FIN-525: Financial Big Data

Group Project:

RFSV Forecasting with High-Frequency Data

Emanuele Sorgente, Giuseppe Cognata

5th February 2023

EPFL

Abstract

In this project, we investigate the use of rough volatility in forecasting realized variance using high frequency data, particularly with assets traded 24 hours a day. We collect and manipulate a large amount of high frequency data from a crypto asset and implement our own algorithms based on previous work with the rough fractional volatility to forecast the volatility of the asset. Our results suggest that incorporating the fractal-like behavior of volatility as captured by rough volatility models leads to very accurate realized variance forecasts compared to traditional methods, but these methods face challenges when the forecast time scale becomes small.

1 Introduction and Outline

The dataset used for the model comes from "Binance Data Collection" [1]. We combined the trading data of a token every second over a 6 month period, specifically from January 1st 2022 to July 1st 2022. Since Cryptos are traded 24/7, we will not need to assume that there will be an opening time and a closing time and a weekend break in the market. This could be a source of difficulty when forecasting volatility at such high frequency, as we expect trades to have some intrinsic seasonality.
The content of the report is as follows. In section 2 we describe the given data and provide a short exploratory data analysis. Sections 3 to 5 are the core body of the report that contains the specification of the models used, the theory behind the idea of rough volatility , the training, testing and fitting modules. In Section 6 we give an interpretation of the results obtained and suggest potential improvements. Finally, in Section 7 we give a short conclusion.

2 Data

The total amount of data in analysis is $N = 181(Days) \times 24(Hours) \times 60(Minutes) \times 60(Seconds) = 15.638.400$. To obtain this dataset, we manually merged six csv files obtained at [1] containing each one month of data The token taken into account for our project is "1INCHBTC" (1-inch Bitcoin).

	datetime	open	high	low	close	volume	num_trades	day_of_week
0	2022-01-01 01:00:00	0.000052	0.000052	0.000052	0.000052	0.0	0.0	5
1	2022-01-01 01:00:01	0.000052	0.000052	0.000052	0.000052	0.0	0.0	5
2	2022-01-01 01:00:02	0.000052	0.000052	0.000052	0.000052	0.0	0.0	5
3	2022-01-01 01:00:03	0.000052	0.000052	0.000052	0.000052	0.0	0.0	5
4	2022-01-01 01:00:04	0.000052	0.000052	0.000052	0.000052	0.0	0.0	5
...	...	...	...	...	...	...	...	...
15638395	2022-07-01 01:59:55	0.000035	0.000035	0.000035	0.000035	0.0	0.0	4
15638396	2022-07-01 01:59:56	0.000035	0.000035	0.000035	0.000035	0.0	0.0	4
15638397	2022-07-01 01:59:57	0.000035	0.000035	0.000035	0.000035	0.0	0.0	4
15638398	2022-07-01 01:59:58	0.000035	0.000035	0.000035	0.000035	0.0	0.0	4
15638399	2022-07-01 01:59:59	0.000035	0.000035	0.000035	0.000035	0.0	0.0	4

Figure 1: Dataframe's visualization

The dataframe assumes the look of Figure 1. As we can see, the values of open, high, low and close are the same, reflecting the high-frequency nature of the data. As a consequence, the columns we'll be interested in are `datetime` and `open`.
Below we show the trend of the asset price during the months considered.

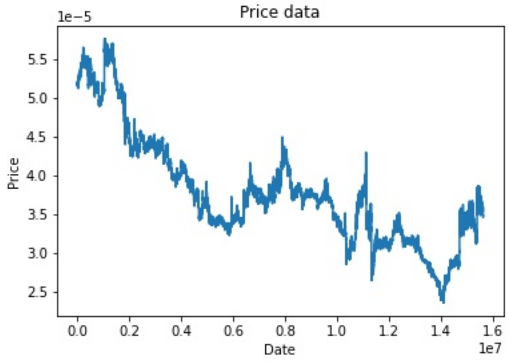

Figure 2: 1INCHBTC' Price January-July 2022

From a first visual analysis we can assume that the trend tends to be decreasing.
Below are showed the different plots of the returns based on different time steps.

