Do the Hype of the Benefits from Using New Data Science Tools Extend to Forecasting Extremely Volatile Assets?

Steven F. Lehrer, Tian Xie, and Guanxi Yi

Abstract This chapter first provides an illustration of the benefits of using machine learning for forecasting relative to traditional econometric strategies. We consider the short-term volatility of the Bitcoin market by realized volatility observations. Our analysis highlights the importance of accounting for nonlinearities to explain the gains of machine learning algorithms and examines the robustness of our findings to the selection of hyperparameters. This provides an illustration of how different machine learning estimators improve the development of forecast models by relaxing the functional form assumptions that are made explicit when writing up an econometric model. Our second contribution is to illustrate how deep learning can be used to measure market-level sentiment from a 10% random sample of Twitter users. This sentiment variable significantly improves forecast accuracy for every econometric estimator and machine algorithm considered in our forecasting application. This provides an illustration of the benefits of new tools from the natural language processing literature at creating variables that can improve the accuracy of forecasting models.

1 Introduction

Over the past few years, the hype surrounding words ranging from big data to data science to machine learning has increased from already high levels. This hype arises

S. F. Lehrer (✉)
Queen's University, Kingston, ON, Canada
NBER, Cambridge, MA, USA
e-mail: lehrers@queensu.ca

T. Xie
Shanghai University of Finance and Economics, Shanghai, China
e-mail: xietian@shufe.edu.cn

G. Yi
Digital Asset Strategies, LLP, Santa Monica, CA, USA
e-mail: Guanxi@das.fund

S. Consoli et al. (eds.), *Data Science for Economics and Finance*,
https://doi.org/10.1007/978-3-030-66891-4_13

in part from three sets of discoveries. Machine learning tools have repeatedly been shown in the academic literature to outperform statistical and econometric techniques for forecasting.[1] Further, tools developed in the natural language processing literature that are used to extract population sentiment measures have also been found to help forecast the value of financial indices. This set of finding is consistent with arguments in the behavioral finance literature (see [23], among others) that the sentiment of investors can influence stock market activity. Last, issues surrounding data security and privacy have grown among the population as a whole, leading governments to consider blockchain technology for uses beyond what it was initially developed for.

Blockchain technology was originally developed for the cryptocurrency Bitcoin, an asset that can be continuously traded and whose value has been quite volatile. This volatility may present further challenges for forecasts by either machine learning algorithms or econometric strategies. Adding to these challenges is that unlike almost every other financial asset, Bitcoin is traded on both the weekend and holidays. As such, modeling the estimated daily realized variance of Bitcoin in US dollars presents an additional challenge. Many measures of conventional economic and financial data commonly used as predictors are not collected at the same points in time. However, since the behavioral finance literature has linked population sentiment measures to the price of different financial assets, we propose measuring and incorporating social media sentiment as an explanatory variable in the forecasting model. As an explanatory predictor, social media sentiment can be measured continuously providing a chance to capture and forecast the variation in the prices at which trades for Bitcoin are made.

In this chapter, we consider forecasts of Bitcoin realized volatility to first provide an illustration of the benefits in terms of forecast accuracy of using machine learning relative to traditional econometric strategies. While prior work contrasting approaches to conduct a forecast found that machine learning does provide gains primarily from relaxing the functional form assumptions that are made explicit when writing up an econometric model, those studies did not consider predicting an outcome that exhibits a degree of volatility of the magnitude of Bitcoin.

Determining strategies that can improve volatility forecasts is of significant value since they have come to play a large role in decisions ranging from asset allocation to derivative pricing and risk management. That is, volatility forecasts are used by traders as a component of their valuation procedure of any risky asset's value (e.g., stock and bond prices), since the procedure requires assessing the level and riskiness of future payoffs. Further, their value to many investors arises when using a strategy that adjust their holdings to equate the risk stemming from the different investments included in a portfolio. As such, more accurate volatility forecasts can provide

[1] See [25, 26], for example, with data from the film industry that conducts horse races between various strategies. Medeiros et al. [31] use the random forest estimator to examine the benefits of machine learning for forecasting inflation. Last, Coulombe et al. [13] conclude that the benefits from machine learning over econometric approaches for macroeconomic forecasting arise since they capture important nonlinearities that arise in the context of uncertainty and financial frictions.

valuable actionable insights for market participants. Finally, additional motivation for determining how to obtain more accurate forecasts comes from the financial media who frequently report on market volatility since it is hypothesized to have an impact on public confidence and thereby can have a significant effect on the broader global economy.

There are many approaches that could be potentially used to undertake volatility forecasts, but each requires an estimate of volatility. At present, the most popular method used in practice to estimate volatility was introduced by Andersen and Bollerslev [1] who proposed using the realized variance, which is calculated as the cumulative sum of squared intraday returns over short time intervals during the trading day.[2] Realized volatility possesses a slowly decaying autocorrelation function, sometimes known as long memory.[3] Various econometric models have been proposed to capture the stylized facts of these high-frequency time series models including the autoregressive fractionally integrated moving average (ARFIMA) models of Andersen et al. [3] and the heterogeneous autoregressive (HAR) model proposed by Corsi [11]. Compared with the ARFIMA model, the HAR model rapidly gained popularity, in part due to its computational simplicity and excellent out-of-sample forecasting performance. [4]

In our empirical exercise, we first use well-established machine learning techniques within the HAR framework to explore the benefits of allowing for general nonlinearities with recursive partitioning methods as well as sparsity using the least absolute shrinkage and selection operator (LASSO) of Tibshirani [39]. We consider alternative ensemble recursive partitioning methods including bagging and random forest that each place equal weight on all observations when making a forecast, as well as boosting that places alternative weight based on the degree of fit. In total, we evaluate nine conventional econometric methods and five easy-to-implement machine learning methods to model and forecast the realized variance of Bitcoin measured in US dollars.

Studies in the financial econometric literature have reported that a number of different variables are potentially relevant for the forecasting of future volatility. A

[2]Traditional econometric approaches to model and forecast such as the parametric GARCH or stochastic volatility models include measures built on daily, weekly, and monthly frequency data. While popular, empirical studies indicate that they fail to capture all information in high-frequency data; see [1, 7, 20], among others.

[3]This phenomenon has been documented by Dacorogna et al. [15] and Andersen et al. [3] for the foreign exchange market and by Andersen et al. [2] for stock market returns.

[4]Corsi et al. [12] provide a comprehensive review of the development of HAR-type models and their various extensions. The HAR model provides an intuitive economic interpretation that agents with three frequencies of trading (daily, weekly, and monthly) perceive and respond to, which changes the corresponding components of volatility. Müller et al. [33] refer to this idea as the Heterogeneous Market Hypothesis. Nevertheless, the suitability of such a specification is not subject to enough verification. Craioveanu and Hillebrand [14] employ a parallel computing method to investigate all of the possible combinations of lags (chosen within a maximum lag of 250) for the last two terms in the additive model, and they compared their in-sample and out-of-sample fitting performance.

secondary goal of our empirical exercise is to determine if there are gains in forecast accuracy of realized volatility by incorporating a measure of social media sentiment. We contrast forecasts using models that both include and exclude social media sentiment. This additional exercise allows us to determine if this measure provides information that is not captured by either the asset-specific realized volatility histories or other explanatory variables that are often included in the information set.

Specifically, in our application social media sentiment is measured by adopting a deep learning algorithm introduced in [17]. We use a random sample of 10% of all tweets posted from users based in the United States from the Twitterverse collected at the minute level. This allows us to calculate a sentiment score that is an equal tweet weight average of the sentiment values of the words within each Tweet in our sample at the minute level.[5] It is well known that there are substantial intraday fluctuations in social media sentiment but its weekly and monthly aggregates are much less volatile. This intraday volatility may capture important information and presents an additional challenge when using this measure for forecasting since the Bitcoin realized variance is measured at the daily level, a much lower time frequency than the minute-level sentiment index that we refer to as the US Sentiment Index (USSI). Rather than make ad hoc assumptions on how to aggregate the USSI to the daily level, we follow Lehrer et al. [28] and adopt the heterogeneous mixed data sampling (H-MIDAS) method that constructs empirical weights to aggregate the high-frequency social media data to a lower frequency.

Our analysis illustrates that sentiment measures extracted from Twitter can significantly improve forecasting efficiency. The gains in forecast accuracy as pseudo R-squared increased by over 50% when social media sentiment was included in the information set for all of the machine learning and econometric strategies considered. Moreover, using four different criteria for forecast accuracy, we find that the machine learning techniques considered tend to outperform the econometric strategies and that these gains arise by incorporating nonlinearities. Among the 16 methods considered in our empirical exercise, both bagging and random forest yield the highest forecast accuracy. Results from the [18] test indicate that the improvements that each of these two algorithms offers are statistically significant at the 5% level, yet the difference between these two algorithms is indistinguishable.

For practitioners, our empirical exercise also contains exercises including examining the sensitivity of our findings to the choices of hyperparameters made when implementing any machine learning algorithm. This provides value since the settings of the hyperparameters with any machine learning algorithm can be thought of in an analogous manner to model selection in econometrics. For example,

[5] We note that the assumption of equal weight is strong. Mai et al. [29] find that social media sentiment is an important predictor in determining Bitcoin's valuation, but not all social media messages are of equal impact. Yet, our measure of social media is collected from all Twitter users, a more diverse group than users of cryptocurrency forums in [29]. Thus, if we find any effect, it is likely a lower bound since our measure of social media sentiment likely has classical measurement error.

with the random forest algorithm, numerous hyperparameters can be adjusted by the researcher including the number of observations drawn randomly for each tree and whether they are drawn with or without replacement, the number of variables drawn randomly for each split, the splitting rule, the minimum number of samples that a node must contain, and the number of trees. Further, Probst and Boulesteix provide evidence that the benefits from changing hyperparameters differ across machine learning algorithms and are higher with the support vector regression than the random forest algorithm we employ. In our analysis, the default values of the hyperparameters specified in software packages work reasonably well, but we stress a caveat that our investigation was not exhaustive so there remains a possibility that there are particular specific combinations of hyperparameters with each algorithm that may lead to changes in the ordering of forecast accuracy in the empirical horse race presented. Thus, there may be a set of hyperparameters where the winning algorithms have a distinguishable different effect from the others that it is being compared to.

This chapter is organized as follows. In the next section, we briefly describe Bitcoin. Sections 3 and 4 provide a more detailed overview of existing HAR strategies as well as conventional machine learning algorithms. Section 5 describes the data we utilize and explains how we measure and incorporate social media data into our empirical exercise. Section 6 presents our main empirical results that compare the forecasting performance of each method introduced in Sects. 3 and 4 in a rolling window exercise. To focus on whether social media sentiment data adds value, we contrast the results of incorporating the USSI variable in each strategy to excluding this variable from the model. For every estimator considered, we find that incorporating the USSI variable as a covariate leads to significant improvements in forecast accuracy. We examine the robustness of our results by considering (1) different experimental settings, (2) different hyperparameters, and (3) incorporating covariates on the value of mainstream assets, in Sect. 7. We find that our main conclusions are robust to both changes in the hyperparameters and various settings, as well as little benefits from incorporating mainstream asset markets when forecasting the realized volatility in the value of Bitcoin. Section 8 concludes by providing additional guidance to practitioners to ensure that they can gain the full value of the hype for machine learning and social media data in their applications.

2 What Is Bitcoin?

Bitcoin, the first and still one of the most popular applications of the blockchain technology by far, was introduced in 2008 by a person or group of people known by the pseudonym, Satoshi Nakamoto. Blockchain technology allows digital information to be distributed but not copied. Basically, a time-stamped series of immutable records of data are managed by a cluster of computers that are not owned by any single entity. Each of these blocks of data (i.e., block) is secured and bound

to each other using cryptographic principles (i.e., chain). The blockchain network has no central authority and all information on the immutable ledger is shared. The information on the blockchain is transparent and each individual involved is accountable for their actions.

The group of participants who uphold the blockchain network ensure that it can neither be hacked or tampered with. Additional units of currency are created by the nodes of a peer-to-peer network using a generation algorithm that ensures decreasing supply that was designed to mimic the rate at which gold was mined. Specifically, when a user/miner discovers a new block, they are currently awarded 12.5 Bitcoins. However, the number of new Bitcoins generated per block is set to decrease geometrically, with a 50% reduction every 210,000 blocks. The amount of time it takes to find a new block can vary based on mining power and the network difficulty.[6] This process is why it can be treated by investors as an asset and ensures that causes of inflation such as printing more currency or imposing capital controls by a central authority cannot take place. The latter monetary policy actions motivated the use of Bitcoin, the first cryptocurrency as a replacement for fiat currencies.

Bitcoin is distinguished from other major asset classes by its basis of value, governance, and applications. Bitcoin can be converted to a fiat currency using a cryptocurrency exchange, such as Coinbase or Kraken, among other online options. These online marketplaces are similar to the platforms that traders use to buy stock. In September 2015, the Commodity Futures Trading Commission (CFTC) in the United States officially designated Bitcoin as a commodity. Furthermore, the Chicago Mercantile Exchange in December 2017 launched a Bitcoin future (XBT) option, using Bitcoin as the underlying asset. Although there are emerging crypto-focused funds and other institutional investors,[7] this market remains retail investor dominated.[8]

[6]Mining is challenging since new blocks and miners are paid any transaction fees as well as a "subsidy" of newly created coins. For the new block to be considered valid, it must contain a proof of work that is verified by other Bitcoin nodes each time they receive a block. By downloading and verifying the blockchain, Bitcoin nodes are able to reach consensus about the ordering of events in Bitcoin. Any currency that is generated by a malicious user that does not follow the rules will be rejected by the network and thus is worthless. To make each new block more challenging to mine, the rate at which a new block can be found is recalculated every 2016 blocks increasing the difficulty.

[7]For example, the legendary former Legg Mason' Chief Investment Officer Bill Miller's fund has been reported to have 50% exposure to crypto-assets. There is also a growing set of decentralized exchanges, including IDEX, 0x, etc., but their market shares remain low today. Furthermore, given the SEC's recent charge against EtherDelta, a well-known Ethereum-based decentralized exchange, the future of decentralized exchanges faces significant uncertainties.

[8]Apart from Bitcoin, there are more than 1600 other alter coin or cryptocurrencies listed over 200 different exchanges. However, Bitcoin still maintains roughly 50% market dominance. At the end of December 2018, the market capitalization of Bitcoin is roughly 65 billion USD with 3800 USD per token. On December 17, 2017, it reached 330 billion USD cap peak with almost 19,000 USD per Bitcoin according to *Coinmarketcap.com*.

There is substantial volatility in BTC/USD, and the sharp price fluctuations in this digital currency greatly exceed that of most other fiat currencies. Much research has explored why Bitcoin is so volatile; our interest is strictly to examine different empirical strategies to forecast this volatility, which greatly exceeds that of other assets including most stocks and bonds.

3 Bitcoin Data and HAR-Type Strategies to Forecast Volatility

The price of Bitcoin is often reported to experience wild fluctuations. We follow Xie [42] who evaluates model averaging estimators with data on the Bitcoin price in US dollars (henceforth BTC/USD) at a 5-min. frequency between May 20, 2015, and Aug 20, 2017. This data was obtained from Poloniex, one of the largest US-based digital asset exchanges. Following Andersen and Bollerslev [1], we estimate the daily realized volatility at day t (RV_t) by summing the corresponding M equally spaced intra-daily squared returns $r_{t,j}$. Here, the subscript t indexes the day, and j indexes the time interval within day t:

$$\mathrm{RV}_t \equiv \sum_{j=1}^{M} r_{t,j}^2 \tag{1}$$

where $t = 1, 2, \ldots, n$, $j = 1, 2, \ldots, M$, and $r_{t,j}$ is the difference between log-prices $p_{t,j}$ ($r_{t,j} = p_{t,j} - p_{t,j-1}$). Poloniex is an active exchange that is always in operation, every minute of each day in the year. We define a trading day using Eastern Standard Time and with data calculate realized volatility of BTC/USD for 775 days. The evolution of the RV data over this full sample period is presented in Fig. 1.

