Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Erasmus School Of Economics

Forecasting Cryptocurrency Price Volatility

Lachman Prashand (523223)

Supervisor:	Aishameriane Venes Schmidt
Second assessor:	Kleen, O
Date final version:	2nd July 2023

Abstract

Accurate prediction of cryptocurrency price volatility is crucial for investors and traders as it enables them to make well-informed decisions regarding risk management and investment strategies, leading to favorable financial outcomes. This study aims to examine six different models, comprising two statistical models and four econometric models, in order to predict cryptocurrency price volatility. The statistical models investigated are the Bayesian Additive Regression Tree (BART) and Heteroscedastic BART, while the econometric models considered are Generalized Auto Regressive Heteroscedastic (GARCH), Exponential GARCH (E-GARCH), Asymmetric Power ARCH (aPARCH), and Glosten-Jagannathan-Runkle GARCH (GJR-GARCH). To evaluate the models' performance, a publicly available dataset from Yahoo Finance spanning the period from January 1, 2014, to January 1, 2023, is utilized. The dataset is splitted into an in-sample period and an out-of-sample period. During the in-sample period, various information criteria and graphical diagnostic tools are employed to compare the model's performance. The out-of-sample period is then utilized to construct 1-step ahead forecasts, which are assessed using loss functions to determine the model's predictive accuracy. The findings of this study reveal that the sGARCH and GJR-GARCH models exhibit the most suitable performance among the GARCH models, while among the statistical models, the HBART model demonstrates a superior fit to the data. Additionally, the GJR-GARCH model demonstrates superior forecasting accuracy when compared to the best fitted models, due to its ability to capture the leverage effect.

1 Introduction

Accurate prediction of future cryptocurrency price volatility is of great interest to investors and traders. It plays a crucial role in enabling efficient risk management strategies, optimizing investment decisions, and achieving favorable financial outcomes. Inaccurate predictions can lead to missed opportunities and significant financial losses. Moreover, the unpredictable nature of cryptocurrency price volatility can affect market stability and impede the widespread adoption of cryptocurrencies as a mainstream asset class. The main thrust of this article is to propose a heteroscedastic elaboration of Bayesian additive regression trees (HBART) Pratola et al. (2020), that accounts for potential heteroscedasticity as a statistical approach for predicting cryptocurrency price volatility and compare it's performance with traditional methods. Therefore, the following research question is constructed: *"Does HBART via Multiplicative Regression Trees outperform traditional methods in predicting cryptocurrency price volatility?"*.

Extensive research has focused on predicting cryptocurrency price volatility, with continuous proposals for potential improvements. Econometric models, particularly Generalized Auto Regressive Conditional Heteroscedastic (GARCH) Naimy & Hayek (2018) models and their extensions, have been widely employed in this domain. Mostafa et al. (2021) and Caporale & Zekokh (2019) showed that these extensions, such as Exponential GARCH (EGARCH), Asymmetric Power Autoregressive Conditional Heteroscedastic (aPARCH), and Glosten Jagannathan Runkle (GJR-GARCH), exhibit promising capabilities in capturing additional effects and producing accurate forecasts. Furthermore, Pratola et al. (2020) showed that the Hierarchical Bayesian Additive Regression Trees (HBART) model, implemented through multiplicative regression trees, is capable of accurately capturing heteroscedasticity in high-dimensional data. For these reasons, we are interested in using this model as a statistical approach in predicting cryptocurrency price volatility.

To investigate the research question, this study utilizes a dataset obtained from Yahoo Finance, specifically focusing on Bitcoin pricing and volume data. Analyzing Bitcoin data offers several advantages due to its market dominance, extensive historical dataset, availability of research and tools, and industry significance. The dataset comprises 3029 daily observations spanning from January 1, 2014, to January 1, 2023. For the purpose of model fitting and parameter estimation, the initial 1817 data points are utilized, while the remaining 40% is set aside for validation, which includes 1212 observations.

The methodology employed in this study encompasses both statistical and econometric models to construct volatility forecasts. The statistical approach involves Homoscedastic Bayesian Additive Regression Trees (BART) Chipman et al. (2010) and Heteroscedastic BART (HBART) Pratola et al. (2020), while the econometric approach utilizes (standard)GARCH, EGARCH, aPARCH, and GJR-GARCH models. The comparison of fitted GARCH models involves the use of information criteria Heij et al. (2004), such as Akaike Information Criteria (AIC) Heij et al. (2004), Bayesian Information Criteria (BIC), and Hannan-Quinn (HQ). Additionally, graphical diagnostic tools like Quantile-Quantile plots and H-Evidence plots are utilized to assess the fit of BART models. The resulting forecasts from the best-fitted GARCH and BART models are evaluated using well-known metrics Heij et al. (2004), such as Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and robust Median Absolute Deviation (MAD).

The main findings indicate that among the sGARCH model and it's extensions, the sGARCH model outperforms the others based on AIC and HQ criteria, while sGARCH and GJR-GARCH exhibit similar data fits according to BIC. Consequently, sGARCH and GJR-GARCH are chosen as the best-fitted GARCH models. Comparing the Bayesian Regression Tree models reveals that the HBART model captures realized variance more accurately. Additionally, the H-evidence plot confirms the presence of statistically significant heteroscedasticity. Therefore, the HBART model is selected as the best-fitted model among the statistical model approach. Evaluating the predictive performance between the two best GARCH models and the HBART model using loss functions demonstrates that GJR-GARCH exhibits superior predictive performance.

The structure of this paper is organized as follows: Section 1 provides an introduction to the research topic. Section 2 presents an overview of the relevant literature. Section 3 describes the dataset and its key characteristics. Section 4 outlines the methodology, including the models, and techniques utilized. Section 5 presents the main findings, followed by a conclusion in Section 6.

2 Literature & Theory

The prediction of cryptocurrency price volatility has received significant attention in the existing academic literature. This is primarily driven by the crucial role it plays in aiding investors and traders in making informed investment decisions. There has been a growing body of research on Bitcoin volatility modeling in recent years, comprehensive comparisons of volatility models both in-sample and out-of-sample remain scarce. The volatility of Bitcoin has been extensively explored in academic literature due to its significant implications. Numerous studies have examined the impact of external shocks on Bitcoin volatility. For example, Aysan et al. (2019) found that Bitcoin volatility increases during periods of high geopolitical risks. Similarly, Wang et al. (2020) observed a positive association between high economic policy uncertainty and increased Bitcoin volatility. Additionally, Lyócsa & Molnár (2016) discovered that Bitcoin volatility tends to rise on days when cryptocurrency-related hacking attacks occur. Furthermore, Ma & Tanizaki (2019)identified a recurring pattern of higher Bitcoin volatility on Mondays and Thursdays. In addition to individual factors, researchers have also investigated the co-movements among different cryptocurrencies. Studies conducted by Beneki et al. (2019) and Katsiampa (2019) have explored the interrelationships and co-movements among various cryptocurrencies, contributing to our understanding of the factors influencing Bitcoin volatility and shedding light on the interconnectedness of different cryptocurrencies.

In order to answer our research question, we aim to answer the following sub-question, which can help guide the research and provide a more comprehensive understanding of the performance and limitations of the HBART method in predicting the volatility of cryptocurrency prices: *"Does HBART via Multiplicative Regression Trees outperform econometrics methods in predicting cryptocurrency price volatility? "*. Previous studies have primarily focused on GARCH models as a econometric approach in predicting cryptocurrency price volatility. In this context, Caporale & Zekokh (2019) conducted a notable study that provides a detailed comparison of GARCH models, not only for Bitcoin but also for other cryptocurrencies such as Ethereum, Ripple, and Litecoin. Their primary emphasis was on Value at Risk and Expected Shortfall.

Notably, a study conducted by Naimy & Hayek (2018) compared the predictive abilities of the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) (1,1), Exponentially Weighted Moving Average (EWMA), and Exponential Generalized Autoregressive Conditional Heteroscedasticity (EGARCH) (1,1) models for predicting Bitcoin price volatility. The findings revealed that the EGARCH (1,1) model outperformed the GARCH (1,1) and EWMA models in both in-sample and out-of-sample contexts, exhibiting increased accuracy during the out-of-sample period.

In another recent study conducted in 2021, Mostafa et al. (2021) found that GJR-GARCH(p,q) models provide a better estimate for cryptocurrency volatility than GARCH(p,q) models. Similarly, Bergsli et al. (2022) investigated accurate forecasting of Bitcoin price volatility and concluded that among the GARCH models, EGARCH and aPARCH performed the best. They employed realized variance as a proxy for true variance to evaluate the models' performance.

However, our study goes beyond GARCH models by including HBART model in the comparison. By considering both GARCH and BART models, our aim is to offer a comprehensive analysis of volatility modeling for Bitcoin, shedding light on the relative performance of these two model groups. The BART and HBART methods can be theoretically justified as suitable approaches for analyzing high-dimensional data, as demonstrated in the study by Pratola et al. (2020) on heteroscedasticity. When it comes to predicting Bitcoin price volatility, these models offer potential advantages in capturing heteroscedasticity by accommodating varying volatility structures across different regions of the predictor space. Moreover, these models are grounded in a Bayesian framework, which enables the integration of prior knowledge or beliefs concerning the data and parameters. By incorporating prior information about volatility patterns or market dynamics, these models can enhance the modeling process and improve the accuracy of volatility predictions.

3 Data

In this research, data is collected from multiple sources. One simulated and two emperical data sets are considered to compare the performance of the HBART and BART methods. At last, a third emperical data set is considered for predicting cryptocurrency price volatility. Some relevant plots, summary statistics and histograms of these data can be found in the appendix A

First, 500 observations (in-sample period) for the model $y_i = 4x_i^2 + 0.2e^{2x_i}Z_i$ are simuated, where each x_i is drawn independently from the uni-form distribution on (0, 1) and each Z_i is drawn independently from the standard normal distribution. Thereafter, the out of sample period is simulated in the exact same way.

Second, alcohol consumption data, collected from the supplemental of Pratola et al. (2020) in the journal of Taylor and Francis is considered. This data set consist over 2463 observations of 36 variables. The variable y_i is selected as the response variable, which refers to the number of alcoholic beverages consumed. The explanatory variables in the analysis encompass demographic and physical characteristics of the respondents, along with a crucial treatment variable indicating whether they received advice from a physician. After creating dummy variables for categorical variables, the dimension of the variable set is 35. For training purposes the first 1477 observations of the data are used, while the testing phase consisted of 986 observations.

