Received: 29 April 2023 | Revised: 24 June 2023 | Accepted: 19 July 2023

DOI: 10.1111/exsy.13414

ORIGINAL ARTICLE

Expert Systems WILEY

A comparison of machine learning and econometric models for pricing perpetual Bitcoin futures and their application to algorithmic trading

Avinash Malik

Department of Electrical, Computer, and Software, Engineering, University of Auckland, Auckland, New Zealand

Correspondence
Avinash Malik, Department of Electrical, Computer, and Software, Engineering, University of Auckland, Auckland, New Zealand.
Email: avinash.malik@auckland.ac.nz

Abstract

Bitcoin (BTC) perpetual futures contracts are highly leveraged speculative trading instruments with *daily* market trading of $45 Billion. BTC perpetual futures are derivative contracts, which depend upon the underlying BTC SPOT (current) price. Pricing perpetual futures fairly is hard, using traditional arbitrage arguments, because of the volatile nature of the so called funding rate, which is used as the replacement of risk free rate in the Cryptocurrency market. This work presents a novel technique for pricing BTC futures contracts using conditional volatility and mean models. Intra-day high-frequency futures' return volatility and mean are modelled using different ML and econometric techniques. A comparison is made using statistical measures to find the model that best captures the intra-day conditional mean and volatility. Exponential generalized autoregressive conditional heteroskedasticity is shown to be an almost unbiased predictor of intra-day volatility, while a constant autoregressive moving average (0, 0) model best captures the conditional mean of the returns. A market directional high frequency trading algorithm is developed using the volatility and mean models. The algorithm first prices the futures contract at some future point of time using the volatility and mean regression models. Next, the slope between the current futures price and the expected price are used to predict the market direction. A long or short position is taken depending upon the expected market direction movement. Extensive back-testing results show absolute returns of 1500%–8000% depending upon the transaction fees and leverage used. On average, the market direction is predicted correctly 85% of the time by the best model. Finally, the trading technique is market neutral, in that it gives large positive returns, with low SD, in both bull and bear markets.

KEYWORDS
Bitcoin, GARCH, HAR, machine learning, pricing futures contracts, random forest, support vector machine, trading, volatility

1 | INTRODUCTION

Bitcoin (BTC) (Nakamoto, 2008) was proposed by 'Nakamoto' in 2008 as a peer to peer decentralized currency. Since its proposal, BTC trading volume and price has grown exponentially (Aalborg et al., 2019). However, BTC was never adopted as a currency for trading goods; rather BTC is

used as a medium of speculation (Griffin & Shams, 2020). The current price of BTC (also called the SPOT price) and futures price on the largest cryptocurrency exchange Binance[1] have significant volume and volatility outflow also affecting other markets (Alexander et al., 2022). The significant volatility and high 24/7 trading volume give a good opportunity for *algorithmic high frequency trading* (*HFT*) using BTC instruments on the Binance exchange.

In general there are two types of algorithmic trading techniques—① market directional and ② market arbitrage (Patton, 2009). Directional trading involves speculating if BTC price will move up or down. If one speculates that the price is going up, then purchasing (also called going long) BTC and waiting for BTC price to go up would result in a profit. Similarly, if a speculator expects that the price of BTC will go down; she can borrow BTC and then sell it in the market (also called going short). Next, wait for the price of BTC to fall and then repurchase the BTC from the market and give back the borrowed BTC, in the process pocketing the price difference. Market arbitrage strategies on the other hand perform arbitrage between price differences of two or more correlated instruments. As an example, it can be the difference between the SPOT price of BTC and the future price of BTC. If the SPOT price is lower than the future (say, the next day') price of BTC. One can buy BTC in the SPOT market and sell a futures contract where the BTC needs to be delivered at a higher price the next day. Arbitrage techniques should make a profit irrespective of whether market goes up or down. However, arbitrage opportunities have become scarce on the cryptocurrency markets (Shynkevich, 2021). This is because of the growing number of institutional traders participating in the BTC market.

The main *contribution* of this paper is to develop a novel *market directional* algorithmic HFT strategy, which has returns similar to arbitrage trading strategy, that is, returns in both bull and bear markets, with lower risk (volatility). In the process of establishing such a trading strategy; a number of technical contributions are made: ① a mean and volatility prediction model is established for BTC perpetual futures market. ② A BTC price direction prediction model is developed using the mean and volatility models. ③ A thorough experimental comparison is performed between econometric and machine learning (ML) techniques to decide on, which prediction model performs best. ④ Experiments are performed over long trading periods to determine, which HFT period is best suited for directional trading in the BTC futures market. Finally, ⑤ transaction costs and their effects are also accounted for in the trading technique.

The rest of the paper is arranged as follows: Section 2 gives the required details for understanding the rest of the paper. Section 3 describes the core of the proposed trading methodology. Section 4 gives the crux of the trading algorithm. Followed by the results in Section 5. Finally, comparison with current trading techniques is performed in Section 6, followed by the conclusions and future improvements in Section 7.

2 | PRELIMINARIES

In this section we give the preliminary information about the trading instrument we work with and its properties.

2.1 | Perpetual futures

A standard futures contract (Jarrow & Oldfield, 1981) is a type of contract between two peers on the cryptocurrency exchange. The buyer of the contract expects the delivery of the underlying BTC at a future expiration date for a *predetermined and negotiated future* BTC price. The seller of the contract on the other hand is obligated to deliver the underlying cryptocurrency to the buyer at the predetermined future price on the contract expiration date. If the SPOT (current) price of the underlying BTC is higher than the predetermined price, on the expiration date then, the buyer can make a profit, by selling the delivered BTC on the open market. If on the other hand, the predetermined price of BTC is lower than the SPOT price, on the expiration date, the seller of the contract makes a profit.

A *perpetual* futures contract (He et al., 2022) is similar to a standard futures contract, except it does not have an expiration date, that is, it is perpetual. Thus, a perpetual futures contract is purely a speculative trading instrument. A trader can buy (or sell) a perpetual futures contract if they expect the price of the underlying cryptocurrency to go up (or down). Selling (or buying) the contract when the price goes up (or down) and pocketing the price difference. Cryptocurrency exchanges employ a process called the 'funding rate' (He et al., 2022), to avoid futures market manipulations and keep the future price of BTC close to the SPOT price. The funding rate requires the short sellers of perpetual futures contract to pay the buyers of the futures contract, if the futures price of BTC is lower than the current SPOT price. Similarly, the buyers of the futures contract will pay the short sellers if the price of futures contract is higher than the current SPOT price. This way the futures price of BTC does not deviate too far from the SPOT price.

Moreover, perpetual futures are highly leveraged trading instruments. A trader can leverage upto 120× the collateral (margin) that they put up. Hence, perpetual futures market have very high liquidity. On Binance alone $45 billion worth of perpetual futures contract are traded *daily*.[2] Finally, in order to increase the trading volume (also termed liquidity) in the futures market, the transaction costs of trading perpetual futures is 10× lower compared to trading on the SPOT market.[3] Overall, perpetual futures market makes a good avenue for *directional market* algorithmic HFT.

2.2 | Properties of perpetual futures

Directional market HFT requires a trader to predict the very short term direction of the market movement. We expect that statistical properties, especially the short term historical volatility would have a large impact on the short term future market direction.

Figure 1 shows the perpetual futures' 'close' price (P_t) of BTC and the logarithmic returns, $r_t = \ln(P_t) - \ln(P_{t-1})$, at any time $t \in \mathbb{Z}$, from 1 January 2020 till 28th of February 2023, sampled at 5 min-frequency. We can see from Figure 1b, *volatility clustering*, that is, high variance of returns are clustered together. We can make use of different econometric and/or ML models to capture these volatility properties for very short market direction prediction in the perpetual futures market. The statistics of returns is shown in Table 1.

As we can see from Table 1, in particular the Jarque Bera test statistic (J. B. stat) (Jarque & Bera, 1980), the returns are *not* normally distributed. Rather we see a very high SD, high left skewness (Ramsey et al., 2002) and very high leptokurtic behaviour (Ramsey et al., 2002), that is, large fat outliers. Skewness and Kurtosis are computed as the third and fourth moments, of return r_t around its mean μ, as shown in Equation (1), with $k = 3$ and $k = 4$, respectively. Finally, augmented Dickey–Fuller statistic (ADF. stat) (Dickey, 2015) shows that the process is stationary. We will utilize these properties in our models for market direction prediction.

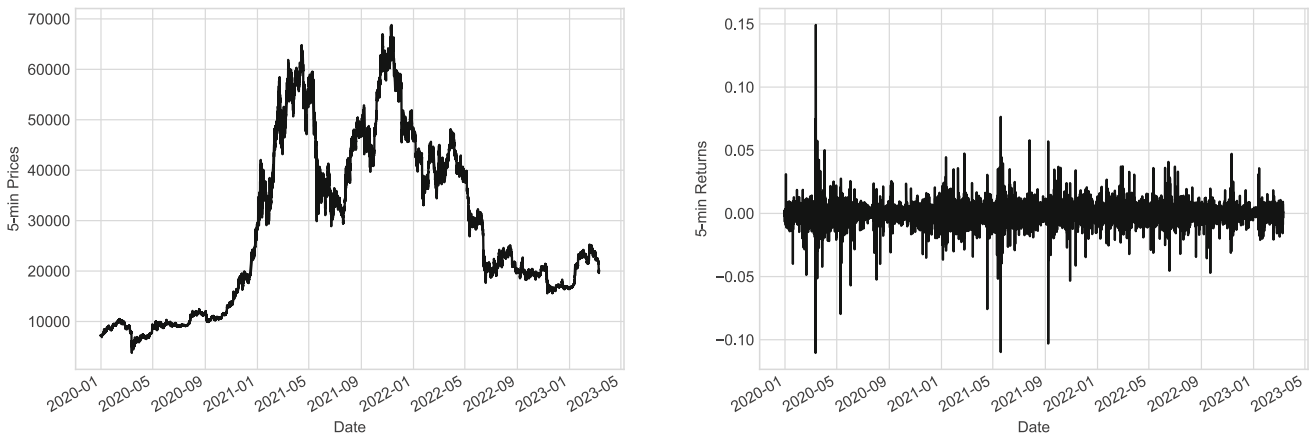
$$\frac{\mathbb{E}\left[(r_t - \mu)^k\right]}{\left(\mathbb{E}\left[(r_t - \mu)^2\right]\right)^{k/2}} \tag{1}$$

3 | METHODOLOGY

Equation (2) shows the different possible cases for profit and loss (P/L) for HFT on the perpetual futures market. If the expected price at some future point time t ($\mathbb{E}[P_t]$) is greater than current price $P_{t-\delta}$, then one can go long futures at $t - \delta$ and wait for the price to increase to sell it at time t to close the position. If the expectation matches reality at time t (case-1, Equation (2)) then we get a profit. Else, we get a loss (case-2, Equation (2)). Similarly, if the expected price $\mathbb{E}[P_t]$ is lower than current price $P_{t-\delta}$ one can go short futures and then repurchase it at time t for a positive return, provided expectations match reality (cases-3 and 4, Equation (2)).

$$r_t = \begin{cases} P_t - P_{t-\delta} > 0, P_t > P_{t-\delta} \wedge \mathbb{E}[P_t] > P_{t-\delta} \\ P_t - P_{t-\delta} < 0, P_t < P_{t-\delta} \wedge \mathbb{E}[P_t] > P_{t-\delta} \\ P_{t-\delta} - P_t > 0, P_t < P_{t-\delta} \wedge \mathbb{E}[P_t] < P_{t-\delta} \\ P_{t-\delta} - P_t < 0, P_t > P_{t-\delta} \wedge \mathbb{E}[P_t] < P_{t-\delta} \end{cases} \tag{2}$$

From Equation (2), we can see that maximizing profit requires the trading algorithm to be built to maximize the objective function in Equation (3). The trader only cares about the overall direction of the price at future time t, *not* the price of the futures contract. This direction is captured in Equation (3), where comparison operators ($=, >, <$) produce a one (true) else produce a zero (false). Given N historical price or return



(a) BTC closing perpetual futures price

(b) BTC 5-min logarithmic return

FIGURE 1 Bitcoin (BTC) futures price and 5-min returns.