In this section, we introduce some HAR-type strategies that are popular in modeling volatility. The standard HAR model of Corsi [11] postulates that the h-step-ahead daily RV_{t+h} can be modeled by[9]

$$log\mathrm{RV}_{t+h} = \beta_0 + \beta_d log\mathrm{RV}_t^{(1)} + \beta_w log\mathrm{RV}_t^{(5)} + \beta_m log\mathrm{RV}_t^{(22)} + e_{t+h}, \tag{2}$$

[9]Using the log to transform the realized variance is standard in the literature, motivated by avoiding imposing positive constraints and considering the residuals of the below regression to have heteroskedasticity related to the level of the process, as mentioned by Patton and Sheppard [34]. An alternative is to implement weighted least squares (WLS) on RV, which does not suit well our purpose of using the least squares model averaging method.

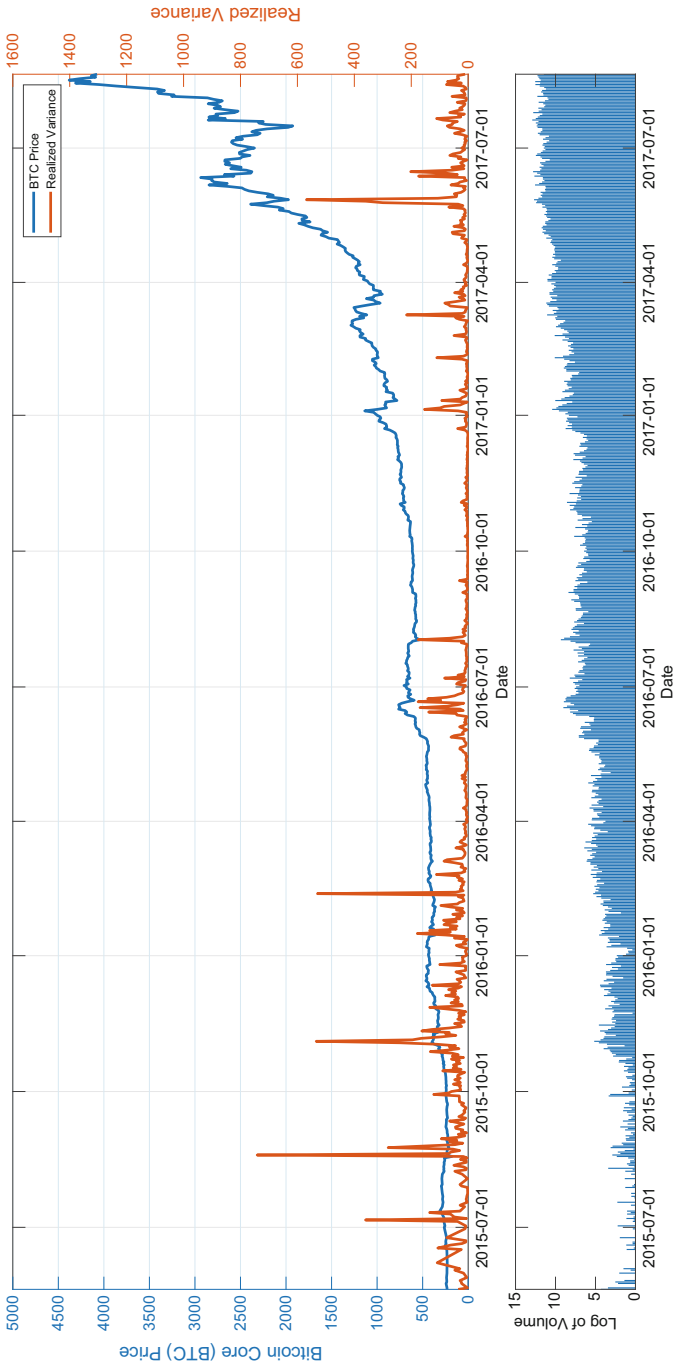

Fig. 1 BTC/USD price, realized variance, and volume on Poloniex

where the βs are the coefficients and $\{e_t\}_t$ is a zero mean innovation process. The explanatory variables take the general form of $log\text{RV}_t^{(l)}$ that is defined as the l period averages of daily log RV:

$$log\text{RV}_t^{(l)} \equiv l^{-1} \sum_{s=1}^{l} log\text{RV}_{t-s}.$$

Another popular formulation of the HAR model in Eq. (2) ignores the logarithmic form and considers

$$\text{RV}_{t+h} = \beta_0 + \beta_d \text{RV}_t^{(1)} + \beta_w \text{RV}_t^{(5)} + \beta_m \text{RV}_t^{(22)} + e_{t+h}, \tag{3}$$

where $\text{RV}_t^{(l)} \equiv l^{-1} \sum_{s=1}^{l} \text{RV}_{t-s}$.

In an important paper, Andersen et al. [4] extend the standard HAR model from two perspectives. First, they added a daily jump component (J_t) to Eq. (3). The extended model is denoted as the HAR-J model:

$$\text{RV}_{t+h} = \beta_0 + \beta_d \text{RV}_t^{(1)} + \beta_w \text{RV}_t^{(5)} + \beta_m \text{RV}_t^{(22)} + \beta^j \text{J}_t + e_{t+h}, \tag{4}$$

where the empirical measurement of the squared jumps is $\text{J}_t = \max(\text{RV}_t - \text{BPV}_t, 0)$ and the standardized realized bipower variation (BPV) is defined as

$$\text{BPV}_t \equiv (2/\pi)^{-1} \sum_{j=2}^{M} |r_{t,j-1}||r_{t,j}|.$$

Second, through a decomposition of RV into the continuous sample path and the jump components based on the Z_t statistic [22], Andersen et al. [4] extend the HAR-J model by explicitly incorporating the two types of volatility components mentioned above. The Z_t statistic respectively identifies the "significant" jumps CJ_t and continuous sample path components CSP_t by

$$\begin{aligned} \text{CSP}_t &\equiv \mathbb{I}(Z_t \leq \Phi_\alpha) \cdot \text{RV}_t + \mathbb{I}(Z_t > \Phi_\alpha) \cdot \text{BPV}_t, \\ \text{CJ}_t &= \mathbb{I}(Z_t > \Phi_\alpha) \cdot (\text{RV}_t - \text{BPV}_t). \end{aligned}$$

where Z_t is the ratio-statistic defined in [22] and Φ_α is the cumulative distribution function(CDF) of a standard Gaussian distribution with α level of significance. The daily, weekly, and monthly average components of CSP_t and CJ_t are then constructed in the same manner as $\text{RV}^{(l)}$. The model specification for the continuous HAR-J, namely, HAR-CJ, is given by

$$\text{RV}_{t+h} = \beta_0 + \beta_d^c \text{CSP}_t^{(1)} + \beta_w^c \text{CSP}_t^{(5)} + \beta_m^c \text{CSP}_t^{(22)} + \beta_d^j \text{CJ}_t^{(1)} + \beta_w^j \text{CJ}_t^{(5)} + \beta_m^j \text{CJ}_t^{(22)} + e_{t+h}. \tag{5}$$

Note that compared with the HAR-J model, the HAR-CJ model explicitly controls for the weekly and monthly components of continuous jumps. Thus, the HAR-J model can be treated as a special and restrictive case of the HAR-CJ model for

$$\beta_d = \beta_d^c + \beta_d^j, \beta^j = \beta_d^j, \beta_w = \beta_w^c + \beta_w^j, \text{ and } \beta_m = \beta_m^c + \beta_m^j.$$

To capture the role of the "leverage effect" in predicting volatility dynamics, Patton and Sheppard [34] develop a series of models using signed realized measures. The first model, denoted as HAR-RS-I, decomposes the daily RV in the standard HAR model (3) into two asymmetric semi-variances RS_t^+ and RS_t^-:

$$\text{RV}_{t+h} = \beta_0 + \beta_d^+ \text{RS}_t^+ + \beta_d^- \text{RS}_t^- + \beta_w \text{RV}_t^{(5)} + \beta_m \text{RV}_t^{(22)} + e_{t+h}, \tag{6}$$

where $\text{RS}_t^- = \sum_{j=1}^M r_{t,j}^2 \cdot \mathbb{I}(r_{t,j} < 0)$ and $\text{RS}_t^+ = \sum_{j=1}^M r_{t,j}^2 \cdot \mathbb{I}(r_{t,j} > 0)$. To verify whether the realized semi-variances add something beyond the classical leverage effect, Patton and Sheppard [34] augment the HAR-RS-I model with a term interacting the lagged RV with an indicator for negative lagged daily returns $\text{RV}_t^{(1)} \cdot \mathbb{I}(r_t < 0)$. The second model in Eq. (7) is denoted as HAR-RS-II:

$$\text{RV}_{t+h} = \beta_0 + \beta_1 \text{RV}_t^{(1)} \cdot \mathbb{I}(r_t < 0) + \beta_d^+ \text{RS}_t^+ + \beta_d^- \text{RS}_t^- + \beta_w \text{RV}_t^{(5)} + \beta_m \text{RV}_t^{(22)} + e_{t+h}, \tag{7}$$

where $\text{RV}_t^{(1)} \cdot \mathbb{I}(r_t < 0)$ is designed to capture the effect of negative daily returns. As in the HAR-CJ model, the third and fourth models in [34], denoted as HAR-SJ-I and HAR-SJ-II, respectively, disentangle the signed jump variations and the BPV from the volatility process:

$$\text{RV}_{t+h} = \beta_0 + \beta_d^j \text{SJ}_t + \beta_d^{bpv} \text{BPV}_t + \beta_w \text{RV}_t^{(5)} + \beta_m \text{RV}_t^{(22)} + e_{t+h}, \tag{8}$$

$$\text{RV}_{t+h} = \beta_0 + \beta_d^{j-} \text{SJ}_t^- + \beta_d^{j+} \text{SJ}_t^+ + \beta_d^{bpv} \text{BPV}_t + \beta_w \text{RV}_t^{(5)} + \beta_m \text{RV}_t^{(22)} + e_{t+h}, \tag{9}$$

where $\text{SJ}_t = \text{RS}_t^+ - \text{RS}_t^-$, $\text{SJ}_t^+ = \text{SJ}_t \cdot \mathbb{I}(\text{SJ}_t > 0)$, and $\text{SJ}_t^- = \text{SJ}_t \cdot \mathbb{I}(\text{SJ}_t < 0)$. The HAR-SJ-II model extends the HAR-SJ-I model by being more flexible to allow the effect of a positive jump variation to differ in unsystematic ways from the effect of a negative jump variation.

The models discussed above can be generalized using the following formulation in practice:

$$y_{t+h} = \boldsymbol{x}_t \boldsymbol{\beta} + e_{t+h}$$

for $t = 1, \ldots, n$, where y_{t+h} stands for RV_{t+h} and variable $\boldsymbol{x}_t$ collects all the explanatory variables such that

$$\boldsymbol{x}_t \equiv \begin{cases} \left[1, \mathrm{RV}_t^{(1)}, \mathrm{RV}_t^{(5)}, \mathrm{RV}_t^{(22)}\right] & \text{for model HAR in (3),} \\ \left[1, \mathrm{RV}_t^{(1)}, \mathrm{RV}_t^{(5)}, \mathrm{RV}_t^{(22)}, \mathrm{J}_t\right] & \text{for model HAR-J in (4),} \\ \left[1, \mathrm{CSP}_t^{(1)}, \mathrm{CSP}_t^{(5)}, \mathrm{CSP}_t^{(22)}, \mathrm{CJ}_t^{(1)}, \mathrm{CJ}_t^{(5)}, \mathrm{CJ}_t^{(22)}\right] & \text{for model HAR-CJ in (5),} \\ \left[1, \mathrm{RS}_t^{-}, \mathrm{RS}_t^{+}, \mathrm{RV}_t^{(5)}, \mathrm{RV}_t^{(22)}\right] & \text{for model HAR-RS-I in (6),} \\ \left[1, \mathrm{RV}_t^{(1)}\mathbb{I}_{r_t<0}, \mathrm{RS}_t^{-}, \mathrm{RS}_t^{+}, \mathrm{RV}_t^{(5)}, \mathrm{RV}_t^{(22)}\right] & \text{for model HAR-RS-II in (7),} \\ \left[1, \mathrm{SJ}_t, \mathrm{BPV}_t, \mathrm{RV}_t^{(5)}, \mathrm{RV}_t^{(22)}\right] & \text{for model HAR-SJ-I in (8),} \\ \left[1, \mathrm{SJ}_t^{-}, \mathrm{SJ}_t^{+}, \mathrm{BPV}_t, \mathrm{RV}_t^{(5)}, \mathrm{RV}_t^{(22)}\right] & \text{for model HAR-SJ-II in (9).} \end{cases}$$

Since y_{t+h} is infeasible in period t, in practice, we usually obtain the estimated coefficient $\hat{\boldsymbol{\beta}}$ from the following model:

$$y_t = \boldsymbol{x}_{t-h}\boldsymbol{\beta} + e_t, \tag{10}$$

in which both the independent and dependent variables are feasible in period $t = 1, \ldots, n$. Once the estimated coefficients $\hat{\boldsymbol{\beta}}$ are obtained, the h-step-ahead forecast can be estimated by

$$\hat{y}_{t+h} = \boldsymbol{x}_t\hat{\boldsymbol{\beta}} \text{ for } t = 1, \ldots, n.$$

4 Machine Learning Strategy to Forecast Volatility

Machine learning tools are increasingly being used in the forecasting literature.[10] In this section, we briefly describe five of the most popular machine learning algorithms that have been shown to outperform econometric strategies when conducting forecast. That said, as Lehrer and Xie [26] stress the "No Free Lunch" theorem of Wolpert and Macready [41] indicates that in practice, multiple algorithms should be considered in any application.[11]

The first strategy we consider was developed to assist in the selection of predictors in the main model. Consider the regression model in Eq. (10), which contains many explanatory variables. To reduce the dimensionality of the set of the explanatory variables, Tibshirani [39] proposed the LASSO estimator of $\hat{\boldsymbol{\beta}}$ that

[10] For example, Gu et al. [19] perform a comparative analysis of machine learning methods for measuring asset risk premia. Ban et al. [6] adopt machine learning methods for portfolio optimization. Beyond academic research, the popularity of algorithm-based quantitative exchange-traded funds (ETF) has increased among investors, in part since as LaFon [24] points out they both offer lower management fees and volatility than traditional stock-picking funds.

[11] This is an impossibility theorem that rules out the possibility that a general-purpose universal optimization strategy exists. As such, researchers should examine the sensitivity of their findings to alternative strategies.

solves

$$\hat{\boldsymbol{\beta}}^{\text{LASSO}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2n} \sum_{t=1}^{n} (y_t - \boldsymbol{x}_{t-h}\boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{L} |\beta_j|, \tag{11}$$

where λ is a tuning parameter that controls the penalty term. Using the estimates of Eq. (11), the h-step-ahead forecast is constructed in an identical manner as OLS:

$$\hat{y}_{t+h}^{\text{LASSO}} = \boldsymbol{x}_t \hat{\boldsymbol{\beta}}^{\text{LASSO}}.$$

The LASSO has been used in many applications and a general finding is that it is more likely to offer benefits relative to the OLS estimator when either (1) the number of regressors exceeds the number of observations, since it involves shrinkage, or (2) the number of parameters is large relative to the sample size, necessitating some form of regularization.