Third, the used cars sales example is considered which has been collected from the rbart package McCulloch et al. (2019). We examine a sample of 1000 observations from a proprietary dataset of used car sales spanning the years 1994 to 2013. The data set consists of the response variable price, two continous variables and 4 categorical variables. Expanding the category variables into dummy variables results in a total of 15 predictor variables. For training purposes the first 600 observations of the data are used, with the last 40% reserved for validation, while the testing phase consist of 400 observations.

Furthermore, in this research, a publicly available data set from Yahoo Finance is used to forecast Bitcoin price volatility. The data set consists over 3029 daily observations of Bitcoin prices spanning a period from 01-01-2014 to 01-01-2023. This data is used to investigate the underlying patterns and trends in the Bitcoin price data, with the aim to accurately forecast price volatility. The prices were extracted and transformed into log-returns and realized variances, which are calculated using a grid of five days.

4 Methodology

This section illustrates the different models employed for forecasting bitcoin price volatility. First the BART and HBART are considered as a statistical approach. Second, the well known GARCH model is condisdered with some extensions, Exponetional GARCH (EGARCH), Asymmetric Power Autoregressive Conditional Heteroskedasticity (aPARCH) and Glosten-Jagannathan-Runkle GARCH (GJR-GARCH).

4.1 Bayesian Additive Regression Trees (BART)

As a first step, statistical methods are examined. In particular, Heteroscedastic Bayesian Additive Regressive Tree (HBART) and Homoscedastic Bayesian Additive Regression Tree (BART) are considered.

4.1.1 Heteroscedastic BART

The Bayesian Additive Regression Tree (BART) method Chipman et al. (2010) is a powerful statistical modeling technique that combines decisiob trees and Bayesian inference to estimate complex relationships between predictors and response variables. The data generating process of general model Pratola et al. (2020) is given by $y(x_i) = f(x_i)+s(x_i)Z_i$, with $Z_i \sim N(0,1)$ and $x_i = (x_1, ..., x_d)$ is a d-dimensional vector of predictor variables. In case of the Homoscedastic process the variance component has contant variance assumption, $s(x_i) = \sigma$, while the Heteroscedastic process assumes non constant variance $s(x_i)$. The mean function $f(x_i)$ and variance funtion $s(x_i)^2$ are both modelled using Bayesian regression trees.

Bayesian regression trees are a flexible method for modeling multidimensional regression functions without relying on fixed assumptions. Instead, these trees are constructed based on observed data, allowing them to adapt to the underlying patterns in the data. Each Bayesian regression tree is composed of interior nodes (referred to as T) and terminal nodes (referred to as M), which are associated with specific parameter values. The tree structure is established using recursive binary partitioning, wherein each interior node (η_i) has a left child ($l(\eta_i)$) and

a right child ($r(\eta_i)$). Additionally, every node, except for the root, has a parent node denoted as $p(\eta_i)$. To distinguish between nodes, a unique integer identifier (e.g., i) is assigned, with the root node being labeled as node 1.

The structure of regression trees consists of internal nodes that contain split rules based on predictor variables and cutpoints. These split rules determine how the tree branches at each internal node. The information regarding these split rules and the arrangement of nodes and edges forms the tree structure, denoted as T. To explain further, let's consider a matrix X with dimensions $n \times d$, where n represents the number of observations and d represents the number of predictor variables. Each column in X represents a predictor variable (v), with values ranging from 1 to d. Each row (x) in the matrix corresponds to a specific combination of predictor variable settings observed in the data. In the context of an internal node, the split rule takes the form $x_v < c$, where x_v represents the selected split variable and c represents the chosen cutpoint for that variable. This rule determines the condition for branching at that particular internal node.

Moreover, in the Bayesian formulation, discrete probability distributions are assigned to the split variables (v) and the possible cutpoint values. This means that for each internal node, there is a distribution over the available predictor variables and their potential cutpoints. The internal structure of a tree (T) can be represented as a set of pairs, denoted as $T = \{(c_1, v_1), (c_2, v_2), \ldots\}$, where each pair indicates a specific cutpoint (c_i) associated with a predictor variable (v_i). To complete the Bayesian specification, prior distributions are also specified for the parameters in the terminal nodes. These parameters are represented by $M = \{\theta_1, \ldots, \theta_{n^g}\}$ in a tree with n^g terminal nodes. By assigning prior distributions to these parameters, the Bayesian approach enhances the model's ability to capture the characteristics of the data in the terminal nodes. Overall, the Bayesian regression tree framework incorporates discrete probability distributions for split variables and cutpoints, uses a set of pairs to represent the internal structure of the tree, and assigns prior distributions to the parameters associated with the terminal nodes (M). Taken all together, the Bayesian regression tree models the mean component $f(x_i)$ and variance component $s(x_i)$ in the following relations:

$$f(x_i) = \sum_{j=1}^{m} g(x_i, T_j, M_j) \tag{1}$$

$$s(x_i)^2 = \prod_{l=1}^{m'} h(x_i; T'_l, M'_l) \tag{2}$$

T_j encodes the structure of the jth tree for the mean, while $M_j = \{\mu_{j_1}, \ldots, \mu_{j_n g^j}\}$ represents the scalar terminal-node parameters for the mean in each tree. Similarly, T'_l encodes the structure of the lth tree for the variance, and in this case, we denote the bottom node mappings as $\theta \equiv s^2$, where $M'_l = \{s^2_{l_1}, \ldots, s^2_{l^h_n}\}$ represents the scalar terminal-node parameters for the variance, with $n^h = |M'_l|$. The heteroscedastic BART model is then obtained as the ensemble sum of m such baysian regression trees and as a product of m' bayesian regression trees, $y(x_i) = \sum_{j=1}^{m} g(x_i, T_j, M_j) + \sqrt{\prod_{j=1}^{m'} h(x_i, T'_l, M'_l)} Z_i$, $Z_i \sim N(0, 1)$, $y(x_i)$ is the observation called at predictor setting x_i. Note that this model is generalization of the homoscedastic process, where

$s(x_i) = \sigma$ Chipman et al. (2010).

The posterior of the heteroscedastic process is considered, to provide a comprehensive and flexible framework for incorporating prior knowledge or beliefs about the parameters into the data analysis process. One of the key advantages of using posteriors is that they allow for the integration of prior information with observed data in a coherent and principled manner. By combining the prior distribution, which represents our initial beliefs, with the likelihood function, which represents the data's probability given the parameters, the posterior distribution represents the updated beliefs about the parameters after considering the observed data. The posterior is defined as in equation 3, where $\pi(M_j|T_k) = \prod_{k=1}^{n^g} \pi(\mu_{jk})$ and $\pi(M'_j|T'_k) = \prod_{k=1}^{n^h} \pi(s^2_{lk})$. Furthermore, when considering μ_{jk} as a function, one may define the likelihood of the kth terminal node of the jth mean tree as in equation 4.

$$\pi(T, M, T', M'|y, X) \propto L(y|T, M, T', M', X) \prod_{j=1}^{m} \pi(T_j \pi(M_j|T_j) \prod_{l=1}^{m'} \pi(T'_l)\pi(M'_l|T'_l), \tag{3}$$

$$L(\mu_{jk}|.) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}s(x_i)} exp\left(\frac{-(r_i - \mu_{jk})^2}{2s^2(x_i)}\right) \tag{4}$$

Where n is the number of observations mapping the particular node and $r_i = y_i - \sum_{g \neq j} g(x_i; T_q; M_q)$, which is the difference between the observed value and the combined predictions from all other trees in the ensemble model. Furthermore, the conjugate posterior distribution of the mean component is given by, $\pi(\mu_{jk}) \sim N(0, \tau^2), \forall_{j,k}$. One could then define the full conditional for the mean compoment and integrated likelihood function as follows:

$$\pi(\mu_{jk}|.) \sim N\left(\frac{\sum_{i=1}^{n} \frac{r_i}{s^2(x_i)}}{\frac{1}{\tau^2} + \sum_{i=1}^{n} \frac{1}{s^2(x_i)}}, \frac{1}{\frac{1}{\tau^2} + \sum_{i=1}^{n} \frac{1}{s^2(x_i)}}\right) \tag{5}$$

$$\int L(\mu_{jk}|.)\pi(\mu_{jk})d\mu_{jk} \propto \left(\tau^2 \sum_{i=1}^{n} \frac{1}{s^2(x_i)} + 1\right)^{-1/2} \exp\left(-\frac{\frac{\tau^2}{2}\left(\sum_{i=1}^{n} \frac{r_i}{s^2(x_i)}\right)^2}{\tau^2 \sum_{i=1}^{n} \frac{1}{s^2(x_i)} + 1}\right) \tag{6}$$

This integration process allows for the incorporation of the prior information and beliefs about μ_{jk}, which makes it easy for the computation of the posterior distribution. The specified prior for μ_{jk}, implies a prior on the mean function, $f(x) \sim N(0, m\tau^2)$. This assumption implies that we believe the mean function is centered around zero and it's variability is determined by the value of m and the precision parameter τ^2. In this specification, $\tau = \frac{y_{max} - y_{min}}{2\sqrt{mk}}$, where y_{max} and y_{min} is the maximum and minimum response value from the observed data. Furthermore, the hyperparameter k is recommended for $k = 2$ in the homoscedastic process, while it is carefully selected using cross validation approach in the heteroscedastic process. A detailed explanation of the cross validation approach can be found in Subsection 4.3.3

When considering s_{lk} as a function, one may define the likelihood of the kth terminal node of the lth variance tree as:

$$L(s_{lk}^2|.) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi} s_{lk}} \exp\left(-\frac{e_i^2}{2s_{lk}^2}\right) \tag{7}$$

Where n denote the count of observations associated with a specific terminal node and $e^2 = \frac{\left(y(x_i) - \sum_{j=1}^{m} g(x_i; T_j; M_j)\right)^2}{s_{-l}^2(x_i)}$, where $s_{-l}^2(x_i) = \prod_{q \neq l} h(x_i; T_q'; M_q')$. A lower value of e^2 indicates that the predicted response are close to the observed data, suggesting a better fit of the model to the data. Furthermore, the conjugate posterior distribution for the variance comoponent is given as, $s_{lk}^2 \sim \chi^{-2}(v', \lambda'), \forall_{l,k}$, then one could define the full conditional for the variance compoment and integrated likelihood function as follows:

$$s_{lk}^2 \sim \chi^{-2}\left(\vartheta' + n, \frac{\vartheta'\lambda'^2 + \sum_{i=1}^{n} \frac{e_i^2}{s_{-l}^2(x_i)}}{\vartheta' + n}\right) \tag{8}$$

$$\int L(s^2(lk|.)\pi(s_{lk}^2) ds_{lk}^2 = \frac{\Gamma(\frac{v'+n}{2}) \left(\frac{v'\lambda'^2}{2}\right)^{\frac{v'}{2}}}{(2\pi)^{n/2} \prod_{i=1}^{n} s_{-l}(x_i) \Gamma(v'/2) (v'\lambda'^2 + \sum_{i=1}^{n} e_i^2)^{\frac{v'+n}{2}}} \tag{9}$$

The integrated likelihood helps assess the overall goodness of fit of the model, incorporating both the observed data and our prior beliefs. The numerator of the integrated likelihood includes the Gamma function and terms related to the shape and scale parameters of the Gamma distribution. These terms represent the prior distribution on the conditional variance. The denominator of the integrated likelihood includes a constant factor, the product of conditional variance estimates, the Gamma function related to the degrees of freedom parameter, and a term representing the sum of squared residuals plus the prior scale parameter.