TABLE 1 Statistical properties of returns sampled at 5-min frequency in Figure 1b.

# Obs	Mean	Min	Max	SD	Skewness	Kurtosis	J.B. stat	ADF. Stat
335807	0.000003	−0.1104	0.149212	0.002451	−0.2825	140.37	2.75×10^8	−74.82

Abbreviation: Augmented Dickey–Fuller statistic.

observations; the trading algorithm is designed with objective in Equation (3), which captures the fact that *average* number of direction predictions, over a number of historical observations N, not just one direction prediction, are correct.

$$\max\left(\frac{1}{N}\left(\sum_{\delta=1}^{N}\left(sgn\left(\mathbb{E}\left[P_{t-(\delta-1)}\right]-P_{t-\delta}\right)=sgn\left(P_{t-(\delta-1)}-P_{t-\delta}\right)\right)\right)\right)$$
$$sgn(x)=\begin{cases}+1, x>0\\ 0, x=0\\ -1, x<0\end{cases} \tag{3}$$

Our trading methodology follows the objective function (Equation (3)) too. We trade at every time point for the given sampling frequency. For example, for the dataset in Figure 1b, we may trade (open and close positions) every 5 min. At current time $t-1$, we make a prediction for the expected price at time t. Then take the sign of $\mathbb{E}[P_t]-P_{t-1}$, if it is positive, then we go long, if it is negative then we go short, else no trade position is opened. In this work, we do one-step ahead prediction. We predict the price for time t using the observations $t-\delta, \forall\delta\in[1,\ldots,N], N\in\mathbb{Z}^+$. The expected price $\mathbb{E}[P_t]$ is conditional upon the historical observed returns, $\{r_{t-\delta}\}, \forall\delta\in[1,\ldots,N]$ (c.f. Figure 1b). Hence, technically the predicted price would be conditional expectation $\mathbb{E}[P_t|r_{t-\delta}]$.

$$\mathbb{E}[P_t|r_{t-\delta}]=P_{t-1}\times e^{r_t} \tag{4}$$

$$r_t=\mu_t+\sigma_t\times\epsilon, \epsilon\sim N(0,1) \tag{5}$$

$$\mathbb{E}[P_t]=\frac{1}{M}\left(\sum_{i=1}^{M}\left(P_{t-1}\times e^{(\mu_t+\sigma_t\times\epsilon_i)}\right)\right) \tag{6}$$

The conditional expected price, at time t, is given in Equation (4), where the conditional return r_t is itself given in Equation (5). Equation (4) follows from the definition of return $r_t=\ln\left(\frac{\mathbb{E}[P_t]}{P_{t-1}}\right)$. Equation (5), follows from first modelling the observed returns (c.f. Figure 1b) as a stationary mean reverting OU (Uhlenbeck & Ornstein, 1930) process. Second, using the discrete version of the OU process for the given sampling frequency. In Equation (5), μ_t captures the mean, while the σ_t captures the volatility. Hence, for M samples ($\epsilon_i=\{\epsilon_1,\ldots,\epsilon_M\}$), drawn from the normal distribution, we get the expected price of the futures contract at time t from Equation (6).

We now have a technique to trade perpetual futures. We only need a model to accurately compute the mean μ_t and the volatility σ_t. We divide the process of predicting the mean and volatility into two separate models themselves, which are described in the upcoming sections.

3.1 | Predicting the conditional mean—μ_t

In a number of studies, the conditional mean of the returns is assumed to be zero (Aras, 2021) or near zero (Akyildirim et al., 2021; Katsiampa, 2017). In this work μ_t is not assumed to be zero, because of the large SDs and kurotosis observed (c.f. Table 1). Moreover, since we are working with very high frequency data (in minutes) the value of μ_t significantly affects the price prediction. The standard way to predict μ_t is using a AR process (Box, 2013). Equation (7) gives the AR process model, where e_t is the error term. From the autocorrelation of the high frequency, 5-min, returns (c.f. Figure 2); we do not see any particularly large correlation between the return value at any time t and its lags. Hence, a simple moving constant mean fit would suffice. We will use the constant mean model in Equation (8) and grid search (using the objective—Equation (3)) from amongst varying lags $\{2,5,10\}$ to predict the mean μ_t. The constant c, in Equation (8), is the average of the returns for a given lag. The error term e_t is used to model the conditional volatility later in Section 3.2.1. The results will be presented in Section 5.3.1. We test the hypothesis: a smaller window size (lag) gives a better prediction for the mean μ_t.

$$\mu_t = c + e_t + \sum_{i=1}^{p} \psi_i \mu_{t-i} \tag{7}$$

$$\mu_t = c + e_t \tag{8}$$

3.2 | Predicting the conditional volatility—σ_t

Volatility cannot be directly observed (plotted) in the returns of the futures price. Hence, we use a proxy to capture historical realized volatility. We use variance at time t of returns—the square of returns, as the proxy. This proxy comes from the derivation in Equation (9). Note that square of the expected mean of returns (Figure 1b and Table 1) is much smaller than the expected mean of the square of returns (Figure 3a). This is because of the negative values occurring in the returns, but no negative values can occur in the square of returns. Furthermore, using square of returns for capturing historical realized volatility is common in the financial literature (Corsi et al., 2008; Zhang, Zhang, et al., 2022);

$$\begin{aligned} Var(r_t) &= \mathbb{E}\left[(r_t - \mathbb{E}[r_t])^2\right] = \mathbb{E}\left[\left(r_t^2 - 2r_t\mathbb{E}[r_t] + \mathbb{E}[r_t]^2\right)\right] \\ &= \mathbb{E}[r_t^2] - 2\mathbb{E}[r_t]^2 + \mathbb{E}[r_t]^2 = \mathbb{E}[r_t^2] - \mathbb{E}[r_t]^2 = \mathbb{E}[r_t^2] \end{aligned} \tag{9}$$

The autocorrelation of squared returns is shown in Figure 3. Unlike autocorrelation of returns (Figure 2), autocorrelation of volatility (Figure 3) shows a pattern. Values of volatility from $t-\delta, \delta \in \{1,\ldots,N\}$ lags can effect the predicted conditional volatility at time t. Given this observation, we will use autoregressive methods both; econometric and ML based to predict conditional volatility—σ_t.

3.2.1 | Econometric models for conditional volatility

We use two types of econometric models to model and predict conditional volatility: ① general autoregressive models and ② autoregressive models for realized volatility (Table 2). General autoregressive models use the error term e_t from Equation (8), to model and predict future volatility, while realized volatility models use the historically observed square of returns from Equation (9) to model and predict future volatility.

1. Generalized autoregressive conditional heteroskedasticity (GARCH): The standard GARCH model (Bollerslev, 1986) (Table 2) captures the volatility as a standard ARMA process; where ω is the intercept, e_{t-i} are the autoregressive error terms (also called symmetric shocks) from Equation (8), and σ_{t-j} are the moving average lagged volatility terms. In this work we perform a grid search, maximizing the objective

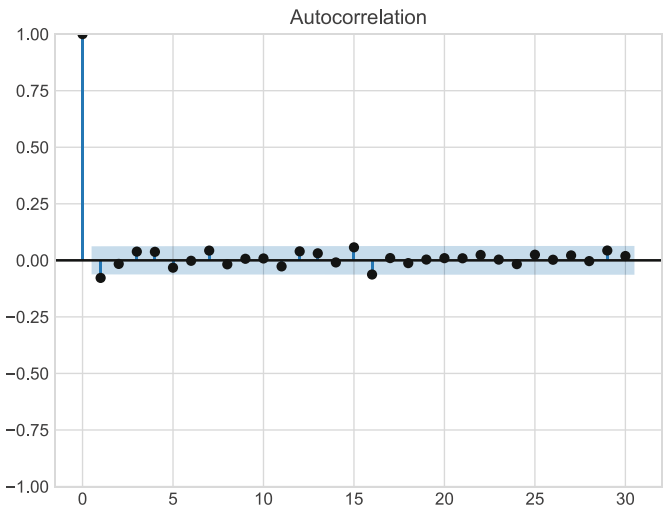
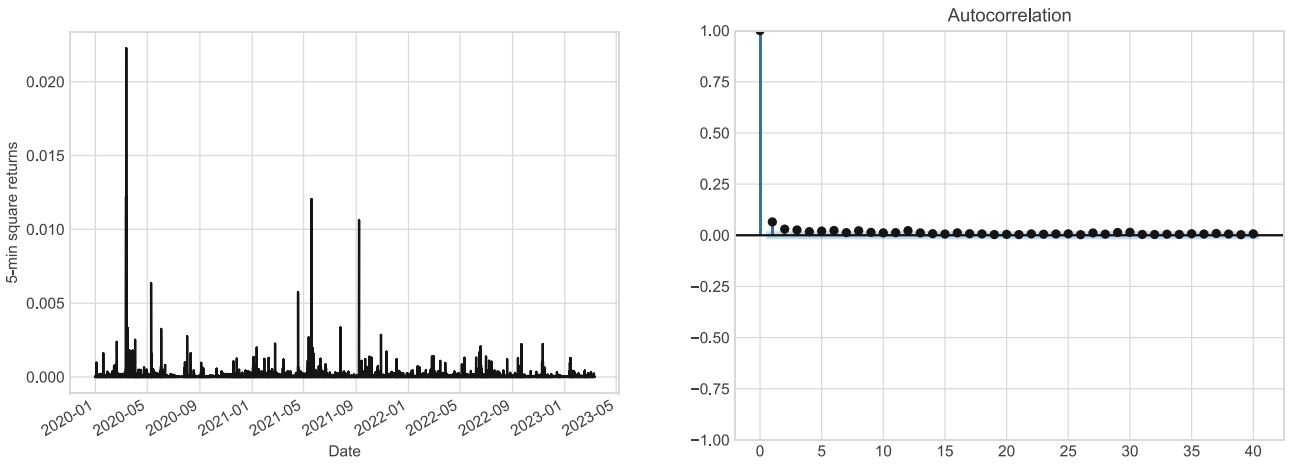


FIGURE 2 Autocorrelation of returns in Figure 1b.