Recursive partitioning methods do not model the relationship between the explanatory variables and the outcome being forecasted with a regression model such as Eq. (10). Breiman et al. [10] propose a strategy known as classification and regression trees (CART), in which classification is used to forecast qualitative outcomes including categorical responses of non-numeric symbols and texts, and regression trees focus on quantitative response variables. Given the extreme volatility in Bitcoin gives rise to a continuous variable, we use regression trees (RT).

Consider a sample of $\{y_t, \boldsymbol{x}_{t-h}\}_{t=1}^{n}$. Intuitively, RT operates in a similar manner to forward stepwise regression. A fast divide and conquer greedy algorithm considers all possible splits in each explanatory variable to recursively partition the data. Formally, at node τ containing n_τ observations with mean outcome $\overline{y}(\tau)$ of the tree can only be split by one selected explanatory variable into two leaves, denoted as τ_L and τ_R. The split is made at the explanatory variable which will lead to the largest reduction of a predetermined loss function between the two regions.[12] This splitting process continues at each new node until the gain to any forecast adds little value relative to a predetermined boundary. Forecasts at each final leaf are the fitted value from a local constant regression model.

Among machine learning strategies, the popularity of RT is high since the results of the analysis are easy to interpret. The algorithm that determines the split allows partitions among the entire covariate set to be described by a single tree. This contrasts with econometric approaches that begin by assuming a linear parametric form to explain the same process and as with the LASSO build a statistical model to make forecasts by selecting which explanatory variables to include. The tree

[12] A best split is determined by a given loss function, for example, the reduction of the sum of squared residuals (SSR). A simple regression will yield a sum of squared residuals, SSR_0. Suppose we can split the original sample into two subsamples such that $n = n_1 + n_2$. The RT method finds the best split of a sample to minimize the SSR from the two subsamples. That is, the SSR values computed from each subsample should follow: $\text{SSR}_1 + \text{SSR}_2 \leq \text{SSR}_0$.

structure considers the full set of explanatory variables and further allows for nonlinear predictor interactions that could be missed by conventional econometric approaches. The tree is simply a top-down, flowchart-like model which represents how the dataset was partitioned into numerous final leaf nodes. The predictions of a RT can be represented by a series of discontinuous flat surfaces forming an overall rough shape, whereas as we describe below visualizations of forecasts from other machine learning methods are not intuitive.

If the data are stationary and ergodic, the RT method often demonstrates gains in forecasting accuracy relative to OLS. Intuitively, we expect the RT method to perform well since it looks to partition the sample into subgroups with heterogeneous features. With time series data, it is likely that these splits will coincide with jumps and structural breaks. However, with primarily cross-sectional data, the statistical learning literature has discovered that individual regression trees are not powerful predictors relative to ensemble methods since they exhibit large variance [21].

Ensemble methods combine estimates from multiple outputs. Bootstrap aggregating decision trees (aka bagging) proposed in [8] and random forest (RF) developed in [9] are randomization-based ensemble methods. In bagging trees (BAG), trees are built on random bootstrap copies of the original data. The BAG algorithm is summarized as below:

(i) Take a random sample with replacement from the data.
(ii) Construct a regression tree.
(iii) Use the regression tree to make forecast, $\hat{f}$.
(iv) Repeat steps (i) to (iii), $b = 1, \ldots, B$ times and obtain $\hat{f}^b$ for each b.
(v) Take a simple average of the B forecasts $\hat{f}_{\text{BAG}} = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b$ and consider the averaged value $\hat{f}_{\text{BAG}}$ as the final forecast.

Forecast accuracy generally increases with the number of bootstrap samples in the training process. However, more bootstrap samples increase computational time. RF can be regarded as a less computationally intensive modification of BAG. Similar to BAG, RF also constructs B new trees with (conventional or moving block) bootstrap samples from the original dataset. With RF, at each node of every tree only a random sample (without replacement) of q predictors out of the total K $(q < K)$ predictors is considered to make a split. This process is repeated and the remaining steps (iii)–(v) of the BAG algorithm are followed. Only if $q = K$, RF is roughly equivalent to BAG. RF forecasts involve B trees like BAG, but these trees are less correlated with each other since fewer variables are considered for a split at each node. The final RF forecast is calculated as the simple average of forecasts from each of these B trees.

The RT method can respond to highly local features in the data and is quite flexible at capturing nonlinear relationships. The final machine learning strategy we consider refines how highly local features of the data are captured. This strategy is known as boosting trees and was introduced in [21, Chapter 10]. Observations responsible for the local variation are given more weight in the fitting process. If the

algorithm continues to fit those observations poorly, we reapply the algorithm with increased weight placed on those observations.

We consider a simple least squares boosting that fits RT ensembles (BOOST). Regression trees partition the space of all joint predictor variable values into disjoint regions R_j, $j = 1, 2, \ldots, J$, as represented by the terminal nodes of the tree. A constant j is assigned to each such region and the predictive rule is $\boldsymbol{X} \in R_j \Rightarrow f(\boldsymbol{X}) = \gamma_j$, where $\boldsymbol{X}$ is the matrix with tth component $\boldsymbol{x}_{t-h}$. Thus, a tree can be formally expressed as $T(\boldsymbol{X}, \Theta) = \sum_{j=1}^{J} \gamma_j \mathbb{I}(\boldsymbol{X} \in R_j)$, with parameters $\Theta = \{R_j, \gamma_j\}_{j=1}^{J}$. The parameters are found by minimizing the risk

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{j=1}^{J} \sum_{\boldsymbol{x}_{t-h} \in R_j} \mathcal{L}(y_t, \gamma_j),$$

where $\mathcal{L}(\cdot)$ is the loss function, for example, the sum of squared residuals (SSR).

The BOOST method is a sum of all trees:

$$f_M(\boldsymbol{X}) = \sum_{m=1}^{M} T(\boldsymbol{X}; \Theta_m)$$

induced in a forward stagewise manner. At each step in the forward stagewise procedure, one must solve

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{n} L\big(y_t, f_{m-1}(\boldsymbol{x}_{t-h}) + T(\boldsymbol{x}_{t-h}; \Theta_m)\big). \tag{12}$$

for the region set and constants $\Theta_m = \{R_{jm}, \gamma_{jm}\}_1^{J_m}$ of the next tree, given the current model $f_{m-1}(\boldsymbol{X})$. For squared-error loss, the solution is quite straightforward. It is simply the regression tree that best predicts the current residuals $y_t - f_{m-1}(\boldsymbol{x}_{t-h})$, and $\hat{\gamma}_{jm}$ is the mean of these residuals in each corresponding region.

A popular alternative to a tree-based procedure to solve regression problems developed in the machine learning literature is the support vector regression (SVR). SVR has been found in numerous applications including Lehrer and Xie [26] to perform well in settings where there a small number of observations (< 500). Support vector regression is an extension of the support vector machine classification method of Vapnik [40]. The key feature of this algorithm is that it solves for a best fitting hyperplane using a learning algorithm that infers the functional relationships in the underlying dataset by following the structural risk minimization induction principle of Vapnik [40]. Since it looks for a functional relationship, it can find nonlinearities that many econometric procedures may miss using a prior chosen mapping that transforms the original data into a higher dimensional space.

Support vector regression was introduced in [16] and the true data that one wishes to forecast was known to be generated as $y_t = f(x_t) + e_t$, where f is unknown to the researcher and e_t is the error term. The SVR framework approximates $f(x_t)$ in terms of a set of basis functions: $\{h_s(\cdot)\}_{s=1}^{S}$:

$$y_t = f(x_t) + e_t = \sum_{s=1}^{S} \beta_s h_s(x_t) + e_t,$$

where $h_s(\cdot)$ is implicit and can be infinite-dimensional. The coefficients $\beta = [\beta_1, \cdots, \beta_S]^\top$ are estimated through the minimization of

$$H(\beta) = \sum_{t=1}^{T} V_\epsilon \left(y_t - f(x_t)\right) + \lambda \sum_{s=1}^{S} \beta_s^2, \tag{13}$$

where the loss function

$$V_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon \\ |r| - \epsilon & \text{otherwise} \end{cases}$$

is called an ϵ-insensitive error measure that ignores errors of size less than ϵ. The parameter ϵ is usually decided beforehand and λ can be estimated by cross-validation.

Suykens and Vandewalle [38] proposed a modification to the classic SVR that eliminates the hyperparameter ϵ and replaces the original ϵ-insensitive loss function with a least squares loss function. This is known as the least squares SVR (LSSVR). The LSSVR considers minimizing

$$H(\boldsymbol{\beta}) = \sum_{t=1}^{T} \left(y_t - f(x_t)\right)^2 + \lambda \sum_{s=1}^{S} \beta_s^2, \tag{14}$$

where a squared loss function replaces $V_e(\cdot)$ for the LSSVR.

Estimating the nonlinear algorithms (13) and (14) requires a kernel-based procedure that can be interpreted as mapping the data from the original input space into a potentially higher-dimensional "feature space," where linear methods may then be used for estimation. The use of kernels enables us to avoid paying the computational penalty implicit in the number of dimensions, since it is possible to evaluate the training data in the feature space through indirect evaluation of the inner products. As such, the kernel function is essential to the performance of SVR and LSSVR since it contains all the information available in the model and training data to perform supervised learning, with the sole exception of having measures of the outcome variable. Formally, we define the kernel function $K(x, x_t) = h(x)h(x_t)^\top$ as the linear dot product of the nonlinear mapping for any input variable x. In our

analysis, we consider the Gaussian kernel (sometimes referred to as "radial basis function" and "Gaussian radial basis function" in the support vector literature):

$$K(\boldsymbol{x}, x_t) = \exp\left(-\frac{\|\boldsymbol{x} - x_t\|^2}{2\sigma_x^2}\right),$$

where the hyperparameters σ_x^2 and γ.

In our main analysis, we use a tenfold cross-validation to pick the tuning parameters for LASSO, SVR, and LSSVR. For tree-type machine learning methods, we set the basic hyperparameters of a regression tree at their default values. These include but not limited to: (1) the split criterion is SSR; (2) the maximum number of split is 10 for BOOST and $n - 1$ for others; (3) the minimum leaf size is 1; (4) the number of predictors for split is $K/3$ for RF and K for others; and (5) the number of learning cycles is $B = 100$ for ensemble learning methods. We examine the robustness to different values for the hyperparameters in Sect. 7.3.

5 Social Media Data

Substantial progress has been made in the machine learning literature on quickly converting text to data, generating real-time information on social media content. To measure social media sentiment, we selected an algorithm introduced in [17] that pre-trained a five-hidden-layer neural model on 124.6 million tweets containing emojis in order to learn better representations of the emotional context embedded in the tweet. This algorithm was developed to provide a means to learn representations of emotional content in texts and is available with pre-processing code, examples of usage, and benchmark datasets, among other features at github.com/bfelbo/deepmoji. The pre-training data is split into a training, validation, and test set, where the validation and test set are randomly sampled in such a way that each emoji is equally represented. This data includes all English Twitter messages without URLs within the period considered that contained an emoji. The fifth layer of the algorithm focuses on attention and takes inputs from the prior levels which uses a multi-class learners to decode the text and emojis itself. See [17] for further details. Thus, an emoji is viewed as a labeling system for emotional content.

The construction of the algorithm began by acquiring a dataset of 55 billion tweets, of which all tweets with emojis were used to train a deep learning model. That is, the text in the tweet was used to predict which emoji was included with what tweet. The premise of this algorithm is that if it could understand which emoji was included with a given sentence in the tweet, then it has a good understanding of the emotional content of that sentence. The goal of the algorithm is to understand the emotions underlying from the words that an individual tweets. The key feature of this algorithm compared to one that simply scores words themselves is that it is better able to detect irony and sarcasm. As such, the algorithm does not score

individual emotion words in a Twitter message, but rather calculates a score based on the probability of each of 64 different emojis capturing the sentiment in the full Twitter message taking the structure of the sentence into consideration. Thus, each emoji has a fixed score and the sentiment of a message is a weighted average of the type of mood being conveyed, since messages containing multiple words are translated to a set of emojis to capture the emotion of the words within.

In brief, for a random sample of 10% of all tweets every minute, the score is calculated as an equal tweet weight average of the sentiment values of the words within them.[13] That is, we apply the pre-trained classifier of Felbo et al. [17] to score each of these tweets and note that there are computational challenges related to data storage when using very large datasets to undertake sentiment analysis. In our application, the number of tweets per hour generally varies between 120,000 and 200,000 tweets per hour in our 10% random sample. We denote the minute-level sentiment index as the U.S. Sentiment Index (USSI).

In other words, if there are 10,000 tweets each hour, we first convert each tweet to a set of emojis. Then we convert the emojis to numerical values based on a fixed mapping related to their emotional content. For each of the 10,000 tweets posted in that hour, we next calculate the average of these scores as the emotion content or sentiment of that individual tweet. We then calculate the equal weighted average of these tweet-specific scores to gain an hourly measure. Thus, each tweet is treated equally irrespective of whether one tweet contains more emojis than the other. This is then repeated for each hour of each day in our sample providing us with a large time series.

Similar to many other text mining tasks, this sentiment analysis was initially designed to deal with English text. It would be simple to apply an off-the-shelf machine translation tool in the spirit of Google translate to generate pseudo-parallel corpora and then learn bilingual representations for downstream sentiment classification task of tweets that were initially posted in different languages. That said, due to the ubiquitous usage of emojis across languages and their functionality of expressing sentiment, alternative emoji powered algorithms have been developed with other languages. These have smaller training datasets since most tweets are in English and it is an open question as to whether they perform better than applying the [17] algorithm to pseudo-tweets.

Note that the way we construct USSI does not necessarily focus on sentiment related to cyptocurrency only as in [29]. Sentiment, in- and off-market, has been a major factor affecting the price of financial asset [23]. Empirical works have documented that large national sentiment swing can cause large fluctuation in asset prices, for example, [5, 37]. It is therefore natural to assume that national sentiment can affect financial market volatility.

[13]This is a 10% random sample of all tweets since the USSI was designed to measure the real-time mood of the nation and the algorithm does not restrict the calculations to Twitter accounts that either mention any specific stock or are classified as being a market participant.

Data timing presents a serious challenge in using minutely measures of the USSI to forecast the daily Bitcoin RV. Since USSI is constructed at minute level, we convert the minute-level USSI to match the daily sampling frequency of Bitcoin RV using the heterogeneous mixed data sampling (H-MIDAS) method of Lehrer et al. [28].[14] This allows us to transform 1,172,747 minute-level observations for USSI variable via a step function to allow for heterogeneous effects of different high-frequency observations into 775 daily observations for the USSI at different forecast horizons. This step function produces a different weight on the hourly levels in the time series and can capture the relative importance of user's emotional content across the day since the type of users varies in a manner that may be related to BTC volatility. The estimated weights used in the H-MIDAS transformation for our application are presented in Fig. 2.

Last, Table 1 presents the summary statistics for the RV data and p-values from both the Jarque–Bera test for normality and the Augmented Dickey–Fuller (ADF) tests for unit root. We consider the first half sample, the second half sample, and full sample. Each of the series exhibits tremendous variability and a large range across the sample period. Further, none of the series are normally distributed or nonstationary at 5% level.