The specified prior for our variance component is given by $s^2(x) \sim \prod_{i=1}^{m'} s_l^2$, with $s_l^2 \sim \chi^-2(v', \lambda')$. The prior parameters for HBART variance component, v', λ' are selected using a different approach. We start with the prior for the homoscedastic case by mathcing the prior means, which is done in the following ways, $E[\sigma^2] = \frac{v\lambda}{v-2}$, $E[s^2(x)] = \prod_{l=1}^{m'}, E[s_l^2] = \lambda^{m'} \left(\frac{v'}{v'-2}\right)^{m'}$. We then match the means by seperately matching the $\lambda piece$ and the $vpiece$ giving $\lambda' = \lambda^{\frac{1}{m'}}$, $v' = \frac{2}{\left(1-(1-\frac{2}{v})^{1/m'}\right)}$

4.1.2 Markov Chain Monte Carlo Algorithm

The heteroscedastic regression tree model, which utilizes a Markov Chain Monte Carlo (MCMC) algorithm, is employed for fitting purposes. The algorithm follows a series of steps outlined in Algorithm 1. Initially, the number of iterations for the algorithm, denoted as N_{mcmc}, is determined. Within each iteration ($j = 1$ to m), the algorithm proceeds with several key steps. In the initialization step, the model parameters are set to their initial values. This includes the tree structure (T), the leaf node parameters (M), as well as an alternate tree structure (T') and its associated leaf node parameters (M'). Subsequently, a new set of candidate values for the model parameters is proposed. These candidates are typically derived by perturbing the current

parameter values, allowing for exploration of the parameter space.

The algorithm then focuses on updating the tree structure (T) by employing specific moves tailored to trees. These moves involve operations like splitting or merging tree nodes, which aim to enhance the model's fit to the observed data. Afterwards, the parameters associated with the leaf nodes (M) are updated. This update is carried out by sampling from the conditional posterior distributions of the leaf node parameters, taking into account the current tree structure (T) and the observed data. The hyperparameters controlling the prior distributions of the model parameters are subsequently updated.

Next, the proposed candidate parameter values are evaluated for acceptance or rejection. This decision is made by comparing the posterior probability of the new parameter values with that of the current values. The acceptance probability is calculated using the Metropolis-Hastings ratio. Steps two to six are repeated for a predefined number of iterations (N_{mcmc}), enabling comprehensive exploration of the parameter space. This iterative process generates a Markov chain composed of samples drawn from the posterior distribution.Finally, the samples obtained from the Markov chain are employed to approximate the posterior distribution of the model parameters.

Algorithm 1: MCMC steps for the proposed HBART model.

Data: $y_1, ..., y_N; x_1, ..., x_N$

Result: Approximate posterior samples drawn from
$\pi(T, M, T', M'|y_1, ..., y_N; x_1, ..., x_N)$ for N_{mcmc} iterations

for $j = 1$ *to* m **do**
  Draw $T_j|\cdot$;
  Draw $M_j|T_j, \cdot$;
end
for $j = 1$ *to* m' **do**
  Draw $T'_j|\cdot$;
  Draw $M'_j|T'_j, \cdot$;
end

4.2 Generalized Autoregressive Conditional Heteroskedasticity

Next, we shift our attention to the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model, which plays a central role in our research. The GARCH models differ in the specification of the volatility process, which captures the time-varying nature of volatility in financial data. Let x_i represent the observed log-returns of Bitcoin prices, μ_i denote the conditional mean, σ_i^2 represent the volatility. The innovation Z_i is distributed according to a standard normal distribution. The GARCH model employed in this study is formulated as follows:

$$x_i = \mu_i + \sigma_i Z_i \tag{10}$$

The GARCH(p,q) model specifies the conditional variance process as:

$$\sigma_i^2 = \alpha_0 + \sum_{t=1}^{q} \alpha_t Z_{i-t}^2 + \sum_{j=1}^{p} \beta_j \sigma_{i-j}^2 \tag{11}$$

Here, p represents the number of lagged conditional variances, and q refers to the number of lags of the squared error. The GARCH parameters must satisfy certain conditions, such as $\alpha_0 > 0$, $\alpha_t \geq 0$ for $t = 1, 2, ..., q$, and $\beta_t > 0$ for $t = 2, ..., p$. These conditions ensure the model's stability and non-negativity of the conditional variance.

To estimate the parameters in the GARCH models, we employ the Maximum Likelihood Estimation (MLE) method. Heij et al. (2004) MLE is a statistical approach that seeks to find the parameter values that maximize the likelihood of observing the given data. By maximizing the likelihood function, we can obtain the most suitable parameter estimates for the GARCH model. Through the application of the GARCH models and the estimation procedure using MLE, we can capture the dynamic characteristics of Bitcoin price volatility and derive insights into its behavior. The GARCH framework provides a flexible and widely-used framework for modeling financial time series data, allowing us to assess the volatility patterns and make informed predictions.

4.2.1 E-GARCH

The Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) model is considered in this study. The EGARCH model is an extension of the GARCH model that incorporates asymmetry in the volatility dynamics, specifically addressing the phenomenon of leverage effects. In the EGARCH(p,q) model, the conditional variance σ_i^2 in equation 12 is modeled on the logarithmic scale as follows:

$$\log(\sigma_i^2) = \alpha_0 + \sum_{j=1}^{p} \beta_j \log(\sigma_{i-j}^2) + \sum_{i=1}^{q} \left(\gamma_i \left(\frac{Z_{i-1}}{\sqrt{\sigma_{i-t}^2}} \right) + \alpha_t \left(\frac{|Z_{i-1}|}{\sqrt{\sigma_{i-t}^2}} - E\left(\frac{|Z_{i-1}|}{\sqrt{\sigma_{i-t}^2}} \right) \right) \right) \tag{12}$$

In this equation, α_0 is the constant term, β_j represents the coefficients associated with the lagged logarithmic conditional variances $\log(\sigma_{i-j}^2)$, γ_t captures the leverage effects, α_t accounts for the asymmetry in the response to shocks, and $E(\cdot)$ represents the expectation operator. The parameter γ_t in the EGARCH model plays a crucial role in capturing the leverage effect. When γ_t is negative ($\gamma_t < 0$), it indicates that negative shocks have a stronger impact on future volatility compared to positive shocks of the same magnitude. This behavior is characteristic of many financial time series, including cryptocurrency prices. Conversely, when γ_t equals zero, it implies perfect symmetry between positive and negative shocks in terms of their influence on volatility. One advantage of using logarithmic transformations in the EGARCH model is that it eliminates the need to impose restrictions on the parameters to be positive. This allows for more flexibility in capturing the dynamics of volatility and better accommodates the asymmetric response to shocks. By incorporating asymmetry and leverage effects, the EGARCH model provides a more comprehensive framework for modeling cryptocurrency price volatility. It enables capturing the

impact of both positive and negative shocks on future volatility, thereby enhancing the accuracy of volatility forecasts and risk management strategies.

4.2.2 aPARCH

The Assymetric Power ARCH (APARCH) model is also considered in this study. The APARCH model is an extension of the traditional GARCH model that incorporates additional parameters to account for asymmetry and the power effect in the conditional variance equation. The power effect captures the nonlinearity observed in the relationship between past shocks and future volatility. The APARCH model introduces a power parameter, denoted as δ, which determines the impact of past shocks on future volatility. When δ is greater than 2, it captures the nonlinear effect and allows for heavier tails in the conditional distribution of the data. It enables the model to capture the presence of extreme events and deviations from normality more effectively. It is worth noting that the standard GARCH model is a special case of the APARCH model, where δ is fixed at 2 and the parameters γ_t are set to 0 for $t = 1, ..., q$. The conditional variance equation for the APARCH(p,q) model is given by:

$$\sigma_i^2 = \omega + \sum_{j=1}^{p} \beta_j \sigma_{i-j}^2 + \sum_{t=1}^{q} \alpha_t \left(|Z_{i-t}| - \gamma_t Z_{i-t} \right)^2 \tag{13}$$

In this equation, σ_i^2 represents the conditional variance at time i, ω is the constant term, β_j are the coefficients associated with the lagged conditional variances σ_{i-j}^2, α_t are the coefficients capturing the asymmetry in the model, γ_t represents the asymmetry parameters. By incorporating the power effect and allowing for asymmetry, the APARCH model provides a more flexible framework for modeling the conditional variance. It enables capturing the nonlinear relationship between shocks and volatility, as well as the asymmetry in the response to positive and negative shocks. This enhanced modeling capability contributes to improving the accuracy of cryptocurrency price volatility forecasts.