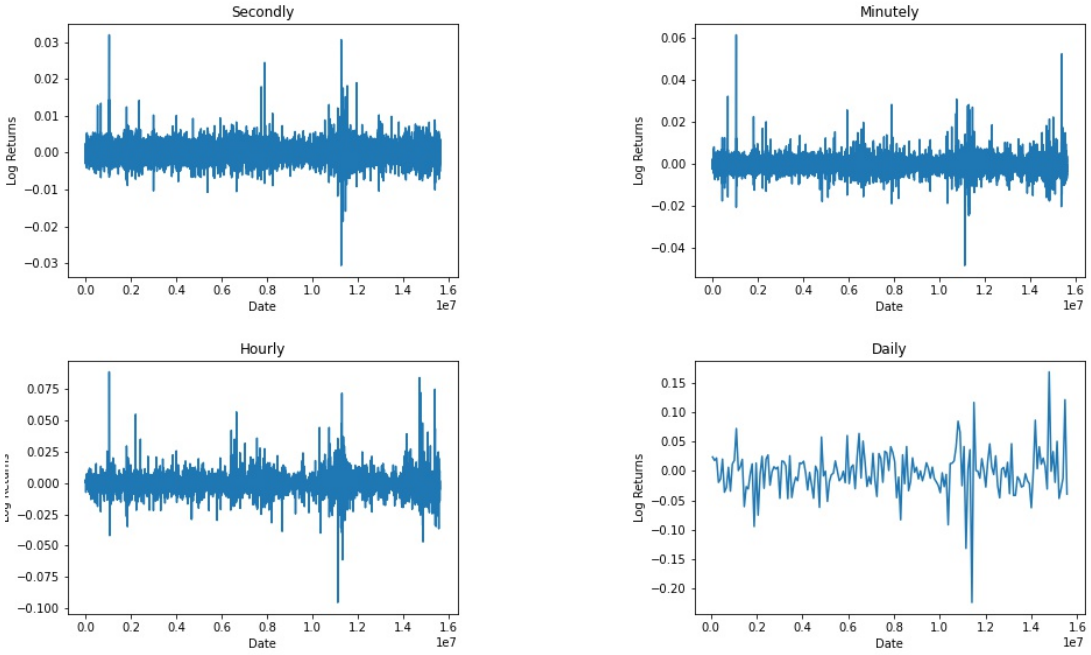

Figure 3: Plots of the returns of 1INCHBTC per second, minute, hour and day

3 The Model: Mathematical context

We summarize the main mathematical context of Rough Volatility Models, stating the results and formulas that we will use for the project. For a detailed account, see [2] and the references therein.

The typical continuous time stochastic model for an asset price process $(S_t)_{t\geq 0}$ is of the form:

$$dS_t = \mu_t dt + \sigma_t dW_t \tag{1}$$

so that S is a semi-martingale. Most innovation on models of this form lies in choosing a suitable volatility process $(\sigma_t)_{t\geq 0}$, but recent work in [2] suggests that volatility may have been far rougher than this type of model suggests. In order to express this roughness, the important generalization made by [2] is to model volatility using a generalization of the classical Brownian motion: fractional Brownian motion (fBM). A fBM $(W_t^H)_{t\geq 0}$ with Hurst exponent $H \in (0,1)$ is a centered Gaussian process with stationary increments such that, for any $\Delta, t \geq 0$ and $q > 0$:

$$\mathbb{E}[|W_{t+\Delta}^H - W_t^H|^q] = \mathbb{E}[|Z|^q]\Delta^{qH} \tag{2}$$

where $Z \sim \mathcal{N}(0,1)$. If $H = \frac{1}{2}$, then the fBM is a simple standard Brownian motion, and the path of the process is rougher the smaller the value of H. Empirically, increments of log-volatility appear to behave very similarly to a fBM with Hurst exponent of the order of 0.1. The model we will be exploring in this project is the Rough Fractional Stochastic Volatility model (RFSV model) from [2]. The simple, intuitive specification to represent these empirical results would be to set:

$$\log \sigma_{t+\Delta} - \log \sigma_t = \nu(W_{t+\Delta}^H - W_t^H) \tag{3}$$

with $\nu > 0$ a constant parameter. In this model, however, the volatility process is not stationary. So, more precisely, the RFSV model assumes that log-volatility follows a stationary fractional Ornstein-Uhlenbeck process:

$$dX_t = \nu dW_t^H - \alpha(X_t - m)dt \tag{4}$$

In the rough model, where $H < \frac{1}{2}$, we also suppose that, focusing our attention on a fixed time horizon $T > 0$, $\alpha \ll \frac{1}{T}$. This assumption makes interpreting this seemingly highly complex model far easier, as it can be shown that in this case, the log-volatility behaves as a fBM at any time scale smaller than T. More formally, we have that:

$$\lim_{\alpha \to 0} \mathbb{E}\left[\sup_{t\in[0,T]} \left|X_t - X_0 - \nu W_t^H\right|\right] = 0 \tag{5}$$

This gives us a mathematical justification for assuming 3. Within this framework, one can show a straight-forward formula for (expected) variance prediction:

$$\mathbb{E}[\sigma_{t+\Delta}^2|\mathcal{F}_t] = \exp\left(\mathbb{E}[\log \sigma_{t+\Delta}^2|\mathcal{F}_t] + 2\frac{\Gamma\left(\frac{3}{2} - H\right)}{\Gamma\left(H + \frac{1}{2}\right)\Gamma(2-2H)}\nu^2\Delta^{2H}\right)$$

where $\mathcal{F}_t$ is the filtration generated by W_t^H, Γ is the Gamma function and:

$$\mathbb{E}[\log \sigma_{t+\Delta}^2|\mathcal{F}_t] = \frac{\cos(H\pi)}{\pi}\Delta^{H+\frac{1}{2}}\int_{-\infty}^{t} \frac{\log \sigma_s^2}{(t-s+\Delta)(t-s)^{H+\frac{1}{2}}}\, ds \tag{6}$$

4 Testing the Model's assumptions

We begin by computing the empirical volatility from the available data. With it being so high-frequency, we cannot apply range volatility estimators, but rather the simple standard deviation of log returns serves as the most effective and natural estimator for volatility. For this computation, we compute the volatility every $\delta t = 300$ seconds (5 minutes), yielding the following time series:

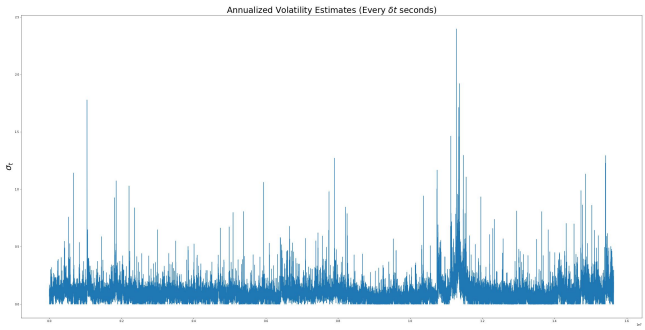

Figure 4: Estimates of Annualized volatility every δt seconds

Treating the log-volatility as a fBM, the main assumption that the model must verify is dW_t^H (and thus $d \log \sigma_t$) being Gaussian. In practice, with high-frequency data on 24-hour traded assets, this assumption might not hold on a regular time scale, as there will be gaps in the trading frequency at weekends or night-time. As a result, this will make the log-volatility increments behave less like a Gaussian variable, making the data unsuitable for the model. We can observe this in the following plot, where the increments are taken every $\Delta = 5$ (where one time step indicates a step of δt seconds):