6 Empirical Exercise

To examine the relative prediction efficiency of different HAR estimators, we conduct an h-step-ahead rolling window exercise of forecasting the BTC/USD RV for different forecasting horizons.[15] Table 2 lists each estimator analyzed in the exercise. For all the HAR-type estimators in Panel A (except the HAR-Full model which uses all the lagged covariates from 1 to 30), we set $l = [1, 7, 30]$. For the machine learning methods in Panel B, the input data includes all covariates as the one for HAR-Full model. Throughout the experiment, the window length is fixed at $WL = 400$ observations. Our conclusions are robust to other window lengths as discussed in Sect. 7.1.

To examine if the sentiment data extracted from social media improves forecasts, we contrasted the forecast from models that exclude the USSI to models that include the USSI as a predictor. We denote methods incorporating the USSI variable with

[14] We provide full details on this strategy in the appendix. In practice, we need to select the lag index $\boldsymbol{l} = [l_1, \ldots, l_p]$ and determine the weight set $\mathcal{W}$ before the estimation. In this study, we set $\mathcal{W} \equiv \{\boldsymbol{w} \in \mathbb{R}^p : \sum_{j=1}^{p} w_j = 1\}$ and use OLS to estimate $\widehat{\beta \boldsymbol{w}}$. We consider $h = 1, 2, 4$, and 7 as in the main exercise. For the lag index, we consider $\boldsymbol{l} = [1 : 5 : 1440]$, given there are 1440 minutes per day.

[15] Additional results using both the GARCH(1, 1) and the ARFIMA(p, d, q) models are available upon request. These estimators performed poorly relative to the HAR model and as such are not included for space considerations.

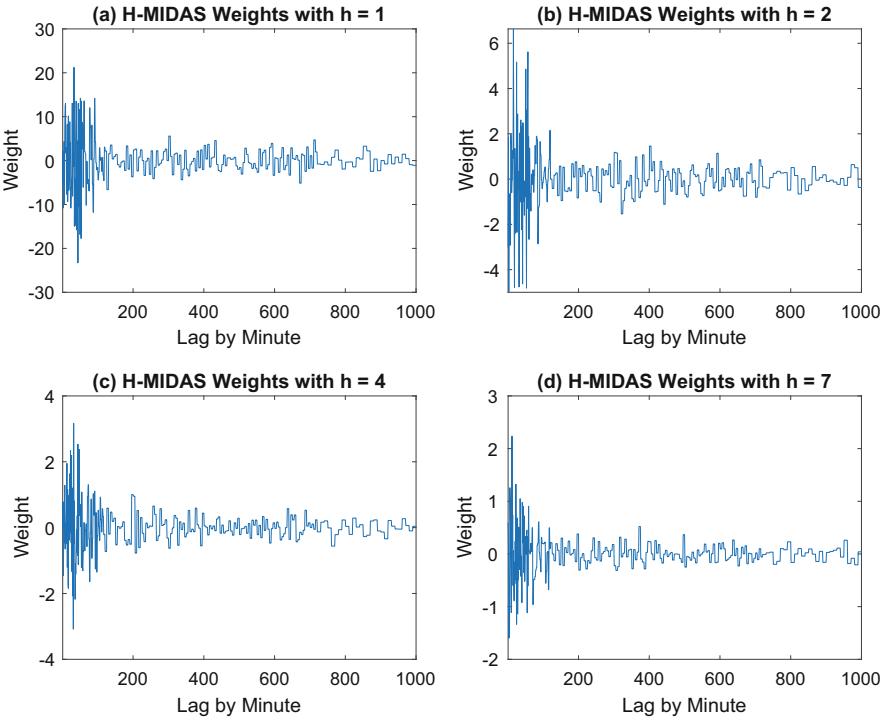

Fig. 2 Weights on the high-frequency observations under different lag indices. (**a**) H-MIDAS weights with h = 1. (**b**) H-MIDAS weights with h = 2. (**c**) H-MIDAS weights with h = 4. (**d**) H-MIDAS weights with h = 7

Table 1 Descriptive statistics

Statistics	Realized variance			USSI
	First half	Second half	Full sample	
Mean	43.4667	12.1959	27.8313	117.4024
Median	31.2213	7.0108	17.4019	125.8772
Maximum	197.6081	115.6538	197.6081	657.4327
Minimum	5.0327	0.5241	0.5241	−866.6793
Std. dev.	38.0177	15.6177	32.9815	179.1662
Skewness	2.1470	3.3633	2.6013	−0.8223
Kurtosis	7.8369	18.2259	11.2147	5.8747
Jarque–Bera	0.0000	0.0000	0.0000	0.0000
ADF test	0.0000	0.0000	0.0000	0.0000

$*$ symbol in each table. The results of the prediction experiment are presented in Table 3. The estimation strategy is listed in the first column and the remaining columns present alternative criteria to evaluate the forecasting performance. The criteria include the mean squared forecast error (MSFE), quasi-likelihood (QLIKE),

Table 2 List of estimators

Panel A: conventional regression		
(1)	AR(1)	A simple autoregressive model
(2)	HAR-Full	The HAR model proposed in [11] with $l = [1, 2, \ldots, 30]$, which is equivalent to AR(30)
(3)	HAR	The conventional HAR model proposed in [11] with $l = [1, 7, 30]$
(4)	HAR-J	The HAR model with jump component proposed in [4]
(5)	HAR-CJ	The HAR model with continuous jump component proposed in [4]
(6)	HAR-RS-I	The HAR model with semi-variance components (Type I) proposed in [34]
(7)	HAR-RS-II	The HAR model with semi-variance components (Type II) proposed in [34]
(8)	HAR-SJ-I	The HAR model with semi-variance and jump components (Type I) proposed in [34]
(9)	HAR-SJ-II	The HAR model with semi-variance and jump components (Type II) proposed in [34]
Panel B: machine learning strategy		
(10)	LASSO	The least absolute shrinkage and selection operator by Tibshirani [39]
(11)	RT	The regression tree method proposed by Breiman et al. [10]
(12)	BOOST	The boosting tree method described in [21]
(13)	BAG	The bagging tree method proposed by Breiman [8]
(14)	RF	The random forest method proposed by Breiman [9]
(15)	SVR	The support vector machine for regression by Drucker et al. [16]
(16)	LSSVR	The least squares support vector regression by Suykens and Vandewalle [38]

mean absolute forecast error (MAFE), and standard deviation of forecast error (SDFE) that are calculated as

$$\text{MSFE}(h) = \frac{1}{V}\sum_{j=1}^{V} e_{T_j,h}^2, \tag{15}$$

$$\text{QLIKE}(h) = \frac{1}{V}\sum_{j=1}^{V}\left(\log \hat{y}_{T_j,h} + \frac{y_{T_j,h}}{\hat{y}_{T_j,h}}\right), \tag{16}$$

$$\text{MAFE}(h) = \frac{1}{V}\sum_{j=1}^{V} |e_{T_j,h}|, \tag{17}$$

$$\text{SDFE}(h) = \sqrt{\frac{1}{V-1}\left(e_{T_j,h} - \frac{1}{V}\sum_{j=1}^{V} e_{T_j,h}\right)^2}, \tag{18}$$

Table 3 Forecasting performance of strategies in the main exercise

Method	MSFE	QLIKE	MAFE	SDFE	Pseudo R^2
Panel A: $h = 1$					
HAR	1666.8492	0.5356	17.0279	40.8271	0.3173
HAR-CJ	1690.4306	0.5299	17.1844	41.1148	0.3076
HAR-RS-II	2377.5159	0.5471	17.6936	48.7598	0.0262
LASSO	1726.2453	0.5649	17.4025	41.5481	0.2929
BOOST	3003.8597	2.3176	27.7473	54.8075	−0.2304
RF	1680.2756	0.4374	16.7922	40.9912	0.3118
BAG	1628.2674	0.4504	16.8285	40.3518	0.3331
SVR	2218.8594	1.3751	20.0765	47.1048	0.0912
LSSVR	1628.6800	0.4858	16.0397	40.3569	0.3329
HAR*	1459.7257	1.5488	19.2790	38.2064	0.4021
HAR-CJ*	1477.1162	1.7526	19.3398	38.4333	0.3950
HAR-RS-II*	2047.5427	1.5013	19.9458	45.2498	0.1613
LASSO*	1497.0621	1.8256	19.1215	38.6919	0.3868
BOOST*	1312.6693	2.4524	18.6123	36.2308	0.4623
RF*	1178.6862	0.3794	14.4059	34.3320	0.5172
BAG*	**1035.7081**	**0.3635**	**13.8235**	**32.1824**	**0.5758**
SVR*	2226.7603	1.4075	20.2407	47.1886	0.0879
LSSVR*	1494.0104	1.2801	16.4454	38.6524	0.3881
Panel B: $h = 2$					
HAR	2066.1864	0.6681	18.6000	45.4553	0.1558
HAR-CJ	2110.0401	0.6696	19.0773	45.9352	0.1379
HAR-RS-II	2028.5347	0.6838	18.8080	45.0393	0.1712
LASSO	2081.8131	0.6936	18.9990	45.6269	0.1494
BOOST	3615.6614	3.1268	28.7990	60.1304	−0.4772
RF	1880.7996	0.5376	17.1419	43.3682	0.2316
BAG	1994.2700	0.5733	17.8611	44.6572	0.1852
SVR	2224.9431	1.3804	20.1089	47.1693	0.0910
LSSVR	1872.4412	0.6192	16.5504	43.2717	0.2350
HAR*	1803.3278	1.5095	21.2684	42.4656	0.2632
HAR-CJ*	1832.2437	1.9863	21.4102	42.8047	0.2514
HAR-RS-II*	1783.0826	2.3170	21.4938	42.2266	0.2715
LASSO*	1817.9238	1.8877	20.8886	42.6371	0.2573
BOOST*	1832.3453	2.8026	21.2695	42.8059	0.2514
RF*	1511.0049	**0.4593**	15.5323	38.8716	0.3827
BAG*	**1428.6900**	0.4654	**15.1394**	**37.7980**	**0.4163**
SVR*	2232.1703	1.4105	20.2573	47.2458	0.0880
LSSVR*	1702.2016	1.0489	17.0578	41.2577	0.3045

(continued)

Table 3 (continued)

Method	MSFE	QLIKE	MAFE	SDFE	Pseudo R^2
Panel C: $h = 4$					
HAR	2064.3686	0.8043	19.5208	45.4353	0.1610
HAR-CJ	2100.3712	0.8181	20.0445	45.8298	0.1464
HAR-RS-II	2057.6179	0.8077	19.6796	45.3610	0.1638
LASSO	2068.0111	0.8231	19.8920	45.4754	0.1595
BOOST	2348.6453	4.6780	24.2304	48.4628	0.0455
RF	1936.6858	0.5980	17.5443	44.0078	0.2129
BAG	2035.9166	0.6470	17.9963	45.1211	0.1726
SVR	2235.8229	1.3882	20.1259	47.2845	0.0913
LSSVR	1963.1437	0.9329	17.3076	44.3074	0.2022
HAR*	1630.8296	2.5250	21.8847	40.3835	0.3372
HAR-CJ*	1641.7051	2.0302	22.0168	40.5180	0.3328
HAR-RS-II*	1638.4781	2.1343	21.9431	40.4781	0.3341
LASSO*	1636.6835	2.3301	21.5890	40.4559	0.3348
BOOST*	1447.7824	3.3492	20.7355	38.0497	0.4116
RF*	1205.4310	**0.4396**	**14.4692**	34.7193	0.5101
BAG*	**1075.4364**	0.4579	14.8433	**32.7938**	**0.5629**
SVR*	2241.9418	1.4129	20.2578	47.3491	0.0889
LSSVR*	1526.7558	1.3300	17.1047	39.0737	0.3795
Panel D: $h = 7$					
HAR	2108.7457	0.8738	19.9327	45.9211	0.1497
HAR-CJ	2119.8357	0.8872	20.2362	46.0417	0.1452
HAR-RS-II	2142.9983	0.9661	20.2572	46.2925	0.1359
LASSO	2100.7324	0.8939	20.2446	45.8337	0.1529
BOOST	2616.8282	2.9902	24.2636	51.1549	-0.0552
RF	1769.0548	0.5524	15.7001	42.0601	0.2867
BAG	1822.8425	0.5648	16.3405	42.6948	0.2650
SVR	2253.5470	1.4045	20.1991	47.4715	0.0913
LSSVR	2000.7088	0.8148	17.7411	44.7293	0.1933
HAR*	1703.6884	1.6255	22.3689	41.2758	0.3130
HAR-CJ*	1705.7788	1.7958	22.2928	41.3011	0.3122
HAR-RS-II*	1716.5970	1.5604	22.4318	41.4318	0.3078
LASSO*	1710.4945	4.1087	22.1347	41.3581	0.3103
BOOST*	1589.2483	2.8654	19.7297	39.8654	0.3592
RF*	1273.7997	**0.4656**	**14.4000**	35.6903	0.4864
BAG*	**1257.6470**	0.5070	15.1803	**35.4633**	**0.4929**
SVR*	2257.5369	1.4195	20.2793	47.5135	0.0897
LSSVR*	1561.7929	1.0831	18.0236	39.5195	0.3702

The best result under each criterion is highlighted in boldface

where $e_{T_j,h} = y_{T_j,h} - \hat{y}_{T_j,h}$ is the forecast error and $\hat{y}_{iT_j,h}$ is the h-day ahead forecast with information up to T_j that stands for the last observation in each of the V rolling windows. We also report the Pseudo R^2 of the Mincer–Zarnowitz regression [32] given by:

$$y_{T_j,h} = a + b\hat{y}_{T_j,h} + u_{T_j}, \text{ for } j = 1, 2, \ldots, V, \tag{19}$$

Each panel in Table 3 presents the result corresponding to a specific forecasting horizon. We consider various forecasting horizons $h = 1, 2, 4$, and 7.

To ease interpretation, we focus on the following representative methods: HAR, HAR-CJ, HAR-RS-II, LASSO, RF, BAG, and LSSVR with and without the USSI variable. Comparison results between all methods listed in Table 2 are available upon request. We find consistent ranking of modeling methods across all forecast horizons. The tree-based machine learning methods (BAG and RF) have superior performance than all others for each panel. Moreover, methods with USSI (indicated by $*$) always dominate those without USSI, which indicates the importance of incorporating social media sentiment data. We also discover that the conventional econometric methods have unstable performance, for example, the HAR-RS-II model without USSI has the worst performance when $h = 1$, but its performance improves when $h = 2$. The mixed performance of the linear models implies that this restrictive formulation may not be robust to model the highly volatile BTC/USD RV process.

To examine if the improvement from the BAG and RF methods is statistically significant, we perform the modified Giacomini–White test [18] of the null hypothesis that the *column method* performs equally well as the *row method* in terms of MAFE. The corresponding p values are presented in Table 4 for $h = 1, 2, 4, 7$. We see that the gains in forecast accuracy from BAG* and RF* relative to all other strategies are statistically significant, although results between BAG* and RF* are statistically indistinguishable.

7 Robustness Check

In this section, we perform four robustness checks of our main results. We first vary the window length for the rolling window exercise in Sect. 7.1. We next consider different sample periods in Sect. 7.2. We explore the use of different hyperparameters for the machine learning methods in Sect. 7.3. Our final robustness check examines if BTC/USD RV is correlated with other types of financial markets by including mainstream assets RV as additional covariates. Each of these robustness checks that are ported in the main text considers $h = 1$.[16]

[16]Although not reported due to space considerations, we investigated other forecasting horizons and our main findings are robust.