(a) Square ofthe returns in Figure 1b,

(b) Autocorrelation ofthe square of returns

FIGURE 3 Volatility proxy and its autocorrelation.

Equation (3), with p and q ranging in sets $\{0,1,2\}$ and $\{1,2\}$, respectively. The models are fitted using MLE. We use the normal distribution for the MLE fit in all GARCH models and its variants.

2. Exponential generalized autoregressive conditional heteroskedasticity (EGARCH): (Nelson, 1991) (Table 2) is a modification of the GARCH model where α_i captures the size effect, while γ_j captures the sign effect. EGARCH models are able to capture leverage effects in volatility. Leverage effect states that; if volatility is positive (negative) at time point $t-1$, it is expected to be positive (negative) at time point t. The sign effect captures these asymmetric shocks. We perform a grid search, maximizing the objective Equation (3), with p, q, and o ranging in the sets $\{0,1,2\}$, $\{1,2\}$, and $\{0,1\}$, respectively.
3. Asymmetric power generalized autoregressive conditional heteroskedasticity (APARCH): APARCH model (Ding et al., 1993) (Table 2) captures the leverage effect using γ_i and also captures the so called Taylor effect (Taylor, 2008), which states that the autocorrelation of the absolute returns is usually higher than the squared returns. Here again we perform a grid search for p, q, and o in the sets $\{0,1,2\}$, $\{1,2\}$, and $\{0,1\}$. The value of δ ranges between $(0.05,4)$ and is set by MLE along with the model coefficients.
4. Component generalized autoregressive conditional heteroskedasticity (CGARCH): CGARCH (Lee & Engle, 1993) (Table 2) decomposes the conditional volatility into a permanent and transitory component so as to investigate the long and short run movements of volatility affecting returns. Letting q_t represent the permanent component of the conditional volatility. The intercept of the CGARCH model is time varying. Moreover, $\left(\sigma_{t-j}^2 - q_{t-j}\right)$ is the transitory component of the conditional volatility.
5. Heterogeneous autoregressive-realized volatility (HAR-RV): HAR-RV model (Corsi et al., 2008) (Table 2), unlike GARCH models, predicts conditional volatility using squared returns (Figure 3a). It is a linear model, with three different autoregressive coefficients (β_1,β_2,and,β_3) capturing the effects of near term and long term lagged volatility. OLS can be used to fit the coefficients of the HAR-RV model. Here again we perform a grid search for W and M, ranging in the sets $\{2,5,7\}$ and $\{10,20,30\}$.

A moving window with 30 days of observations are used to fit the models described above.

3.2.2 | ML models for conditional volatility

We model conditional volatility using two different ML models: Random forest (RF) (Breiman, 2001) and support vector machines (SVM) (Kj, 2003). We choose these two models out of the plethora of other models, because they have been previously shown to appropriately predict *long term* SPOT BTC prices (Akyildirim et al., 2021; Aras, 2021).

1. RF: RFs build a number of decision trees (Rokach & Maimon, 2005) each predicting the outcome for a regression or classification problem. The average value of outputs of decision trees is used to decide the outcome of the RF. RFs can avoid over-fitting that usually occurs with decision trees. When constructing each decision tree, a bootstrap sample is generated by randomly selecting observations from training dataset with replacements. Using the randomly chosen number of features leads to the correlation between the trees to be low, thereby reducing over-fitting. We predict the square of conditional volatility σ_t^2, using lagged square of returns $r_{t-\delta}^2, \delta \in [1,\ldots,N]$. Each lagged $r_{t-\delta}^2$ is an input feature to

TABLE 2 Econometric models for conditional volatility prediction.

GARCH	$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i e_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$
EGARCH	$\ln \sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \left(e_{t-i} - \sqrt{2/\pi} \right) + \sum_{j=1}^{o} \gamma_j e_{t-j} + \sum_{k=1}^{p} \beta_k \ln \sigma_{t-k}^2$
APARCH	$\sigma_t^\delta = \omega + \sum_{i=1}^{q} \alpha_i \left(e_{t-i} - \gamma_i 1_{[o \geq i]} e_{t-i} \right)^\delta + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^\delta$
CGARCH	$\sigma_t^2 = q_t + \sum_{i=1}^{q} \left(e_{t-i}^2 - q_{t-i} \right) + \sum_{j=1}^{p} \beta_j \left(\sigma_{t-j}^2 - q_{t-j} \right)$ $q_t = \omega + \rho q_{t-1} + \phi \left(e_{t-1}^2 - \sigma_{t-1}^2 \right)$
HAR-RV	$\sigma_t^2 = \beta_0 + \beta_1 r_{t-1}^2 + (\beta_2/W) \sum_{i=1}^{W} r_{t-i}^2 + (\beta_3/M) \sum_{j=1}^{M} r_{t-j}^2$

Abbreviations: APARCH, asymmetric power generalized autoregressive conditional heteroskedasticity; CGARCH, component generalized autoregressive conditional heteroskedasticity; EGARCH, exponential generalized autoregressive conditional heteroskedasticity; GARCH, generalized autoregressive conditional heteroskedasticity; HAR-RV, heterogeneous autoregressive-realized volatility.

TABLE 3 The range of parameters for model tuning.

Model	Parameters	Set of possible values
RF	# Trees (q)	{100}
	min_split_samples (ss)	{2, 5, 10, 15}
	min_samples_leaf (sl)	{1, 5, 10, 15, 30, 50}
	N	{2, 3, 10}
SVM	C	{1, 10, 100, 1000, 5000}
	ϵ	{0, 0.01, 0.001}
	γ	{0.1, 0.01, 0.001, 0.0001, 0.00001}
	N	{2, 3, 10}

Abbreviations: RF, random forest; SVM, support vector machines.

the RF. Thus, N total input features are used to predict conditional volatility at time t. A whole month of previous observations are used for training the RF model. Hence, the total training set $S_m = \left\{ \left(\left[r1_{t-1}^2, .., r1_{t-N}^2 \right], y_1 \right), .., \left(\left[rm_{t-1}^2, .., rm_{t-N}^2 \right], y_m \right) \right\}$ for $m = 30$ days of observations with N lags each. For q decision trees represented as $\left(S_m^{\theta_1}, ..., S_m^{\theta_q} \right)$ and $\hat{h}\left(X, S_m^{\theta_q} \right)$ prediction function with input vector X; the output of the RF is: $(1/q) \sum_{i-1}^{q} \hat{h}\left(X, S_m^{\theta_q} \right)$. We use grid search, maximizing the objective Equation (3), to tune the parameters for RF. The grid search parameters for RF and their bounds are given in Table 3.

2. SVM: Like RF, SVM also predicts the squared conditional volatility σ_t^2 using N lagged values. Moreover, just like RF, a moving window of previous 30 days of observations are used to fit the parameters of SVM. Unlike RF though, SVM does *not* build decision trees. SVM divides the input space using hyperplanes using kernel functions. We use a RBF kernel with ϵ-sensitive loss function to predict volatility. Given the training set $\{(x_1, y_1), ..., (x_m, y_m)\}$, where $x_i = \left[ri_{t-1}^2, ..., ri_{t-N}^2 \right]$ is the lagged square of returns and y_i is the expected output for $i \in \{1, ..., M\}$. SVM solves the optimisation problem in Equation (10).

$$\begin{aligned} &\min_{\alpha, \alpha^*} (1/2)(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + \epsilon \sum_{i-1}^{n} (\alpha_i + \alpha_i^*) + \sum_{i-1}^{n} y_i (\alpha_i - \alpha_i^*) \\ &s.t.\ 1^T(\alpha - \alpha^*) = 0, 0 \leq \alpha_i, \alpha_i^* \leq C \end{aligned} \tag{10}$$

In Equation (10), α and α_i^* are dual variables, K is the RBF kernel with γ coefficient, and $Q_{ij} = K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$. Finally, C is the regularization parameter avoiding over-fitting. We use grid search, maximizing the objective Equation (3), for tuning the SVM. The search bounds of the SVM parameters are given in Table 3.

The range of parameters are chosen to follow the work by Aras (2021) and Akyildirim et al. 2021; as these works have shown fair results and it also helps us in comparing with these works.

4 | SIMULATION ALGORITHM

Algorithm 1 gives the training and trading pseudo-code. The main method (Line 44) starts with taking in very high frequency data, sampled at 1-min, along with the various trading frequencies that we need to explore in order to find the best frequency to trade at.

The re-sampled data is then split into training (in-sample) and testing sets (out-of-sample). The mean and volatility parameters are determined using grid search, maximizing Equation (3), using the training dataset (Lines 49–51). Function paramMeanvolatility function performs the grid search on the parameter space of the given volatility model (Section 3.2) and the mean model (Section 3.1).

Algorithm 1 **Algorithm for Training and Testing/Trading**

```
def trade(data, vm, meanParams, volParams, D=30, leverage=1,
          BET=100, transactionCost=0.012/100):
 N = (D*24*(60/freq)) # Window D (default 30) days
 posOpen = 0
 for t, y in enumerate(data[1:]):
  # Fit models at every time step t
  fvmodel = fitVol(data[t-N:t], volParams) #uses previous D days obs
  fmmodel = fitMean(data[t-meanParams:t]) # uses previous meanParams window obs
  μ_t, σ_t = fmmodel.predict(1), fvmodel.predict(1)
  # Get the predicted return at point t (one step ahead)
  ep_t = Equation(6)(μ_t, σ_t)
  if posOpen: #Close any open positions
   fund += close(transactionCost)
   posOpen = 0 # Long positions are closed by selling, short by buying
  if (ep_t - data[t-1] > 0):
   fund += buy(BET, leverage, transactionCost) # Go Long
   posOpen = 1
  elif (ep_t - data[t-1] < 0):
   fund += sell(BET, leverage, transactionCost) # Go Short
   posOpen = 1

def paramMeanVolatility(vm, train_data, freq, D=30):
 # Get the Cartesian product of model parameters
 N = (D*24*(60/freq)) # Window moves for D (default 30) days
 dir = {} # Dictionary holding direction prediction for each fitted model
 for params in product(vm.param_sets):
  for w in [2, 5, 10]: # Mean model lags Section 3.1,
   # Using moving window fit and one-step ahead lookup (Figure 4).
   for t, y in enumerate(train_data[len(train_data)-N:]):
    # Get the fitted volatility model for given params
    fvmodel = fitVol(train_data[t-N:t], params)
    # Get the fitted mean model (Equation (8))
    fmmodel = fitMean(train_data[t-w:t])
    μ_t, σ_t = fmmodel.predict(1), fvmodel.predict(1)
    # Get the predicted return at point t (one step ahead)
    ep_t = Equation(6)(μ_t, σ_t)
    # Apply (Equation (3)) and store in directory
    dir[params, q] += (sign(ep_t - train_data[t-1]) = sign(y - train_data[t-1]))
 # Get the best average from the directions
 best_index = maxidx(dir.average())
 # Return the best volatility params and mean model lags
 return dir.keys()[best_index]

def main(orig_data, freqs):
 for f in freqs:   # Re-sample data for the given user frequencies
  sdata = orig_data.resample(f)
  # Split the dataset into testing data and training data
  test_data, train_data = split(sdata)
  for vm in vModels:
   # Get the mean and volatility parameters fitted from grid search
   mean_params, vol_params = paramMeanVolatility(vm, train_data, f)
   trade(test_data, vm, mean_params, vol_params) # Trade on test data
```

Function paramMeanVolatility iterates through the Cartesian product of the parameters of the given volatility model. For example, the GARCH model has the parameter sets $\{0,1,2\}$ and $\{1,2\}$ for p and q, respectively. Hence, the Cartesian product $\{(0,1),(0,2),(1,1),(1,2)\}$ is built for (p,q). Same for all other volatility models, both econometric and ML, in Section 3.2. The function also iterates over the window size w for the mean model (Line 27).