4.2.3 GJR-GARCH

The GJR-GARCH (Glosten-Jagannathan-Runkle GARCH) model is an extension of the traditional GARCH model that addresses certain limitations in capturing the characteristics of monthly excess returns. It specifically incorporates the concept of leverage effects, which refer to the asymmetric impact of positive and negative shocks on volatility. Unlike the standard GARCH model, which assumes symmetric effects, the GJR-GARCH model recognizes that volatility dynamics may differ depending on the sign of the shocks. The GJR-GARCH model introduces an additional term in the conditional variance equation to account for the leverage effect. This term is multiplied by an indicator function, denoted as I_{i-t}, which takes a value of 1 if Z_{i-t} is negative and 0 otherwise. The conditional variance equation is formulated as follows:

$$\sigma_i^2 = \alpha_0 + \sum_{t=1}^{q} (\alpha_t + \gamma_t I_{i-t}) Z_{i-t}^2 + \sum_{j=1}^{p} \beta_j \sigma_{i-j}^2 \tag{14}$$

In this equation, σ_i^2 represents the conditional variance at time t, α_0 is the constant term, α_t

and γ_t are coefficients associated with the leverage effect, and β_j represents the coefficients associated with the lagged conditional variances σ^2_{i-j}. The GJR-GARCH model allows for capturing the asymmetric response of volatility to shocks, with negative shocks having a potentially larger impact on future volatility compared to positive shocks of the same magnitude. By incorporating the leverage effect, the GJR-GARCH model enhances its ability to capture the dynamics of the data, particularly in cases where the presence of leverage effects is observed. This additional flexibility in modeling the conditional variance improves the model's predictive performance and enables it to provide more accurate forecasts of cryptocurrency price volatility.

4.3 Model Quality Metrics

In order to evaluate the fitted models and predictive performance of the models, several metrics are used. First, H-evidence plot ,Quantile-Quantile Plot and e-statistics are considered for selecting best fitted BART model. Second, well known Information Criteria's such as, Akaike Information Criteria, Bayesian Information Criteria and Hannan-Quinn are considered for selecting best performing GARCH model. At last, the best fitted BART model is compared with the best fitted GARCH model using loss functions such as, Root Mean Squared Error (RMSE), Mean Absolute Deviation (MAD), and Mean Squared Error (MSE).

4.3.1 Graphical Diagnostic Tools

We consider the H-evidence plot, which is a graphical representation used to display posterior intervals for $\hat{s}(x_i)$ in a sorted manner based on their corresponding estimated values. This plot provides a simple way to identify evidence supporting the presence of heteroscedasticity. By examining the intervals, we can quickly assess whether the conditional variance is influenced by predictors, gaining insights into the predictor-dependence of variability.

Additionally, we propose an extension of quantile-quantile (QQ) plots beyond their traditional use in linear regression. Our predictive QQ plot utilizes a sample of observations (x_i, y_i) and calculates the quantiles of y_i based on the predictive distribution $y|x_i$ obtained from our model. In an accurate model, these quantiles should resemble samples drawn from a uniform distribution. We employ QQ plots to compare the calculated quantiles with samples from a uniform distribution, allowing us to visually assess the goodness of fit from a distributional perspective.

Moreover, we introduce the e-statistic, also known as the energy statistic, as an alternative numerical metric for assessing quality. The e-statistic is defined as follows: when we have independent random vectors $U_1, ..., U_{n1}$ drawn from $\mathbb{R}^d$ following the distribution F_1, and $V_1, ..., V_{n2}$ drawn from $\mathbb{R}^d$ following the distribution F_2, where $d > 1$. The test statistic is defined as follows:

$$e = \frac{n_1 n_2}{n_1 + n_2} \left(\frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j-1}^{n_2} ||U_i - V_j|| - \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j-1}^{n_1} ||U_i - U_j|| - \frac{1}{n_2^2} \sum_{i=1}^{n_2} \sum_{j-1}^{n_2} ||V_i - V_j|| \right) \quad (15)$$

where $||.||$ refers to the Euclidean norm. The e-statistic offers a practical and efficient way to test whether two distributions, F_1 and F_2, are equal based solely on samples from these distributions.

4.3.2 Information Criteria

In order to compare the performance of different GARCH models in-sample, common information criteria are employed. Specifically, the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Hannan-Quinn Criterion (HQ) are considered for selecting the best-fitting GARCH model.

The AIC, BIC, and HQ criteria are defined based on the number of parameters in the model (k), the maximum value of the likelihood function ($\hat{L}$), and the number of observations (n). The formulas for these criteria are as follows:

$$AIC = 2k - 2\ln(\hat{L}) \tag{16}$$

$$BIC = \ln(n)k - 2\ln(\hat{L}) \tag{17}$$

$$HQ = -2\log(\hat{L}) + 2k\log(\log(n)) \tag{18}$$

These information criteria provide measures of the trade-off between model fit and model complexity. The lower the value of AIC, BIC, or HQ, the better the model is considered in terms of balancing goodness of fit and parsimony. By considering these information criteria, the selection of the best-fitted GARCH model is based on the model that minimizes AIC, BIC, or HQ. These criteria allow for an objective comparison of different GARCH models and assist in determining the most appropriate model for capturing the volatility dynamics of the cryptocurrency price series.

4.3.3 Hyperparameter Tuning

The selection of hyperparameters can have a large impact on the quality of the ML forecasts, as demonstrated by Hutter et al. (2014). Therefore, it is crucial to select these parameters as appropriately as possibly. Tuning these hyperparameters on the test data can lead to overfitting. Hence, the performance of some hyperparameter selections are evaluated based on the validation set, in our case through 5-fold validation. In the case of the proposed HBART model, it is important to judiciously select the prior hyperparameter κ in specifying the prior mean model. This parameter essentially controls whether BART will tend to fit smoother mean functions with a preference for greater heteroscedasticity (large κ) or fit more complex mean functions with a preference for homoscedasticity (small κ). The e-statistic is employed in this process. In each fold, for each (x, y) pair in the held-out set, we calculate the percentile of y in the predictive distribution of $Y|x$, computed using the remaining folds. Using the `energy` package in R, we compute the e-statistic by comparing these percentiles to the uniform distribution. This procedure is repeated for several plausible values of κ.

4.3.4 Loss Functions

In order to evaluate the forecast accuracy of the models, several loss functions are considered, with three specifically examined in this study. These loss functions provide a quantitative

measure of the discrepancy between the predicted values and the actual values obtained for each model.

The first loss function is the Mean Squared Error (MSE) and its derived metric, the Root Mean Squared Error (RMSE). The MSE is computed as the average of the squared differences between the predicted values and the actual values: $MSE = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$. The RMSE is then obtained by taking the square root of the MSE: $RMSE = \sqrt{MSE}$. The MSE is a widely used loss function, but it has a tendency to be sensitive to outliers due to its quadratic penalty.

The second loss function considered is the Mean Absolute Deviation (MAD). It is calculated as the average of the absolute differences between the predicted values and the actual values: $MAD = \frac{1}{n}\sum|y_i - \hat{y}_i|$. Unlike the mean-based measures, such as the Mean Absolute Error (MAE), the MAD utilizes the median of the absolute differences, providing a measure that is less influenced by extreme values. By using the median, which is less sensitive to outliers, the MAD offers increased robustness compared to mean-based measures.

By employing these loss functions, the study evaluates the forecast accuracy of the models. The MSE and RMSE capture the overall squared differences between predictions and actual values, while the MAD focuses on the median of the absolute differences. These loss functions enable a comprehensive assessment of the models' performance, taking into account both the average and robustness to outliers.

5 Results

In this section, the results of our analysis are presented. First we illustrate the simulated example, and two emperical examples, which gives a introduction on the application of the HBART and BART method. Then our focus is moved to forecasting Bitcoin price volatility, which is main purpose of our research.

5.1 Results: Simulated Example

We simulated 500 observations from the model $y_i = 4x_i^2 + 0.2e^{2x_i}Z_i$, so that $y_i = f(x_i) + s(x_i)Z_i$ with $f(x_i) = 4x_i^2$ and $s(x_i) = 0.2e^{2x_i}$. Each x_i is drawn independently from the uniform distribution on $(0, 1)$, and each Z_i is drawn independently from the standard normal distribution. We then simulated an independent dataset in the exact same manner to serve as out-of-sample data.

The right panel in Figure 1 presents the H-evidence plot, which visually represents the inference for $s(x_i)$, extending to higher dimensional x_i. In this plot, the observations are sorted based on the values of $\hat{s}(x_i)$, with $\hat{s}(x_i)$ displayed on the horizontal axis and posterior intervals for $s(x_i)$ shown on the vertical axis. The plot also includes a solid horizontal line representing the estimate of σ obtained from the homoscedastic version of BART. The H-evidence plot effectively demonstrates the presence of heteroscedasticity as the posterior intervals for $s(x_i)$ are noticeably separated from the horizontal line. This clear separation indicates a significant discovery of heteroscedasticity.

The left panel and centered panel of Figure 1 assesses the fit of our model by looking at QQ-plots based on the predictive distribution obtained from our model. Draws are obtained

from $p(y_i|x_i)$ and then compute the percentile of the observed y_i in these draws. If the model is correct, these percentiles should look like draws from the uniform distribution on the interval $(0, 1)$. We use the QQ-plot to compare these percentiles to draws from the uniform. The (x_i, y_i) pairs are from the test data, so we are evaluating the out-of-sample predictive performance. The left panel of Figure 1 shows the predictive QQ-plot obtained from our heteroscedastic model. We see that our predictive percentiles match the uniform draws very well. In the centered panel it is observed that our predictive draws from our homoscedastic model are failing.

Although the root mean square error (RMSE) values for both HBART and BART models are quite close, with 0.77 for HBART and 0.76 for BART, the e-statistic reveals a significant difference in performance. The e-statistic scores are 0.54 for HBART and 3.50 for BART, indicating that HBART outperforms BART substantially. While the RMSE suggests similar mean prediction performance between the two models, the e-statistic and graphical diagnostics provide strong evidence that HBART is the superior model overall. Therefore, in this example, HBART is selected as the best-performing model.