Table 4 Giacomini–White test results

	HAR	HAR-CJ	RS-II	LASSO	BOOST	RF	BAG	SVR	LSSVR	HAR*	HAR-CJ*	RS-II*	LASSO*	BOOST*	RF*	BAG*	SVR*
Panel A: h = 1																	
HAR	–	–	–	–	–	–	–	–	–	–	–	–	–	-	–	–	–
HAR-CJ	0.1609	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II	0.3155	0.4160	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO	**0.0169**	0.2295	0.6622	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–
RF	0.5722	0.4883	0.4241	0.3457	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
BAG	0.9714	0.9095	0.6806	0.7371	**0.0000**	0.0811	–	–	–	–	–	–	–	–	–	–	–
SVR	**0.0091**	**0.0178**	0.1527	**0.0211**	**0.0006**	**0.0002**	**0.0038**	–	–	–	–	–	–	–	–	–	–
LSSVR	0.3449	0.2975	0.3122	0.1931	**0.0000**	0.3631	0.0946	**0.0000**	–	–	–	–	–	–	–	–	–
HAR*	**0.0013**	**0.0033**	0.1384	**0.0091**	**0.0000**	**0.0134**	0.0549	0.5354	**0.0047**	–	–	–	–	–	–	–	–
HAR-CJ*	**0.0008**	**0.0020**	0.1133	**0.0065**	**0.0000**	**0.0135**	0.0526	0.5773	**0.0050**	0.4450	–	–	–	–	–	–	–
HAR-RS-II*	**0.0005**	**0.0006**	**0.0014**	**0.0025**	**0.0002**	**0.0265**	0.0643	0.9384	**0.0172**	0.3066	0.3243	–	–	–	–	–	–
LASSO*	**0.0017**	**0.0043**	0.1688	**0.0100**	**0.0000**	**0.0162**	0.0662	0.4478	**0.0059**	0.2134	0.1287	0.2044	–	–	–	–	–
BOOST*	0.2661	0.3247	0.6207	0.4001	**0.0000**	0.1030	0.2551	0.3348	0.0500	0.6188	0.5917	0.4380	0.7039	–	–	–	–
RF*	**0.0147**	**0.0129**	**0.0402**	**0.0061**	**0.0000**	**0.0009**	**0.0001**	**0.0000**	**0.0077**	**0.0000**	**0.0000**	**0.0003**	**0.0000**	**0.0001**	–	–	–
BAG*	**0.0104**	**0.0095**	**0.0322**	**0.0048**	**0.0000**	**0.0029**	**0.0004**	**0.0000**	**0.0085**	**0.0000**	**0.0000**	**0.0002**	**0.0000**	**0.0000**	0.4810	–	–
SVR*	**0.0061**	**0.0125**	0.1269	**0.0146**	**0.0008**	**0.0001**	**0.0024**	**0.0000**	**0.0000**	0.4554	0.4960	0.8617	0.3744	0.2843	**0.0000**	**0.0000**	–
LSSVR*	0.5912	0.5127	0.4497	0.3794	**0.0000**	0.9770	0.3969	**0.0001**	0.3323	**0.0031**	**0.0036**	**0.0202**	**0.0044**	0.0772	**0.0002**	**0.0005**	**0.0000**
Panel B: h = 2																	
HAR	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-CJ	**0.0012**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II	0.4668	0.4161	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO	**0.0038**	0.7427	0.5100	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–

RF	0.1446	0.0648	0.0850	0.0534	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
BAG	0.5707	0.3177	0.4267	0.3205	**0.0000**	**0.0287**	–	–	–	–	–	–	–	–	–	–	–
SVR	0.1475	0.3618	0.1610	0.2555	**0.0000**	**0.0054**	0.0715	–	–	–	–	–	–	–	–	–	–
LSSVR	0.0596	**0.0324**	**0.0291**	**0.0172**	**0.0000**	0.3543	0.0981	**0.0000**	–	–	–	–	–	–	–	–	–
HAR*	**0.0007**	**0.0091**	**0.0036**	**0.0041**	**0.0002**	**0.0002**	**0.0052**	0.3441	**0.0001**	–	–	–	–	–	–	–	–
HAR-CJ*	**0.0002**	**0.0034**	**0.0016**	**0.0017**	**0.0002**	**0.0002**	**0.0040**	0.3019	**0.0001**	0.2366	–	–	–	–	–	–	–
HAR-RS-II*	**0.0004**	**0.0054**	**0.0008**	**0.0020**	**0.0002**	**0.0000**	**0.0020**	0.2221	**0.0000**	0.3331	0.7645	–	–	–	–	–	–
LASSO*	**0.0029**	**0.0275**	**0.0103**	**0.0131**	**0.0001**	**0.0006**	**0.0110**	0.5042	**0.0001**	**0.0010**	**0.0015**	**0.0068**	–	–	–	–	–
BOOST*	0.0872	0.1611	0.1083	0.1471	**0.0002**	**0.0041**	**0.0297**	0.4905	**0.0022**	0.9994	0.9240	0.8782	0.7968	–	–	–	–
RF*	**0.0146**	**0.0065**	**0.0074**	**0.0043**	**0.0000**	**0.0316**	**0.0048**	**0.0002**	0.3616	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–
BAG*	**0.0127**	**0.0065**	**0.0078**	**0.0048**	**0.0000**	0.0569	**0.0157**	**0.0003**	0.1931	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.2217	–	–
SVR*	0.1131	0.2985	0.1201	0.1992	**0.0000**	**0.0037**	0.0549	**0.0005**	**0.0000**	0.4098	0.3609	0.2762	0.5892	0.5486	**0.0001**	**0.0002**	–
LSSVR*	0.1666	0.0934	0.1012	0.0688	**0.0000**	0.8533	0.3002	**0.0015**	0.2814	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0031**	0.0689	**0.0344**	**0.0009**
Panel C: h = 4																	
HAR	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-CJ	**0.0224**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II	0.6981	0.5251	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO	0.0789	0.6490	0.6093	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST	**0.0033**	**0.0083**	**0.0048**	**0.0081**	–	–	–	–	–	–	–	–	–	–	–	–	–
RF	**0.0106**	**0.0063**	**0.0006**	**0.0030**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
BAG	0.1467	0.0771	0.0527	0.0694	**0.0000**	**0.0098**	–	–	–	–	–	–	–	–	–	–	–
SVR	0.4655	0.9347	0.5465	0.7681	**0.0322**	**0.0034**	0.0717	–	–	–	–	–	–	–	–	–	–
LSSVR	**0.0014**	**0.0012**	**0.0000**	**0.0001**	**0.0000**	0.8528	0.3372	**0.0014**	–	–	–	–	–	–	–	–	–
HAR*	**0.0276**	0.1108	**0.0359**	0.0621	0.1960	**0.0004**	**0.0075**	0.1806	**0.0001**	–	–	–	–	–	–	–	–
HAR-CJ*	**0.0157**	0.0717	**0.0240**	**0.0391**	0.2175	**0.0003**	**0.0057**	0.1552	**0.0001**	0.3451	–	–	–	–	–	–	–
HAR-RS-II*	**0.0355**	0.1292	**0.0314**	0.0696	0.2109	**0.0003**	**0.0055**	0.1622	**0.0001**	0.8194	0.8270	–	–	–	–	–	–

(continued)

Table 4 (continued)

	HAR	HAR-CJ	RS-II	LASSO	BOOST	RF	BAG	SVR	LSSVR	HAR*	HAR-CJ*	RS-II*	LASSO*	BOOST*	RF*	BAG*	SVR*
LASSO*	**0.0466**	0.1687	0.0597	0.0984	0.1420	**0.0008**	**0.0120**	0.2546	**0.0002**	**0.0013**	**0.0044**	0.1774	–	–	–	–	–
BOOST*	0.4973	0.7091	0.5281	0.6364	**0.0477**	**0.0252**	0.0991	0.7525	**0.0392**	0.4558	0.4091	0.4175	0.5780	–	–	–	–
RF*	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0031**	**0.0004**	**0.0000**	**0.0041**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–
BAG*	**0.0002**	**0.0001**	**0.0000**	**0.0000**	**0.0000**	**0.0272**	**0.0058**	**0.0001**	**0.0265**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.6820	–	–
SVR*	0.3783	0.8311	0.4400	0.6475	**0.0393**	**0.0025**	0.0577	**0.0117**	**0.0010**	0.2179	0.1882	0.1972	0.3025	0.8057	**0.0000**	**0.0001**	–
LSSVR*	**0.0113**	**0.0069**	**0.0010**	**0.0026**	**0.0000**	0.9283	0.3331	**0.0057**	0.7714	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0078**	**0.0004**	**0.0052**	**0.0042**
Panel D: $h = 7$																	
HAR	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-CJ	0.1065	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II	0.1331	0.9319	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO	0.2138	0.9811	0.9725	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST	0.0533	0.0770	0.0790	0.0687	–	–	–	–	–	–	–	–	–	–	–	–	–
RF	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
BAG	**0.0001**	**0.0001**	**0.0001**	**0.0000**	**0.0000**	**0.0032**	–	–	–	–	–	–	–	–	–	–	–
SVR	0.7526	0.9693	0.9494	0.9552	0.0607	**0.0000**	**0.0005**	–	–	–	–	–	–	–	–	–	–
LSSVR	**0.0022**	**0.0012**	**0.0035**	**0.0002**	**0.0009**	**0.0001**	**0.0087**	**0.0138**	–	–	–	–	–	–	–	–	–
HAR*	**0.0412**	0.0756	0.0913	0.0672	0.4768	**0.0000**	**0.0000**	0.1501	**0.0004**	–	–	–	–	–	–	–	–
HAR-CJ*	**0.0445**	0.0781	0.0954	0.0748	0.4590	**0.0000**	**0.0000**	0.1712	**0.0005**	0.6254	–	–	–	–	–	–	–
HAR-RS-II*	**0.0302**	0.0568	0.0667	0.0539	0.4943	**0.0000**	**0.0000**	0.1388	**0.0004**	0.7164	0.4060	–	–	–	–	–	–
LASSO*	0.0571	0.1043	0.1253	0.0918	0.4178	**0.0000**	**0.0000**	0.1812	**0.0004**	0.1073	0.4999	0.2253	–	–	–	–	–
BOOST*	0.9169	0.7966	0.7895	0.7876	**0.0118**	**0.0040**	**0.0199**	0.8192	0.2342	0.1673	0.1791	0.1613	0.2063	–	–	–	–
RF*	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.1657	**0.0276**	**0.0002**	**0.0009**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–
BAG*	**0.0008**	**0.0004**	**0.0005**	**0.0003**	**0.0000**	0.8017	0.3219	**0.0023**	**0.0316**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0517	–	–
SVR*	0.6853	0.9647	0.9809	0.9663	0.0673	**0.0000**	**0.0004**	**0.0373**	**0.0121**	0.1670	0.1894	0.1547	0.2013	0.7898	**0.0002**	**0.0021**	–
LSSVR*	**0.0423**	**0.0192**	**0.0301**	**0.0168**	**0.0056**	**0.0027**	**0.0389**	0.0967	0.7300	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.2758	**0.0000**	**0.0013**	0.0871

p-values smaller than 5% are highlighted in boldface

7.1 Different Window Lengths

In the main exercise, we set the window length $WL = 400$. In this section, we also tried other window lengths $WL = 300$ and 500. Table 5 shows the forecasting performance of all the estimators for various window lengths. In all the cases BAG* and RF* yield smallest MSFE, MAFE, and SDFE and the largest Pseudo R^2. We examine the statistical significance of the improvement on forecasting accuracy in Table 6. The small p-values on testing BAG* and RF* against other strategies indicate that the forecasting accuracy improvement is statistically significant at the 5% level.

7.2 Different Sample Periods

In this section, we partition the entire sample period in half: the first subsample period runs from May 20, 2015, to July 29, 2016, and the second subsample period runs from July 30, 2016, to Aug 20, 2017. We carry out the similar out-of-sample analysis with $WL = 200$ for the two subsamples in Table 7 Panels A and B, respectively. We also examine the statistical significance in Table 8. The previous conclusions remain basically unchanged under the subsamples.

7.3 Different Tuning Parameters

In this section, we examine the effect of different tuning parameters for the machine learning methods. We consider a different set of tuning parameters: $B = 20$ for RF and BAG, and $\lambda = 0.5$ for LASSO, SVR, and LSSVR. The machine learning methods with the second set of tuning parameters are labeled as RF2, BAG2, and LASSO2. We replicate the main empirical exercise in Sect. 6 and compare the performance of machine learning methods with different tuning parameters.

The results are presented in Tables 9 and 10. Changes in the considered tuning parameters generally have marginal effects on the forecasting performance, although the results for the second tuning parameters are slightly worse than those under the default setting. Last, social media sentiment data plays a crucial role on improving the out-of-sample performance in each of these exercises.

7.4 Incorporating Mainstream Assets as Extra Covariates

In this section, we examine if the mainstream asset class has spillover effect on BTC/USD RV. We include the RVs of the S&P and NASDAQ indices ETFs (ticker

Table 5 Forecasting performance by different window lengths ($h = 1$)

Method	MSFE	QLIKE	MAFE	SDFE	Pseudo R^2
Panel A: $WL = 300$					
HAR	1626.1783	0.4658	17.3249	40.3259	0.3036
HAR-CJ	1691.6375	0.4806	17.3407	41.1295	0.2756
HAR-RS-II	2427.8630	0.4611	17.8985	49.2733	-0.0397
LASSO	1676.9910	0.4912	17.6299	40.9511	0.2819
BOOST	3902.5683	5.0682	30.7322	62.4705	-0.6712
RF	1725.6296	0.4611	18.2421	41.5407	0.2610
BAG	1633.2346	0.4540	17.5508	40.4133	0.3006
SVR	2017.5537	1.3343	19.3042	44.9172	0.1360
LSSVR	1632.6040	0.4961	17.3568	40.4055	0.3009
HAR*	1473.7240	1.8110	19.4883	38.3891	0.3689
HAR-CJ*	1526.2976	2.4053	19.6475	39.0679	0.3464
HAR-RS-II*	2159.5044	1.6874	20.1350	46.4705	0.0752
LASSO*	1510.2217	2.0658	19.4269	38.8616	0.3533
BOOST*	1531.6126	5.0383	20.4951	39.1358	0.3441
RF*	1277.5211	0.3751	15.7195	35.7424	0.4529
BAG*	**1182.1547**	**0.3602**	**14.7103**	**34.3825**	**0.4938**
SVR*	2022.3680	1.3688	19.4026	44.9707	0.1340
LSSVR*	1492.9071	1.8484	17.1765	38.6382	0.3607
Panel B: $WL = 500$					
HAR	2149.6161	0.5193	20.8155	46.3640	0.3510
HAR-CJ	2219.6210	0.5281	20.1791	47.1129	0.3298
HAR-RS-II	2851.7670	0.5199	21.5077	53.4019	0.1390
LASSO	2205.3996	0.5226	20.7104	46.9617	0.3341
BOOST	3106.4917	4.1749	29.2914	55.7359	0.0621
RF	2144.2577	0.4679	20.7959	46.3061	0.3526
BAG	2256.8494	0.4779	21.5526	47.5063	0.3186
SVR	2870.1779	1.2920	22.2445	53.5740	0.1334
LSSVR	2216.1386	0.4999	19.2678	47.0759	0.3309
HAR*	1686.7126	1.5249	21.6946	41.0696	0.4907
HAR-CJ*	1737.9884	1.5219	21.5992	41.6892	0.4753
HAR-RS-II*	2228.9633	2.0233	22.6721	47.2119	0.3270
LASSO*	1731.5366	1.6110	21.5009	41.6117	0.4772
BOOST*	1595.2616	4.8013	23.3670	39.9407	0.5184
RF*	1380.9952	0.3759	16.9718	37.1617	0.5830
BAG*	**1115.9729**	**0.3669**	**16.1018**	**33.4062**	**0.6631**
SVR*	2879.3386	1.3206	22.3949	53.6595	0.1307
LSSVR*	1890.4027	2.3489	19.2429	43.4788	0.4292

