional predictive value.

For the sentiment time series, we find that aggregate sentiment Granger-causes volatility up to lag 2, while the orthogonal component of the Fear and Greed Index Granger-causes volatility up to lag 19.

Testing Procedure & Cross-validation

To evaluate the performance of our time series models, we employ a rolling window approach for training and validation. Specifically, we partition the time series into ten sequential intervals. For each iteration, models are trained on three consecutive intervals, validated on the fourth, and then retrained using the optimal hyperparameters across the first four intervals before being evaluated on the fifth.

Cross-validation in financial time series requires particular caution due to serial dependence. As noted by López de Prado [2018], improper splitting can introduce data leakage between training and validation sets. To mitigate this risk, we account for the autocorrelation structure of our volatility series, which shows significant autocorrelation up to lag 5. Accordingly, we exclude the first five observations of each validation and test interval to prevent leakage from past information.

Additionally, we address the risk of test set overfitting. Using the test set for both model selection and evaluation can lead to overoptimistic performance estimates. Our split—separating training, validation, and testing intervals—aims to ensure a fair and robust evaluation of model generalization.

Hyperparameter Tuning

This section outlines the hyperparameters tuned for each model. In cases where standard grid search on the cross-validation setup described in Section 3.3 is not feasible, we specify the alternative approach used.

Temporal Mixture Models: Several hyperparameters are associated with the temporal mixture models, primarily the lookback windows for volatility, order book features, and sentiment. The volatility lookback window is fixed at $l_v = 5$, based on the observation that the autocorrelation of realized volatility becomes statistically insignificant beyond five lags. For the sentiment component, we use only the aggregate sentiment series and set the lookback window to 2 as indicated by the GCT. The lookback window for order book features, along with all regularization hyperparameters, is selected via cross-validation.

Flexible HAR: The regularization parameter λ in the Lasso stage of the Flexible HAR model is selected via time-series cross-validation. The training set is split into five sequential folds using `TimeSeriesSplit` from `scikit-learn`, which preserves temporal ordering. Lasso is then trained over a grid of 100 logarithmically spaced λ values between 10^{-4} and 1, and the value minimizing average prediction error across folds is chosen.

LSTM Models: For both LSTM-based models, training minimizes the RMSE with L2 regularization. We use the AdamW optimizer, which decouples weight decay from gradient updates. The following hyperparameters are tuned via grid search: Hidden state and dense layer sizes: $\{32, 64, 128, 256\}$, learning rate: $\{0.0005, 0.001\}$, weight decay: $\{0.0001, 0.001\}$, dropout rate: $\{0.2, 0.4, 0.6, 0.8\}$.

GARCH: The only model that does not follow the training-evaluation protocol outlined in the previous section is GARCH. As noted by Hansen and Lunde [2005], GARCH models are typically re-estimated at each prediction point using a rolling window of the most recent data, rather than adhering to a fixed train-test split as in conventional machine learning.

This forecasting approach introduces a hyperparameter: the size of the rolling window. Accurate tuning of this parameter is critical. If, instead of rolling, one incrementally expands the training window without resetting, the model's forecasts for conditional volatility σ_t tend to converge to a constant, diminishing its responsiveness to new information.

XGBoost:
The final benchmark model is XGBoost (Chen and Guestrin [2016]). We give XGBoost past lags as well as volatilites' rolling means similarly to the HAR models as inputs. We tune several key hyperparameters, including the maximum tree depth, number of trees, minimum loss reduction, and the L2 regularization term. Additionally, we train XGBoost using a custom loss function defined as $L(\hat{y}, y) = \sum_{i=1}^{|y|} e^{\alpha \frac{y_i}{\max_i y_i}} (y_i - \hat{y}_i)^2$ which emphasizes larger target values. This formulation encourages the model to better capture volatility spikes by assigning greater weight to higher-magnitude prediction errors. The parameter alpha from the loss is also tuned during CV.

Discussion of the results

The test results for each interval are provided in Appendix B due to space constraints.

Across all experiments, one consistent result stands out: univariate models such as HAR and Flexible HAR, which rely solely on past volatility, perform on par with—or only marginally below—their multivariate counterparts that incorporate order book features. This similarity in performance highlights a key challenge: while alternative features theoretically offer richer information, extracting meaningful predictive signal from them is difficult in practice. The mean-reverting nature of volatility and the effectiveness of temporal smoothing allow these simpler models to deliver robust, low-RMSE forecasts without the added complexity.