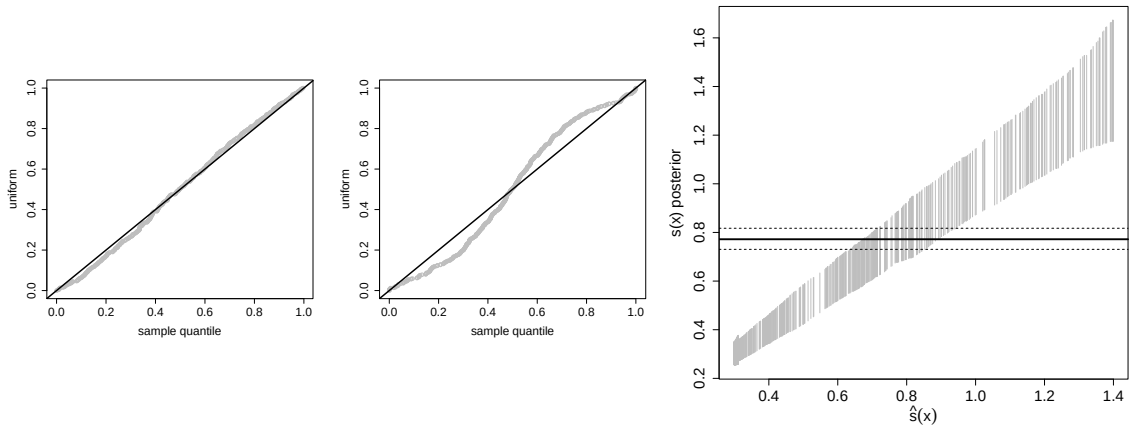

Figure 1: Simulation Example: Left Panel Displays HBART QQ-Plot, Centered Panel Displays BART QQ-Plot, Right Panel Displays H-Evidence Plot

5.2 Results: Emperical Examples

Our focus is turned to the empirical examples, starting with the analysis of alcohol consumption and used car sales. These examples serve as illustrations of the statistical models employed in our study. Finally, we delve into the Bitcoin Price Volatility example, which is the primary focus and objective of our research.

5.2.1 Results: Alcohol Consumption Example

In this specific example, the response variable is restricted to positive values, and there are observations where $y_i = 0$. This means that our model $y_i = f(x_i) + s(x_i)Z$ does not fully account for these aspects of the response variable y_i.

The H-evidence plot in the right panel in Figure 2 clearly shows that the posterior intervals are not seperated from the horizontal line, indicating that there is no significant indication of

heteroscedasticity. This implies that the variability in the response variable across different levels of the predictor is relatively consistent, supporting the assumption of constant variance in the model.

The left and centered panel in Figure 2 provide an assessment of our model's fit by utilizing QQ-plots. Upon examining the QQ-plots for both the HBART and BART models, we can observe that the plotted points do not strictly follow a linear pattern but exhibit similar trends. This suggests that both models share common characteristics in their predictions, even though they may not align perfectly with the expected percentiles. The deviation from a straight line indicates some discrepancies between the model's predictions and the observed data. However, the presence of similar patterns suggests that both models capture certain aspects of the underlying data distribution.

Furthermore, upon evaluating the performance of the homoscedastic and heteroscedastic models, we find that they yield similar results in terms of the root mean squared error (RMSE), with both models having an RMSE of 1.34. However, when considering the e-statistics, we observe slightly different values. The homoscedastic model has an e-statistic of 2.26, while the heteroscedastic model has an e-statistic of 2.5. Based on these metrics, we conclude that the BART model performs better overall and is selected as the best-performing model in this particular example.

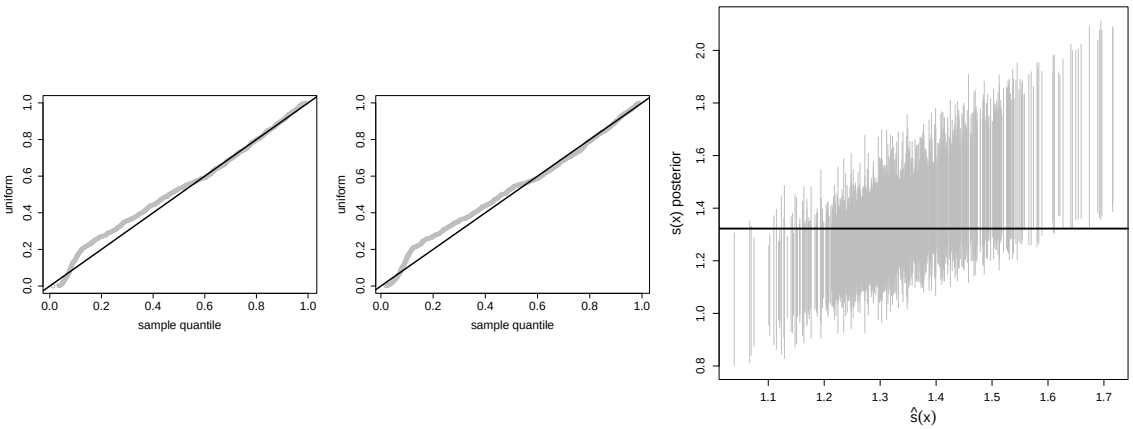

Figure 2: Alcohol Example: Left Panel Displays HBART QQ-Plot, Centered Panel Displays BART QQ-Plot, Right Panel Display H-Evidence Plot

5.2.2 Results: Used Car Sales Example

This example is an extended real example in which we develop a model to predict the price of a used car using car characteristics. Some key relationships between predictor variables can be found in the Appendix A.

As mentioned previously (see Section 4.3.3), a five-fold cross-validation procedure is conducted in this example to determine the appropriate hyperparameters. The results of this cross-validation are presented in Table 1. Based on the average e-statistics, the suggested value for the homoscedastic model is $\kappa = 1$, while for the heteroscedastic model, a cross-validated value

of $\kappa = 0.25$ is recommended. These values are chosen based on achieving the lowest average e-statistics across the cross-validation folds.

Table 1: Average e-statics after performing 5 fold cross validation for BART and HBART

κ	**0.25**	**0.5**	**1**	**2**	**5**	**10**	**20**
BART	0.98	1.13	0.83	1.24	1.40	1.96	2.80
HBART	0.23	0.43	0.27	0.31	0.36	0.78	1.89

The H-evidence plot in Figure 3 is used to assess the presence of heteroscedasticity. A noteworthy observation is that a significant portion of the data falls outside the posterior 90% credible intervals for the standard deviation estimate obtained from the homoscedastic model. This indicates that there are variations in the variability of the data across different levels. Additionally, Figure 3 examines the goodness of fit of our model using qq-plots. It is important to note that the plotted points do not align perfectly along a straight line, but they exhibit similar patterns. This suggests that both models capture similar trends in their predictions, although they deviate slightly from the expected percentiles. The deviation from a straight line indicates some discrepancies between the models' predictions and the observed data. However, the presence of similar patterns suggests that both models capture certain aspects of the underlying data distribution. Furthermore, the comparison of the two models reveals that HBART performs better in terms of predictive accuracy. The RMSE for HBART is 4763.79, whereas for BART it is 4794.79. Additionally, the e-statistic, which serves as a measure of overall model performance, is better for HBART (0.39) compared to BART (0.76). Based on these findings, it can be concluded that the HBART model outperforms the BART model in this specific example.

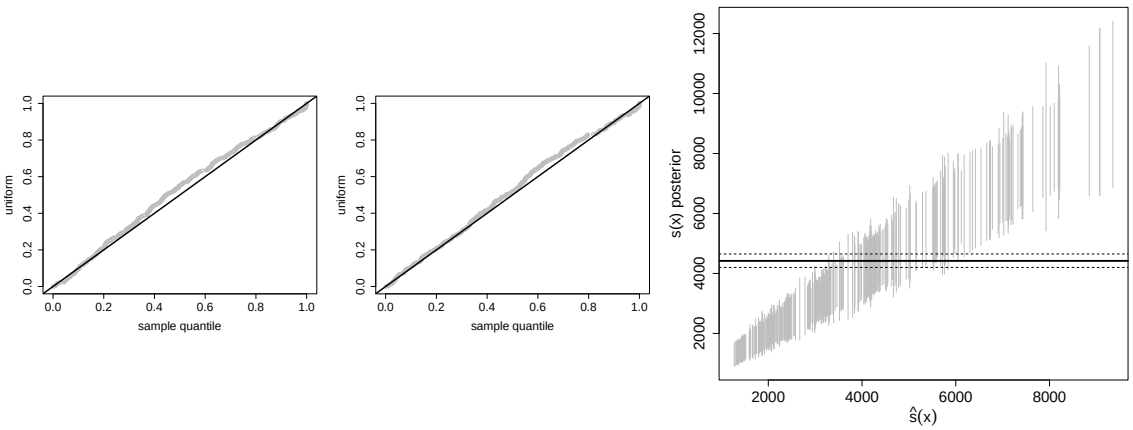

Figure 3: Cars Example: Left Panel Displays HBART QQ-Plot, Centered Panel Displays BART QQ-Plot, Right Panel Displays H-Evidence Plot

5.3 Results: Bitcoin Price Volatility Example

Our attention is now turned to the results of the Bitcoin price volatility analysis. Firstly, we evaluate the performance of the GARCH models during the in-sample period to assess how

well they fit the entire dataset. This evaluation involves comparing information criteria such as AIC, BIC, and HQ, which provide indicators of model fit. To compare the statistical models separately, we utilize diagnostic tools such as QQ-plots and H-evidence plots. These plots allow us to assess the goodness of fit and detect any deviations from expected patterns. Moving on, we assess the forecasting accuracy of the best-performing GARCH and BART models using loss function measures including RMSE (Root Mean Squared Error), MAD (Median Absolute Deviation), and MSE (Mean Squared Error). These measures provide quantitative insights into the predictive performance of the models and allow for a meaningful comparison between them.

5.3.1 Results: In-Sample-Comparison

All GARCH-type models were applied to the in-sample period, which contained 2422 observations. Table 2 presents the AIC, BIC and HQ values for the different models used. It is evident that the models chosen based on AIC, BIC and HQ criteria differ significantly. The standard GARCH model is selected as the best model based on AIC and HQ criterion, indicating that it provides a good balance between model fit and complexity. On the other hand, the GJR-GARCH & sGARCH are chosen as the best models based on the BIC criterion. These results suggest that different criteria may lead to different model selections, highlighting the importance of considering multiple criteria when choosing the most appropriate model for the data. Note, that from Table 2, it is also evident that GARCH and GJR-GARCH have similar performance.