A moving window technique (Figure 4) is used to fit the parameters and perform grid search (Lines 31–38). The last D, default 30 days worth of samples, are used to perform grid search. For a 5-min sampling frequency this would entail $N = 30 \times 24 \times (60/5) = 8640$ samples being used to find best parameters for a given model. The return model is divided into two parts (Equation (6))—conditional volatility and conditional mean. Both models are fit (Lines 31 and 33). One step-ahead prediction starting from the first sample in the last D days at time t is performed (Line 36). Finally, Equation (3) is used (Line 40) to return the best fitted volatility and mean parameters for the given volatility and the mean model.

The function trade (Line 1) performs trade using out-of-sample testing dataset. The function performs a trade at every sampled time point t. The volatility and mean models are fitted for the given parameters (Lines 7–8) at every trading time t. If the predicted price ep_t is higher than the current price then, a long position is opened, else a short position is opened (Lines 15–20). A \$100 bet is placed every time, with a default leverage of 1, that is, with no leverage. The transaction costs are also accounted for. Finally, if an open position exists, then at point $t+1$, the position is closed by reverting the open position (Line 13).

The result of running the training and trading algorithm, with different sampling frequencies, different transaction costs, and so forth will be presented in the next section.

5 | EXPERIMENTAL RESULTS

In this section we describe the outcomes of applying the training and trading algorithm (Algorithm 1) applied to out-of-sample dataset. First the experimental setup, including the dataset, the different experiments carried out are described. Next, the results are presented.

5.1 | Setup

The dataset we use is 1-min sampled BTC futures traded contracts obtained from Binance.[4] We re-sample data (Algorithm 1 line 45) at frequencies of 5, 15, 30, and 60 min intervals to investigate, which frequency performs best in terms of trading outcomes (Section 5.3). Our dataset ranges from 1 January 2020 to 28 February 2023. We split this dataset into two parts: ① dataset from 1 January 2023 to 31 February 2023 is used for training and fitting the model parameters (Section 5.2). The out-of-sample testing dataset from 1 February 2023 to 28 February 2023 is then used to investigate the performance of different models in terms of trading outcomes. The trading/prediction best model obtained from this test is then used for back-testing on long scale dataset from 2020 to 2023; including in bull and bear markets (Sections 5.3.4 and 5.3.5). A number of statistical measures are used to test various hypothesis on the testing outcomes.

5.1.1 | Transaction costs and leverage

We include transaction costs in all our tests. Binance has a number of different transaction costs depending upon the transaction fees generated for Binance over the last trading month.[5] We use the *limit order* fees of 0.012% for experiments, except when stated otherwise (Section 5.3.5). Similarly, we do not use leverage for experiments, except when stated otherwise (Section 5.3.5).

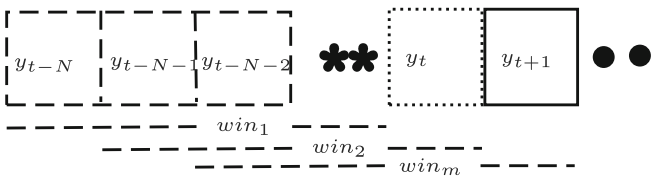


FIGURE 4 Moving window on-step ahead fitting. Flowers represent the input vectors. Dots represent the next output in grid search training dataset. The dashed boxes are the inputs to generalized autoregressive conditional heteroskedasticity (GARCH) or machine learning (ML) models following Section 3.2. Dotted box shows the output used for fitting at point t. However, for prediction at time $t+1$, the observed value of y_t is used as an input; *not* the predicted value.

5.1.2 | Handling erroneous model prediction

It is well known that econometric models can predict erroneously small or large outcomes (Bollerslev et al., 2016; Clements & Preve, 2021). We apply a simple filter in such cases; if the prediction is very small or large, we simply drop that outcome and do not perform a trade. More precisely, if the erroneous prediction happens for time t, then the slope of $\mathbb{E}[P_t] - P_{t-1}$ is equal to 0.

5.2 | Best parameter fits for prediction models

The first step of our algorithm is to fit the model parameters (Algorithm 1 line 51) via grid search. We perform the grid search for all our volatility models on four different sampled frequencies. The result of grid search is shown in Table 4 and Table 5. As stated in Algorithm 1, line 38, the direction of prediction is used as the objective function. The result of direction prediction for the best parameter fits for each of the model is shown in Table 4 and Table 5.

TABLE 4 Best parameters found after grid search for various sampling frequencies for training dataset from 1 January 2023 to 31 January 2023.

(a) Best parameters for models, sampled at 5-min frequency							
	GARCH	EGARCH	APARCH	CGARCH	HAR-RV	RF	SVM
p	2	0	1	2			
q	2	2	2	1			
o		1	1				
D					1		
W					5		
M					10		
ss						15	
sl						30	
C							10
ϵ							0.01
γ							0.001
N						2	2
Equation (3)	0.703	0.854	0.726	0.727	0.729	0.729	0.731
(b) Best parameters for models, sampled at 15-min frequency							
	GARCH	EGARCH	APARCH	CGARCH	HAR-RV	RF	SVM
p	2	0	1	1			
q	1	2	2	1			
o		1	1				
D					1		
W					2		
M					20		
ss						15	
sl						1	
C							1000
ϵ							0
γ							0.001
N						2	2
Equation (3)	0.698	0.857	0.77	0.7269	0.715	0.712	0.713

Note: The parameters not applicable to certain model are left empty.
Abbreviations: APARCH, asymmetric power generalized autoregressive conditional heteroskedasticity; CGARCH, component generalized autoregressive conditional heteroskedasticity; EGARCH, exponential generalized autoregressive conditional heteroskedasticity; GARCH, generalized autoregressive conditional heteroskedasticity; HAR-RV, heterogeneous autoregressive-realized volatility; RF, random forest; SVM, support vector machines.

As we can see, from Table 4 and Table 5, EGARCH is the best performing model in terms of direction prediction. It is also parsimonious, with its parameters remaining constant for any given sampling frequency. On average EGARCH is able to predict the correct direction 85% of the times. The other models on average give a correct direction prediction ≈75% of the time. However, the parameter values keep on changing with the sampling frequency.

These direction predictions include the fitted mean model (Equation 8) with a lag window of size 2. A more through investigation of the mean model lag size is presented later in Section 5.3.2. Finally, the number of samples used for fitting depends upon the sampling frequencies. For example, for the 5-min sampling frequency, the total dataset from 1 January 2023 to 31 January 2023 includes: $(31 \times 24 \times (60/5)) = 8928$ samples.

The primary reason for the out performance of EGARCH over other models is so called *leverage effects*. In case of BTC futures; first of all we have volatility clustering (Figure 1b). Moreover, in these clusters, when volatility is increasing at point $t-1$, it continues to increase at point t and vice-versa. This fact is captured by the sign of volatility (parameter γ) in the EGARCH model (c.f. Table 2). APARCH also captures this leverage effect. However, the EGARCH model outperforms APARCH to a very large extent.

Given these fitted parameters we now show the results of trading (Algorithm 1) on out-of-sample data.

5.3 | Trading results

This section presents the results of trading on out-of-sample data. A number of hypothesis are tested using statistical tests to validate the efficacy of the models.

5.3.1 | Best trading frequency

The very first question we want to answer is: *What frequency is best for trading?*. The fitted models from Section 5.2, are executed on out-of-sample dataset from 1 February 2023 to 28 February 2023, sampled at different frequencies, to answer this question. The process follows Algorithm 1 lines 1–20. First, an expected price ($\mathbb{E}[P_t]$) prediction for next time point t is made. Next, the expected direction is computed using $\mathbb{E}[P_t] - P_{t-1}$. If the slope is positive, then a long order is made by buying \$100 worth of futures contract. If the slope is negative then a short order is made by selling \$100 worth of futures contract. Else, no position is taken. In the next step, the open position is closed by reversing the orders and a new position is opened. Every order (long or short) consumes a transaction fees of 0.012% (Table 5).

The result of trading is shown in Figure 5. Once again the EGARCH model outperforms every other model for any given sampling frequency. The top panel in all sub-figures shows the percentage profit and loss (P/L) while the bottom panel shows the absolute rise or fall in funds. All funds start with \$1000. The best performance amongst all the sampling frequencies is for 5 min. As the sampling frequency decreases the total P/L decreases. This is expected, because there are fewer trading opportunities with a longer sampling period.

The trading model has two components; the mean model (Section 3.1) and the volatility model (Section 3.2). We investigate each of these models separately.

5.3.2 | Best fit for the mean model

The mean model (Equation 8) is common to all trading techniques. The mean model is a constant model, which is fitted using lagged observation (Section 3.1). We want to answer the question: *What is a statistically good lagged window size to fit the mean model.* The answer to this question in presented in Figures 6 and A1.

Figure 6a shows a 3D graph for the trading results with different lagged window sizes used for fitting the mean model with volatility captured using EGARCH. The X, Y, and Z axes show the total P/L (in percentage), the so called total *overall wins*—a positive P/L, after accounting for transaction costs, and total direction wins, respectively. The first thing to note is that a shorter lagged window size used for fitting the mean model generally gives a better result. An interesting point of note is that; a high number of directional wins (Equation 3) does not necessarily entail a positive P/L. This is because, certain parameter values for the volatility model might give a number of erroneous predictions, which results in very few trades. In turn, resulting in very low total P/L. Hence, the overall trading objective function should maximize directional wins, the overall wins (after transaction costs), and the total P/L together. Such an objective function can be constructed by multiplying the three values together. The mean model fit with lagged window of 2, and EGARCH volatility model with $p = 0$, $q = 2$, and $o = 1$ gives the best results considering all these three outcomes.

This observation is further re-enforced by carrying out a Welch t test (Figure 6b) between the outcomes for lagged windows. This t test tests for the null hypothesis (H0) that the expected total P/L multiplied with the overall wins, multiplied by the direction wins, for different lagged

TABLE 5 Best parameters found after grid search for various sampling frequencies for training dataset from 1 January 2023 to 31 January 2023.