However, there are moments where models using alternative data show a clear edge. Figure 2 demonstrates this with the LSTM model capturing a volatility spike in real time—an event that HAR-based models tend to smooth over or lag behind. This suggests that order book and similar features may indeed carry predictive signals, but they are buried in noise and only become useful in specific contexts, such as sudden market shifts.

Further evidence of this difficulty is visible in the TM-model gating plots (Figure 1), where the contribution of historical volatility overwhelmingly dominates. Order book components, especially those not shown to Granger-cause volatility, contribute little to the model's final prediction. This implies that the current architecture i.e bilinear gating, may be too limited to properly leverage high-frequency, noisy signals like those found in order book data.

Hybrid models that combine different paradigms, such as SXGB × GARCH, demonstrate competitive performance and, in some intervals, achieve the lowest RMSE. For example, in Interval 6, it achieves the best result among all compared variants (0.0454), outperforming both SXGB (0.0494) and GARCH (0.0464). However, this superiority is not consistent across all periods—SXGB alone matches or even slightly outperforms the hybrid in other intervals (e.g., Interval 4 and 5). These results suggest that while structural complementarity can be beneficial, its effectiveness depends on the volatility regime.

Models incorporating sentiment data also fail to show consistent improvement. Comparisons between TM-G and TM-SG, or between XGB-GCT and SXGB-GCT, indicate minimal benefit from sentiment features—likely due to data quality issues.

Lastly, the IG-TM model had to be discarded due to instability. Its use of the Inverse Gamma distribution led to erratic predictions and outlier RMSE values, likely due to the heavy-tailed nature of the distribution undermining the model's ability to produce stable outputs.

In short, while past volatility remains the most reliable predictor, there are isolated cases—like the LSTM's success in Figure 2—where alternative data provides real value. Unlocking this potential will require better architectures and more selective integration strategies rather than naïvely adding more inputs.

4 Conclusion

To conclude, we explored the task of forecasting Bitcoin's realized volatility using a diverse set of modeling approaches. The central question was whether dynamically combining historical returns with limit order book data or news sentiment could produce more accurate hourly forecasts of Bitcoin volatility than traditional econometric or deep learning models. Our results suggest that while advanced modeling techniques and alternative data sources hold promise, their predictive advantage remains marginal and is primarily driven by a small subset of well-chosen features—underscoring the critical importance of feature quality.

Future research could focus on developing more effective methods for extracting meaningful signals from the inherently noisy order book data and on constructing or integrating higher-quality sentiment indicators. These enhancements may unlock more of the potential embedded in alternative data sources and improve forecasting performance in high-frequency crypto markets.

References

Laura Alessandretti, Abeer ElBahrawy, Luca Maria Aiello, and Andrea Baronchelli. Anticipating cryptocurrency prices using machine learning. *arXiv:1805.08550*, 2019. URL https://doi.org/10.48550/arXiv.1805.08550.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 785–794. ACM, 2016. doi: 10.1145/2939672.2939785.

Shaen Corbet, Charles Larkin, Brian M. Lucey, Andrew Meegan, and Larisa Yarovaya. The impact of macroeconomic news on bitcoin returns. *The European Journal of finance*, 2019. URL https://doi.org/10.1080/1351847X.2020.1737168.

Fulvio Corsi. A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196, 2009. URL https://statmath.wu.ac.at/~hauser/LVs/FinEtricsQF/References/Corsi2009JFinEtrics_LMmodelRealizedVola.pdf. University of Lugano and Swiss Finance Institute.

D. Dickey and Wayne Fuller. Distribution of the estimators for autoregressive time series with a unit root. *JASA. Journal of the American Statistical Association*, 74, 06 1979. doi: 10.2307/2286348.

Thomas Dimpfl Dirk G. Baur. The volatility of bitcoin and its role as a medium of exchange and a store of value. *Empirical Economics*, 2021. URL https://doi.org/10.1007/s00181-020-01990-5.

Andreas M. Fischer, Rafael P. Greminger, Christian Grisse, and Sylvia Kaufmann. Portfolio rebalancing in times of stress. *Journal of International Money and Finance*, 2021. URL https://doi.org/10.1016/j.jimonfin.2021.102360.

Clive W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, 37(3):424–438, 1969.

Tian Guo, Albert Bifet, and Nino Antulov-Fantulin. Bitcoin volatility forecasting with a glimpse into buy and sell orders. *arXiv preprint arXiv:1802.04065*, 2019. URL http://arxiv.org/abs/1802.04065.