The estimated parameters of the GARCH model and its extension can be found in Table 3. It is noteworthy that the estimated coefficient α_1 of 0.133 in the sGARCH model indicates a moderate level of persistence in Bitcoin price volatility. This suggests that past volatility has a lasting impact on future volatility. Additionally, the estimated coefficient β_1 of 0.865 in the sGARCH model carries important implications for understanding the dynamics of Bitcoin price volatility. It signifies the degree of persistence observed in the volatility of the Bitcoin prices. The estimated coefficient γ_1 of -0.059 in the GJR-GARCH model suggests the presence of a leverage effect in Bitcoin price volatility. This indicates that negative returns in the previous period significantly and persistently influence future volatility, reflecting an asymmetric response of volatility to negative shocks. The positive coefficient $\alpha_1 = 0.154$ in the GJR-GARCH model suggests that it captures the presence of volatility clustering in Bitcoin prices, where periods of high volatility tend to be followed by further periods of high volatility. Comparing the GARCH models considered in the analysis, the sGARCH and GJR-GARCH models exhibit the lowest log likelihood values. The lower log likelihood values indicate that these models more accurately and effectively capture the patterns of volatility compared to the other GARCH models examined.

Moreover, our analysis of the Homoscedastic and Heteroscedastic models reveals that the homoscedastic BART model is also suitable for the data. This finding is supported by the similarity observed in the QQ plots of the homoscedastic and heteroscedastic models, as depicted in the left and centered panel in Figure 4. In addition to the QQ-plots, the right panel in Figure 4 provides an additional check on whether the HBART model effectively captures the variance of Bitcoin returns. In this plot, $\hat{s}(x_i)$ is represented on the horizontal axis, while the vertical axis displays posterior intervals for $s(x_i)$. The solid horizontal line represents the estimate of σ obtained from the heteroscedastic version of BART. Our analysis reveals that the posterior

intervals in our study significantly deviate from the horizontal line, indicating the presence of statistically significant heteroscedasticity. Based on this inference, we select the HBART model as the best-fitting model for the data.

Table 2: Information Criteria GARCH models

Models	**AIC**	**BIC**	**HQ**
GARCH	**-4.139**	**-4.126**	**-4.134**
EGARCH	-4.149	-4.145	-4.154
aPARCH	-4.159	-4.143	-4.153
GJR-GARCH	-4.141	**-4.126**	-4.136

Table 3: Estimated Parameters Of The Fitted GARCH Models

Models	μ	ω	α_1	β_1	δ_1	γ_1	**Log-Lik**
GARCH(1,1)	0.01873	0.000014	0.133394	0.865606	-	-	3777.923
EGARCH(1,1)	0.001751	-0.068167	0.051850	0.989187	-	0.320541	3798.451
aPARCH(1,1)	0.001692	0.000664	0.198645	0.879226	0.886128	-0.188747	3801.388
GJR-GARCH(1,1)	0.001942	0.000011	0.154316	0.874443	-	**-0.059518**	3781.102

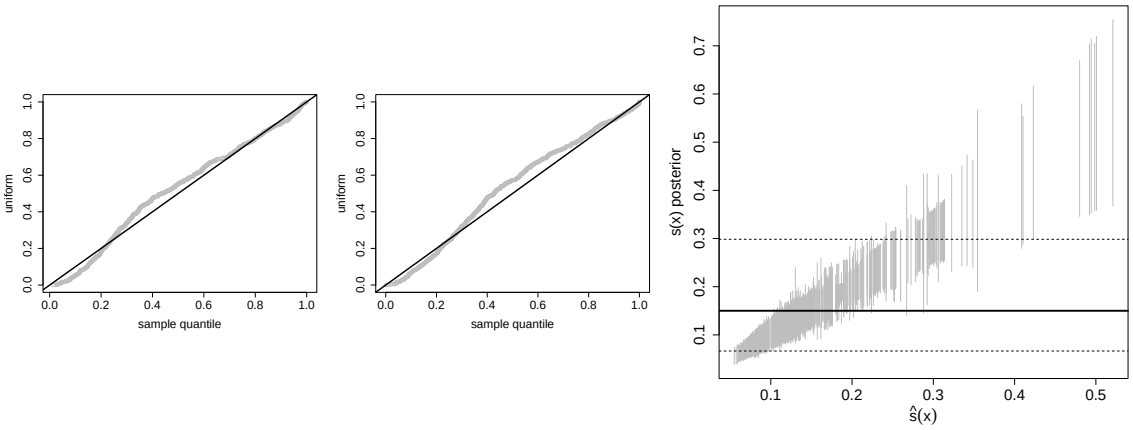

Figure 4: Bitcoin Example: Left Panel: Displays QQ-Plot H-BART, Centered Panel: QQ-Plot BART, Right Panel: Displays H-Evidence Plot

5.4 Results: Forecasting Performance

We have evaluated and compared the predictive performance of the best GARCH models and BART models. Specifically, we focused on the sGARCH and GJR-GARCH models for 1-step ahead forecasting horizon and compared their predictive accuracy with that of the HBART model using loss functions. Table 4 presents the loss functions of the volatility forecasts for the sGARCH and GJR-GARCH models at 1-step ahead forecast horizons, alongside the loss functions of the HBART model. It is important to note that both the sGARCH and GJR-GARCH models outperform the HBART model in terms of predictive accuracy. Notably, the GJR-GARCH model emerges as the superior choice in terms of forecasting performance. One

possible explanation for the GJR-GARCH model's superior performance could be its ability to capture the leverage effect, which likely enhances its predictive capabilities.

Table 4: Loss Functions GARCH vs HBART

Models	**Horizon**	**RMSE**	**MAD**	**MSE**
sGARCH	1-day	0.0456	0.0456	0.0021
GJR-GARCH	1-day	**0.0455**	**0.0455**	**0.0020**
HBART	1-day	0.1736	0.1306	0.0299

6 Conclusion

Accurately predicting cryptocurrency price volatility is of utmost importance for investors and traders, as it allows them to effectively manage risks, make informed investment decisions, and enhance financial outcomes. To address the question of achieving accurate volatility predictions, we examine two statistical models, namely HBART and BART, as well as four econometric models: sGARCH, EGARCH, aPARCH, and GJR-GARCH. Using a real-world dataset obtained from Yahoo Finance, covering the period from January 1, 2014, to January 1, 2023, we apply these models to forecast Bitcoin price volatility. Our analysis focuses on assessing the ability of these models to capture heteroscedasticity in high-dimensional data. In addition to the empirical analysis, we also present a simulated example to illustrate the effectiveness of the statistical models in capturing heteroscedasticity. By considering these examples, we aim to provide insights into the performance and suitability of the statistical models in predicting cryptocurrency price volatility.

Our analysis based on information criteria, suggests that the sGARCH and GJR-GARCH models provide the best fits for the data. Specifically, the sGARCH model outperforms the GJR-GARCH model based on AIC and HQ, while their performance is comparable according to BIC. These findings highlight the importance of using different criteria for model selection, as they can lead to different outcomes. In terms of graphical diagnostic metrics, the HBART model emerges as the preferred choice over the BART model, indicating its superior fit.

In terms of predictive accuracy, our analysis indicates that the GARCH models outperform the statistical-based models, particularly when evaluated using loss functions. Among the top performers, the GJR-GARCH model shows superior predictive accuracy. However, it is worth noting that the sGARCH model performs similarly, with minimal differences in loss function values. Based on our analysis, it can be concluded that the HBART model does not outperform traditional econometric methods.

Despite the valuable insights gained from this research, there are limitations that suggest avenues for future investigation. Firstly, our study utilized a relatively small number of observations in the time dimension. To incorporate additional effects, it would be interesting to apply our methods to high frequency datasets. Secondly, due to time and resource constraints, we did not incorporate macroeconomic variables and sentiment data. Including these factors in future research could enhance the analysis.

References

Aysan, A. F., Demir, E., Gozgor, G. & Lau, C. K. M. (2019). Effects of the geopolitical risks on bitcoin returns and volatility. *Research in International Business and Finance*, *47*, 511–518.

Beneki, C., Koulis, A., Kyriazis, N. A. & Papadamou, S. (2019). Investigating volatility transmission and hedging properties between bitcoin and ethereum. *Research in International Business and Finance*, *48*, 219–227.

Caporale, G. M. & Zekokh, T. (2019). Modelling volatility of cryptocurrencies using markov-switching garch models. *Research in International Business and Finance*, *48*, 143–155.

Chipman, H. A., George, E. I. & McCulloch, R. E. (2010). Bart: Bayesian additive regression trees.

Heij, C., de Boer, P., Kloek, P. & Franses, P. (2004). *Econometric methods with applications in business and economics*. Athenaeum Uitgeverij.

Hutter, F., Hoos, H. & Leyton-Brown, K. (2014). An efficient approach for assessing hyperparameter importance. In *International conference on machine learning* (pp. 754–762).

Katsiampa, P. (2019). An empirical investigation of volatility dynamics in the cryptocurrency market. *Research in International Business and Finance*, *50*, 322–335.

Lyócsa, Š. & Molnár, P. (2016). Volatility forecasting of strategically linked commodity etfs: gold-silver. *Quantitative Finance*, *16*(12), 1809–1822.

Ma, D. & Tanizaki, H. (2019). The day-of-the-week effect on bitcoin return and volatility. *Research in International Business and Finance*, *49*, 127–136.

McCulloch, R., Pratola, M. T. & Chipman, H. (2019). rbart: Bayesian trees for conditional mean and variance [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=rbart (R package version 0.9)

Mostafa, F., Saha, P., Islam, M. R. & Nguyen, N. (2021). Gjr-garch volatility modeling under nig and ann for predicting top cryptocurrencies. *Journal of Risk and Financial Management*, *14*(9), 421.

Naimy, V. Y. & Hayek, M. R. (2018). Modelling and predicting the bitcoin volatility using garch models. *International Journal of Mathematical Modelling and Numerical Optimisation*, *8*(3), 197–215.

Pratola, M. T., Chipman, H. A., George, E. I. & McCulloch, R. E. (2020). Heteroscedastic bart via multiplicative regression trees. *Journal of Computational and Graphical Statistics*, *29*(2), 405–417.

Wang, P., Li, X., Shen, D. & Zhang, W. (2020). How does economic policy uncertainty affect the bitcoin market? *Research in International Business and Finance*, *53*, 101234.

A Data

A.1 Simulated Example

Figure 5 provides a clear visual representation of the HBART estimates for the given data. It includes the true mean function and the error standard deviation, represented by dashed lines, as well as the HBART estimates of these curves, depicted by solid lines.