The best result under each criterion is highlighted in boldface

Table 6 Giacomini–White test results by different window lengths ($h = 1$)

	HAR	HAR-CJ	RS-II	LASSO	BOOST	RF	BAG	SVR	LSSVR	HAR*	HAR-CJ*	RS-II*	LASSO*	BOOST*	RF*	BAG*	SVR*
Panel A: $WL = 300$																	
HAR	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-CJ	0.9338	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II	0.4818	0.4462	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO	**0.0307**	0.2119	0.7466	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–
RF	0.3721	0.4172	0.8416	0.5419	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
BAG	0.8110	0.8383	0.8332	0.9316	**0.0000**	**0.0477**	–	–	–	–	–	–	–	–	–	–	–
SVR	0.0736	0.1037	0.4429	0.1229	**0.0000**	0.2329	**0.0387**	–	–	–	–	–	–	–	–	–	–
LSSVR	0.9751	0.9885	0.7617	0.7860	**0.0000**	0.1028	0.7148	**0.0085**	–	–	–	–	–	–	–	–	–
HAR*	**0.0011**	**0.0025**	0.1650	**0.0059**	**0.0000**	0.2636	0.0697	0.8791	0.0594	–	–	–	–	–	–	–	–
HAR-CJ*	**0.0003**	**0.0006**	0.1028	**0.0024**	**0.0000**	0.2279	0.0615	0.7862	0.0548	0.2951	–	–	–	–	–	–	–
HAR-RS-II*	**0.0025**	**0.0015**	**0.0010**	**0.0079**	**0.0000**	0.2672	0.1192	0.6477	0.1185	0.4168	0.4972	–	–	–	–	–	–
LASSO*	**0.0010**	**0.0023**	0.1759	**0.0046**	**0.0000**	0.2727	0.0687	0.9174	0.0599	0.6547	0.2794	0.3800	–	–	–	–	–
BOOST*	**0.0090**	**0.0128**	0.1289	**0.0188**	**0.0000**	**0.0483**	**0.0073**	0.3769	**0.0100**	0.4046	0.4957	0.8260	0.3761	–	–	-	–
RF*	0.0932	0.1171	0.1826	**0.0480**	**0.0000**	**0.0000**	**0.0002**	**0.0000**	**0.0012**	**0.0002**	**0.0003**	**0.0073**	**0.0002**	**0.0000**	–	–	–
BAG*	**0.0230**	**0.0346**	0.0877	**0.0110**	**0.0000**	**0.0000**	**0.0002**	**0.0000**	**0.0009**	**0.0000**	**0.0000**	**0.0020**	**0.0000**	**0.0000**	0.1109	–	–
SVR*	0.0607	0.0877	0.4118	0.1027	**0.0000**	0.1932	**0.0293**	**0.0000**	**0.0058**	0.9436	0.8467	0.6872	0.9837	0.4177	**0.0000**	**0.0000**	–
LSSVR*	0.8887	0.8863	0.6880	0.6641	**0.0000**	0.0908	0.5488	**0.0079**	0.6627	**0.0164**	**0.0173**	0.0767	**0.0160**	**0.0045**	**0.0016**	**0.0008**	**0.0055**
Panel B: $WL = 500$																	
HAR	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-CJ	**0.0007**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II	0.5914	0.3132	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO	0.6862	0.0706	0.5393	–	–	–	–	–	–	–	–	–	–	–	–	–	–

(continued)

Table 6 (continued)

	HAR	HAR-CJ	RS-II	LASSO	BOOST	RF	BAG	SVR	LSSVR	HAR*	HAR-CJ*	RS-II*	LASSO*	BOOST*	RF*	BAG*	SVR*
BOOST	**0.0001**	**0.0000**	**0.0058**	**0.0001**	–	–	–	–	–	–	–	–	–	–	–	–	–
RF	0.9827	0.5018	0.7248	0.9271	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
BAG	0.3567	0.0966	0.9803	0.3041	**0.0001**	0.0950	–	–	–	–	–	–	–	–	–	–	–
SVR	0.1930	**0.0459**	0.7289	0.1745	**0.0037**	0.2397	0.5826	–	–	–	–	–	–	–	–	–	–
LSSVR	0.0867	0.3051	0.2939	0.1249	**0.0000**	**0.0194**	**0.0034**	**0.0018**	–	–	–	–	–	–	–	–	–
HAR*	0.3361	0.1173	0.9117	0.3096	**0.0006**	0.4461	0.8991	0.7049	**0.0397**	–	–	–	–	–	–	–	–
HAR-CJ*	0.3795	0.1278	0.9562	0.3464	**0.0006**	0.4930	0.9666	0.6494	**0.0453**	0.3877	–	–	–	–	–	–	–
HAR-RS-II*	0.1598	0.0691	0.1972	0.1440	**0.0133**	0.3375	0.5270	0.8385	0.0968	0.4019	0.3561	–	–	–	–	–	–
LASSO*	0.4266	0.1472	0.9967	0.3629	**0.0005**	0.5362	0.9615	0.5975	0.0501	0.3700	0.6529	0.3185	–	–	–	–	–
BOOST*	0.1798	0.1015	0.4720	0.1766	**0.0070**	0.1471	0.3213	0.6014	**0.0325**	0.3494	0.3291	0.7675	0.3065	–	–	–	–
RF*	**0.0011**	**0.0095**	**0.0333**	**0.0024**	**0.0000**	**0.0001**	**0.0000**	**0.0008**	**0.0325**	**0.0000**	**0.0000**	**0.0019**	**0.0000**	**0.0002**	–	–	–
BAG*	**0.0002**	**0.0018**	**0.0115**	**0.0005**	**0.0000**	**0.0001**	**0.0000**	**0.0002**	**0.0081**	**0.0000**	**0.0000**	**0.0003**	**0.0000**	**0.0000**	0.1547	–	–
SVR*	0.1520	**0.0330**	0.6768	0.1374	**0.0046**	0.1969	0.5057	**0.0059**	**0.0012**	0.6298	0.5754	0.8949	0.5260	0.6515	**0.0006**	**0.0001**	–
LSSVR*	0.1008	0.3379	0.2743	0.1405	**0.0000**	0.0646	**0.0103**	**0.0103**	0.9700	**0.0008**	**0.0012**	0.0516	**0.0015**	**0.0237**	**0.0060**	**0.0014**	**0.0073**

p-values smaller than 5% are highlighted in boldface

Table 7 Forecasting performance by different sample periods ($h = 1$)

Method	MSFE	QLIKE	MAFE	SDFE	Pseudo R^2
Panel A: first half					
HAR	2124.1237	0.4650	22.9310	46.0882	0.2335
HAR-CJ	2355.2555	0.4492	21.7508	48.5310	0.1500
HAR-RS-II	2603.1374	0.4914	24.0043	51.0210	0.0606
LASSO	2138.6650	0.4666	23.3848	46.2457	0.2282
BOOST	3867.2799	2.0069	32.0598	62.1875	-0.3956
RF	2099.9254	0.3727	19.6797	45.8249	0.2422
BAG	2106.6674	0.4048	19.5280	45.8984	0.2398
SVR	2153.3053	0.5631	22.7778	46.4037	0.2229
LSSVR	2040.2006	0.3860	21.1425	45.1686	0.2637
HAR*	1489.5345	2.9636	26.6309	38.5945	0.4625
HAR-CJ*	1541.1336	7.4995	26.9735	39.2573	0.4438
HAR-RS-II*	1711.2464	1.9009	27.7648	41.3672	0.3825
LASSO*	1448.5859	1.9891	26.4592	38.0603	0.4772
BOOST*	1273.8670	1.4514	22.1323	35.6913	0.5403
RF*	1201.8716	**0.2606**	16.8897	34.6680	0.5663
BAG*	**840.0199**	0.2629	**15.4812**	**28.9831**	**0.6969**
SVR*	2153.5420	0.5633	22.7812	46.4063	0.2228
LSSVR*	1331.7041	2.7236	19.8550	36.4925	0.5194
Panel B: second half					
HAR	3412.6612	0.4790	23.4856	58.4180	0.2370
HAR-CJ	3591.3391	0.4739	24.8167	59.9278	0.1970
HAR-RS-II	5357.5796	0.4995	25.1334	73.1955	−0.1979
LASSO	3575.5839	0.5118	24.0981	59.7962	0.2005
BOOST	6151.3787	4.0402	41.1825	78.4307	−0.3754
RF	3314.1729	0.5416	25.1547	57.5689	0.2590
BAG	3152.0846	0.5716	24.3284	56.1434	0.2952
SVR	3917.5789	1.9247	23.9854	62.5906	0.1241
LSSVR	3187.9434	0.5683	24.3457	56.4619	0.2872
HAR*	2747.1766	1.4813	24.0375	52.4135	0.3858
HAR-CJ*	2908.1546	1.4502	24.5958	53.9273	0.3498
HAR-RS-II*	4324.7752	2.3995	25.4931	65.7630	0.0330
LASSO*	2869.5404	0.7703	24.2617	53.5681	0.3584
BOOST*	2624.4054	5.9681	30.0566	51.2290	0.4132
RF*	2337.9213	**0.3759**	21.4734	48.3521	0.4773
BAG*	**2110.7631**	0.3847	**20.6086**	**45.9430**	**0.5281**
SVR*	3924.9867	1.9806	24.0556	62.6497	0.1224
LSSVR*	2952.6849	0.5104	24.0650	54.3386	0.3398

The best result under each criterion is highlighted in boldface

Table 8 Giacomini–White test results by different sample periods ($h = 1$)

	HAR	HAR-CJ	RS-II	LASSO	BOOST	RF	BAG	SVR	LSSVR	HAR*	HAR-CJ*	RS-II*	LASSO*	BOOST*	RF*	BAG*	SVR*
Panel A: first half																	
HAR	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-CJ	0.1354	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II	0.3850	**0.0068**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO	0.0646	0.0526	0.6283	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST	**0.0005**	**0.0002**	**0.0055**	**0.0008**	–	–	–	–	–	–	–	–	–	–	–	–	–
RF	**0.0004**	0.0688	**0.0035**	**0.0001**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
BAG	**0.0004**	0.0605	**0.0035**	**0.0001**	**0.0000**	0.6653	–	–	–	–	–	–	–	–	–	–	–
SVR	0.8702	0.4822	0.5117	0.5202	**0.0011**	**0.0157**	**0.0117**	–	–	–	–	–	–	–	–	–	–
LSSVR	**0.0001**	0.5021	**0.0349**	**0.0000**	**0.0000**	**0.0478**	0.0510	0.1174	–	–	–	–	–	–	–	–	–
HAR*	0.0680	**0.0375**	0.3060	0.1094	0.0758	**0.0012**	**0.0011**	0.0599	**0.0073**	–	–	–	–	–	–	–	–
HAR-CJ*	**0.0359**	**0.0146**	0.1972	0.0632	0.0923	**0.0004**	**0.0004**	**0.0393**	**0.0032**	0.4765	–	–	–	–	–	–	–
HAR-RS-II*	**0.0171**	**0.0036**	0.0654	**0.0313**	0.1620	**0.0002**	**0.0002**	**0.0275**	**0.0014**	0.2676	0.2651	–	–	–	–	–	–
LASSO*	0.0778	**0.0410**	0.3285	0.1230	0.0656	**0.0014**	**0.0012**	0.0695	**0.0084**	0.5238	0.3100	0.1910	–	–	–	–	–
BOOST*	0.7816	0.9011	0.5709	0.6643	**0.0066**	0.3808	0.3515	0.8233	0.7283	**0.0441**	**0.0338**	**0.0231**	**0.0489**	–	–	–	–
RF*	**0.0000**	**0.0053**	**0.0005**	**0.0000**	**0.0000**	**0.0243**	**0.0322**	**0.0002**	**0.0015**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0164**	–	–	–
BAG*	**0.0014**	**0.0156**	**0.0027**	**0.0007**	**0.0000**	0.0601	0.0693	**0.0019**	**0.0122**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0001**	0.1855	–	–
SVR*	0.8731	0.4808	0.5130	0.5227	**0.0011**	**0.0156**	**0.0116**	0.4224	0.1169	0.0601	**0.0394**	**0.0276**	0.0698	0.8224	**0.0002**	**0.0019**	–
LSSVR*	**0.0169**	0.2273	**0.0259**	**0.0066**	**0.0000**	0.8966	0.8164	**0.0438**	0.2843	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.3193	**0.0034**	**0.0047**	**0.0436**
Panel B: second half																	
HAR	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-CJ	**0.0005**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II	0.4168	0.8623	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO	0.1433	0.1745	0.6130	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–

RF	0.5305	0.9051	0.9962	0.6828	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
BAG	0.7330	0.8538	0.8522	0.9237	**0.0000**	0.2049	–	–	–	–	–	–	–	–	–	–	–
SVR	0.8590	0.7862	0.8069	0.9680	**0.0000**	0.4623	0.8289	–	–	–	–	–	–	–	–	–	–
LSSVR	0.7489	0.8699	0.8634	0.9255	**0.0000**	0.3092	0.9838	0.8023	–	–	–	–	–	–	–	–	–
HAR*	0.6384	0.5469	0.6862	0.9624	**0.0000**	0.6605	0.9023	0.9845	0.9018	–	–	–	–	–	–	–	–
HAR-CJ*	0.3223	0.8517	0.8323	0.6860	**0.0000**	0.8360	0.9156	0.8308	0.9254	0.0696	–	–	–	–	–	–	–
HAR-RS-II*	0.2815	0.6915	0.7691	0.4684	**0.0000**	0.9354	0.7722	0.7294	0.7861	0.4615	0.6166	–	–	–	–	–	–
LASSO*	0.4982	0.6525	0.7429	0.8880	**0.0000**	0.7126	0.9764	0.9170	0.9723	0.5963	0.5101	0.5357	–	–	–	–	–
BOOST*	**0.0119**	0.0521	0.2097	**0.0251**	**0.0029**	0.0575	**0.0232**	**0.0323**	**0.0237**	**0.0097**	**0.0232**	0.1910	**0.0136**	–	–	–	–
RF*	0.5825	0.3362	0.4912	0.4401	**0.0000**	**0.0104**	0.0576	0.2770	**0.0350**	0.3694	0.2905	0.3893	0.3121	**0.0003**	–	–	–
BAG*	0.3290	0.1772	0.3417	0.2432	**0.0000**	**0.0168**	**0.0390**	0.0942	**0.0288**	0.1591	0.1250	0.2463	0.1385	**0.0000**	0.2344	–	–
SVR*	0.8395	0.8039	0.8185	0.9879	**0.0000**	0.4896	0.8635	**0.0000**	0.8404	0.9946	0.8500	0.7416	0.9381	**0.0343**	0.2613	0.0878	–
LSSVR*	0.8282	0.7923	0.8140	0.9900	**0.0000**	0.2548	0.7907	0.9576	0.6384	0.9904	0.8291	0.7258	0.9287	**0.0125**	**0.0198**	**0.0242**	0.9950

p-values smaller than 5% are highlighted in boldface

Table 9 Forecasting performance by different tuning parameters ($h = 1$)

Method	MSFE	QLIKE	MAFE	SDFE	Pseudo R^2
Panel A: without sentiment					
LASSO	1726.2453	0.5649	17.4025	41.5481	0.2929
BOOST	3003.8597	2.3176	27.7473	54.8075	-0.2304
RF	1680.2756	**0.4374**	16.7922	40.9912	0.3118
BAG	**1628.2674**	0.4504	16.8285	**40.3518**	**0.3331**
SVR	2218.8594	1.3751	20.0765	47.1048	0.0912
LSSVR	1628.6800	0.4858	**16.0397**	40.3569	0.3329
LASSO2	1736.6334	0.5546	17.4325	41.6729	0.2887
BOOST2	2965.5740	2.1399	27.2208	54.4571	-0.2147
RF2	1765.2329	0.4706	17.2435	42.0147	0.2770
BAG2	1659.4408	0.4611	16.7576	40.7362	0.3203
SVR2	2218.8594	1.3751	20.0765	47.1048	0.0912
LSSVR2	1635.2935	0.4900	16.0911	40.4388	0.3302
Panel B: with sentiment					
LASSO*	1497.0621	1.8256	19.1215	38.6919	0.3868
BOOST*	1312.6693	2.4524	18.6123	36.2308	0.4623
RF*	1178.6862	0.3794	14.4059	34.3320	0.5172
BAG*	T1035.7081	**0.3635**	**13.8235**	32.1824	0.5758
SVR*	2226.7603	1.4075	20.2407	47.1886	0.0879
LSSVR*	1494.0104	1.2801	16.4454	38.6524	0.3881
LASSO2*	1501.9018	2.1237	19.3177	38.7544	0.3848
BOOST2*	1324.7603	14.1393	18.2779	36.3973	0.4574
RF2*	1250.0685	0.3932	14.8282	35.3563	0.4880
BAG2*	**1007.2093**	0.3842	13.9225	**31.7366**	**0.5874**
SVR2*	2226.7603	1.4075	20.2407	47.1886	0.0879
LSSVR2*	1504.4609	1.7125	16.4577	38.7874	0.3838

The best result under each criterion is highlighted in boldface

names: SPY and QQQ, respectively) and the CBOE Volatility Index (VIX) as extra covariates. For SPY and QQQ, we proxy daily spot variances by daily realized variance estimates. For the VIX, we collect the daily data from CBOE. The extra covariates are described in Table 11

The data range is from May 20, 2015, to August 18, 2017, with 536 total observations. Fewer observations are available since mainstream asset exchanges are closed on the weekends and holidays. We truncate the BTC/USD data accordingly. We compare forecasts from models with two groups of covariate data: one with only the USSI variable and the other which includes both the USSI variable and the mainstream RV data (SPY, QQQ, and VIX). Estimates that include the larger covariate set are denoted by the symbol $**$.