Peter Reinhard Hansen and Asger Lunde. A forecast comparison of volatility models: Does anything beat a garch(1,1)? *Journal of Applied Econometrics*, 20(7):873–889, 2005. doi: 10.1002/jae.800.

Yu-Chen Wei Her-Jiun Sheu. Effective options trading strategies based on volatility forecasting recruiting investor sentiment. *Expert Systems with Applications*, 2011. URL https://doi.org/10.1016/j.eswa.2010.07.007.

Denis Kwiatkowski, Peter C.B. Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1-3):159–178, 1992. doi: 10.1016/0304-4076(92)90104-Y.

Marcos López de Prado. *Advances in Financial Machine Learning*. John Wiley & Sons, Hoboken, NJ, 2018. ISBN 9781119482086.

Jürgen Schmidhuber Sepp Hochreiter. Long short-term memory. *Neural Computation*, 1997. URL https://www.researchgate.net/publication/13853244_Long_Short-Term_Memory.

Pamela Dent Stavros Degiannakis, Christos Floros. Forecasting value-at-risk and expected shortfall using fractionally integrated models of conditional volatility: International evidence. *International review of financial analysis*, 2013. URL https://doi.org/10.1016/j.irfa.2012.06.001.

Appendix

A Temporal Mixture Model - A formal definition

Let $(v_t)_t$ denote a time series, and let $(F_h^0)_h, \ldots, (F_h^{m-1})_h$ represent m distinct feature sequences hypothesized to carry predictive information about $(v_t)_t$. Define $(z_h)_h$ as a sequence of latent categorical variables taking values in $\{0, \ldots, m-1\}$, where each z_h indicates the feature index relevant for modeling the conditional distribution of v_h. A temporal mixture model defines the joint distribution over the sequence $(v_1, \ldots, v_H)$ by marginalizing over the latent variables, resulting in the following formulation:

$$
\begin{aligned}
p(v_1, ..., v_H \mid \mathcal{D}; \Theta) &= \sum_{z_1} \cdots \sum_{z_H} p(v_1, \ldots, v_H, z_1, \ldots, z_H \mid \mathcal{D}; \Theta) \\
&= \prod_h \big[\sum_{i=0}^{m-1} p(v_h, z_h = i \mid \mathcal{D}; \Theta) \big] \\
&= \prod_h \big(\sum_{i=0}^{m-1} p(v_h \mid F_h^i, z_h = i; \Theta) \cdot g_h^i \big)
\end{aligned}
$$

where $g_h^i = p(z_h = i \mid \mathcal{D}; \Theta)$

B Results

All figures, plots, and extended results are available at our GitHub repository: `https://github.com/AugusteDP-git/bitcoin-volatility-forecasting`.

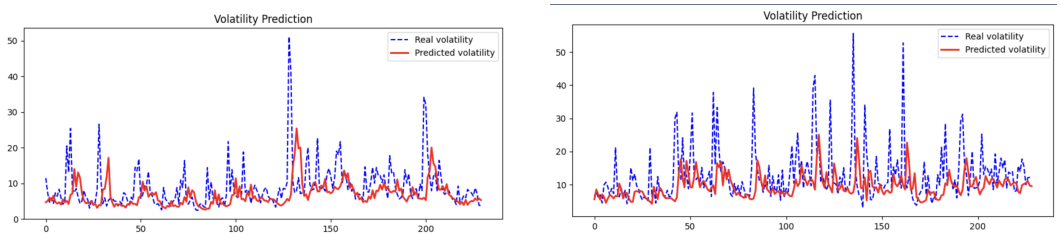
Table 1: RMSE Across Intervals for Different Models exclusively with GCT features

Interval	LSTM	HAR	Flexible HAR	HARX(10)	Flexible HARX(10)	GCT TM-G	TM-SG	TM-G
Interval 1	**0.0651**	0.0658	0.0657	0.0706	0.0702	0.0739	0.0742	0.762
Interval 2	0.0810	0.0756	**0.0753**	0.0773	0.0770	0.0907	0.0908	0.873
Interval 3	0.0602	0.0552	0.0547	0.0530	**0.0529**	0.0637	0.0633	0.641
Interval 4	0.0824	0.0708	0.0719	0.0745	0.0746	0.0873	0.0881	0.904
Interval 5	0.0627	0.0509	0.0515	0.0486	**0.0485**	0.0531	0.0534	0.532
Interval 6	0.0524	0.0365	0.0363	**0.0358**	0.0359	0.0363	0.0464	0.463