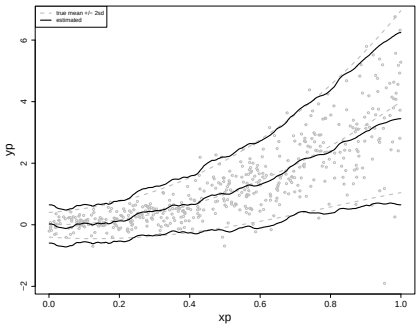

Figure 5: Simulated Example:Out-Of-Sample Simulated Data

A.2 Cars Example

In this section, we provide a comprehensive description of the variables used in the Used Cars Example dataset, as presented in Table 5. Additionally, we analyze the relationship between log(price) and year/mileage in Figure 6 and Figure 7. By taking the logarithm of the price variable, we address potential issues related to heteroscedasticity and non-linear relationships in the data. Plotting log(price) against mileage and year allows us to uncover any underlying patterns or trends that may not be apparent when examining the variables in their original scales. This transformation aids in understanding the functional form of the relationship and enhances the accuracy of modeling and interpretation of the results.

Table 5: Cars Example: Data types, levels and descriptions of the Original Variables

Variable	**Data type**	**Level**	**Description**
price	Continous	-	Price car
mileage	Continous	-	Number of miles
year	Continous	-	production year
trim	Categorical	trim.430, trim.500, trim.550, trim.other	Car version
color	Categorical	Black, Silver, White, Other	Car color
displacement	Categorical	displacement.46, displacement.55, displacement.other	Total engine vol
isOneOwner	Categorical	True, False	if prev owner = 1

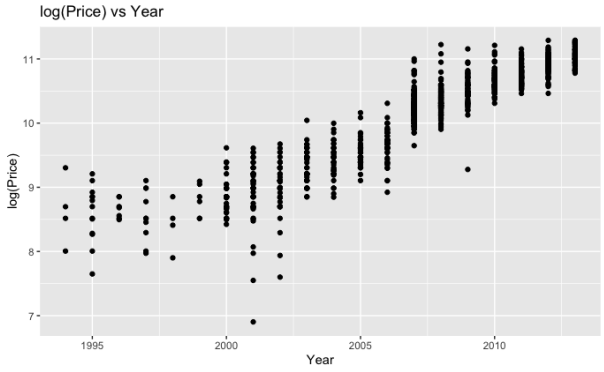

Figure 6: Cars Example: LogPrice Vs Year

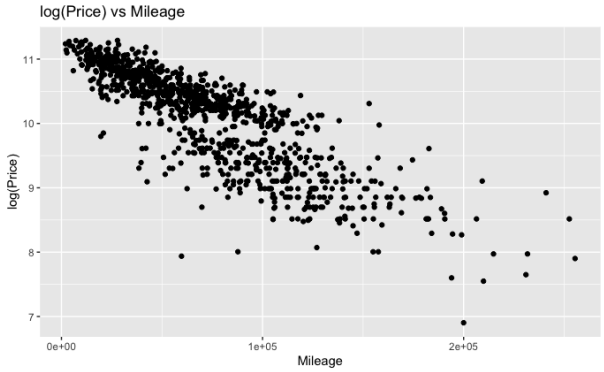

Figure 7: Cars: LogPrice Vs Mileage

A.3 Bitcoin Example

Table 6 represents the summary statistics and test results of the daily log returns of Bitcoin from yahoo-finance. We can see that the data is left skewed and has excess kurtosis, which indicates non-normality. This is confirmed by the Jarque–Bera test. The augmented Dickey–Fuller test statistic rejects the null hypothesis of a unit root at 5% significance level, meaning that returns can be viewed as stationary. This can also be seen in Figure 8, where the log returns have been plotted. From this plot we also see evidence of volatility clustering, which violates stationary.

The histogram 9 of the log-returns displays a distribution with fat tails, indicating the presence of extreme outcomes. This observation aligns with the substantial price fluctuations observed in Bitcoin. The kurtosis value in the summary statistics further confirms this characteristic, indicating a higher probability of extreme events. Additionally, the summary statistics reveal a left-skewed distribution, where the mean is lower than the median and negative skewness is present. These findings suggest that the dataset is slightly more inclined towards extreme negative returns compared to positive returns.

Table 6: Summary Statistics Bitcoin Log-Returns

Statistics	**Mean**	**Max**	**Skewness**	**Kurtosis**	**JB-Test**	**Dickey-Fuller**	**P-Value**
Value	0.0012	0.0017	-0.7748	10.8522	2340.7200	-13.4130	0.00

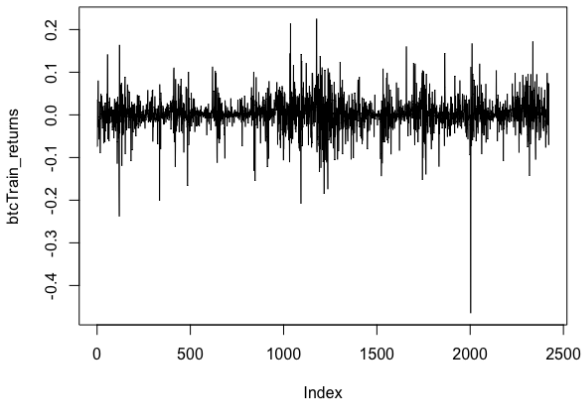

Figure 8: Bitcoin Example: BTC Returns In-sample Period

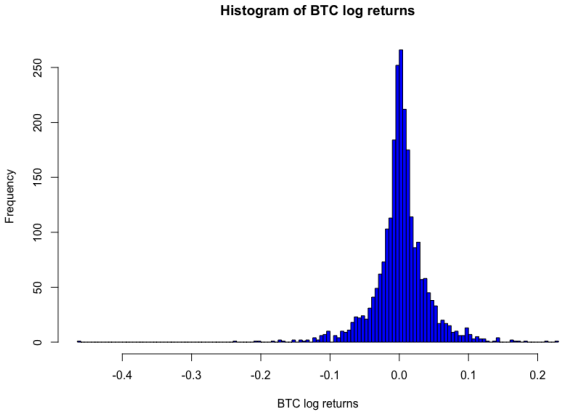

Figure 9: Histogram of BTC log Returns

B Results

B.1 Simulated Example

Figure 10 illustrates the uncertainty associated with our estimates of functions f and s. The left panel shows the inference for function f, while the right panel focuses on function s. The dashed line represents the true function, the solid line represents the estimated function obtained through averaging MCMC draws, and the dot-dash line represents the point-wise 95% posterior intervals for each f(xi) (in the left panel) and s(xi) (in the right panel). These intervals are estimated using quantiles from the MCMC draws of f(xi) and s(xi), where the xi values are derived from the test data.

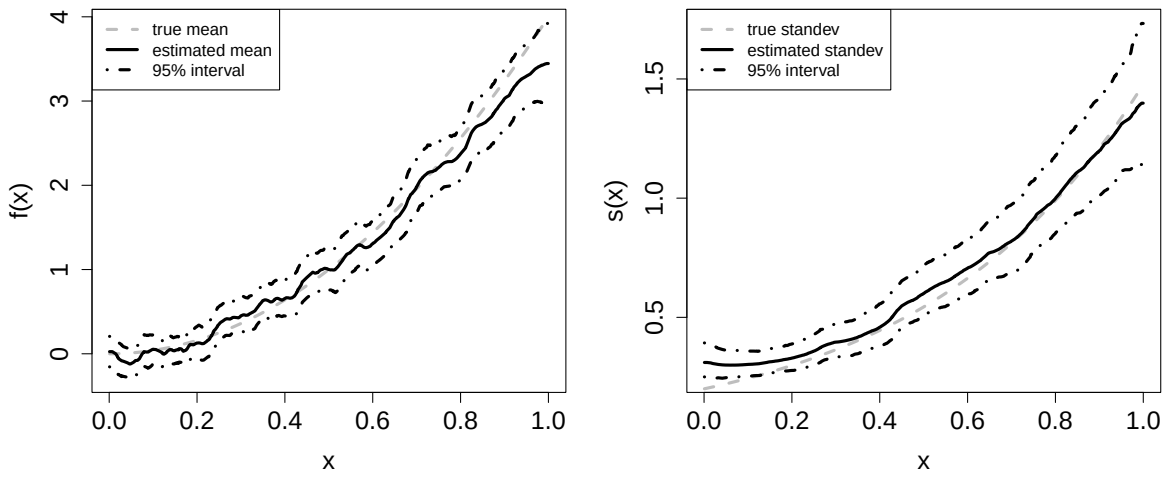

Figure 10: Simulated Data with their true, estimated and 95% posterior intervals

B.2 Cars Example

It is worth noting that there is collinearity observed between the categorical variables and the predictor variables, as shown in Figures 11 and 12. Figure 12 indicates a non-linear relationship between mileage and price, where higher mileage is associated with lower prices. On the other hand, Figure 11 reveals a non-linear relationship between year and price, with higher year (younger car) corresponding to higher prices. Despite the significant contribution of the "trim.550" variable to the increase in prices from 2007 onwards, it does not fully explain the observed phenomenon, as other cars with different trim levels ("trim.other") are distributed across the entire range of years in the dataset. Additionally, there appears to be a noticeable change in price variability starting from 2007, suggesting the presence of heteroscedasticity, which indicates that factors beyond the average values of the predictors may influence the price variability.

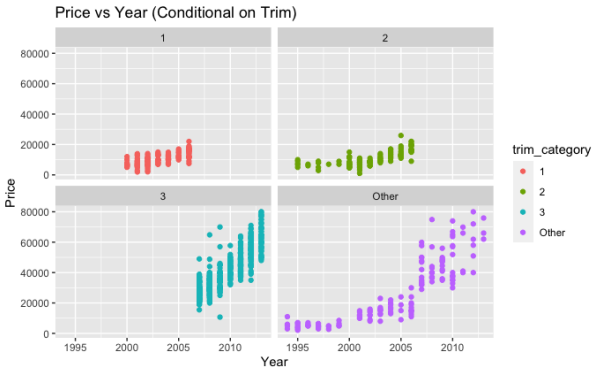

Figure 11: Price vs Year for trim levels

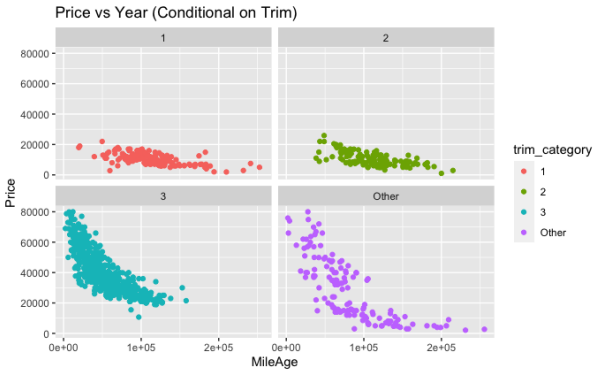

Figure 12: Price vs Mileage for trim levels

B.3 Bitcoin Example

The fitted variance of the GJR-GARCH and (Standard) GARCH models, as shown in Figure 13 and Figure 14 respectively, does not accurately capture the significant spikes in volatility observed in the realized variance. This is also the case for the E-GARCH and aPARCH models, as seen in Figure 15 and Figure 16. This is expected because GARCH models have a tendency to smooth out variance, leading to a less precise depiction of sharp volatility jumps.