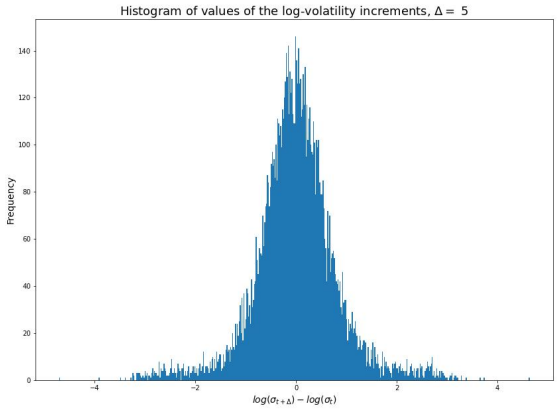

Figure 5: Histogram of increments in log-volatility every $\Delta = 5$ intervals of δt seconds

We would thus want to remove the seasonality of (log-)volatility by re-scaling it by a factor that accounts for this periodicity. One way to do this is to account for seasonality by day of the week and hour of the day. If we are in day $d \in \{0, 1, ..., 6\}$ and hour $h \in \{0, 1, ..., 23\}$, we can compute a rescaling factor f defined as:

$$f_{d,h} = \sqrt{\widehat{\mathbb{E}}[(\log \sigma_t)^2 | d, h]} \tag{7}$$

where $\widehat{\mathbb{E}}$ denoted the empirical average, and the conditioning indicates that we compute this average over all the log-volatility measurements made on day d and hour h. In order to avoid explosive numbers, we can normalize these factors by dividing all of them by their mean, hence ensuring that they are on average equal to one. This will take into account the seasonality of trading (hence of log-volatility), and should make the volatility of log-volatility more constant, and the increments of log-volatility more Gaussian. We thus obtain a total of 168 (7 days a week, 24 hours per day) rescaling factors. In the figure below, we plot them in increasing hour of the day, where each red line separates the different days of the week:

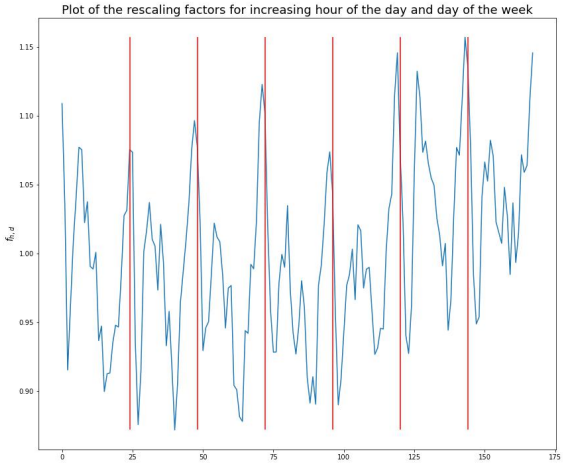

Figure 6: Plot of the scaling factor $f_{h,d}$. Red lines separate the days of the week (Monday through Sunday)

In the figure above we can observe that, for any given day of the week, there is a peak at the beginning of the day, and another, usually smaller one, around the middle of the day. These are the times where log volatility is itself more volatile, such as times of day where trades are more frequent. We can then rescale the log volatility by the corresponding factor and, using again a lag of $\Delta = 5$, the histogram of values of $\log \sigma_{t+\Delta} - \log \sigma_t$ now becomes:

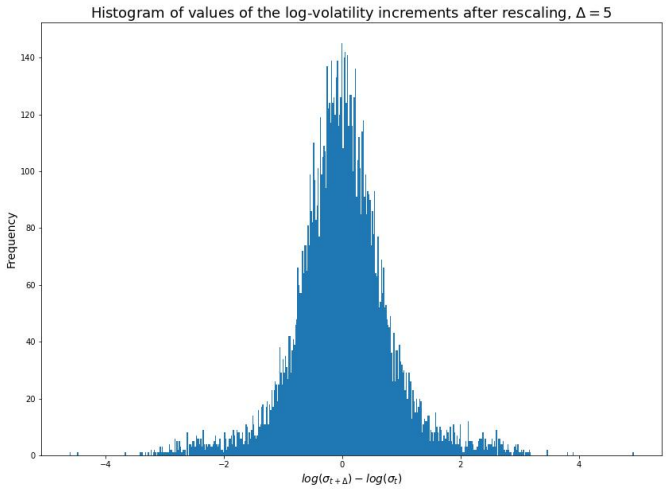

Figure 7: Histogram of increments in the rescaled log-volatility every $\Delta = 5$ intervals of δt seconds

This resembles more closely a Gaussian distribution, as was the desired effect.
Note that, due to a significant presence of zero values in the volatility estimates, when computing $\log \sigma_t$, actually $\log(\sigma_t + \sigma_{min})$ was calculated, where $\sigma_{\min} = \inf\{\sigma_t | \sigma_t > 0\}$.