(a) Best parameters for models, sampled at 30-min frequency							
	GARCH	EGARCH	APARCH	CGARCH	HAR-RV	RF	SVM
p	2	0	2	2			
q	2	2	2	2			
o		1	1				
D					1		
W					2		
M					10		
ss						2	
sl						5	
C							100
ϵ							0.01
γ							0.01
N						2	2
Equation (3)	0.719	0.851	0.784	0.755	0.728	0.731	0.734
(b) Best parameters for models, sampled at 60-min frequency							
	GARCH	EGARCH	APARCH	CGARCH	HAR-RV	RF	SVM
p	1	0	2	2			
q	1	2	2	2			
o		1	1				
D					1		
W					5		
M					30		
ss						2	
sl						50	
C							1
ϵ							0.01
γ							0.1
N						2	2
Equation (3)	0.72	0.823	0.775	0.727	0.691	0.727	0.731

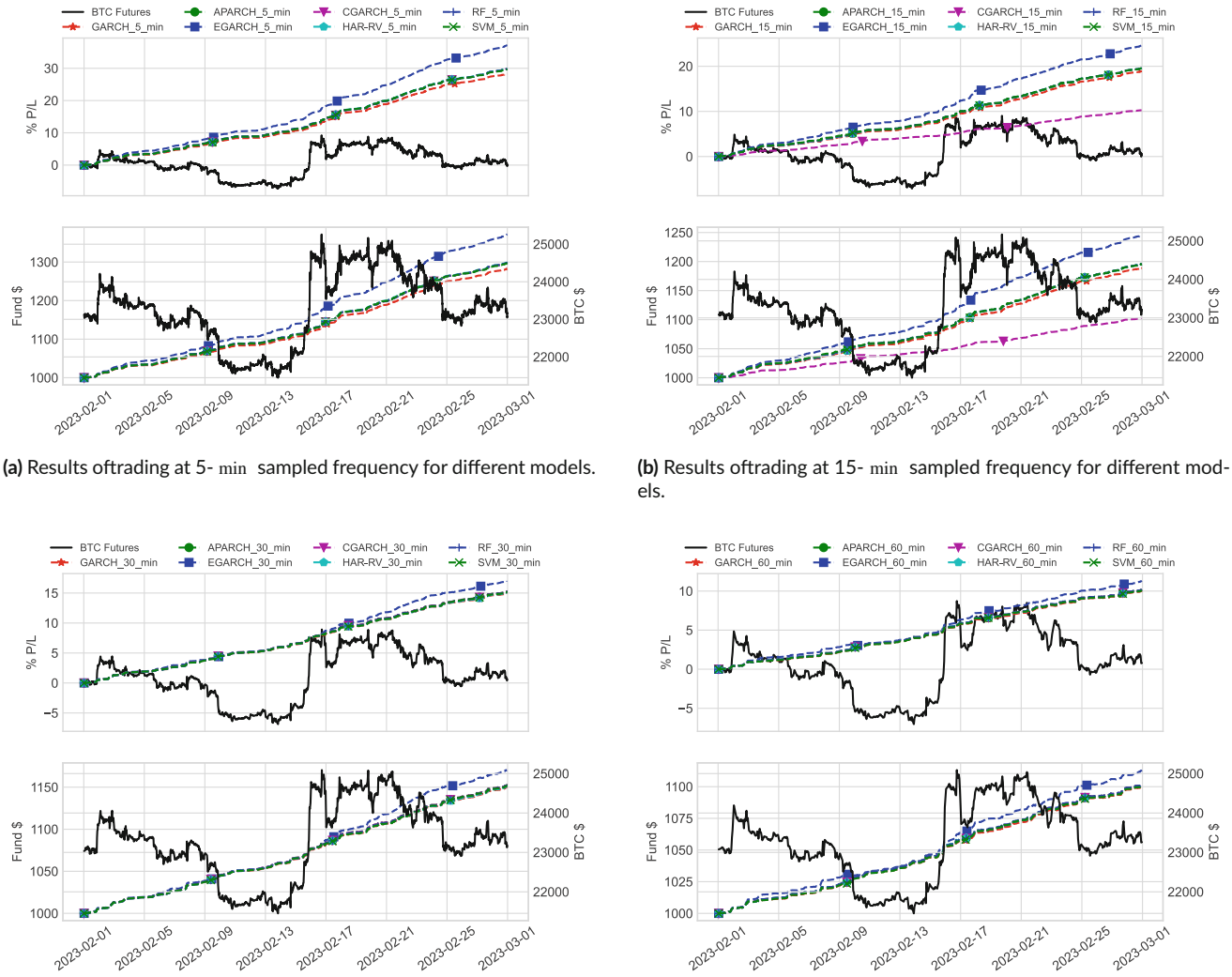
Note: The parameters not applicable to certain model are left empty.
Abbreviations: APARCH, asymmetric power generalized autoregressive conditional heteroskedasticity; CGARCH, component generalized autoregressive conditional heteroskedasticity; EGARCH, exponential generalized autoregressive conditional heteroskedasticity; GARCH, generalized autoregressive conditional heteroskedasticity; HAR-RV, heterogeneous autoregressive-realized volatility; RF, random forest; SVM, support vector machines.

window sizes is the same. The t-statistic along with the *p* values, in brackets, shows that the expected value with a shorter lagged window size of 2 is significantly different to lagged window size of 5 and 10.

5.3.3 | Unbiased predictor for high frequency volatility

We have seen that a small lagged window size for the mean model gives better trading results. The other component of the price/direction prediction model is volatility. Here in we investigate the hypothesis: *Is the best fitted EGARCH model an unbiased predictor of high-frequency volatility of BTC futures returns*.

In order to test for this hypothesis, a linear model of the form $(\text{Real volatility}) = \beta_0 + \beta_1(\text{Predicted volatility})$ is fit for every time point *t*, for the given fitted EGARCH model with $p = 0$, $q = 2$, $o = 1$. Real volatility is the square of logarithmic returns (Equation (9)) (Zhang, Zhang, et al., 2022). If the fitted EGARCH model is a perfect unbiased predictor then, the value of β_0 should be zero and β_1 should be one.



(a) Results oftrading at 5- min sampled frequency for different models.

(b) Results oftrading at 15- min sampled frequency for different models.

(c) Results oftrading at 30- min sampled frequency for different models.

(d) Results oftrading at 60- min sampled frequency for different models.

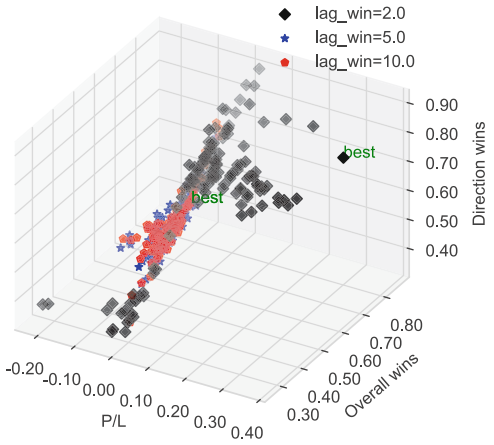
FIGURE 5 Trade results on out-of-sample dataset from 1 February 2023 to 28 February 2023. Every order has a transaction cost of 0.012%. Bet $100 every trade. No leverage.

The results of predicted conditional volatility along with the real volatility is shown in Figure 7. The results for the hypothesis test are shown in Table 6. The fitted EGARCH model is almost an unbiased predictor, but it is not perfect. We attribute this to the fact that there are some very large jumps (c.f. Figure 7), which are not captured correctly by the volatility model. More precisely, the height of the jump is not predicted correctly. Hence, in the future it might be prudent to modify the EGARCH model with jump diffusion process (Zhang, Chen, & Peng, 2022). However, note that even with the lack of the jump diffusion process; the predicted volatility from the EGARCH model has a low MAE (c.f. Table 6) and the trading performance is quite good.

Now that we have a good mean and volatility model along with evidence that it is best to trade at a 5-min frequency; we will see the performance of this model over very long term trading intervals—this is termed back-testing in the trading literature.

5.3.4 | Results in bull and bear markets

First we test the fitted prediction model on Bull and Bear markets (Pagan & Sossounov, 2003) with sampling frequency of 5 min. A Bull market is one where the BTC futures price is on an up-trend. A Bear market is one where the BTC futures price is on a down-trend. We choose to test the prediction and trading model on these two scenarios, because one of our objective is to get market neutrality, that is, irrespective of the overall trend of the BTC market we want to make a profit.



(a) EGARCH model results for different window sizes for μ_t and all parameters values for EGARCH used in grid search. The best result (best P/L × best overall wins × best direction wins) for different window sizes is marked as "best" Notice that the black cluster (lag/window size 2) size performs much better in general then the blue (lag/window size 5) and red clusters (lag/window size 10). The black cluster has on average higher directional wins, higher overall wins, and a larger P/L.

	5	10
2	8.27 (pval: 7e-14)	8.47 (pval: 2.3e-14)
5		1.04 (pval: 0.29)

(b) EGARCH model, H0 results.

FIGURE 6 Welch t-statistic and *p* values (in brackets) for H0: Expected (total P/L × overall wins × direction wins) are same. Comparing different lags for μ_t. Sampled at 5, 15, 30, and 60 min (s). Transaction costs of 0.012% for every order. Trading on out-of-sample dataset from 1 February 2023 to 28 February 2023. Bet $100 every trade. No leverage.