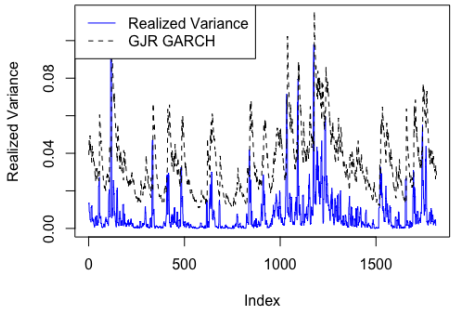

Figure 13: GJR-GARCH

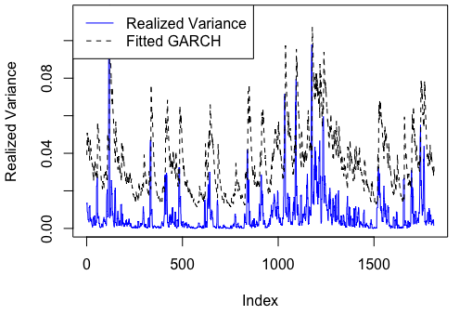

Figure 14: sGARCH

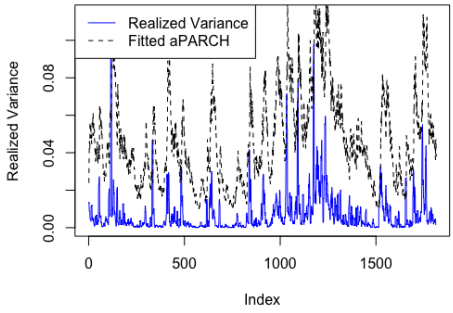

Figure 15: aPARCH

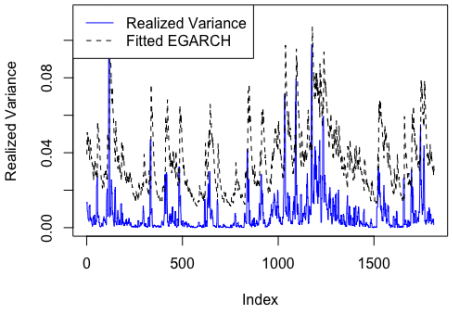

Figure 16: E-GARCH

C Programming code

In the *thesis.code* folder, there are four different R files, four txt files, a "csv" file and a README.txt file, which contains the programming codes and descriptions of our analysis performed in R. It is structured in the following way:

- **Simulated.R**: The R programming code for the Simulated Example
- **Simulated.txt:** The description of the code and analysis performed in the Simulated Example
- **Alcohol.R:** The R programming code for the Alcohol Consumption Example

- **Alcohol.txt:** The description of the code and the analysis performed in the Alcohol Example
- **Alcohol.csv:** The data set used in our Alcohol Consumption Example
- **Cars.R:** The R programming code for the Used Cars Sales Example
- **Cars.txt** The description of the code and analysis for the Used Cars Sale Example
- **Bitcoin.R:** The R programming code for the Bitcoin Price Volatility Prediction Example
- **Bitcoin.txt:** The description for the code and analysis for the Bitcoin Example
- **README.txt:** A detailed description of the aim, directory structure and usage of this project

Simulated.R

From the *thesis.code* folder, we first consider the Simualted example from Subsection 5.1. The code for this example is considered within the **Simulated.R** file.

Input:

We run the prepare data part of the code in the Simulated.R file to obtain the in-sample data and out-of-sample data, which is used a input for our analysis.

Output:

- The RMSE of the BART model
- The RMSE of the HBART model
- The *e-statistic* of the BART model
- The *e-statistic* of the HBART model
- The H-Evidence Plot
- The Predictive QQ-Plot
- The Data Plot
- Plot of the True and Point Estimates
- Plot of the mean and standard deviation intervals

Dependencies:

- rbart package: for fitting and predicting BART and HBART model
- energy package: for calculating the e-statistics
- ggplot: for making the plots

Usage:

- First run with loading the packages and library's
- Second, run the "start with preparing your in-sample and out-of-sample data"
- Third, run the part of fitting and prediciting the HBART and BART model
- At last, run the part which will give the desired outputs such as: RMSE, e-statistics, QQ-plots, H-evidence plots etc

Alcohol.R

From the *thesis.code* folder, the Alcohol Consumption example from Subsubsection 5.2.1 is considered. The code for this example is considered within the **Alcohol.R** file.

Input:

In this part of the analysis, we first import the *alcohol.csv* from the *thesis.code* folder in the R working space. Furthermore, we run the make *" Load Data and Make training/test sets"* part of the code in the **Alcohol.R** file to obtain the in-sample and out-of-sample data for our analysis.

Output:

- The RMSE of the BART model
- The RMSE of the HBART model
- The *e-statistic* of the BART model
- The *e-statistic* of the HBART model
- The H-Evidence Plot
- The Predictive QQ-Plot

Dependencies

- rbart package: for fitting and predicting BART and HBART model
- energy package: for calculating the e-statistics
- ggplot: for making the plots

Usage

- First run with loading the packages and library's
- Second, run the "start with preparing your in-sample and out-of-sample data"
- Third, run the part of fitting and prediciting the HBART and BART model

- At last, run the part which will give the desired outputs such as: RMSE, e-statistics, QQ-plots, H-evidence plots etc

Cars.R

From the *thesis.code* folder, we now consider the Used Cars Sales example from Subsubsection 5.2.2. The code for this example is considered within the **Cars.R** file.

Input:

In this part of the analysis, we first run the *"Load Data"* part of the code and therafter we run the *"Prepare the data for fitting the BART and HBART"* part of the code, inorder to obtain the in-sample and out-of-sample data for our analysis

Output:

- The (MileAge vs Price | Trim) plot
- The log transformation of price vs mileage plot
- The Average *e-statistics* of the BART model, based on 5 fold crass validation technique
- The Average *e-statistics* of the HBART model, based on 5 fold crass validation technique
- The RMSE of the BART model
- The RMSE of the HBART model
- The *e-statistic* of the BART model
- The *e-statistic* of the HBART model
- The H-Evidence Plot
- The Predictive QQ-Plot

Dependencies:

- rbart package: for fitting and predicting BART and HBART model
- energy package: for calculating the e-statistics
- ggplot: for making the plots

Usage

- First run with loading the packages and library's
- Second, run the "start with preparing your in-sample and out-of-sample data"
- Third, run the cross validation part for selecting hyperparameters

- fourth, run the part of fitting and prediciting the HBART and BART model
- At last, run the part which will give the desired outputs such as: RMSE, e-statistics, QQ-plots, H-evidence plots etc

Bitcoin.R

From the *thesis.code* folder, the Bitcoin Price Volatility Prediction Example from Subsubsection 5.3 is considered at last. The code for this example is considered within the **Cars.R** file. This part of the analysis consists of two parts. First, the HBART and BART approach is considered. Second the GARCH approach is condisered.

Input1:

First, the *"Prepare data for HBART and BART* part of the code is considered. This part of the code , imports the Bitcoin data from yahoo finance on the specified period. Furthermore, the data is split into training and testing data.

Output1:

- The RMSE of the BART model
- The RMSE of the HBART model
- The MADof the BART model
- The MAD of the HBART model
- The MSE of the BART model
- The MSE of the HBART model
- The *e-statistic* of the BART model
- The *e-statistic* of the HBART model
- The Predictive QQ-Plot
- The H-Evidence Plot

Input2:

Second, the *"Load and Prepare data for the GARCH models "* part is considered. This part of the code, prepares data for the GARCH analysis. Furthermore, the data is split into training and testing data for this part of the analysis.

Output2:

- The histogram of Bitcoin Returns
- The Skewness of the Bitcoinn Returns
- The Kurtosis of the Bitcoin Returns
- The Summary of the Bitcoin Returns data, including the mean, maximum and minimum value.
- Print the Fit of the sGARCH(1,1) model
- The fitted variance vs realized variance plot of the sGARCH(1,1) model
- The MSE of the sGARCH(1,1) model
- The MAD of the sGARCH(1,,1) model
- Print the Fit of the E-GARCH(1,1) model
- The fitted variance vs realized variance plot of the E-GARCH(1,1) model
- The MSE of the E-GARCH(1,1) model
- The MAD of the E-GARCH(1,,1) model
- Print the Fit of the aPARCH(1,1) model
- The fitted variance vs realized variance plot of the aPARCH(1,1) model
- The MSE of the aPARCH(1,1) model
- The MAD of the aPARCH(1,,1) model
- Print the Fit of the GJR-GARCH(1,1) model
- The fitted variance vs realized variance plot of the GJR-GARCH(1,1) model
- The MSE of the GJR-GARCH(1,1) model
- The MAD of the GJR-GARCH(1,,1) model

Dependencies

- rbart package: for fitting and predicting BART and HBART model
- energy package: for calculating the e-statistics
- ggplot: for making the plots

Usage

- First run with loading the packages and library's
- Second, run the "start with preparing your in-sample and out-of-sample data" for the statistical models
- Third, run the part of fitting and prediciting the HBART and BART model
- Fourth, run the part which will give the desired outputs such as: RMSE, MSE and MAD, e-statistics, QQ-plots, H-evidence plots.
- Fifth, run the "start preparing your in-sample and out-of-sample data" for the econometric models
- Third, run the part of fitting and prediciting the GARCH models
- Fourth, run the part which will give the desired outputs such as: Information Criteria and Loss functions values.