The rolling window forecasting results with $WL = 300$ are presented in Table 12. Comparing results across any strategy between Panels A and B, we do not observe obvious improvements in forecasting accuracy. This implies that

Table 10 Giacomini–White test results by different tuning parameters ($h = 1$)

	LASSO	BOOST	RF	BAG	SVR	LSSVR	LASSO2	BOOST2	RF2	BAG2	SVR	LSSVR2	LASSO*	BOOST*	RF*	BAG*	SVR*	LSSVR*	LASSO2*	BOOST2*	RF2*	SVR2*	LSSVR2*
LASSO	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
RF	0.5830	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BAG	0.6383	**0.0000**	0.7623	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
SVR	**0.0211**	**0.0006**	**0.0013**	**0.0020**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LSSVR	0.1931	**0.0000**	0.1252	0.1172	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO2	0.7415	**0.0000**	0.5644	0.6197	**0.0263**	0.1895	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST2	**0.0000**	**0.0114**	**0.0000**	**0.0000**	**0.0013**	**0.0000**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
RF2	0.5977	**0.0000**	0.9806	0.8041	**0.0013**	0.1549	0.5797	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BAG2	0.9837	**0.0000**	0.1410	0.1615	**0.0097**	**0.0324**	0.9606	**0.0000**	0.2193	–	–	–	–	–	–	–	–	–	–	–	–	–	–
SVR2	**0.0211**	**0.0006**	**0.0013**	**0.0020**	1.0000	**0.0000**	**0.0263**	**0.0013**	**0.0013**	**0.0097**	–	–	–	–	–	–	–	–	–	–	–	–	–
LSSVR2	0.2158	**0.0000**	0.1568	0.1455	**0.0000**	**0.0290**	0.2118	**0.0000**	0.1867	**0.0424**	**0.0000**	–	–	–	–	–	–	–	–	–	–	–	–
LASSO*	**0.0100**	**0.0000**	**0.0435**	**0.0458**	0.4478	**0.0059**	**0.0125**	**0.0001**	**0.0491**	0.1237	0.4478	**0.0072**	–	–	–	–	–	–	–	–	–	–	–
BOOST*	0.4001	**0.0000**	0.1872	0.2141	0.3348	0.0500	0.4163	**0.0000**	0.1809	0.3648	0.3348	0.0557	0.7039	–	–	–	–	–	–	–	–	–	–
RF*	**0.0107**	**0.0000**	**0.0016**	**0.0009**	**0.0000**	**0.0466**	**0.0105**	**0.0000**	**0.0018**	**0.0003**	**0.0000**	**0.0407**	**0.0000**	**0.0001**	–	–	–	–	–	–	–	–	–
BAG*	**0.0042**	**0.0000**	**0.0013**	**0.0005**	**0.0000**	**0.0076**	**0.0045**	**0.0000**	**0.0012**	**0.0002**	**0.0000**	**0.0065**	**0.0000**	**0.0000**	**0.0451**	–	–	–	–	–	–	–	–
SVR*	**0.0146**	**0.0008**	**0.0007**	**0.0012**	**0.0000**	**0.0000**	**0.0186**	**0.0017**	**0.0008**	**0.0063**	**0.0000**	**0.0000**	0.3744	0.2843	**0.0000**	**0.0000**	–	–	–	–	–	–	–
LSSVR*	0.3794	**0.0000**	0.5488	0.4744	**0.0001**	0.3323	0.3702	**0.0000**	0.5667	0.2064	**0.0001**	0.3976	**0.0044**	0.0772	**0.0039**	**0.0004**	**0.0000**	–	–	–	–	–	–
LASSO2*	**0.0047**	**0.0000**	**0.0308**	**0.0321**	0.5551	**0.0041**	**0.0058**	**0.0001**	**0.0355**	0.0910	0.5551	**0.0051**	**0.0000**	0.6014	**0.0000**	**0.0000**	0.4734	**0.0028**	–	–	–	–	–
BOOST2*	0.5432	**0.0000**	0.2858	0.3220	0.2362	0.0894	0.5610	**0.0000**	0.2776	0.5100	0.2362	0.0982	0.5292	**0.0441**	**0.0005**	**0.0000**	0.1969	0.1358	0.4418	–	–	–	–
RF2*	**0.0084**	**0.0000**	**0.0011**	**0.0008**	**0.0000**	**0.0275**	**0.0084**	**0.0000**	**0.0013**	**0.0003**	**0.0000**	**0.0238**	**0.0000**	**0.0001**	0.3103	0.2140	**0.0000**	**0.0013**	**0.0000**	**0.0005**	–	–	–
BAG2*	**0.0065**	**0.0000**	**0.0016**	**0.0007**	**0.0000**	**0.0093**	**0.0067**	**0.0000**	**0.0015**	**0.0003**	**0.0000**	**0.0079**	**0.0000**	**0.0000**	0.1363	0.6847	**0.0000**	**0.0008**	**0.0000**	**0.0000**	0.3800	–	–
SVR2*	**0.0146**	**0.0008**	**0.0007**	**0.0012**	**0.0000**	**0.0000**	**0.0186**	**0.0017**	**0.0008**	**0.0063**	**0.0000**	**0.0000**	0.3744	0.2843	**0.0000**	**0.0000**	1.0000	**0.0000**	0.4734	0.1969	**0.0000**	**0.0000**	–
LSSVR2*	0.3899	**0.0000**	0.5604	0.4858	**0.0000**	0.3061	0.3806	**0.0000**	0.5773	0.2131	**0.0000**	0.3684	**0.0053**	0.0807	**0.0041**	**0.0005**	**0.0000**	0.6459	**0.0034**	0.1408	**0.0014**	**0.0008**	**0.0000**

p-values smaller than 5% are highlighted in boldface

Table 11 Descriptive statistics

Statistics	SPY	QQQ	VIX
Mean	0.3839	0.7043	15.0144
Median	0.2034	0.3515	13.7300
Maximum	12.1637	70.6806	40.7400
Minimum	0.0143	0.0468	9.3600
Std. Dev.	0.6946	3.1108	4.5005
Skewness	10.1587	21.3288	1.6188
Kurtosis	158.5806	479.5436	6.3394
Jarque–Bera	0.0010	0.0010	0.0010
ADF Test	0.0010	0.0010	0.0010

Table 12 Forecasting performance

Method	MSFE	QLIKE	MAFE	SDFE	Pseudo R^2
Panel A: with sentiment					
HAR*	1265.3736	1.7581	21.7060	35.5721	0.4299
HAR-CJ*	1258.1112	1.4488	21.4721	35.4699	0.4332
HAR-RS-II*	1312.9602	1.6025	22.4346	36.2348	0.4085
LASSO*	1251.4556	1.7235	21.3984	35.3759	0.4362
BOOST*	1135.0482	9.2958	19.0763	33.6905	0.4886
RF*	1015.7416	0.3845	15.1202	31.8707	0.5424
BAG*	**884.8778**	**0.3674**	**14.3677**	**29.7469**	**0.6013**
SVR*	1934.5500	1.4254	21.1660	43.9835	0.1284
LSSVR*	1311.5350	1.2829	18.2171	36.2151	0.4091
Panel B: with sentiment and extra covariates					
HAR**	1298.6001	8.7030	21.6841	36.0361	0.4149
HAR-CJ**	1299.4404	1.4853	21.7684	36.0478	0.4145
HAR-RS-II**	1349.2130	2.0542	22.4713	36.7316	0.3921
LASSO**	1251.6195	1.3544	21.1397	35.3782	0.4361
BOOST**	1489.1772	4.9792	22.1760	38.5899	0.3291
RF**	1024.0401	0.3846	15.3587	32.0006	0.5386
BAG**	**885.8634**	**0.3687**	**14.3526**	**29.7635**	**0.6009**
SVR**	1934.5502	1.4254	21.1660	43.9835	0.1284
LSSVR**	1336.3343	1.2665	17.7219	36.5559	0.3979

The best result under each criterion is highlighted in boldface

mainstream asset markets RV does not affect BTC/USD volatility, which reinforces the fact that crypto-assets are sometimes considered as a hedging device for many investment companies.[17]

Last, we use the GW test to formally explore if there are no differences in forecast accuracy between the panels in Table 13. For each estimator, we present the *p*-

[17]PwC-Elwood [36] suggests that the capitalization of cryptocurrency hedge funds increases at a steady pace since 2016.

Table 13 Giacomini–White test results

	HAR*	HAR-CJ*	RS-II*	LASSO*	BOOST*	RF*	BAG*	SVR*	LSSVR*	HAR**	HAR-CJ**	RS-II**	LASSO**	BOOST**	RF**	BAG**	SVR**
HAR*	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-CJ*	0.2800	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HAR-RS-II*	**0.0308**	**0.0370**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LASSO*	0.1862	0.7852	**0.0159**	–	–	–	–	–	–	–	–	–	–	–	–	–	–
BOOST*	0.1080	0.1468	**0.0382**	0.1496	–	–	–	–	–	–	–	–	–	–	–	–	–
RF*	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0011**	–	–	–	–	–	–	–	–	–	–	–	–
BAG*	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0761	–	–	–	–	–	–	–	–	–	–	–
SVR*	0.7212	0.8405	0.3878	0.8732	0.2890	**0.0000**	**0.0000**	–	–	–	–	–	–	–	–	–	–
LSSVR*	**0.0000**	**0.0001**	**0.0000**	**0.0000**	0.5719	**0.0002**	**0.0004**	**0.0077**	–	–	–	–	–	–	–	–	–
HAR**	0.9044	0.4386	**0.0467**	0.2991	0.1156	**0.0000**	**0.0000**	0.7276	**0.0000**	–	–	–	–	–	–	–	–
HAR-CJ**	0.8329	0.1081	0.1848	0.2450	0.1071	**0.0000**	**0.0000**	0.6919	**0.0001**	0.7446	–	–	–	–	–	–	–
HAR-RS-II**	0.0597	0.0511	0.8469	**0.0244**	**0.0390**	**0.0000**	**0.0000**	0.3653	**0.0000**	**0.0270**	0.1627	–	–	–	–	–	–
LASSO**	**0.0421**	0.2672	**0.0050**	0.0524	0.2018	**0.0000**	**0.0000**	0.9853	**0.0001**	**0.0277**	**0.0421**	**0.0042**	–	–	–	–	–
BOOST**	0.8039	0.7097	0.8906	0.6749	**0.0234**	**0.0000**	**0.0000**	0.6384	**0.0246**	0.7977	0.8313	0.8769	0.5775	–	–	–	–
RF**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0016**	0.7798	0.0725	**0.0000**	**0.0001**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	–	–	–
BAG**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0002**	0.6173	**0.0045**	**0.0004**	**0.0054**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.5515	–	–
SVR**	0.7212	0.8405	0.3878	0.8732	0.2890	**0.0000**	**0.0000**	0.4629	**0.0077**	0.7276	0.6919	0.3653	0.9853	0.6384	**0.0000**	**0.0004**	–
LSSVR**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.3831	**0.0025**	**0.0022**	**0.0011**	**0.0038**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0127**	**0.0015**	**0.0194**	**0.0011**

p-values smaller than 5% are highlighted in boldface

values from different covariate groups in bold. Each of these p-values exceeds 5%, which support our finding that mainstream asset RV data does not improve forecasts sharply, unlike the inclusion of social media data.

8 Conclusion

In this chapter, we compare the performance of numerous econometric and machine learning forecasting strategies to explain the short-term realized volatility of the Bitcoin market. Our results first complement a rapidly growing body of research that finds benefits from using machine learning techniques in the context of financial forecasting. Our application involves forecasting an asset that exhibits significantly more variation than much of the earlier literature which could present challenges in settings such as ours with fewer than 800 observations. Yet, our result further highlights that what drives the benefits of machine learning is the accounting for nonlinearities and there are much smaller gains from using regularization or cross-validation. Second, we find substantial benefits from using social media data in our forecasting exercise that hold irrespective of the estimator. These benefits are larger when we consider new econometric tools to more flexibly handle the difference in the timing of the sampling of social media and financial data.

Taken together, there are benefits from using both new data sources from the social web and predictive techniques developed in the machine learning literature for forecasting financial data. We suggest that the benefits from these tools will likely increase as researchers begin to understand why they work and what they measure. While our analysis suggests nonlinearities are important to account for, more work is needed to incorporate heterogeneity from heteroskedastic data in machine learning algorithms.[18] We observe significant differences between SVR and LSSVR so the change in loss function can explain a portion of the gains within machine learning relative to econometric strategies, but not to the same extent as nonlinearities, which the tree-based strategies also account for and use a similar loss function based on SSR.

Our investigation focused on the performance of what are currently the most popular algorithms considered by social scientists. There have been many advances developing powerful algorithms in the machine learning literature including deep learning procedures which consider more hidden layers than the neural network procedures considered in the econometrics literature between 1995 and 2015. Similarly, among tree-based procedures, we did not consider eXtreme gradient boosting which applies more penalties in the boosting equation when updating

[18] Lehrer and Xie [26] pointed out that all of the machine learning algorithms considered in this paper assume homoskesdastic data. In their study, they discuss the consequences of heteroskedasticity for these algorithms and the resulting predictions, as well as propose alternatives for this data.

trees and residual compared to the classic boosting method we employed. Both eXtreme gradient boosting and deep learning methods present significant challenges regarding interpretability relative to the algorithms we examined in the empirical exercise.