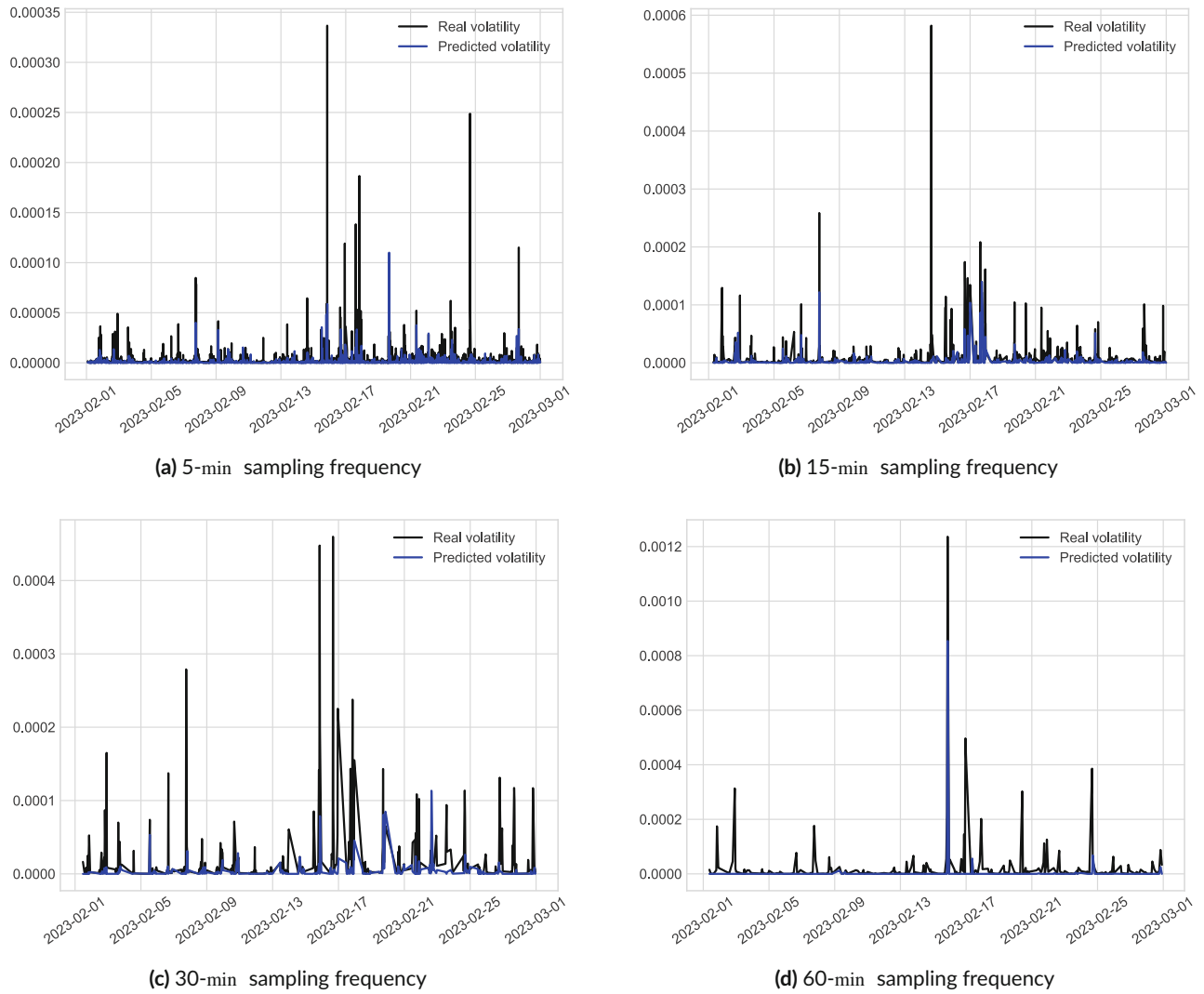
Figure 8 shows the trading result on Bull and Bear markets. BTC had a Bull market from 1 January 2020 to 1 March 2021, and a Bear market trend started from 1 November 2021 to 1 January 2023. Same as before, we trade at every 5-min interval. Open a long (short) trade if expected market direction is up (down), closing the position every 5 min too by reversing the trades. The transaction cost for every order (long or short and its reversal) is 0.012%. We bet $100 every time without leverage.

As we can see, the proposed trading technique produces a profit under both Bull and Bear markets. The trading metrics are shown in Table 7. Total number of positions (Total Pos) taken over both market cycles is greater than a 100,000. Note that this means over 200,000 orders were placed. Since every position includes two orders opening and closing the position to realize the profit and loss (P/L). The algorithm takes twice as many long position (Long Pos) as shorts (Short Pos). However, short positions have a 96% win ratio (Shorts Won). After considering transaction costs, only 72% of the total positions (Overall Trades Won) give a positive P/L—these are the overall wins (c.f. Figure 6a). The algorithm is likely to win twice as many times (after accounting for transaction costs) as lose (Win/Loss ratio). This highlights that the BTC futures market is highly inefficient. More precisely, the futures contracts are *not* priced fairly and the proposed technique exploits these inefficiencies to generate a positive P/L.

On average (avg. win), the technique generates a few cents every 5 min in P/L. Similarly, the average loss (avg. loss) is also a few cents. The maximum win (max. win) and maximum loss (max. loss) are large compared to the average wins and loss. The absolute return (Strat. Abs. Ret) in both markets eclipses the buy and hold strategy (BH-Bull/BH-Bear). Similarly, the annualized return (Strat. Ann. Ret) and the risk adjusted Sharpe ratio (Sharpe, 1998) (Strat. Ann. Sharpe) is very high. Sharpe ratio calculate the excess return given by the trading strategy over risk free investment, for example, in US T-bills. Sharpe ratio is computed as shown in Equation (11). In Equation (11), we adopt a risk free annualized return of 4.5% and Std. Dev. Ret is the SD of our return curve (c.f. Figure 8), which is very small resulting in very large Sharpe ratios.

$$SR = (\text{Strat. Ann. Ret} - \text{Risk Free Ret.})/(\text{Std. Dev. Ret}) \tag{11}$$

Finally, two metrics comparing the out-performance of the proposed trading strategy vis-á-vis the Buy–Hold strategy are α (Strat Ann. α) and β (Strat β). The α gives the percentage excess return of the strategy over the Buy–Hold benchmark and β gives the correlation of the strategy returns with the Buy–Hold benchmark returns. These values are obtained by regressing the generated 5-min returns over the Buy-Hold returns. A larger value of α means a better (more profitable) trading strategy. A positive value of β means the trading strategy is very highly correlated with the benchmark. A zero or negative value of β means the trading strategy is uncorrelated with the benchmark. Our trading strategy has a zero correlation with the Bull market and a negative correlation with the Bear Market, which shows the market neutrality of the proposed strategy.



(a) 5-min sampling frequency

(b) 15-min sampling frequency

(c) 30-min sampling frequency

(d) 60-min sampling frequency

FIGURE 7 Predicted conditional volatility versus real observed volatility for the exponential generalized autoregressive conditional heteroskedasticity (EGARCH) model for different sampling frequencies.

5.3.5 | Long term trading with and without leverage

In this section we carry out back-tests over the complete dataset, sampled at 5-min frequency. However, this time the transaction costs vary along with leverage.

The back-test is carried out on dataset from 1 January 2020 to 12 February 2023, two different transaction costs are used: 0.012% for every order and 0.03% for every order. These two transaction costs account for limit and market orders, respectively. A limit order is one, where an order is placed and it does not match immediately resulting in increased market liquidity. A market order is one where a futures contract is sold (bought) at whatever current price is available in the market. Hence, it reduces market liquidity.

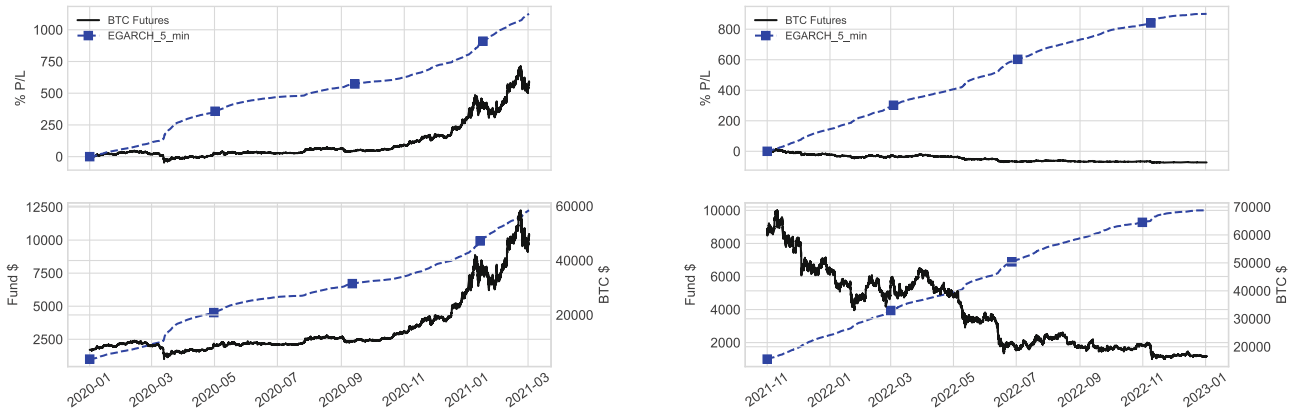
The results of the back-tests with different transaction costs along with 1× and 3× leverage are shown in Figure 9. As expected, a higher transaction cost, results in substantially low positive P/L (c.f. Figure 9a,c). Almost half the profit is lost paying transaction fees. Hence, a limit order based trading technique is preferable. However, this requires one to continuously monitor the order book to keep tabs on the incoming orders. Real-time access to deep order books is usually very expensive. Market orders still generate a profit even without leverage (c.f. Figure 9c).

Taking leverage (c.f. Figure 9b,d) substantially improves the positive P/L. However, one needs to be careful that large orders do not result in changing the market direction itself. The trading metrics for Figure 9 are given in Table 8. Similar to the Bull and Bear market returns; the proposed trading strategy generates excess profits over the Buy–Hold (BH) strategy. Moreover, with very high Sharpe ratios. On average, the trading models (mean and volatility) are able to correctly predict the future price of BTC contracts for very HFT.

TABLE 6 Fit values and metrics for $(\text{Real volatility}) = \beta_0 + \beta_1(\text{Predicted volatility})$, Figure 7.

	5-min	15-min	30-min	60-min
β_0	2.1e-6	5.45e-6	1.6e-5	2.17e-5
	(SE: 1.7e-7)	(SE: 8.5e-7)	(SE: 2.6e-6)	(SE: 4.02e-6)
β_1	0.9126	1.45	0.9461	1.4115
	(SE: 0.05)	(SE: 0.091)	(SE: 0.228)	(SE: 0.073)
Adj. R^2	0.089	0.209	0.042	0.61
MAE	2.3e-6	7.3e-6	1.79e-5	2.4e-5

Abbreviations: MAE, mean absolute error; SE, standard error.



(a) Bull market results

(b) Bear market results

FIGURE 8 Trading results of the fitted exponential generalized autoregressive conditional heteroskedasticity (EGARCH) and mean model in bull and bear markets. Transaction cost of 0.012% for every order. Bull market trade on dataset from 1 January 2020 to 1 March 2021. Bear market trade from 1 November 2021 to 1 January 2023. Bet $100 on every trade without leverage. Dataset sampled at 5-min frequency. All Funds start with $1000.

6 | RELATED WORK AND DISCUSSION

Futures are derivative contracts, that is, the price of a futures contract depends upon the underlying SPOT price of the basic instrument such as BTC. Futures are usually priced using so called arbitrage arguments (Hull, 2003). A fair price for a perpetual BTC futures contract is also based on such arbitrage arguments (He et al., 2022). For example, given a risk free rate of r% per-annum and the price of the perpetual BTC futures contract greater than the BTC SPOT price $\times e^{rT6}$ at some future point T. Then, the trader can perform an arbitrage by borrowing the cash (at rate r%); purchasing the BTC in the SPOT market and immediately selling the futures contract in the futures market for the purchased BTC. Thereby, instantaneously locking in a profit. The risk free rate to use for fairly pricing a futures contract, so that no such arbitrage opportunities exist, for a novel instrument such as BTC is ambiguous. In fact, the risk free rate (also called the funding rate in the Cryptocurrency world) depends upon the price difference of BTC in the SPOT and futures market. Since BTC is highly volatile, the funding rate is also very volatile. In essence, this work captures this funding rate using ML and econometric techniques modelling volatility.