5 Fitting and prediction

In order to use the predictive formula in (3), we need to first calibrate the parameters of the model: ν and H. To estimate H, we use a two-timescale estimator given by equation (11) in [4], so that, defining two time increments $\tau_1, \tau_2 > 0$, we have that the Hurst exponent in the time interval $[t - T, t]$ (with T the size of a rolling calibration window) is given by:

$$\hat{H}_t = \frac{1}{2\log\left(\frac{\tau_1}{\tau_2}\right)} \log\left(\frac{(T-\tau_1)\sum_{i=0}^{T-\tau_1}(\log(\sigma_{t-i}) - \log(\sigma_{t-i-\tau_1}))^2}{(T-\tau_2)\sum_{i=0}^{T-\tau_2}(\log(\sigma_{t-i}) - \log(\sigma_{t-i-\tau_2}))^2}\right) \tag{8}$$

which essentially computes a scaled difference in empirical log-variance of increments of sizes τ_1 and τ_2. Computing an estimate for H in a series of rolling windows containing 500 points, we obtain the following time-varying estimates:

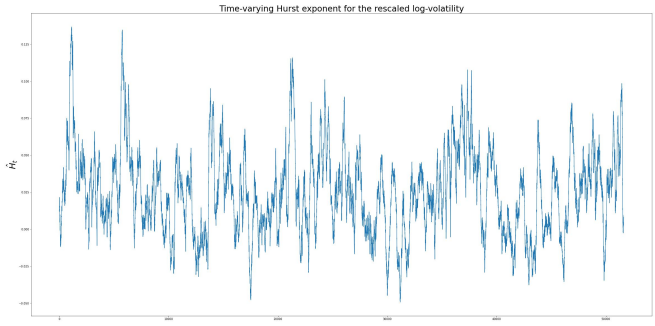

Figure 8: Rolling H estimates in windows with 500, $\tau_1 = 245$, $\tau_2 = 20$

Notice the extremely low variance of H, and how the estimates do not exceed 0.13, as is consistent with the findings in [2]. Furthermore, notice how some of the estimates are negative (although extremely small). This is likely a consequence of the data's extremely high frequency. Then, once we know H, we can estimate ν by using Corollary 3.2 in [2], which states that for any $t > 0$:

$$\mathrm{Cov}[\log \sigma_t, \log \sigma_{t+\Delta}] = \mathbb{V}\mathrm{ar}(\log \sigma_t) - \frac{1}{2}\nu^2 \Delta^{2H} + o(1) \tag{9}$$

Hence, we can find an estimate for ν regressing the linear (in Δ^{2H}) model above. However, we find empirically that for the linear model to be accurate, we require a large amount of covariance data, hence we cannot apply the same rolling idea in this case. Thus, we will calibrate H in a rolling manner, but only perform a single calibration of ν over the entire dataset, (using an estimate for H that also encompasses the whole history of the data in this case). The plot below shows the linear fit to the empirical autocovariance of the log-volatility:

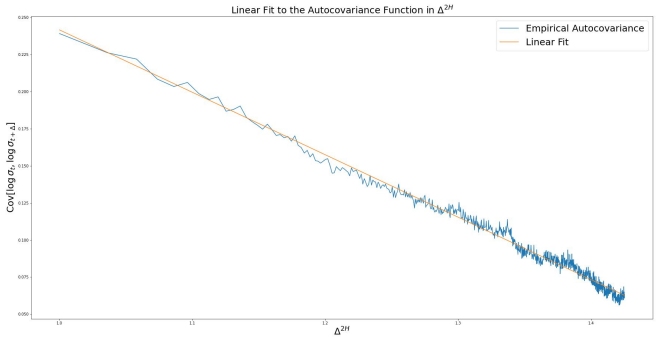

Figure 9: Linear regression of the autocovariance of log-volatility on Δ^{2H}

We can see the regression is fairly accurate ($R^2 = 0.98$) and the data looks very linear. This regression leads us to our estimate $\nu \approx 0.91$.