Further, machine learning algorithms were not developed for time series data and more work is needed to develop methods that can account for serial dependence, long memory, as well as the consequences of having heterogeneous investors.[19] That is, while time series forecasting is an important area of machine learning (see [19, 30], for recent overviews that consider both one-step-ahead and multi-horizon time series forecasting), concepts such as autocorrelation and stationarity which pervade developments in financial econometrics have received less attention. We believe there is potential for hybrid approaches in the spirit of Lehrer and Xie [25] with group LASSO estimators. Further, developing machine learning approaches that consider interpretability appears crucial for many forecasting exercises whose results need to be conveyed to business leaders who want to make data-driven decisions. Last, given the random sample of Twitter users from which we measure sentiment, there is likely measurement error in our sentiment and our estimate should be interpreted as a lower bound.

Given the empirical importance of incorporating social media data in our forecasting models, there is substantial scope for further work that generates new insights with finer measures of this data. For example, future work could consider extracting Twitter messages that only capture the views of market participants rather than the entire universe of Twitter users. Work is also needed to clearly identify bots and consider how best to handle fake Twitter accounts. Similarly, research could strive to understand shifting sentiment for different groups on social media in response to news events. This can help improve our understanding of how responses to unexpected news leads lead investors to reallocate across asset classes.[20]

In summary, we remain at the early stages of extracting the full set of benefits from machine learning tools used to measure sentiment and conduct predictive analytics. For example, the Bitcoin market is international but the tweets used to estimate sentiment in our analysis were initially written in English. Whether the findings are robust to the inclusion of Tweets posted in other languages represents

[19] Lehrer et al. [27] considered the use of model averaging with HAR models to account for heterogeneous investors.

[20] As an example, following the removal of Ivanka Trump's fashion line from their stores, President Trump issued a statement via Twitter:

> My daughter Ivanka has been treated so unfairly by @Nordstrom. She is a great person – always pushing me to do the right thing! Terrible!

The general public response to this Tweet was to disagree with President Trump's stance on Nordstrom so aggregate Twitter sentiment measures rose and the immediate negative effects from the Tweet on Nordstrom stock of a decline of 1% in the minute following the tweet were fleeting since the stock closed the session posting a gain of 4.1%. See http://www.marketwatch.com/story/nordstrom-recovers-from-trumps-terrible-tweet-in-just-4-minutes-2017-02-08 for more details on this episode.

an open question for future research. As our understanding of how to account for real-world features of data increases with these data science tools, the full hype of machine learning and data science may be realized.

Acknowledgments We wish to thank Yue Qiu, Jun Yu, and Tao Zeng, seminar participants at Singapore Management University, for helpful comments and suggestions. Xie's research is supported by the Natural Science Foundation of China (71701175), the Chinese Ministry of Education Project of Humanities and Social Sciences (17YJC790174), and the Fundamental Research Funds for the Central Universities. Contact Tian Xie (e mail: xietian@shufe.edu.cn) for any questions concerning the data and/or codes. The usual caveat applies.

Appendix: Data Resampling Techniques

Substantial progress has been made in the machine learning literature on quickly converting text to data, generating real-time information on social media content. In this study, we also explore the benefits of incorporating an aggregate measure of social media sentiment, the Wall Street Journal-IHS Markit US Sentiment Index (USSI) in forecasting the Bitcoin RV. However, data timing presents a serious challenge in using minutely measures of the USSI to forecast the daily Bitcoin RV. To convert minutely USSI measure to match the sampling frequency of Bitcoin RV, we hereby introduce a few popular data resampling techniques.

Let y_{t+h} be target h-step-ahead future a low-frequency variable (e.g., the daily realized variance) that is sampled at periods denoted by a time index t for $t = 1, \ldots, n$. Consider a higher-frequency (e.g., the USSI) predictor $\boldsymbol{X}_t^{hi}$ that is sampled m times within the period of t:

$$\boldsymbol{X}_t^h \equiv \left[X_t^{hi}, X_{t-\frac{1}{m}}^{hi}, \ldots, X_{t-\frac{m-1}{m}}^{hi} \right]^\top. \tag{20}$$

A specific element among the high-frequency observations in $\boldsymbol{X}_t^{hi}$ is denoted by $X_{t-\frac{i}{m}}^{hi}$ for $i = 0, \ldots, m-1$. Denoting $L^{i/m}$ as the lag operator, then $X_{t-\frac{i}{m}}^{hi}$ can be reexpressed as $X_{t-\frac{i}{m}}^{hi} = L^{i/m} X_t^{hi}$ for $i = 0, \ldots, m-1$.

Since $\boldsymbol{X}_t^h$ on y_{t+h} is measured at different frequencies, we need to convert the higher-frequency data to match the lower-frequency data. A simple average of the high-frequency observations $\boldsymbol{X}_t^h$:

$$\bar{X}_t = \frac{1}{m} \sum_{i=0}^{m-1} L^{i/m} X_t^h,$$

where $\bar{X}_t$ is likely the easiest way to estimate a low-frequency X_t that can match the frequency of y_{t+h}. With the variables y_{t+h} and $\bar{X}_t$ being measured in the same time domain, a regression approach is simply

$$y_{t+h} = \alpha + \gamma \bar{X}_t + \epsilon_t = \alpha + \frac{\gamma}{m} \sum_{i=0}^{m-1} L^{i/m} X_t^h + \epsilon_t, \tag{21}$$

where α is the intercept and γ is the slope coefficient on the time-averaged $\bar{X}_t$. This approach assumes that each element in $\boldsymbol{X}_t^h$ has an identical effect on explaining y_{t+h}.

These homogeneity assumptions may be quite strong in practice. One could assume that each of the slope coefficients for each element in $\boldsymbol{X}_t^{hi}$ is unique. Following Lehrer et al. [28], extending Model (21) to allow for heterogeneous effects of the high-frequency observations generates

$$y_{t+h} = \alpha + \sum_{i=0}^{m-1} \gamma_i L^{i/m} X_t^{hi} + \epsilon_t, \tag{22}$$

where γ_i represents a set of slope coefficients for all high-frequency observations $X_{t-\frac{i}{m}}^{hi}$.

Since γ_i is unknown, estimating these parameters can be problematic when m is a relatively large number. The heterogeneous mixed data sampling (H-MIDAS) method by Lehrer et al. [28] uses a step function to allow for heterogeneous effects of different high-frequency observations on the low-frequency dependent variable. A low-frequency $\bar{X}_t^{(l)}$ can be constructed following

$$\bar{X}_t^{(l)} \equiv \frac{1}{l} \sum_{i=0}^{l-1} L^{i/m} X_t^{hi} = \frac{1}{l} \sum_{i=0}^{l-1} X_{t-\frac{i}{m}}^{hi}, \tag{23}$$

where l is a predetermined number and $l \leq m$. Equation (23) implies that we compute variable $\bar{X}_t^{(l)}$ by a simple average of the first l observations in $\boldsymbol{X}_t^{hi}$ and ignored the remaining observations. We consider different values of l and group all $\bar{X}_t^{(l)}$ into $\tilde{\boldsymbol{X}}_t$ such that

$$\tilde{\boldsymbol{X}}_t = \left[\bar{X}_t^{(l_1)}, \bar{X}_t^{(l_2)}, \ldots, \bar{X}_t^{(l_p)} \right],$$

where we set $l_1 < l_2 < \cdots < l_p$. Consider a weight vector $\boldsymbol{w} = \left[w_1, w_2, \ldots, w_p\right]^\top$ with $\sum_{j=1}^{p} w_j = 1$; we can construct regressor X_t^{new} as $X_t^{new} = \tilde{\boldsymbol{X}}_t \boldsymbol{w}$. The regression based on the H-MIDAS estimator can be expressed as

$$y_{t+h} = \beta X_t^{new} + \epsilon_t = \beta \sum_{s=1}^{p} \sum_{j=s}^{p} \frac{w_j}{l_j} \sum_{i=l_{s-1}}^{l_s-1} L^{i/m} X_t^h + \epsilon_t = \beta \sum_{s=1}^{p} \sum_{i=l_{s-1}}^{l_s-1} w_s^* L^{i/m} X_t^h + \epsilon_t, \tag{24}$$

where $l_0 = 0$ and $w_s^* = \sum_{j=s}^{p} \frac{w_j}{l_j}$.

The weights $\boldsymbol{w}$ play a crucial role in this procedure. We first estimate $\widehat{\beta \boldsymbol{w}}$ following

$$\widehat{\beta \boldsymbol{w}} = \underset{\boldsymbol{w} \in \mathcal{W}}{\arg\min} \left\| y_{t+h} - \tilde{\boldsymbol{X}}_t \cdot \beta \boldsymbol{w} \right\|^2$$

by any appropriate econometric method necessary, where $\mathcal{W}$ is some predetermined weight set. Once $\widehat{\beta \boldsymbol{w}}$ is obtained, we estimate the weight vector $\hat{\boldsymbol{w}}$ by rescaling following

$$\hat{\boldsymbol{w}} = \frac{\widehat{\beta \boldsymbol{w}}}{\mathrm{Sum}(\widehat{\beta \boldsymbol{w}})},$$

since the coefficient β is a scalar.

References

1. Andersen, T. G., & Bollerslev, T. (1998). Answering the skeptics: yes, standard volatility models do provide accurate forecasts. *International Economic Review, 39*(4), 885–905.
2. Andersen, T., Bollerslev, T., Diebold, F., & Ebens, H. (2001). The distribution of realized stock return volatility. *Journal of Financial Economics, 61*(1), 43–76.
3. Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2001). The distribution of realized exchange rate volatility. *Journal of the American Statistical Associatio, 96*(453), 42–55.
4. Andersen, T. G., Bollerslev, T., & Diebold, F. X. (2007). Roughing it up: including jump components in the measurement, modelling, and forecasting of return volatility. *The Review of Economics and Statistics, 89*(4), 701–720.
5. Baker, M., & Wurgler, J. (2007). Investor sentiment in the stock market. *Journal of Economic Perspectives, 21*(2), 129–152.
6. Ban, G.-Y., Karoui, N. E., & Lim, A. E. B. (2018). Machine learning and portfolio optimization. *Management Science, 64*(3), 1136–1154.
7. Blair, B. J., Poon, S.-H., & Taylor, S. J. (2001). Forecasting S&P 100 volatility: the incremental information content of implied volatilities and high-frequency index returns. *Journal of Econometrics, 105*(1), 5–26.
8. Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123–140.
9. Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

10. Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. New York: Chapman and Hall/CRC.
11. Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics, 7*(2), 174–196.
12. Corsi, F., Audrino, F., & Renó, R. (2012). HAR modelling for realized volatility forecasting. In *Handbook of volatility models and their applications* (pp. 363–382). Hoboken: : John Wiley & Sons.
13. Coulombe, P. G., Leroux, M., Stevanovic, D., & Surprenant, S. (2019). How is machine learning useful for macroeconomic forecasting? In *Cirano Working Papers, CIRANO*. https://economics.sas.upenn.edu/system/files/2019-03/GCLSS_MC_MacroFcst.pdf
14. Craioveanu, M., & Hillebrand, E. (2012). *Why it is ok to use the har-rv (1, 5, 21) model*. Technical Report 1201, University of Central Missouri. https://ideas.repec.org/p/umn/wpaper/1201.html
15. Dacorogna, M. M., Müller, U. A., Nagler, R. J., Olsen, R. B., & Pictet, O. V. (1993). A geographical model for the daily and weekly seasonal volatility in the foreign exchange market. *Journal of International Money and Finance, 12*(4), 413–438.
16. Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., & Vapnik, V. (1996). Support vector regression machines. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems* (Vol. 9, pp. 155–161). Cambridge: MIT Press.
17. Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1615–1625). Stroudsburg: Association for Computational Linguistics.
18. Giacomini, R., & White, H. (2006). Tests of conditional predictive ability. *Econometrica, 74*(6), 1545–1578.
19. Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies, 33*(5), 2223–2273. Society for Financial Studies.
20. Hansen, P. R., & Lunde, A. (2005). A forecast comparison of volatility models: does anything beat a garch(1,1)? *Journal of Applied Econometrics, 20*(7), 873–889.
21. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning. Springer series in statistics*. New York, NY: Springer.
22. Huang, X., & Tauchen, G. (2005). The relative contribution of jumps to total price variance. *Journal of Financial Econometrics, 3*(4), 456–499.
23. Ke, Z. T., Kelly, B. T., & Xiu, D. (2019). Predicting returns with text data. In *NBER Working Papers 26186*. Cambridge: National Bureau of Economic Research, Inc.
24. LaFon, H. (2017). Should you jump on the smart beta bandwagon? https://money.usnews.com/investing/funds/articles/2017-08-24/are-quant-etfs-worth-buying
25. Lehrer, S. F., & Xie, T. (2017). Box office buzz: does social media data steal the show from model uncertainty when forecasting for hollywood? *Review of Economics and Statistics, 99*(5), 749–755.
26. Lehrer, S. F., & Xie, T. (2018). The bigger picture: Combining econometrics with analytics improve forecasts of movie success. In *NBER Working Papers 24755*. Cambridge: National Bureau of Economic Research.
27. Lehrer, S. F., Xie, T., & Zhang, X. (2019). *Does adding social media sentiment upstage admitting ignorance when forecasting volatility?* Technical Report, Queen's University, NY. Available at: http://econ.queensu.ca/faculty/lehrer/mahar.pdf
28. Lehrer, S. F., Xie, T., & Zeng, T. (2019). Does high frequency social media data improve forecasts of low frequency consumer confidence measures? In *NBER Working Papers 26505*. Cambridge: National Bureau of Economic Research.
29. Mai, F., Shan, J., Bai, Q., Wang, S., & Chiang, R. (2018). How does social media impact bitcoin value? A test of the silent majority hypothesis. *Journal of Management Information Systems, 35*, 19–52.
30. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: concerns and ways forward. *PloS One, 13*(3), Article No. e0194889. https://doi.org/10.1371/journal.pone.0194889

31. Medeiros, M. C., Vasconcelos, G. F. R., Veiga, Á., & Zilberman, E. (2019). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *Journal of Business & Economic Statistics, 39*(1), 98–119. https://doi.org/10.1080/07350015.2019.1637745
32. Mincer, J., & Zarnowitz, V. (1969). The evaluation of economic forecasts. In *Economic forecasts and expectations: Analysis of forecasting behavior and performance* (pp. 3–46). Cambridge: National Bureau of Economic Research, Inc.
33. Müller, U. A., Dacorogna, M. M., Davé, R. D., Pictet, O. V., Olsen, R. B., & Ward, J. (1993). *Fractals and intrinsic time – a challenge to econometricians*. Technical report SSRN 5370. https://ssrn.com/abstract=5370
34. Patton, A. J., & Sheppard, K. (2015). Good volatility, bad volatility: signed jumps and the persistence of volatility. *The Review of Economics and Statistics, 97*(3), 683–697.
35. Probst, P., Boulesteix, A., & Bischl, B. (2019). Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research, 20*, 1–32.
36. PwC-Elwood. (2019). 2019 crypto hedge fund report. https://www.pwc.com/gx/en/financial-services/fintech/assets/pwc-elwood-2019-annual-crypto-hedge-fund-report.pdf
37. Schumaker, R. P., Zhang, Y., Huang, C.-N., & Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems, 53*(3), 458–464.
38. Suykens, J., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*, 293–300.
39. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B, 58*, 267–288.
40. Vapnik, V. N. (1996). *The nature of statistical learning theory*. New York, NY: Springer-Verlag.
41. Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82.
42. Xie, T. (2019). Forecast bitcoin volatility with least squares model averaging. *Econometrics, 7*(3), 40:1–40:20.