This is not the first work to capture volatility of BTC. A number of other works (Aras, 2021; Basher & Sadorsky, 2022; Chi & Hao, 2021; Katsiampa, 2017) also use econometric, ML, and combination of these two to model the volatility of BTC. However, most of these works are targeted at modelling and predicting the volatility of the daily (also called inter-day volatility) BTC SPOT market rather than high frequency intra-day volatility of the perpetual futures market like we do. Moreover, many of these other works assume the mean of the return to be zero. Making this assumption works well for inter-day long time pricing. However, fails for pricing very short high frequency futures contracts. Finally, these techniques are targeted at exploiting options trading based arbitrage opportunities, in contrast to the proposed trading technique, which is market directional.

The work by Akyildirim et al. (2021) uses ML algorithms to predict the price of BTC futures contracts. This work also targets price prediction for high-frequency intra-day trading. However, unlike the proposed approach, where the ML and econometric models model the volatility as a regression process. Their work (Akyildirim et al., 2021) does future price prediction as a classification problem. A number of ML algorithms are explored such as logistic regression, RF, SVM, KNN, and so forth, which take the previous time point's futures prices as input and then predict a

TABLE 7 Trading metrics for bull and bear markets (Figure 8).

	Bull (1 Jan 2020–1 Mar 2021)	Bear (1 Nov 2021–1 Jan 2023)	BH-Bull	BH-Bear
Total Pos	118,470	118,627		
Longs Pos	73,931	73,807		
Short Pos	44,113	44,393		
Longs won (%)	78.47	77.84		
Shorts won (%)	96.85	96.63		
Total direction wins (%)	85.34	84.9		
Overall trades won (%)	72.82	71.19		
Win/loss ratio	2.6	2.47		
Avg. win ($)	0.15	0.13		
Avg. loss ($)	−0.072	−0.06		
Max. win ($)	16.06	4.8		
Max. loss ($)	−3.71	−3.9		
Strat abs. ret. (%)	1125.37	900.34	590.72	−72.93
Strat ann. ret. (%)	755.92	616.02	423.75	−67.277
Strat ann. Sharpe	320.08	334.788	4.71	−1.06
Strat ann. α (%)	217.56	199.13		
Strat β	0.0006	−5.34		

Note: No leverage, transaction costs 0.012% for every order.

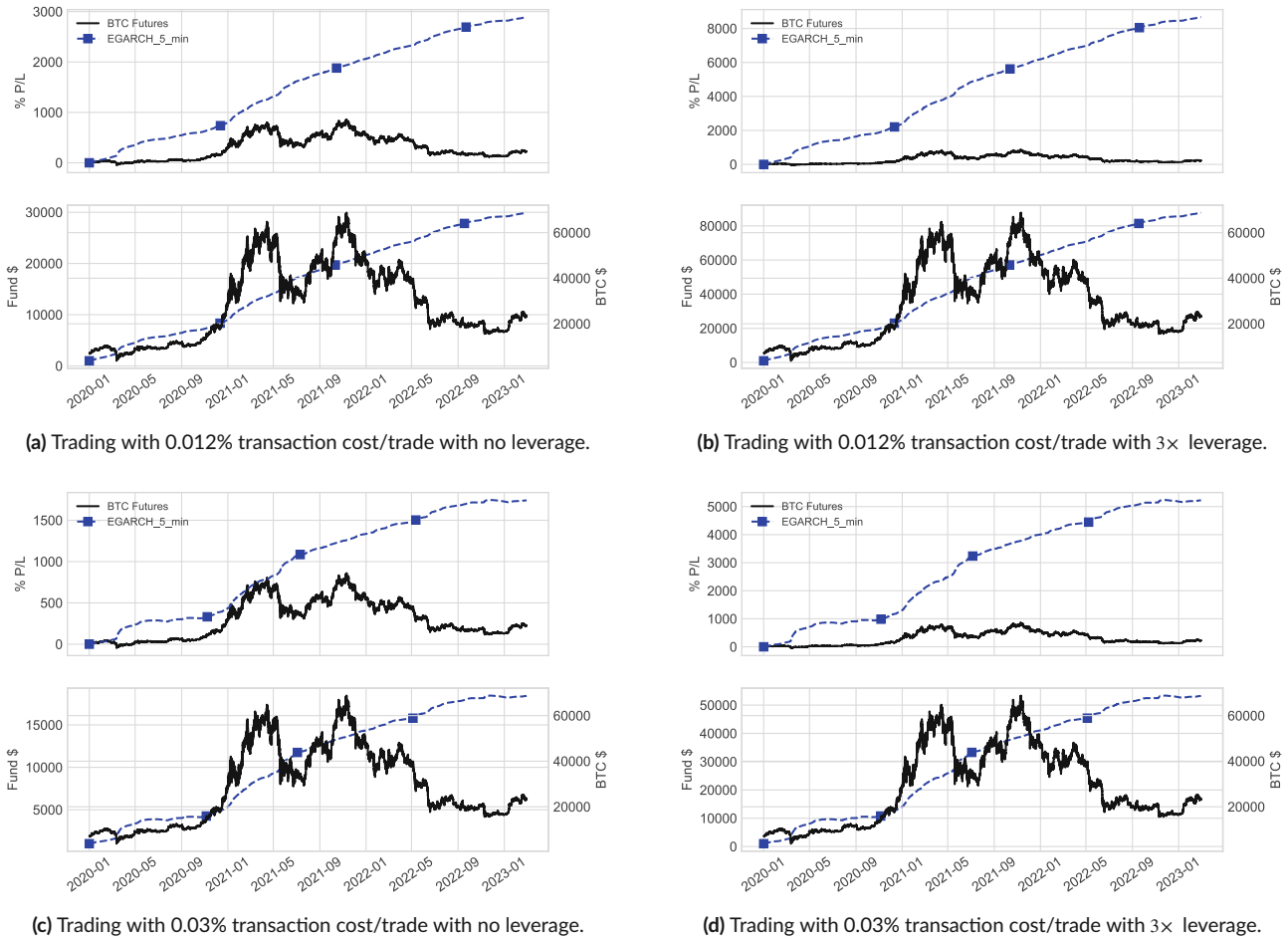
1 (for up) and 0 (for down) for the next time point. The best out-of-sample results for this classification based technique is in the range of 50% accuracy for the RF model. This is in stark contrast to the proposed regression based technique, where we use ML and econometric, regression, models to first predict the volatility and then use the predicted next time point volatility (Section 3.2) along with the predicted conditional mean, via the mean regression model (Section 3.1) to predict the futures market direction. Our best results are for the EGARCH volatility model with ≈85% accuracy. Moreover, even the RF and SVM volatility models result in ≈73% accuracy (c.f. Figures 6a and A1). Hence, overall the proposed technique clearly outperforms the work by Akyildirim et al. (2021).

There are range based volatility models; such as those designed by Chou (2005). These models do *not* work on a return time series (c.f. Figure 1b); rather on the range or price differences of an asset for a given time period. For example, if we consider the start of trading at 9:00 AM and the close of trading day at 4:00 PM, then the difference between the maximum and minimum price of an asset during this trading period would be a single point in the range time series. We can do this for a number of days to get a range time series. Then we can model the range time series using GARCH (or its variant) to predict volatility. Such a range based volatility model is good for pricing options, because closed form formulae exist for pricing options (Black & Scholes, 1973). However, these models cannot be used for pricing Cryptocurrency futures, because of multiple fundamental reasons: ① Cryptocurrency futures are traded continuously, there is no defined start and end of a trading period. Hence, one would need to choose an arbitrary start and end trading period. ② The range based volatility model, does not give the return volatility. Hence, it is not obvious how the future should be priced from the predicted range based volatility, unlike options. ③ In our particular case, we are trading at a very high frequency (in minutes). Hence, we simply do not have enough number of time points within our trading period to get a valid range. As commented by Chou (2005), the greater the number of time points between the start and ending of trading period, the better the volatility prediction. Hence, we cannot use range based models in this work.

In general, Cryptocurrency trading is targeted more towards arbitrage trading rather than market directional trading. A good overview of different trading techniques in the Cryptocurrency market is provided in Fang et al. (2022). One of the most profitable arbitrage trading technique in the early days of the Cryptocurrency market was Geographical arbitrage trading. BTC in the Korean and Japanese SPOT markets was extremely over-priced. Thus, traders took the opportunity of buying BTC in the SPOT market, say in the United States, and immediately selling it in the Korean/Japanese markets thereby reaping profits greater than 80% (Makarov & Schoar, 2020). Such arbitrage trading opportunities have now become scarce. The proposed trading strategy, uses market direction prediction, to generate substantial returns in Bull, and Bear markets.

7 | CONCLUSION AND FUTURE WORK

Perpetual BTC futures contracts are speculative trading instruments, where traders buy and sell contracts speculating on market direction. Today $45 Billion worth of contracts are traded daily on the most popular Cryptocurrency exchange: Binance. High leverage and low transaction fees



(a) Trading with 0.012% transaction cost/trade with no leverage.

(b) Trading with 0.012% transaction cost/trade with 3× leverage.

(c) Trading with 0.03% transaction cost/trade with no leverage.

(d) Trading with 0.03% transaction cost/trade with 3× leverage.

FIGURE 9 Long term results of trading from 1 January 2020 to 28 February 2023, sampled at 5 min. Bet $100 every time. All funds start with $1000. P/L stands for profit and loss.

TABLE 8 Trading metrics for long term trading, with different transaction costs and with or without leverage (Figure 9).

	TC-0.012	TC-0.03	TC-0.012-Lev-3	TC-0.03-Lev-3	BH
Total Pos	319,886	321,041	321,041	321,041	
Longs Pos	199,957	199,975	200,016	199,982	
Short Pos	119,929	119,911	119,870	119,904	
Longs won (%)	78.1	78.09	78.08	78.1	
Shorts won (%)	96.84	96.85	96.85	96.8	
Total direction wins (%)	85.13	85.12	85.12	85.13	
Overall trades won (%)	72.75	56.05	72.75	56.05	
Win/loss ratio	2.67	1.27	2.67	1.27	
Avg. win ($)	0.15	0.15	0.45	0.46	
Avg. loss ($)	−0.071	−0.07	−0.21	−0.22	
Max. win ($)	16.06	16.02	48.19	48.08	
Max. loss ($)	−9.81	−9.84	−29.43	−29.54	
Strat abs. ret. (%)	2892.83	1741.63	8676.52	5224.21	222.94
Strat ann. ret. (%)	192.73	151.08	311.26	251.177	44.84
Strat ann. Sharpe	127.18	75.11	130.38	74.933	0.49
Strat ann. α (%)	106.35	90.47	141.61	125.28	
Strat β	0.0002	0.0003	0.00038	0.00055	

make futures market a very lucrative avenue for trading. Fairly pricing perpetual BTC futures is hard, because of the volatility of the so called funding rate. This work proposed a novel way of pricing BTC futures contracts. The price of the futures contract is modelled by a combination of conditional mean and conditional volatility models. The conditional mean model is a constant autoregressive moving average (ARMA) (0, 0) process. A number of different ML and econometric models are used for predicting the conditional volatility. Turns out that exponential GARCH (EGARCH) is an almost un-biased predictor of volatility in the BTC futures market.