Using these two procedures, and an implementation of equation 3 found in [3] that approximates the integral as a Riemann sum, we can predict the volatility every δt seconds, using a rolling window with 500 calibration points to compute H and the integral. Superimposing the predicted volatility to the actual data (the first 500 points are omitted, as we cannot predict them) we obtain the following plot:

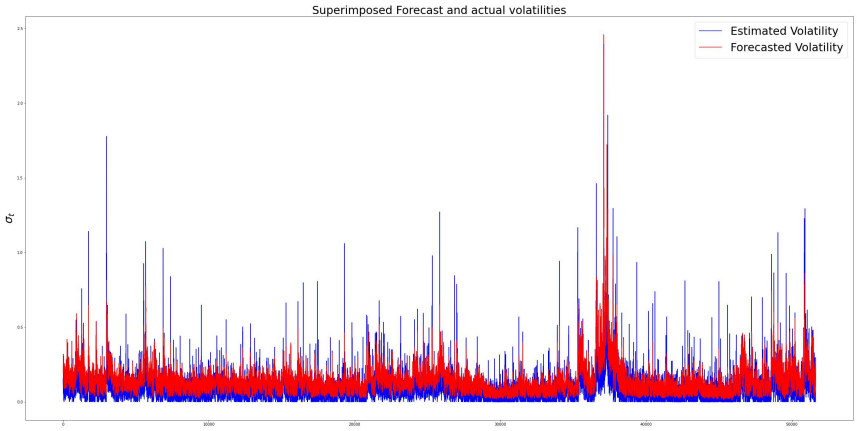

Figure 10: Superimposed forecasted volatility to estimated volatility

If we focus on a smaller subset of the data, we can observe the accuracy of the prediction in more detail:

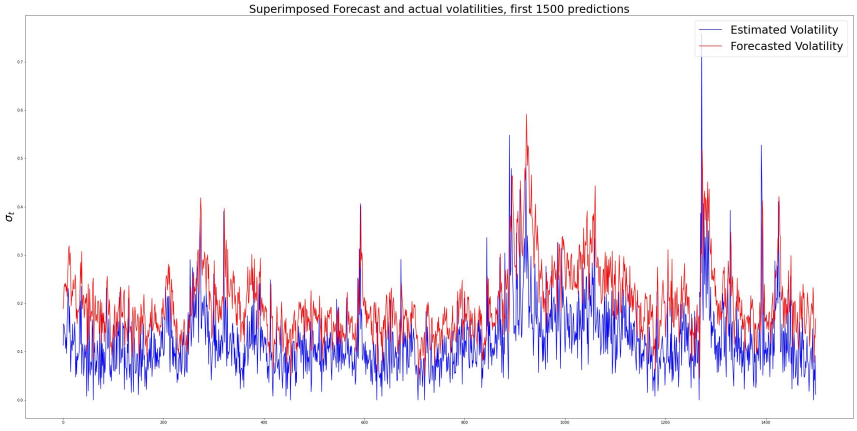

Figure 11: Superimposition of the first 1500 forecasted points to the actual estimates

Alternatively, we can visualize the èredictive power of the model using a scatter plot:

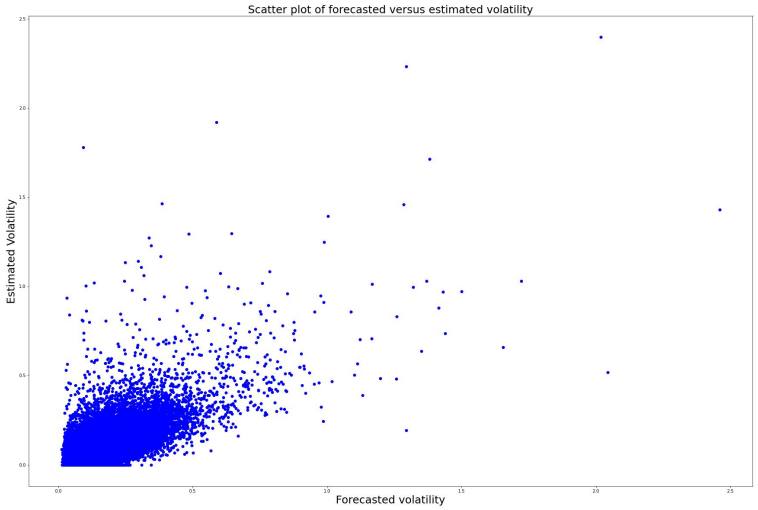

Figure 12: Scatter plot of forecasted volatility against estimated volatility

6 Interpretation and Potential Improvements

As we can see form figures 10 and 11, the model is very accurate at predicting volatility every five minutes, even when using rolling calibration. The prediction closely follows the trends, spikes and dips in the time series, yielding a mean squared error of around 0.007. The scatter plot in figure 12 also shows particularly promising precision for values smaller than 0.5, where the majority of the data is concentrated. Considering the range of the data, this is very promising, and suggests that, if the data is treated properly, the RFSV model can be used to precisely forecast the volatility of a price process. In particular, when applying the model's predictive formulas to assets traded 24 hours a day, one must ensure that volatility is properly de-seasonalized, in order to account for jumps due to regular increases or decreases in trading activity.

The first challenge we came up against in this project was the high number of zero values in the volatility estimates, which is a given when using data of such high frequency. We opted to use a pseudocount, which is common practice when facing similar issues. However, this likely introduces a slight bias in the computations, which can be seen in figure 10 in the form of the forecasted volatility being a touch higher than the actual estimates from the data. It's important to notice that this apparent bias is however very small, also thanks to the large amount of data available. Alternatively this problem is reduced when using a larger volatility sampling window. However, this makes this method less suitable for very high frequency operations, as one might require forecasts of volatility over shorter time windows such as one minute.

Then, the fact that the estimator in 8 may produce negative values is certainly an issue worth tackling. One could consider implementing more sophisticated or robust estimators for the Hurst

exponent, otherwise a more thorough analysis on the choice of the hyper-parameters τ_1 and τ_2 should be conducted. We simply attempted a few parameter combinations until the result had mostly positive values.
Finally, the issue of calibrating ν should be considered, as it relies heavily on large amounts of data. As such, this parameter cannot be calibrated as frequently as H. An alternative could be to calibrate ν every few months, once one has enough data for the asymptotic properties of 9 to come into play.

7 Conclusion

In this report, we explored the mathematical background of the RFSV model and it's implications. We then applied the tools of this model to a high-frequency crypto dataset containing information spanning six months of trades. In doing so, we faced the challenges of the data not always conforming to the assumptions on the model, in particularly due to the seasonality of high-frequency volatility estimates, and suggested potential ways to tackle this issue. We then used a combination of estimating techniques to calibrate the model and predict the history of the volatility, obtaining very accurate and reliable results. We then presented the difficulties that we faced in our work, suggesting potential solutions. In conclusion, while the RFSV model has very high predictive power, it faces a number of challenges when applied to frequently traded assets and high-frequency tasks.

References

[1] Binance Data Collection, https://data.binance.vision/?prefix=data/spot/monthly/klines/1INCHBTC/1s/

[2] Volatility is rough; Jim Gatheral, Thibault Jaisson, Mathieu Rosenbaum; September 30, 2017. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2509457

[3] Rough Volatility with Python; Jim Gatheral, May 6, 2016. https://tpq.io/p/rough_volatility_with_python.html

[4] Forecasting with fractional Brownian motion: a financial perspective; Matthieu Garcin, May 20, 2022. https://www.tandfonline.com/doi/abs/10.1080/14697688.2022.2071758