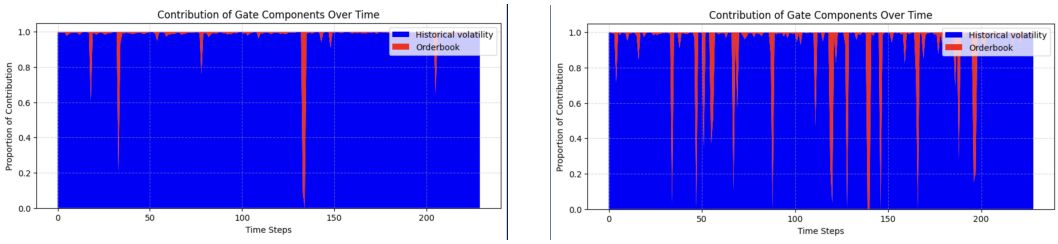
Table 2: RMSE Across Intervals for XGBoost with Different Implementations

Interval	GARCH	XGB	SXGB	XGB-GCT	SXGB-GCT	SXGB × GARCH	LSTM × GARCH
Interval 1	0.0818	0.0684	0.702	0.0832	0.1165	0.0704	0.0693
Interval 2	0.0881	0.0777	0.0792	0.0939	0.0870	0.0781	0.0835
Interval 3	0.0683	0.0653	0.0713	0.0559	0.0566	0.0646	0.0642
Interval 4	0.0868	0.0708	**0.0704**	0.0720	0.0706	0.0740	0.0830
Interval 5	0.0635	0.0524	0.0532	0.0529	0.0529	0.0546	0.0565
Interval 6	0.0464	0.0459	0.0494	0.0912	0.0886	0.0454	0.0454

NB: The acronym SXGB model denotes the XGBoost model with sentiment features on top of historical volatility. Moreover GCT features denote the 5 features that the Granger causality test found significant.



Left: Forecast vs. Truth **Right:** Forecast vs. Truth



Left: Gating values **Right:** Gating values

Figure 1: Prediction and gating of the TM-G behavior for two selected time intervals. Top row: predicted vs. actual volatility. Bottom row: temporal gating weights assigned to order book vs. volatility signal.

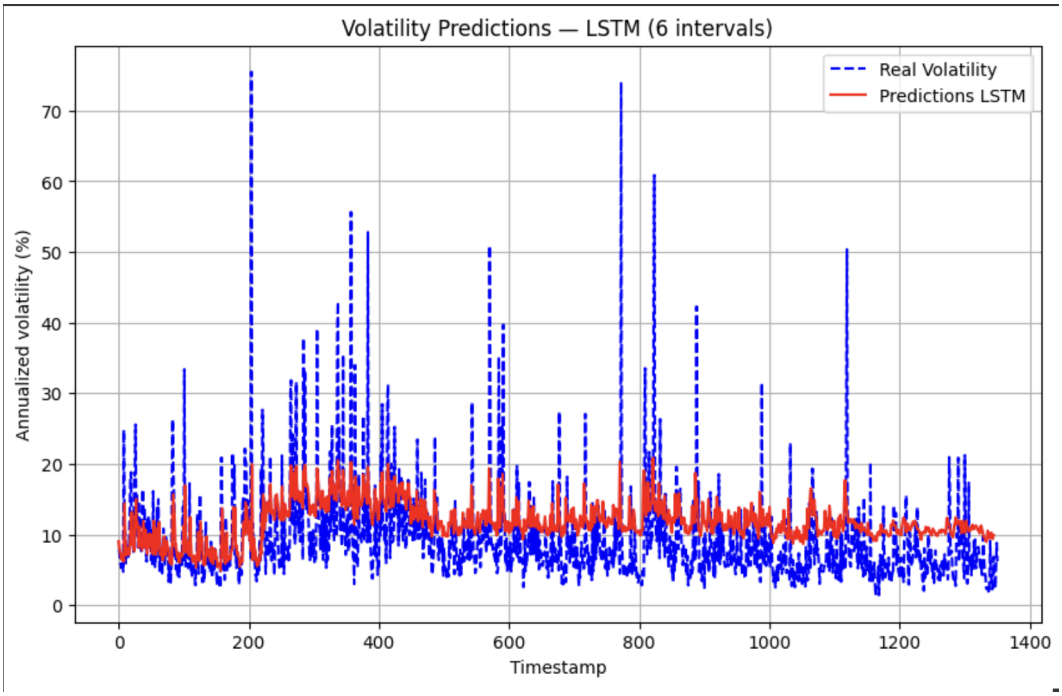


Figure 2: LSTM rolling prediction