Once the perpetual futures are priced, a market directional HFT technique is developed. The trading strategy is simple; it predicts the direction of the futures contract price using the slope between the current futures contract price and the predicted futures price at the next trading sample period. Trading every 5 min generates the largest profits. Extensive experiments on out-of-sample dataset shows that substantial absolute profits can be generated ranging from 1500% to 8000% depending upon the transaction costs and leverage taken. The EGARCH model along with the conditional mean model are able to predict the future price direction correctly 85% of the time, which forms the basis for the large positive profit and loss (P/L). Finally, the trading strategy is able to produce a positive P/L is both Bear (down-trend) and Bull (up-trend) markets, which shows market neutrality—usually only associated with arbitrage strategies.

In future, the volatility model can be improved by including Poisson jump processes, which are currently under-estimated by the EGARCH conditional volatility process.

DATA AVAILABILITY STATEMENT

Data is available directly from Binance website. A link is here: https://data.binance.vision/?prefix=data/futures/.

ORCID

Avinash Malik https://orcid.org/0000-0002-7524-8292

ENDNOTES

[1] http://www.futures.binance.com.

[2] https://www.coingecko.com/en/exchanges/binance_futures.

[3] https://www.binance.com/en/fee/futureFee.

[4] https://data.binance.vision/?prefix=data/.

[5] https://www.binance.com/en/fee/futureFee.

[6] Continuously compounding interest.

REFERENCES

Aalborg, H. A., Molnár, P., & Vries, D. J. E. (2019). What can explain the price, volatility and trading volume of Bitcoin? *Finance Research Letters*, *29*, 255–265.

Akyildirim, E., Cepni, O., Corbet, S., & Uddin, G. S. (2021). Forecasting mid-price movement of Bitcoin futures using machine learning. *Annals of Operations Research*, 1–32.

Alexander, C., Heck, D. F., & Kaeck, A. (2022). The role of binance in Bitcoin volatility transmission. *Applied Mathematical Finance*, *29*(1), 1–32.

Aras, S. (2021). Stacking hybrid GARCH models for forecasting Bitcoin volatility. *Expert Systems with Applications*, *174*, 114747.

Basher, S. A., & Sadorsky, P. (2022). Forecasting Bitcoin price direction with random forests: How important are interest rates, inflation, and market volatility? *Machine Learning with Applications*, *9*, 100355.

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, *81*(3), 637–654.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327.

Bollerslev, T., Patton, A. J., & Quaedvlieg, R. (2016). Exploiting the errors: A simple approach for improved volatility forecasting. *Journal of Econometrics*, *192*(1), 1–18.

Box, G. (2013). *Box and Jenkins: Time series analysis, forecasting and control* (pp. 161–215). Springer.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Chi, Y., & Hao, W. (2021). Volatility models for cryptocurrencies and applications in the options market. *Journal of International Financial Markets, Institutions and Money*, *75*, 101421.

Chou, R. Y. (2005). Forecasting financial volatilities with extreme values: The conditional autoregressive range (CARR) model. *Journal of Money, Credit and Banking*, *37*, 561–582.

Clements, A., & Preve, D. P. (2021). A practical guide to harnessing the HAR volatility model. *Journal of Banking & Finance*, *133*, 106285.

Corsi, F., Mittnik, S., Pigorsch, C., & Pigorsch, U. (2008). The volatility of realized volatility. *Econometric Reviews*, *27*(1–3), 46–78.

Dickey, D. A. (2015). *Stationarity issues in time series models* (p. 30). SAS Users Group International.

Ding, Z., Granger, C. W., & Engle, R. F. (1993). A long memory property of stock market returns and a new model. *Journal of Empirical Finance*, *1*(1), 83–106.

Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., & Li, L. (2022). Cryptocurrency trading: A comprehensive survey. *Financial Innovation*, *8*(1), 1–59.

Griffin, J. M., & Shams, A. (2020). Is Bitcoin really untethered? *The Journal of Finance*, *75*(4), 1913–1964.

He, S., Manela, A., Ross, O., & Wachter, v V. (2022). Fundamentals of perpetual futures. *arXiv preprint arXiv:2212.06888*.

Hull, J. C. (2003). *Options futures and other derivatives*. Pearson Education India.

Jarque, C. M., & Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, *6*(3), 255–259.
Jarrow, R. A., & Oldfield, G. S. (1981). Forward contracts and futures contracts. *Journal of Financial Economics*, *9*(4), 373–382.
Katsiampa, P. (2017). Volatility estimation for Bitcoin: A comparison of GARCH models. *Economics Letters*, *158*, 3–6.
Kj, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1–2), 307–319.
Lee, G. G., & Engle, R. F. (1993). A permanent and transitory component model of stock return volatility. *SSRN 5848*.
Makarov, I., & Schoar, A. (2020). Trading and arbitrage in cryptocurrency markets. *Journal of Financial Economics*, *135*(2), 293–319.
Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.
Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the Econometric Society*, *59*, 347–370.
Pagan, A. R., & Sossounov, K. A. (2003). A simple framework for analysing bull and bear markets. *Journal of Applied Econometrics*, *18*(1), 23–46.
Patton, A. J. (2009). Are "market neutral" hedge funds really market neutral? *The Review of Financial Studies*, *22*(7), 2495–2530.
Ramsey, J. B., Newton, H. J., & Harvill, J. L. (2002). *The elements of statistics: With applications to economics and the social sciences*. Duxbury/Thomson Learning.
Rokach, L., & Maimon, O. (2005). Decision trees. *Data Mining and Knowledge Discovery Handbook*, *2*(2005),165–192.
Sharpe, W. F. (1998). The Sharpe ratio. *Streetwise–The Best of the Journal of Portfolio Management*, *3*, 169–185.
Shynkevich, A. (2021). Bitcoin arbitrage. *Finance Research Letters*, *40*, 101698.
Taylor, S. J. (2008). *Modelling financial time series*. World Scientific.
Uhlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the Brownian motion. *Physical Review*, *36*(5), 823–841.
Zhang, C., Chen, H., & Peng, Z. (2022). Does Bitcoin futures trading reduce the normal and jump volatility in the spot market? Evidence from GARCH-jump models. *Finance Research Letters*, *47*, 102777.
Zhang, C., Zhang, Y., Cucuringu, M., & Qian, Z. (2022). Volatility forecasting with machine learning and intraday commonality. *arXiv preprint arXiv: 2202.08962*.

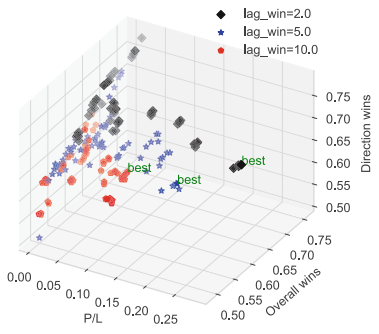
AUTHOR BIOGRAPHY

Avinash Malik is a Senior Lecturer at the University of Auckland, New Zealand. His main research interest lies in applying AI in the biomedical and financial fields. He has worked at organizations such as INRIA in France, Trinity College Dublin, IBM research Ireland, and IBM Watson. He holds B.E. and PhD degrees from the University of Auckland.

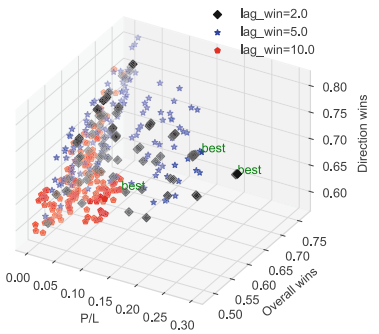
How to cite this article: Malik, A. (2023). A comparison of machine learning and econometric models for pricing perpetual Bitcoin futures and their application to algorithmic trading. *Expert Systems*, *40*(10), e13414. https://doi.org/10.1111/exsy.13414

APPENDIX A

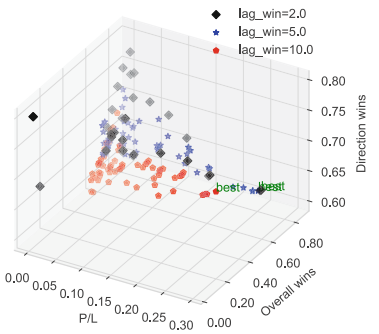
TRADING RESULTS FOR DIFFERENT VOLATILITY MODELS WITH DIFFERENT WINDOW SIZES FOR MEAN MODEL



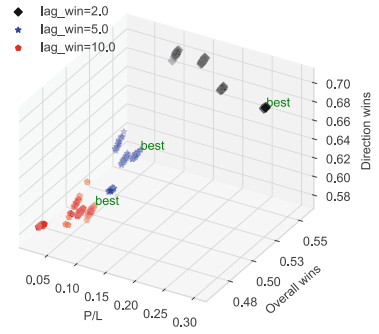
(a) GARCH model results for different window sizes for μ_t and all parameters values for GARCH used in grid search. The best result (best P/L $\times$ best overall wins $\times$ best direction wins) for different window sizes is marked as "best".



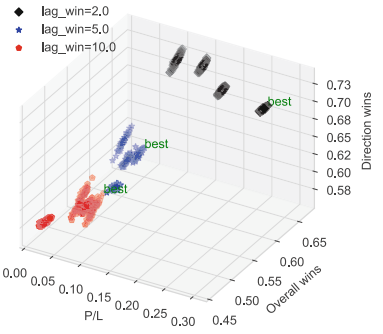
(b) APARCH model results for different window sizes for μ_t and all parameters values for APARCH used in grid search. The best result (best P/L $\times$ best overall wins $\times$ best direction wins) for different window sizes is marked as "best".



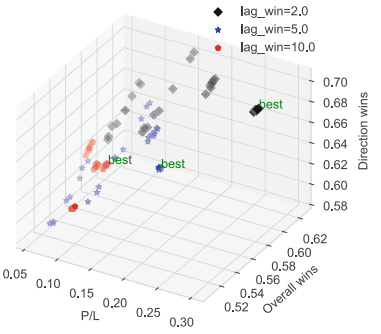
(c) CGARCH model results for different window sizes for μ_t and all parameters values for CGARCH used in grid search. The best result (best P/L $\times$ best overall wins $\times$ best direction wins) for different window sizes is marked as "best".



(d) RF model results for different window sizes for μ_t and all parameters values for RF used in grid search. The best result (best P/L $\times$ best overall wins $\times$ best direction wins) for different window sizes is marked as "best".



(e) SVM model results for different window sizes for μ_t and all parameters values for SVM used in grid search. The best result (best P/L $\times$ best overall wins $\times$ best direction wins) for different window sizes is marked as "best".



(f) HAR-RV model results for different window sizes for μ_t and all parameters values for HAR-RV used in grid search. The best result (best P/L $\times$ best overall wins $\times$ best direction wins) for different window sizes is marked as "best".

FIGURE A1 Comparing different lags for μ_t. Sampled at 5, 15, 30, and 60 min (s). Transaction costs of 0.012% for every order. Trading on out-of-sample dataset from 1 February 2023 to 28 February 2023. Bet $100 every trade. No leverage.